



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería en Electrónica Industrial y Automática

ANÁLISIS Y CÁLCULO DE CIRCUITOS CON AMPLIFICADORES OPERACIONALES

Autor:

González Ramos, Sergio

Tutor:

**Buey Cuesta, José Julio
Departamento de Tecnología
Electrónica**

Valladolid, mayo de 2019.

RESUMEN

Este trabajo tiene su fundamento en desarrollar una herramienta con la que podamos obtener, de manera rápida y sencilla, la representación gráfica y los parámetros más interesantes de circuitos con amplificadores operacionales. Todos estos circuitos se obtienen de la asignatura “*Electrónica Analógica*” de 3º del grado en *Ingeniería en Electrónica Industrial y Automática*.

El amplificador operacional y sus aplicaciones en circuitos se explicará al inicio de la memoria de manera escueta, pues el trabajo se centrará en explicar la construcción del programa en todos sus niveles. Este programa lo llevaremos a cabo mediante el lenguaje de programación *Visual Basic Application* de *Microsoft Office Excel*.

PALABRAS CLAVE

Amplificador operacional, circuito, aplicación, programa, Excel, VBA.

CONTENIDO

1. Introducción.....	1
2. Objetivos	3
3. Análisis de los amplificadores operacionales	5
3.1. Estructura del amplificador operacional	5
3.2. Notación y terminología de un amplificador operacional.....	6
3.3. Modos de funcionamiento del amplificador operacional.....	7
3.4. Encapsulado del amplificador operacional.....	9
3.5. Diferencias entre un A.O. ideal y un A.O. real	10
3.6. Aplicaciones lineales y no lineales de los A.O.....	11
4. Programa para la resolución de circuitos con A.O.....	13
4.1. Descripción del software empleado	13
4.2. Estructura interna del programa.....	15
4.2.1. Formularios del programa	17
4.2.1.1. CircCom.....	18
4.2.1.2. CircHis.....	19
4.2.1.3. CircInt.....	19
4.2.1.4. CircLim	20
4.2.1.5. CircRectPre	20
4.2.1.6. CircRes.....	21
4.2.1.7. CircSum	22
4.2.1.8. CompCeroSat	22
4.2.1.9. CompSatSat.....	25
4.2.1.10. CompSchmitt.....	25
4.2.1.11. CompSimple	26
4.2.1.12. ConvCorrTen	27
4.2.1.13. ConvTenCorr	28
4.2.1.14. DetectPico	29
4.2.1.15. Diferenciador.....	30
4.2.1.16. FuenteCorriente	30
4.2.1.17. FuenteTension.....	32

4.2.1.18.	Histeresis1.....	32
4.2.1.19.	Histeresis2.....	33
4.2.1.20.	IntegradorInversor.....	34
4.2.1.21.	IntegradorNoInversor.....	36
4.2.1.22.	IntegradorRestador.....	37
4.2.1.23.	IntegradorSumador.....	39
4.2.1.24.	Inversor.....	40
4.2.1.25.	LimitadorDiodoNormal.....	40
4.2.1.26.	LimitadorDiodoNormalDiodoZener.....	41
4.2.1.27.	LimitadorDosZener.....	42
4.2.1.28.	NoInversor.....	43
4.2.1.29.	OndaCuadrada.....	44
4.2.1.30.	OndaTriangular.....	45
4.2.1.31.	RectMediaOnda.....	45
4.2.1.32.	RectOndaCompleta.....	46
4.2.1.33.	RestadorAltImp.....	47
4.2.1.34.	RestadorSimple.....	48
4.2.1.35.	SumadorInversor.....	49
4.2.1.36.	SumadorNoInversor.....	49
4.2.1.37.	ValorMedioValorEficaz.....	50
4.2.1.38.	GraficaCosSen.....	51
4.2.1.39.	GraficaDetectPico.....	53
4.2.1.40.	GraficaDif.....	53
4.2.1.41.	GraficaHisteresis.....	54
4.2.1.42.	GraficaSchmitt.....	55
4.2.1.43.	GraficaVot.....	56
4.2.1.44.	GraficaVot_2.....	57
4.2.1.45.	GraficaVoVi.....	58
4.2.1.46.	GraficaVoVi_2.....	59
4.2.1.47.	Informacion.....	60
4.2.1.48.	Informacion1.....	61
4.2.1.49.	Informacion2.....	62
4.2.1.50.	Informacion3.....	63
4.2.1.51.	Informacion4.....	64

4.2.1.52.	Informacion5	65
4.2.1.53.	InformacionDiferencial	66
4.2.1.54.	InformacionInt1.....	67
4.2.1.55.	InformacionInt2.....	68
4.2.1.56.	InformacionOndas.....	69
4.2.1.57.	TeoriaCompCeroSat1.....	71
4.2.1.58.	TeoriaCompCeroSat2.....	71
4.2.1.59.	TeoriaCompSatSat	71
4.2.1.60.	TeoriaCompSchmitt	72
4.2.1.61.	TeoriaCompSimple.....	72
4.2.1.62.	TeoriaConvCorrTen.....	73
4.2.1.63.	TeoriaConvTenCorr.....	74
4.2.1.64.	TeoriaDetectPico	74
4.2.1.65.	TeoriaDiferenciador	75
4.2.1.66.	TeoriaFuenteCorriente.....	75
4.2.1.67.	TeoriaFuenteTension	76
4.2.1.68.	TeoriaHisteresis	77
4.2.1.69.	TeoriaHisteresis1	77
4.2.1.70.	TeoriaHisteresis2	78
4.2.1.71.	TeoriaIntInv.....	78
4.2.1.72.	TeoriaIntNoInv	79
4.2.1.73.	TeoriaIntRes	79
4.2.1.74.	TeoriaIntSum	80
4.2.1.75.	TeoriaInv	80
4.2.1.76.	TeoriaLimDiodoNormalDiodoZener1	81
4.2.1.77.	TeoriaLimDiodoNormalDiodoZener2	81
4.2.1.78.	TeoriaLimDosZener.....	81
4.2.1.79.	TeoriaLimSimple1	82
4.2.1.80.	TeoriaLimSimple2	83
4.2.1.81.	TeoriaMediaOnda.....	83
4.2.1.82.	TeoriaMedioEficaz.....	83
4.2.1.83.	TeoriaNoInv	84
4.2.1.84.	TeoriaOndaCompleta1.....	85
4.2.1.85.	TeoriaOndaCompleta2.....	85

4.2.1.86.	TeoriaOndaCuadrada.....	85
4.2.1.87.	TeoriaOndaTriangular	86
4.2.1.88.	TeoriaRestAltImp.....	87
4.2.1.89.	TeoriaRestSimple.....	87
4.2.1.90.	TeoriaSumInv	88
4.2.1.91.	TeoriaSumNoInv.....	88
4.2.2.	Módulos del programa.....	89
4.2.2.1.	Circuitos.....	90
4.2.2.2.	ModCircComp.....	90
4.2.2.3.	ModCircLim.....	92
4.2.2.4.	ModCircRectPre	93
4.2.2.5.	ModConvTenCorr y ModConvCorrTen.....	93
4.2.2.6.	ModDetectPico.....	94
4.2.2.7.	ModDiferenciador	95
4.2.2.8.	ModFuenteCorriente y ModFuenteTension.....	95
4.2.2.9.	ModHisteresis	95
4.2.2.10.	ModIntInv y ModIntNoInv.....	96
4.2.2.11.	ModIntRes y ModIntSum	97
4.2.2.12.	ModInversor y ModNoInversor	97
4.2.2.13.	ModOndaCua.....	97
4.2.2.14.	ModOndaTri.....	98
4.2.2.15.	ModRestador y ModSumador	98
4.3.	Funcionamiento del programa.....	99
5.	Conclusiones	105
6.	Referencias bibliográficas.....	107
7.	Anexos	109

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Estructura interna del Amplificador Operacional $\mu A741$	5
Ilustración 2. Circuito eléctrico equivalente del Amplificador Operacional.....	6
Ilustración 3. Representación del Amplificador Operacional.....	7
Ilustración 4. Característica de transferencia en lazo abierto del Amplificador Operacional	8
Ilustración 5. Encapsulado del Amplificador Operacional $\mu A741$	9
Ilustración 6. Tabla comparativa de un A.O. ideal y un A.O. real	10
Ilustración 7. Activación de la pestaña Programador.....	14
Ilustración 8. Acceso a Visual Basic	14
Ilustración 9. Ventana principal de Visual Basic.....	15
Ilustración 10. Formularios del programa.....	16
Ilustración 11. Módulos del programa	17
Ilustración 12. Formulario CircCom	18
Ilustración 13. Formulario CircInt	19
Ilustración 14. Formulario CircLim.....	20
Ilustración 15. Formulario CircRectPre	21
Ilustración 16. Formulario CircRes	21
Ilustración 17. Formulario CircSum	22
Ilustración 18. Formulario CompCeroSat Ejemplo 1	24
Ilustración 19. Formulario CompCeroSat Ejemplo 2	24
Ilustración 20. Formulario CompSatSat	25
Ilustración 21. Formulario CompSchmitt	26
Ilustración 22. Formulario CompSimple.....	27
Ilustración 23. Formulario ConvCorrTen Ejemplo 1.....	27
Ilustración 24. Formulario ConvCorrTen Ejemplo 2.....	28
Ilustración 25. Formulario ConvTenCorr.....	28
Ilustración 26. Formulario DetectPico	29
Ilustración 27. Formulario Diferenciador	30
Ilustración 28. Formulario FuenteCorriente Ejemplo 1	31
Ilustración 29. Formulario FuenteCorriente Ejemplo 2	31
Ilustración 30. Formulario FuenteTension	32
Ilustración 31. Formulario Histeresis1	33
Ilustración 32. Formulario Histeresis2 Ejemplo 1	34

Ilustración 33. Formulario Histeresis2 Ejemplo 2	34
Ilustración 34. Formulario IntegradorInversor Tipo1.....	35
Ilustración 35. Formulario IntegradorInversor Tipo2.....	36
Ilustración 36. Formulario IntegradorNoInversor Tipo1	37
Ilustración 37. Formulario IntegradorNoInversor Tipo2	37
Ilustración 38. Formulario IntegradorRestador Tipo1	38
Ilustración 39. Formulario IntegradorRestador Tipo2	38
Ilustración 40. Formulario IntegradorSumador Tipo1	39
Ilustración 41. Formulario IntegradorSumador Tipo2	39
Ilustración 42. Formulario Inversor.....	40
Ilustración 43. Formulario LimitadorDiodoNormal Ejemplo 1	41
Ilustración 44. Formulario LimitadorDiodoNormal Ejemplo 2	41
Ilustración 45. Formulario LimitadorDiodoNormalDiodoZener Ejemplo 1.....	42
Ilustración 46. Formulario LimitadorDiodoNormalDiodoZener Ejemplo 2.....	42
Ilustración 47. Formulario LimitadorDosZener	43
Ilustración 48. Formulario NoInversor.....	44
Ilustración 49. Formulario OndaCuadrada.....	44
Ilustración 50. Formulario OndaTriangular	45
Ilustración 51. Formulario RectMediaOnda	46
Ilustración 52. Formulario RectOndaCompleta Ejemplo 1.....	47
Ilustración 53. Formulario RectOndaCompleta Ejemplo 2.....	47
Ilustración 54. Formulario RestadorAltImp	48
Ilustración 55. Formulario RestadorSimple	48
Ilustración 56. Formulario SumadorInversor	49
Ilustración 57. Formulario SumadorNoInversor.....	50
Ilustración 58. Formulario ValorMedioValorEficaz.....	51
Ilustración 59. Formulario GraficaCosSen	52
Ilustración 60. Formulario GraficaDetectPico	53
Ilustración 61. Formulario GraficaDif	54
Ilustración 62. Formulario GraficaHisteresis	55
Ilustración 63. Formulario GraficaSchmitt	56
Ilustración 64. Formulario GraficaVot.....	57
Ilustración 65. Formulario GraficaVot_2	58
Ilustración 66. Formulario GraficaVoVi.....	59
Ilustración 67. Formulario GraficaVoVi_2	60
Ilustración 68. Formulario Informacion	61

Ilustración 69. Formulario Informacion1.....	62
Ilustración 70. Formulario Informacion2.....	63
Ilustración 71. Formulario Informacion3.....	64
Ilustración 72. Formulario Informacion4.....	65
Ilustración 73. Formulario Informacion5.....	66
Ilustración 74. Formulario InformacionDiferencial.....	67
Ilustración 75. Formulario InformacionInt1.....	68
Ilustración 76. Formulario InformacionInt2.....	69
Ilustración 77. Formulario InformacionOndas.....	70
Ilustración 78. Formulario TeoriaCompCeroSat1.....	71
Ilustración 79. Formulario TeoriaCompCeroSat2.....	71
Ilustración 80. Formulario TeoriaCompSatSat.....	72
Ilustración 81. Formulario TeoriaCompSchmitt.....	72
Ilustración 82. Formulario TeoriaCompSimple.....	73
Ilustración 83. Formulario TeoriaConvCorrTen.....	73
Ilustración 84. Formulario TeoriaConvTenCorr.....	74
Ilustración 85. Formulario TeoriaDetectPico.....	74
Ilustración 86. Formulario TeoriaDiferenciador.....	75
Ilustración 87. Formulario TeoriaFuenteCorriente.....	76
Ilustración 88. Formulario TeoriaFuenteTension.....	76
Ilustración 89. Formulario TeoriaHisteresis.....	77
Ilustración 90. Formulario TeoriaHisteresis1.....	77
Ilustración 91. Formulario TeoriaHisteresis2.....	78
Ilustración 92. Formulario TeoriaIntInv.....	78
Ilustración 93. Formulario TeoriaIntNoInv.....	79
Ilustración 94. Formulario TeoriaIntRes.....	79
Ilustración 95. Formulario TeoriaIntSum.....	80
Ilustración 96. Formulario TeoriaInv.....	80
Ilustración 97. Formulario TeoriaLimDiodoNormalDiodoZener1.....	81
Ilustración 98. Formulario TeoriaLimDiodoNormalDiodoZener2.....	81
Ilustración 99. Formulario TeoriaLimDosZener Página 1.....	82
Ilustración 100. Formulario TeoriaLimDosZener Página 2.....	82
Ilustración 101. Formulario TeoriaLimSimple1.....	82
Ilustración 102. Formulario TeoriaLimSimple2.....	83
Ilustración 103. Formulario TeoriaMediaOnda.....	83
Ilustración 104. Formulario TeoriaMedioEficaz.....	84

Ilustración 105. Formulario TeoriaNoInv.....	84
Ilustración 106. Formulario TeoriaOndaCompleta1.....	85
Ilustración 107. Formulario TeoriaOndaCompleta2.....	85
Ilustración 108. Formulario TeoriaOndaCuadrada.....	86
Ilustración 109. Formulario TeoriaOndaTriangular Página 1.....	86
Ilustración 110. Formulario TeoriaOndaTriangular Página 2.....	87
Ilustración 111. Formulario TeoriaRestAltImp.....	87
Ilustración 112. Formulario TeoriaRestSimple.....	88
Ilustración 113. Formulario TeoriaSumInv.....	88
Ilustración 114. Formulario TeoriaSumNoInv.....	89
Ilustración 115. Pantalla principal del programa.....	99
Ilustración 116. Pestaña del circuito inversor.....	100
Ilustración 117. Mensaje de error del circuito inversor I.....	100
Ilustración 118. Mensaje de error del circuito inversor II.....	101
Ilustración 119. Mensaje de error del circuito inversor III.....	101
Ilustración 120. Resultados del circuito inversor.....	102
Ilustración 121. Error de representación gráfica del circuito inversor.....	103



1. Introducción

El proyecto que se lleva cabo comienza con un pequeño estudio del **amplificador operacional** y sus diferentes aplicaciones en circuitos. Estas aplicaciones son de gran interés, ya que podemos realizar operaciones matemáticas con señales que introduzcamos al amplificador, comparaciones entre dichas señales de entrada o modificar la salida del circuito a nuestra conveniencia, entre otras aplicaciones más que se verán en los anexos.

Siguiendo con la breve explicación general de los amplificadores operacionales, nos meteremos de lleno con la realización de la herramienta para el cálculo de circuitos con amplificadores operacionales. En este apartado se mostrará el uso del lenguaje VBA de Excel, en este caso orientado a las necesidades de nuestra aplicación, la estructura interna que se ha creado para dicha herramienta, la explicación de su funcionamiento y varios ejemplos visuales para demostrar que la aplicación funciona correctamente.

En el apartado 5. *Conclusiones* se recogen todas las reflexiones, ideas finales y conclusiones a cerca del trabajo, además de todas las referencias bibliográficas y recursos web que se han utilizado y consultado para el trabajo, reflejadas en el apartado 6. *Referencias bibliográficas*.

Por último, se muestran los anexos en el apartado 7, donde se hará un análisis extenso de las aplicaciones en circuitos de los amplificadores operacionales. Se incluirá también todo el código referido al programa.





2. Objetivos

El objetivo principal es que la aplicación que se desarrolla se pueda introducir en el método de aprendizaje de los amplificadores operacionales y su uso en circuitos. Es decir, que se pueda practicar con cualquier circuito de los incluidos en la aplicación, y que el usuario pueda recurrir a ella para comprobar resultados, gráficas o simplemente para conocer el funcionamiento de un circuito en particular.

En segundo lugar, pero no menos importante, cabe destacar el objetivo personal de afianzar los conocimientos sobre el A.O. y su integración en circuitos, al igual que conocer y manejar la herramienta VBA de Microsoft Office Excel.



3. Análisis de los amplificadores operacionales

Un amplificador operacional es un dispositivo electrónico de gran versatilidad que se usa muy a menudo a la hora de implementar circuitos electrónicos. Se trata de un circuito integrado que funciona como un amplificador multietapa de alta ganancia directamente acoplado.

Estos dispositivos cuentan con dos señales de entrada y una de salida. De las señales de entrada nos interesará la señal diferencial que obtengamos de estas. La señal de salida será el resultado del término diferencial de las señales de entrada, pero amplificada proporcionalmente por la ganancia que disponga nuestro amplificador operacional, obteniendo una señal asimétrica a la salida.

3.1. Estructura del amplificador operacional

En la *ilustración 1* se muestra la estructura interna de un amplificador operacional, en concreto el modelo $\mu A741$, que es el empleado en las prácticas de laboratorio de la asignatura *Electrónica Analógica*, perteneciente al tercer curso del grado en *Ingeniería en Electrónica Industrial y Automática*:

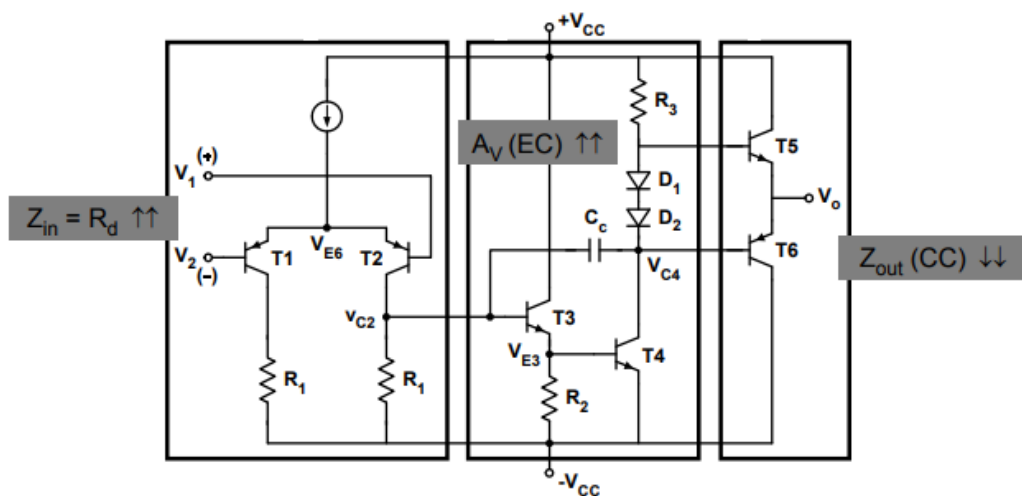


Ilustración 1. Estructura interna del Amplificador Operacional $\mu A741$

En el primer bloque, la parte izquierda de la estructura, se integra un amplificador diferencial al que están conectadas las dos señales de entrada que se ha comentado anteriormente. De esta etapa, se obtiene una señal asimétrica que será llevada hacia una etapa intermedia de amplificación de alta ganancia empleando una configuración emisor-común. Finalmente, pasa por una etapa de salida en configuración colector-común, que es la que nos proporciona una menor distorsión a la salida.

En la siguiente ilustración, mostramos el circuito eléctrico equivalente del anterior montaje. El valor de Z_{in} será muy elevado, del orden de $M\Omega$, que hará que por las patillas de entrada dispongamos de una corriente de polarización prácticamente nula. En el caso de Z_{out} , su valor será muy pequeño, en torno a $0-100 \Omega$.

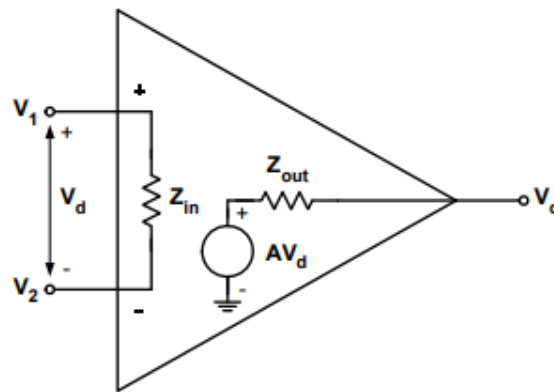


Ilustración 2. Circuito eléctrico equivalente del Amplificador Operacional.

3.2. Notación y terminología de un amplificador operacional

Una vez analizada la estructura interna que compone un amplificador operacional, nos centramos en la representación más universal de lo que se conoce como amplificador operacional. En la *ilustración 3*, se puede ver dicha representación con sus correspondientes señales de entrada (V_x y V_y) y salida (V_o). Todas estas señales están referidas a masa.

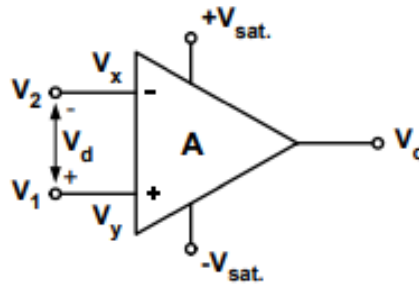


Ilustración 3. Representación del Amplificador Operacional

A continuación, se definen los términos de un amplificador operacional:

- Entrada no inversora (V_y): Si se aplica señal por esta entrada, siendo la señal de la otra patilla nula, aparecerá en la salida sin invertir.
- Entrada inversora (V_x): Si se aplica señal por esta entrada, siendo la señal de la otra patilla nula, aparece en la salida dicha señal con la fase opuesta o invertida.
- Entrada diferencial (V_d): Señal que representa el valor diferencial entre las dos señales de entrada.
- Ganancia (A): Es una magnitud adimensional que representa la amplificación de la señal diferencial de entrada. Dependiendo del modo de funcionamiento del amplificador operacional, tendremos una ganancia u otra.
- Tensión de saturación ($+V_{sat}$ y $-V_{sat}$): Establecen los límites de tensión que podemos llegar a tener en la salida de nuestro amplificador operacional.
- Señal de salida (V_o): Como su propio nombre indica, representa el valor de tensión que obtenemos a la salida del amplificador operacional.

3.3. Modos de funcionamiento del amplificador operacional

Después de haber establecido la notación y terminología de un amplificador operacional, debemos conocer los modos de funcionamiento que permite el amplificador operacional. Son los siguientes:

→ FUNCIONAMIENTO EN LAZO ABIERTO

En este caso, el amplificador operacional queda definido como en la *ilustración 3*, es decir, no existe ninguna realimentación que haga que la señal de entrada quede influida por la señal de salida. Con este modo, se obtiene una ganancia A muy elevada, del orden de $>10^5$, haciendo que el tiempo de conmutación en el paso de $-V_{sat}$ a $+V_{sat}$ sea casi despreciable. La tensión a la salida queda definida de la siguiente forma:

$$V_o = A \cdot V_d = A \cdot (V_y - V_x)$$

Aunque el valor de V_d adquiera valores muy pequeños, debemos de tener en cuenta que la ganancia del amplificador es muy grande. Esto implica que la salida tenderá siempre a la saturación, ya sea a saturación positiva o saturación negativa. Por tanto, para este modo de funcionamiento, los amplificadores operacionales se usan como comparador de las señales de entrada que se conecten al propio amplificador operacional.

En la *ilustración 4*, se representa la función de transferencia de este caso de amplificador operacional.

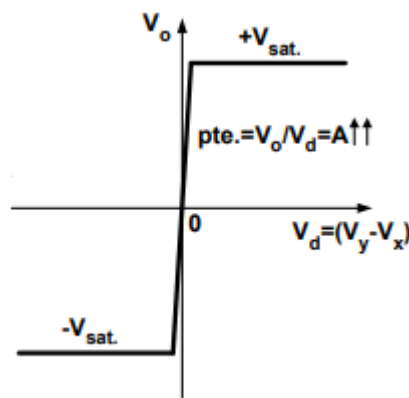


Ilustración 4. Característica de transferencia en lazo abierto del Amplificador Operacional

→ FUNCIONAMIENTO CON REALIMENTACIÓN NEGATIVA

Si la salida de nuestro amplificador operacional se encuentra conectado con la patilla inversora del mismo, entonces nuestro amplificador operacional tiene realimentación negativa. Esto implica que el valor de la tensión de la entrada inversora (V_x) y el de la no inversora (V_y) sea el mismo, y que, por tanto, V_d tenga valor nulo. Otro aspecto a tener en cuenta es que

el valor de la ganancia de tensión A estará limitado por la relación de impedancias que exista entre la entrada realimentada y la salida.

A partir de este montaje, podremos realizar numerosas operaciones matemáticas, las cuales se pueden ver más adelante, más concretamente en el apartado 3.6. Aplicaciones lineales y no lineales de los A.O.

3.4. Encapsulado del amplificador operacional

Otro aspecto de interés de los amplificadores operacionales es su encapsulado. Existen varios tipos de encapsulados, pero únicamente nos centraremos en uno solo, más concretamente en el encapsulado del amplificador operacional $\mu A741$, que es el mismo con el que hemos analizado la estructura interna del amplificador operacional. En la *ilustración 5*, mostramos dicho encapsulado.

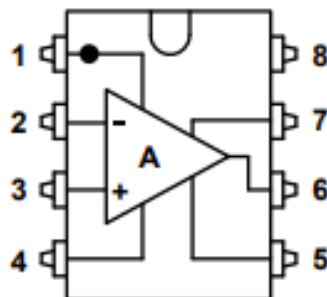


Ilustración 5. Encapsulado del Amplificador Operacional $\mu A741$

Como podemos observar, el encapsulado dispone de 8 patillas, 7 de ellas utilizables, y cuya descripción comentamos a continuación:

- Patilla 1: Ajuste del offset
- Patilla 2: Entrada inversora
- Patilla 3: Entrada no inversora
- Patilla 4: Alimentación negativa
- Patilla 5: Ajuste del offset
- Patilla 6: Salida
- Patilla 7: Alimentación positiva

En cuanto a las patillas 4 y 7, dichas señales se refieren a los valores de tensión límite que puede adoptar la salida, es decir, los valores de saturación máximo y mínimo del amplificador operacional.

Por otro lado, también nos centramos en las patillas referidas al ajuste del offset, patillas 1 y 5. En un amplificador operacional, puede aparecer *tensión offset de entrada*. Esto es que existe una tensión diferencial a la entrada cuando la tensión de salida es nula. Mediante unos pequeños ajustes, los cuales no se desarrollan ya que no es de interés para este trabajo, podemos conseguir que esta tensión se haga nula y obtener un mejor comportamiento del amplificador operacional.

3.5. Diferencias entre un A.O. ideal y un A.O. real

Para terminar con este breve análisis de los amplificadores operacionales, en la siguiente ilustración mostramos las características principales tanto de un amplificador operacional ideal como de un amplificador operacional real.

Características	A.O. ideal	A.O. real
Ganancia en lazo abierto (A)	Infinita	Muy alta $\approx 2 \cdot 10^5 = 106 \text{ dB}$
Impedancia de entrada (Z_{in})	Infinita	Muy elevada $\approx 2 \text{ M}\Omega$
Impedancia de salida (Z_{out})	Nula	Muy pequeña $\approx 75 \Omega$
Tensión offset (V_{IO})	Nula	Muy pequeña $\approx 2 \text{ mV}$
Corrientes de polarización (I_B)	Nulas	Muy reducidas $\approx 75 \text{ nA}$
Tiempo de conmutación (SR)	Nulo	Muy pequeño $\approx 5 \text{ V}/\mu\text{s}$
Ancho de banda (f_T)	Infinito	Finito $\approx 1 \text{ MHz}$

Ilustración 6. Tabla comparativa de un A.O. ideal y un A.O. real



3.6. Aplicaciones lineales y no lineales de los A.O.

Los amplificadores operacionales tienen dos aplicaciones a la hora de integrarse en los circuitos. Estas aplicaciones se conocen como lineales y no lineales.

Para el caso de las aplicaciones lineales de los A.O., se obtienen con un modo de funcionamiento del amplificador operacional en realimentación negativa. Así, podemos conseguir realizar diversas operaciones matemáticas referidas a la relación entre las entradas y la salida. Los circuitos con aplicaciones lineales de los A.O. que se tratan en este proyecto son los siguientes:

- Circuito inversor
- Circuito no inversor
- Circuitos sumadores
 - Circuito sumador inversor
 - Circuito sumador no inversor
- Circuitos restadores
 - Circuito restador simple
 - Circuito restador con alta impedancia de entrada
- Circuitos integradores
 - Circuito integrador inversor
 - Circuito integrador no inversor
 - Circuito integrador sumador
 - Circuito integrador restador
- Circuito diferenciador
- Fuentes de tensión de referencia
 - Fuente con diodo Zener
- Fuentes de corriente
- Convertidor corriente-tensión
- Convertidor tensión-corriente

Por otro lado, tenemos los circuitos con aplicación no lineal de los A.O., en los que el modo de funcionamiento del amplificador operacional puede ser en realimentación, tanto positiva como negativa, o en lazo abierto. De esta forma, podremos realizar operaciones que van más allá de una relación lineal entre entrada y salida. Si incluimos otros procesos a esta relación de entrada-salida de nuestro amplificador operacional, podremos conseguir circuitos de gran



utilidad y con una función específica. En la siguiente lista, mostramos los circuitos de este tipo que se van a tratar en este proyecto:

- Circuitos limitadores
 - Circuito limitador con diodo normal
 - Circuito limitador con diodo normal y diodo zener en serie
 - Circuito limitador con dos diodos zener
- Detectores de tensión o comparadores
 - Detector con salida saturación-saturación y las dos señales de entrada en el mismo terminal
 - Detector con salida cero-saturación
 - Disipador de Schmitt
- Comparadores con histéresis
 - Comparador con histéresis simple
 - Comparador con histéresis y limitación a la salida
- Generador de onda cuadrada
- Generador de onda triangular
- Rectificadores de precisión
 - Rectificador de media onda
 - Rectificador de onda completa
 - Circuitos de valor medio y valor eficaz
- Detectores de pico

4. Programa para la resolución de circuitos con A.O.

Como se ha comentado en el inicio del proyecto, el objetivo principal es realizar una **herramienta que nos permita resolver circuitos con amplificadores operacionales**. Circuitos como los que se han comentado en el anterior apartado y que se explican de manera extensa en los apartados 7.1.1. Aplicaciones lineales de los amplificadores operacionales y 7.1.2. Aplicaciones no lineales de los amplificadores operacionales de los anexos de la memoria. A continuación, se explica el desarrollo de dicha herramienta y su funcionamiento.

4.1. Descripción del software empleado

Para la creación de la aplicación, se ha empleado la herramienta *Microsoft Excel*, un software desarrollado por Microsoft que, mediante hojas de cálculo, permite realizar numerosas operaciones matemáticas o gestión de datos, entre muchas cosas más. Las funcionalidades básicas de Excel son mundialmente conocidas, pero si buscamos realizar tareas que van más allá de gestionar datos que introduzcamos a nuestra hoja de cálculo, como pueden ser acciones repetitivas sobre estos datos, Microsoft Excel dispone de lo que se conoce como *Visual Basic Application* (VBA).

VBA es una herramienta integrada en el office de Excel, que interacciona con las hojas de cálculo a través de *macros*. Las macros son instrucciones que se encargan de registrar una serie de comandos para que puedan ser ejecutados automáticamente por el usuario. Además de esto, VBA dispone del *Editor de Visual Basic*, un entorno de programación en lenguaje Visual Basic, con el que podemos realizar programas de gran utilidad. La comunicación entre VBA y la hoja de cálculo de Excel se realiza con las macros comentadas anteriormente. Para poder realizar todo esto, debemos activar previamente la opción *Programador* de Excel para poder trabajar con Visual Basic. Para ello, tenemos que seguir la ruta *Archivo -> Opciones -> Personalizar cinta de opciones*, llegando a la siguiente ventana:

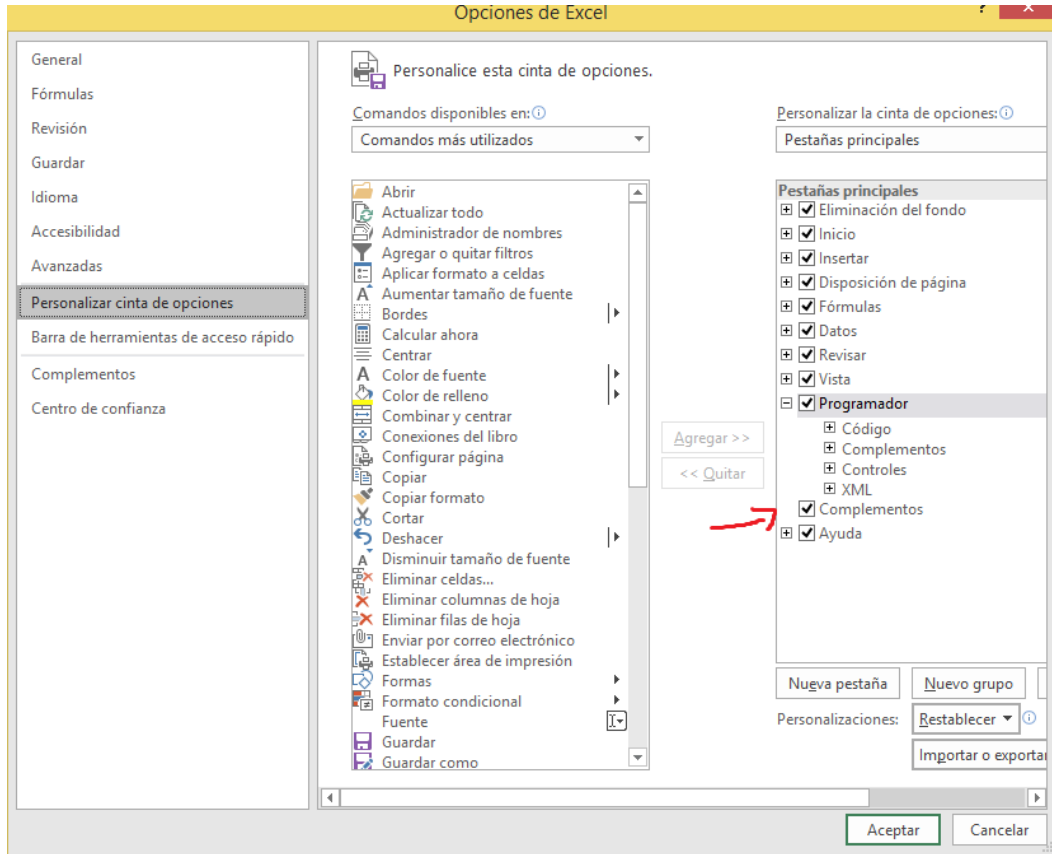


Ilustración 7. Activación de la pestaña Programador

Buscamos el apartado *Programador* en nuestros comandos y lo agregamos a nuestras pestañas principales, para posteriormente activarlo.

De esta forma, podremos acceder a la herramienta de Visual Basic, como mostramos en la siguiente ilustración:

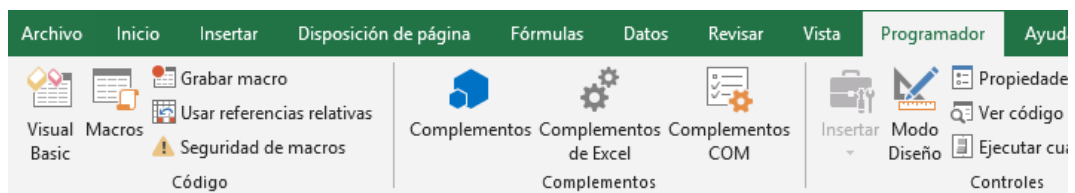


Ilustración 8. Acceso a Visual Basic

La ventana principal de Visual Basic es la que mostramos en la *ilustración 9*:

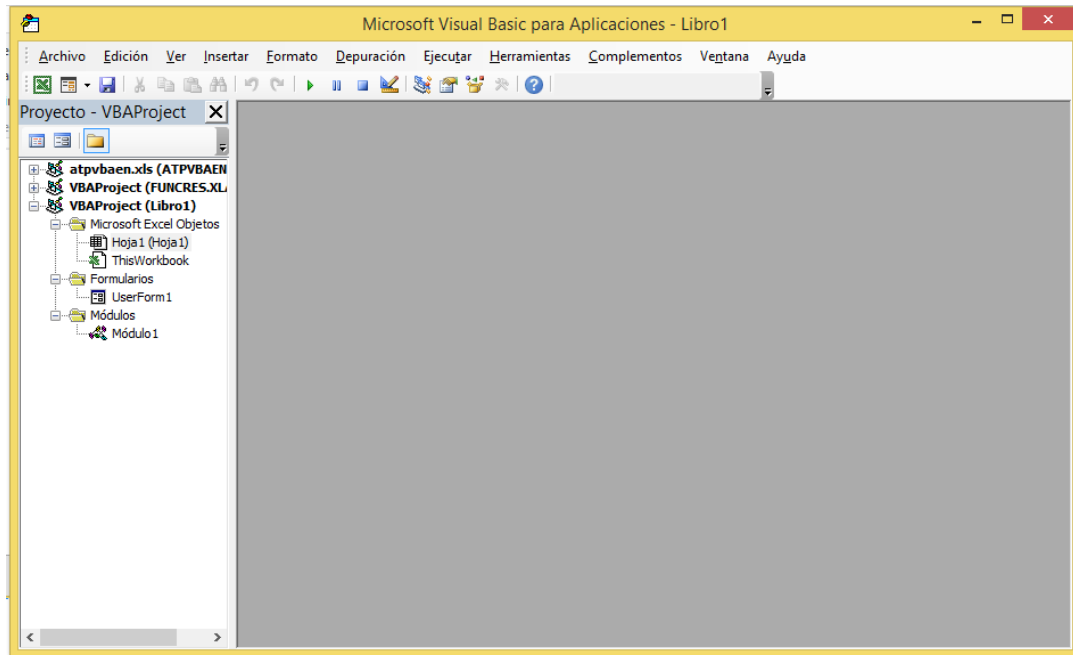


Ilustración 9. Ventana principal de Visual Basic

La programación se basa en el uso de *Formularios* y *Módulos*.

De los formularios usaremos los *UserForm*, unas viñetas que se visualizan mediante la ejecución de macros desde la hoja de cálculo. Con ellos, podemos mostrar cuadros de texto o imágenes. También se pueden introducir botones o pedir que el usuario introduzca datos en la viñeta, entre otras acciones. El control de todas estas acciones se realiza mediante código.

Por otra parte, Visual Basic dispone de los módulos. Se trata de un bloque de instrucciones que nos permite simplificar las tareas del código, como pueden ser cálculos matemáticos, obtener información del programa, manipular datos de formularios o de las hojas de cálculo, etc. Nos basaremos en módulos *Sub*, y se ejecutarán llamándolos desde los *UserForm*.

4.2. Estructura interna del programa

El programa consta de varios formularios y módulos desarrollados para cada uno de los circuitos incluidos en el programa. Todos ellos se muestran en las siguientes ilustraciones:

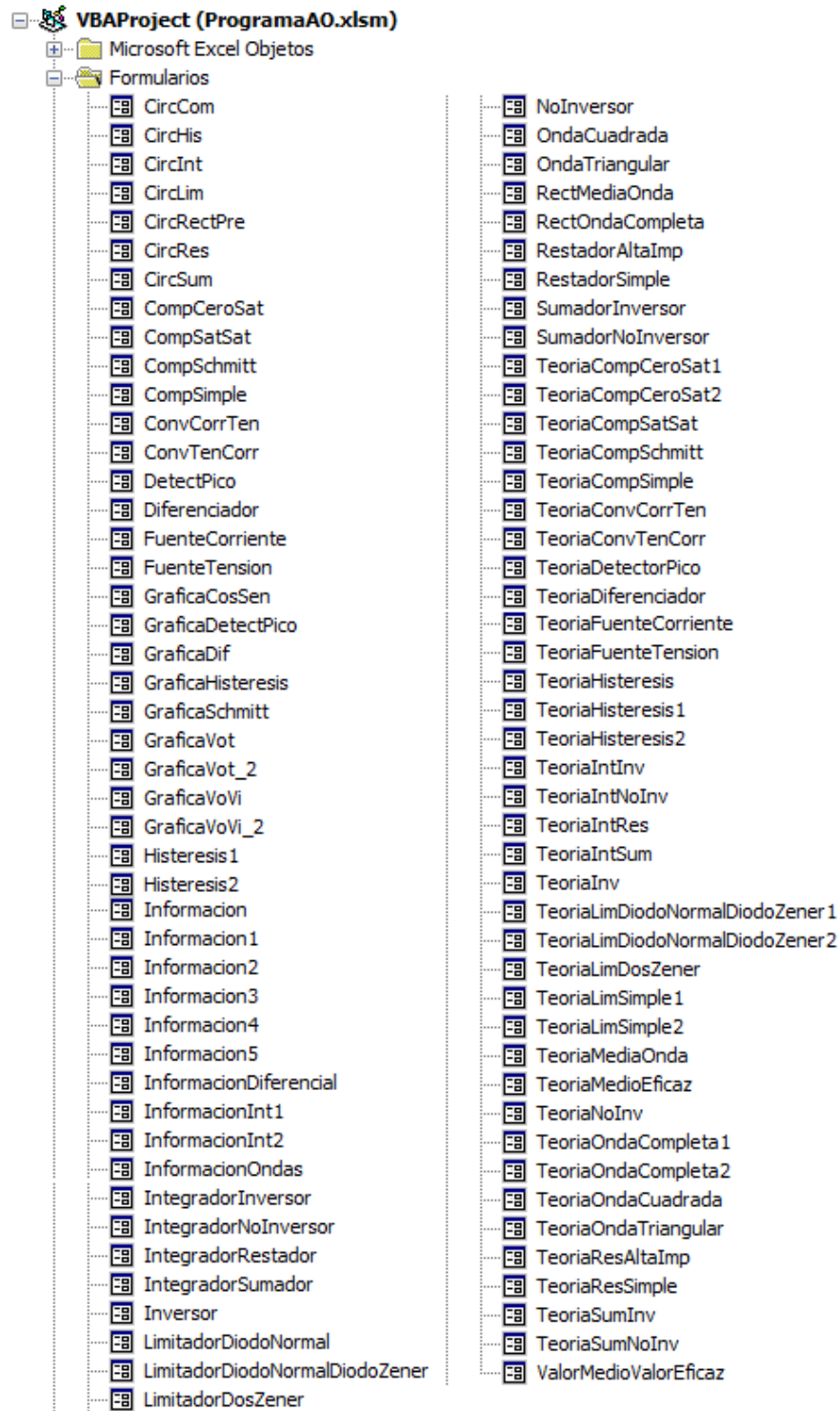


Ilustración 10. Formularios del programa

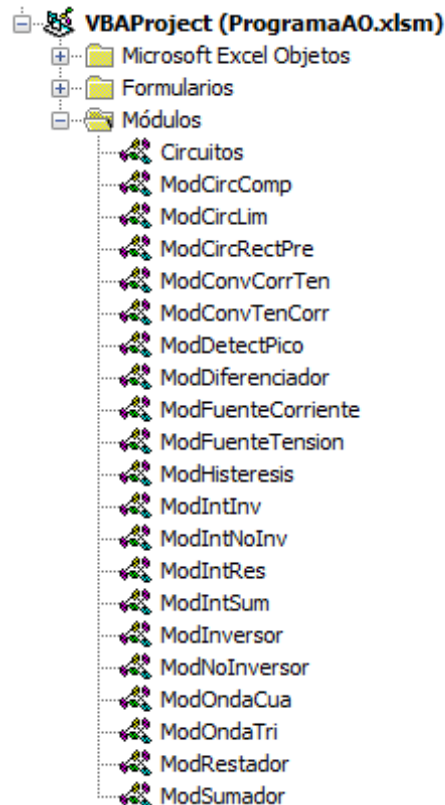


Ilustración 11. Módulos del programa

En los dos próximos apartados se explicarán los aspectos más importantes de todos los formularios y módulos respectivamente, mostrados en las ilustraciones anteriores. Es conveniente que, durante los dos próximos apartados, se tenga a mano el apartado 7.2. Código del programa de los anexos de la memoria para ir revisando el código de cada uno de los archivos que se describen.

4.2.1. Formularios del programa

La metódica a seguir en este apartado será mostrar la viñeta del formulario correspondiente y explicar cómo se ha desarrollado dicho formulario. No se mostrará el código que contenga cada formulario, pues en los anexos se recogerá todo el código.

Podemos diferenciar cuatro tipos de formularios, en función de su utilidad y su contenido.

! El primer tipo de formulario corresponde a los apartados entre el 4.2.1.1. y el 4.2.1.7. Estos *UserForm* contienen únicamente las funciones *CommandButton*, unos botones de comando que, al pulsarlos, la aplicación ejecuta el código que tenga asociado dicho botón. Cada botón está asociado a un circuito del programa, de tal forma que, cuando cliquemos en un botón, la aplicación muestre la ventana del circuito seleccionado. La propiedad *Caption* de la función *CommandButton* es donde introduciremos el nombre del circuito en cuestión. El código de cada botón llama al procedimiento *mostrarcircuito* correspondiente del módulo asociado a los circuitos que estemos tratando. Cabe añadir que, en el caso de los circuitos con aplicaciones lineales de los A.O., el color de la ventana del formulario y de los botones tienen un tono verde. En el caso de los de aplicaciones no lineales, tendrán un color naranja.

4.2.1.1. CircCom

Este *UserForm* corresponde con los detectores de tensión o comparadores. Tenemos un botón para cada circuito comparador que dispone la aplicación.

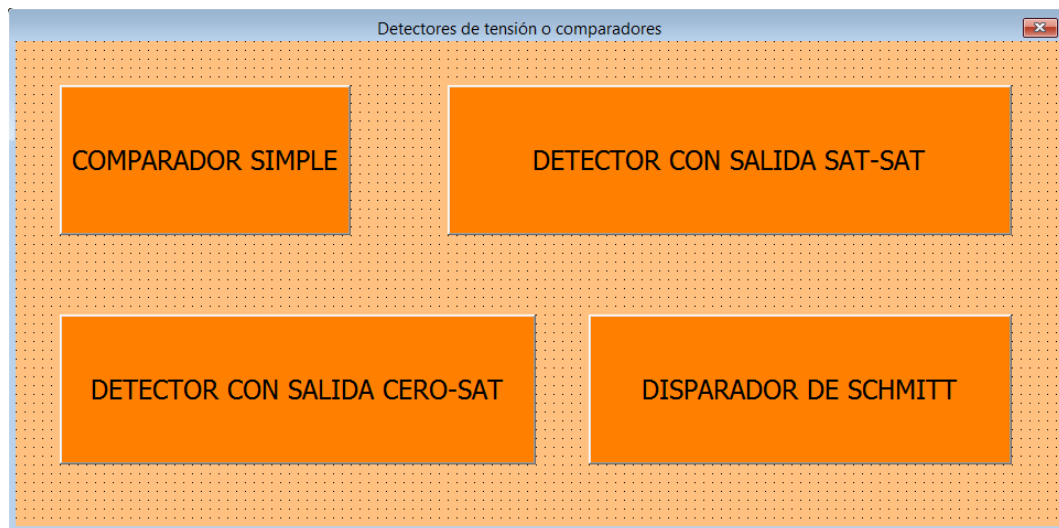


Ilustración 12. Formulario CircCom

4.2.1.2. Circhis

En este *UserForm*, los botones corresponden con los dos circuitos comparadores con histéresis.

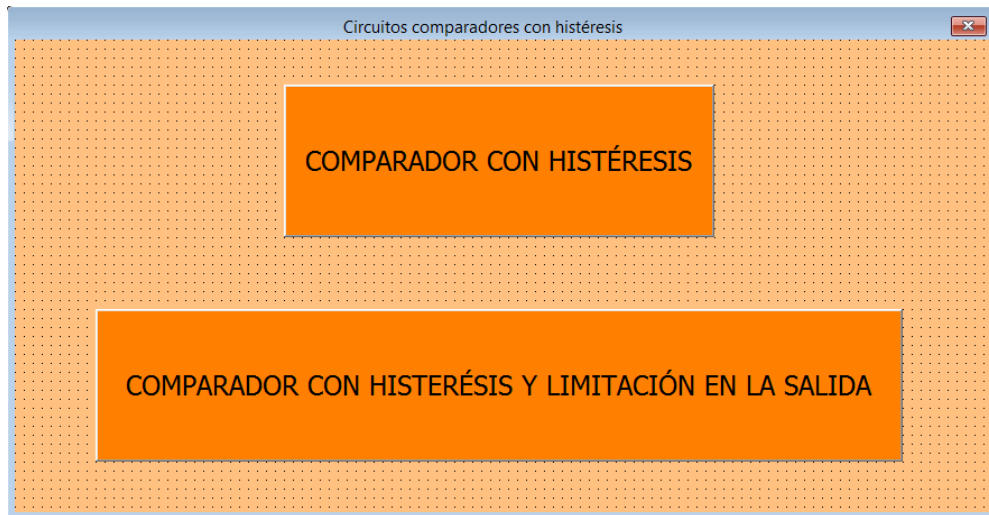


Ilustración 13. Formulario Circhis

4.2.1.3. Circlnt

Circlnt corresponde con los circuitos integradores, en el que se incluyen los botones integrador inversor, integrador no inversor, integrador sumador e integrador restador.

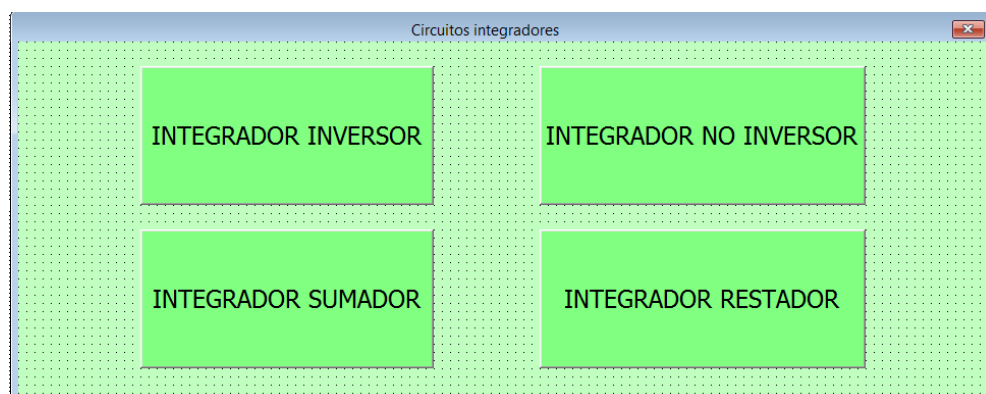


Ilustración 13. Formulario Circlnt

4.2.1.4. Circlim

En la siguiente ilustración mostramos el *UserForm* correspondiente a los circuitos limitadores. Dispondremos, como en los anteriores casos, de un botón para cada circuito limitador.

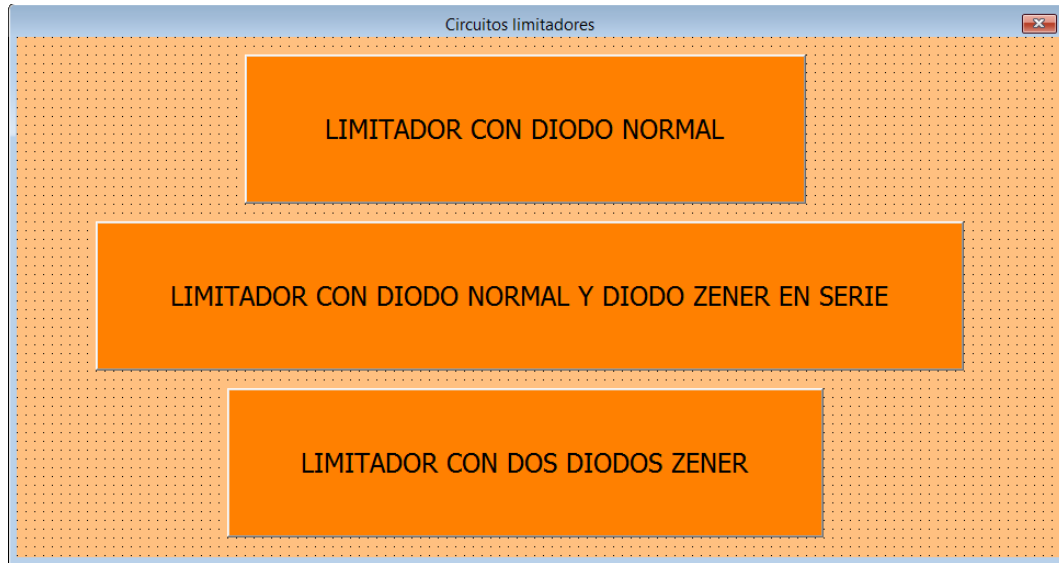


Ilustración 14. Formulario Circlim

4.2.1.5. CircRectPre

Este *UserForm* muestra los circuitos rectificadores de precisión con sus correspondientes botones.

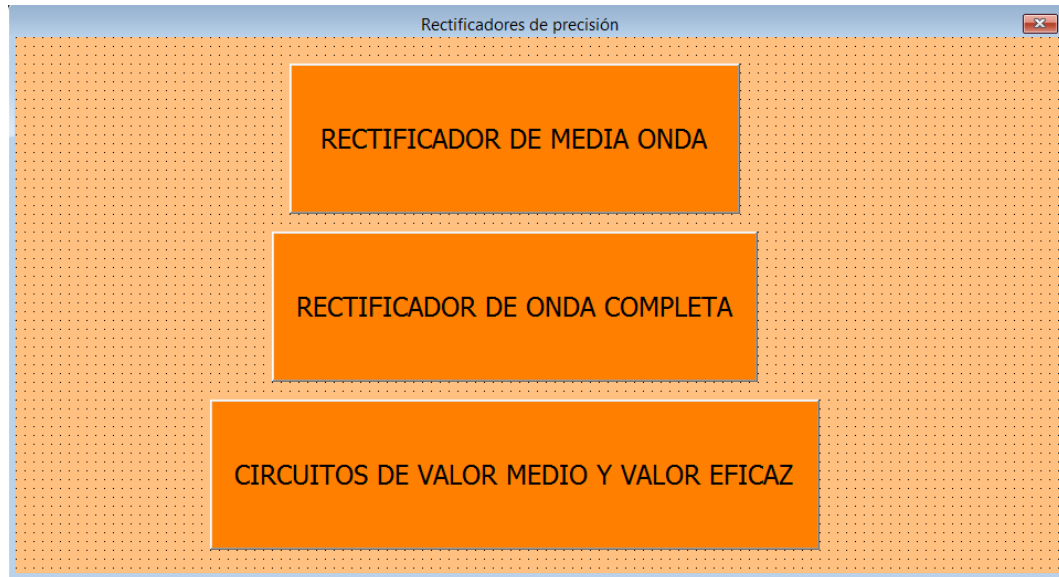


Ilustración 15. Formulario CircRectPre

4.2.1.6. CircRes

En la *ilustración 77* se muestra el formulario con los dos circuitos restadores de la aplicación.

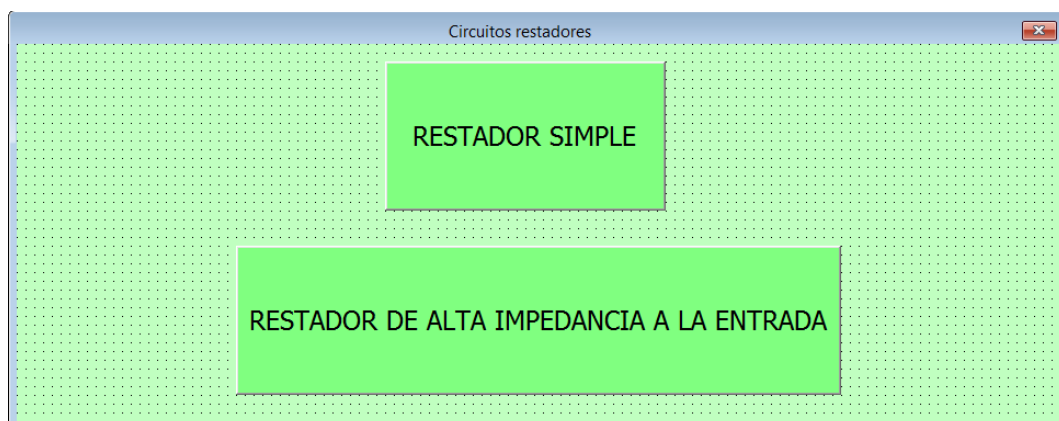


Ilustración 16. Formulario CircRes

4.2.1.7. CircSum

Por último, se muestra el formulario referido a los circuitos sumadores y los dos botones correspondientes a cada uno de los circuitos.

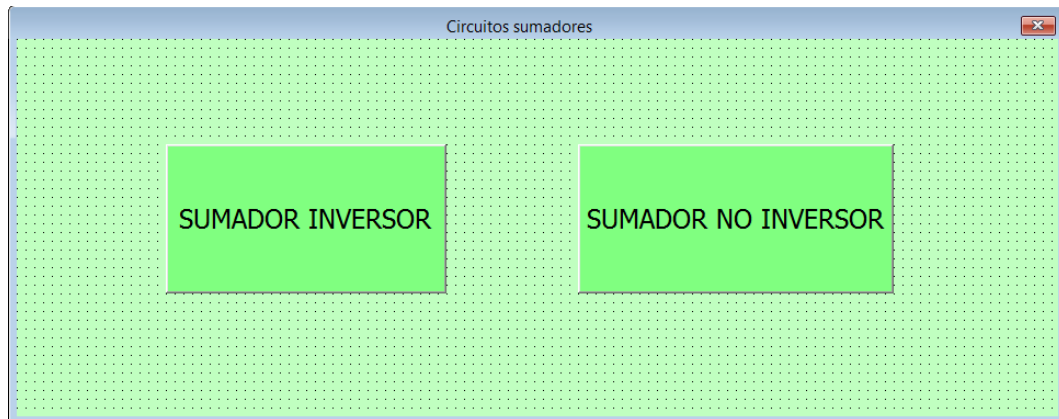


Ilustración 17. Formulario CircSum

II: El siguiente tipo de formulario es el empleado para mostrar las ventanas de los circuitos de nuestra aplicación. Aunque más o menos siguen una misma línea de estructura, cada formulario tiene sus propias especificaciones, pues para cada circuito se necesitan calcular unos parámetros. A continuación, se muestran cada uno de los formularios referidos a cada circuito y la explicación de cómo se ha desarrollado. De nuevo, para los circuitos con aplicaciones lineales se establece un tono verde y para los de aplicaciones no lineales se emplea un tono anaranjado.

4.2.1.8. CompCeroSat

El formulario *CompCeroSat* se emplea para la representación del circuito comparador con salida cero-saturación. De este circuito disponemos de dos ejemplos, por lo que pondremos dos pestañas en nuestro formulario. Para ello, introducimos el elemento *Page*, con el que conseguimos poner 2 páginas a nuestro formulario. Dentro de cada elemento *Page* introduciremos un ejemplo del circuito. Para este circuito, la estructura de cada una de las páginas del formulario es exactamente igual. Introducimos un elemento *Image* en la que definimos el circuito al que nos referimos. Dicha imagen se colocará en la parte



izquierda de la ventana. En la parte derecha del formulario, colocaremos unas etiquetas para hacer referencia a los valores que se introducen. Los dos elementos *Label* en negrita nos sirven para identificar el tipo de valores que se están tratando. Debajo de estas etiquetas, se colocan unos elementos llamados *TextBox*, una especie de casillas en las que se introducen valores. Al lado de cada una de ellas, se colocan otros elementos *Label*, para saber qué valores estamos introduciendo. Debajo del *TextBox* referido al valor *Vu* se dispone de una etiqueta en la que se mostrará si el diodo *D* conduce o no. Debajo de todos estos elementos introduciremos unas etiquetas, que nos servirán como elementos para mostrar los resultados del circuito. Estas 2 etiquetas con fondo anaranjado son un conjunto de 3 elementos *Label*; una primera etiqueta con la que se identifica qué valor se está calculando, otra etiqueta a continuación de la anterior donde se muestra el resultado y una última etiqueta donde se muestra la unidad en la que se obtiene el resultado. A su vez, debajo de los valores de tensión se añade otra etiqueta en la que se indica si la salida está saturada o no. En la zona inferior de la parte de la derecha se colocarán también otros tres elementos *CommandButton*; el botón *Calcular*, con el que conseguimos obtener los resultados al presionarlo, el botón *Restaurar valores*, con el que vaciamos los valores que se hayan introducido en los *TextBox* y el botón *Representación Vo-Vi*, con el que se muestra la ventana para generar la gráfica del circuito. Disponemos de otros dos botones más. Debajo de la imagen disponemos del botón *Teoría*, con el que se mostrará la teoría del circuito. Por otro lado, en la esquina superior de la derecha se encuentra el botón *Información*, con el que se consigue mostrar la ventana de información del circuito al clicarlo.

En cuanto al código referido al objeto, se definirán funciones del tipo *Private Sub* (se definen las funciones como privadas ya que no van a ser llamadas desde otros objetos) relacionadas con cada botón del formulario. Tendremos las funciones *Teoria_Click()*, *Informacion_Click()*, *Calcular_Click()*, *Restaurar_Click()* y *Vo_Vi_Click()*. Estas funciones ejecutarán las llamadas a su módulo correspondiente, en este caso *ModCircComp*, seguido del procedimiento correspondiente. Para el caso del botón *Teoría*, la sentencia será *ModCircComp.MostrarTeoria*. Esta sentencia se repetirá para el resto de los botones; para el botón *Calcular* se define la función *CalcularValores*, para el botón *Restaurar valores* la función *RestaurarValores* y para el botón *Información* la sentencia *MostrarInformacion*. Para el caso del botón de *Representación de Vo-Vi* se introducirá una condición *If - Else - End If*, estableciendo que, si no se ha obtenido un valor en la casilla de la tensión de salida, se muestre un mensaje diciendo que no se puede generar la representación gráfica. Si esto no sucede, se cargará el formulario correspondiente mediante tres líneas de código; *load grafica* (cargamos el formulario que define la gráfica del circuito), *grafica.replaint* (refrescamos la

gráfica con los últimos datos introducidos) y *grafica.show* (mostramos la gráfica). Todas estas funciones tendrán que ser definidas para cada una de las páginas de nuestro formulario. Para ello, tendremos que haber definido todos los elementos con nombres distintos para cada página, al igual que las funciones que se han descrito. Por último, se establece una función *UserForm_Terminate()*, cuyo código se ejecuta cuando el formulario se ha cerrado. En esta función vaciaremos las casillas que se han utilizado en la hoja de cálculo correspondiente para generar la gráfica.

En las siguientes ilustraciones se muestran las dos páginas que se han desarrollado para el formulario.

Introduzca el valor de las resistencias (en $K\Omega$)

R1= R2=

Introduzca el valor de los voltajes (en V)

Vi= Vu= +Vsat= -Vsat=

Vref=

Calcular **Restaurar valores**

R3 = **(KΩ)** Vo = **(V)**

Teoría **Representación Vo-Vi**

Ilustración 18. Formulario CompCeroSat Ejemplo 1

Introduzca el valor de las resistencias (en $K\Omega$)

R1= R2=

Introduzca el valor de los voltajes (en V)

Vi= Vu= +Vsat= -Vsat=

Vref=

Calcular **Restaurar valores**

R3 = **(KΩ)** Vo = **(V)**

Teoría **Representación Vo-Vi**

Ilustración 19. Formulario CompCeroSat Ejemplo 2

4.2.1.9. CompSatSat

Este formulario se emplea para la representación de la ventana del circuito comparador con salida saturación-saturación. La estructura del objeto presenta la misma forma que el anterior formulario, con la diferencia de que, para este circuito, solo disponemos de un solo ejemplo. Esto hace que no sea necesario utilizar el elemento *Page*. Los elementos *Label* y *TextBox* de la parte derecha del formulario son prácticamente los mismos que para el anterior caso, quitando el parámetro *Vu*, que para este caso no es necesario al no tener diodo en el circuito. La forma del código será la misma que la del formulario *ComCeroSat* pero sin tener que poner dos veces cada una de las funciones referidas a los botones, ya que solo disponemos de una página de formulario. Debido a esto, no se comenta el código.

Ilustración 20. Formulario CompSatSat

4.2.1.10. CompSchmitt

El disipador de Schmitt emplea este formulario para que se ejecute su ventana. En este caso, no se disponen de elementos *Label* y *TextBox* para resistencias, pues el circuito no dispone de ellas. En los *TextBox* de las tensiones, la tensión de entrada tiene forma senoidal, por lo que los elementos a introducir serán la amplitud y el periodo de la onda. Del código cabe comentar que para la función *Private Sub UserForm_Terminate()* se recogen celdas de dos hojas de cálculo

distintas para vaciarlas, pues en la representación gráfica se emplean dos elementos *Image* para representar V_i - t y V_o - t . El resto de código es igual que en los casos anteriores.

Ilustración 21. Formulario *CompSchmitt*

4.2.1.11. *CompSimple*

El formulario *CompSimple* está referido al circuito comparador simple, tal y como vemos en la siguiente ilustración. Al igual que para el caso anterior, no disponemos de elementos *Label* y *TextBox* para resistencias, pues el circuito no dispone de ellas. Por el resto del formulario, la estructura del objeto es igual que en el anterior caso, salvo que las tensiones de entrada del A.O. son constantes en vez de introducir una señal senoidal por la patilla inversora como era el caso del disipador de Schmitt. Otra diferencia es que, para la función *UserForm_Terminate()*, en el código sólo se vacían casillas de una hoja de cálculo, pues solo se representa una gráfica del circuito.

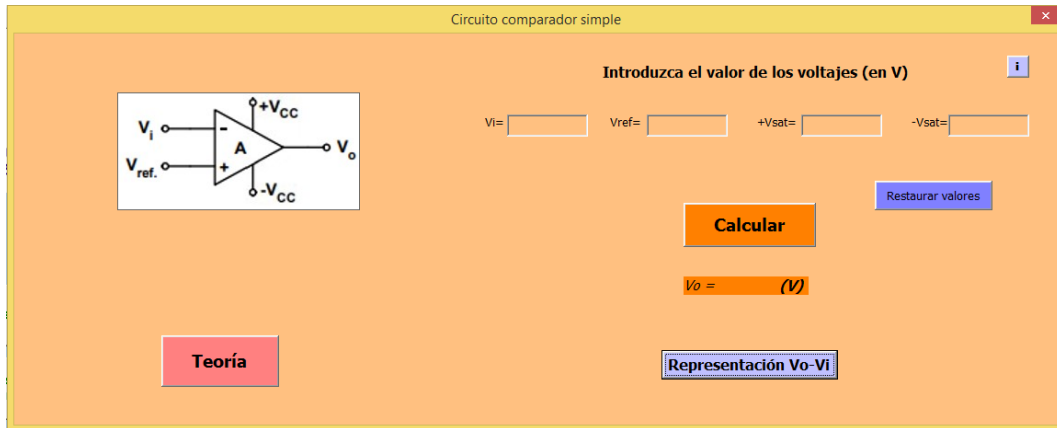


Ilustración 22. Formulario CompSimple

4.2.1.12. ConvCorrTen

El formulario *ConvCorrTen* se emplea para mostrar la ventana de los circuitos convertidores de corriente-tensión. Este caso dispone de dos ejemplos, por lo que el objeto dispondrá de un elemento *Page* para establecer dos páginas distintas. En cada página definiremos los *Label* y *TextBox* necesarios como en los anteriores casos. En la entrada tendremos corriente de entrada en vez de tensión de entrada, y estará definida en mA. En la etiqueta de K se define una etiqueta debajo de esta indicando si la constante de proporcionalidad es mayor o menor que cero. Los botones serán iguales que para el resto de casos, salvo que para estos circuitos no obtenemos la representación gráfica, por lo que suprimimos el botón referido a la representación gráfica. El código se basará en la definición de las operaciones para cada botón, estableciendo las sentencias referidas al procedimiento correspondiente.

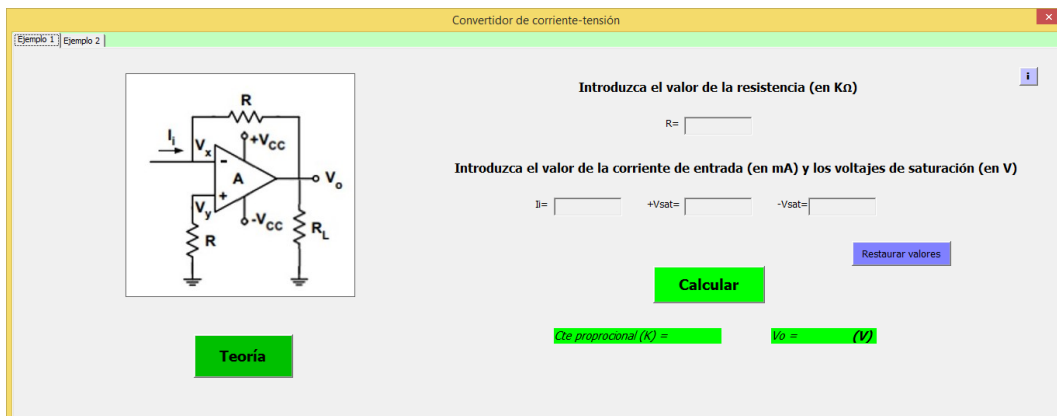


Ilustración 23. Formulario ConvCorrTen Ejemplo 1

Ilustración 24. Formulario ConvCorrTen Ejemplo 2

4.2.1.13. ConvTenCurr

El circuito convertidor de tensión-corriente tiene asociado el formulario *ConvTenCurr*. La estructura del objeto es básicamente la misma que la vista en los anteriores casos. En este caso, obtenemos a la salida un valor de corriente en vez de un valor de tensión. En cuanto al código, solo se definen las funciones para los botones *Calcular*, *Restaurar valores*, *Teoría* e *Información*. Para este caso no dispondremos de botón para la representación gráfica.

Ilustración 25. Formulario ConvTenCurr

4.2.1.14. DetectPico

El formulario *DetectPico* se emplea para representar la ventana del circuito detector de pico. La estructura del objeto posee una diferencia en comparación con el resto de formularios. Los elementos *CommandButton*, *Label* y *TextBox* siguen la misma estructura que en los anteriores casos, pero con la diferencia de que se introduce un botón más, *Interruptor S*, debido a las exigencias del circuito. El funcionamiento de este botón se explicará más adelante, concretamente en el módulo *ModDetectPico*, que es donde se desarrolla el código de este circuito. Para el formulario actual, dicho botón funciona exactamente igual que el resto de botones, siendo su función *Interruptor_Click()* y ejecutando la sentencia que nos lleva a la función correspondiente al interruptor del procedimiento del detector de pico. Cabe también destacar a nivel de código del objeto que se introduce la función *UserForm_Initialize()*, cuyo código se ejecuta cuando este formulario se abre. En esta función se definen unas variables que nos servirán como contadores. En este caso se definen sus valores iniciales. Será en la función *Calcular_Click()* en la que iremos aumentando el valor de estos contadores, es decir, cada vez que presionemos el botón calcular, aumentaremos el valor de los contadores. Este proceso nos servirá para ir almacenando el valor de pico de la tensión de salida.

The screenshot shows a software window titled "Detector de pico". On the left is a circuit diagram of a precision rectifier using two operational amplifiers (A1, A2), two diodes (D1, D2), a resistor (R), a capacitor (C), and a load resistor (RL). The input is Vi and the output is Vo. The circuit is powered by +VCC and -VCC. Below the diagram is a red button labeled "Teoría". To the right of the diagram is a control panel with the following elements:

- A header: "Introduzca el valor de los voltajes (en V)" with an information icon (i).
- Input fields: $V_i = \text{[]} \cdot \text{sen}(\text{[]} \cdot t)$, $+V_{sat} = \text{[]}$, and $-V_{sat} = \text{[]}$.
- Buttons: "Calcular" (orange), "Restaurar valores" (blue), and "Interruptor S" (blue).
- Output labels: $V_o \text{ (pico)} = \text{[]} \text{ (V)}$ and $V_o \text{ (pico) máx. alcanzado} = \text{[]} \text{ (V)}$.
- Bottom button: "Representación Vo-t" (blue).

Ilustración 26. Formulario DetectPico

4.2.1.15. Diferenciador

El formulario *Diferenciador* se emplea para visualizar la ventana del circuito Diferenciador. La estructura del objeto sigue la misma línea que resto del formulario con algunas excepciones. Se introduce un *TextBox* para un condensador, cuya unidad se define en μF . En cuanto a la tensión de entrada, se define un valor que se multiplicará al parámetro del tiempo elevado a una potencia, cuyo valor también lo introducirá el usuario. El valor del tiempo para el que calcularemos la salida se define mediante un nuevo *TextBox*, definido debajo de los elementos referidos a la tensión. En cuanto al código del objeto, sigue la misma estructura que los casos anteriores, con la diferencia de que en nuestra función *UserForm_Terminate()* tendremos que vaciar celdas de dos hojas de cálculo distintas al generar dos gráficas cuando clicamos el botón *Representación de Vi y Vo frente a t*.

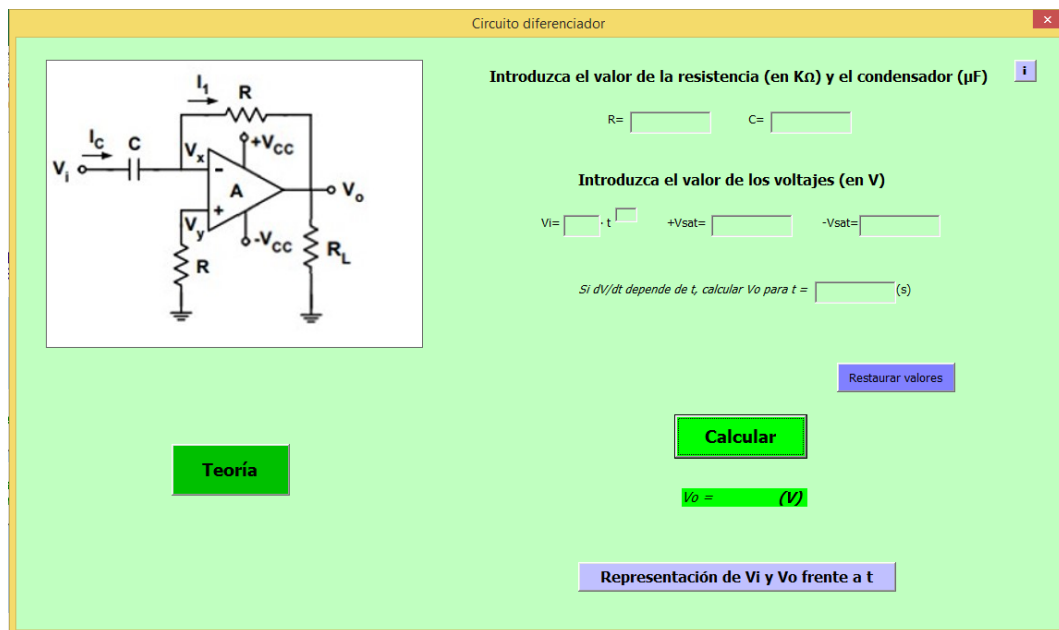


Ilustración 27. Formulario Diferenciador

4.2.1.16. FuenteCorriente

El formulario *FuenteCorriente* se emplea para mostrar la ventana de las fuentes de corriente. Para este caso disponemos de dos ejemplos de circuito, por lo que introduciremos un elemento *Page* para definir dos páginas en el formulario.

En cada una de ellas, definiremos un ejemplo de fuente de corriente, el cual dispone de los elementos *Label* y *TextBox* que venimos comentando en los anteriores apartados. La salida que obtenemos con estos circuitos será un valor de corriente, el cual vendrá definido en las unidades de mA. En cuanto al código del objeto, tendremos que definir las funciones de los botones dos veces, una por cada ejemplo. Para este caso, no se define la función para vaciar las celdas de la hoja de cálculo, pues en estos circuitos no se realiza la representación gráfica.

Ilustración 28. Formulario FuenteCorriente Ejemplo 1

Ilustración 29. Formulario FuenteCorriente Ejemplo 2

4.2.1.17. FuenteTension

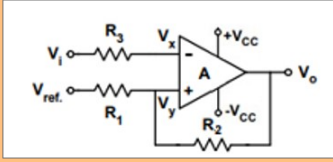
FuenteTension será el formulario que usamos para generar la ventana del circuito fuente de tensión. La estructura del objeto sigue la misma línea que el resto de formularios. Cabe comentar que, para este caso, no se dispone del botón para la representación gráfica, pues no es una información de interés para el circuito. El código quedará definido por las funciones de los botones de los que se dispone únicamente.

Ilustración 30. Formulario FuenteTension

4.2.1.18. Histeresis1

Histeresis1 es el formulario que empleamos para definir la ventana del circuito comparador con histéresis. Este formulario dispone de unas etiquetas añadidas a las que se han visto en los anteriores casos. Estas etiquetas son las definidas en color amarillo, que nos sirven para diferenciar las dos representaciones que se generan en la gráfica V_o - V_i . Para cada una de las representaciones definimos unos *Label* y *TextBox* proporcionando el valor de V_i en el que cambia de saturación la salida y el valor de V_o que se obtiene para el valor de V_i que se ha introducido por medio de su *TextBox*. Internamente, el código del objeto dispone de las mismas funciones que se han ido comentando en los anteriores apartados, pero generando las sentencias para el procedimiento del circuito comparador con histéresis.

Comparador con histéresis



Introduzca el valor de las resistencias (en $\text{k}\Omega$)

R1= R2=

Introduzca el valor de los voltajes (en V)

Vi= Vref= +Vsat= -Vsat=

Teoría

Calcular Restaurar valores

Inicialmente +Vsat Cambio para Vi > (V) Vo = (V)

Inicialmente -Vsat Cambio para Vi < (V) Vo = (V)

Representación Vo-Vi

Ilustración 31. Formulario Histeresis1

4.2.1.19. Histeresis2

El formulario *Histeresis2* es empleado para mostrar la pestaña de los circuitos comparadores con histéresis y limitación a la salida. La estructura del objeto es la misma que en el anterior caso, salvo que para este formulario debemos introducir un elemento *Page* para que podamos disponer de dos páginas, una para cada ejemplo. Además, se introducen dos nuevos elementos; la tensión umbral del diodo V_u junto a la etiqueta que indica si dicho diodo conduce o no, y la tensión de salida del A.O. que en este caso se denomina V_o . El código también tiene la misma construcción que el caso anterior, con la diferencia de que, al tener dos páginas en el formulario, tendremos que tener el doble de funciones para los botones de ambas páginas del formulario.

Comparador con histéresis y limitación a la salida

Ejemplo 1 | Ejemplo 2

Introduzca el valor de las resistencias (en K Ω)

R1= R2=

Introduzca el valor de los voltajes (en V)

Vi= Vu= +Vsat= -Vsat=

Vref=

Restaurar valores

Teoría **Calcular**

Inicialmente +Vsat	Cambio para Vi >	V'o = (V)	Vo = (V)
Inicialmente -Vsat	Cambio para Vi <	V'o = (V)	Vo = (V)

Representación Vo-Vi

Ilustración 32. Formulario Histeresis2 Ejemplo 1

Comparador con histéresis y limitación a la salida

Ejemplo 1 | Ejemplo 2

Introduzca el valor de las resistencias (en K Ω)

R1= R2=

Introduzca el valor de los voltajes (en V)

Vi= Vu= +Vsat= -Vsat=

Vref=

Restaurar valores

Teoría **Calcular**

Inicialmente +Vsat	Cambio para Vi >	V'o = (V)	Vo = (V)
Inicialmente -Vsat	Cambio para Vi <	V'o = (V)	Vo = (V)

Representación Vo-Vi

Ilustración 33. Formulario Histeresis2 Ejemplo 2

4.2.1.20. IntegradorInversor

Para mostrar la ventana de los circuitos integradores inversores se emplea el formulario *IntegradorInversor*. Dicho formulario dispone de un elemento *Page* para definir dos páginas, una para cada circuito. En cuanto a la estructura de

las páginas de los circuitos tipo 1 y tipo 2, es exactamente igual salvo un elemento, la tensión de entrada al circuito. Para el tipo 1, la tensión de entrada queda definida por un valor que actúa como ganancia al que se le multiplicará el valor del tiempo elevado a una potencia, valor que podrá introducir el usuario mediante un *TextBox*, al igual que dicha ganancia. Por otro lado, en el circuito tipo 2 se introduce una tensión de entrada en forma de onda, ya sea senoidal o cosenoidal. Para ello, utilizaremos un *TextBox* para definir la amplitud de la onda y otro para el periodo. Para definir si tenemos una onda senoidal o cosenoidal se empleará el elemento *ComboBox*, con el que podremos seleccionar “sen” o “cos”. En ambas páginas tendremos unos *Label* y *TextBox* para definir el valor del tiempo para el que queremos calcular la tensión de salida V_o y el valor de V_o con el que se inicia el circuito. En cuanto al código del objeto, dispondremos de las mismas funciones que se emplean en los casos con elementos *Page* que hemos visto anteriormente, añadiendo una función. Para introducir los elementos “sen” y “cos” en nuestro *ComboBox* se define la función *UserForm_Activate()*. Esto quiere decir que el código que contiene dicha función se activa cuando se abre el formulario. En esta función definimos las sentencias *Nombre_Combobox.AddItem* para “sen” y “cos”, consiguiendo así que en el elemento *ComboBox* se genere una lista con estos dos parámetros, pudiendo seleccionar una de las opciones de la lista.

Ilustración 34. Formulario IntegradorInversor Tipo1

Circuito integrador inversor

Tipo 1 Tipo 2

Introduzca el valor de la resistencia (en K Ω) y el condensador (en μ F) i

R= C=

Introduzca el valor de los voltajes (en V)

Vi= (· t) +Vsat= -Vsat=

Introduzca la condición inicial (en V) y final (en s) del circuito

Para $t < 0$, Vo =

Obtener Vo para t =

[Restaurar valores](#)

Calcular

Vo = (V)

[Representación de Vi y Vo frente a t](#)

Teoría

Ilustración 35. Formulario IntegradorInversor Tipo2

4.2.1.21. IntegradorNoInversor

El formulario *IntegradorNoInversor* nos muestra la ventana de los circuitos integradores no inversores. La forma y el código del formulario es exactamente igual que el formulario *IntegradorInversor*, cambiando la imagen del circuito en el objeto y las llamadas en las funciones del código del objeto, que para este caso se llamara al procedimiento del circuito integrador no inversor.

Circuito integrador no inversor

Tipo 1 Tipo 2

Introduzca el valor de la resistencia (en $K\Omega$) y el condensador (en μF)

R= C=

Introduzca el valor de los voltajes (en V)

$V_i =$ $\cdot t^{\square}$ +Vsat= -Vsat=

Introduzca la condición inicial (en V) y final (en s) del circuito

Para $t < 0$, $V_o =$

Obtener V_o para $t =$

Restaurar valores

Calcular

$V_o =$ (V)

Representación de V_i y V_o frente a t

Teoría

Ilustración 36. Formulario IntegradorNoInversor Tipo1

Circuito integrador no inversor

Tipo 1 Tipo 2

Introduzca el valor de la resistencia (en $K\Omega$) y el condensador (en μF)

R= C=

Introduzca el valor de los voltajes (en V)

$V_i =$ \cdot $\cdot t$ +Vsat= -Vsat=

Introduzca la condición inicial (en V) y final (en s) del circuito

Para $t < 0$, $V_o =$

Obtener V_o para $t =$

Restaurar valores

Calcular

$V_o =$ (V)

Representación de V_i y V_o frente a t

Teoría

Ilustración 37. Formulario IntegradorNoInversor Tipo2

4.2.1.22. IntegradorRestador

El formulario *IntegradorRestador* hace que se visualice la ventana de los circuitos integradores restadores. La estructura del objeto y de su código es igual que para los casos anteriores de los circuitos integradores. La única

diferencia es que se introduce una tensión de entrada a mayores en los dos tipos de circuitos, lo que implica que se añadan más *Label*, *TextBox* y *ComboBox* al objeto. Esto hace también que, en la función de *UserForm_Activate()*, se definan los dos nuevos elementos (“sen” y “cos”) en el nuevo *ComboBox* que se ha introducido.

Circuito integrador restador

Tipo 1 Tipo 2

Introduzca el valor de la resistencia (en $\text{K}\Omega$) y el condensador (en μF)

R= C=

Introduzca el valor de los voltajes (en V)

V1= · t V2= · t +Vsat= -Vsat=

Introduzca la condición inicial (en V) y final (en s) del circuito

Para $t < 0$, $V_o =$

Obtener V_o para $t =$

Restaurar valores

Calcular

$V_o =$ (V)

Teoría

Ilustración 38. Formulario IntegradorRestador Tipo1

Circuito integrador restador

Tipo 1 Tipo 2

Introduzca el valor de la resistencia (en $\text{K}\Omega$) y el condensador (en μF)

R= C=

Introduzca el valor de los voltajes (en V)

V1= · t (· t) V2= · t (· t) +Vsat= -Vsat=

Introduzca la condición inicial (en V) y final (en s) del circuito

Para $t < 0$, $V_o =$

Obtener V_o para $t =$

Restaurar valores

Calcular

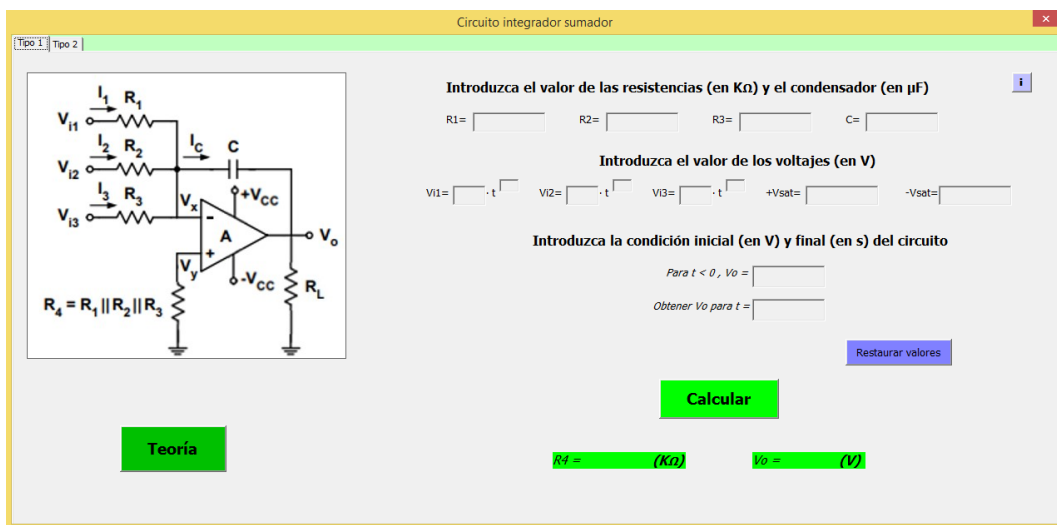
$V_o =$ (V)

Teoría

Ilustración 39. Formulario IntegradorRestador Tipo2

4.2.1.23. IntegradorSumador

IntegradorSumador es el último formulario de los circuitos integradores. En este caso, con este formulario definimos la ventana de los circuitos integradores sumadores. Sucede lo mismo que para el formulario de los circuitos integradores restadores. Se añade una tercera señal de entrada en ambos tipos de circuitos, lo que implica nuevos *Label*, *TextBox* y *ComboBox*. También habrá que definir los parámetros “sen” y “cos” del nuevo *ComboBox* en la función *UserForm_Activate()*.



Circuito integrador sumador

Introduzca el valor de las resistencias (en KΩ) y el condensador (en μF)

R1= R2= R3= C=

Introduzca el valor de los voltajes (en V)

V1= · t V2= · t V3= · t +Vsat= -Vsat=

Introduzca la condición inicial (en V) y final (en s) del circuito

Para $t < 0$, $V_o =$

Obtener V_o para $t =$

Restaurar valores

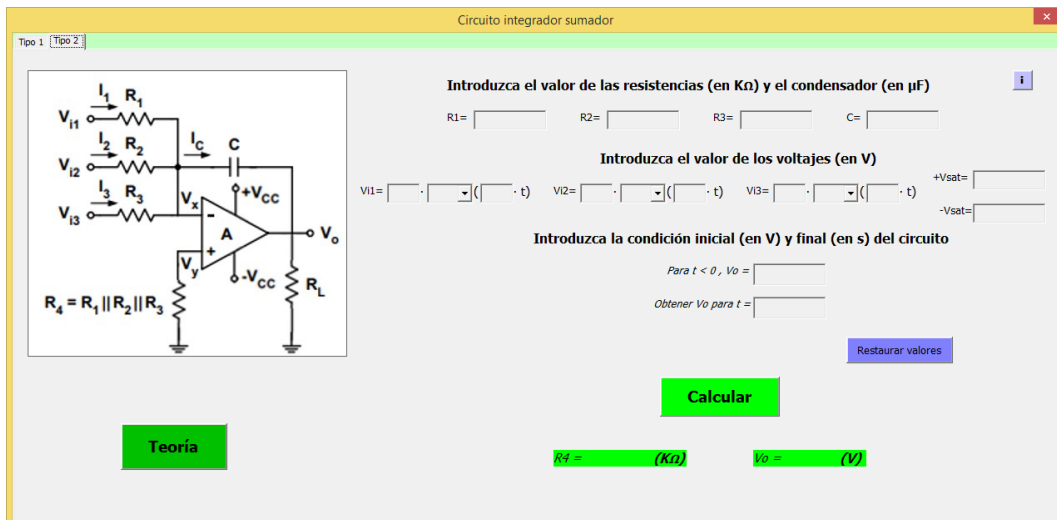
Calcular

Teoría

$R_4 = R_1 \parallel R_2 \parallel R_3$

$R_4 =$ (KΩ) $V_o =$ (V)

Ilustración 40. Formulario IntegradorSumador Tipo1



Circuito integrador sumador

Introduzca el valor de las resistencias (en KΩ) y el condensador (en μF)

R1= R2= R3= C=

Introduzca el valor de los voltajes (en V)

V1= · (· t) V2= · (· t) V3= · (· t) +Vsat= -Vsat=

Introduzca la condición inicial (en V) y final (en s) del circuito

Para $t < 0$, $V_o =$

Obtener V_o para $t =$

Restaurar valores

Calcular

Teoría

$R_4 = R_1 \parallel R_2 \parallel R_3$

$R_4 =$ (KΩ) $V_o =$ (V)

Ilustración 41. Formulario IntegradorSumador Tipo2

4.2.1.24. Inversor

El formulario *Inversor* se emplea para mostrar la ventana del circuito inversor. La estructura del objeto y del código del objeto dispone de elementos que ya se ha comentado en los anteriores casos. *Label* y *TextBox* para definir los valores de las resistencias, los parámetros de tensión que introduce el usuario y para los parámetros que hay que calcular. En cuanto al código, se definirán las funciones para cada botón, además de la función para vaciar las celdas de la hoja de cálculo donde generamos la gráfica.

Ilustración 42. Formulario Inversor

4.2.1.25. LimitadorDiodoNormal

LimitadorDiodoNormal es el formulario con el que conseguimos mostrar la ventana de los circuitos limitadores con diodo normal. Se introduce un elemento *Page* para definir dos páginas, una para cada ejemplo de estos circuitos. A los *Label* y *TextBox* de los que se definen en los formularios anteriores añadimos unos *Label* donde se indican si el diodo está en conducción o no, si está polarizado en directa o en inversa y si nos encontramos en la zona lineal o saturada de la representación gráfica. En cuanto al código, se definen las funciones para cada botón y la función *UserForm_Terminate()* para vaciar las celdas de la hoja de cálculo.

Circuito limitador con diodo normal

Ejemplo 1 | Ejemplo 2

Introduzca el valor de las resistencias (en $K\Omega$)

R1= R2=

Introduzca el valor de los voltajes (en V)

Vi= Vu= +Vsat= -Vsat=

Calcular

Restaurar valores

Ganancia (A) = Vo = (V)

Teoría

Representación Vo-Vi

Ilustración 43. Formulario LimitadorDiodoNormal Ejemplo 1

Circuito limitador con diodo normal

Ejemplo 1 | Ejemplo 2

Introduzca el valor de las resistencias (en $K\Omega$)

R1= R2=

Introduzca el valor de los voltajes (en V)

Vi= Vu= +Vsat= -Vsat=

Calcular

Restaurar valores

Ganancia (A) = Vo = (V)

Teoría

Representación Vo-Vi

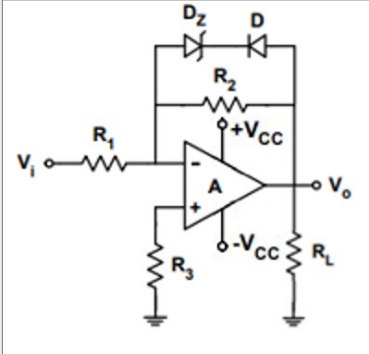
Ilustración 44. Formulario LimitadorDiodoNormal Ejemplo 2

4.2.1.26. LimitadorDiodoNormalDiodoZener

El formulario *LimitadorDiodoNormalDiodoZener* muestra la ventana de los circuitos limitadores con diodo normal y diodo Zener en serie. Sigue la misma estructura que el formulario anterior, pero añadiendo las etiquetas y *TextBox* referidas al diodo Zener y su tensión V_z . El código también dispone de la misma estructura que el caso anterior.

Circuito limitador con diodo normal y diodo Zener en serie

Ejemplo 1 | Ejemplo 2



Introduzca el valor de las resistencias (en K Ω)

R1= R2=

Introduzca el valor de los voltajes (en V)

Vi= Vu= +Vsat= -Vsat=

Vz=

Calcular [Restaurar valores](#)

Ganancia (A) = Vo = (V)

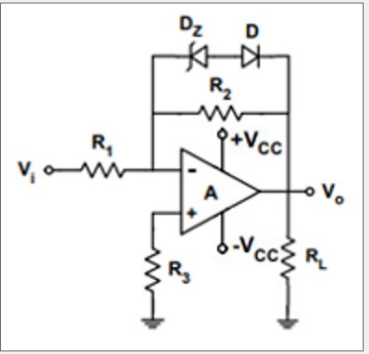
[Representación Vo-Vi](#)

[Teoría](#)

Ilustración 45. Formulario LimitadorDiodoNormalDiodoZener Ejemplo 1

Circuito limitador con diodo normal y diodo Zener en serie

Ejemplo 1 | Ejemplo 2



Introduzca el valor de las resistencias (en K Ω)

R1= R2=

Introduzca el valor de los voltajes (en V)

Vi= Vu= +Vsat= -Vsat=

Vz=

Calcular [Restaurar valores](#)

Ganancia (A) = Vo = (V)

[Representación Vo-Vi](#)

[Teoría](#)

Ilustración 46. Formulario LimitadorDiodoNormalDiodoZener Ejemplo 2

4.2.1.27. LimitadorDosZener

El formulario *LimitadorDosZener* muestra la ventana del circuito limitador con dos diodos Zener en serie. En este caso, no se introduce un elemento Page al disponer solo un ejemplo de dicho circuito. A partir de los *Label* y *TextBox* que se vienen repitiendo en los anteriores casos, se introducen nuevos *Label* y

TextBox para definir los dos diodos Zener, la tensión umbral V_u de ambos, que en este caso será común, y las tensiones Zener de cada uno V_{z1} y V_{z2} . El código sigue siendo la misma estructura que se viene repitiendo.

Ilustración 47. Formulario LimitadorDosZener

4.2.1.28. NoInversor

El formulario *NoInversor* nos muestra la ventana del circuito no inversor. La estructura del objeto y del código del objeto es exactamente igual que el formulario *Inversor*, pero modificando la imagen del circuito y las llamadas de las funciones de los botones, en la que se llamará al procedimiento del circuito no inversor en vez del circuito inversor.

Circuito no inversor

Introduzca el valor de las resistencias (en $K\Omega$)

R1= R2=

Introduzca el valor de los voltajes (en V)

Vi= +Vsat= -Vsat=

Restaurar valores

Calcular

R3 = ($K\Omega$) Ganancia (A) = Vo = (V)

Teoría

Representación Vo-Vi

Ilustración 48. Formulario Nolnversor

4.2.1.29. OndaCuadrada

El formulario *OndaCuadrada* nos muestra la ventana del circuito generador de onda cuadrada. El objeto dispone de elementos *Label* y *TextBox* para definir los parámetros necesarios. Para este caso, a la salida buscamos solo el valor del periodo y la frecuencia de la onda generada, por lo que los *Label* y *TextBox* referidos a la salida definirán estos valores. El código del formulario dispone de las funciones que se han comentado en los apartados anteriores.

Generador de onda cuadrada

Introduzca el valor de las resistencias (en $K\Omega$) y el condensador (en μF)

R= R1= R2= C=

Introduzca el valor de los voltajes (en V)

+Vsat= -Vsat=

Restaurar valores

Calcular

T = (s) f = (KHz)

Teoría

Representación Vo-t

Ilustración 49. Formulario OndaCuadrada

4.2.1.30. OndaTriangular

OndaTriangular es el formulario empleado para mostrar la ventana del circuito generador de onda triangular. La forma y el código del objeto es exactamente igual que el formulario *OndaCuadrada*, cambiando el circuito del elemento *Image* y las llamadas de las funciones del código, que llamarán al procedimiento del circuito generador de onda triangular.

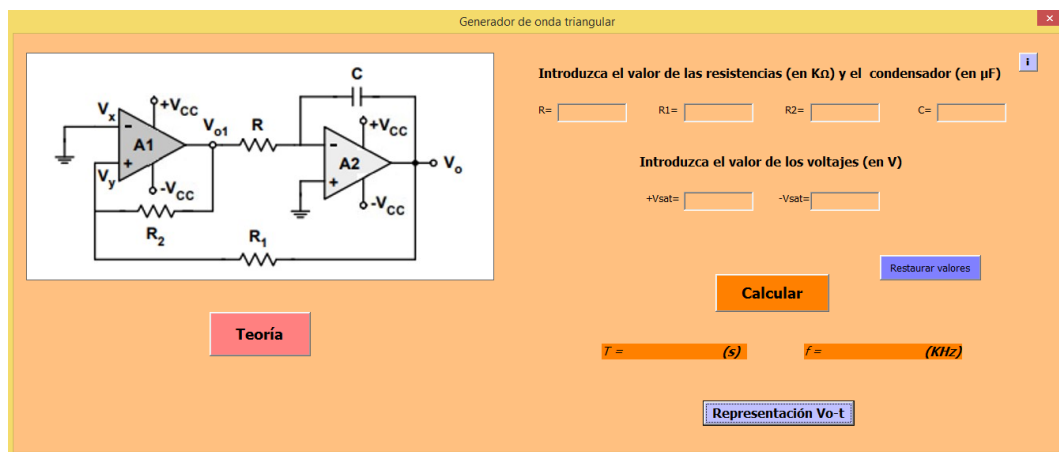


Ilustración 50. Formulario OndaTriangular

4.2.1.31. RectMediaOnda

El formulario *RectMediaOnda* nos muestra la ventana del circuito rectificador de media onda. En cuanto a la forma del objeto, nos basamos de la estructura que se viene repitiendo en todos los formularios. Cabe comentar que, para definir la tensión de entrada del circuito, se definen unos *Label* y *TextBox* para definir una tensión senoidal, introduciendo en los *TextBox* la amplitud y el periodo de la onda. Debajo de *Vu* se define un *Label* para indicar si los diodos conducen o no. En la salida se indica mediante los *Label* el valor de la amplitud de la tensión de salida y qué semionda de *Vi* no es rectificadas. El código del objeto dispondrá de las funciones referidas a los botones del formulario y de la función para vaciar las celdas de las hojas de cálculo, en este caso dos hojas de cálculo, ya que se generan las gráficas *Vi-t* y *Vo-t*.

Rectificador de media onda

Introduzca el valor de las resistencias (en K Ω)

R1= R2=

Introduzca el valor de los voltajes (en V)

Vi= · sen(· t) Vu= +Vsat= -Vsat=

Restaurar valores

Calcular

Amplitud de la salida Vo de Vi = (V)

Representación de Vi y Vo frente a t

Ilustración 51. Formulario RectMediaOnda

4.2.1.32. RectOndaCompleta

RectOndaCompleta es el formulario con el que se consigue mostrar la ventana con los dos circuitos rectificadores de onda cuadrada. Al disponer de dos circuitos, se necesita de un elemento *Page* para poner los ejemplos en páginas separadas. En cada una de las páginas, los *Label* y *TextBox* son bastante parecidos, con la diferencia de que varía el número resistencias en las que tenemos que introducir un valor. Para el caso del ejemplo 2, solo tendremos que introducir dos resistencias debido a la simplificación que se consigue. En los *Label* de salida se indica la amplitud de V_o que se consigue para la semionda positiva y negativa de V_i . Para el ejemplo 2, la amplitud será igual para ambas semiondas debido a la simplificación que se ha comentado antes. El código sigue la misma estructura con la diferencia de que tendremos el doble de funciones referidas a los botones del objeto, ya que tenemos dos páginas.

Ilustración 52. Formulario RectOndaCompleta Ejemplo 1

Ilustración 53. Formulario RectOndaCompleta Ejemplo 2

4.2.1.33. RestadorAltImp

El formulario *RestadorAltImp* nos permite mostrar la ventana del circuito restador con alta impedancia a la entrada. Definimos los *Label* y *TextBox* necesarios para los valores de las impedancias y tensiones. Los parámetros que calculamos los definimos en unas etiquetas con fondo en verde. Para este caso, no se hace uso del botón de representación gráfica, pues no se necesita

dicha información para el circuito. El código del objeto estará compuesto por las funciones referidas a los botones.

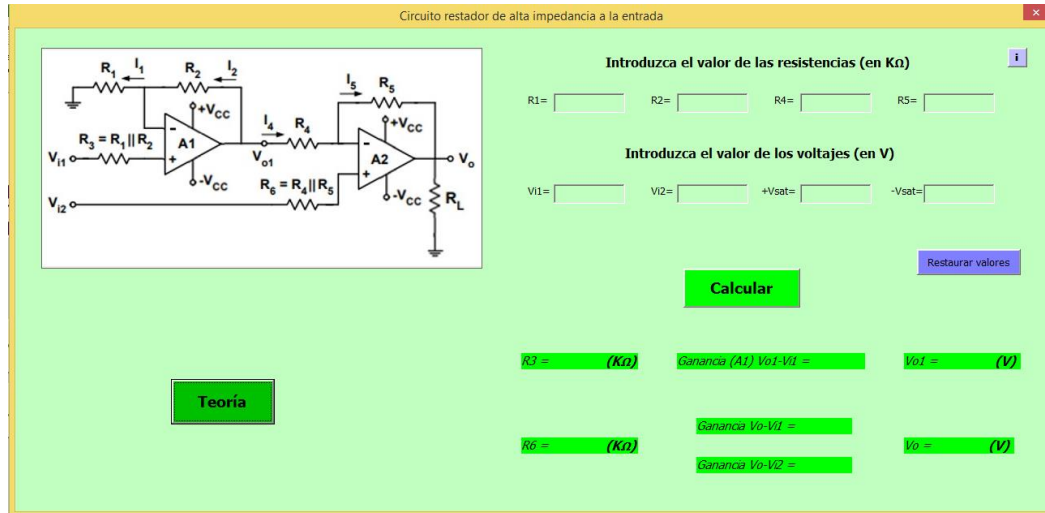


Ilustración 54. Formulario RestadorAltaImp

4.2.1.34. RestadorSimple

RestadorSimple es el formulario con el que se muestra la ventana del circuito restador simple. Seguimos con elementos *Label* y *TextBox* para definir los parámetros necesarios para calcular los valores de salida y unas etiquetas para dichas salidas. La estructura del código es la misma que para el caso anterior.

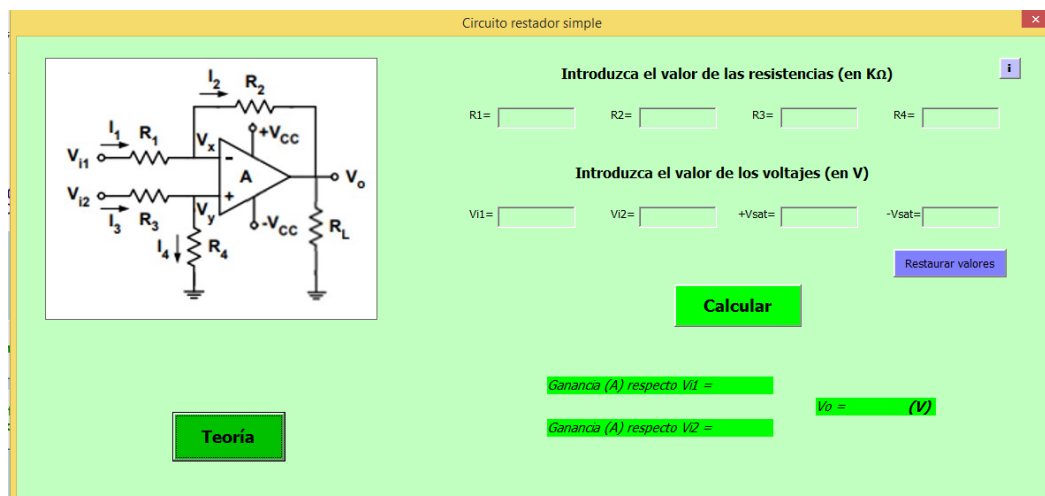


Ilustración 55. Formulario RestadorSimple

4.2.1.35. SumadorInversor

El formulario *SumadorInversor* corresponde con el circuito sumador inversor. Se sigue la misma estructura de los casos anteriores en cuanto a los elementos *Label* y *TextBox*. El código define las funciones de los botones que incluye el objeto.

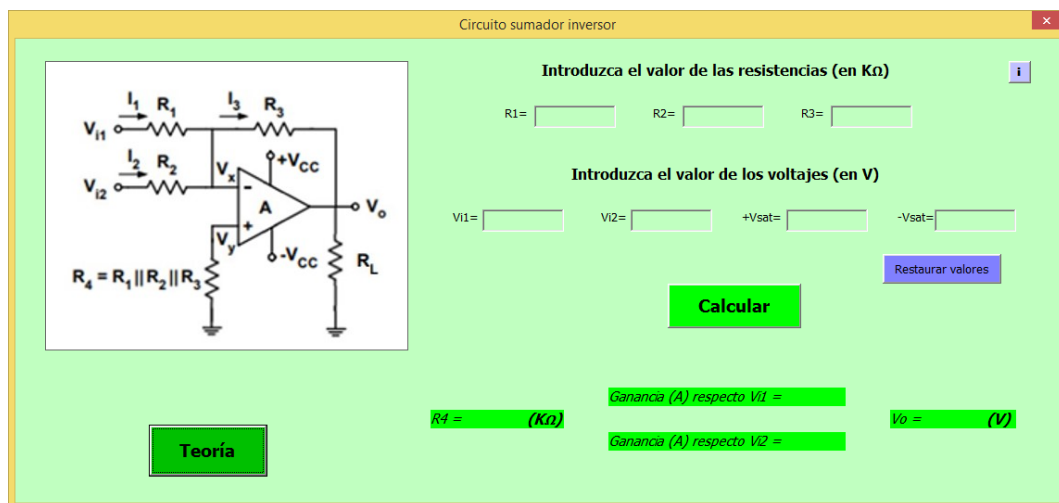


Ilustración 56. Formulario SumadorInversor

4.2.1.36. SumadorNoInversor

SumadorNoInversor es el formulario con el que se consigue mostrar la ventana del circuito sumador no inversor. Los *Label* y *TextBox* son exactamente los mismos que en el caso del *SumadorInversor*, salvo que para este caso no se define la etiqueta para mostrar el valor calculado de R_4 si no que se introduce un *TextBox* para introducir dicho valor como dato. El código dispone de las mismas funciones que el caso anterior, pero llamando al procedimiento del circuito sumador no inversor.

Circuito sumador no inversor

Introduzca el valor de las resistencias (en K Ω)

R1= R2= R3= R4=

Introduzca el valor de los voltajes (en V)

V1= V2= +Vsat= -Vsat=

Restaurar valores

Calcular

Teoría

Ganancia (A) respecto V1 =

Ganancia (A) respecto V2 =

Vo = (V)

Ilustración 57. Formulario SumadorNoInversor

4.2.1.37. ValorMedioValorEficaz

El último formulario de este tipo es *ValorMedioValorEficaz*, el cual nos muestra la ventana del circuito para obtener el valor medio y el valor eficaz. Debido a las necesidades del circuito, se introducen nuevos *Label* y *TextBox* para definir los parámetros del máximo rizado y la ganancia que se obtiene a la salida del circuito. También será necesario definir el valor de R y la frecuencia mínima a la que trabaja el circuito. La tensión de entrada será una onda en la que, mediante *TextBox*, definimos su amplitud y periodo. Los *Label* donde se mostrarán los valores calculados serán para R5, C y la tensión de salida Vo. Al lado de la etiqueta del valor de Vo se definirá otra, donde se mostrará si dicha salida es el valor medio o el valor eficaz. Si no es ninguno de los dos, esta etiqueta no se mostrará en el formulario. En cuanto al código del objeto, se define una función para cada botón que dispone el formulario y la función *UserForm_Terminate()*, con la que vaciaremos las celdas de las hojas de cálculo que se han utilizado para generar las gráficas. En este caso, tendremos que vaciar celdas de dos hojas de cálculo, ya que se generan dos gráficas; Vi frente al tiempo y Vo frente al tiempo.

Circuitos de valor medio y valor eficaz

Introduzca el rizado máximo (en %) y la ganancia (G)

máximo rizado= G=

Introduzca el valor de la resistencia (en K Ω) y la frecuencia (en Hz)

R= freq. mínima=

Introduzca el valor de los voltajes (en V)

Vi= sen(· t) +Vsat= -Vsat=

Restaurar valores

Calcular

Teoría

R5 = (K Ω) C = (μ F) V θ = (V) \rightarrow V θ =

Representación de Vi y Vo frente a t

Ilustración 58. Formulario ValorMedioValorEficaz

III: El tercer tipo de formulario es el que corresponde con las pestañas que muestran las gráficas de los circuitos. Debido a que ciertos circuitos necesitan sus propias gráficas, se hace necesario realizar varios *UserForm*. Desde el apartado 4.2.1.38. hasta el 4.2.1.46. se muestran todos ellos. Los formularios referidos a los circuitos con aplicaciones lineales tendrán un fondo verde, mientras que los circuitos con aplicaciones no lineales tendrán un tono anaranjado.

4.2.1.38. GraficaCosSen

El formulario *GraficaCosSen* representa las gráficas de los circuitos tipo 2 del integrador inversor y del integrador no inversor. Debido a que en ambos circuitos se busca representar la tensión de entrada y salida respecto al tiempo, se introducen dos elementos *Image* en el formulario. Además de estos dos elementos, se introducen varios elementos *Label*, que son unas etiquetas en la que se introduce texto. Como se puede apreciar en la ilustración del formulario, hay ciertas etiquetas en las que ya se ha introducido texto, mientras que otras se encuentran vacías esperando a que se ejecute el código para que puedan ser rellenadas.

Ilustración 59. Formulario GraficaCosSen

Las etiquetas nos servirán para comprender mejor las gráficas que se muestren en las imágenes. Se indica cuales son los ejes en cada una de las gráficas y su función de transferencia.

En cuanto al desarrollo interno, el formulario se inicia mediante la función `UserForm_Initialize()`, es decir, que una vez abierto el formulario, el código se ejecuta. El funcionamiento del mismo se basa en una operación `If - Elseif - End If`. En cada uno de los `if` se establece la condición para cada uno de los circuitos. Si la pestaña del circuito está abierta y se ha obtenido un valor de la salida, ejecutamos su código correspondiente. En este código, se realizan sentencias para recoger valores, ya sea a partir de los valores que se han introducido en la pestaña del circuito, o de las celdas de la hoja de cálculo en la que se ha generado la gráfica. Estos valores que obtenemos son los que se introducen en las etiquetas vacías para mostrarlos en el formulario. En cuanto a generar las gráficas y mostrarlas en el formulario utilizamos la función `Set`, con la que conseguimos almacenar en nuestro elemento `Image` la gráfica que se genera en la hoja de cálculo correspondiente. Al tener dos elementos `Image`, necesitamos que las gráficas provengan de dos hojas de cálculo distintas para que no tengamos problemas. Para la representación de Vi-t usamos la hoja de cálculo “Gráfica Vo-t” y para la de Vo-t la hoja de cálculo “Gráfica CosSen”.

4.2.1.39. GraficaDetectPico

Este formulario se emplea para mostrar la gráfica resultante del circuito detector de pico exclusivamente. Este es un caso particular, pues la gráfica no va acompañada de ninguna etiqueta para mostrar la función de transferencia o los valores que se consiguen con el circuito. La única información que nos interesa es ver cómo va evolucionando los valores de pico que va generando el circuito con respecto al tiempo.

La estructura interna es la misma que para el anterior caso. Se establece la función *UserForm_Initialize()* seguido del código *If - End If*, siendo la condición que la ventana del circuito detector de pico esté abierta. Si esto se cumple, se captura la gráfica generada en la hoja de cálculo y se muestra por el elemento *Image*.

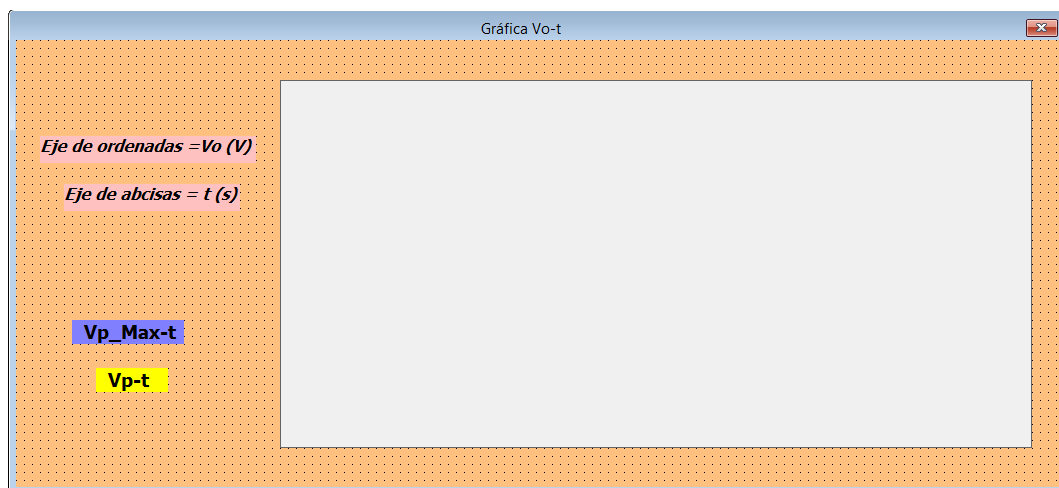


Ilustración 60. Formulario GraficaDetectPico

4.2.1.40. GraficaDif

El formulario *GraficaDif* tiene la función de mostrar la representación gráfica de V_i-t y V_o-t del circuito Diferenciador. Se dispone de dos elementos *Image* por la que se muestran ambas gráficas y varios elementos *Label*. Con la gráfica V_i-t se tienen varias etiquetas para mostrar la función de transferencia, mientras que las etiquetas que acompañan a la gráfica V_o-t (imagen inferior del formulario) están superpuestas las unas con las otras. Esto es debido a la situación que se plantea si la potencia a la que está elevado el tiempo en la

ecuación de la tensión de entrada tiene valor uno. Si esto sucede, la salida será un valor constante. Para este caso tendremos unas etiquetas para mostrar la salida constante. Por otro lado, si este índice es mayor que 1, la salida dependerá del tiempo, de tal forma que necesitaremos otras etiquetas para mostrar la ecuación de salida en función del tiempo. Para cada caso, haremos que las etiquetas que se necesiten estén visibles y la del caso contrario no lo estén. Esto lo conseguiremos poniendo en el código del formulario la sentencia *GraficaDif.variable.Visible = True* o *GraficaDif.Variable.Visible = False*.

El código del formulario tendrá la misma estructura que los casos anteriores añadiendo las nuevas sentencias de código que se han comentado este apartado.



Ilustración 61. Formulario GraficaDif

4.2.1.41. GraficaHisteresis

GraficaHisteresis es el formulario empleado para mostrar las representaciones gráficas de los circuitos comparadores con histéresis. En este formulario se hace uso de un elemento *Image*, donde se representa V_o frente a V_i . En la gráfica se mostrarán dos líneas de representación, en función de si el valor de V_o es inicialmente $+V_{sat}$ o es $-V_{sat}$, tal y como se muestra en las etiquetas amarillas. También se mostrará en forma de etiquetas el valor de la banda de

histéresis y los valores de A y B, que son los valores de V_i para los que la tensión pasa de $+V_{sat}$ a $-V_{sat}$ y viceversa.

En el caso de circuitos comparadores con histéresis tenemos tres tipos; comparador con histéresis simple y dos casos de comparador con histéresis y limitación a la salida. Debido a esto, la estructura interna se basa en un código *If - Elseif - Elseif - End If*. En cada uno de los casos, la condición para que se ejecute el código correspondiente será que la pestaña del circuito este abierta y que se haya obtenido un valor de V_o . Para el comparador con histéresis simple con que este visible la ventana de dicho circuito será suficiente. El código correspondiente a cada caso será el pasar los valores a las etiquetas y almacenar la gráfica en la imagen del formulario.

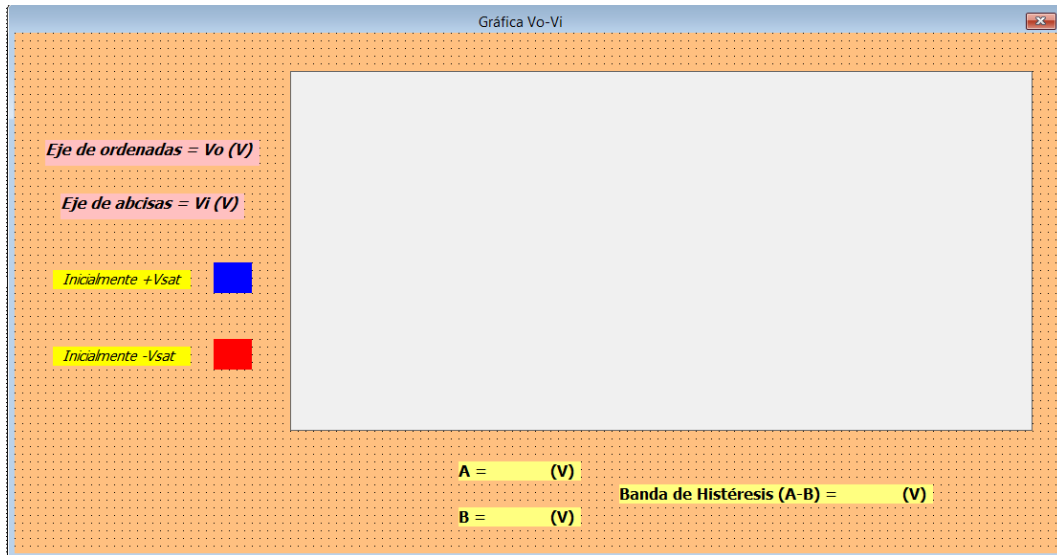


Ilustración 62. Formulario GraficaHisteresis

4.2.1.42. GraficaSchmitt

El formulario *GraficaSchmitt* es utilizado por los siguientes circuitos; disipador de Schmitt, rectificador de media onda, los dos ejemplos del rectificador de onda completa y el circuito para el valor medio y el valor eficaz. Para este *UserForm* tenemos la situación de que para cada circuito tenemos sus propias etiquetas, pues cada uno de ellos completan las gráficas con sus datos de mayor interés. De nuevo, tendremos dos elementos *Image* para representar las gráficas V_i -t y V_o -t. El código interno del objeto seguirá la misma estructura que los casos anteriores, recogiendo los valores del formulario del circuito

correspondiente o de la hoja de cálculo donde se estén ejecutando la gráfica. Para la representación de las gráficas se ejecutará la sentencia Set.

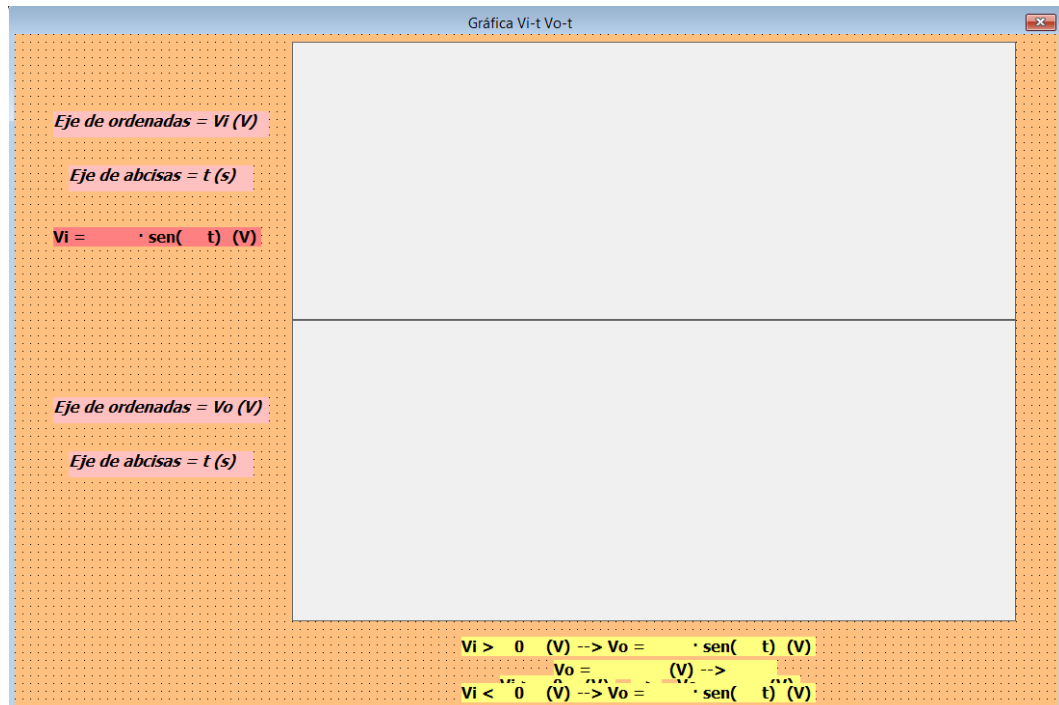


Ilustración 63. Formulario GraficaSchmitt

4.2.1.43. GraficaVot

GraficaVot es el formulario encargado de representar las gráficas de los circuitos integradores inversores y no inversores tipo 1. Se representarán la tensión de entrada y de salida en función del tiempo en gráficas separadas. Acompañando a las gráficas se encuentran unas etiquetas en las que se definen la función de transferencia de cada una de ellas, y los valores de saturación en el caso de la tensión de salida. En el código vendrán las sentencias necesarias para pasar los valores que correspondan a las etiquetas en cada caso y la función Set para almacenar en la imagen del formulario las gráficas generadas en las hojas de cálculo correspondientes.



Gráfica Vi-t Vo-t

Eje de ordenadas = Vi (V)

Eje de abcisas = t (s)

Vi = . t^

Eje de ordenadas = Vo (V)

Eje de abcisas = t (s)

Entre t = 0 y t =

Vo = . t^

Valor de t para el que Vo satura:

= (V) -> t = (s)

Ilustración 64. Formulario GraficaVot

4.2.1.44. GraficaVot_2

El formulario *GraficaVot_2* es el que nos muestra la gráfica Vo-t de los circuitos generadores de onda cuadrada y triangular. De estos circuitos nos interesan únicamente el periodo y la frecuencia de las ondas generadas. En el código se establece la estructura *If - Elseif - End If* para diferenciar las sentencias de cada uno de los circuitos, siendo la condición que la ventana del circuito correspondiente esté abierta. El valor de la frecuencia y el periodo lo recogemos del formulario del circuito correspondiente, y mediante la función *Set* mostraremos la gráfica que se ha ido generando en la hoja de cálculo “*Ondas*”.

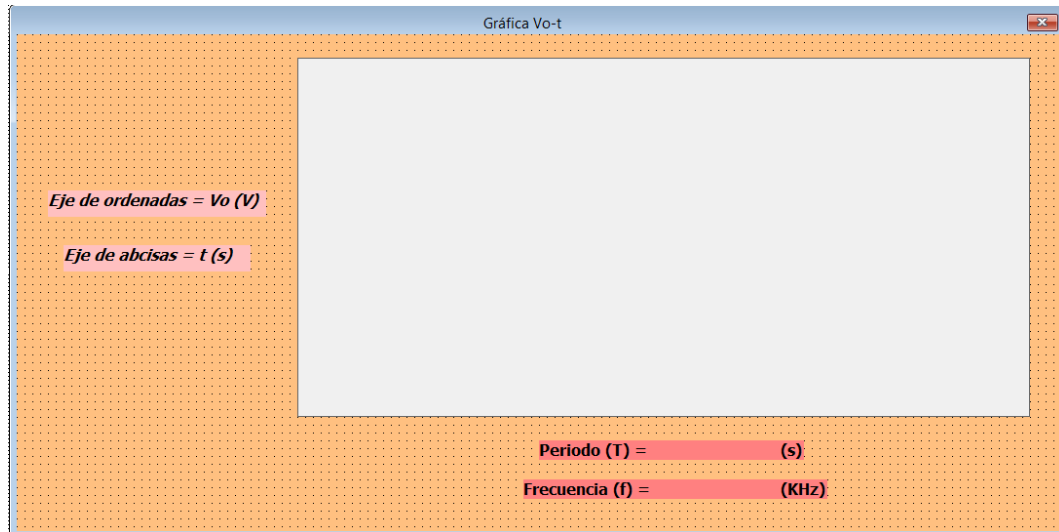


Ilustración 65. Formulario GraficaVot_2

4.2.1.45. GraficaVoVi

El formulario *GraficaVoVi* se emplea para la representación gráfica de V_o - V_i para los circuitos inversor y no inversor. Para ello, empleamos un único elemento *Image* y varios elementos *Label* donde definiremos la función de transferencia de V_o frente a V_i y los valores de V_i para los que V_o satura.

La estructura interna del objeto se basa en el ya visto *If - Elseif - End If*. Para cada circuito la condición de que se ejecute el código es que el formulario del circuito correspondiente esté visible. Si se cumple esto, pasamos los valores definidos en la hoja de cálculo “*Gráfica Vo-Vi*” a las etiquetas correspondientes y asignamos la gráfica generada en dicha hoja de cálculo a la imagen del formulario.

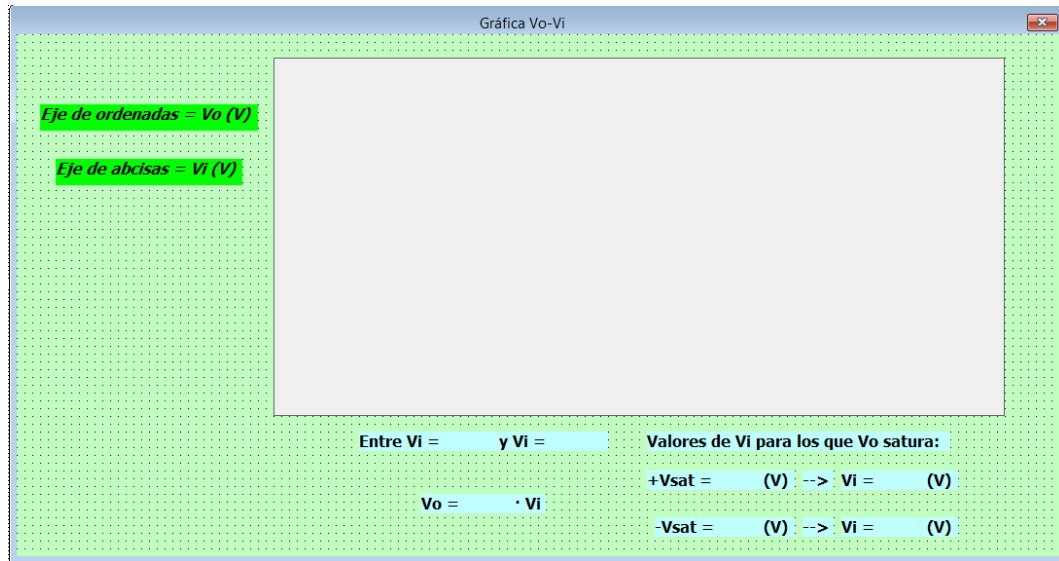


Ilustración 66. Formulario GráficaVoVi

4.2.1.46. GráficaVoVi_2

Para terminar con este tipo de formularios, analizamos el formulario *GráficaVoVi_2*, donde se implementarán las gráficas Vo-Vi de los siguientes circuitos; los dos ejemplos del circuito limitador con diodo normal, los dos ejemplos del circuito limitador con diodo normal y diodo Zener en serie, el circuito limitador con dos diodos Zener en serie, el circuito comparador simple, el circuito comparador con salida sat-sat y los dos ejemplos del circuito comparador con salida cero-sat. El objeto está formado por un elemento *Image* y varias etiquetas. En este caso, tendremos varias etiquetas para cada circuito, por lo que las tendremos superpuestas. Por lo tanto, en el código tendremos que establecer qué variables están visibles y cuales no en cada uno de los circuitos. A mayores, y como en el resto de formularios de este tipo, definiremos el valor de cada una de las etiquetas a partir de los valores generados en la hoja de cálculo “Gráfica Vo-Vi” y definiremos la gráfica generada en dicha hoja de cálculo en nuestra imagen del formulario.

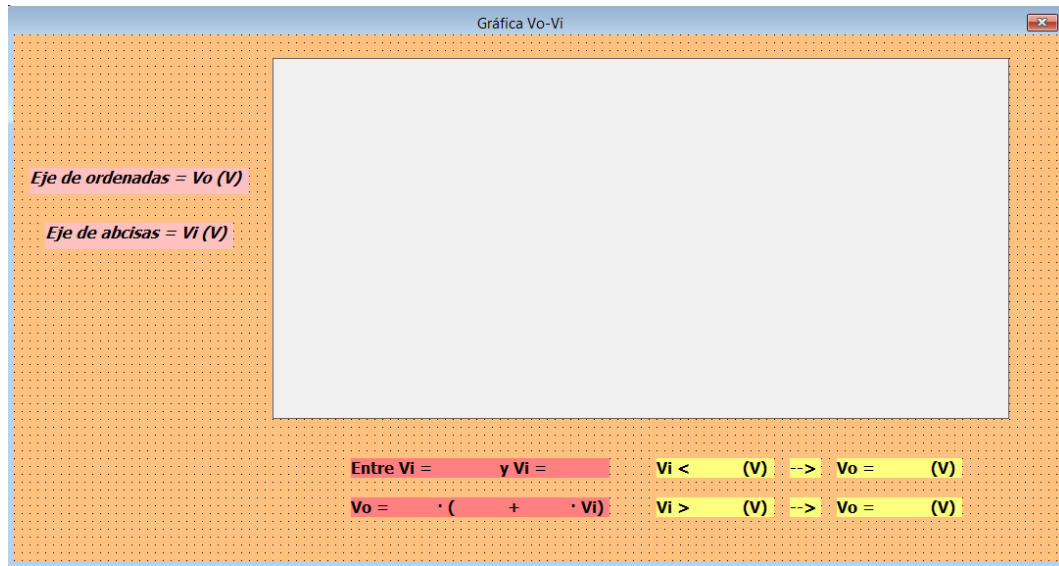


Ilustración 67. Formulario GraficaVoVi_2

IV: Para acabar, mostramos el último tipo de formulario que se puede apreciar en la aplicación. Este tipo de formulario no dispone de código en su estructura interna, pues estos formularios solo sirven para mostrar una ventana con información únicamente. El primer caso de este tipo de formulario es para aportar información acerca del uso de la aplicación en las ventanas del circuito en concreto. Como hemos visto anteriormente, las ventanas de cada circuito disponen de un botón azul oscuro con la letra “i”. A partir de este botón llegamos a la ventana que te proporciona la información necesaria para utilizar correctamente la aplicación. Estos *UserForm* contienen únicamente elementos *Label* donde se introducen los datos necesarios para cada circuito. A continuación, se muestran cada uno de los formularios de la aplicación.

4.2.1.47. Información

Este formulario se usa para el circuito inversor, el circuito no inversor, el limitador con dos diodos Zener, el comparador simple, el detector con salida sat-sat, el detector con salida cero-sat y el comparador con histéresis simple.

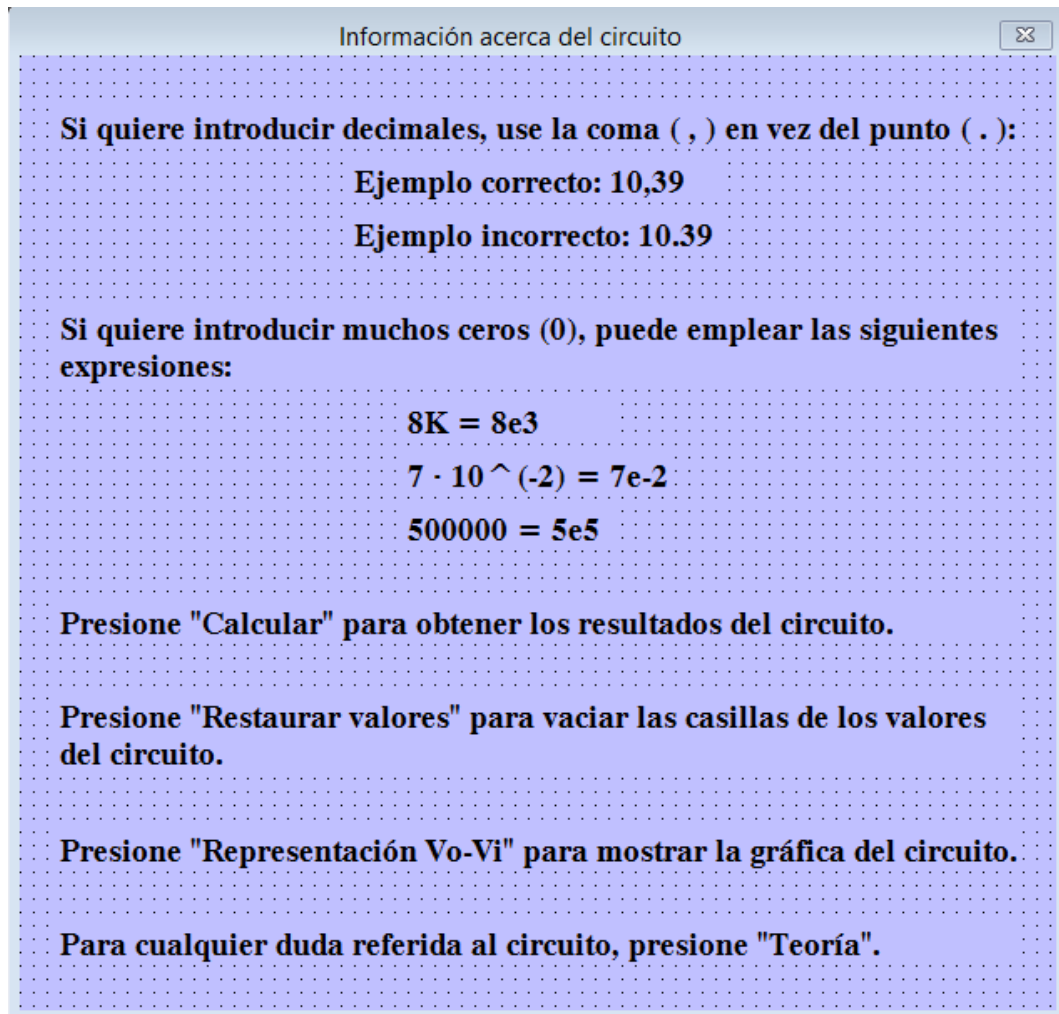


Ilustración 68. Formulario Informacion

4.2.1.48. Informacion1

Este *UserForm* de información se utiliza para los circuitos de aplicaciones lineales en los que no se muestran representación gráfica, es decir, los circuitos restadores, los circuitos sumadores, las fuentes de corriente y de tensión y los circuitos convertidores.

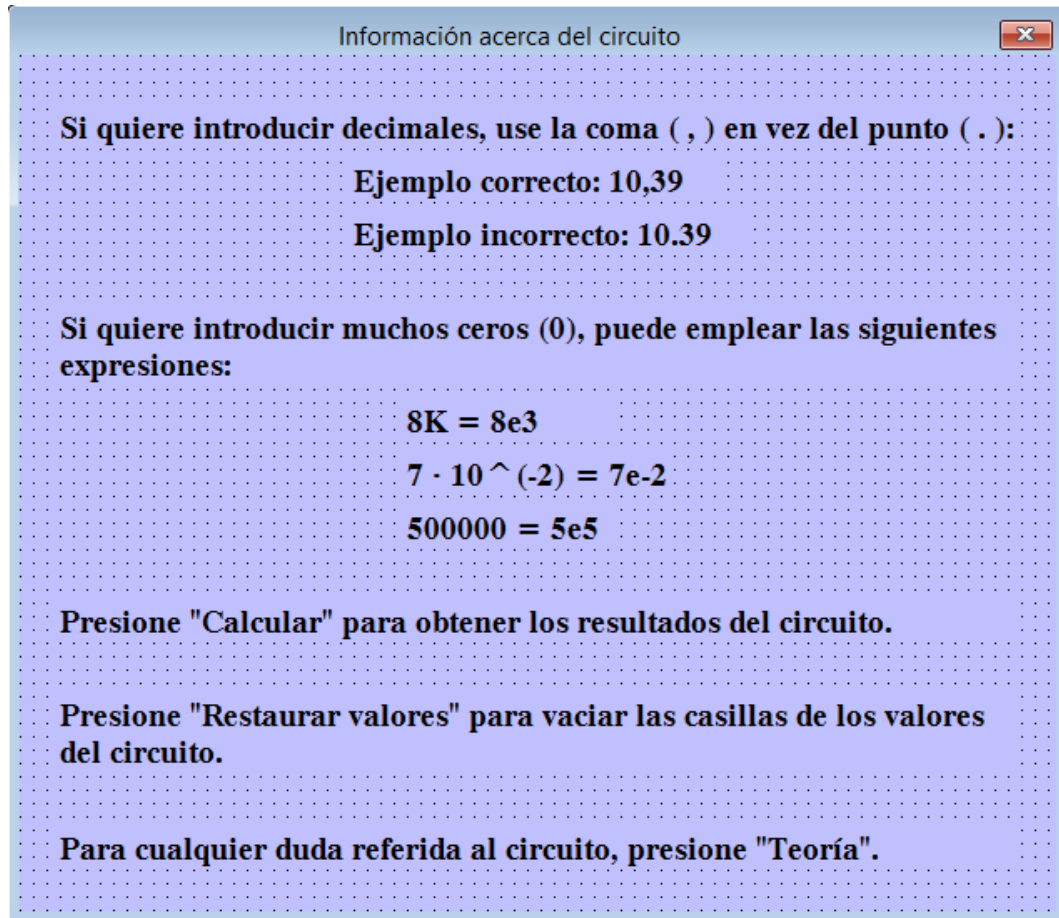


Ilustración 69. Formulario Informacion1

4.2.1.49. Informacion2

Este *UserForm* está dedicado únicamente al circuito detector de pico, ya que es el único circuito que dispone de un interruptor.

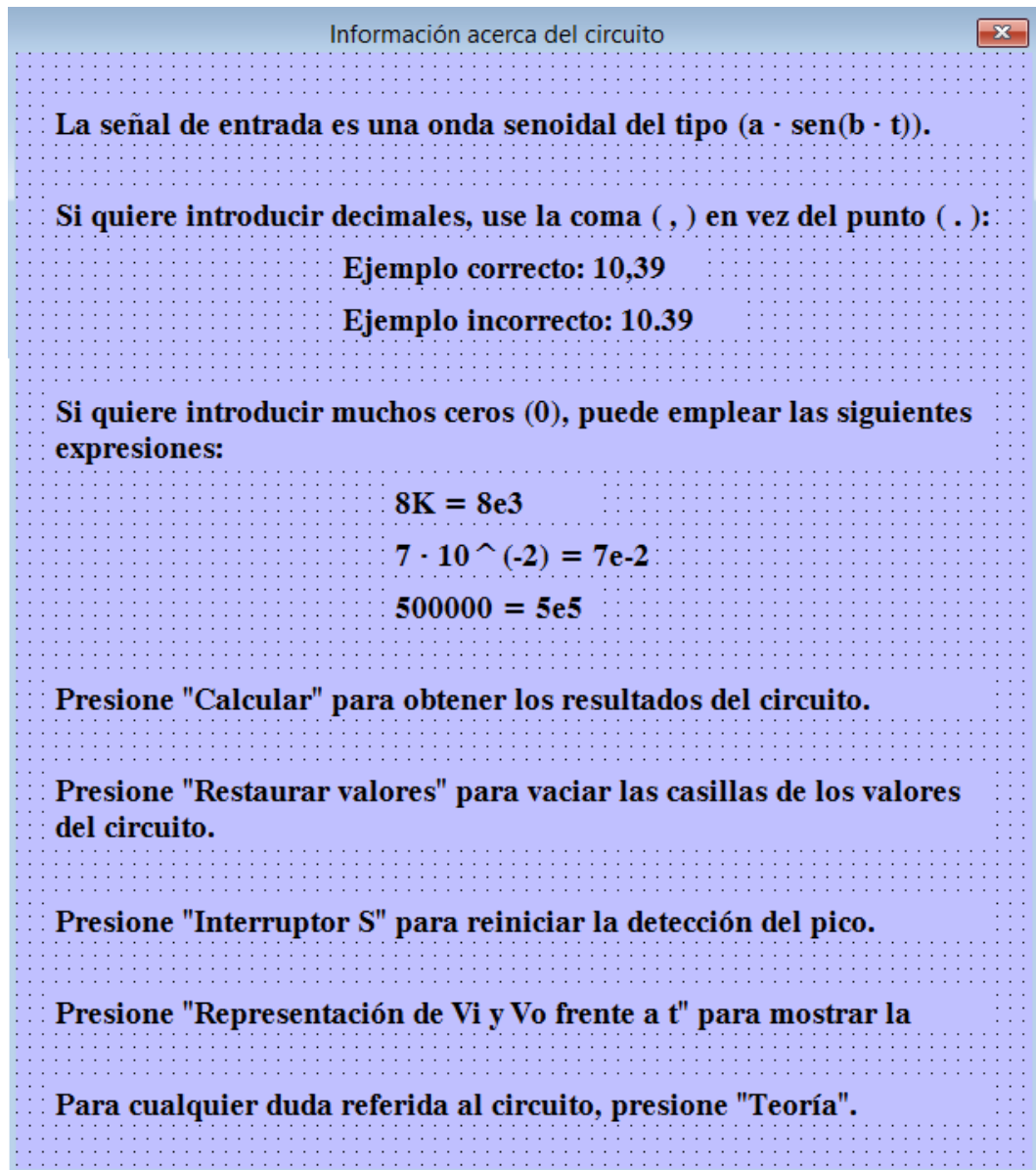


Ilustración 70. Formulario Informacion2

4.2.1.50. Informacion3

Este formulario será utilizado para los circuitos disipador de Schmitt, el rectificador de media onda y el circuito para la obtención del valor medio y el valor eficaz.

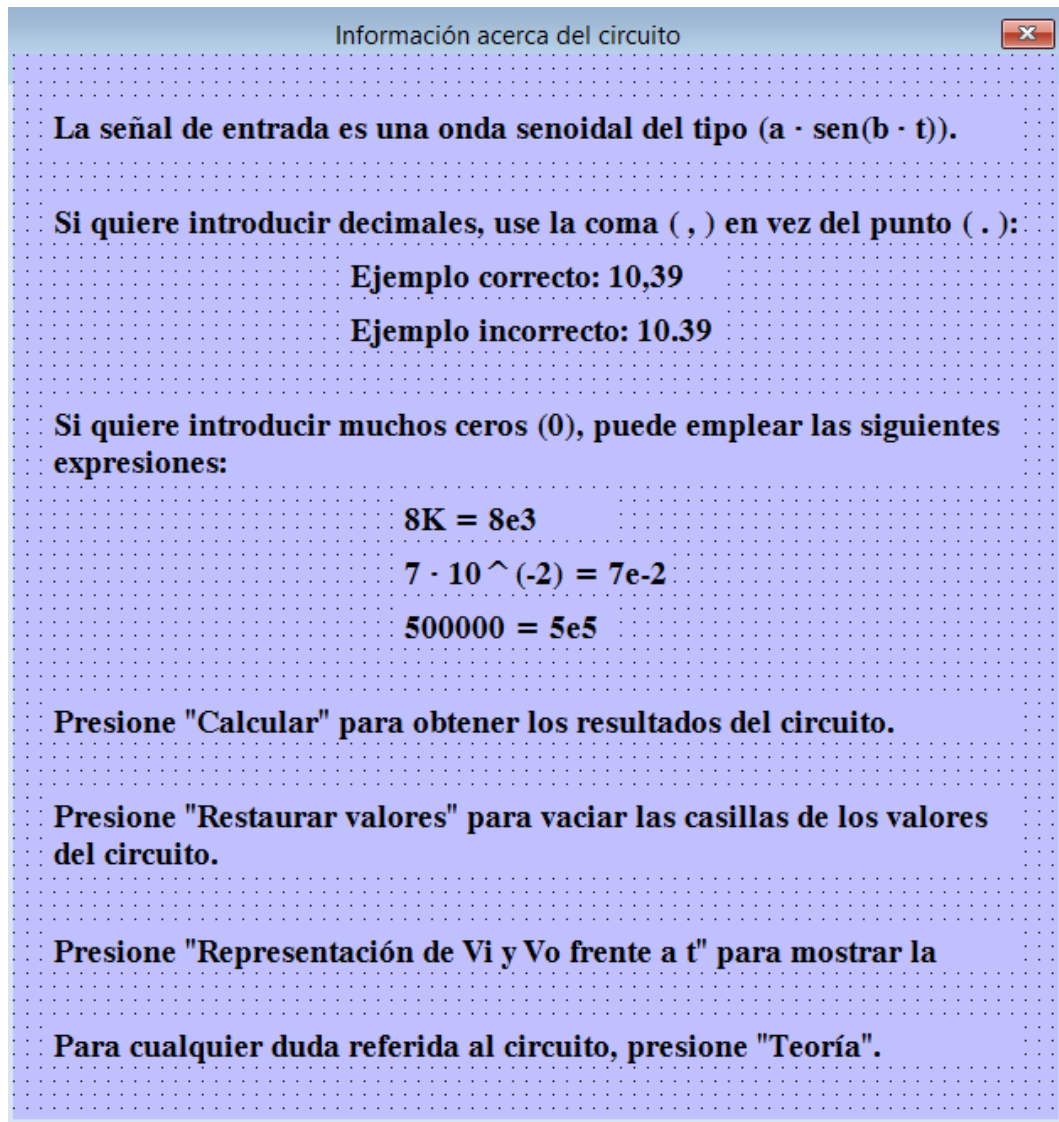


Ilustración 71. Formulario Informacion3

4.2.1.51. Informacion4

En los circuitos limitador con diodo normal, limitador con diodo normal más diodo Zener y comparadores con histéresis y limitación a la salida, al haber dos casos de cada circuito, se debe borrar los valores del circuito que se ha calculado si queremos utilizar el otro tipo de circuito. Para ello, se realiza un *UserForm* propio para ellos.

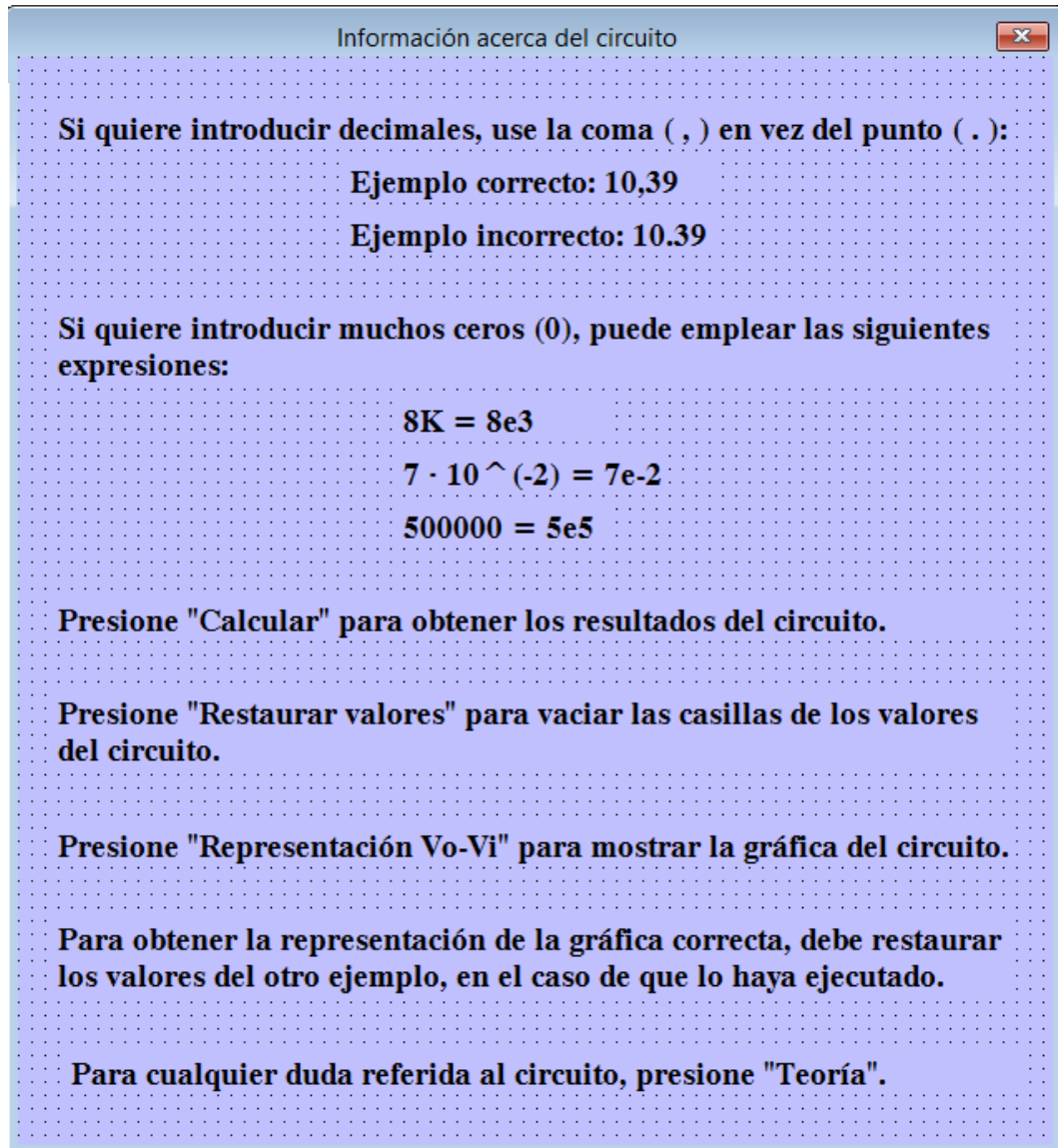


Ilustración 72. Formulario Informacion4

4.2.1.52. Informacion5

Este formulario es igual que el anterior, pero con la diferencia de que, en los circuitos del anterior formulario, la representación gráfica es tensión de salida frente a tensión de entrada. Sin embargo, para el rectificador de onda completa, el cual tiene el mismo problema que los circuitos del anterior apartado, la gráfica que se representa es de la tensión de salida y la tensión de entrada frente al tiempo.

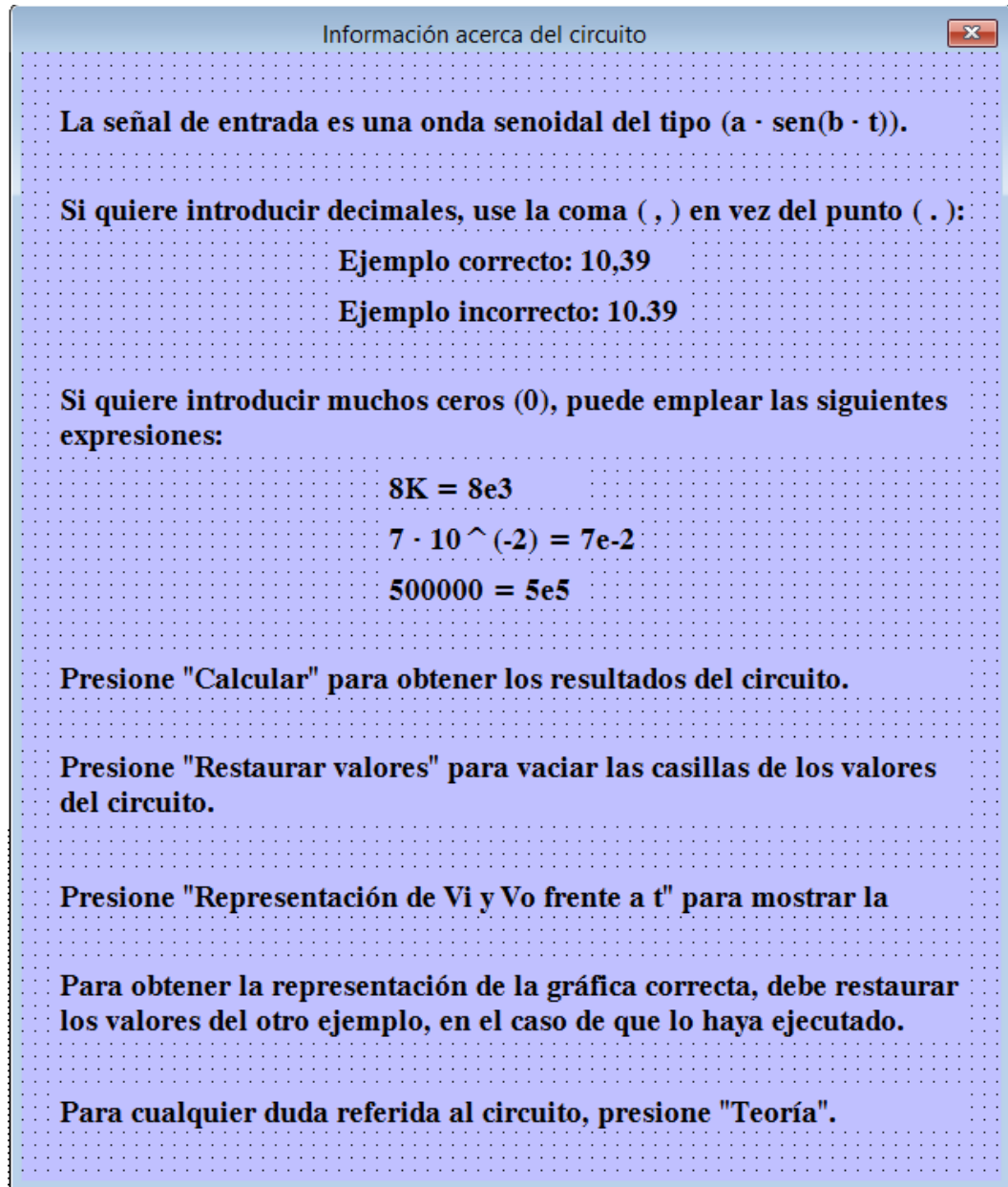


Ilustración 73. Formulario Informacion5

4.2.1.53. InformacionDiferencial

Este formulario es exclusivamente para el circuito diferenciador.

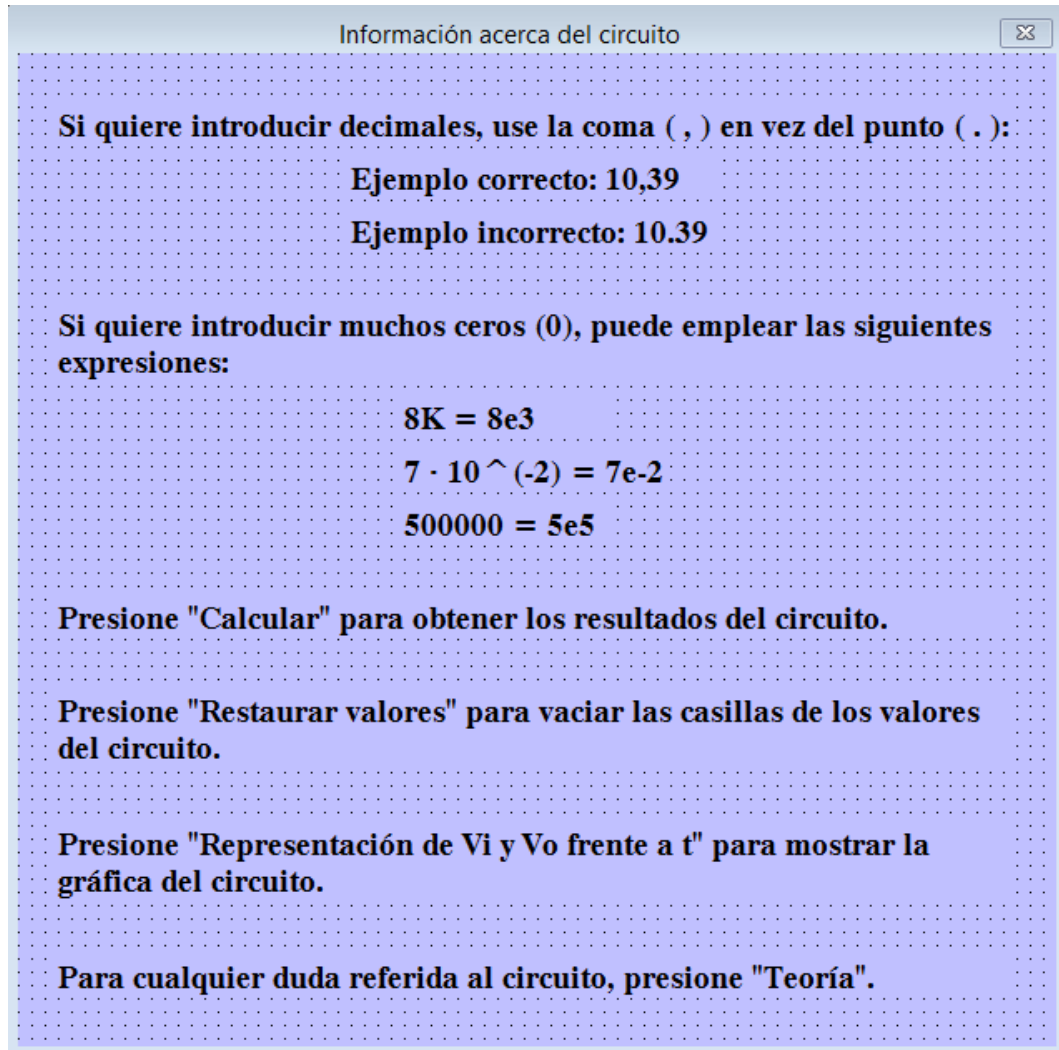


Ilustración 74. Formulario InformacionDiferencial

4.2.1.54. InformacionInt1

Este *UserForm* está destinado para los circuitos integradores inversores y no inversores, que son los que proporcionan una representación gráfica de los circuitos integradores.

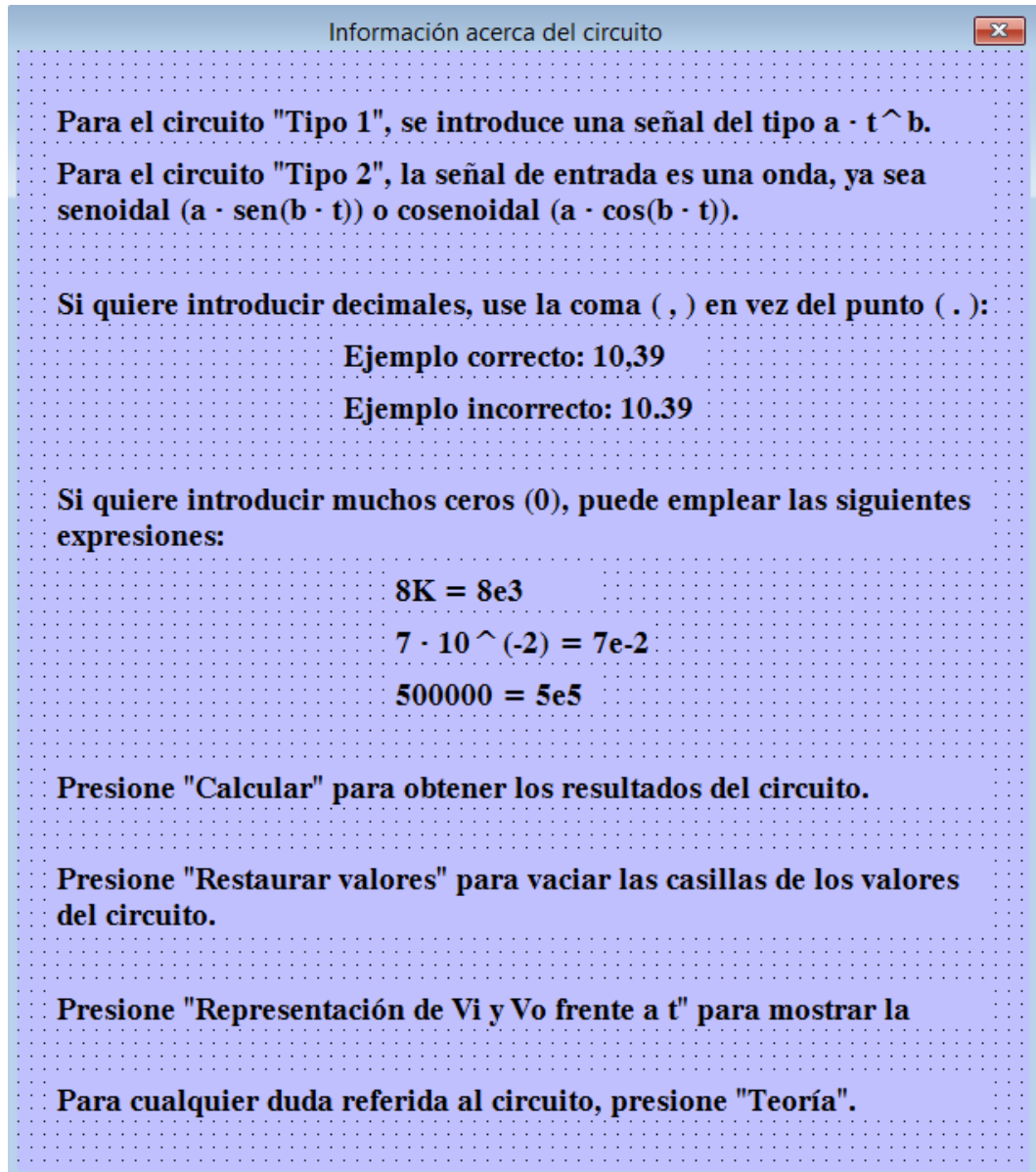


Ilustración 75. Formulario InformacionInt1

4.2.1.55. InformacionInt2

Este formulario se usa en los circuitos integradores en los que no se muestran una representación gráfica. Es decir, en los integradores restadores y sumadores.

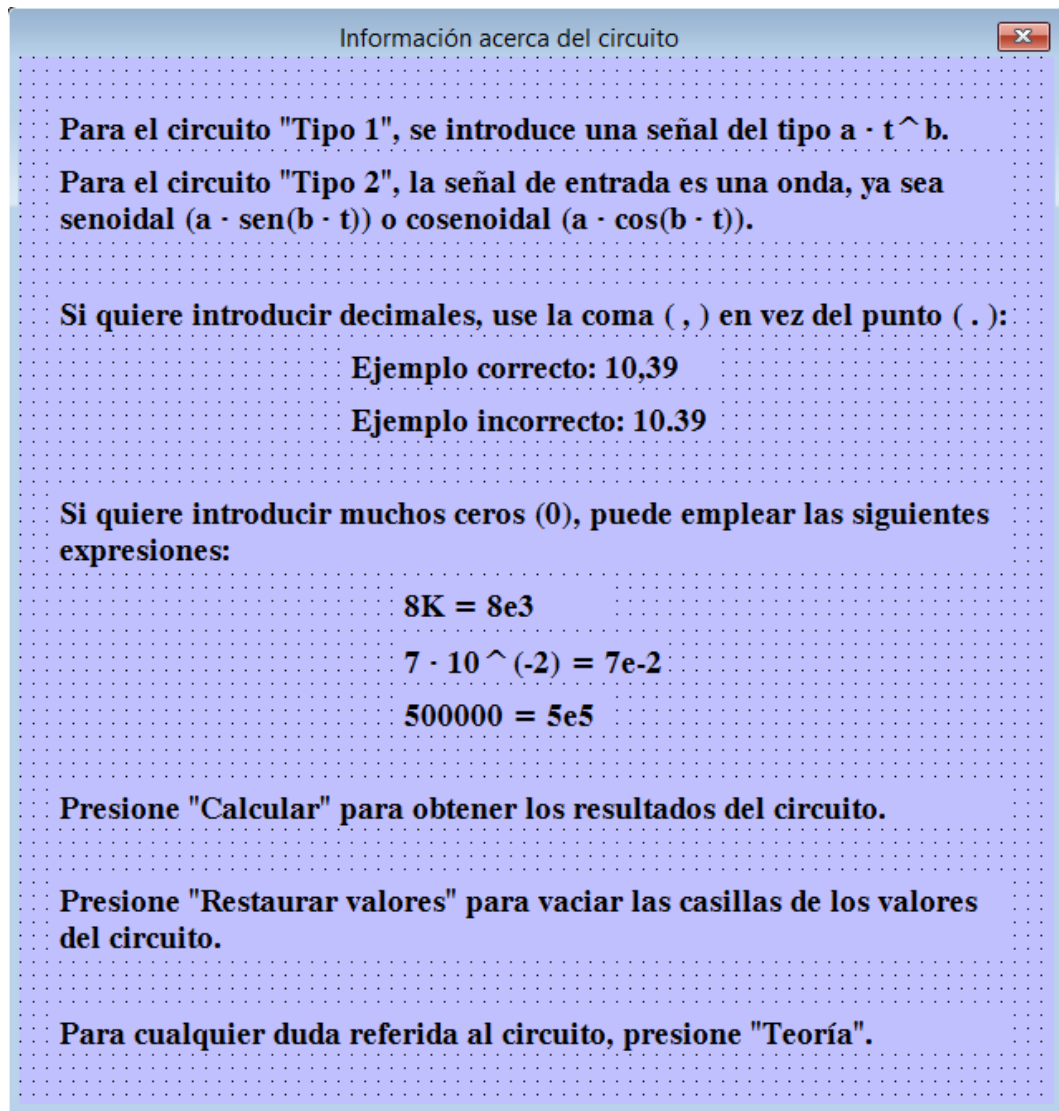


Ilustración 76. Formulario InformacionInt2

4.2.1.56. InformacionOndas

Por último, se encuentra el *UserForm* dedicado a los circuitos generador de onda cuadrada y generador de onda triangular.

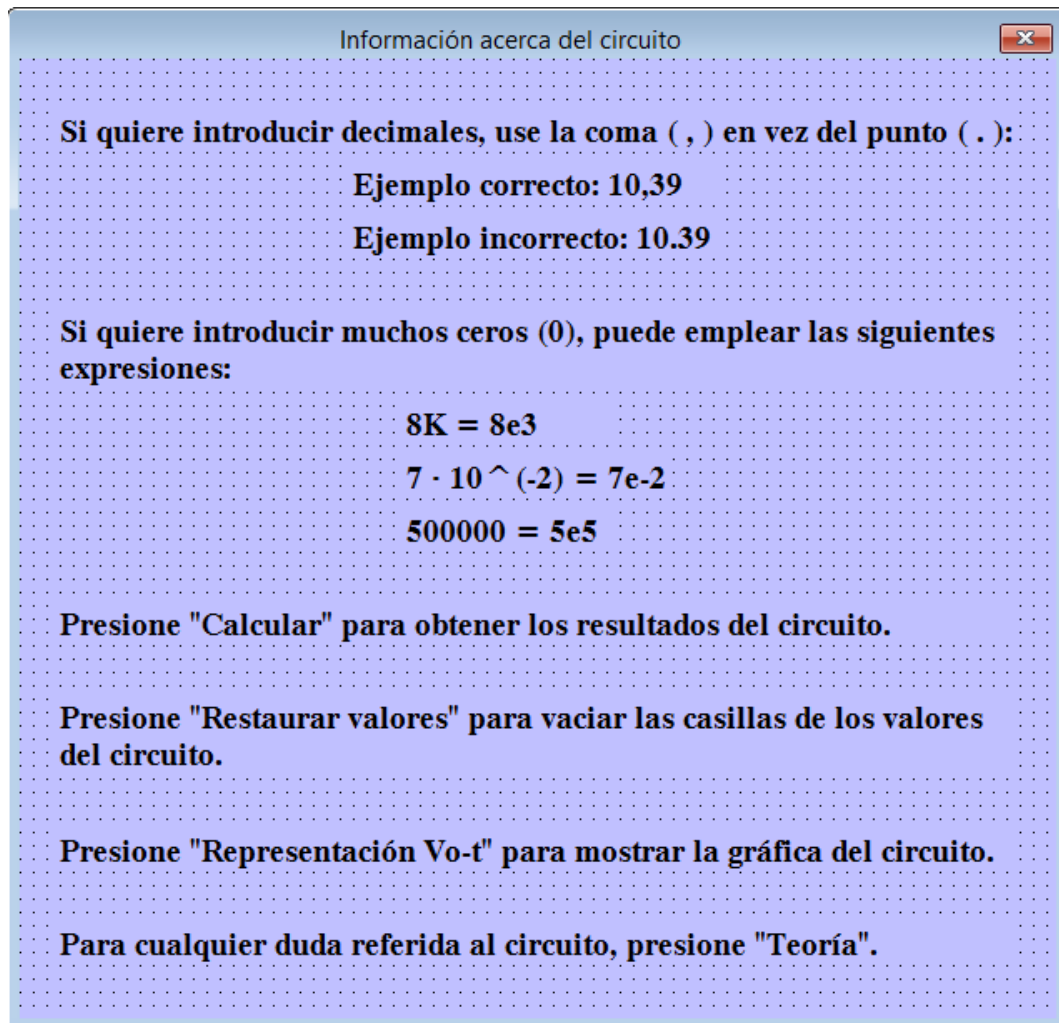


Ilustración 77. Formulario InformacionOndas

El otro tipo de formulario de este caso son los desarrollados para mostrar la teoría de cada uno de los circuitos del programa. Estos formularios disponen básicamente de una imagen donde se recoge la teoría del circuito.

4.2.1.57. TeoriaCompCeroSat1

Teoría circuito detector con salida cero-saturación

Circuito detector con salida cero-saturación

• D CONDUCE ⇒ REALIMENTACIÓN

$$V_x = V_y = 0 \quad ; \quad I_D = I_1 - I_2 > 0 \quad \Rightarrow \quad V_O = -V_U$$

$$I_1 = \frac{V_1 - V_x}{R_1} = \frac{V_1}{R_1} \quad ; \quad I_2 = \frac{V_x - V_{ref}}{R_2} = \frac{-V_{ref}}{R_2}$$

$$I_D = \frac{V_1}{R_1} - \frac{-V_{ref}}{R_2} = \frac{VR_2 + V_{ref}R_1}{R_1R_2} > 0 \Rightarrow$$

• D NO CONDUCE si $I_D \rightarrow (< 0)$

↓

NO REALIMENTACIÓN ⇒ $V_O = +V_{sat}$

Característica de transferencia

Ilustración 78. Formulario TeoriaCompCeroSat1

4.2.1.58. TeoriaCompCeroSat2

Teoría circuito detector con salida cero-saturación

Circuito detector con salida cero-saturación

• D CONDUCE ⇒ REALIMENTACIÓN

$$V_x = V_y = 0 \quad ; \quad I_D = I_1 - I_2 < 0 \quad \Rightarrow \quad V_O = V_U$$

$$I_1 = \frac{V_1 - V_x}{R_1} = \frac{V_1}{R_1} \quad ; \quad I_2 = \frac{V_x - V_{ref}}{R_2} = \frac{VR_2 + V_{ref}R_1}{R_1R_2} < 0 \Rightarrow$$

• D NO CONDUCE si $I_D \rightarrow (> 0)$

↓

NO REALIMENTACIÓN ⇒ $V_O = -V_{sat}$

Característica de transferencia

Ilustración 79. Formulario TeoriaCompCeroSat2

4.2.1.59. TeoriaCompSatSat

Teoría circuito detector con salida saturación-saturación y las dos señales de entrada en el mismo terminal

Circuito detector con salida saturación-saturación y las dos señales de entrada en el mismo terminal

$V_y = 0 \quad ; \quad I_1 = I_2$

$V_o = A(V_y - V_x) \Rightarrow V_o = -AV_x$

$I_1 = \frac{V_i - V_x}{R_1}$

$I_2 = \frac{V_x - V_{ref}}{R_2}$

$\left. \begin{matrix} I_1 = I_2 \\ I_2 = \frac{V_x - V_{ref}}{R_2} \end{matrix} \right\} \frac{V_i - V_x}{R_1} = \frac{V_x - V_{ref}}{R_2} \Rightarrow V_x = \frac{VR_2 + V_{ref}R_1}{R_1 + R_2}$

- Si $V_x = \frac{VR_2 + V_{ref}R_1}{R_1 + R_2} > 0 \quad \left(V_i > V_{ref} \frac{R_1}{R_2} \right) \Rightarrow V_o = -V_{sat}$

- Si $V_x = \frac{VR_2 + V_{ref}R_1}{R_1 + R_2} < 0 \quad \left(V_i < V_{ref} \frac{R_1}{R_2} \right) \Rightarrow V_o = +V_{sat}$

Característica de transferencia

Ilustración 80. Formulario TeoriaCompSatSat

4.2.1.60. TeoriaCompSchmitt

Teoría circuito disparador de Schmitt

Circuito disparador de Schmitt

$V_x = V_i \quad ; \quad V_y = 0$

$V_o = A(V_y - V_x) \Rightarrow V_o = -AV_i$

$\left\{ \begin{matrix} - \text{ Si } V_i < 0 \Rightarrow V_o = +V_{sat} \\ - \text{ Si } V_i > 0 \Rightarrow V_o = -V_{sat} \end{matrix} \right.$

Característica de transferencia

Ilustración 81. Formulario TeoriaCompSchmitt

4.2.1.61. TeoriaCompSimple

Teoría circuito comparador simple

Circuito comparador simple

$V_x = V_i$; $V_y = V_{ref}$
 $V_o = A(V_y - V_x) \Rightarrow V_o = A(V_{ref} - V_i)$
 - Si $V_i < V_{ref} \Rightarrow V_o = +V_{sat}$
 - Si $V_i > V_{ref} \Rightarrow V_o = -V_{sat}$

Característica de transferencia

Ilustración 82. Formulario TeoríaCompSimple

4.2.1.62. TeoríaConvCorrTen

Teoría convertidores de corriente-tensión

Ejemplo 1

$V_x = V_y = 0$
 $I_i = \frac{V_x - V_o}{R} = \frac{-V_o}{R} \Rightarrow V_o = -R I_i$ ($K = -R < 0$)

Ejemplo 2

A1 → Convertidor corriente-tensión
 $V_{o1} = -R I_i$
A2 → Amplificador inversor
 $V_o = -V_{o1} \frac{R_2}{R_1} \Rightarrow V_o = R \frac{R_2}{R_1} I_i$
 $(K = R \frac{R_2}{R_1} > 0)$

Ilustración 83. Formulario TeoríaConvCorrTen

4.2.1.63. TeoriaConvTenCorr

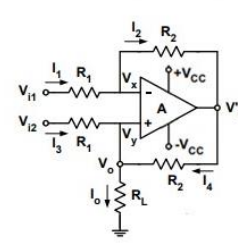
Teoría convertidor tensión-corriente

Sabiendo que:

$V_x = V_y = V_o$; $I_1 = I_2$; $I_o = I_3 + I_4$

Obtenemos:

Convertidor tensión-corriente



$$I_1 = \frac{V_{11} - V_x}{R_1}$$

$$I_2 = \frac{V_x - V_o}{R_2}$$

$$I_3 = \frac{V_{12} - V_y}{R_1}$$

$$I_4 = \frac{V_o - V_y}{R_2}$$

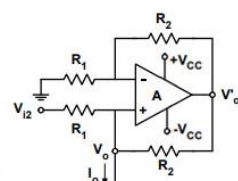
$$\frac{V_{11} - V_x}{R_1} = \frac{V_x - V_o}{R_2} \Rightarrow \frac{V_o}{R_2} = \frac{V_x}{R_2} + \frac{V_x - V_{11}}{R_1}$$

$$I_o = \frac{V_{12} - V_y}{R_1} + \frac{V_o - V_y}{R_2} = \frac{V_{12}}{R_1} \cdot \frac{V_y}{R_1} + \left(\frac{V_x}{R_2} + \frac{V_x - V_{11}}{R_1} \right) \cdot \frac{V_y}{R_2}$$

$$I_o = \frac{V_{12} - V_{11}}{R_1}$$

- Si $V_{11} = 0$

$$I_o = \frac{V_{12}}{R_1}$$

$$K = \frac{1}{R_1} > 0$$


- Si $V_{12} = 0$

$$I_o = \frac{-V_{11}}{R_1}$$

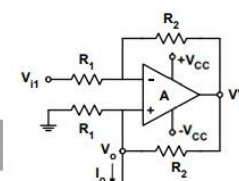
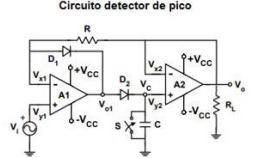
$$K = \frac{-1}{R_1} < 0$$


Ilustración 84. Formulario TeoriaConvTenCorr

4.2.1.64. TeoriaDetectPico

Teoría circuito detector de pico

Circuito detector de pico



FUNCIONAMIENTO:

- Con $V_i > 0 \rightarrow V_{o1} > 0$ (D_1 OFF y D_2 ON) cargándose el condensador C.
 - Si V_i crece y $V_i > V_C \rightarrow$ seguirá D_2 ON cargando C $\Rightarrow V_o = V_C = V_{i1} - V_{i1} - V_i$ ($I_C = 0$)
 - $V_{o1} = V_C + V_i = V_i + V_C$; $V_{D1} = V_{i1} - V_{o1} = V_i - (V_i + V_C) = -V_C \rightarrow$ seguirá D_1 OFF
- Si V_i disminuye y $V_i < V_C \rightarrow V_{o1}$ se reduce y D_2 OFF $\Rightarrow V_o = V_C = V_i$ (máx.) anterior
- $V_{o1} \rightarrow -V_{sat}$ ($V_{i1} = V_C$ y $V_{i1} = V_i < V_C$) $\rightarrow D_1$ ON y $V_{o1} = V_i - V_C$
- Si V_i vuelve a crecer y $V_i > V_C \rightarrow$ de nuevo D_1 OFF y D_2 ON, cargándose C hasta el nuevo valor máximo de $V_i \Rightarrow V_o = V_C = V_i$ (máx.) nuevo

A1 \rightarrow Carga el condensador C, a través de D_2 , solo cuando $V_i > V_C$.

A2 \rightarrow Circuito separador (buffer) con $V_o = V_C$.

- Su alta impedancia de entrada evita la descarga de C.

EN LA SALIDA SIEMPRE SE TENDRÁ $V_o = V_C$ QUE SERÁ EL MAYOR VALOR POSITIVO DE V_i .

EL INTERRUPTOR "S" PERMITE DESCARGAR C PARA REINICIAR LA DETECCIÓN DEL PICO. (Suele emplearse como interruptor un transistor en conmutación)

PARA DETECTAR PICOS NEGATIVOS BASTARÍA CON INVERTIR AMBOS DIODOS.

Ilustración 85. Formulario TeoriaDetectPico

4.2.1.65. TeoriaDiferenciador

Teoría circuito diferenciador

Circuito Diferenciador

Sabiendo que:

$$V_x = V_y = 0 \quad ; \quad I_c = I_1$$

Obtenemos:

$$I_c = C \frac{d(V_i - V_x)}{dt} = C \frac{dV_i}{dt}$$

$$I_1 = \frac{V_x - V_o}{R} = \frac{-V_o}{R}$$

$$\left. \begin{array}{l} I_c = C \frac{dV_i}{dt} \\ I_1 = \frac{-V_o}{R} \end{array} \right\} C \frac{dV_i}{dt} = \frac{-V_o}{R} \Rightarrow V_o = -RC \frac{dV_i}{dt}$$

Ilustración 86. Formulario TeoriaDiferenciador

4.2.1.66. TeoriaFuenteCorriente

Teoría fuentes de corriente

Ejemplo 1

Sabiendo que:

$$V_x = V_y = V_i \quad ; \quad I_L = I_1$$

Obtenemos:

$$I_1 = \frac{V_x}{R_1} = \frac{V_i}{R_1} \Rightarrow I_L = \frac{V_i}{R_1} = \text{cte.}$$

Ejemplo 2

Sabiendo que:

$$V_x = V_y = 0 \quad ; \quad I_1 = I_2 \quad ; \quad I_L = I_2 + I_3$$

Obtenemos:

$$\left. \begin{aligned} I_1 &= \frac{V_x - V_i}{R_1} = \frac{-V_i}{R_1} \\ I_2 &= \frac{V_o - V_x}{R_2} = \frac{V_o}{R_2} \\ I_3 &= \frac{V_o}{R_3} \end{aligned} \right\} \begin{aligned} \frac{-V_i}{R_1} &= \frac{V_o}{R_2} \Rightarrow V_o = -V_i \frac{R_2}{R_1} \\ I_L &= \frac{V_o}{R_2} + \frac{V_o}{R_3} = V_o \left(\frac{1}{R_2} + \frac{1}{R_3} \right) \Rightarrow \\ &\Rightarrow I_L = -V_i \frac{1}{R_1} \left(1 + \frac{R_2}{R_3} \right) = \text{cte.} \end{aligned}$$

Ilustración 87. Formulario TeoriaFuenteCorriente

4.2.1.67. TeoriaFuenteTension

Teoría fuente con diodo Zener

Fuente con diodo Zener

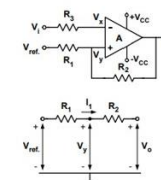
$$V_{ref.} = -V_A \frac{R_2}{R_1} = -V_Z \frac{R_2}{R_1}$$

Ilustración 88. Formulario TeoriaFuenteTension

4.2.1.68. TeoriaHisteresis

Teoría circuito comparador con histeresis

Comparador con Histeresis



- No tiene realimentación negativa.

$$V_o = A(V_y - V_x)$$

$$V_x = V_i$$

$$V_y = V_{ref} - I_f R_1$$

$$I_f = \frac{V_{ref} - V_o}{R_1 + R_2}$$

$$V_y = \frac{V_{ref} R_2 + V_o R_1}{R_1 + R_2}$$

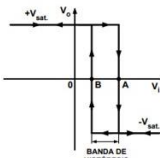
- Suponiendo inicialmente $V_o = +V_{sat} \Rightarrow V_y = \frac{V_{ref} R_2 + V_{sat} R_1}{R_1 + R_2}$

Si $V_i > V_y \Rightarrow V_o = -V_{sat}$

- Suponiendo que ahora $V_o = -V_{sat} \Rightarrow V_y = \frac{V_{ref} R_2 - V_{sat} R_1}{R_1 + R_2}$

Si $V_i < V_y \Rightarrow V_o = +V_{sat}$

Característica de transferencia



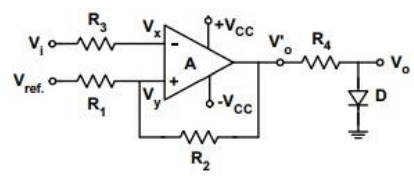
ANCHO DE LA BANDA DE HISTERESIS $\rightarrow A - B = 2V_{sat} \frac{R_1}{R_1 + R_2}$

Ilustración 89. Formulario TeoriaHisteresis

4.2.1.69. TeoriaHisteresis1

Teoría circuito comparador con histeresis y limitación a la salida

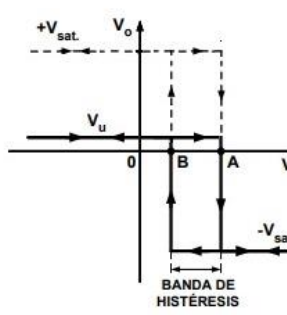
Comparador con Histeresis



- Si $V'_o = +V_{sat} \Rightarrow D$ conduce $\Rightarrow V_o = V_u$

- Si $V'_o = -V_{sat} \Rightarrow D$ no conduce $\Rightarrow V_o = -V_{sat}$

Característica de transferencia



BANDA DE HISTERESIS

Ilustración 90. Formulario TeoriaHisteresis1

4.2.1.70. TeoriaHisteresis2

Teoría circuito comparador con histeresis y limitación a la salida

Comparador con Histéresis

- Si $V_o = +V_{sat}$. \Rightarrow D no conduce $\Rightarrow V_o = +V_{sat}$.

- Si $V_o = -V_{sat}$. \Rightarrow D conduce $\Rightarrow V_o = -V_u$.

Característica de transferencia

Ilustración 91. Formulario TeoriaHisteresis2

4.2.1.71. TeoriaIntInv

Teoría circuito integrador inversor

Circuito Integrador Inversor

Sabiendo que:

$V_x = V_y = 0$; $I_1 = I_C$

Obtenemos:

$$\left. \begin{aligned} I_1 &= \frac{V_1 - V_x}{R} = \frac{V_1}{R} \\ I_C &= C \frac{d(V_x - V_o)}{dt} = -C \frac{dV_o}{dt} \end{aligned} \right\} \frac{V_1}{R} = -C \frac{dV_o}{dt} \Rightarrow dV_o = -\frac{1}{RC} V_1 dt$$

$$V_o = -\frac{1}{RC} \int V_1 dt$$

Ilustración 92. Formulario TeoriaIntInv

4.2.1.72. TeoriaIntNoInv

Teoría circuito integrador no inversor

Circuito Integrador no Inversor

Sabiendo que:

$$V_x = V_y \quad ; \quad I_1 = I_2 \quad ; \quad I_c = I_3 + I_4$$

Obtenemos:

$$I_1 = \frac{V_0 - V_x}{R}$$

$$I_2 = \frac{V_x}{R}$$

$$I_3 = \frac{V_1 - V_y}{R}$$

$$I_4 = \frac{V_0 - V_y}{R}$$

$$I_c = C \frac{dV_y}{dt}$$

$$\left. \begin{array}{l} I_1 = \frac{V_0 - V_x}{R} \\ I_2 = \frac{V_x}{R} \end{array} \right\} \frac{V_0 - V_x}{R} = \frac{V_x}{R} \Rightarrow V_x = \frac{V_0}{2}$$

$$\left. \begin{array}{l} I_3 = \frac{V_1 - V_y}{R} \\ I_4 = \frac{V_0 - V_y}{R} \\ I_c = C \frac{dV_y}{dt} \end{array} \right\} C \frac{dV_y}{dt} = \frac{V_1 - V_y}{R} + \frac{V_0 - V_y}{R} = \frac{V_1 - 2V_y + V_0}{R} \Rightarrow C \frac{d\left(\frac{V_0}{2}\right)}{dt} = \frac{V_1 - 2\left(\frac{V_0}{2}\right) + V_0}{R} \Rightarrow$$

$$\Rightarrow \frac{RC}{2} \frac{dV_0}{dt} = V_1 \Rightarrow dV_0 = \frac{2}{RC} V_1 dt \Rightarrow V_0 = \frac{2}{RC} \int V_1 dt$$

Ilustración 93. Formulario TeoriaIntNoInv

4.2.1.73. TeoriaIntRes

Teoría circuito integrador restador

Circuito Integrador Restador

Sabiendo que:

$$V_x = V_y \quad ; \quad I_1 = I_{c1} \quad ; \quad I_2 = I_{c2}$$

Obtenemos:

$$I_2 = \frac{V_{12} - V_y}{R}$$

$$I_{c2} = C \frac{dV_y}{dt}$$

$$I_1 = \frac{V_{11} - V_x}{R}$$

$$I_{c1} = C \frac{d(V_x - V_0)}{dt}$$

$$\left. \begin{array}{l} I_2 = \frac{V_{12} - V_y}{R} \\ I_{c2} = C \frac{dV_y}{dt} \end{array} \right\} \frac{V_{12} - V_y}{R} = C \frac{dV_y}{dt}$$

$$\left. \begin{array}{l} I_1 = \frac{V_{11} - V_x}{R} \\ I_{c1} = C \frac{d(V_x - V_0)}{dt} \end{array} \right\} \frac{V_{11} - V_x}{R} = C \frac{d(V_x - V_0)}{dt} \Rightarrow C \frac{dV_0}{dt} = C \frac{dV_x}{dt} - \frac{V_{11} - V_x}{R} \Rightarrow$$

$$\Rightarrow C \frac{dV_0}{dt} = \frac{V_{12} - V_y}{R} - \frac{V_{11} - V_x}{R} = \frac{V_{12} - V_{11}}{R} \Rightarrow dV_0 = \frac{1}{RC} (V_{12} - V_{11}) dt \Rightarrow$$

$$\Rightarrow V_0 = \frac{1}{RC} \int (V_{12} - V_{11}) dt$$

Ilustración 94. Formulario TeoriaIntRes

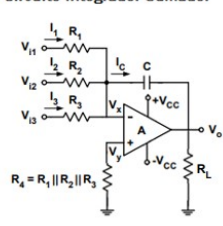
4.2.1.74. TeoriaIntSum

Teoría circuito integrador sumador

Sabido que:
 $V_x = V_y = 0$; $I_c = I_1 + I_2 + I_3$

Obtenemos:

Circuito Integrador Sumador



$R_4 = R_1 \parallel R_2 \parallel R_3$

$$I_1 = \frac{V_{i1} - V_x}{R_1} = \frac{V_{i1}}{R_1}$$

$$I_2 = \frac{V_{i2} - V_x}{R_2} = \frac{V_{i2}}{R_2}$$

$$I_3 = \frac{V_{i3} - V_x}{R_3} = \frac{V_{i3}}{R_3}$$

$$I_c = C \frac{d(V_x - V_o)}{dt} = -C \frac{dV_o}{dt}$$

$$-C \frac{dV_o}{dt} = \frac{V_{i1}}{R_1} + \frac{V_{i2}}{R_2} + \frac{V_{i3}}{R_3} \Rightarrow dV_o = -\frac{1}{C} \left(\frac{V_{i1}}{R_1} + \frac{V_{i2}}{R_2} + \frac{V_{i3}}{R_3} \right) dt \Rightarrow$$

$$V_o = -\frac{1}{C} \int \left(\frac{V_{i1}}{R_1} + \frac{V_{i2}}{R_2} + \frac{V_{i3}}{R_3} \right) dt$$

Si $R_1 = R_2 = R_3 = R \Rightarrow V_o = -\frac{1}{RC} \int (V_{i1} + V_{i2} + V_{i3}) dt$

Ilustración 95. Formulario TeoriaIntSum

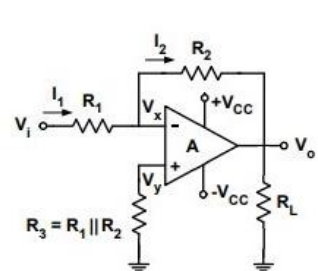
4.2.1.75. TeoriaInv

Teoría circuito inversor

Sabido que:
 $V_x = V_y = 0$; $I_1 = I_2$

Obtenemos:

Circuito Inversor



$R_3 = R_1 \parallel R_2$

$$I_1 = \frac{V_i - V_x}{R_1} = \frac{V_i}{R_1}$$

$$I_2 = \frac{V_x - V_o}{R_2} = \frac{-V_o}{R_2}$$

$$\frac{V_i}{R_1} = \frac{-V_o}{R_2} \Rightarrow V_o = -V_i \frac{R_2}{R_1}$$

GANANCIA = $\left(-\frac{R_2}{R_1} \right)$

Representación Vo-Vi

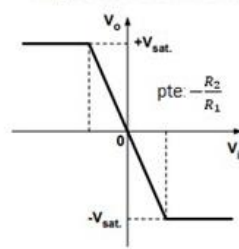
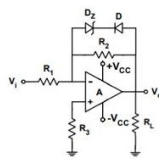


Ilustración 96. Formulario TeoriaInv

4.2.1.76. TeoriaLimDiodoNormalDiodoZener1

Teoría circuito limitador con diodo normal y diodo zener en serie

Circuito Limitador con diodo normal y diodo zener en serie



- Con $V_i > 0 \rightarrow V_o < 0 \rightarrow$ D polarizado en inversa.
→ D₂ polarizado en directa. } → NO CONDUCCIÓN ⇒ $V_o = -V \frac{R_2}{R_1}$
- Con $V_i < 0 \rightarrow V_o > 0 \rightarrow$ D polarizado en directa.
→ D₂ polarizado en inversa. }
- Si $V_o < (V_u + V_z) \rightarrow$ D polarizado en directa.
→ D₂ no en zona zener. } → NO CONDUCCIÓN ⇒ $V_o = -V \frac{R_2}{R_1}$
- Si $V_o = (V_u + V_z) \rightarrow$ D polarizado en directa.
→ D₂ en zona zener. } → CONDUCCIÓN ⇒ $V_o = -V \frac{R_2}{R_1} - (V_u + V_z)$

↓
 $V_i = -(V_u + V_z) \frac{R_1}{R_2}$

- Si $V_i < -(V_u + V_z) \frac{R_1}{R_2} \rightarrow$ D y D₂ siempre CONDUCE. ⇒ $V_o = -(V_u + V_z)$

Característica de transferencia

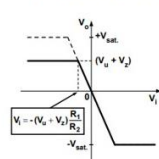
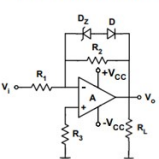


Ilustración 97. Formulario TeoriaLimDiodoNormalDiodoZener1

4.2.1.77. TeoriaLimDiodoNormalDiodoZener2

Teoría circuito limitador con diodo normal y diodo zener en serie

Circuito Limitador con diodo normal y diodo zener en serie



- Con $V_i < 0 \rightarrow V_o > 0 \rightarrow$ D polarizado en inversa.
→ D₂ polarizado en directa. } → NO CONDUCCIÓN ⇒ $V_o = -V \frac{R_2}{R_1}$
- Con $V_i > 0 \rightarrow V_o < 0 \rightarrow$ D polarizado en directa.
→ D₂ polarizado en inversa. }
- Si $V_o > -(V_u + V_z) \rightarrow$ D polarizado en directa.
→ D₂ no en zona zener. } → NO CONDUCCIÓN ⇒ $V_o = -V \frac{R_2}{R_1}$
- Si $V_o = -(V_u + V_z) \rightarrow$ D polarizado en directa.
→ D₂ en zona zener. } → CONDUCCIÓN ⇒ $V_o = -V \frac{R_2}{R_1} - (V_u + V_z)$

↓
 $V_i = (V_u + V_z) \frac{R_1}{R_2}$

- Si $V_i > (V_u + V_z) \frac{R_1}{R_2} \rightarrow$ D y D₂ siempre CONDUCE. ⇒ $V_o = -(V_u + V_z)$

Característica de transferencia

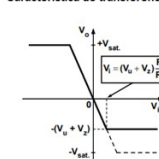


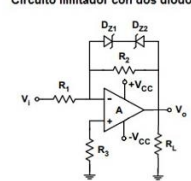
Ilustración 98. Formulario TeoriaLimDiodoNormalDiodoZener2

4.2.1.78. TeoriaLimDosZener

Teoría circuito limitador con dos diodos zener en serie

Página 1 | **Página 2**

Circuito limitador con dos diodos zener



• Con $V_i > 0 \rightarrow V_o < 0 \rightarrow D_{z1}$ polarizado en directa.
 $\rightarrow D_{z2}$ polarizado en inversa.

- Si $V_o > -(V_u + V_{z2}) \rightarrow D_{z1}$ polarizado en directa.
 $\rightarrow D_{z2}$ no en zona zener. } **NO CONDUCCIÓN** $\Rightarrow V_o = -V_i \frac{R_2}{R_1}$

- Si $V_o = -(V_u + V_{z2}) \rightarrow D_{z1}$ polarizado en directa.
 $\rightarrow D_{z2}$ en zona zener. } **CONDUCCIÓN** $\Rightarrow V_o = -V_i \frac{R_2}{R_1} - (V_u + V_{z2})$

- Conducción de D_{z1} en directa y de D_{z2} en zona zener. $\Rightarrow V_i = (V_u + V_{z2}) \frac{R_1}{R_2}$

- Si $V_i > (V_u + V_{z2}) \frac{R_1}{R_2} \rightarrow D_{z1}$ en directa y D_{z2} en zener, siempre **CONDUCE**. $\Rightarrow V_o = -(V_u + V_{z2})$

- Característica de los diodos zener.

• $D_{z1} \rightarrow (V_u ; V_{z1})$
 • $D_{z2} \rightarrow (V_u ; V_{z2})$ } $V_{z2} > V_{z1}$

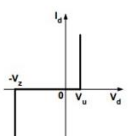


Ilustración 99. Formulario TeoríaLimDosZener Página 1

Teoría circuito limitador con dos diodos zener en serie

Página 1 | **Página 2**

• Con $V_i < 0 \rightarrow V_o > 0 \rightarrow D_{z1}$ polarizado en inversa.
 $\rightarrow D_{z2}$ polarizado en directa.

- Si $V_o < (V_u + V_{z1}) \rightarrow D_{z1}$ no en zona zener.
 $\rightarrow D_{z2}$ polarizado en directa. } **NO CONDUCCIÓN** $\Rightarrow V_o = -V_i \frac{R_2}{R_1}$

- Si $V_o = (V_u + V_{z1}) \rightarrow D_{z1}$ en zona zener.
 $\rightarrow D_{z2}$ polarizado en directa. } **CONDUCCIÓN** $\Rightarrow V_o = -V_i \frac{R_2}{R_1} + (V_u + V_{z1})$

- Conducción de D_{z1} en zona zener y de D_{z2} en directa. $\Rightarrow V_i = -(V_u + V_{z1}) \frac{R_1}{R_2}$

- Si $V_i < -(V_u + V_{z1}) \frac{R_1}{R_2} \rightarrow D_{z1}$ en zener y D_{z2} en directa, siempre **CONDUCE**. $\Rightarrow V_o = (V_u + V_{z1})$

Característica de transferencia

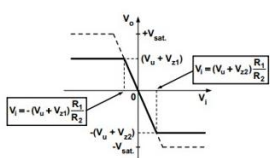
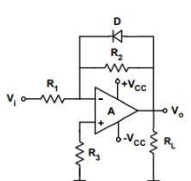


Ilustración 100. Formulario TeoríaLimDosZener Página 2

4.2.1.79. TeoríaLimSimple1

Teoría circuito limitador simple

Circuito limitador simple



• Con $V_i > 0 \rightarrow V_o < 0 \rightarrow D$ polarizado en inversa. $\Rightarrow V_o = -V_i \frac{R_2}{R_1}$

• Con $V_i < 0 \rightarrow V_o > 0 \rightarrow D$ polarizado en directa.

- Si $V_o < V_u \rightarrow D$ en directa, pero **NO CONDUCE**. $\Rightarrow V_o = -V_i \frac{R_2}{R_1}$

- Si $V_o = V_u \rightarrow D$ en directa y **CONDUCE**. $\Rightarrow V_o = -V_i \frac{R_2}{R_1} = V_u \Rightarrow V_i = -V_u \frac{R_1}{R_2}$

- Si $V_i < -V_u \frac{R_1}{R_2} \rightarrow D$ siempre **CONDUCE**. $\Rightarrow V_o = V_u$

Característica de transferencia

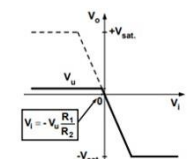
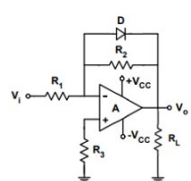


Ilustración 101. Formulario TeoríaLimSimple1

4.2.1.80. TeoriaLimSimple2

Teoría circuito limitador simple

Circuito limitador simple



- Con $V_i < 0 \rightarrow V_o > 0 \rightarrow$ D polarizado en **inversa**. $\Rightarrow V_o = -V_i \frac{R_2}{R_1}$
- Con $V_i > 0 \rightarrow V_o < 0 \rightarrow$ D polarizado en **directa**.
- Si $V_o > -V_u \rightarrow$ D en **directa**, pero **NO CONDUCE**. $\Rightarrow V_o = -V_i \frac{R_2}{R_1}$
- Si $V_o = -V_u \rightarrow$ D en **directa** y **CONDUCE**. $\Rightarrow V_o = -V_i \frac{R_2}{R_1} = -V_u \Rightarrow V_i = V_u \frac{R_1}{R_2}$
- Si $V_i > V_u \frac{R_1}{R_2} \rightarrow$ D siempre **CONDUCE**. $\Rightarrow V_o = -V_u$

Característica de transferencia

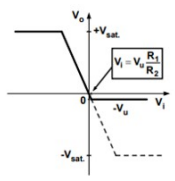
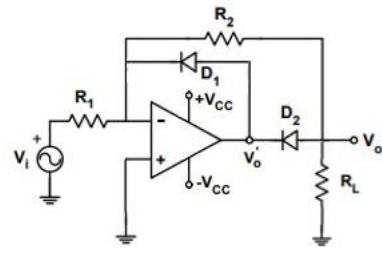


Ilustración 102. Formulario TeoriaLimSimple2

4.2.1.81. TeoriaMediaOnda

Teoría circuito rectificador de media onda

Circuito rectificador de media onda



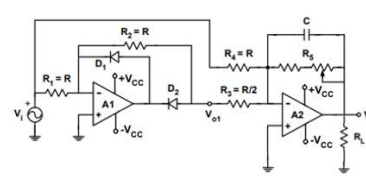
- Con $V_i > 0 \rightarrow V'_o < 0$
D1 OFF y D2 ON ($V'_o = V_o - V_u$) $\Rightarrow V_o = -V_i \frac{R_2}{R_1}$
- Con $V_i < 0 \rightarrow V'_o > 0$
D1 ON y D2 OFF ($V'_o = V_u$) $\Rightarrow V_o = V_x = V_y = 0$

Ilustración 103. Formulario TeoriaMediaOnda

4.2.1.82. TeoriaMedioEficaz

Teoría circuito de valor medio y valor eficaz

Circuito de valor medio y valor eficaz



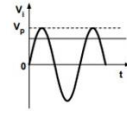
- Condensador C, filtro pasa bajo para obtener V_{med} .
- Ganancia $G = R_5 / R$ ajustable mediante el potenciómetro

$$\left\{ \begin{array}{l} \text{Si } G = 1,11 \rightarrow V_o = V_{rms} \\ \text{Si } G = 1 \rightarrow V_o = V_{med} \end{array} \right.$$

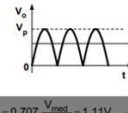
Frecuencia de corte del filtro $f_c = \frac{1}{2\pi R_5 C}$

$f_c \ll f_{min}$ de operación del circuito

Rectificador duplica la frecuencia $\left\{ \begin{array}{l} C \gg \frac{1}{4\pi R_5 f_{min}} \\ \text{Regla práctica} \rightarrow C = \frac{1}{\max. \text{ rizado}} \frac{1}{4\pi R_5 f_{min}} \end{array} \right.$



$V_{rms} = \frac{V_p}{\sqrt{2}} = 0,707 V_p$



$V_{med} = \frac{2V_p}{\pi} = 0,637 V_p$

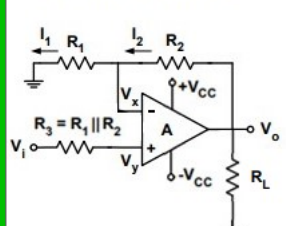
$$V_{rms} = \frac{V_p}{\sqrt{2}} = 0,707 V_p = 0,707 \frac{V_{med}}{0,637} = 1,11 V_{med}$$

Ilustración 104. Formulario TeoriaMedioEficaz

4.2.1.83. TeoriaNoInv

Teoría circuito no inversor

Circuito no Inversor



Sabido que:

$$V_x = V_y = V_i \quad ; \quad I_1 = I_2$$

Obtenemos:

$$\left. \begin{array}{l} I_1 = \frac{V_x}{R_1} = \frac{V_i}{R_1} \\ I_2 = \frac{V_o - V_x}{R_2} = \frac{V_o - V_i}{R_2} \end{array} \right\} \frac{V_i}{R_1} = \frac{V_o - V_i}{R_2} \Rightarrow V_o = V_i \frac{R_1 + R_2}{R_1}$$

GANANCIA = $\left(\frac{R_1 + R_2}{R_1} \right)$

Representación Vo-Vi

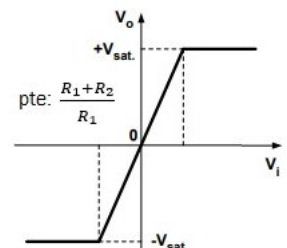


Ilustración 105. Formulario TeoriaNoInv

4.2.1.84. TeoriaOndaCompleta1

Teoria circuito rectificador de onda completa

Circuito rectificador de onda completa

A1 → Rectificador de media onda

- Con $V_i > 0 \Rightarrow V_{o1} = -V_i \frac{R_2}{R_1}$
- Con $V_i < 0 \Rightarrow V_{o1} = V_{i1} = V_{i2} = 0$

A2 → Sumador inversor

$$V_o = - \left(V_{o1} \frac{R_3}{R_3} + V_i \frac{R_5}{R_4} \right)$$

- Con $V_i > 0 \Rightarrow V_{o1} = -V_i \frac{R_2}{R_1} \Rightarrow V_o = V_i \frac{R_2 R_5}{R_1 R_3} - V_i \frac{R_5}{R_4}$
- Con $V_i < 0 \Rightarrow V_{o1} = 0 \Rightarrow V_o = -V_i \frac{R_5}{R_4}$

Para tener la misma salida en valor absoluto, debe cumplirse $\frac{R_2 R_5}{R_1 R_3} = \frac{2 R_5}{R_4}$ ó $R_3 = \frac{R_2 R_4}{2 R_1}$

Normalmente $R_1 = R_2 = R_4 = R$ con lo que $R_3 = R/2$ y la ganancia sería $G = R_5/R$, igual a la unidad si $R_5 = R$.

• INCONVENIENTE → Conseguir tres o cuatro resistencias iguales, debido a su tolerancia.

Ilustración 106. Formulario TeoriaOndaCompleta1

4.2.1.85. TeoriaOndaCompleta2

Teoria circuito rectificador de onda completa

Circuito rectificador de onda completa

A1 → Rectificador de media onda

- Con $V_i > 0 \Rightarrow V_{o1} = -V_i \frac{R_2 + R_3}{R_2}$
- Con $V_i < 0 \Rightarrow V_{o1} = -V_i \frac{R_2 + R_3}{R_1}$

A2 → Sumador inversor

$$V_o = - \left(V_{o1} \frac{R_3}{R_3} + V_i \frac{R_4}{R_4} \right)$$

- Con $V_i > 0 \Rightarrow V_o = V_i \frac{R_2 + R_3}{R_2}$
- Con $V_i < 0 \Rightarrow V_o = -V_i \frac{R_2 + R_3}{R_1}$

- Igual salida en valor absoluto si $R_1 = R_2 = R$

$$V_o = V_i \frac{R + R_3}{R} = V_i \left(1 + \frac{R_3}{R} \right)$$

La ganancia del montaje será $\rightarrow G = \left(1 + \frac{R_3}{R} \right) \Rightarrow R_3 = (G-1)R$

$\rightarrow G = 1 (V_o = |V_i|) \Rightarrow R_3 = 0$

Ilustración 107. Formulario TeoriaOndaCompleta2

4.2.1.86. TeoriaOndaCuadrada

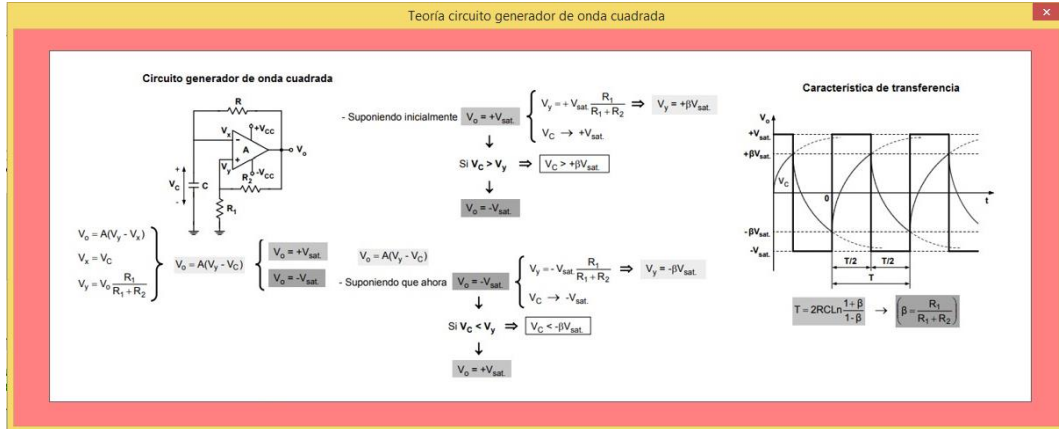


Ilustración 108. Formulario TeoríaOndaCuadrada

4.2.1.87. TeoríaOndaTriangular

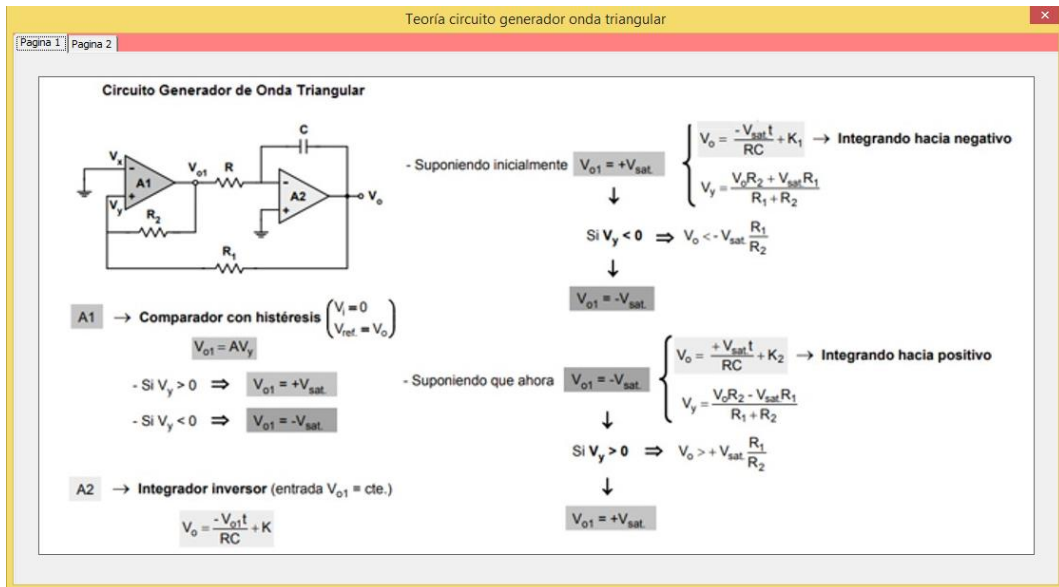


Ilustración 109. Formulario TeoríaOndaTriangular Página 1

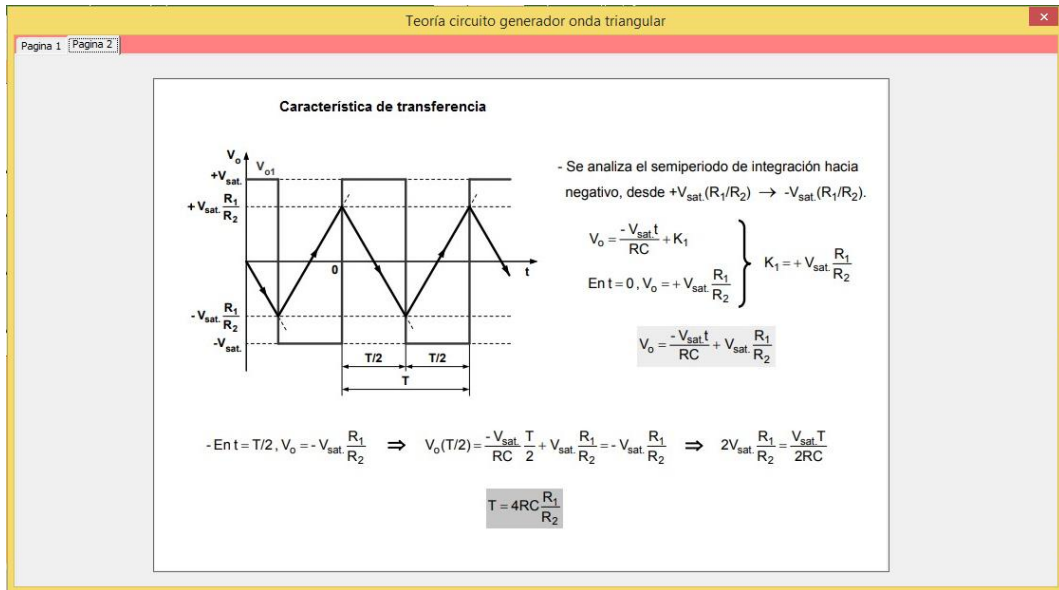


Ilustración 110. Formulario TeoriaOndaTriangular Página 2

4.2.1.88. TeoriaRestAltImp

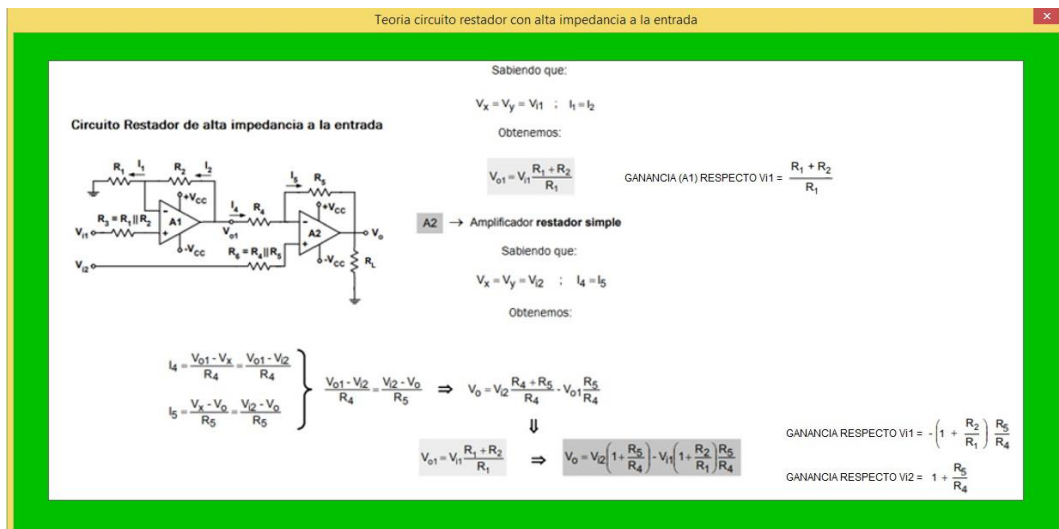


Ilustración 111. Formulario TeoriaRestAltImp

4.2.1.89. TeoriaRestSimple

Teoría circuito restador simple

Sabido que:

$$V_x = V_y \quad ; \quad I_1 = I_2 \quad ; \quad I_3 = I_4$$

Obtenemos:

Circuito Restador Simple

$$\begin{cases}
 I_1 = \frac{V_{11} - V_x}{R_1} \\
 I_2 = \frac{V_x - V_o}{R_2} \\
 I_3 = \frac{V_{12} - V_y}{R_3} \\
 I_4 = \frac{V_y}{R_4}
 \end{cases}
 \Rightarrow
 \begin{cases}
 \frac{V_{11} - V_x}{R_1} = \frac{V_x - V_o}{R_2} \Rightarrow V_o = V_x \frac{R_1 + R_2}{R_1} - V_{11} \frac{R_2}{R_1} \\
 \frac{V_{12} - V_y}{R_3} = \frac{V_y}{R_4} \Rightarrow V_y = V_{12} \frac{R_4}{R_3 + R_4}
 \end{cases}$$

$$V_o = V_{12} \frac{R_4}{R_3 + R_4} \frac{R_1 + R_2}{R_1} - V_{11} \frac{R_2}{R_1}$$

GANANCIA RESPECTO $V_{11} = -\frac{R_2}{R_1}$

GANANCIA RESPECTO $V_{12} = \frac{R_4}{R_3 + R_4} \frac{R_1 + R_2}{R_1}$

Ilustración 112. Formulario TeoriaRestSimple

4.2.1.90. TeoriaSumInv

Teoría circuito sumador inversor

Sabido que:

$$V_x = V_y = 0 \quad ; \quad I_3 = I_1 + I_2$$

Obtenemos:

Circuito Sumador Inversor

$$\begin{cases}
 I_1 = \frac{V_{11} - V_x}{R_1} = \frac{V_{11}}{R_1} \\
 I_2 = \frac{V_{12} - V_x}{R_2} = \frac{V_{12}}{R_2} \\
 I_3 = \frac{V_x - V_o}{R_3} = \frac{-V_o}{R_3}
 \end{cases}
 \Rightarrow
 \frac{-V_o}{R_3} = \frac{V_{11}}{R_1} + \frac{V_{12}}{R_2} \Rightarrow V_o = -\left(V_{11} \frac{R_3}{R_1} + V_{12} \frac{R_3}{R_2} \right)$$

GANANCIA RESPECTO $V_{11} = -\frac{R_3}{R_1}$

GANANCIA RESPECTO $V_{12} = -\frac{R_3}{R_2}$

Ilustración 113. Formulario TeoriaSumInv

4.2.1.91. TeoriaSumNoInv

Teoría circuito sumador no inversor

Sabiendo que:

$$V_x = V_y \quad ; \quad I_1 = I_2 \quad ; \quad I_3 = I_4$$

Obtenemos:

Circuito Sumador no Inversor

$$\left. \begin{aligned} I_1 &= \frac{V_x}{R_1} \\ I_2 &= \frac{V_0 - V_x}{R_2} \\ I_3 &= \frac{V_1 - V_y}{R_3} \\ I_4 &= \frac{V_y - V_0}{R_4} \end{aligned} \right\} \begin{aligned} \frac{V_x}{R_1} - \frac{V_0 - V_x}{R_2} &\Rightarrow V_0 = V_x \frac{R_1 + R_2}{R_1} \\ \frac{V_1 - V_y}{R_3} - \frac{V_y - V_0}{R_4} &\Rightarrow V_y = \frac{V_1 R_4 + V_0 R_3}{R_3 + R_4} \end{aligned}$$

$$\Rightarrow V_0 = V_1 \left(\frac{R_4}{R_3 + R_4} \frac{R_1 + R_2}{R_1} \right) + V_2 \left(\frac{R_3}{R_3 + R_4} \frac{R_1 + R_2}{R_1} \right)$$

GANANCIA RESPECTO V1 = $\frac{R_4}{R_3 + R_4} \frac{R_1 + R_2}{R_1}$

GANANCIA RESPECTO V2 = $\frac{R_3}{R_3 + R_4} \frac{R_1 + R_2}{R_1}$

Ilustración 114. Formulario TeoriaSumNoInv

4.2.2. Módulos del programa

Los módulos del programa son los archivos donde se define todo el código que hace que funcionen los formularios que hemos visto anteriormente. Por lo general, todos los módulos disponen de los mismos procedimientos, pero en cada uno de ellos se desarrollará un código distinto, en función del circuito al que esté asociado el módulo. Por lo general siguen la siguiente estructura:

Se define una función para mostrar el formulario del circuito, llamada *MostrarNombreDelFormulario()*, con la que se ejecuta la sentencia *NombreDelFormulario.Show*.

Para el botón *Teoría* de los circuitos se define la función *MostrarTeoriaNombreDelCircuito()*. En dicha función se ejecutará la sentencia *NombreDelFormularioDeTeoria.Show*. Con esta línea de código se consigue mostrar el formulario del circuito que queremos.

Para el botón de información de cada circuito tenemos la función *MostrarInformacion()*, con la que se mostrará el formulario de información al que este asociado el circuito correspondiente. Esto lo conseguimos con la sentencia *FormularioInformacion.Show*.

Se define un procedimiento para el botón *Calcular* del circuito en concreto. Este procedimiento es el que irá variando para cada circuito, en función de los valores que se introducen en el formulario, los valores que se quieren calcular, y las necesidades de la representación gráfica de cada circuito. Este procedimiento lo comentaremos para cada módulo.



El último procedimiento que se repite en cada uno de los módulos es el referido al botón de restaurar los valores del formulario. A este procedimiento se le denomina *RestaurarValoresNombreDelCircuito()*. En este procedimiento se vaciaran las casillas del formulario correspondiente mediante la sentencia *NombreDelFormulario.VariableDelFormulario.Text = ""* para el caso de los valores que actúan como variables de entrada del formulario, y para los elementos *Label* en los que se han definido los resultados del circuito la sentencia *NombreDelFormulario.VariableDelFormulario.Caption = ""*.

4.2.2.1. Circuitos

El módulo *Circuitos* se encarga de que se muestren por pantalla los formularios del tipo 1 que se han visto en el apartado anterior. Para ello, se define una función *Sub* con el nombre *MostrarNombreDelFormulario()* y en dicha función se ejecuta la sentencia *NombreDelFormulario.Show*.

4.2.2.2. ModCircComp

ModCircComp es el módulo que dispone de los procedimientos que contienen el código de ejecución de los circuitos comparadores. Estos procedimientos son con los que definen el funcionamiento de los botones de los formularios correspondientes a dichos circuitos. Para cada circuito tendremos la secuencia de procedimientos *mostrar formulario*, *mostrar información*, *calcular valores* y *restaurar valores* que se ha comentado anteriormente. Entraremos a analizar el procedimiento de calcular valores, que es el que contiene el código más fundamental del módulo. Este procedimiento es similar para cada circuito, por lo que se analizará solo una vez.

Inicialmente definimos unas variables (*Dim Variable As Single*) con las que se trabajara dentro del módulo. Estas variables estarán referenciadas a los valores que se introducen en el formulario correspondiente. Una vez definidas todas estas variables, ejecutamos la sentencia *On Error Resume Next*, con la que se establece que, de aquí en adelante, todo error que se detecte en el código se omitirá para que permita que se ejecute el módulo al completo.

Primero definimos un proceso *If - Else - End If*, en el que se establece la condición de que, si no se introducen valores en las casillas del formulario, y en el caso de las casillas de las resistencias se introduce un cero, salga un



mensaje de error y vacíe las etiquetas del formulario donde se definen los resultados del cálculo. Si esto no sucede, continuamos dentro del código en el *Else* de este proceso. Llegados a esta situación, asociamos nuestras variables del procedimiento con los elementos del formulario correspondiente (*Variable = NombreFormulario.Variable.Text*) y establecemos otro proceso *If - Else - End If* anidado, en el que se establece la condición de que, si la tensión de saturación negativa es mayor que la positiva, salga un mensaje de error, vacíe las etiquetas del formulario donde se definen los resultados del formulario y no siga ejecutándose el código. Si esto no ha sucedido, el código sigue mediante el *Else* de este proceso. Ahora se definen las ecuaciones con las que obtenemos los valores de salida del circuito. En este caso, para los circuitos comparadores, el valor de la tensión de salida depende del valor de la tensión de entrada. Por lo tanto, establecemos unos procesos *If* en el que se analiza el valor de la tensión de entrada. Si este valor es mayor o menor que otro valor determinado para cada circuito, el valor de la tensión de salida será la saturación positiva o negativa o cero. Al terminar este proceso *If* almacenamos en la etiqueta de la tensión de salida el valor calculado en dicho proceso mediante la variable del procedimiento (*NombreFormulario.Valor_Vo.Caption = Variable_Vo*). Si en otros circuitos debemos calcular otros valores, se repetirá el mismo proceso para dichas variables.

En el caso de haber calculado el valor de la tensión de salida del circuito, como en este caso, debemos comprobar si este valor no supera los valores de saturación del amplificador operacional. Para ello, definimos un proceso *If*, estableciendo las condiciones de que, si la variable correspondiente a *Vo* del procedimiento es mayor o igual que la variable de la tensión de saturación positiva, la etiqueta de la tensión de salida del formulario guarda el valor de la tensión de saturación positiva. A su vez se establece en otro *Elseif* la misma condición, pero para la tensión de saturación negativa.

Por último, debemos pasar los valores que sean necesarios a la hoja de cálculo correspondiente para poder generar la gráfica. Para ello, se establece *ActiveWorkbook.Worksheets("Hoja de cálculo").Range("Celda") = Variable* como sentencia para almacenar dicha variable en la celda que se defina para generar la gráfica. En la hoja de cálculo correspondiente, se definirán ecuaciones mediante los valores de las celdas a las que se pasen los valores de estas variables y generar las gráficas que sean necesarias.

Después de cerrar todos los procesos, establecemos la sentencia *On Error GoTo 0*, con la que paramos el control de errores que se ha arrancado al inicio del módulo.

Esta estructura de código se repite para cada circuito comparador al que hace referencia el módulo, salvo para el disipador de Schmitt. Para este circuito, el desarrollo de código a la hora de pasar los valores necesarios a la hoja de



cálculo correspondiente es más complejo. A parte de pasar valores a celdas, necesitamos ejecutar bucles *For* para ir almacenando los valores que se van obteniendo en función del parámetro del tiempo. Para ello, se necesitan definir variables del tipo *Integer* que nos sirvan de contador para el bucle. A este *For* se le pasa el número de la primera y la última celda de la columna en la que se van guardando estos valores. Al tener que generar dos gráficas, tenemos que definir dos bucles *For*, uno para cada gráfica. Para la gráfica *Vi-t* empleamos la hoja de cálculo "*Gráfica Vo-t*", en donde definimos el valor del tiempo en la columna S. El valor de *Vi* se genera mediante ecuaciones en dicha hoja de cálculo y se va generando un valor de *Vi* para cada valor del tiempo que vamos introduciendo. Para el bucle *For* referido a la gráfica *Vo-t* se repite el mismo proceso, pero con la hoja de cálculo "*Gráfica CosSen*". Además, se añaden sentencias para comprobar si el valor de la tensión de salida que se almacene en las celdas es mayor que los valores de saturación, y si es así, guardar en dichas celdas el valor de la tensión de saturación.

4.2.2.3. ModCircLim

ModCircLim es el módulo en el que se genera el código que hace que funcionen las ventanas de los circuitos limitadores. La estructura del procedimiento *CalcularValores()* sigue la misma línea que para el módulo anterior. A la hora de definir las ecuaciones para obtener los valores que se quieren calcular, se desarrollaran las correspondientes ecuaciones para cada uno de los circuitos limitadores, las cuales ya se definieron al inicio del proyecto. Un cambio significativo en el código de este módulo es el tramo dedicado para generar la representación gráfica. Cada vez que se ejecute el procedimiento *CalcularValores()* de cada circuito se vaciarán todas las casillas de la hoja de cálculo que se emplean para generar las prácticas, pues hay celdas que se comparten para varios circuitos limitadores y pueden generar una gráfica que no sea la correcta. Una vez realizado esto, ya pasamos los valores que se necesitan a las celdas de la hoja de cálculo para cada circuito. Para este caso, comprobamos si la tensión umbral del diodo, o la tensión Zener para el caso de los circuitos que dispongan de diodos Zener, superan los valores de saturación. Esto se realiza mediante procesos *If*.

4.2.2.4. ModCircRectPre

Este módulo recoge el código para que se ejecuten los formularios de los circuitos rectificadores y el circuito para el valor medio y el valor eficaz. Partiendo de la estructura del procedimiento para el botón *Calcular* que se ha comentado anteriormente, de los tres circuitos rectificadores que se disponen caben comentar el código referido a la representación gráfica. Al igual que para el procedimiento referido al circuito disipador de Schmitt, se emplean dos bucles *For*, uno para cada gráfica que se genera. Como se veía en dicho procedimiento, se definen varias variables *Integer* para recorrer dichos bucles. Con estos bucles, vamos definiendo el valor del tiempo en las casillas correspondientes y el valor de la tensión, V_i para uno y V_o para el otro bucle, para cada valor de t que vamos definiendo. Se establecen procesos *If* para comprobar si dichas tensiones son mayores que los valores de saturación. Todo esto acompaña a las sentencias para pasar valores concretos a las celdas concretas de las hojas de cálculo para definir las ecuaciones necesarias para generar las gráficas.

Para el caso del procedimiento para el circuito para el valor medio y el valor eficaz, también se emplea un bucle *For* para generar la gráfica V_i - t , pasándole valores del tiempo a la columna de la hoja de cálculo correspondiente. A partir de estos valores del tiempo, se va generando en otra columna de la hoja de cálculo la tensión de entrada, mediante ecuaciones definidas en esta hoja de cálculo. Para la otra gráfica, V_o - t , analizamos si hemos obtenido a la salida la tensión media o la tensión eficaz. Este valor lo pasamos a la hoja de cálculo correspondiente, generando una señal continua del valor que hemos pasado.

4.2.2.5. ModConvTenCorr y ModConvCorrTen

Estos módulos sirven para ejecutar los formularios de los circuitos convertidores tensión-corriente y corriente-tensión respectivamente. La estructura del procedimiento *CalcularValores()* es muy similar en ambos módulos y sigue la misma línea que para los procedimientos de los anteriores módulos. Para cada procedimiento se definirán las ecuaciones pertinentes usando las variables internas del procedimiento y se pasan los resultados a las etiquetas del formulario donde se definen los resultados. En ambos circuitos no se realiza representación gráfica, por lo que no se define código para ello.

4.2.2.6. ModDetectPico

El módulo *ModDetectPico* es el que contiene el código que define el formulario del circuito detector de pico. En este módulo, aparte de tener una estructura diferente en el procedimiento *CalcularValores()*, contiene un procedimiento a mayores denominado *Interruptor()*, el cual lo comentaremos más adelante.

En cuanto al procedimiento *CalcularValores()*, nos seguimos basando en la estructura que se ha visto en los anteriores apartados. Cabe destacar que, a la hora de ir definiendo los resultados de los valores de pico de la tensión de salida, se va diferenciando cuantas veces damos al botón calcular. Para ello, es necesario definir variables que nos sirvan de contador, de tipo *Integer*, al igual que ya se hizo para el código del formulario del detector de pico. Volviendo con el procedimiento, se establece en el código unas líneas específicas para cuando se ha apretado el botón por primera vez. Almacenamos el valor que se obtiene y se muestra en la etiqueta del formulario. A su vez, vamos guardando el valor en la hoja de cálculo a la vez que vamos generando una onda senoidal. Todo esto lo conseguimos mediante un bucle *For* y varias condiciones *If* analizando si satura o no dicho valor, y comparando entre el valor de la onda para cada instante de tiempo y el valor del instante anterior. Una vez pasada la primera vez que hemos presionado el botón de calcular, desarrollamos otras líneas de código para cuando hemos presionado dos o más veces dicho botón. Para este caso, se repiten las líneas del caso en el que se presiona una sola vez el botón, pero se añade un proceso *If* para ir analizando, con la variable de la tensión de pico máxima, cuándo conseguimos el máximo valor. Esto se consigue comparando el valor de pico que se consigue en la iteración en la que se ha presionado el botón con el valor de la iteración anterior.

Al final de este procedimiento, vamos recorriendo mediante un *For* el número de veces que se ha presionado el botón *Calcular* y almacenamos en la hoja de cálculo un contador iniciado a cero para generar en la gráfica los valores de pico máximos que vamos obteniendo.

En cuanto al procedimiento *Interruptor()*, su objetivo es que si se presiona el botón *Interruptor*, se vacíen todos los valores de pico que se han ido almacenando cada vez que hemos presionado el botón. En el código definimos primero un proceso *If* analizando si se ha obtenido al menos una vez un valor de pico de la tensión de salida. Si no se ha obtenido, saldrá un mensaje de error diciendo que no se han almacenado valores. Si ya se ha presionado el botón *Calcular* al menos una vez, podremos presionar el botón *Interruptor*, con el que se vacían y se reinician todas las variables empleadas para ir obteniendo los valores de pico para cada iteración y los valores de pico máximo que se han obtenido hasta ese momento.



4.2.2.7. ModDiferenciador

ModDiferenciador contiene el código para la ejecución del formulario del circuito diferenciador. El procedimiento para el botón de calcular tiene la diferencia, en comparación con este mismo procedimiento del resto de módulos, de que se define un código diferente en función de si la potencia a la que se eleva el valor del tiempo es la unidad o mayor que la unidad en el tramo en el que se genera la representación gráfica. Para cada uno de los casos se le pasan los valores necesarios a la hoja de cálculo para que se genere la gráfica. En el caso de que la potencia sea uno, la representación será una recta, por lo que se debe analizar si el valor continuo que se obtiene supera el valor de saturación del A.O.

4.2.2.8. ModFuenteCorriente y ModFuenteTension

Estos módulos ejecutan los formularios de las fuentes de corriente y de tensión respectivamente. El procedimiento *CalcularValores()* sigue la estructura típica que se viene repitiendo en los anteriores módulos. El tramo en el que se calcula los parámetros de salida es lo que se va modificando para cada caso. Para el primer ejemplo de fuente de corriente, no se calcula la tensión de salida, por lo que no se realiza el código en el que se analiza si satura o no la tensión de salida del circuito.

4.2.2.9. ModHisteresis

ModHisteresis es el módulo que contiene el código para los formularios de los circuitos comparadores con histéresis. En referencia a los procedimientos para el botón de calcular, sigue el mismo patrón de estructura que se viene comentando. Después de establecer la condición de que la tensión de saturación negativa no sea mayor que la de saturación positiva, definimos las ecuaciones necesarias para obtener las variables que se quieren calcular. Después comprobamos si la tensión de salida satura o no, a lo que le sigue las líneas de código para la representación gráfica. Como la hoja de cálculo a la que pasamos los valores comparte celdas entre los circuitos comparadores con



histéresis, para cada circuito debemos de vaciar varias celdas a parte de las sentencias con las que pasamos los valores necesarios a otras celdas para generar la gráfica.

4.2.2.10. ModIntInv y ModIntNoInv

Los módulos *ModIntInv* y *ModIntNoInv* son los que contienen el código para los dos casos de los circuitos integrador inversor y no inversor respectivamente. Los procedimientos *CalcularValores()* de ambos módulos son prácticamente iguales salvo el código para definir las ecuaciones para calcular los valores que se desean, por lo que comentaremos dichos procedimientos una sola vez para ambos.

Para el procedimiento de los circuitos tipo 1 cabe comentar el código referido a la representación gráfica. Vaciamos las celdas correspondientes para el circuito tipo 2. Una vez realizado esto, le pasamos los valores necesarios a la hoja de cálculo “*GráficaCosSen*”. Para ello, se hace diferencia entre si la potencia a la que elevamos el valor del tiempo es cero o mayor que cero. Para cada caso, le pasaremos unos valores a ciertas casillas y vaciaremos otras. Por último, pasamos los valores correspondientes a las celdas de la hoja de cálculo “*Gráfica Vo-t*” y vaciamos las que hagan falta para generar la otra gráfica.

Para el procedimiento de los circuitos tipo 2, de nuevo analizamos el código para la representación gráfica. Comenzamos vaciando las celdas de las hojas de cálculo referidas al circuito tipo 1 y pasamos los valores necesarios a la hoja de cálculo “*Gráfica CosSen*”. Acto seguido realizamos proceso *If - Elseif - End If* en el que diferenciamos entre si definimos senoidal “*sen*” o cosenoidal “*cos*” la señal de entrada. Para cada caso, regeneramos los valores de las celdas que se emplean para realizar la gráfica vaciando dichas celdas. A su vez pasamos las variables *a* y *b* a sus celdas correspondientes en las dos hojas de cálculo. Acto seguido se define un bucle *For*, con el que vamos definiendo el valor del tiempo en ambas hojas de cálculo, rellenando así la columna de esas hojas de cálculo. Esto hará que, a su vez, se vaya generando los valores de tensión en ambas hojas de cálculo. Después se analiza si la tensión de salida para cada valor del tiempo satura.



4.2.2.11. ModIntRes y ModIntSum

ModIntRes y *ModIntSum* son los módulos referidos a los formularios de los circuitos integradores restadores y sumadores. Ambos circuitos disponen de dos tipos, como para el caso anterior. Lo destacado de los procedimientos *CalcularValores()* para ambos módulos son las ecuaciones que se definen para obtener los valores que se quieren obtener. Se analizan para el caso de los circuitos tipo 2 la selección entre “sen” y “cos” de cada una de las señales de entrada para los dos circuitos, y se define el resultado de la tensión de salida para todas las combinaciones que se tengan. Para los de tipo 1, se definen las ecuaciones necesarias.

4.2.2.12. ModInversor y ModNoInversor

Los módulos *ModInversor* y *ModNoInversor* son los que contienen el código para ejecutar los formularios de los circuitos inversor y no inversor respectivamente. Los procedimientos para el botón calcular son muy similares para ambos módulos. Se diferencian en las ecuaciones que se definen para obtener los resultados. La estructura de dichos procedimientos sigue la misma línea que la que se viene repitiendo.

4.2.2.13. ModOndaCua

ModOndaCuadrada es el módulo que contiene el código para el formulario del circuito generador de onda cuadrada. Como se viene repitiendo, nos centraremos en su procedimiento *CalcularValores()*, pues el resto sigue la misma estructura que se explicó al principio del apartado de los módulos. Dentro del procedimiento nos centraremos en el trozo de código dedicado a la representación gráfica. Para poder pasar correctamente los valores necesarios a la hoja de cálculo, vamos a vaciar todas las columnas de celdas empleadas. Una vez que se hayan vaciado, comenzamos a introducir valores. Para conseguir generar la onda cuadrada, vamos a desarrollar un bucle *For*. Dentro de este, vamos a ir realizando un proceso *If* para cada tramo de subida y de bajada de la señal V_c . Definimos los valores necesarios para generar el primer tramo de la onda V_c y la de V_o . Cuando la onda de V_c haya superado el valor máximo definido por teoría, saldremos del *If* para entrar en otro *If* anidado al



mismo. Se vuelve a plantear la misma condición, pero en este caso se analiza si la onda de V_c es menor al valor en el que la onda debe invertirse. Este proceso se repite 20 veces, consiguiendo así una onda cuadrada generada con 20 periodos de onda de V_c . Una vez que se haya salido de estos procesos *If* anidados, analizamos si el valor de V_o satura para cada caso. Finalmente, con otro bucle *For* definimos los valores del tiempo que se necesitan pasar a la hoja de cálculo para que se pueda generar la gráfica en la hoja de cálculo. Para que se puedan realizar todas las ecuaciones definidas en la hoja de cálculo, pasamos los valores de la frecuencia y el periodo, además de un cero, a las celdas correspondientes.

4.2.2.14. ModOndaTri

El módulo *ModOndaTri* es el que contiene todo el código para que se ejecute el formulario del circuito generador de onda triangular. En cuanto al procedimiento del botón *Calcular*, sigue la misma dinámica que se ha comentado para el módulo *ModOndaCuadrada* en cuanto al código referido a la representación gráfica, por lo que no se explicará. El resto del procedimiento sigue la misma línea de realización que para los anteriores casos.

4.2.2.15. ModRestador y ModSumador

Los dos últimos módulos del programa son *ModRestador* y *ModSumador*, los cuales contienen el código para los formularios de los circuitos restadores y sumadores respectivamente. En ambos módulos disponemos de dos procedimientos referidos al botón para calcular los valores del circuito, pues se disponen de dos tipos de circuitos para cada grupo de circuitos. En cada uno de ellos, el código sigue la misma estructura que se ha estado repitiendo para cada circuito, variando las sentencias para definir las ecuaciones con las que se obtienen las variables a calcular. Para estos circuitos, no es necesario definir el código para la representación gráfica, pues no se genera gráfica en estos circuitos.

4.3. Funcionamiento del programa

El programa arranca y la pantalla principal que aparece es la que se muestra en la *ilustración 115*:

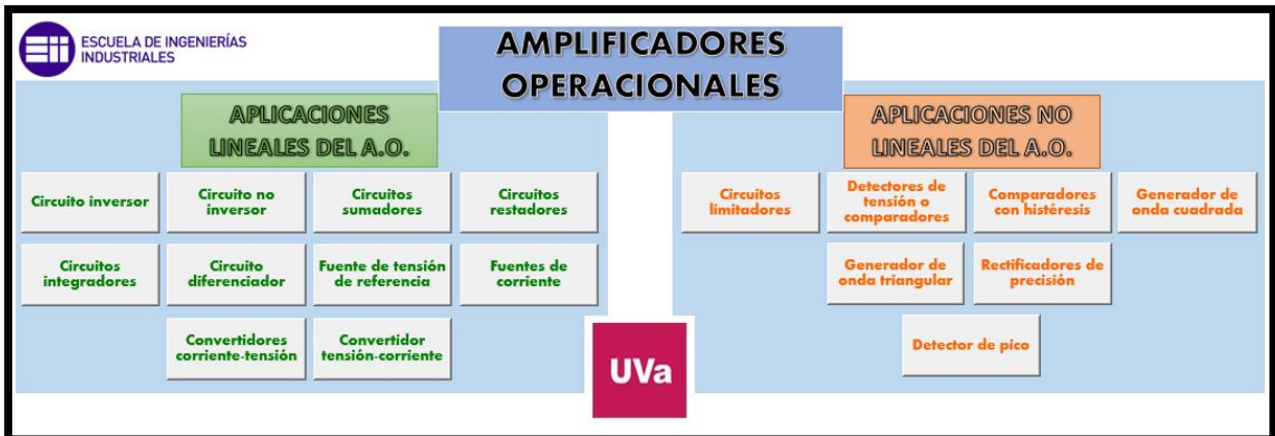


Ilustración 115. Pantalla principal del programa

Como se puede apreciar, en el centro de la pantalla se define el título “*Amplificadores operacionales*”, haciendo referencia al elemento sobre el que se centra el programa. En el lado izquierdo se muestran los circuitos con aplicaciones lineales del A.O., en tonos verdes, y a la derecha los circuitos con aplicaciones no lineales del A.O., en tonos anaranjados. En cada uno de estos bloques, se definen botones para cada uno de los circuitos de los que dispone el programa. A parte de lo que se ha comentado, se han introducido dos imágenes sobre la pantalla principal, haciendo referencia a la Universidad de Valladolid y a la Escuela de Ingenierías Industriales de Valladolid.

Para comprobar el funcionamiento de la aplicación, se realiza un ejemplo con uno de los circuitos. Se ha elegido como ejemplo de muestra el circuito inversor.

Desde la pantalla principal (Ilustración 115), clicamos en el botón “*Circuito inversor*” y nos aparece la siguiente pestaña:

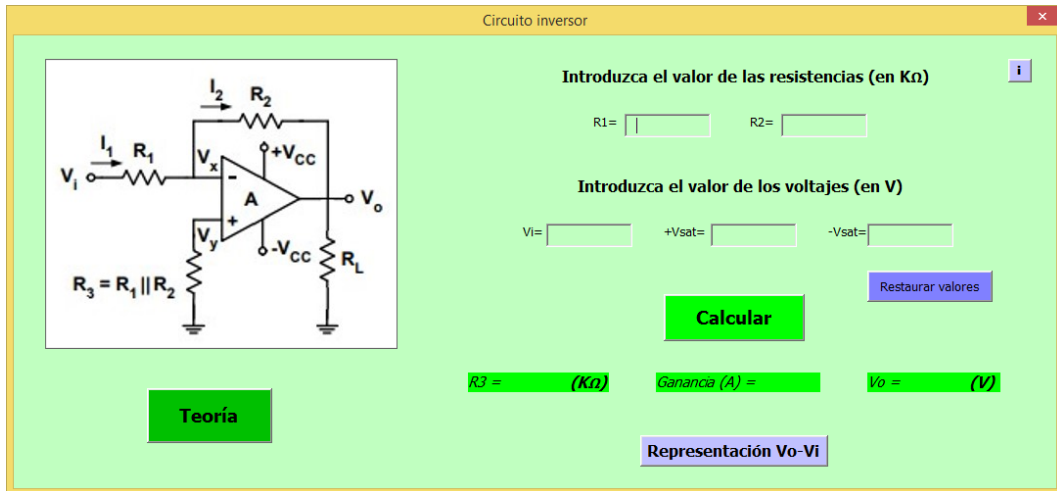


Ilustración 116. Pestaña del circuito inversor

Para rubricar su funcionamiento, introduciremos algunas erratas en las casillas donde se definen los valores de los parámetros.

Primero probamos a introducir un valor nulo en las casillas de las resistencias. Para este caso, introducimos un 0 en la casilla de R1. Al apretar el botón “Calcular” nos salta el siguiente mensaje:

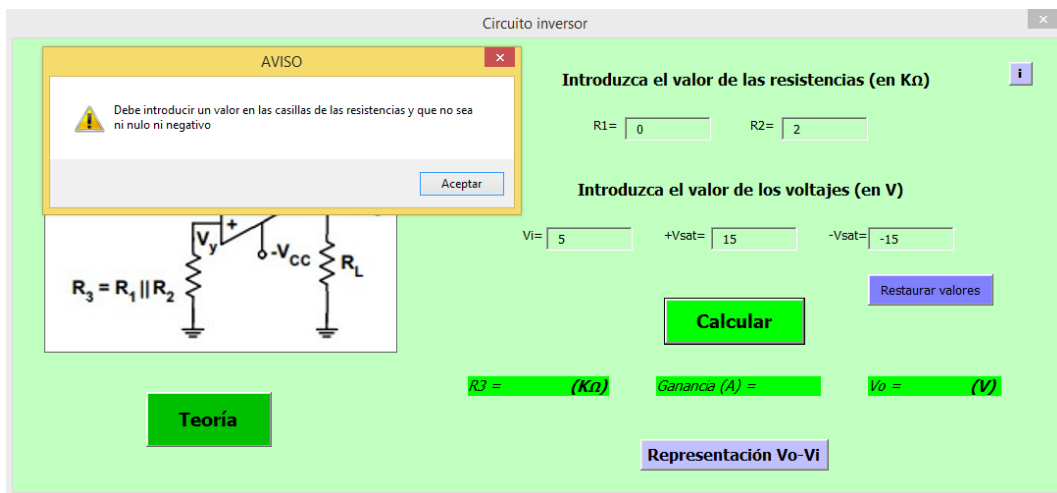


Ilustración 117. Mensaje de error del circuito inversor I

Este mensaje saltará cada vez que no se introduzca algún valor en las casillas de las resistencias o se introduzca un valor nulo, como puede ser un valor negativo o un cero, como en este caso.

Ahora probamos a no introducir ningún valor en las casillas de los voltajes, en concreto para la casilla de V_i en este caso. El resultado es el siguiente:

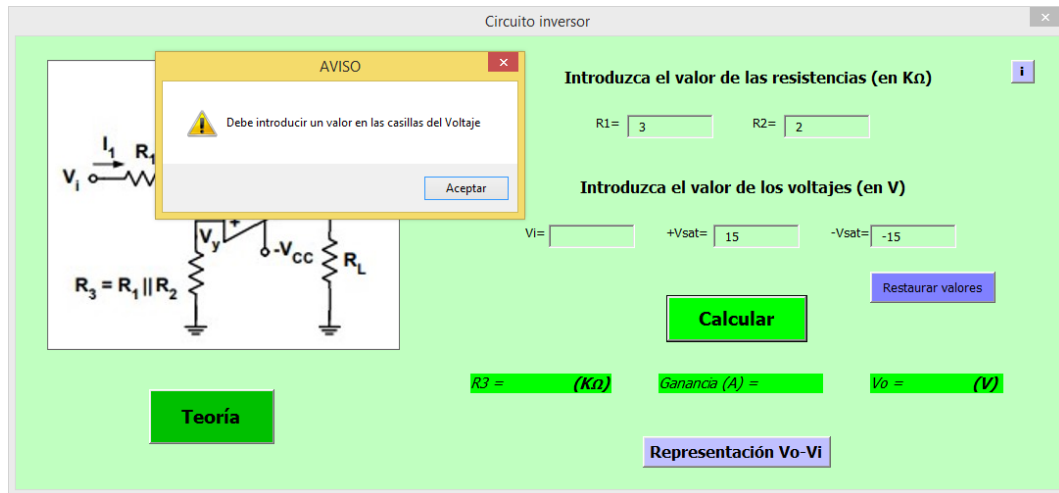


Ilustración 118. Mensaje de error del circuito inversor II

Como vemos en la *ilustración 118*, nos salta un mensaje diciendo que se debe de introducir un valor en las casillas del voltaje. Esto sucederá cada vez que se deje vacía una casilla del voltaje y presionemos el botón de calcular.

Por último, probamos a definir un valor de la tensión de saturación negativa del A.O. mayor que la positiva. Si clicamos en “Calcular”, nos salta el siguiente mensaje:

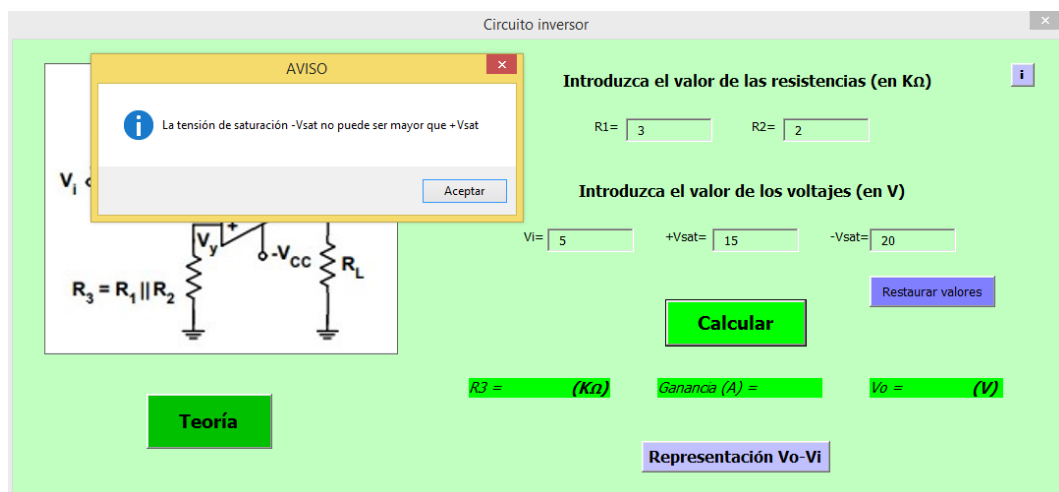


Ilustración 119. Mensaje de error del circuito inversor III

Tras dar al botón “Aceptar” de este mensaje, se borrarán los valores introducidos en las casillas de las tensiones de saturación del A.O. y los resultados obtenidos del circuito si se habían obtenido anteriormente unos valores con parámetros correctos.

Evitando todos estos fallos e introduciendo unos valores válidos, al pinchar el botón de calcular, obtendremos los resultados de los valores a calcular del circuito, tal y como se puede ver en la siguiente ilustración:

The screenshot shows a software window titled "Circuito inversor". On the left is a circuit diagram of an inverting amplifier with an operational amplifier (A), input resistor R_1 , feedback resistor R_2 , and load resistor R_L . The input voltage is V_i and the output is V_o . The feedback network is labeled $R_3 = R_1 \parallel R_2$. On the right, there are input fields for resistances (R1=3, R2=2) and voltages (Vi=5, +Vsat=15, -Vsat=-15). A "Calcular" button is highlighted in green. Below the inputs, the results are displayed: $R_3 = 1,2 \text{ (K}\Omega\text{)}$, $\text{Ganancia (A)} = -0,6666$, and $V_o = -3,3333 \text{ (V)}$. A "Teoría" button is located below the circuit diagram, and a "Representación Vo-Vi" button is at the bottom.

Ilustración 120. Resultados del circuito inversor

Tras obtener unos resultados del circuito, podremos clicar el botón “Representación Vo-Vi”, con el que se muestra una nueva ventana en donde viene definida la gráfica Vo-Vi del circuito. Si por el contrario no se hubieran obtenido ningún valor de los parámetros a calcular, al presionar el botón de la representación gráfica, nos habría saltado un mensaje de error como el siguiente:

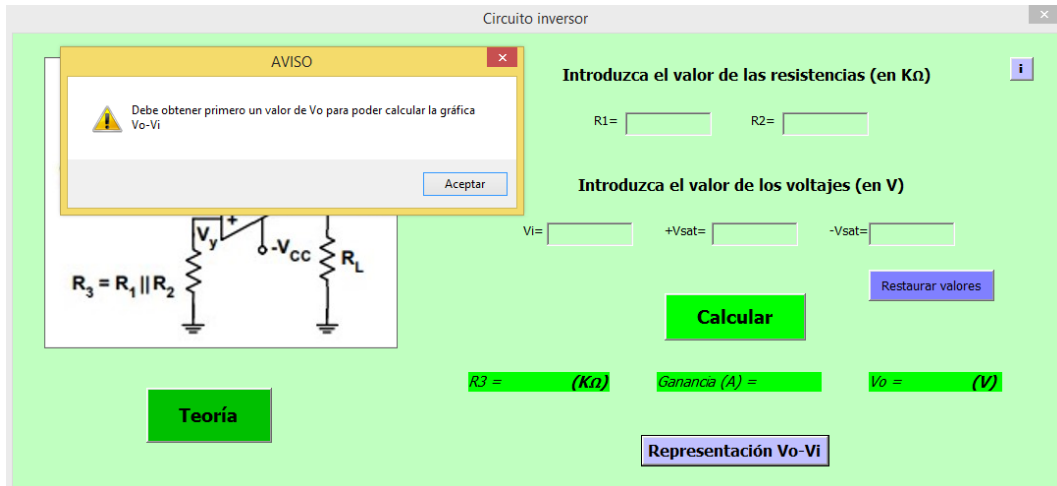


Ilustración 121. Error de representación gráfica del circuito inversor

Como se había indicado antes, si se introducen valores válidos y obtenemos los resultados de los parámetros a calcular en el circuito, podremos obtener la representación gráfica de los resultados, la cual se muestra en la *ilustración 122*:

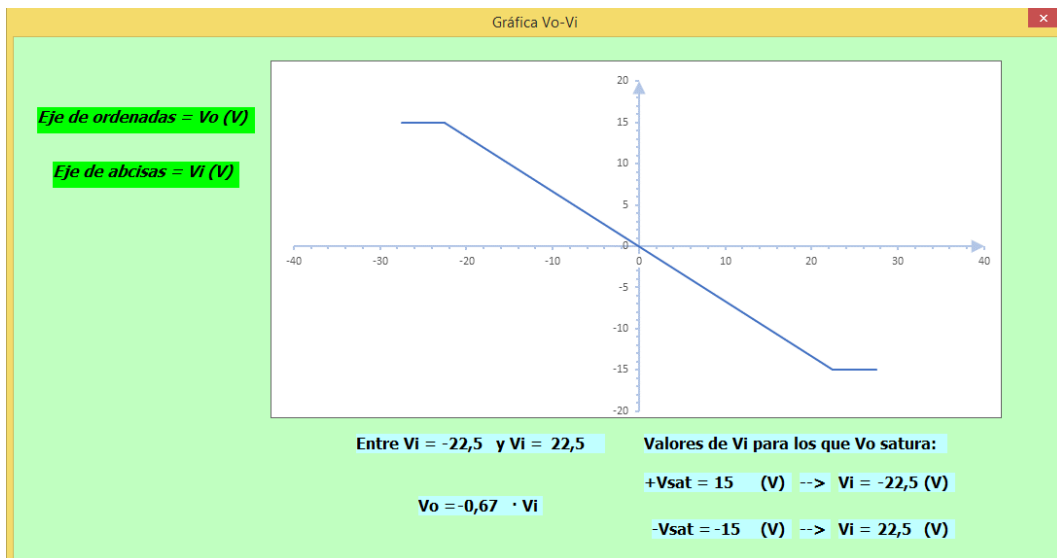
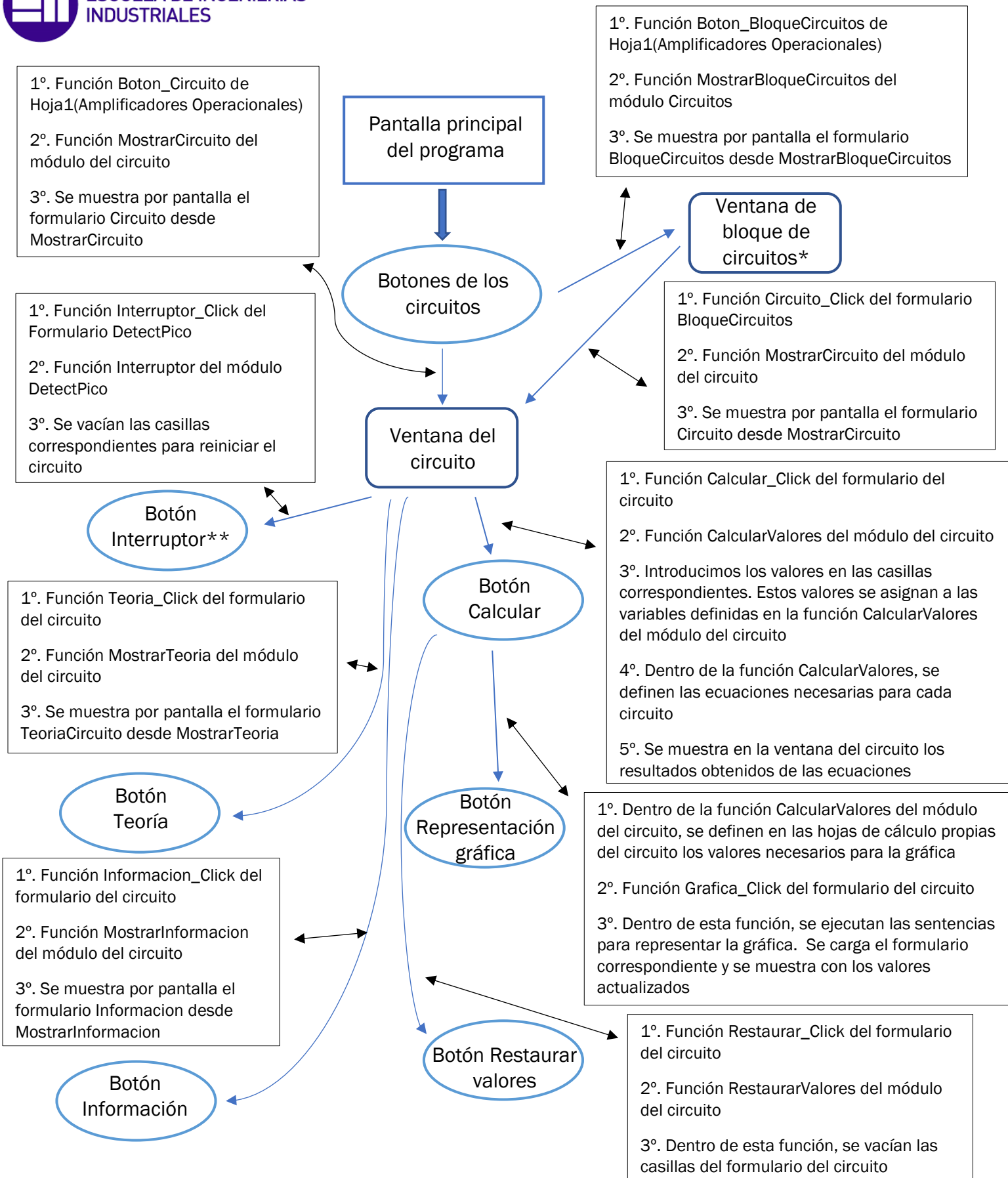


Ilustración 122. Representación gráfica del circuito inversor

Aunque el código de todo el programa es muy explicativo y dispone de muchos comentarios para comprender cómo se ha desarrollado, se ha realizado un esquema demostrativo de cómo funciona el programa para que sea más sencillo de comprender.



*Ventana para los siguientes bloques de circuitos: Circuitos sumadores, Circuitos restadores, Circuitos integradores, Circuitos limitadores, Detectores de tensión, Comparadores con histéresis y Rectificadores de precisión.

**El botón Interruptor es únicamente para el circuito Detector de pico.



5. Conclusiones

La realización de este proyecto cumple con los objetivos marcados al inicio del mismo. Se ha conseguido una herramienta de trabajo que nos sirva de gran apoyo a la hora de estudiar circuitos con amplificadores operacionales y que proporciona datos que son relevantes para comprender el funcionamiento de cada uno de los circuitos incluidos en el programa.

Cabe comentar que, salvo algunas excepciones, los resultados de cada uno de los circuitos del programa se proporcionan al instante. Las excepciones a las que se hace alusión tienen lugar en algunos de los circuitos que generan gráficas representando las tensiones de salida y entrada en función del tiempo. Debido a que se deben generar los instantes de tiempo para los que queremos obtener las señales de tensión, la gráfica aparecerá por pantalla pasados unos 10 segundos aproximadamente de presionar al botón de representar la gráfica.

Esta aplicación ha sido elaborada mientras se aprendía a manejar la herramienta con la que se ha realizado, lo que implica que puede que el código realizado pueda ser optimizable. Aunque esto fuese así, se garantiza un correcto funcionamiento de la aplicación.





6. Referencias bibliográficas

Ilustraciones y ecuaciones del apartado “3. Análisis de los Amplificadores Operacionales”:

BUEY CUESTA, José Julio. *Diapositivas de los temas 6, 7 y 8 de la asignatura Electrónica Analógica del grado en Ingeniería Electrónica Industrial y Automática.*

Imágenes de los circuitos del programa y la teoría de cada uno de ellos:

BUEY CUESTA, José Julio. *Diapositivas de los temas 6, 7 y 8 de la asignatura Electrónica Analógica del grado en Ingeniería Electrónica Industrial y Automática.*

<http://moisesrbb.tripod.com/vb.html>

BURGOS, Moisés. *Curso de Visual Basic*

<https://excelyvba.com/curso-de-vba/>

ARRANZ, Quique. *Curso de VBA para Excel*

<https://byspel.com/variables-y-constantes-en-visual-basic-net/2/>

ANÓNIMO. *Variables y constantes de VBA, por Byspel*

<https://www.todoexpertos.com/categorias/tecnologia-e-internet/software-y-aplicaciones/microsoft-excel/respuestas/1951967/graficos-en-formularios-vba>

ANÓNIMO. *Gráficos en formularios VBA*

<http://miexcelymas.blogspot.com/2013/08/mostrar-graficos-en-formularios.html>

QUISPE, Luis. *Mostrar Gráficos en formularios UserForm de VBA*



<https://support.office.com/en-us/article/Overview-of-forms-Form-controls-and-ActiveX-controls-on-a-worksheet-15BA7E28-8D7F-42AB-9470-FFB9AB94E7C2>

Kathleen Dollard, olprod, yishengjin1413 y Saisang Cai (usuarios de github.com). Información general de formularios, controles de formulario y controles ActiveX en una hoja de cálculo

<https://docs.microsoft.com/es-es/dotnet/visual-basic/language-reference/statements/on-error-statement>

Kathleen Dollard, olprod, OpenLocalizationService y Saisang Cai (usuarios de github.com). Información general de formularios, controles de formulario y controles ActiveX en una hoja de cálculo



7. Anexos



CONTENIDO DE ANEXOS

7.1.	Aplicaciones de los amplificadores operacionales	1
7.1.1.	Aplicaciones lineales de los amplificadores operacionales	1
7.1.1.1.	Circuito inversor	1
7.1.1.2.	Circuito no inversor	2
7.1.1.3.	Circuitos sumadores	3
7.1.1.4.	Circuitos restadores	5
7.1.1.5.	Circuitos integradores	8
7.1.1.6.	Circuito diferenciador	13
7.1.1.7.	Fuentes de tensión de referencia	13
7.1.1.8.	Fuentes de corriente	15
7.1.1.9.	Convertidores corriente-tensión y tensión-corriente	17
7.1.2.	Aplicaciones no lineales de los amplificadores operacionales	20
7.1.2.1.	Circuitos limitadores	20
7.1.2.2.	Detectores de tensión o comparadores	25
7.1.2.3.	Comparadores con histéresis	29
7.1.2.4.	Generador de onda cuadrada	34
7.1.2.5.	Generador de onda triangular	36
7.1.2.6.	Rectificadores de precisión	38
7.1.2.7.	Detectores de pico	45
7.2.	Código del programa	47
7.2.1.	Código de hojas Excel	47
7.2.1.1.	Hoja 1 (Amplificadores operacionales)	47
7.2.1.2.	ThisWorkbook	49
7.2.2.	Código de formularios	49
7.2.2.1.	Detectores de tensión o comparadores (CircCom)	49
7.2.2.2.	Comparadores con histéresis (CircHis)	50
7.2.2.3.	Circuitos integradores (CircInt)	50
7.2.2.4.	Circuitos limitadores (CircLim)	51
7.2.2.5.	Rectificadores de precisión (CircRectPre)	51
7.2.2.6.	Circuitos restadores (CircRes)	52
7.2.2.7.	Circuitos sumadores (CircSum)	52
7.2.2.8.	Detector con salida Cero-Sat (CompCeroSat)	52

7.2.2.9.	Detector con salida Sat-Sat (CompSatSat).....	54
7.2.2.10.	Disipador de Schmitt (CompSchmitt)	55
7.2.2.11.	Comparador Simple (CompSimple).....	57
7.2.2.12.	Convertidores corriente-tensión (ConvCorrTen).....	58
7.2.2.13.	Convertidores tensión-corriente (ConvTenCorr).....	59
7.2.2.14.	Detector de pico (DetectPico)	59
7.2.2.15.	Circuito diferenciador (Diferenciador)	61
7.2.2.16.	Fuentes de corriente (FuenteCorriente)	62
7.2.2.17.	Fuente de tensión de referencia (FuenteTension).....	63
7.2.2.18.	GraficaCosSen.....	64
7.2.2.19.	GraficaDetectPico	68
7.2.2.20.	GraficaDif.....	68
7.2.2.21.	GraficaHisteresis.....	72
7.2.2.22.	GraficaSchmitt.....	74
7.2.2.23.	GraficaVot.....	81
7.2.2.24.	GraficaVot_2.....	84
7.2.2.25.	GraficaVoVi	85
7.2.2.26.	GraficaVoVi_2.....	87
7.2.2.27.	Comparador con histéresis (Histeresis1)	95
7.2.2.28.	Comparador con histéresis y limitación a la salida (Histeresis2)	97
7.2.2.29.	Integrador inversor (IntegradorInversor)	99
7.2.2.30.	Integrador no inversor (IntegradorNoInversor)	102
7.2.2.31.	Integrador restador (IntegradorRestador)	105
7.2.2.32.	Integrador sumador (IntegradorSumador)	106
7.2.2.33.	Circuito inversor (Inversor)	107
7.2.2.34.	Limitador con diodo normal (LimitadorDiodoNormal)	108
7.2.2.35.	Limitador con diodo normal y diodo Zener en serie (LimitadorDiodoNormalDiodoZener).....	110
7.2.2.36.	Limitador con dos diodos Zener (LimitadorDosZener)	112
7.2.2.37.	Circuito no inversor (NoInversor)	114
7.2.2.38.	Generador de onda cuadrada (OndaCuadrada)	115
7.2.2.39.	Generador de onda triangular (OndaTriangular).....	116
7.2.2.40.	Rectificador de media onda (RectMediaOnda).....	117
7.2.2.41.	Rectificador de onda completa (RectOndaCompleta)	118

7.2.2.42.	Restador de alta impedancia a la entrada (RestadorAltaImp).....	120
7.2.2.43.	Restador simple (RestadorSimple)	121
7.2.2.44.	Sumador inversor (SumadorInversor).....	121
7.2.2.45.	Sumador no inversor (SumadorNoInversor).....	122
7.2.2.46.	Circuitos de valor medio y valor eficaz (ValorMedioValorEficaz).....	122
7.2.3.	Código de módulos	124
7.2.3.1.	Botones de la hoja principal (Circuitos).....	124
7.2.3.2.	Detectores de tensión o comparadores (ModCircComp)	125
7.2.3.3.	Circuitos limitadores (ModCircLim).....	140
7.2.3.4.	Rectificadores de precisión (ModCircRectPre).....	169
7.2.3.5.	Convertidores corriente-tensión (ModConvCorrTen)	187
7.2.3.6.	Convertidor tensión-corriente (ModConvTenCorr)	192
7.2.3.7.	Detector de pico (ModDetectPico).....	195
7.2.3.8.	Diferenciador (ModDiferenciador)	201
7.2.3.9.	Fuentes de corriente (ModFuenteCorriente).....	205
7.2.3.10.	Fuente de tensión de referencia (ModFuenteTension)	209
7.2.3.11.	Comparadores con histéresis (ModHisteresis)	212
7.2.3.12.	Integrador inversor (ModIntInv).....	224
7.2.3.13.	Integrador no inversor (ModIntNoInv).....	234
7.2.3.14.	Integrador restador (ModIntRes).....	245
7.2.3.15.	Integrador sumador (ModIntSum)	251
7.2.3.16.	Circuito inversor (ModInversor).....	259
7.2.3.17.	Circuito no inversor (ModNoInversor).....	262
7.2.3.18.	Generador de onda cuadrada (ModOndaCua).....	265
7.2.3.19.	Generador de onda triangular (ModOndaTri).....	273
7.2.3.20.	Circuitos restadores (ModRestador).....	281
7.2.3.21.	Circuitos sumadores (ModSumador).....	287

ÍNDICE DE ILUSTRACIONES ANEXOS

Ilustración 123. Circuito inversor.....	1
Ilustración 124. Representación Vo-Vi del circuito inversor	2
Ilustración 125. Circuito no inversor	2
Ilustración 126. Representación Vo-Vi del circuito no inversor	3
Ilustración 127. Circuito sumador inversor.....	4
Ilustración 128. Circuito sumador no inversor.....	4
Ilustración 129. Circuito restador simple.....	5
Ilustración 130. Restador alta impedancia de entrada con 3 A.O.	7
Ilustración 131. Restador alta impedancia de entrada con 2 A.O.	7
Ilustración 132. Circuito integrador inversor	8
Ilustración 133. Circuito integrador inversor con interruptores	9
Ilustración 134. Representación Vo-t de integrador inversor	10
Ilustración 135. Circuito integrador no inversor	10
Ilustración 136. Circuito integrador sumador	11
Ilustración 137. Circuito integrador restador.....	12
Ilustración 138. Integrador restador con 2 A.O.	12
Ilustración 139. Circuito diferenciador.....	13
Ilustración 140. Fuente con pila patrón	14
Ilustración 141. Fuente con diodo zener	14
Ilustración 142. Fuente de referencia de doble polaridad	15
Ilustración 143. Ejemplo 1 fuente de corriente.....	16
Ilustración 144. Ejemplo 2 fuente de corriente.....	16
Ilustración 145. Convertidor corriente-tensión con $k < 0$	17
Ilustración 146. Convertidor corriente-tensión con $k > 0$	18
Ilustración 147. Convertidor tensión-corriente.....	19
Ilustración 148. Circuito limitador con diodo normal I	20
Ilustración 149. Circuito limitador con diodo normal II.....	21
Ilustración 150. Circuito limitador con diodo normal y diodo zener I.....	22
Ilustración 151. Circuito limitador con diodo normal y diodo zener II.....	23
Ilustración 152. Circuito limitador con dos diodos zener.....	24
Ilustración 153. Comparador simple	26
Ilustración 154. Comparador con salida saturación-saturación	27
Ilustración 155. Comparador con salida cero-saturación I.....	27

Ilustración 156. Comparador con salida cero-saturación II.....	28
Ilustración 157. Disipador de Schmitt.....	29
Ilustración 158. Señal de salida de un comparador simple distorsionada	30
Ilustración 159. Señal de salida de un comparador simple con histéresis.....	30
Ilustración 160. Comparador con histéresis.....	31
Ilustración 161. Representación Vo-Vi de un comparador con histéresis.....	32
Ilustración 162. Comparador con histéresis y limitación de salida I.....	33
Ilustración 163. Representación Vo-Vi de un comparador con histéresis y limitación de salida I	33
Ilustración 164. Comparador con histéresis y limitación de salida II.....	33
Ilustración 165. Representación Vo-Vi de un comparador con histéresis y limitación de salida II	34
Ilustración 166. Circuito generador de onda cuadrada	34
Ilustración 167. Representación Vo-t del circuito generador de onda cuadrada.....	36
Ilustración 168. Circuito generador de onda triangular	37
Ilustración 169. Representación Vo-t del circuito generador de onda triangular	38
Ilustración 170. Señal Vi del rectificador de media onda	39
Ilustración 171. Rectificador de media onda.....	39
Ilustración 172. Representación Vo-t del rectificador de media onda.....	40
Ilustración 173. Rectificador de onda completa I	40
Ilustración 174. Representación Vo-t del rectificador de onda completa I.....	41
Ilustración 175. Rectificador de onda completa II	42
Ilustración 176. Representación Vo-t del rectificador de onda completa II.....	42
Ilustración 177. Circuito del valor medio y valor eficaz.....	43
Ilustración 178. Representación Vi-t Vo-t del circuito del valor medio y valor eficaz..	43
Ilustración 179. Circuito detector de pico	45
Ilustración 180. Representación Vi-t Vo-t del circuito detector de pico	46

7.1. Aplicaciones de los amplificadores operacionales

En este apartado se van a mostrar una gran cantidad de casos prácticos en los que podemos emplear un amplificador operacional. Dividiremos estos casos en dos grandes grupos; aplicaciones lineales, cuando la salida es una función lineal de la entrada, y aplicaciones no lineales, cuando la salida no depende de la señal de entrada de forma lineal y entran en juego otros factores que se comentarán más adelante.

7.1.1. Aplicaciones lineales de los amplificadores operacionales

7.1.1.1. Circuito inversor

Mediante ese montaje, mostrado en la *ilustración 1*, conseguimos en la salida invertir la señal que introducimos a la entrada.

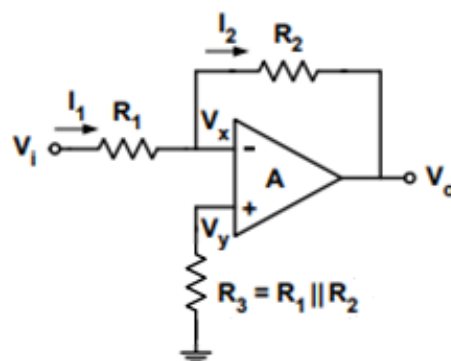


Ilustración 123. Circuito inversor

La patilla no inversora está conectada a tierra, por lo que V_x y V_y tendrán valor nulo.

La corriente I_1 que circula por la patilla inversora será la misma corriente que circula por la realimentación, I_2 , obteniendo la siguiente ecuación:

$$V_o = -V_i \cdot \frac{R_2}{R_1} \quad (1)$$

Siendo la ganancia $\left(-\frac{R_2}{R_1}\right)$.

La función de transferencia resultante se muestra a continuación, siendo la pendiente de la zona lineal la ganancia.

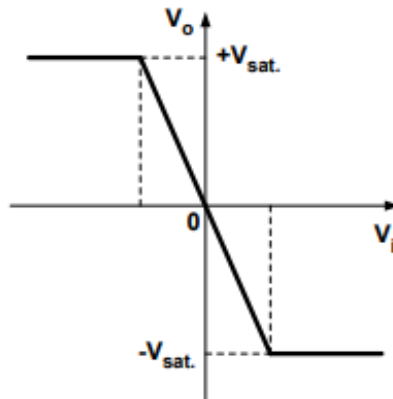


Ilustración 124. Representación V_o - V_i del circuito inversor

7.1.1.2. Circuito no inversor

Es similar al circuito inversor, en cuanto a montaje del circuito se refiere, pero con la gran diferencia de que se conecta en la patilla no inversora la señal de entrada y la patilla inversora se conecta a tierra. Esto implica que en la salida obtengamos una señal con la misma fase que la de la entrada.

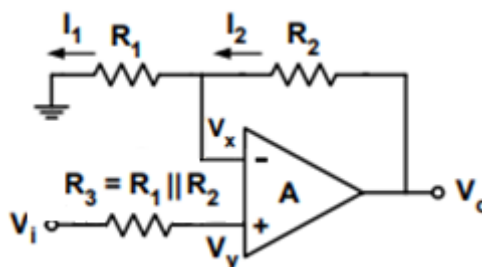


Ilustración 125. Circuito no inversor

Tenemos que:

$$V_i = V_y \rightarrow V_x = V_i \quad (2)$$

Como en este caso, la corriente que circula por la realimentación (I_2) será la misma que circule por la patilla inversora (I_1), tenemos que:

$$V_o = V_i \cdot \frac{R_1 + R_2}{R_1} \quad (3)$$

Siendo la ganancia $\left(\frac{R_1 + R_2}{R_1}\right)$.

La función de transferencia resultante se muestra en la *ilustración 4*, siendo la pendiente de la zona lineal la ganancia.

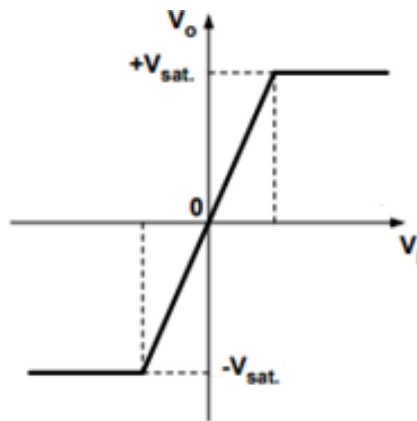


Ilustración 126. Representación Vo-Vi del circuito no inversor

7.1.1.3. Circuitos sumadores

Lo que se busca con estos circuitos es juntar en la salida del amplificador operacional dos señales que se han introducido en la entrada del mismo. En función de donde conectamos las dos señales que queremos sumar, conseguiremos a la salida la suma de ambas señales de manera invertida o no invertida.

→ SUMADOR INVERSOR

En este caso, las dos señales que queremos sumar las conectamos en la patilla inversora. Como en el caso del circuito inversor, la patilla no inversora está conectada a tierra, de tal forma que V_x y V_y tendrán valor nulo.

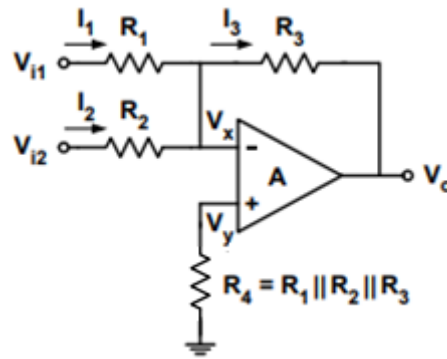


Ilustración 127. Circuito sumador inversor

Ya que la corriente de entrada por el amplificador operacional es cero, la suma de las corrientes que proporcionan ambas tensiones por la patilla inversora será la misma que la que circula por la realimentación, es decir, $I_3 = I_1 + I_2$. De esta forma, conseguimos la siguiente ecuación:

$$V_o = - \left(V_{i1} \cdot \frac{R_3}{R_1} + V_{i2} \cdot \frac{R_3}{R_2} \right) \quad (4)$$

→ SUMADOR NO INVERSOR

Si las señales de tensión que queremos sumar se conectan a la patilla no inversora, entonces se trata de un circuito sumador no inversor. El circuito es prácticamente similar al del circuito no inversor, salvo la diferencia de introducir dos señales de tensión por la patilla no inversora en vez de solo una.

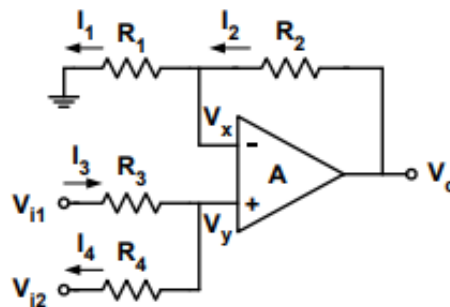


Ilustración 128. Circuito sumador no inversor

De este circuito obtenemos dos ecuaciones; una ecuación a partir de la igualdad de las corrientes I_1 e I_2 , pertenecientes a las ramas inversora y de realimentación respectivamente, y otra ecuación de la igualdad de corrientes I_3 e I_4 de la patilla no inversora.

- $I_1=I_2$

$$V_o = V_x \cdot \frac{R_1+R_2}{R_1} \quad (5)$$

- $I_3=I_4$

$$V_y = \frac{V_{i1}R_4+V_{i2}R_3}{R_3+R_4} \quad (6)$$

Ya que V_x es igual a V_y al haber realimentación negativa, sustituimos la ecuación (6) en la ecuación (5):

$$V_o = \frac{V_{i1}R_4+V_{i2}R_3}{R_3+R_4} \cdot \frac{R_1+R_2}{R_1} \quad (7)$$

Si consideramos que $R_1 \parallel R_2 = R_3 \parallel R_4$ en la ecuación (7), obtenemos la siguiente y definitiva ecuación:

$$V_o = V_{i1} \cdot \frac{R_2}{R_3} + V_{i2} \cdot \frac{R_2}{R_4} \quad (8)$$

7.1.1.4. Circuitos restadores

El objetivo de estos circuitos es conseguir a la salida del amplificador operacional la diferencia de dos señales de entrada. Estas señales se conectarán por separado en las patillas de entrada, una por la patilla inversora y otra por la patilla no inversora. A partir del circuito, obtendremos dos ecuaciones, una por medio de la patilla inversora y la rama de realimentación igualando I_1 e I_2 , y la otra igualando las corrientes I_3 e I_4 de la patilla no inversora.

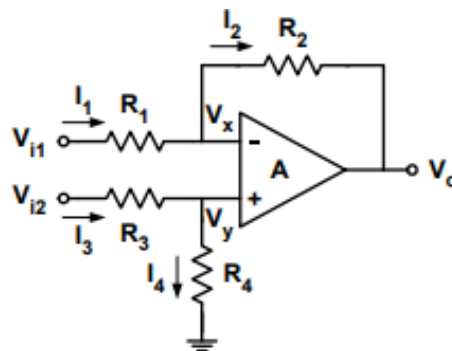


Ilustración 129. Circuito restador simple



- $I_1=I_2$

$$V_o = V_x \cdot \frac{R_1+R_2}{R_1} - V_{i1} \cdot \frac{R_2}{R_1} \quad (9)$$

- $I_3=I_4$

$$V_y = V_{i2} \cdot \frac{R_4}{R_3+R_4} \quad (10)$$

Si sustituimos la ecuación (10) en la (9), ya que V_y es igual a V_x , obtenemos la siguiente ecuación:

$$V_o = V_{i2} \cdot \frac{R_4}{R_3+R_4} \cdot \frac{R_1+R_2}{R_1} - V_{i1} \cdot \frac{R_2}{R_1} \quad (11)$$

Si consideramos que $R_1=R_3$ y $R_2=R_4$, obtenemos finalmente que:

$$V_o = (V_{i2} - V_{i1}) \cdot \frac{R_2}{R_1} \quad (12)$$

A parte del circuito restador simple, podemos desarrollar otro tipo de circuitos, que son los **restadores con alta impedancia de entrada**. El fundamento de estos circuitos es que las señales de tensión conectadas a la entrada se aplicarán únicamente en la patilla no inversora. Esto implicará que las corrientes que generamos a la entrada sean casi nulas, ya que no estarán realimentadas con la salida, de tal forma que tendremos unas impedancias elevadas de entrada. Con esto, conseguimos que las entradas y las salidas están aisladas entre ellas. Para conseguir el mismo efecto de los circuitos restadores simples, será necesario utilizar más de un amplificador operacional. A continuación, mostramos dos ejemplos, uno con tres amplificadores operacionales y otro con dos.

- CIRCUITO CON 3 AMPLIFICADORES OPERACIONALES

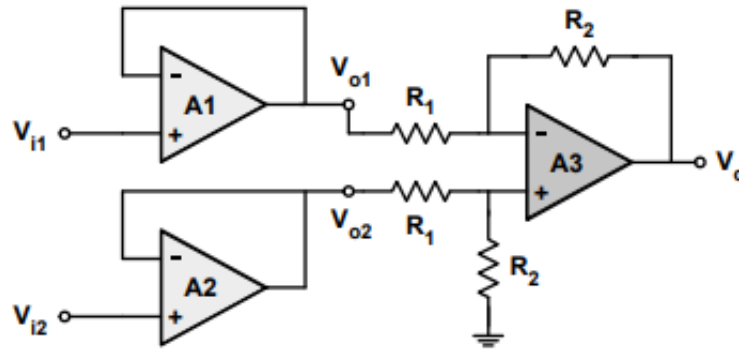


Ilustración 130. Restador alta impedancia de entrada con 3 A.O.

Los amplificadores A1 y A2 actúan como simples buffers, consiguiendo dos señales más limpias a la salida.

$$V_{i1} = V_{o1} ; V_{i2} = V_{o2} \quad (13)$$

Después de los dos primeros amplificadores operacionales, pasamos a un tercero, que actúa como un restador simple. La salida que se obtiene se define en la siguiente ecuación:

$$V_o = (V_{i2} - V_{i1}) \cdot \frac{R_2}{R_1} \quad (14)$$

Esta aplicación del amplificador operacional no estará en el programa debido a que conseguimos el mismo efecto usando dos de ellos, como vemos en el siguiente ejemplo.

- CIRCUITO CON 2 AMPLIFICADORES OPERACIONALES

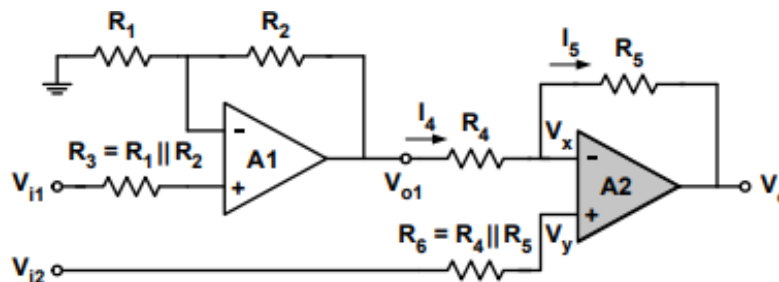


Ilustración 131. Restador alta impedancia de entrada con 2 A.O.

En este caso A1 actúa como un circuito no inversor, con el que conseguimos la señal que se introduce en la patilla inversora del restador simple A2.

$$V_{o1} = V_{i1} \cdot \frac{R_1 + R_2}{R_1} \quad (15)$$

$$V_o = V_{i2} \cdot \frac{R_4 + R_5}{R_4} - V_{o1} \cdot \frac{R_5}{R_4} \quad (16)$$

Si sustituimos (15) en (16) y consideramos que $\frac{R_1}{R_2} = \frac{R_5}{R_4}$ obtenemos que:

$$V_o = (V_{i2} - V_{i1}) \cdot \left(1 + \frac{R_5}{R_4}\right) \quad (17)$$

7.1.1.5. Circuitos integradores

Los circuitos integradores consiguen que obtengamos a la salida el valor integrado de la señal que hemos introducido por el amplificador operacional. En estos circuitos, se introducen a su vez las anteriores operaciones que hemos visto.

→ INTEGRADOR INVERSOR

Se introducen los conceptos del circuito inversor, pero con la diferencia de que introducimos un capacitor en la rama de realimentación, que hará el efecto de integrar la señal de tensión que introducimos por la entrada. El circuito resultante se muestra en la *ilustración 10*.

Como condición inicial tenemos que, al ser circuito inversor, $V_x = V_y = 0$.

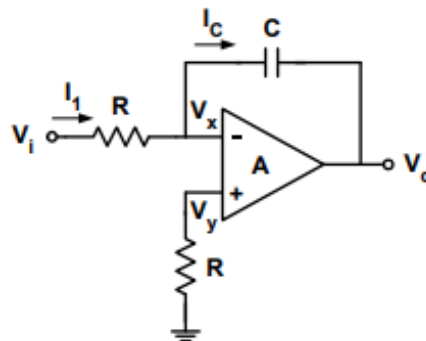


Ilustración 132. Circuito integrador inversor

La corriente que circula por el condensador del circuito es:

$$I_c = C \frac{d(V_x - V_o)}{dt} = -C \frac{dV_o}{dt} \quad (18)$$

Si igualamos las corrientes que circulan por la patilla inversora y por la realimentación, obtenemos la siguiente ecuación:

$$V_o = -\frac{1}{RC} \int V_i dt \quad (19)$$

Ya que existe la condición del tiempo en la integral, debemos de establecer unas condiciones iniciales e introducir un elemento en el circuito que haga que se produzca un cambio de dichas condiciones tras pasar un periodo de tiempo.

Para ello se emplean interruptores que, tras pasar un periodo de tiempo, se abren o se cierran y modifican el circuito, tal y como se muestra en la siguiente ilustración.

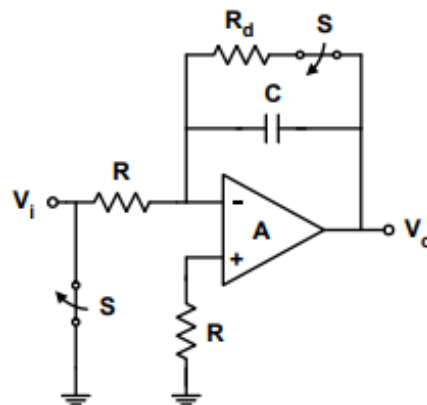


Ilustración 133. Circuito integrador inversor con interruptores

Analizamos el circuito de la *ilustración 11*. Previo a cierto instante de tiempo, por ejemplo, en $t=0$, los interruptores están cerrados. Esto hace que el valor de la tensión de entrada sea nulo y el condensador esté descargado, haciendo que el valor de la salida sea cero. Pero a partir de ese instante de tiempo, los interruptores se abren y se proporciona un valor a la salida. De esta forma, tenemos un valor a la salida para un instante de tiempo determinado, en este caso para $t=0$. A continuación, mostramos la representación V_o-t resultante.

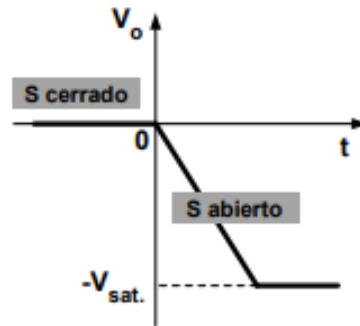


Ilustración 134. Representación V_o-t de integrador inversor

→ INTEGRADOR NO INVERSOR

Para este caso, el circuito cambia considerablemente en comparación con el circuito no inversor. Como podemos ver en la *ilustración 13*, el amplificador operacional se encuentra realimentando por ambas patillas, inversora y no inversora, y el condensador se coloca en la rama de la patilla no inversora que descarga en masa. En cuanto a la patilla inversora, ésta se mantiene conectada a tierra como en el caso del circuito no inversor.

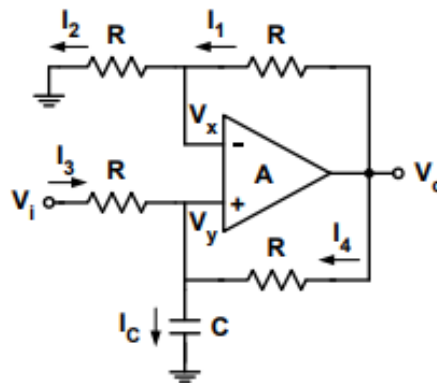


Ilustración 135. Circuito integrador no inversor

Como se ha explicado antes, la presencia del condensador hace que aparezca en el valor de la tensión de salida un producto integrador. Estableciendo la ley de nodos por ambas patillas y que $V_y = V_x$, obtenemos que:

$$V_o = \frac{2}{RC} \int V_i dt \quad (20)$$

→ INTEGRADOR SUMADOR

Partiendo del circuito sumador inversor, si colocamos un condensador en la rama de realimentación, obtenemos nuestro integrador sumador, mostrado en la siguiente ilustración:

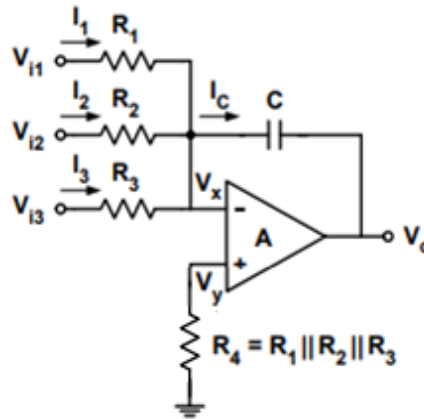


Ilustración 136. Circuito integrador sumador

Sabiendo que $V_x = V_y = 0$ y estableciendo que $I_1 + I_2 + I_3 = I_c$ (I_c está definida en la ecuación 18), podemos determinar el valor de V_o .

$$V_o = -\frac{1}{C} \int \left(\frac{V_{i1}}{R_1} + \frac{V_{i2}}{R_2} + \frac{V_{i3}}{R_3} \right) dt \quad (21)$$

Si consideramos que $R_1 = R_2 = R_3 = R$, tenemos que:

$$V_o = -\frac{1}{RC} \int (V_{i1} + V_{i2} + V_{i3}) dt \quad (22)$$

→ INTEGRADOR RESTADOR

Por último, llevamos a estudio los circuitos integrador restador. Al igual que el resto, partimos de un circuito previo, en este caso, del circuito restador simple, e introducimos elementos capacitivos en la rama de la patilla no inversora y en la rama de realimentación, tal y como podemos ver en la siguiente ilustración.

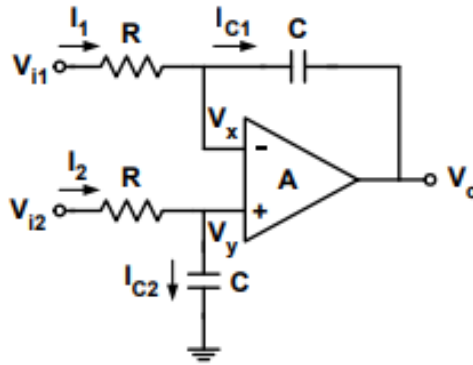


Ilustración 137. Circuito integrador restador

A partir de los dos nodos obtenemos una serie de ecuaciones, con las que obtendremos nuestra ecuación final:

$$V_o = \frac{1}{RC} \int (V_{i2} - V_{i1}) dt \quad (23)$$

También podemos obtener un circuito integrador restador mediante dos amplificadores operacionales, tal y como mostramos en la *ilustración 16*. Este ejemplo tampoco se encuentra en el programa realizado.

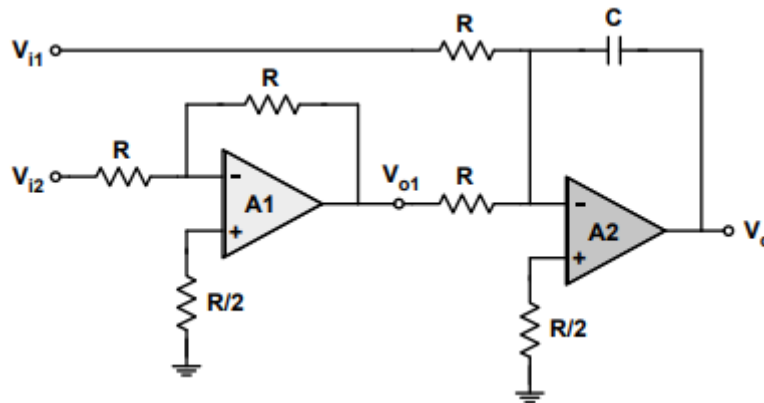


Ilustración 138. Integrador restador con 2 A.O.

El amplificador operacional A1 actúa como circuito inversor, del que conseguimos que:

$$V_{o1} = -V_{i2} \quad (24)$$

Por otro lado, A2 se emplea en un circuito integrador sumador. Como de la primera etapa obtenemos una señal negativa, al implementar un sumador

obtendremos como resultado final una diferencia de señales. La señal de salida final que se obtiene es la misma que con un circuito integrador restador con un amplificador operacional:

$$V_o = \frac{1}{RC} \int (V_{i2} - V_{i1}) dt \quad (25)$$

7.1.1.6. Circuito diferenciador

Lo que se busca con estos circuitos es obtener en la salida el valor de la entrada derivado. Para ello, se debe introducir una tensión de entrada que varíe en el tiempo, ya que, si se introduce una señal continua, el valor a la salida será cero.

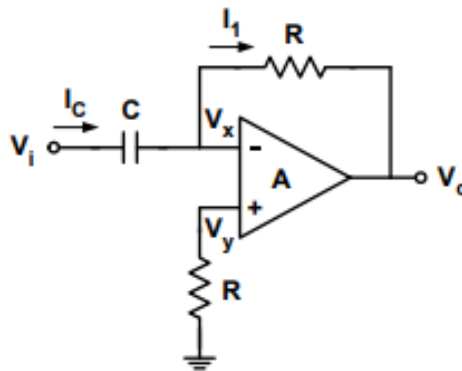


Ilustración 139. Circuito diferenciador

Como vemos en la ilustración, basándonos en un circuito inversor, introducimos un condensador en la entrada inversora. Sabiendo la corriente que circula por un condensador (definida en la ecuación 18), y que $V_x=V_y=0$, obtenemos la siguiente ecuación:

$$V_o = -RC \frac{dv_i}{dt} \quad (26)$$

7.1.1.7. Fuentes de tensión de referencia

Los amplificadores operaciones también pueden emplearse para proporcionar una tensión continua estable. Estas tensiones se pueden emplear posteriormente como referencia para otro tipo de circuitos. A continuación, se definen algunas formas de como proporcionar esta tensión de referencia.

→ FUENTE CON PILA PATRÓN

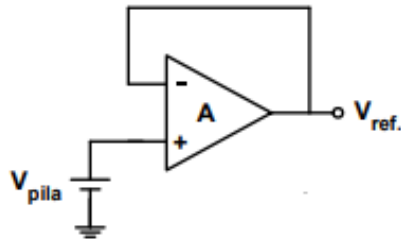


Ilustración 140. Fuente con pila patrón

$$V_y = V_x ; V_{ref} = V_x ; V_y = V_{pila} \rightarrow V_{ref} = V_{pila} \quad (27)$$

Esta aplicación del amplificador operacional no estará en el programa debido a la sencillez del circuito.

→ FUENTE CON DIODO ZENER

En este caso, disponemos de un diodo zener que se polariza inversamente, ya que el ánodo está conectado a masa y por la rama del cátodo se suministra una tensión +V. Esto hará que el diodo genere tensión zener V_z .

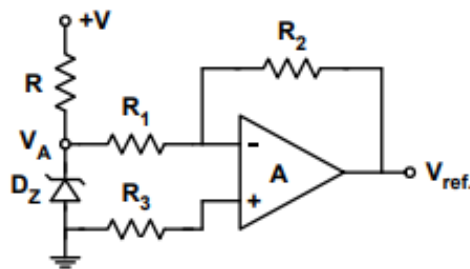


Ilustración 141. Fuente con diodo zener

$$V_{ref} = -V_A \cdot \frac{R_2}{R_1} ; V_A = V_z \rightarrow V_{ref} = -V_z \cdot \frac{R_2}{R_1} \quad (28)$$

→ FUENTE DE REFERENCIA DE DOBLE POLARIDAD

Con este montaje se necesitan dos amplificadores operacionales que actúen como inversores, tal y como se muestra en la *ilustración 20*. Se busca que podamos obtener la misma señal, tanto positiva como negativa, de nuestra tensión de referencia. Esta aplicación del amplificador operacional no estará en el programa realizado.

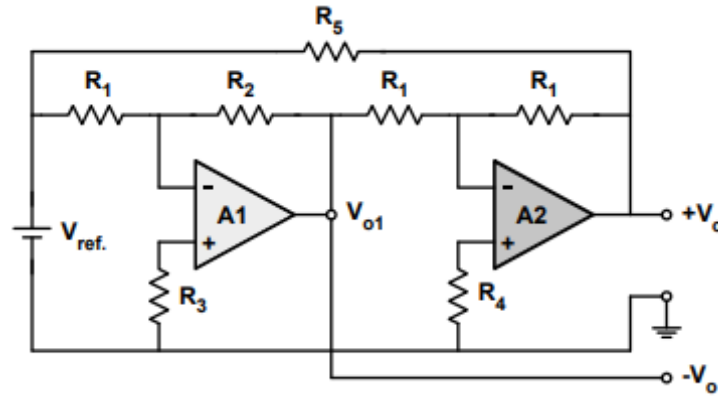


Ilustración 142. Fuente de referencia de doble polaridad

Mediante A1, obtenemos la señal negativa de V_{ref} :

$$-V_o = V_{o1} = -V_{ref} \cdot \frac{R_2}{R_1} \quad (29)$$

Por otro lado, con A2 conseguiremos la señal positiva de V_{ref} . Como la señal de salida que obtenemos de A1 se utiliza como señal de entrada para la patilla inversora de A2, arrastramos su valor negativo y su ganancia. Por ello, la ganancia de este inversor deberá ser (-1), que se consigue empleando resistencias tanto en la patilla inversora como en la rama de realimentación. Estas resistencias deben tener el mismo valor, en este caso R_1 .

$$+V_o = -V_{o1} = V_{ref} \cdot \frac{R_2}{R_1} \quad (30)$$

7.1.1.8. Fuentes de corriente

Las fuentes de corriente son circuitos que proporcionan una corriente constante, es decir, que su valor no depende del valor de la resistencia (R_L) por la que circule dicha corriente. Mediante amplificadores operacionales, podemos construir dichos circuitos, tal y como mostramos en los siguientes ejemplos.

- EJEMPLO 1

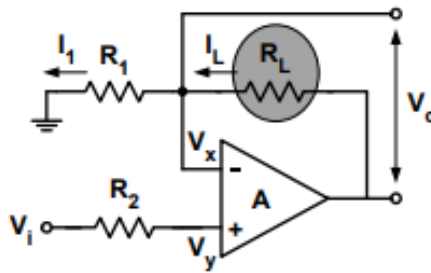


Ilustración 143. Ejemplo 1 fuente de corriente

$$V_y = V_x ; V_y = V_i ; I_L = I_1 ; I_1 = \frac{V_x}{R_1} \rightarrow I_L = \frac{V_i}{R_1} = cte \quad (31)$$

• EJEMPLO 2

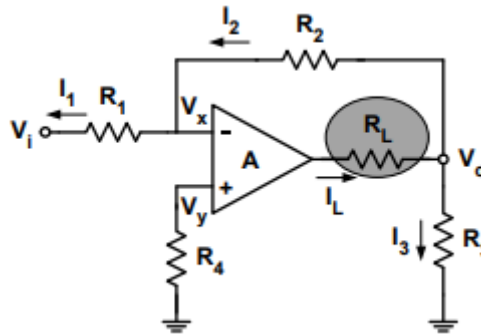


Ilustración 144. Ejemplo 2 fuente de corriente

$$I_2 = I_1 \rightarrow \frac{V_o}{R_2} = -\frac{V_i}{R_1} \rightarrow V_o = -\frac{R_2}{R_1} \cdot V_i \rightarrow I_2 = -\frac{V_i}{R_1} \quad (32)$$

$$I_3 = \frac{V_o}{R_3} ; V_o = -\frac{R_2}{R_1} \cdot V_i \rightarrow I_3 = -\frac{R_2}{R_1 \cdot R_3} \cdot V_i \quad (33)$$

$$I_L = I_2 + I_3 \quad (34)$$

Sustituyendo (32) y (33) en (34), obtenemos que:

$$I_L = \left(-\frac{V_i}{R_1}\right) + \left(-\frac{R_2}{R_1 \cdot R_3} \cdot V_i\right) = -V_i \cdot \frac{1}{R_1} \cdot \left(1 + \frac{R_2}{R_3}\right) = cte \quad (34)$$

7.1.1.9. Convertidores corriente-tensión y tensión-corriente

Para terminar con las aplicaciones lineales de los amplificadores operacionales, llevamos a estudio los convertidores tanto de corriente-tensión como de tensión-corriente. A continuación, entramos un poco en detalle con cada uno de ellos.

→ CONVERTIDORES CORRIENTE-TENSIÓN

Mediante este tipo de convertidores conseguimos una tensión de salida proporcional a la corriente que introducimos a la entrada ($V_o = K \cdot I_i$). La constante proporcional K podrá ser positiva o negativa, en función del montaje que realicemos.

- $K < 0$

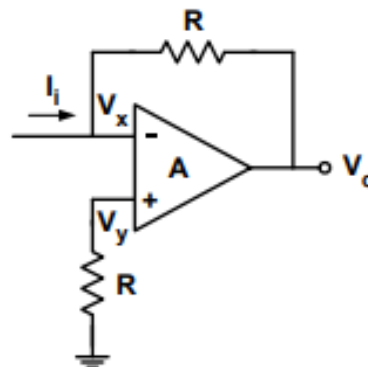


Ilustración 145. Convertidor corriente-tensión con $k < 0$

Sabiendo que $V_y = V_x = 0$, obtenemos la ecuación que relaciona la corriente de entrada I_i con la tensión de salida V_o mediante la realimentación:

$$I_i = \frac{V_x - V_o}{R} = -\frac{V_o}{R} \rightarrow V_o = -R \cdot I_i \quad (35)$$

Siendo $K = -R < 0$.

- $K > 0$

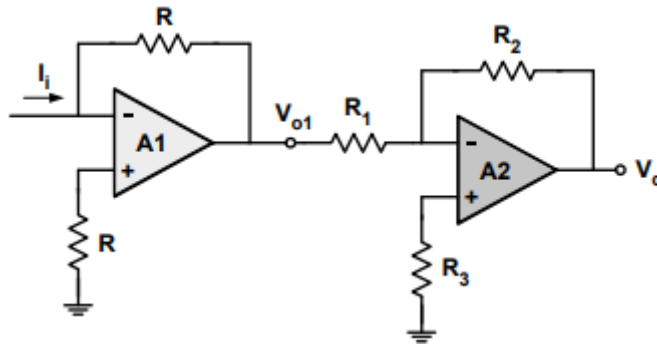


Ilustración 146. Convertidor corriente-tensión con $k > 0$

Para conseguir que nuestra constante proporcional sea positiva, el montaje debe contar con dos amplificadores. Partiendo del montaje realizado para $K < 0$, le añadimos a este un amplificador inversor y así conseguir una ganancia positiva.

$$A1 \rightarrow V_{o1} = -R \cdot I_i \quad (36)$$

$$A2 \rightarrow V_o = -V_{o1} \cdot \frac{R_2}{R_1} \quad (37)$$

Sustituyendo (36) en (37) obtenemos que:

$$V_o = R \cdot \frac{R_2}{R_1} \cdot I_i \quad (38)$$

Siendo $K = R \cdot \frac{R_2}{R_1} > 0$

→ CONVERTIDORES TENSIÓN-CORRIENTE

Por otro lado, tenemos los convertidores tensión-corriente, con los que conseguimos que la corriente a la salida del circuito sea proporcional a la tensión de la entrada ($I_o = K \cdot V_i$). En este caso, la constante proporcional K no dependerá del montaje para que sea positiva o negativa, si no que dependerá del valor de las señales que introduzcamos al circuito.

Partimos del circuito mostrado en la *ilustración 25*.

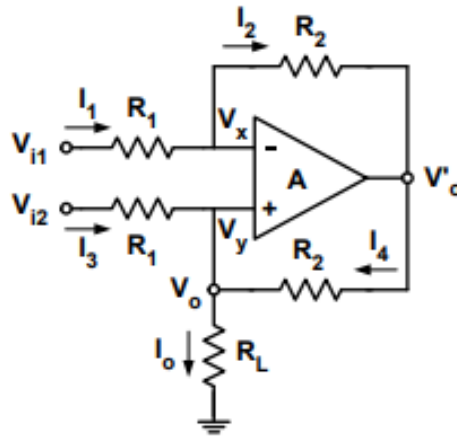


Ilustración 147. Convertidor tensión-corriente

Estableciendo que:

$$V_x = V_y = V_o ; I_1 = I_2 ; I_o = I_3 + I_4 \quad (39)$$

Podemos obtener la ecuación final del circuito:

$$I_o = \frac{V_{i2} - V_{i1}}{R_1} \quad (40)$$

La diferencia entre las señales de entrada nos definirá el signo de la constante proporcional K. Si queremos asegurarnos de que dicha constante K sea positiva o negativa, una de las dos tensiones de entrada tiene que ser nula.

- Si $V_{i1} = 0$

Dejamos de introducir tensión por la patilla inversora y la conectamos a tierra. El resultado que obtenemos es el siguiente:

$$I_o = \frac{V_{i2}}{R_1} \quad (41)$$

Siendo $K = \frac{1}{R_1} > 0$

- Si $V_{i2} = 0$

Si, por el contrario, dejamos de introducir tensión por la patilla no inversora y la conectamos a tierra, el resultado que obtenemos es el siguiente:

$$I_o = -\frac{V_{i1}}{R_1} \quad (42)$$

Siendo $K = -\frac{1}{R_1} > 0$

7.1.2. Aplicaciones no lineales de los amplificadores operacionales

7.1.2.1. Circuitos limitadores

Estos circuitos parten de los circuitos inversores simples vistos anteriormente, definidos de nuevo por la ecuación (43), pero con la cualidad de que nos permiten recortar a un determinado valor la tensión de salida del conjunto. Para ello, emplearemos diodos simples y diodos Zener, tal y como mostramos en los siguientes ejemplos.

➔ LIMITADOR CON DIODO NORMAL

Para conseguir limitar la tensión a la salida, emplearemos un diodo simple en paralelo con la rama de realimentación del amplificador operacional. En función de cómo coloquemos el diodo, limitaremos la tensión de salida cuando la tensión de entrada sea positiva o negativa.

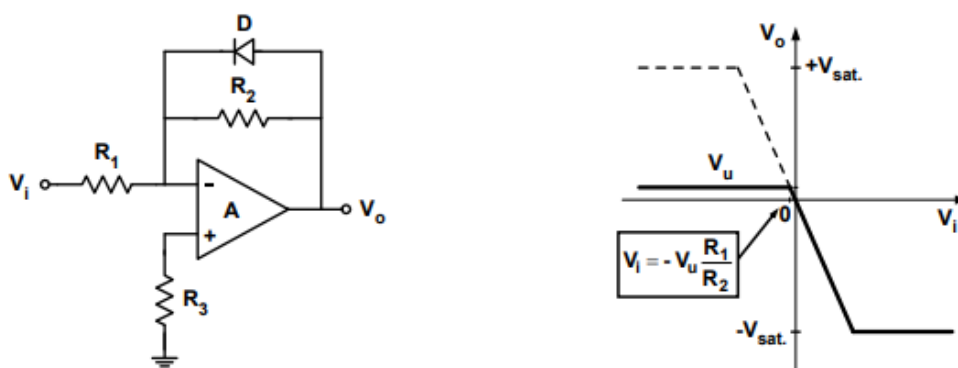


Ilustración 148. Circuito limitador con diodo normal I

En el caso de la *ilustración 26* tenemos que:

- Si $V_i > 0 \rightarrow V_o < 0 \rightarrow$ Diodo polarizado en inversa

$$V_o = -V_i \cdot \frac{R_2}{R_1} \quad (43)$$

- Si $V_i < 0 \rightarrow V_o > 0 \rightarrow$ Diodo polarizado en directa

- Si $V_o < V_u \rightarrow$ No conduce

$$V_o = -V_i \cdot \frac{R_2}{R_1} \quad (43)$$

- Si $V_o = V_u \rightarrow$ Conduce

$$V_o = -V_i \cdot \frac{R_2}{R_1} = V_u \rightarrow V_i = -V_u \cdot \frac{R_1}{R_2} \quad (44)$$

- Si $V_i < -V_u \cdot \frac{R_1}{R_2} \rightarrow$ Siempre conduce

$$V_o = V_u \quad (45)$$

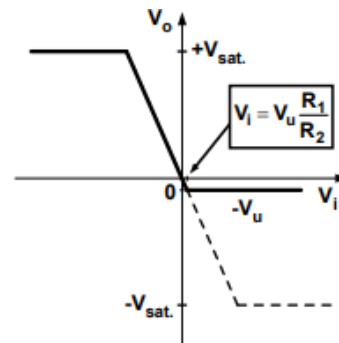
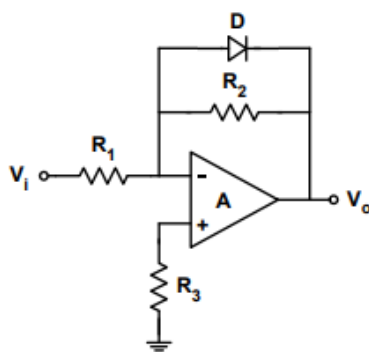


Ilustración 149. Circuito limitador con diodo normal II

Por otro lado, en la *ilustración 27* se establece que:

- Si $V_i < 0 \rightarrow V_o > 0 \rightarrow$ Diodo polarizado en inversa

$$V_o = -V_i \cdot \frac{R_2}{R_1} \quad (43)$$

- Si $V_i > 0 \rightarrow V_o < 0 \rightarrow$ Diodo polarizado en directa

- Si $V_o > -V_u \rightarrow$ No conduce

$$V_o = -V_i \cdot \frac{R_2}{R_1} \quad (43)$$

- Si $V_o = -V_u \rightarrow$ **Conduce**

$$V_o = -V_i \cdot \frac{R_2}{R_1} = -V_u \rightarrow V_i = V_u \cdot \frac{R_1}{R_2} \quad (46)$$

- Si $V_i > V_u \cdot \frac{R_1}{R_2} \rightarrow$ **Siempre conduce**

$$V_o = -V_u \quad (46)$$

→ LIMITADOR CON DIODO NORMAL Y DIODO ZENER EN SERIE

Otra manera de limitar la tensión de salida es introducir un diodo simple y un diodo zener en serie entre ellos y en paralelo con la rama realimentada del amplificador operacional. Como en el caso anterior, en función de la colocación de ambos diodos, la salida quedará limitada cuando la tensión de entrada sea positiva o negativa, tal y como se muestran en las dos siguientes ilustraciones.

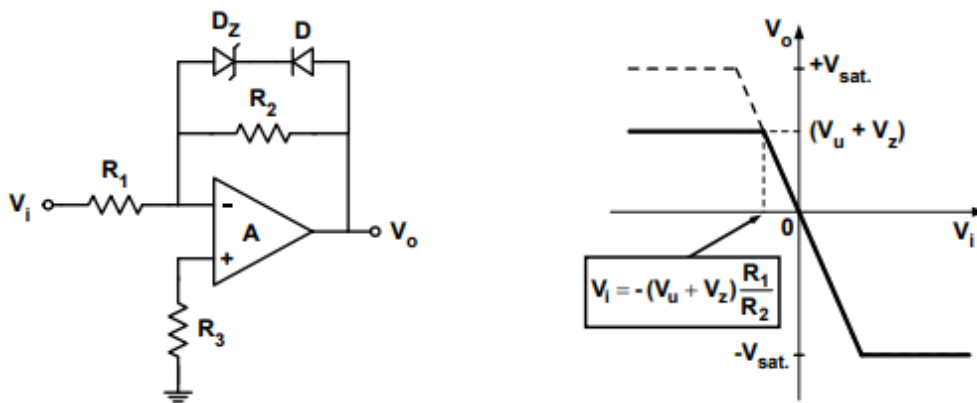


Ilustración 150. Circuito limitador con diodo normal y diodo zener I

Para la *ilustración 28* tenemos que:

- Si $V_i > 0 \rightarrow V_o < 0 \rightarrow$ **Diodo simple polarizado en inversa y diodo zener polarizado en directa \rightarrow No conducción**

$$V_o = -V_i \cdot \frac{R_2}{R_1} \quad (43)$$

- Si $V_i < 0 \rightarrow V_o > 0 \rightarrow$ Diodo simple polarizado en directa y diodo zener polarizado en inversa

- Si $V_o < (V_u + V_z) \rightarrow$ No conducción

$$V_o = -V_i \cdot \frac{R_2}{R_1} \quad (43)$$

- Si $V_o = (V_u + V_z) \rightarrow$ Conducción

$$V_o = -V_i \cdot \frac{R_2}{R_1} = (V_u + V_z) \rightarrow V_i = -(V_u + V_z) \cdot \frac{R_1}{R_2} \quad (48)$$

- Si $V_i < -(V_u + V_z) \cdot \frac{R_1}{R_2} \rightarrow$ Siempre conducen

$$V_o = (V_u + V_z) \quad (49)$$

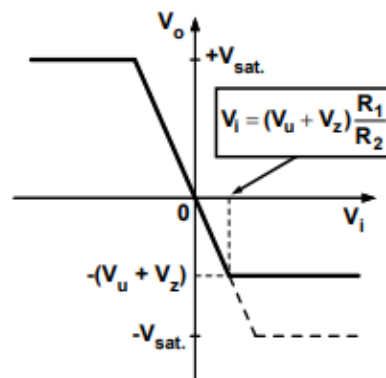
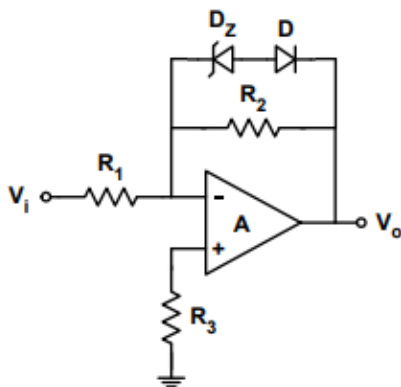


Ilustración 151. Circuito limitador con diodo normal y diodo zener II

Por otro lado, en la *ilustración 29* se establece que:

- Si $V_i < 0 \rightarrow V_o > 0 \rightarrow$ Diodo simple polarizado en inversa y diodo zener polarizado en directa \rightarrow No conducción

$$V_o = -V_i \cdot \frac{R_2}{R_1} \quad (43)$$

- Si $V_i > 0 \rightarrow V_o < 0 \rightarrow$ Diodo simple polarizado en directa y diodo zener polarizado en inversa

- Si $V_o > -(V_u + V_z) \rightarrow$ No conducción

$$V_o = -V_i \cdot \frac{R_2}{R_1} \quad (43)$$

- Si $V_o = -(V_u + V_z) \rightarrow$ **Conducción**

$$V_o = -V_i \cdot \frac{R_2}{R_1} = -(V_u + V_z) \rightarrow V_i = (V_u + V_z) \cdot \frac{R_1}{R_2} \quad (50)$$

- Si $V_i > (V_u + V_z) \cdot \frac{R_1}{R_2} \rightarrow$ **Siempre conducen**

$$V_o = -(V_u + V_z) \quad (51)$$

→ LIMITADOR CON DOS DIODOS ZENER

Por último, podemos conseguir limitar la tensión de salida empleando dos diodos zener en serie. Estos diodos deben estar colocados de tal forma que el cátodo de cada uno debe estar en frente del otro. Además, la tensión zener del diodo más próximo a la salida debe ser mayor que la del otro diodo zener. Como en los casos anteriores, los dos zener en serie se colocan en paralelo con la rama de realimentación del amplificador operacional.

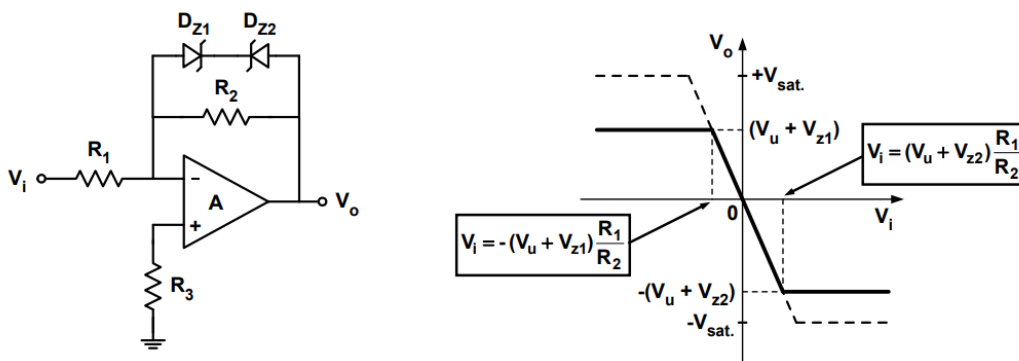


Ilustración 152. Circuito limitador con dos diodos zener

- Si $V_i > 0 \rightarrow V_o < 0 \rightarrow$ Diodo zener 1 polarizado en directa y diodo zener 2 polarizado en inversa
- Si $V_o > -(V_u + V_{z2}) \rightarrow D_{z1}$ polarizado en directa y D_{z2} no en zona zener \rightarrow **No conducción**

$$V_o = -V_i \cdot \frac{R_2}{R_1} \quad (43)$$

- Si $V_o = -(V_u + V_{z2}) \rightarrow$ **Dz1 polarizado en directa y Dz2 en zona zener**
 \rightarrow Conducción

$$V_o = -V_i \cdot \frac{R_2}{R_1} = -(V_u + V_{z2}) \rightarrow V_i = (V_u + V_{z2}) \cdot \frac{R_1}{R_2} \quad (52)$$

- Si $V_i > (V_u + V_{z2}) \cdot \frac{R_1}{R_2} \rightarrow$ **Dz1 polarizado en directa y Dz2 en zona zener**
 \rightarrow Siempre conducen

$$V_o = -(V_u + V_{z2}) \quad (53)$$

- Si $V_i < 0 \rightarrow V_o > 0 \rightarrow$ **Diodo zener 1 polarizado en inversa y diodo zener 2 polarizado en directa**

- Si $V_o < (V_u + V_{z1}) \rightarrow$ **Dz1 no en zona zener y Dz2 polarizado en directa**
 \rightarrow No conducción

$$V_o = -V_i \cdot \frac{R_2}{R_1} \quad (43)$$

- Si $V_o = (V_u + V_{z1}) \rightarrow$ **Dz1 en zona zener y Dz2 polarizado en directa** **\rightarrow Conducción**

$$V_o = -V_i \cdot \frac{R_2}{R_1} = (V_u + V_{z1}) \rightarrow V_i = -(V_u + V_{z1}) \cdot \frac{R_1}{R_2} \quad (54)$$

- Si $V_i < -(V_u + V_{z1}) \cdot \frac{R_1}{R_2} \rightarrow$ **Dz1 en zona zener y Dz2 polarizado en directa** **\rightarrow Siempre conducen**

$$V_o = (V_u + V_{z1}) \quad (55)$$

7.1.2.2. Detectores de tensión o comparadores

Como bien indica el nombre, estos circuitos sirven para determinar o detectar cuando alcanza un determinado valor, el que nosotros queramos, la señal que entra a nuestro circuito, que será la señal de entrada del propio amplificador operacional.

Para ello, dicha señal se compara con otra señal que nos sirve de referencia. Existen varios circuitos con los que podemos conseguir dicho efecto.

➔ DETECTOR O COMPARADOR SIMPLE

Mediante un amplificador operacional, introducimos dos señales de entrada. La señal que se va a llevar a estudio en la patilla inversora, y la de referencia en la no inversora. No se introduce rama de realimentación al circuito. Esto, sumado a que se tienen señales distintas en las patillas del amplificador operacional, hace que el funcionamiento del mismo se realice en lazo abierto, con una ganancia A muy elevada.

La tensión de salida queda definida de tal forma:

$$V_o = A \cdot (V_{ref} - V_i) \quad (56)$$

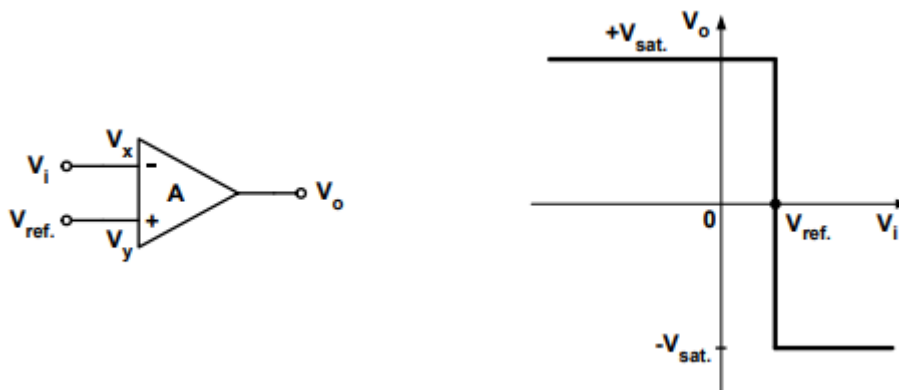


Ilustración 153. Comparador simple

Como podemos ver en la *ilustración 31*, tenemos que:

$$\text{Si } V_i < V_{ref} \rightarrow V_o = +V_{sat}$$

$$\text{Si } V_i > V_{ref} \rightarrow V_o = -V_{sat}$$

➔ COMPARADOR CON SALIDA SATURACIÓN-SATURACIÓN Y LAS DOS SEÑALES DE ENTRADA EN EL MISMO TERMINAL

Este detector se emplea cuando la señal de referencia que disponemos es distinta en valor a la tensión de entrada que queremos comparar. En este caso, estas dos señales están conectadas en la patilla inversora, mientras que la patilla no inversora está conectada a tierra. Seguimos sin tener realimentación en el amplificador operacional y trabajando en lazo abierto.

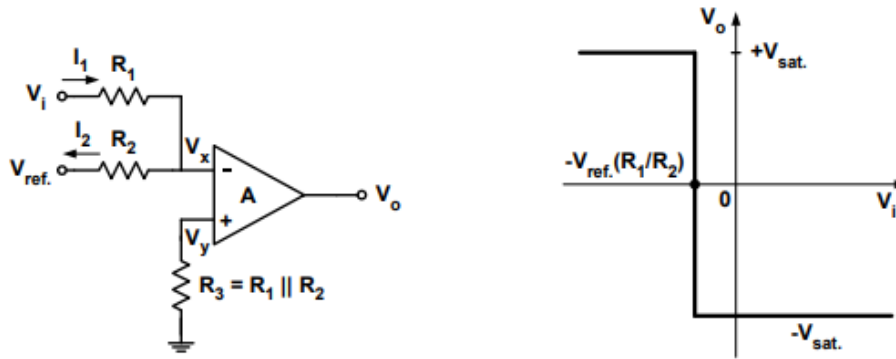


Ilustración 154. Comparador con salida saturación-saturación

Partiendo de la *ilustración 32*, establecemos una igualdad de corrientes ($I_1 = I_2$)

$$I_1 = \frac{V_i - V_x}{R_1} ; I_2 = \frac{V_x - V_{ref}}{R_2} \quad (57)$$

$$I_1 = I_2 \rightarrow \frac{V_i - V_x}{R_1} = \frac{V_x - V_{ref}}{R_2} \rightarrow V_x = \frac{V_i R_2 + V_{ref} R_1}{R_1 + R_2} \quad (58)$$

$$\text{Si } V_x < 0 \rightarrow V_i < -V_{ref} \cdot \frac{R_1}{R_2} \rightarrow V_o = +V_{sat}$$

$$\text{Si } V_x > 0 \rightarrow V_i > -V_{ref} \cdot \frac{R_1}{R_2} \rightarrow V_o = -V_{sat}$$

➔ COMPARADOR CON SALIDA CERO-SATURACIÓN

Este caso sigue la misma línea que los anteriores comparadores, pero con la diferencia de que se establece la rama de realimentación del amplificador operacional con un diodo simple. Con esto conseguimos limitar una de las salidas a la tensión umbral del diodo en cuestión. En función de cómo conectemos el diodo en la realimentación, conseguimos limitar una salida u otra.

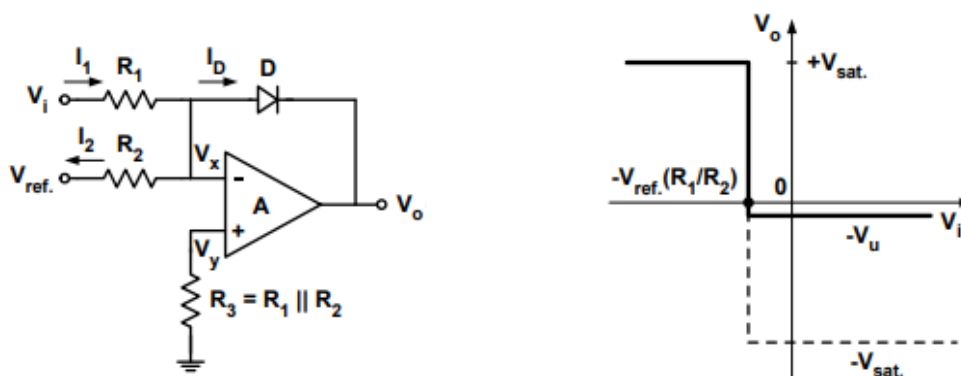


Ilustración 155. Comparador con salida cero-saturación I

Establecemos una igualación de corrientes ($I_D = I_1 - I_2$) en la *ilustración 33*.

$$I_1 = \frac{V_i - V_x}{R_1} ; I_2 = \frac{V_x - V_{ref}}{R_2} \quad (57)$$

- Si $I_D > 0 \rightarrow$ **Diodo conduce** \rightarrow Realimentación $\rightarrow V_x = V_y = 0$

$$I_D > \frac{V_i - V_x}{R_1} - \frac{V_x - V_{ref}}{R_2} \rightarrow I_D > \frac{V_i R_2 + V_{ref} R_1}{R_1 R_2} \rightarrow V_i > -V_{ref} \cdot \frac{R_1}{R_2} \quad (59)$$

$$V_i > -V_{ref} \cdot \frac{R_1}{R_2} \rightarrow V_o = -V_u \quad (60)$$

- Si $I_D < 0 \rightarrow$ **Diodo no conduce** \rightarrow No realimentación

$$I_D < 0 \rightarrow V_i < -V_{ref} \cdot \frac{R_1}{R_2} \rightarrow V_o = +V_{sat} \quad (61)$$

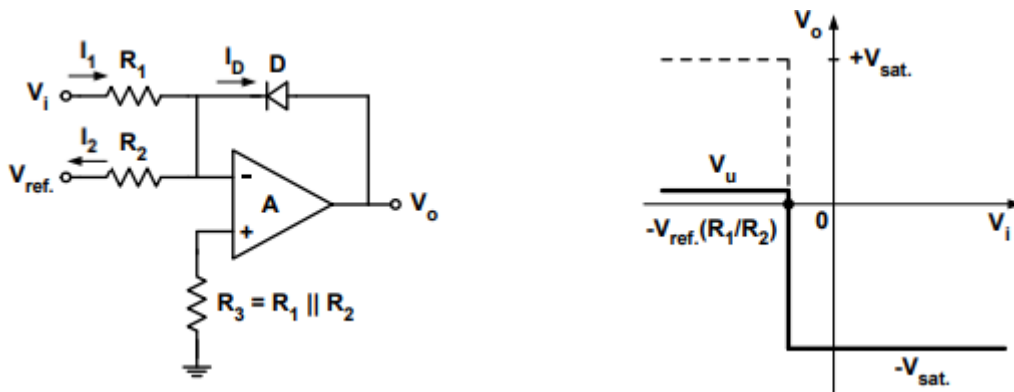


Ilustración 156. Comparador con salida cero-saturación II

Volvemos a establecer una igualación de corrientes ($I_D = I_1 - I_2$), en este caso para la *ilustración 34*.

$$I_1 = \frac{V_i - V_x}{R_1} ; I_2 = \frac{V_x - V_{ref}}{R_2} \quad (57)$$

- Si $I_D < 0 \rightarrow$ **Diodo conduce** \rightarrow Realimentación $\rightarrow V_x = V_y = 0$

$$I_D < \frac{V_i - V_x}{R_1} - \frac{V_x - V_{ref}}{R_2} \rightarrow I_D < \frac{V_i R_2 + V_{ref} R_1}{R_1 R_2} \rightarrow V_i < -V_{ref} \cdot \frac{R_1}{R_2} \quad (62)$$

$$V_i < -V_{ref} \cdot \frac{R_1}{R_2} \rightarrow V_o = V_u \quad (63)$$

- Si $I_D > 0 \rightarrow$ **Diodo no conduce** \rightarrow No realimentación

$$I_D > 0 \rightarrow V_i > -V_{ref} \cdot \frac{R_1}{R_2} \rightarrow V_o = -V_{sat} \quad (64)$$

\rightarrow DISIPADOR DE SCHMITT

Para terminar con los comparadores, se lleva a estudio el disipador de Schmitt. Mediante este circuito comparador, obtenemos una señal cuadrada a partir de una señal senoidal, tal como se muestra en la *ilustración 35*.

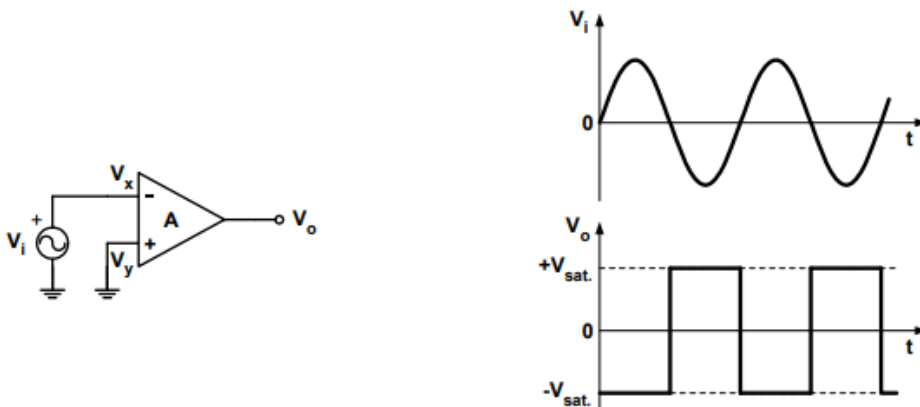


Ilustración 157. Disipador de Schmitt

Al no haber realimentación, trabajamos en lazo abierto, obteniendo así las siguientes ecuaciones:

$$V_x = V_i ; V_y = 0 \quad (65)$$

$$V_o = A \cdot (V_y - V_x) \rightarrow V_o = -A \cdot V_i \quad (66)$$

$$\text{Si } V_i < 0 \rightarrow V_o = +V_{sat}$$

$$\text{Si } V_i > 0 \rightarrow V_o = -V_{sat}$$

7.1.2.3. Comparadores con histéresis

Quando se trabaja con comparadores, pueden surgir algunos problemas que hacen que su funcionamiento no sea el correcto. Cuando la señal que queremos comparar alcanza el valor de la señal de referencia que disponemos, puede surgir un ruido en

nuestra señal de salida que haga que se distorsione durante un periodo de tiempo, tal y como se muestra en la siguiente ilustración.

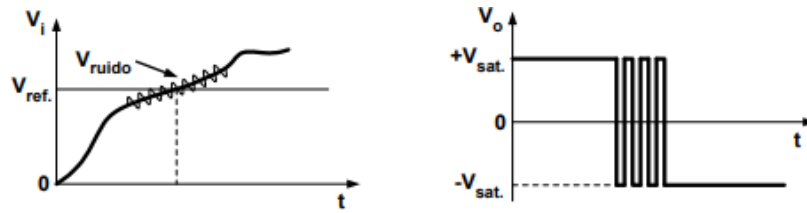


Ilustración 158. Señal de salida de un comparador simple distorsionada

Para solucionar este ruido, establecemos en los comparadores lo que se conoce como *histéresis*. Mediante este método, conseguimos transformar el tramo distorsionado en una conmutación más directa del valor positivo al negativo. Analizamos para que valores de señal de entrada a comparar se distorsiona la señal (V_1 y V_2). La diferencia entre estos valores es lo que denominamos *banda de histéresis*. Esto se muestra en la *ilustración 36*.

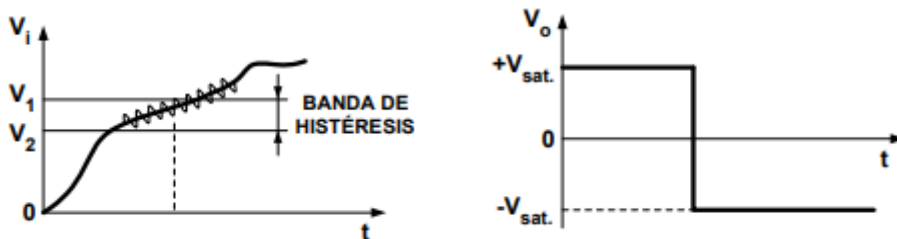


Ilustración 159. Señal de salida de un comparador simple con histéresis

$$\text{Si } V_i > V_1 \rightarrow V_o = -V_{sat}$$

$$\text{Si } V_i < V_2 \rightarrow V_o = +V_{sat}$$

→ COMPARADOR CON HISTERESIS

Partiendo del comparador simple que se ha visto anteriormente, realizamos el circuito con la diferencia de que se introduce una rama de realimentación con la patilla no inversora.

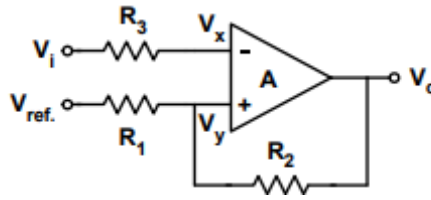


Ilustración 160. Comparador con histéresis

La tensión de salida queda definida de tal manera:

$$V_o = A \cdot (V_y - V_i) \quad (67)$$

$$\text{Si } V_i > V_y \Rightarrow V_o = -V_{sat}$$

$$\text{Si } V_i < V_y \Rightarrow V_o = +V_{sat}$$

A continuación, determinamos el valor de V_y estableciendo una igualación de corrientes:

$$\frac{V_{ref} - V_y}{R_1} = \frac{V_y - V_o}{R_2} ; V_y = \frac{V_{ref}R_2 + V_oR_1}{R_1 + R_2} \quad (68)$$

Para determinar para que valor de V_i conmuta la salida de saturación positiva a negativa y viceversa, debemos suponer primero el valor de la salida V_o .

- $V_o = +V_{sat}$

Tendremos que:

$$V_y = \frac{V_{ref}R_2 + (+V_{sat})R_1}{R_1 + R_2} = A \quad (69)$$

El valor de la salida V_o será el valor de la saturación positiva $+V_{sat}$ hasta que la tensión de entrada V_i alcance el valor de A, yendo dicha señal de entrada desde valores negativos hacia valores positivos. En dicho momento, V_o pasará a valer $-V_{sat}$.

$$\text{Si } V_i > V_y \Rightarrow V_i > A \Rightarrow V_o = -V_{sat}$$

- $V_o = -V_{sat}$

Para este caso tenemos que:

$$V_y = \frac{V_{ref}R_2 + (-V_{sat})R_1}{R_1 + R_2} = B \quad (70)$$

Ahora tendremos el efecto contrario. El valor de la señal de entrada V_i irá decreciendo hasta que alcance el nuevo valor de V_y , B. En este momento, el valor de V_o pasará de ser $-V_{sat}$ a $+V_{sat}$.

$$\text{Si } V_i < V_y \rightarrow V_i < B \rightarrow V_o = +V_{sat}$$

Este funcionamiento queda reflejado en la *ilustración 38*.

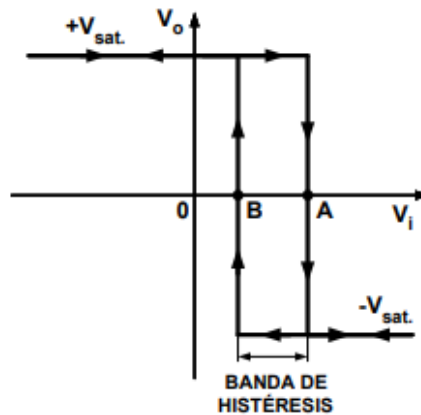


Ilustración 161. Representación V_o - V_i de un comparador con histéresis

→ COMPARADOR CON HISTÉRESIS Y LIMITACIÓN EN LA SALIDA

Este circuito es exactamente el mismo que el anterior, pero introduciendo un diodo simple conectado entre masa y el punto de la tensión de salida.

El funcionamiento del circuito sigue el mismo ciclo explicado anteriormente, pero con la diferencia de que la salida, en vez de saturar a la tensión de saturación del amplificador operacional, lo hará para el valor de la tensión umbral del diodo V_u . En función de la colocación del diodo, la salida quedará limitada cuando alcance el valor positivo de saturación del amplificador operacional, o cuando alcance el valor negativo.

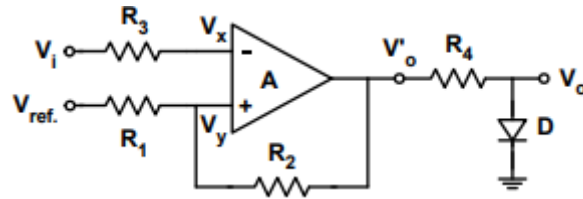


Ilustración 162. Comparador con histéresis y limitación de salida I

Lo que era en el anterior ejemplo V_o ahora lo es V'_o . Basándonos en el funcionamiento comentado anteriormente y sumándole el efecto del diodo, tendremos que:

Si $V'_o = +V_{sat}$ \Rightarrow Diodo conduce $\Rightarrow V_o = V_u$

Si $V'_o = -V_{sat}$ \Rightarrow Diodo no conduce $\Rightarrow V_o = -V_{sat}$

Esto queda representado en la siguiente ilustración:

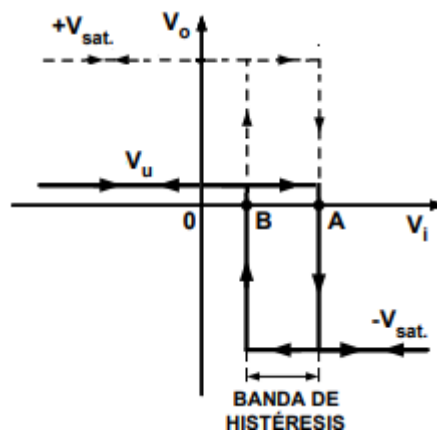


Ilustración 163. Representación V_o - V_i de un comparador con histéresis y limitación de salida I

Analizamos ahora la *ilustración 41*.

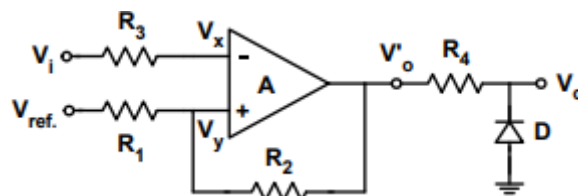


Ilustración 164. Comparador con histéresis y limitación de salida II

Para este caso, tendremos que:

Si $V'_o = +V_{sat} \Rightarrow$ Diodo conduce $\Rightarrow V_o = V_u$

Si $V'_o = -V_{sat} \Rightarrow$ Diodo no conduce $\Rightarrow V_o = -V_{sat}$

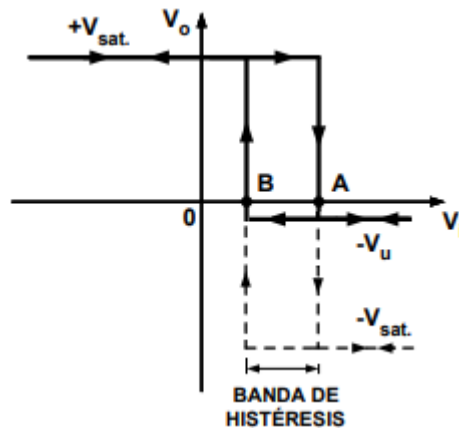


Ilustración 165. Representación V_o - V_i de un comparador con histéresis y limitación de salida II

7.1.2.4. Generador de onda cuadrada

También llamado *multivibrador a estable*, este circuito parte del circuito comparador con histéresis, pero sin introducir una señal de referencia a la entrada del amplificador. A su vez, la tensión de entrada a comparar V_i se sustituye introduciendo un condensador en la patilla inversora del amplificador operacional, que generará una tensión V_c . La carga y descarga del condensador se consigue introduciendo una resistencia R mediante realimentación negativa.

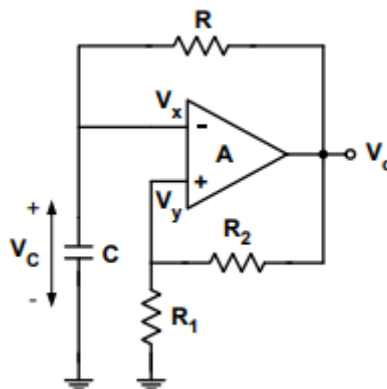


Ilustración 166. Circuito generador de onda cuadrada

Definimos la tensión de salida:

$$V_o = A \cdot (V_y - V_C) \quad (71)$$

Además, tenemos que:

$$V_y = V_o \cdot \frac{R_1}{R_1 + R_2} \quad (72)$$

Para saber el valor de nuestra tensión de entrada (V_C), debemos suponer un valor de V_o primero.

- $V_o = +V_{sat}$

$$V_y = +V_{sat} \cdot \frac{R_1}{R_1 + R_2} \rightarrow V_y = +\beta V_{sat} \quad (73)$$

Con este valor de V_o , el condensador C se irá cargando de forma creciente hacia $+V_{sat}$.

$$V_C \rightarrow +V_{sat} \quad (74)$$

La ecuación que define la curva de crecimiento de V_C es la siguiente:

$$V_C = +V_{sat} + (-V_{sat}) \cdot (\beta + 1) \cdot e^{-\frac{t}{RC}} \quad (75)$$

$$\text{Si } V_C > V_y \rightarrow V_C > +\beta V_{sat} \rightarrow V_o = -V_{sat}$$

- $V_o = -V_{sat}$

$$V_y = -V_{sat} \cdot \frac{R_1}{R_1 + R_2} \rightarrow V_y = -\beta V_{sat} \quad (76)$$

Con este valor de V_o , el condensador C se irá descargando de forma decreciente hacia $-V_{sat}$.

$$V_C \rightarrow -V_{sat} \quad (77)$$

La ecuación que define la curva de decrecimiento de V_C es la siguiente:

$$V_C = -V_{sat} + (+V_{sat}) \cdot (\beta + 1) \cdot e^{-\frac{t}{RC}} \quad (78)$$

$$\text{Si } V_C < V_y \Rightarrow V_C < +\beta V_{sat} \Rightarrow V_o = +V_{sat}$$

Este ciclo representa el funcionamiento del circuito para genera una onda cuadrada como la que se muestra en la siguiente ilustración.

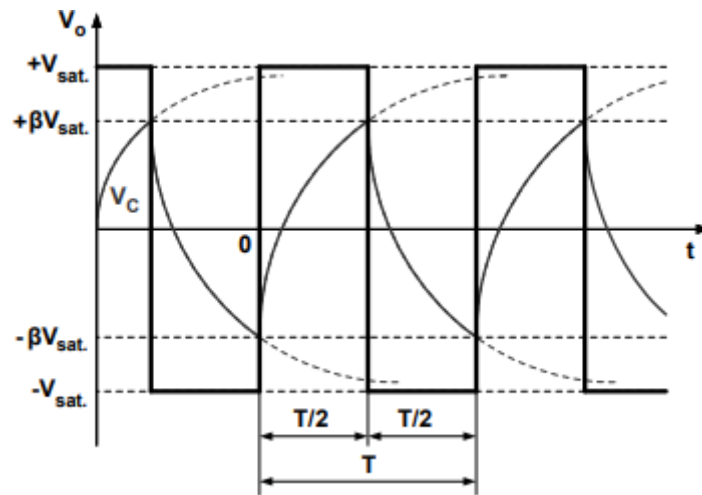


Ilustración 167. Representación Vo-t del circuito generador de onda cuadrada

El periodo de dicha onda cuadrada es el siguiente:

$$T = 2 \cdot R \cdot C \cdot \ln \left(\frac{1+\beta}{1-\beta} \right) \quad (79)$$

Siendo $\beta = \frac{R_1}{R_1+R_2}$

Estos circuitos pueden usarse en un intervalo de frecuencia de 10 Hz a 10 kHz, ya que, si superamos este intervalo, la forma de onda cuadrada quedaría muy distorsionada debido al *slew-rate*.

7.1.2.5. Generador de onda triangular

El circuito generador de onda triangular está formado por dos bloques de circuitos. Dispone de un circuito comparador con histéresis (A1) que genera una onda

cuadrada continua en el tiempo. En serie con este bloque se conecta un integrador inversor (A2), cuya salida se realimenta al comparador mediante su patilla no inversora. El resultado es el circuito de la *ilustración 44*.

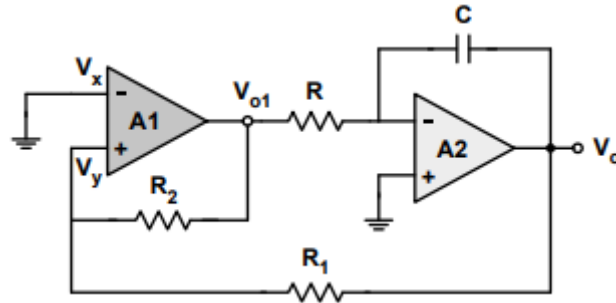


Ilustración 168. Circuito generador de onda triangular

Del bloque del comparador con histéresis tenemos que:

$$V_o = A \cdot V_y \quad (80)$$

Siendo V_y :

$$V_y = \frac{V_o R_2 + V_{o1} R_1}{R_1 + R_2} \quad (81)$$

Si $V_y > 0 \Rightarrow V_{o1} = +V_{sat}$

Si $V_y < 0 \Rightarrow V_{o1} = -V_{sat}$

Debido a la realimentación entre ambos bloques, debemos suponer un valor de V_{o1} para definir el funcionamiento del bloque integrador inversor.

- $V_{o1} = +V_{sat}$

$$V_y = \frac{V_o R_2 + (+V_{sat}) R_1}{R_1 + R_2} \quad (82)$$

$$V_o = \frac{-V_{sat} \cdot t}{RC} + (+V_{sat}) \cdot \frac{R_1}{R_2} \quad (83)$$

Esto implica que se integra hacia negativo.

Si $V_y < 0 \Rightarrow V_o < -V_{sat} \cdot \frac{R_1}{R_2} \Rightarrow V_{o1} = -V_{sat}$

- $V_{o1} = -V_{sat}$

$$V_y = \frac{V_o R_2 + (-V_{sat}) R_1}{R_1 + R_2} \quad (84)$$

$$V_o = \frac{+V_{sat} \cdot t}{RC} + (-V_{sat}) \cdot \frac{R_1}{R_2} \quad (85)$$

Esto implica que se integra hacia positivo.

$$\text{Si } V_y > 0 \Rightarrow V_o > +V_{sat} \cdot \frac{R_1}{R_2} \Rightarrow V_{o1} = +V_{sat}$$

Este desarrollo a lo largo del tiempo determina el funcionamiento del circuito con el que conseguimos una onda triangular como la mostrada en la siguiente ilustración.

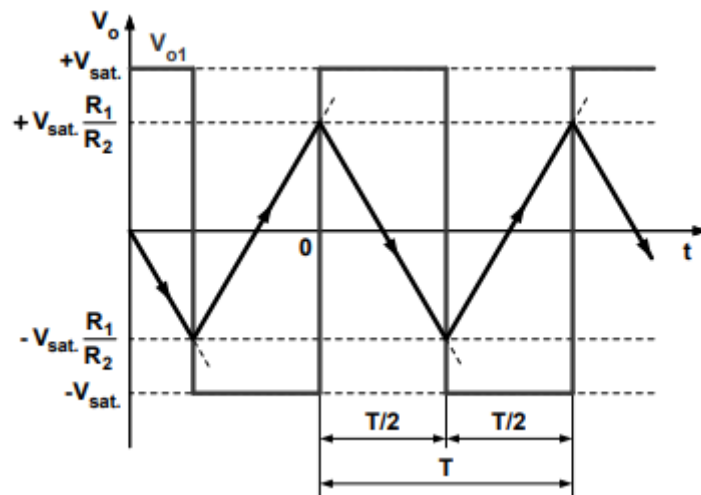


Ilustración 169. Representación V_o - t del circuito generador de onda triangular

El periodo de esta onda es el siguiente:

$$T = 4 \cdot R \cdot C \cdot \frac{R_1}{R_2} \quad (86)$$

7.1.2.6. Rectificadores de precisión

Los rectificadores de precisión son circuitos formados por amplificadores operacionales y diodos, de tal forma que, colocando los diodos en unas ramas

determinadas del circuito, se consigue rectificar la salida. La señal de entrada que se introduce al circuito es de pequeña amplitud. El objetivo final del circuito no es convertir energía a corriente continua, ya que la entrada es alterna, si no obtener información de la señal que introducimos al circuito.

➔ RECTIFICADOR DE MEDIA ONDA

La tensión introducida al circuito (V_i) es una señal senoidal de amplitud V_p .

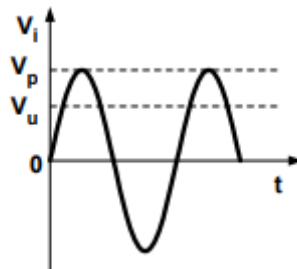


Ilustración 170. Señal V_i del rectificador de media onda

El montaje es el siguiente:

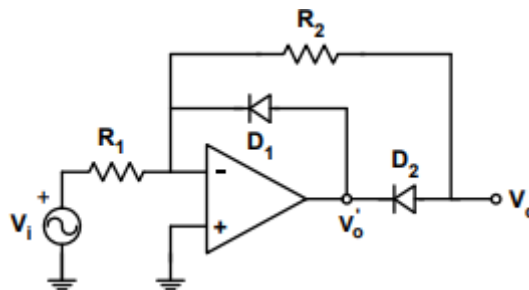


Ilustración 171. Rectificador de media onda

En función del signo de la señal de entrada, los diodos trabajarán de un modo u otro.

Si $V_i > 0$ ➔ D1 en OFF y D2 en ON ➔ $V'_o = V_o - V_u$

$$V_o = -V_i \cdot \frac{R_2}{R_1} \quad (87)$$

Si $V_i < 0$ ➔ D1 en ON y D2 en OFF ➔ $V'_o = V_u$

$$V_o = V_x = V_y = 0 \quad (88)$$

La onda resultante es la que se muestra en la *ilustración 47*, donde podemos ver como se rectifica media onda y la otra media se deja en valor nulo.

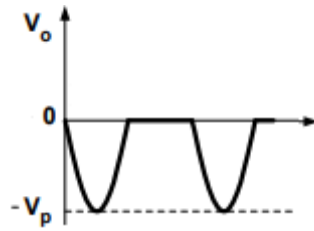


Ilustración 172. Representación Vo-t del rectificador de media onda

➔ RECTIFICADOR DE ONDA COMPLETA

Partimos de la misma señal de entrada que para el de media onda, reflejada en la *ilustración 48*. Para este circuito, se necesitarán dos amplificadores operacionales, ya que este montaje se divide en dos bloques. El primer bloque se trata de un rectificador de media onda, con la diferencia de que, en lo que en el anterior ejemplo era V_o , ahora es V_{o1} . El final del circuito es un sumador inversor con el que conseguimos rectificar la media onda que no se consigue rectificar con el anterior bloque, y además se rectifica para valores positivos, de ahí que se disponga de un sumador inversor y no de uno no inversor. En resumen, el montaje completo es el que se muestra en la siguiente ilustración:

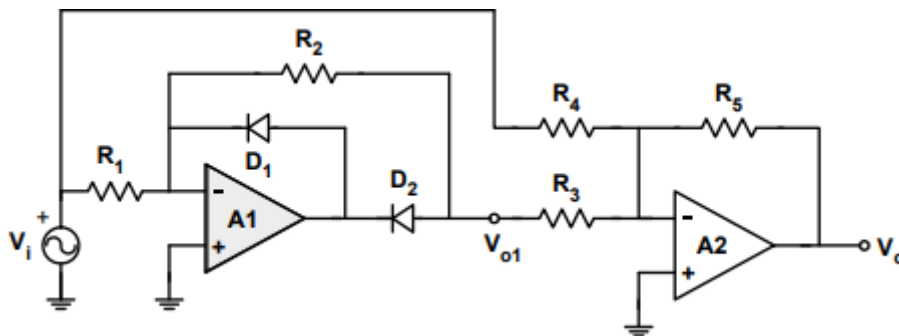


Ilustración 173. Rectificador de onda completa I

Como se ha explicado anteriormente, con un rectificador de media onda tenemos que:

Si $V_i > 0$ ➔ D1 en OFF y D2 en ON ➔ $V_{o1} = -V_i \cdot \frac{R_2}{R_1}$

Si $V_i < 0$ ➔ D1 en ON y D2 en OFF ➔ $V_{o1} = V_{x1} = V_{y1} = 0$

Por otro lado, del sumador inversor podemos definir el valor de salida.

$$V_o = - \left(V_{o1} \cdot \frac{R_5}{R_3} + V_i \cdot \frac{R_5}{R_4} \right) \quad (89)$$

Una vez establecido el funcionamiento de los dos bloques, podemos definir que:

$$\text{Si } V_i > 0 \rightarrow V_{o1} = -V_i \cdot \frac{R_2}{R_1} \rightarrow V_o = V_i \cdot \frac{R_2 R_5}{R_1 R_3} - V_i \cdot \frac{R_5}{R_4}$$

$$\text{Si } V_i < 0 \rightarrow V_{o1} = 0 \rightarrow V_o = -V_i \cdot \frac{R_5}{R_4}$$

Como podemos ver, el valor de V_o es distinto en las dos medias onda de un ciclo. Para conseguir una amplitud igual se debe cumplir que:

$$\frac{R_2 R_5}{R_1 R_3} = \frac{2R_5}{R_4} \quad \text{ó} \quad R_3 = \frac{R_2 R_4}{2R_1}$$

Con esta condición, el valor de V_o sería el siguiente:

$$V_o = \left| V_i \cdot \frac{R_5}{R_4} \right| \quad (90)$$

Siendo la forma de onda la siguiente:

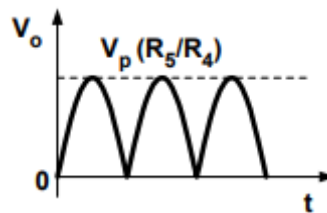


Ilustración 174. Representación V_o - t del rectificador de onda completa I

Una solución más rápida es que $R_1 = R_2 = R_4 = R$, con lo que que $R_3 = R/2$, consiguiendo así que la ganancia sea R_5/R .

Debido a este inconveniente de tener que conseguir varias resistencias iguales, como alternativa a este circuito se lleva a análisis otro rectificador de onda completa, el cual se muestra en la *ilustración 50*.

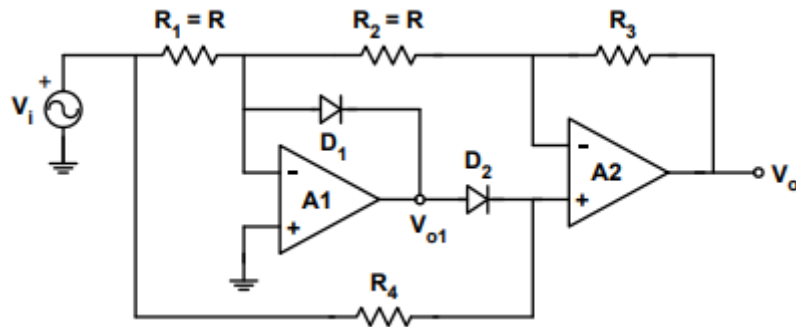


Ilustración 175. Rectificador de onda completa II

Para este montaje tendríamos que:

$$\text{Si } V_i > 0 \rightarrow V_o = V_i \cdot \frac{R_2 + R_3}{R_2}$$

$$\text{Si } V_i < 0 \rightarrow V_o = -V_i \cdot \frac{R_2 + R_3}{R_1}$$

Como ya viene indicado en el circuito, si establecemos que $R_1 = R_2 = R$, tenemos como salida, en valor absoluto:

$$V_o = \left| V_i \cdot \frac{R + R_3}{R} \right| = \left| V_i \cdot \left(1 + \frac{R_3}{R} \right) \right| \quad (91)$$

La onda resultante es la mostrada en esta ilustración.

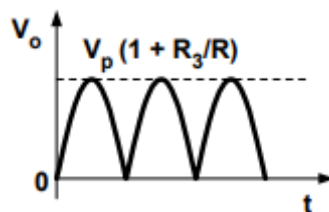


Ilustración 176. Representación V_o - t del rectificador de onda completa II

Como podemos ver, esta segunda solución de un rectificador de media onda plantea menos inconvenientes que la primera.

→ CIRCUITOS DE VALOR MEDIO Y VALOR EFICAZ

Este circuito está basado en un rectificador de onda completa, como el de la *ilustración 48*, pero incluyendo un condensador a modo de filtro paso bajo y así obtener el valor medio de la tensión de salida. El circuito es el siguiente:

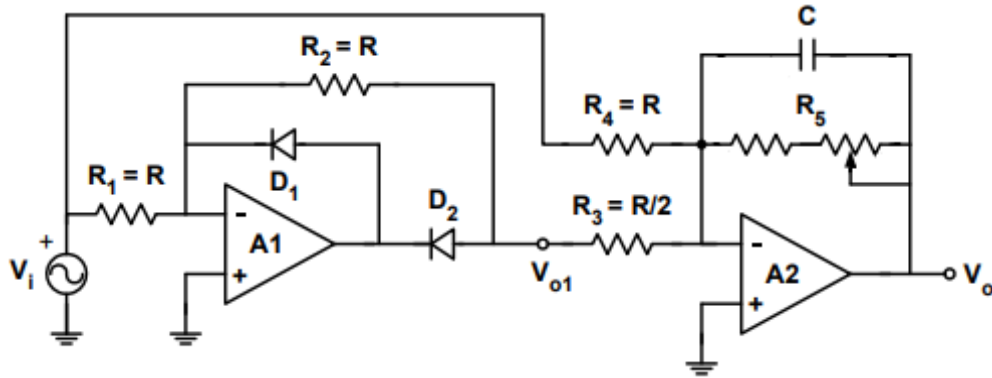


Ilustración 177. Circuito del valor medio y valor eficaz

La forma de onda tanto de la señal de entrada como de la salida resultante se muestra en la siguiente ilustración:

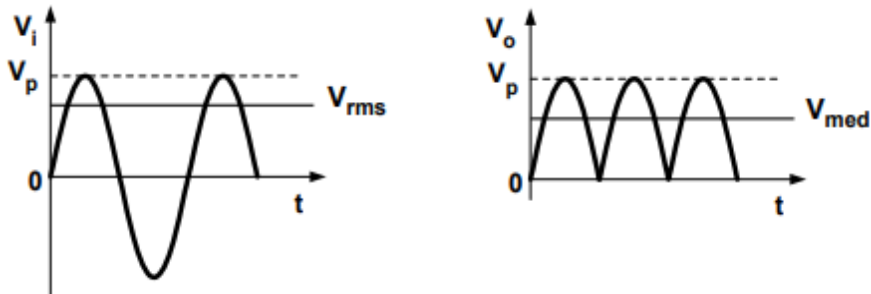


Ilustración 178. Representación V_i - t V_o - t del circuito del valor medio y valor eficaz

El valor eficaz de una señal senoidal se define con la siguiente ecuación:

$$V_{rms} = \frac{V_p}{\sqrt{2}} = 0,707 \cdot V_p \quad (92)$$

El valor de pico de la señal de entrada y de la de salida es el mismo. Por otro lado, podemos saber el valor medio de la señal de salida mediante esta ecuación:

$$V_{med} = \frac{2V_p}{\pi} = 0,637 \cdot V_p \quad (93)$$

El valor eficaz de la tensión de salida lo obtenemos despejando el valor de pico de (95) y sustituyéndolo en (94):

$$V_{rms} = \frac{V_p}{\sqrt{2}} = 0,707 \cdot V_p = 0,707 \cdot \frac{V_{med}}{0,637} = 1,11 \cdot V_{med} \quad (94)$$

La ganancia que conseguimos con el conjunto del circuito es:

$$G = \frac{R_5}{R} \quad (95)$$

Mediante esta ganancia, ajustable mediante un potenciómetro como se muestra en la *ilustración 52*, podemos conseguir a la salida tanto V_{rms} como V_{med} :

$$\text{Si } G = 1,11 \rightarrow V_o = V_{rms}$$

$$\text{Si } G = 1 \rightarrow V_o = V_{med}$$

Por último, para el cálculo del condensador debemos de tener en cuenta la frecuencia de corte del filtro:

$$f_o = \frac{1}{2\pi R_5 C} \quad (96)$$

Para un bien funcionamiento, esta frecuencia debe ser mucho más pequeña que la frecuencia mínima de operación del circuito. Si sumamos que el rectificador duplica la frecuencia a la que se trabaja, tenemos que:

$$C \gg \frac{1}{4\pi R_5 f_{min}} \quad (97)$$

Una aproximación que nos puede valer para obtener el valor exacto del condensador es:

$$C = \frac{1}{\text{max. rizado}} \cdot \frac{1}{4\pi R_5 f_{min}} \quad (98)$$

7.1.2.7. Detectores de pico

La última aplicación no lineal de los amplificadores operacionales es para detectar el valor máximo que alcanza una señal de entrada. Los circuitos detectores de pico están formados por dos amplificadores operacionales.

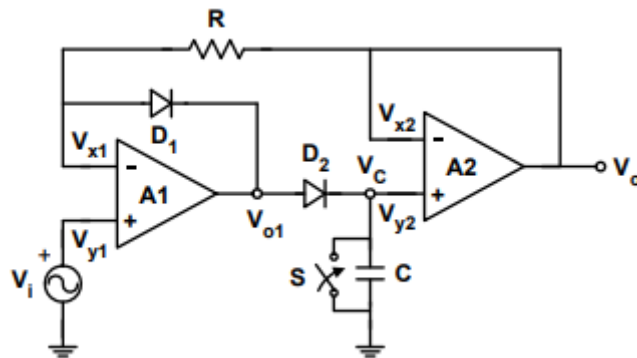


Ilustración 179. Circuito detector de pico

Con el primer amplificador operacional, conseguiremos cargar de tensión al condensador en función del signo de la tensión de entrada y del estado de los diodos.

El segundo amplificador operacional servirá como circuito separador (buffer), de tal forma que $V_o = V_c$. La alta impedancia en las patillas del amplificador hará que el condensador nunca se descargue.

El funcionamiento es el siguiente:

Con $V_i > 0 \Rightarrow V_{o1} > 0 \Rightarrow D_1$ en OFF y D_2 en ON \Rightarrow Cargando C

- Si V_i crece y $V_i > V_c \Rightarrow D_2$ continua en ON

Vo obtendrá el valor de V_i , que hasta ese momento ha sido el máximo que ha alcanzado.

$$V_o = V_c = V_{x1} = V_{y1} = V_i \quad (99)$$

$$V_{o1} = V_c + V_u = V_i + V_u \quad (100)$$

$$V_{D1} = V_{x1} - V_{o1} \quad (101)$$

Sustituimos (102) en (103) y obtenemos que:

$$V_{D1} = V_i - (V_i + V_u) = -V_u \rightarrow \text{D1 continua en OFF}$$

- Si V_i disminuye y $V_i < V_C \rightarrow V_{o1}$ se reduce y D2 pasa a ON

En este caso, V_o mantendrá el valor anterior, que será el máximo que haya alcanzado V_i hasta ese momento.

$$V_o = V_C = V_i \text{ (máx) anterior} \quad (102)$$

$$V_{o1} \rightarrow -V_{sat} ; V_{x1} \approx V_C \text{ y } V_{y1} = V_i < V_C \rightarrow V_{o1} = V_i - V_u \text{ y D1 pasa a ON}$$

- Si V_i vuelve a crecer y $V_i > V_C \rightarrow$ De nuevo D1 en OFF y D2 en ON

El condensador se volverá a cargar hasta el nuevo valor máximo que ha alcanzado V_i .

$$V_o = V_C = V_i \text{ (máx) nuevo} \quad (103)$$

En la *ilustración 66*, se puede ver un ejemplo del funcionamiento que se ha descrito:

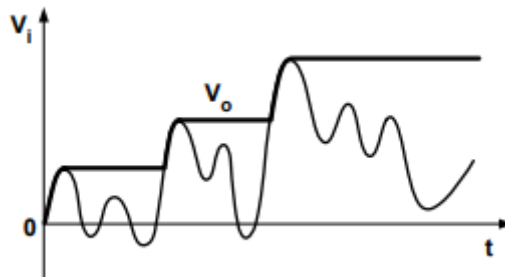


Ilustración 180. Representación Vi-t Vo-t del circuito detector de pico

Mediante el interruptor S descargamos la tensión acumulada en el condensador, y, por tanto, en la salida del circuito, con el fin de reiniciar la detección del pico de la señal de entrada.

Si se quieren detectar picos negativos, con invertir de sentido los dos diodos del circuito sería suficiente.



7.2. Código del programa

7.2.1. Código de hojas Excel

7.2.1.1. Hoja 1 (Amplificadores operacionales)

```
Private Sub boton_inversor_Click()      'Botón "Circuito inversor"'
ModInversor.MostrarInversor

End Sub

Private Sub boton_no_inversor_Click()  'Botón "Circuito no
inversor"'
ModNoInversor.MostrarNoInversor

End Sub

Private Sub boton_circuitos_sumadores_Click()  'Botón "Circuitos
sumadores"'
Circuitos.MostrarCircSum

End Sub

Private Sub boton_circuitos_restadores_Click() 'Botón "Circuitos
restadores"'
Circuitos.MostrarCircRes

End Sub

Private Sub boton_circuitos_integradores_Click() 'Botón
"Circuitos integradores"'
Circuitos.MostrarCircInt

End Sub

Private Sub boton_diferenciador_Click()      'Botón "Circuito
diferenciador"'
ModDiferenciador.MostrarDiferenciador

End Sub

Private Sub boton_Fuente_Tension_Click()    'Botón "Fuente de
```



```
tensión de referencia"'

ModFuenteTension.MostrarFuenteTension

End Sub

Private Sub boton_Fuente_Corriente_Click()      'Botón "Fuentes de
corriente"'

ModFuenteCorriente.MostrarFuenteCorriente

End Sub

Private Sub boton_Conv_CorrTen_Click()          'Botón "Convertidores
corriente-tensión"'

ModConvCorrTen.MostrarConvCorrTen

End Sub

Private Sub boton_Conv_TenCorr_Click()          'Botón "Convertidores
tensión-corriente"'

ModConvTenCorr.MostrarConvTenCorr

End Sub

Private Sub boton_circuitos_limitadores_Click() 'Botón "Circuitos
limitadores"'

Circuitos.MostrarCircLim

End Sub

Private Sub boton_comparadores_Click()          'Botón "Detectores de
tensión comparadores"'

Circuitos.MostrarCircCom

End Sub

Private Sub boton_histeresis_Click()            'Botón "Comparadores con
histéresis"'

ModHisteresis.MostrarHisteresis

End Sub

Private Sub boton_onda_cuadrada_Click()         'Botón "Generador de onda
cuadrada"'

ModOndaCua.MostrarOndaCuadrada

End Sub

Private Sub boton_onda_triangular_Click()       'Botón "Generador de
onda triangular"'

ModOndaTri.MostrarOndaTriangular

End Sub
```



```
Private Sub boton_rect_precision_Click()           'Botón "Rectificadores  
de precisión"  
  
Circuitos.MostrarCircRectPre  
  
End Sub  
  
Private Sub boton_detector_pico_Click()           'Botón "Detector de pico"  
  
ModDetectPico.MostrarDetectPico  
  
End Sub
```

7.2.1.2. ThisWorkbook

```
'Código para hacer que la aplicación ocupe toda la pantalla'  
  
Private Sub Workbook_Open()  
  
Application.DisplayFullScreen = True  
  
End Sub  
  
Private Sub Workbook_BeforeClose(Cancel As Boolean)  
  
On Error Resume Next  
  
Application.DisplayFullScreen = False  
  
End Sub
```

7.2.2. Código de formularios

7.2.2.1. Detectores de tensión o comparadores (CircCom)

```
Private Sub ComparadorSimple_Click()           'Botón "Comparador simple"  
  
ModCircComp.MostrarSimple  
  
End Sub  
  
Private Sub ComparadorSatSat_Click()           'Botón "Detector con salida  
sat-sat"  
  
ModCircComp.MostrarSatSat
```



```
End Sub
```

```
Private Sub ComparadorCeroSat_Click() 'Botón "Detector con salida  
cero-sat"
```

```
ModCircComp.MostrarCeroSat
```

```
End Sub
```

```
Private Sub ComparadorSchmitt_Click() 'Botón "Disipador de Schmitt"
```

```
ModCircComp.MostrarSchmitt
```

```
End Sub
```

7.2.2.2. Comparadores con histéresis (CircHis)

```
Private Sub HisteresisSimple_Click() 'Botón "Comparador con  
histéresis"
```

```
ModHisteresis.MostrarHisteresis1
```

```
End Sub
```

```
Private Sub HisteresisLimitado_Click() 'Botón "Comparador con  
histéresis y limitación la salida"
```

```
ModHisteresis.MostrarHisteresis2
```

```
End Sub
```

7.2.2.3. Circuitos integradores (CircInt)

```
Private Sub IntegradorInversor_Click() 'Botón "Integrador inversor"
```

```
ModIntInv.MostrarIntegradorInversor
```

```
End Sub
```

```
Private Sub IntegradorNoInversor_Click() 'Botón "Integrador no  
inversor"
```

```
ModIntNoInv.MostrarIntegradorNoInversor
```

```
End Sub
```

```
Private Sub IntegradorSumador_Click() 'Botón "Integrador sumador"
```

```
ModIntSum.MostrarIntegradorSumador
```



```
End Sub
```

```
Private Sub IntegradorRestador_Click() 'Botón "Integrador restador"
```

```
ModIntRes.MostrarIntegradorRestador
```

```
End Sub
```

7.2.2.4. Circuitos limitadores (CircLim)

```
Private Sub LimitadorDiodoNormal_Click() 'Botón "Limitador con  
diodo normal"
```

```
ModCircLim.MostrarDiodoNormal
```

```
End Sub
```

```
Private Sub LimitadorDiodoNormalDiodoZener_Click() 'Botón "Limitador  
con diodo normal y diodo zener en serie"
```

```
ModCircLim.MostrarDiodoNormalDiodoZener
```

```
End Sub
```

```
Private Sub LimitadorDosZener_Click() 'Botón "Limitador con dos  
diodos zener"
```

```
ModCircLim.MostrarDosZener
```

```
End Sub
```

7.2.2.5. Rectificadores de precisión (CircRectPre)

```
Private Sub RectMediaOnda_Click() 'Botón "Rectificador de media  
onda"
```

```
ModCircRectPre.MostrarMediaOnda
```

```
End Sub
```

```
Private Sub RectOndaCompleta_Click() 'Botón "Rectificador de onda  
completa"
```

```
ModCircRectPre.MostrarOndaCompleta
```

```
End Sub
```

```
Private Sub ValorMedioValorEficaz_Click() 'Botón "Circuitos de valor  
medio y valor eficaz"
```

```
ModCircRectPre.MostrarValorMedioValorEficaz
```



End Sub

7.2.2.6. Circuitos restadores (CircRes)

```
Private Sub RestadorSimple_Click() 'Botón "Restador simple"
```

```
ModRestador.MostrarRestadorSimple
```

End Sub

```
Private Sub RestadorAltaImp_Click() 'Botón "Restador de alta  
impedancia a la entrada"
```

```
ModRestador.MostrarRestadorAltaImp
```

End Sub

7.2.2.7. Circuitos sumadores (CircSum)

```
Private Sub SumadorInversor_Click() 'Botón "Sumador inversor"
```

```
ModSumador.MostrarSumadorInversor
```

End Sub

```
Private Sub SumadorNoInversor_Click() 'Botón "Sumador no inversor"
```

```
ModSumador.MostrarSumadorNoInversor
```

End Sub

7.2.2.8. Detector con salida Cero-Sat (CompCeroSat)

```
Private Sub Teoria1_Click() 'Botón "Teoría" del ejemplo 1'
```

```
ModCircComp.MostrarTeoriaCompCeroSat1
```

End Sub

```
Private Sub Teoria2_Click() 'Botón "Teoría" del ejemplo 2'
```

```
ModCircComp.MostrarTeoriaCompCeroSat2
```

End Sub

```
Private Sub Calcular1_Click() 'Botón "Calcular" del ejemplo 1'
```




```
ModCircComp.CalcularValoresCompCeroSat1

End Sub

Private Sub Calcular2_Click() 'Botón "Calcular" del ejemplo 2'
ModCircComp.CalcularValoresCompCeroSat2

End Sub

Private Sub Restaurar1_Click() 'Botón "Restaurar valores" del
ejemplo 1'
ModCircComp.RestaurarValoresCompCeroSat1

End Sub

Private Sub Restaurar2_Click() 'Botón "Restaurar valores" del
ejemplo 2'
ModCircComp.RestaurarValoresCompCeroSat2

End Sub

Private Sub Informacion1_Click() 'Botón "Información" del ejemplo 1'
ModCircComp.MostrarInformacionCompCeroSat

End Sub

Private Sub Informacion2_Click() 'Botón "Información" del ejemplo 2'
ModCircComp.MostrarInformacionCompCeroSat

End Sub

Private Sub Vo_Vi_1_Click() 'Botón "Representación Vo-Vi" del
ejemplo 1'

'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If CompCeroSat.Valor_Vo_1.Caption = Empty Then

    MsgBox "Debe obtener primero un valor de Vo para poder calcular la
gráfica Vo-Vi", vbExclamation + vbOKOnly, "AVISO"

Else

    'Cargamos la gráfica'
    Load GraficaVoVi_2

    'Refrescamos la gráfica'
    GraficaVoVi_2.Repaint

    'Mostramos la gráfica'
    GraficaVoVi_2.Show

End If

End Sub
```



```
Private Sub Vo_Vi_2_Click() 'Botón "Representación Vo-Vi" del
ejemplo 2'

'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If CompCeroSat.Valor_Vo_2.Caption = Empty Then

    MsgBox "Debe obtener primero un valor de Vo para poder calcular la
gráfica Vo-Vi", vbExclamation + vbOKOnly, "AVISO"

Else

    'Cargamos la gráfica'
    Load GraficaVoVi_2

    'Refrescamos la gráfica'
    GraficaVoVi_2.Repaint

    'Mostramos la gráfica'
    GraficaVoVi_2.Show

End If

End Sub

Private Sub UserForm_terminate()

'Vaciamos las celdas de la hoja de cálculo "Gráfica Vo-Vi"'
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s6") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t6") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s9") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t9") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s10") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t10") = Empty

End Sub
```

7.2.2.9. Detector con salida Sat-Sat (CompSatSat)

```
Private Sub Teoria_Click() 'Botón "Teoría"'

ModCircComp.MostrarTeoriaCompSatSat

End Sub

Private Sub Calcular_Click() 'Botón "Calcular"'

ModCircComp.CalcularValoresCompSatSat

End Sub

Private Sub Restaurar_Click() 'Botón "Restaurar valores"'
```



```
ModCircComp.RestaurarValoresCompSatSat

End Sub

Private Sub Informacion_Click() 'Botón "Información"

ModCircComp.MostrarInformacionCompSatSat

End Sub

Private Sub Vo_Vi_Click() 'Botón "Representación Vo-Vi"

'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de
Vo'
If CompSatSat.Valor_Vo.Caption = Empty Then

MsgBox "Debe obtener primero un valor de Vo para poder calcular la
gráfica Vo-Vi", vbExclamation + vbOKOnly, "AVISO"

Else

'Cargamos la gráfica'
Load GraficaVoVi_2

'Refrescamos la gráfica'
GraficaVoVi_2.Repaint

'Mostramos la gráfica'
GraficaVoVi_2.Show

End If

End Sub

Private Sub UserForm_terminate()

'Vaciamos las celdas de la hoja de calculo "Gráfica Vo-Vi"
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s6") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t6") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s9") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t9") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s10") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t10") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("q3") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("r3") = Empty

End Sub
```

7.2.2.10. Disipador de Schmitt (CompSchmitt)

```
Private Sub Teoria_Click() 'Botón "Teoría"

ModCircComp.MostrarTeoriaCompSchmitt
```



```
End Sub

Private Sub Calcular_Click() 'Botón "Calcular"'
ModCircComp.CalcularValoresCompSchmitt
End Sub

Private Sub Restaurar_Click() 'Botón "Restaurar valores"'
ModCircComp.RestaurarValoresCompSchmitt
End Sub

Private Sub Informacion_Click() 'Botón "Información"'
ModCircComp.MostrarInformacionCompSchmitt
End Sub

Private Sub ViVo_t_Click() 'Botón "Representación de Vi y Vo frente
a t"'

'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If CompSchmitt.Valor_Vo.Caption = Empty Then

    MsgBox "Debe obtener primero un valor de Vo para poder calcular
las gráficas Vi-t Vo-t", vbExclamation + vbOKOnly, "AVISO"

Else

    'Cargamos la gráfica'
    Load GraficaSchmitt

    'Refrescamos la gráfica'
    GraficaSchmitt.Repaint

    'Mostramos la gráfica'
    GraficaSchmitt.Show

End If

End Sub

Private Sub UserForm_terminate()

'Vaciamos las celdas de la hoja de cálculo "Gráfica Vo-t"'
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("t5") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("u5") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v5") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("s8:s728") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v8:v728") = Empty

'Vaciamos las celdas de la hoja de cálculo "Gráfica CosSen"'
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("a8:a728") = Empty
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("w8:w728") = Empty

End Sub
```



7.2.2.11. Comparador Simple (CompSimple)

```
Private Sub Teoria_Click() 'Botón "Teoría"
ModCircComp.MostrarTeoriaCompSimple
End Sub

Private Sub Calcular_Click() 'Botón "Calcular"
ModCircComp.CalcularValoresCompSimple
End Sub

Private Sub Restaurar_Click() 'Botón "Restaurar valores"
ModCircComp.RestaurarValoresCompSimple
End Sub

Private Sub Informacion_Click() 'Botón "Información"
ModCircComp.MostrarInformacionCompSimple
End Sub

Private Sub Vo_Vi_Click() 'Botón "Representación Vo-Vi"

'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If CompSimple.Valor_Vo.Caption = Empty Then

MsgBox "Debe obtener primero un valor de Vo para poder calcular la
gráfica Vo-Vi", vbExclamation + vbOKOnly, "AVISO"

Else

'Cargamos la gráfica'
Load GraficaVoVi_2

'Refrescamos la gráfica'
GraficaVoVi_2.Repaint

'Mostramos la gráfica'
GraficaVoVi_2.Show

End If

End Sub

Private Sub UserForm_terminate()

'Vaciamos las celdas de la hoja de cálculo "Gráfica Vo-Vi"
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s6") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t6") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s9") = Empty
```



```
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t9") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s10") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t10") = Empty
```

```
End Sub
```

7.2.2.12. Convertidores corriente-tensión (ConvCorrTen)

```
Private Sub Teoria_1_Click() 'Botón "Teoría" del ejemplo 1'
```

```
ModConvCorrTen.MostrarTeoriaConvCorrTen
```

```
End Sub
```

```
Private Sub Teoria_2_Click() 'Botón "Teoría" del ejemplo 2'
```

```
ModConvCorrTen.MostrarTeoriaConvCorrTen
```

```
End Sub
```

```
Private Sub Calcular_1_Click() 'Botón "Calcular" del ejemplo 1'
```

```
ModConvCorrTen.CalcularValores1
```

```
End Sub
```

```
Private Sub Calcular_2_Click() 'Botón "Calcular" del ejemplo 2'
```

```
ModConvCorrTen.CalcularValores2
```

```
End Sub
```

```
Private Sub Restaurar_1_Click() 'Botón "Restaurar valores" del ejemplo  
1'
```

```
ModConvCorrTen.RestaurarValores1
```

```
End Sub
```

```
Private Sub Restaurar_2_Click() 'Botón "Restaurar valores" del ejemplo  
2'
```

```
ModConvCorrTen.RestaurarValores2
```

```
End Sub
```

```
Private Sub Informacion1_Click() 'Botón "Información" del ejemplo 1'
```

```
ModConvCorrTen.MostrarInformacion
```

```
End Sub
```

```
Private Sub Informacion2_Click() 'Botón "Información" del ejemplo 2'
```



```
ModConvCorrTen.MostrarInformacion
```

```
End Sub
```

7.2.2.13. Convertidores tensión-corriente (ConvTenCorr)

```
Private Sub Teoria_Click() 'Botón "Teoría"
```

```
ModConvTenCorr.MostrarTeoriaConvTenCorr
```

```
End Sub
```

```
Private Sub Calcular_Click() 'Botón "Calcular"
```

```
ModConvTenCorr.CalcularValores
```

```
End Sub
```

```
Private Sub Restaurar_Click() 'Botón "Restaurar valores"
```

```
ModConvTenCorr.RestaurarValores
```

```
End Sub
```

```
Private Sub Informacion_Click() 'Botón "Información"
```

```
ModConvTenCorr.MostrarInformacion
```

```
End Sub
```

7.2.2.14. Detector de pico (DetectPico)

```
Private Sub Teoria_Click() 'Botón "Teoría"
```

```
ModDetectPico.MostrarTeoriaDetectorPico
```

```
End Sub
```

```
Private Sub UserForm_Activate() 'Definimos el inicio de los  
contadores para almacenar que pico es el mínimo'
```

```
Dim i As Integer
```

```
Dim k1 As Integer
```

```
Dim k2 As Integer
```

```
i = 5
```

```
k1 = 5
```

```
k2 = 0
```

```
DetectPico.i.Caption = i
```



```
DetectPico.i.Visible = False
DetectPico.k1.Caption = k1
DetectPico.k1.Visible = False
DetectPico.k2.Caption = k2
DetectPico.k2.Visible = False

End Sub

Private Sub Calcular_Click()      'Botón "Calcular"'

ModDetectPico.CalcularValores

'Si hemos obtenido un valor de pico, comenzamos a guardar valores y
aumentamos los contadores cada vez que calculemos el valor de pico'
If DetectPico.Valor_Vp.Caption <> Empty Then

    Dim i As Integer
    Dim k1 As Integer
    Dim k2 As Integer

    DetectPico.i.Caption = DetectPico.i.Caption + 1
    DetectPico.i.Visible = False
    DetectPico.k1.Caption = DetectPico.k1.Caption + (360 / b)
    DetectPico.k1.Visible = False
    DetectPico.k2.Caption = DetectPico.k2.Caption + (360 / b)
    DetectPico.k2.Visible = False

End If

End Sub

Private Sub Restaurar_Click()    'Botón "Restaurar valores"'

ModDetectPico.RestaurarValores

End Sub

Private Sub Interruptor_Click() 'Botón "Interruptor S"'

ModDetectPico.Interruptor

End Sub

Private Sub Informacion_Click() 'Botón "Información"'

ModDetectPico.MostrarInformacion

End Sub

Private Sub Vo_t_Click()        'Botón "Representación Vo-t"'

'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If DetectPico.Valor_Vp.Caption = Empty Then

    MsgBox "Debe obtener primero un valor de Vo para poder calcular la
gráfica Vo-t", vbExclamation + vbOKOnly, "AVISO"

Else

    'Cargamos la gráfica'
```




```
Load GraficaDetectPico

'Refrescamos la gráfica'
GraficaDetectPico.Repaint

'Mostramos la gráfica'
GraficaDetectPico.Show

End If

End Sub

Private Sub UserForm_terminate()

'Vaciamos las celdas de la hoja de cálculo "DetectPico"'
ActiveWorkbook.Worksheets("DetectPico").Range("b2") = Empty
ActiveWorkbook.Worksheets("DetectPico").Range("c2") = Empty
ActiveWorkbook.Worksheets("DetectPico").Range("a5:a1200") = Empty
ActiveWorkbook.Worksheets("DetectPico").Range("c5:c1200") = Empty
ActiveWorkbook.Worksheets("DetectPico").Range("d5:d1200") = Empty
ActiveWorkbook.Worksheets("DetectPico").Range("q5:q1200") = Empty
ActiveWorkbook.Worksheets("DetectPico").Range("r5:r1200") = Empty
ActiveWorkbook.Worksheets("DetectPico").Range("s5:s1200") = Empty

End Sub
```

7.2.2.15. Circuito diferenciador (Diferenciador)

```
Private Sub Teoria_Click() 'Botón "Teoría"'

ModDiferenciador.MostrarTeoriaDiferenciador

End Sub

Private Sub Calcular_Click() 'Botón "Calcular"'

ModDiferenciador.CalcularValores

End Sub

Private Sub Restaurar_Click() 'Botón "Restaurar valores"'

ModDiferenciador.RestaurarValores

End Sub

Private Sub Informacion_Click() 'Botón "Información"'

ModDiferenciador.MostrarInformacion

End Sub

Private Sub ViVo_t_Click() 'Botón "Representación de Vi y Vo frente  
a t"'
```



```
'Si no hemos obtenido previamente un valor de Vo, no se mostrará la  
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'  
If Diferenciador.Valor_Vo.Caption = Empty Then
```

```
MsgBox "Debe obtener primero un valor de Vo para poder calcular  
las gráficas Vi-t Vo-t", vbExclamation + vbOKOnly, "AVISO"
```

```
Else
```

```
  'Cargamos la gráfica'  
  Load GraficaDif
```

```
  'Refrescamos la gráfica'  
  GraficaDif.Repaint
```

```
  'Mostramos la gráfica'  
  GraficaDif.Show
```

```
End If
```

```
End Sub
```

```
Private Sub UserForm_terminate()
```

```
  'Vaciamos las celdas de la hoja de cálculo "Diferenciador"
```

```
  ActiveWorkbook.Worksheets("Diferenciador").Range("c6") = Empty  
  ActiveWorkbook.Worksheets("Diferenciador").Range("c7") = Empty  
  ActiveWorkbook.Worksheets("Diferenciador").Range("c8") = Empty  
  ActiveWorkbook.Worksheets("Diferenciador").Range("c9") = Empty  
  ActiveWorkbook.Worksheets("Diferenciador").Range("c15") = Empty  
  ActiveWorkbook.Worksheets("Diferenciador").Range("c16") = Empty  
  ActiveWorkbook.Worksheets("Diferenciador").Range("c18") = Empty  
  ActiveWorkbook.Worksheets("Diferenciador").Range("d18") = Empty  
  ActiveWorkbook.Worksheets("Diferenciador").Range("c19") = Empty  
  ActiveWorkbook.Worksheets("Diferenciador").Range("d19") = Empty  
  ActiveWorkbook.Worksheets("Diferenciador").Range("c21") = Empty  
  ActiveWorkbook.Worksheets("Diferenciador").Range("d21") = Empty
```

```
  'Vaciamos las celdas de la hoja de cálculo "Gráfica Vo-t"
```

```
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c36") = Empty  
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c37") = Empty  
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c47") = Empty  
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("d47") = Empty
```

```
End Sub
```

7.2.2.16. Fuentes de corriente (FuenteCorriente)

```
Private Sub Teoria_1_Click()      'Botón "Teoría" del ejemplo 1'
```

```
ModFuenteCorriente.MostrarTeoriaFuenteCorriente
```

```
End Sub
```



```
Private Sub Teoria_2_Click() 'Botón "Teoría" del ejemplo 2'  
ModFuenteCorriente.MostrarTeoriaFuenteCorriente  
End Sub  
  
Private Sub Calcular_1_Click() 'Botón "Calcular" del ejemplo 1'  
ModFuenteCorriente.CalcularValores1  
End Sub  
  
Private Sub Calcular_2_Click() 'Botón "Calcular" del ejemplo 2'  
ModFuenteCorriente.CalcularValores2  
End Sub  
  
Private Sub Restaurar_1_Click() 'Botón "Restaurar valores" del ejemplo  
1'  
ModFuenteCorriente.RestaurarValores1  
End Sub  
  
Private Sub Restaurar_2_Click() 'Botón "Restaurar valores" del ejemplo  
2'  
ModFuenteCorriente.RestaurarValores2  
End Sub  
  
Private Sub Informacion1_Click() 'Botón "Información" del ejemplo 1'  
ModFuenteCorriente.MostrarInformacion  
End Sub  
  
Private Sub Informacion2_Click() 'Botón "Información" del ejemplo 2'  
ModFuenteCorriente.MostrarInformacion  
End Sub
```

7.2.2.17. Fuente de tensión de referencia (FuenteTension)

```
Private Sub Teoria_Click() 'Botón "Teoría"  
ModFuenteTension.MostrarTeoriaFuenteTension  
End Sub  
  
Private Sub Calcular_Click() 'Botón "Calcular"
```



```
ModFuenteTension.CalcularValores

End Sub

Private Sub Restaurar_Click() 'Botón "Restaurar valores"

ModFuenteTension.RestaurarValores

End Sub

Private Sub Informacion_Click() 'Botón "Información"

ModFuenteTension.MostrarInformacion

End Sub
```

7.2.2.18. GraficaCosSen

```
Private Sub UserForm_Initialize()

'Para poder mostrar las gráficas Vi-t Vo-t, antes tiene que estar
visible la pestaña del circuito integrador inversor'
If IntegradorInversor.Visible = True And
IntegradorInversor.Valor_Vo_2.Caption <> Empty Then

    'Cada valor lo obtenemos a partir de las hojas de cálculo "Gráfica
    CosSen" y Gráfica Vo-t, definidos en sus respectivas celdas'

    'En función de si la entrada es un seno o un coseno, recogemos
    unos valores u otros'
    If IntegradorInversor.Sen_Cos.Text = "sen" Then

        GraficaCosSen.SenCos_1.Caption = "sen"
        GraficaCosSen.a_1.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-t").Range("u5"), 2)
        GraficaCosSen.b_1.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-t").Range("v5"), 2)

        GraficaCosSen.SenCos_2.Caption = "cos"
        GraficaCosSen.a_2.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("n8"), 2)
        GraficaCosSen.b_2.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("o8"), 2)

        'Si hay tensi□nicial la definimos en nuestro formulario'
        If IntegradorInversor.Valor_VoInicial_2 > 0 Then

            GraficaCosSen.Vo_Inicial.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("p8"), 2)
            GraficaCosSen.signo.Caption = "+"
```



```
ElseIf IntegradorInversor.Valor_VoInicial_2 < 0 Then

    GraficaCosSen.Vo_Inicial.Caption =
Abs (Application.WorksheetFunction.Round (ActiveWorkbook.Worksheets ("Grá
fica CosSen").Range ("p8"), 2))
    GraficaCosSen.signo.Caption = "-"

ElseIf IntegradorInversor.Valor_VoInicial_2 = 0 Then

    GraficaCosSen.Vo_Inicial.Visible = False
    GraficaCosSen.Vo_Inicial.Caption = ""
    GraficaCosSen.signo.Caption = ""

End If

ElseIf IntegradorInversor.Sen_Cos.Text = "cos" Then

    GraficaCosSen.SenCos_1.Caption = "cos"
    GraficaCosSen.a_1.Caption =
Application.WorksheetFunction.Round (ActiveWorkbook.Worksheets ("Gráfica
Vo-t").Range ("w5"), 2)
    GraficaCosSen.b_1.Caption =
Application.WorksheetFunction.Round (ActiveWorkbook.Worksheets ("Gráfica
Vo-t").Range ("x5"), 2)

    GraficaCosSen.SenCos_2.Caption = "sen"
    GraficaCosSen.a_2.Caption =
Application.WorksheetFunction.Round (ActiveWorkbook.Worksheets ("Gráfica
CosSen").Range ("n9"), 2)
    GraficaCosSen.b_2.Caption =
Application.WorksheetFunction.Round (ActiveWorkbook.Worksheets ("Gráfica
CosSen").Range ("o9"), 2)

    'Si hay tensiñnicial la definimos en nuestro formulario'
    If IntegradorInversor.Valor_VoInicial_2 > 0 Then

        GraficaCosSen.Vo_Inicial.Caption =
Application.WorksheetFunction.Round (ActiveWorkbook.Worksheets ("Gráfica
CosSen").Range ("p9"), 2)
        GraficaCosSen.signo.Caption = "+"

    ElseIf IntegradorInversor.Valor_VoInicial_2 < 0 Then

        GraficaCosSen.Vo_Inicial.Caption =
Abs (Application.WorksheetFunction.Round (ActiveWorkbook.Worksheets ("Grá
fica CosSen").Range ("p9"), 2))
        GraficaCosSen.signo.Caption = "-"

    ElseIf IntegradorInversor.Valor_VoInicial_2 = 0 Then

        GraficaCosSen.Vo_Inicial.Visible = False
        GraficaCosSen.Vo_Inicial.Caption = ""
        GraficaCosSen.signo.Caption = ""

    End If

End If

'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-t"
Set Grafico = Sheets ("Gráfica Vo-t").ChartObjects (1).Chart
```



```
'Definimos dicho gráfico como un GIF'
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &
"temp.gif"
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"

'En la imagen de nuestro formulario mostramos el GIF capturado,
qué representa la gráfica'
Image1.Picture = LoadPicture(NombreArchivo)

'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica CosSen"'
Set Grafico = Sheets("Gráfica CosSen").ChartObjects(1).Chart

'Definimos dicho gráfico como un GIF'
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &
"temp.gif"
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"

'En la imagen de nuestro formulario mostramos el GIF capturado,
qué representa la gráfica'
Image2.Picture = LoadPicture(NombreArchivo)

'Para poder mostrar las gráficas Vi-t Vo-t, antes tiene que estar
visible la pestaña del circuito integrador no inversor'
ElseIf IntegradorNoInversor.Visible = True And
IntegradorNoInversor.Valor_Vo_2 <> Empty Then

    'Cada valor lo obtenemos a partir de las hojas de cálculo "Gráfica
    CosSen" y Gráfica Vo-t, definidos en sus respectivas celdas'

    'En función de si la entrada es un seno o un coseno, recogemos
    unos valores u otros'
    If IntegradorNoInversor.Sen_Cos.Text = "sen" Then

        GraficaCosSen.SenCos_1.Caption = "sen"
        GraficaCosSen.a_1.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-t").Range("y5"), 2)
        GraficaCosSen.b_1.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-t").Range("z5"), 2)

        GraficaCosSen.SenCos_2.Caption = "cos"
        GraficaCosSen.a_2.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("n10"), 2)
        GraficaCosSen.b_2.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("o10"), 2)

        If IntegradorNoInversor.Valor_VoInicial_2 > 0 Then

            GraficaCosSen.Vo_Inicial.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("p10"), 2)
            GraficaCosSen.signo.Caption = "+"

        ElseIf IntegradorNoInversor.Valor_VoInicial_2 < 0 Then

            GraficaCosSen.Vo_Inicial.Caption =
Abs(Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Grá
fica CosSen").Range("p10"), 2))

        End If

    End If

End If
```



```
GraficaCosSen.signo.Caption = "-"

ElseIf IntegradorNoInversor.Valor_VoInicial_2 = 0 Then

    GraficaCosSen.Vo_Inicial.Visible = False
    GraficaCosSen.Vo_Inicial.Caption = ""
    GraficaCosSen.signo.Caption = ""

End If

ElseIf IntegradorNoInversor.Sen_Cos.Text = "cos" Then

    GraficaCosSen.SenCos_1.Caption = "cos"
    GraficaCosSen.a_1.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-t").Range("aa5"), 2)
    GraficaCosSen.b_1.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-t").Range("ab5"), 2)

    GraficaCosSen.SenCos_2.Caption = "sen"
    GraficaCosSen.a_2.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("n11"), 2)
    GraficaCosSen.b_2.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("o11"), 2)

    If IntegradorNoInversor.Valor_VoInicial_2 > 0 Then

        GraficaCosSen.Vo_Inicial.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("p11"), 2)
        GraficaCosSen.signo.Caption = "+"

    ElseIf IntegradorNoInversor.Valor_VoInicial_2 < 0 Then

        GraficaCosSen.Vo_Inicial.Caption =
Abs(Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Grá
fica CosSen").Range("p11"), 2))
        GraficaCosSen.signo.Caption = "-"

    ElseIf IntegradorNoInversor.Valor_VoInicial_2 = 0 Then

        GraficaCosSen.Vo_Inicial.Visible = False
        GraficaCosSen.Vo_Inicial.Caption = ""
        GraficaCosSen.signo.Caption = ""

    End If

End If

'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-t"
Set Grafico = Sheets("Gráfica Vo-t").ChartObjects(1).Chart

'Definimos dicho gráfico como un GIF'
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &
"temp.gif"
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"
```



```
'En la imagen de nuestro formulario mostramos el GIF capturado,  
qué representa la gráfica'  
Image1.Picture = LoadPicture(NombreArchivo)  
  
'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica CosSen"  
Set Grafico = Sheets("Gráfica CosSen").ChartObjects(1).Chart  
  
'Definimos dicho gráfico como un GIF'  
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &  
"temp.gif"  
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"  
  
'En la imagen de nuestro formulario mostramos el GIF capturado,  
qué representa la gráfica'  
Image2.Picture = LoadPicture(NombreArchivo)  
  
End If  
  
End Sub
```

7.2.2.19. GraficaDetectPico

```
Private Sub UserForm_Initialize()  
  
'Para poder mostrar la gráfica Vo-t, antes tiene que estar visible la  
pestaña del circuito detector de pico'  
If DetectPico.Visible = True Then  
  
    'Recogemos la gráfica de nuestra hoja de cálculo "DetectPico"  
    Set Grafico = Sheets("DetectPico").ChartObjects(1).Chart  
  
    'Definimos dicho gráfico como un GIF'  
    NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &  
    "temp.gif"  
    Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"  
  
    'En la imagen de nuestro formulario mostramos el GIF capturado,  
    que representa la gráfica'  
    Image.Picture = LoadPicture(NombreArchivo)  
  
End If  
  
End Sub
```

7.2.2.20. GraficaDif

```
Private Sub UserForm_Initialize()  
  
'Para poder mostrar las gráficas Vi-t Vo-t, antes tiene que estar  
visible la pestaña del circuito diferenciador'  
If Diferenciador.Visible = True Then
```




```
'En función de si la entrada es positiva o negativa, tendremos
unos valores u otros'
If Diferenciador.Valor_Vi < 0 Then

    'A su vez, diferenciamos dos casos, si la potencia de la
    entrada es 1 o mayor que 1'
    If Diferenciador.n.Value = 1 Then

        'Establecemos qué valores queremos mostrar en la pestaña
        de la gráfica'
        GraficaDif.texto1.Visible = False
        GraficaDif.t_final2.Visible = False
        GraficaDif.texto2.Visible = False
        GraficaDif.Vsat.Visible = False
        GraficaDif.igual.Visible = False
        GraficaDif.Valor_Vsat.Visible = False
        GraficaDif.flecha.Visible = False
        GraficaDif.V.Visible = False
        GraficaDif.s.Visible = False
        GraficaDif.t_final.Visible = False
        GraficaDif.V.Visible = False
        GraficaDif.tigual.Visible = False
        GraficaDif.n1.Visible = False
        GraficaDif.t.Visible = False

        'Analizamos si alcanzamos la tensión de saturación'
        If
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Diferen
ciador").Range("g3"), 2) > Diferenciador.Valor_VsatPos.Value Then

            'Establecemos qué valores queremos mostrar en la
            pestaña de la gráfica'
            GraficaDif.Vosatura.Visible = True
            GraficaDif.igual_2.Visible = True
            GraficaDif.Vsat_2.Visible = True
            GraficaDif.Valor_Vsat_2.Visible = True

            GraficaDif.Vsat_2.Caption = "+Vsat"
            GraficaDif.Valor_Vsat_2.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Diferen
ciador").Range("c15"), 2)

        Else

            'Establecemos qué valores queremos mostrar en la
            pestaña de la gráfica'
            GraficaDif.Vosatura.Visible = False
            GraficaDif.igual_2.Visible = False
            GraficaDif.Vsat_2.Visible = False
            GraficaDif.Valor_Vsat_2.Visible = False

        End If

    ElseIf Diferenciador.n.Value > 1 Then

        'Establecemos qué valores queremos mostrar en la pestaña
        de la gráfica'
        GraficaDif.texto1.Visible = True
        GraficaDif.t_final2.Visible = True
        GraficaDif.texto2.Visible = True
        GraficaDif.Vsat.Visible = True
```



```
GraficaDif.igual.Visible = True
GraficaDif.Valor_Vsat.Visible = True
GraficaDif.flecha.Visible = True
GraficaDif.V.Visible = True
GraficaDif.s.Visible = True
GraficaDif.t_final.Visible = True
GraficaDif.V.Visible = True
GraficaDif.tigual.Visible = True
GraficaDif.n1.Visible = True
GraficaDif.t.Visible = True
GraficaDif.Vosatura.Visible = False
GraficaDif.igual_2.Visible = False
GraficaDif.Vsat_2.Visible = False
GraficaDif.Valor_Vsat_2.Visible = False

GraficaDif.Vsat.Caption = "+Vsat"
GraficaDif.Valor_Vsat.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Diferen
ciador").Range("c15"), 2)

End If

ElseIf Diferenciador.Valor_Vi > 0 Then

    If Diferenciador.n.Value = 1 Then

        'Establecemos qué valores queremos mostrar en la pestaña
de la gráfica'
        GraficaDif.texto1.Visible = False
        GraficaDif.t_final2.Visible = False
        GraficaDif.texto2.Visible = False
        GraficaDif.Vsat.Visible = False
        GraficaDif.igual.Visible = False
        GraficaDif.Valor_Vsat.Visible = False
        GraficaDif.flecha.Visible = False
        GraficaDif.V.Visible = False
        GraficaDif.s.Visible = False
        GraficaDif.t_final.Visible = False
        GraficaDif.V.Visible = False
        GraficaDif.tigual.Visible = False
        GraficaDif.n1.Visible = False
        GraficaDif.t.Visible = False

        If
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Diferen
ciador").Range("g3"), 2) < Diferenciador.Valor_VsatNeg.Value Then

            'Establecemos qué valores queremos mostrar en la
pestaña de la gráfica'
            GraficaDif.Vosatura.Visible = True
            GraficaDif.igual_2.Visible = True
            GraficaDif.Vsat_2.Visible = True
            GraficaDif.Valor_Vsat_2.Visible = True

            GraficaDif.Vsat_2.Caption = "-Vsat"
            GraficaDif.Valor_Vsat_2.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Diferen
ciador").Range("c16"), 2)

        Else
```



```
'Establecemos qué valores queremos mostrar en la
pestaña de la gráfica'
GraficaDif.Vosatura.Visible = False
GraficaDif.igual_2.Visible = False
GraficaDif.Vsat_2.Visible = False
GraficaDif.Valor_Vsat_2.Visible = False

End If

ElseIf Diferenciador.n.Value > 1 Then

'Establecemos qué valores queremos mostrar en la pestaña
de la gráfica'
GraficaDif.texto1.Visible = True
GraficaDif.t_final2.Visible = True
GraficaDif.texto2.Visible = True
GraficaDif.Vsat.Visible = True
GraficaDif.igual.Visible = True
GraficaDif.Valor_Vsat.Visible = True
GraficaDif.flecha.Visible = True
GraficaDif.V.Visible = True
GraficaDif.s.Visible = True
GraficaDif.t_final.Visible = True
GraficaDif.V.Visible = True
GraficaDif.tigual.Visible = True
GraficaDif.n1.Visible = True
GraficaDif.t.Visible = True
GraficaDif.Vosatura.Visible = False
GraficaDif.igual_2.Visible = False
GraficaDif.Vsat_2.Visible = False
GraficaDif.Valor_Vsat_2.Visible = False

GraficaDif.Vsat.Caption = "-Vsat"
GraficaDif.Valor_Vsat.Caption =

Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Diferen
ciador").Range("c16"), 2)

End If

End If

'Cada valor lo obtenemos a partir de la hoja de cálculo
"Diferenciador" y la hoja de cálculo "Gráfica Vo-t", definidos en sus
respectivas celdas'
GraficaDif.Ganancia_Vo.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Diferen
ciador").Range("g3"), 2)
GraficaDif.n1.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Diferen
ciador").Range("d8"), 2)
GraficaDif.t_final.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Diferen
ciador").Range("c18"), 2)
GraficaDif.t_final2.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Diferen
ciador").Range("c18"), 2)
GraficaDif.Ganancia_Vi.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-t").Range("c36"), 2)
```



```
GraficaDif.n2.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-t").Range("c37"), 2)  
  
'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-t"  
Set Grafico = Sheets("Gráfica Vo-t").ChartObjects(1).Chart  
  
'Definimos dicho gráfico como un GIF'  
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &  
"temp.gif"  
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"  
  
'En la imagen de nuestro formulario mostramos el GIF capturado,  
que representa la gráfica'  
Image1.Picture = LoadPicture(NombreArchivo)  
  
'Recogemos la gráfica de nuestra hoja de cálculo "Diferenciador"  
Set Grafico = Sheets("Diferenciador").ChartObjects(1).Chart  
  
'Definimos dicho gráfico como un GIF'  
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &  
"temp.gif"  
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"  
  
'En la imagen de nuestro formulario mostramos el GIF capturado,  
que representa la gráfica'  
Image2.Picture = LoadPicture(NombreArchivo)  
  
End If  
  
End Sub
```

7.2.2.21. GraficaHisteresis

```
Private Sub UserForm_Initialize()  
  
'Para poder mostrar la gráfica Vo-Vi, antes tiene que estar visible la  
pestaña del circuito comparador con histéresis'  
If Histeresis1.Visible = True Then  
  
'Cada valor lo obtenemos a partir de la hoja de cálculo "Gráfica  
Vo-Vi" definidos en sus respectivas celdas'  
GraficaHisteresis.a.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("s14"), 2)  
GraficaHisteresis.b.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("v14"), 2)  
GraficaHisteresis.Banda.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("w17"), 2)  
  
'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-Vi"  
Set Grafico = Sheets("Gráfica Vo-Vi").ChartObjects(1).Chart  
  
'Definimos dicho gráfico como un GIF'
```



```
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &
"temp.gif"
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"

'En la imagen de nuestro formulario mostramos el GIF capturado,
que representa la gráfica'
Image.Picture = LoadPicture(NombreArchivo)

'Para poder mostrar la gráfica Vo-Vi, antes tiene que estar visible la
pestaña del ejemplo 1 de un circuito comparador con histéresis y
limitación en la salida'
ElseIf Histeresis2.Visible = True And Histeresis2.Valor_Vo_1_1.Caption
<> Empty Then

    'Cada valor lo obtenemos a partir de la hoja de cálculo "Gráfica
Vo-Vi" definidos en sus respectivas celdas'
    GraficaHisteresis.a.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("s30"), 2)
    GraficaHisteresis.b.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("v30"), 2)
    GraficaHisteresis.Banda.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("w17"), 2)

    'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-Vi"
Set Grafico = Sheets("Gráfica Vo-Vi").ChartObjects(1).Chart

    'Definimos dicho gráfico como un GIF'
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &
"temp.gif"
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"

'En la imagen de nuestro formulario mostramos el GIF capturado,
que representa la gráfica'
Image.Picture = LoadPicture(NombreArchivo)

'Para poder mostrar la gráfica Vo-Vi, antes tiene que estar visible la
pestaña del ejemplo 2 de un circuito comparador con histéresis y
limitación en la salida'
ElseIf Histeresis2.Visible = True And Histeresis2.Valor_Vo_1_2.Caption
<> Empty Then

    'Cada valor lo obtenemos a partir de la hoja de cálculo "Gráfica
Vo-Vi" definidos en sus respectivas celdas'
    GraficaHisteresis.a.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("s23"), 2)
    GraficaHisteresis.b.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("v23"), 2)
    GraficaHisteresis.Banda.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("w17"), 2)

    'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-Vi"
Set Grafico = Sheets("Gráfica Vo-Vi").ChartObjects(1).Chart

    'Definimos dicho gráfico como un GIF'
```



```
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &
"temp.gif"
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"

'En la imagen de nuestro formulario mostramos el GIF capturado,
que representa la gráfica'
Image.Picture = LoadPicture(NombreArchivo)

End If

End Sub
```

7.2.2.22. GraficaSchmitt

```
Private Sub UserForm_Initialize()

'Para poder mostrar la gráfica Vo-t, antes tiene que estar visible la
pestaña del circuito disipador de Schmitt'
If CompSchmitt.Visible = True Then

'Establecemos qué valores queremos mostrar en la pestaña de la
gráfica'
GraficaSchmitt.ViMenor.Visible = True
GraficaSchmitt.ViMayor.Visible = True
GraficaSchmitt.Vi_Neg_1.Visible = True
GraficaSchmitt.Vi_Pos_1.Visible = True
GraficaSchmitt.V_1.Visible = True
GraficaSchmitt.V_2.Visible = True
GraficaSchmitt.Flecha1.Visible = True
GraficaSchmitt.Flecha2.Visible = True
GraficaSchmitt.Vo1.Visible = True
GraficaSchmitt.Vo2.Visible = True
GraficaSchmitt.Vo_1.Visible = True
GraficaSchmitt.Vo_2.Visible = True
GraficaSchmitt.V_3.Visible = True
GraficaSchmitt.V_4.Visible = True
GraficaSchmitt.Vo3.Visible = False
GraficaSchmitt.Vo_med_rms.Visible = False
GraficaSchmitt.flecha3.Visible = False
GraficaSchmitt.vmed_vrms.Visible = False
GraficaSchmitt.V_5.Visible = False
GraficaSchmitt.ViMayor1.Visible = False
GraficaSchmitt.ViMenor1.Visible = False
GraficaSchmitt.Vi_Pos.Visible = False
GraficaSchmitt.Vi_Neg.Visible = False
GraficaSchmitt.V_6.Visible = False
GraficaSchmitt.V_7.Visible = False
GraficaSchmitt.V_8.Visible = False
GraficaSchmitt.V_9.Visible = False
GraficaSchmitt.Flecha4.Visible = False
GraficaSchmitt.Flecha5.Visible = False
GraficaSchmitt.Vo4.Visible = False
GraficaSchmitt.Vo5.Visible = False
GraficaSchmitt.Ganancia.Visible = False
GraficaSchmitt.Ganancia_2.Visible = False
GraficaSchmitt.sen_2.Visible = False
GraficaSchmitt.sen_3.Visible = False
```



```
GraficaSchmitt.b_2.Visible = False
GraficaSchmitt.b_3.Visible = False
GraficaSchmitt.t_sen_2.Visible = False
GraficaSchmitt.t_sen_3.Visible = False

'Definimos los valores que queremos mostrar en la pestaña'
GraficaSchmitt.a.Caption = CompSchmitt.a.Value
GraficaSchmitt.b.Caption = CompSchmitt.b.Value
GraficaSchmitt.Vo_1.Caption = CompSchmitt.Valor_VsatPos.Value
GraficaSchmitt.Vo_2.Caption = CompSchmitt.Valor_VsatNeg.Value

'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-t"'
Set Grafico = Sheets("Gráfica Vo-t").ChartObjects(1).Chart

'Definimos dicho gráfico como un GIF'
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &
"temp.gif"
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"

'En la imagen de nuestro formulario mostramos el GIF capturado,
que representa la gráfica'
Image1.Picture = LoadPicture(NombreArchivo)

'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica CosSen"'
Set Grafico = Sheets("Gráfica CosSen").ChartObjects(1).Chart

'Definimos dicho gráfico como un GIF'
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &
"temp.gif"
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"

'En la imagen de nuestro formulario mostramos el GIF capturado,
que representa la gráfica'
Image2.Picture = LoadPicture(NombreArchivo)

'Para poder mostrar la gráfica Vo-t, antes tiene que estar visible la
pestaña del circuito rectificador de media onda'
ElseIf RectMediaOnda.Visible = True Then

'Establecemos qué valores queremos mostrar en la pestaña de la
gráfica'
GraficaSchmitt.ViMayor1.Visible = True
GraficaSchmitt.ViMenor1.Visible = True
GraficaSchmitt.Vi_Pos.Visible = True
GraficaSchmitt.Vi_Neg.Visible = True
GraficaSchmitt.V_6.Visible = True
GraficaSchmitt.V_7.Visible = True
GraficaSchmitt.V_8.Visible = True
GraficaSchmitt.V_9.Visible = True
GraficaSchmitt.Flecha4.Visible = True
GraficaSchmitt.Flecha5.Visible = True
GraficaSchmitt.Vo4.Visible = True
GraficaSchmitt.Vo5.Visible = True
GraficaSchmitt.Ganancia.Visible = True
GraficaSchmitt.Ganancia_2.Visible = True
GraficaSchmitt.sen_2.Visible = True
GraficaSchmitt.sen_3.Visible = True
GraficaSchmitt.b_2.Visible = True
GraficaSchmitt.b_3.Visible = True
GraficaSchmitt.t_sen_2.Visible = True
GraficaSchmitt.t_sen_3.Visible = True
```



```
GraficaSchmitt.ViMenor.Visible = False
GraficaSchmitt.ViMayor.Visible = False
GraficaSchmitt.Vi_Neg_1.Visible = False
GraficaSchmitt.Vi_Pos_1.Visible = False
GraficaSchmitt.V_1.Visible = False
GraficaSchmitt.V_2.Visible = False
GraficaSchmitt.Flecha1.Visible = False
GraficaSchmitt.Flecha2.Visible = False
GraficaSchmitt.Vo1.Visible = False
GraficaSchmitt.Vo2.Visible = False
GraficaSchmitt.Vo_1.Visible = False
GraficaSchmitt.Vo_2.Visible = False
GraficaSchmitt.V_3.Visible = False
GraficaSchmitt.V_4.Visible = False
GraficaSchmitt.Vo3.Visible = False
GraficaSchmitt.Vo_med_rms.Visible = False
GraficaSchmitt.flecha3.Visible = False
GraficaSchmitt.vmed_vrms.Visible = False
GraficaSchmitt.V_5.Visible = False

'Obtenemos los valores necesarios a partir del formulario y de la
hoja de cálculo "Gráfica Vo-t 2"'
GraficaSchmitt.a.Caption = RectMediaOnda.a.Value
GraficaSchmitt.b.Caption = RectMediaOnda.b.Value
GraficaSchmitt.b_2.Caption = RectMediaOnda.b.Value
GraficaSchmitt.Ganancia.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-t 2").Range("b5"), 2)
GraficaSchmitt.Ganancia_2.Caption = 0
GraficaSchmitt.b_3.Visible = False
GraficaSchmitt.sen_3.Visible = False
GraficaSchmitt.t_sen_3.Visible = False
GraficaSchmitt.V_9.Visible = False

'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-t"'
Set Grafico = Sheets("Gráfica Vo-t").ChartObjects(1).Chart

'Definimos dicho gráfico como un GIF'
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &
"temp.gif"
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"

'En la imagen de nuestro formulario mostramos el GIF capturado,
que representa la gráfica'
Image1.Picture = LoadPicture(NombreArchivo)

'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-t 2"'
Set Grafico = Sheets("Gráfica Vo-t 2").ChartObjects(1).Chart

'Definimos dicho gráfico como un GIF'
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &
"temp.gif"
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"

'En la imagen de nuestro formulario mostramos el GIF capturado,
que representa la gráfica'
Image2.Picture = LoadPicture(NombreArchivo)

'Para poder mostrar la gráfica Vo-t, antes tiene que estar visible la
pestaña del ejemplo 1 de un circuito rectificador de onda completa'
```




```
ElseIf RectOndaCompleta.Visible = True And  
RectOndaCompleta.Valor_Vo1_1.Caption <> Empty Then  
  
    'Establecemos qué valores queremos mostrar en la pestaña de la  
    gráfica'  
    GraficaSchmitt.ViMayor1.Visible = True  
    GraficaSchmitt.ViMenor1.Visible = True  
    GraficaSchmitt.Vi_Pos.Visible = True  
    GraficaSchmitt.Vi_Neg.Visible = True  
    GraficaSchmitt.V_6.Visible = True  
    GraficaSchmitt.V_7.Visible = True  
    GraficaSchmitt.V_8.Visible = True  
    GraficaSchmitt.V_9.Visible = True  
    GraficaSchmitt.Flecha4.Visible = True  
    GraficaSchmitt.Flecha5.Visible = True  
    GraficaSchmitt.Vo4.Visible = True  
    GraficaSchmitt.Vo5.Visible = True  
    GraficaSchmitt.Ganancia.Visible = True  
    GraficaSchmitt.Ganancia_2.Visible = True  
    GraficaSchmitt.sen_2.Visible = True  
    GraficaSchmitt.sen_3.Visible = True  
    GraficaSchmitt.b_2.Visible = True  
    GraficaSchmitt.b_3.Visible = True  
    GraficaSchmitt.t_sen_2.Visible = True  
    GraficaSchmitt.t_sen_3.Visible = True  
    GraficaSchmitt.ViMenor.Visible = False  
    GraficaSchmitt.ViMayor.Visible = False  
    GraficaSchmitt.Vi_Neg_1.Visible = False  
    GraficaSchmitt.Vi_Pos_1.Visible = False  
    GraficaSchmitt.V_1.Visible = False  
    GraficaSchmitt.V_2.Visible = False  
    GraficaSchmitt.Flecha1.Visible = False  
    GraficaSchmitt.Flecha2.Visible = False  
    GraficaSchmitt.Vo1.Visible = False  
    GraficaSchmitt.Vo2.Visible = False  
    GraficaSchmitt.Vo_1.Visible = False  
    GraficaSchmitt.Vo_2.Visible = False  
    GraficaSchmitt.V_3.Visible = False  
    GraficaSchmitt.V_4.Visible = False  
    GraficaSchmitt.Vo3.Visible = False  
    GraficaSchmitt.Vo_med_rms.Visible = False  
    GraficaSchmitt.flecha3.Visible = False  
    GraficaSchmitt.vmed_vrms.Visible = False  
    GraficaSchmitt.V_5.Visible = False  
  
    'Obtenemos los valores necesarios a partir del formulario y de la  
    hoja de cálculo "Gráfica Vo-t 2"  
    GraficaSchmitt.a.Caption = RectOndaCompleta.a1.Value  
    GraficaSchmitt.b.Caption = RectOndaCompleta.b1.Value  
    GraficaSchmitt.b_2.Caption = RectOndaCompleta.b1.Value  
    GraficaSchmitt.b_3.Caption = RectOndaCompleta.b1.Value  
    GraficaSchmitt.Ganancia.Caption =  
Abs(Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Grá  
fica Vo-t 2").Range("b5"), 2))  
    GraficaSchmitt.Ganancia_2.Caption =  
Abs(Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Grá  
fica Vo-t 2").Range("c5"), 2))  
  
    'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-t"  
    Set Grafico = Sheets("Gráfica Vo-t").ChartObjects(1).Chart
```



```
'Definimos dicho gráfico como un GIF'  
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &  
"temp.gif"  
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"  
  
'En la imagen de nuestro formulario mostramos el GIF capturado,  
que representa la gráfica'  
Image1.Picture = LoadPicture(NombreArchivo)  
  
'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-t 2"  
Set Grafico = Sheets("Gráfica Vo-t 2").ChartObjects(1).Chart  
  
'Definimos dicho gráfico como un GIF'  
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &  
"temp.gif"  
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"  
  
'En la imagen de nuestro formulario mostramos el GIF capturado,  
que representa la gráfica'  
Image2.Picture = LoadPicture(NombreArchivo)  
  
'Para poder mostrar la gráfica Vo-t, antes tiene que estar visible la  
pestaña del ejemplo 2 de un circuito rectificador de onda completa'  
ElseIf RectOndaCompleta.Visible = True And  
RectOndaCompleta.Valor_Vo_2.Caption <> Empty Then  
  
'Establecemos qué valores queremos mostrar en la pestaña de la  
gráfica'  
GraficaSchmitt.ViMayor1.Visible = True  
GraficaSchmitt.ViMenor1.Visible = True  
GraficaSchmitt.Vi_Pos.Visible = True  
GraficaSchmitt.Vi_Neg.Visible = True  
GraficaSchmitt.V_6.Visible = True  
GraficaSchmitt.V_7.Visible = True  
GraficaSchmitt.V_8.Visible = True  
GraficaSchmitt.V_9.Visible = True  
GraficaSchmitt.Flecha4.Visible = True  
GraficaSchmitt.Flecha5.Visible = True  
GraficaSchmitt.Vo4.Visible = True  
GraficaSchmitt.Vo5.Visible = True  
GraficaSchmitt.Ganancia.Visible = True  
GraficaSchmitt.Ganancia_2.Visible = True  
GraficaSchmitt.sen_2.Visible = True  
GraficaSchmitt.sen_3.Visible = True  
GraficaSchmitt.b_2.Visible = True  
GraficaSchmitt.b_3.Visible = True  
GraficaSchmitt.t_sen_2.Visible = True  
GraficaSchmitt.t_sen_3.Visible = True  
GraficaSchmitt.ViMenor.Visible = False  
GraficaSchmitt.ViMayor.Visible = False  
GraficaSchmitt.Vi_Neg_1.Visible = False  
GraficaSchmitt.Vi_Pos_1.Visible = False  
GraficaSchmitt.V_1.Visible = False  
GraficaSchmitt.V_2.Visible = False  
GraficaSchmitt.Flecha1.Visible = False  
GraficaSchmitt.Flecha2.Visible = False  
GraficaSchmitt.Vo1.Visible = False  
GraficaSchmitt.Vo2.Visible = False  
GraficaSchmitt.Vo_1.Visible = False  
GraficaSchmitt.Vo_2.Visible = False  
GraficaSchmitt.V_3.Visible = False
```



```
GraficaSchmitt.V_4.Visible = False
GraficaSchmitt.Vo3.Visible = False
GraficaSchmitt.Vo_med_rms.Visible = False
GraficaSchmitt.flecha3.Visible = False
GraficaSchmitt.vmed_vrms.Visible = False
GraficaSchmitt.V_5.Visible = False

'Obtenemos los valores necesarios a partir del formulario y de la
hoja de cálculo "Gráfica Vo-t 2"'
GraficaSchmitt.a.Caption = RectOndaCompleta.a2.Value
GraficaSchmitt.b.Caption = RectOndaCompleta.b2.Value
GraficaSchmitt.b_2.Caption = RectOndaCompleta.b2.Value
GraficaSchmitt.b_3.Caption = RectOndaCompleta.b2.Value
GraficaSchmitt.Ganancia.Caption =
Abs(Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Grá
fica Vo-t 2").Range("b5"), 2))
GraficaSchmitt.Ganancia_2.Caption =
Abs(Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Grá
fica Vo-t 2").Range("c5"), 2))

'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-t"'
Set Grafico = Sheets("Gráfica Vo-t").ChartObjects(1).Chart

'Definimos dicho gráfico como un GIF'
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &
"temp.gif"
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"

'En la imagen de nuestro formulario mostramos el GIF capturado,
que representa la gráfica'
Image1.Picture = LoadPicture(NombreArchivo)

'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-t 2"'
Set Grafico = Sheets("Gráfica Vo-t 2").ChartObjects(1).Chart

'Definimos dicho gráfico como un GIF'
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &
"temp.gif"
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"

'En la imagen de nuestro formulario mostramos el GIF capturado,
que representa la gráfica'
Image2.Picture = LoadPicture(NombreArchivo)

'Para poder mostrar la gráficas Vi-t Vo-t, antes tiene que estar
visible la pestaña del circuito para obtener el valor medio y el valor
eficaz'
ElseIf ValorMedioValorEficaz.Visible = True Then

'Establecemos qué valores queremos mostrar en la pestaña de la
gráfica'
GraficaSchmitt.Vo3.Visible = True
GraficaSchmitt.Vo_med_rms.Visible = True
GraficaSchmitt.flecha3.Visible = True
GraficaSchmitt.vmed_vrms.Visible = True
GraficaSchmitt.V_5.Visible = True
GraficaSchmitt.ViMenor.Visible = False
GraficaSchmitt.ViMayor.Visible = False
GraficaSchmitt.Vi_Neg_1.Visible = False
GraficaSchmitt.Vi_Pos_1.Visible = False
GraficaSchmitt.V_1.Visible = False
```



```
GraficaSchmitt.V_2.Visible = False
GraficaSchmitt.Flecha1.Visible = False
GraficaSchmitt.Flecha2.Visible = False
GraficaSchmitt.Vo1.Visible = False
GraficaSchmitt.Vo2.Visible = False
GraficaSchmitt.Vo_1.Visible = False
GraficaSchmitt.Vo_2.Visible = False
GraficaSchmitt.V_3.Visible = False
GraficaSchmitt.V_4.Visible = False
GraficaSchmitt.ViMayor1.Visible = False
GraficaSchmitt.ViMenor1.Visible = False
GraficaSchmitt.Vi_Pos.Visible = False
GraficaSchmitt.Vi_Neg.Visible = False
GraficaSchmitt.V_6.Visible = False
GraficaSchmitt.V_7.Visible = False
GraficaSchmitt.V_8.Visible = False
GraficaSchmitt.V_9.Visible = False
GraficaSchmitt.Flecha4.Visible = False
GraficaSchmitt.Flecha5.Visible = False
GraficaSchmitt.Vo4.Visible = False
GraficaSchmitt.Vo5.Visible = False
GraficaSchmitt.Ganancia.Visible = False
GraficaSchmitt.Ganancia_2.Visible = False
GraficaSchmitt.sen_2.Visible = False
GraficaSchmitt.sen_3.Visible = False
GraficaSchmitt.b_2.Visible = False
GraficaSchmitt.b_3.Visible = False
GraficaSchmitt.t_sen_2.Visible = False
GraficaSchmitt.t_sen_3.Visible = False

'Cada valor lo obtenemos a partir de la hoja de cálculo "Gráfica
CosSen" y el formulario'
GraficaSchmitt.a.Caption = ValorMedioValorEficaz.a.Value
GraficaSchmitt.b.Caption = ValorMedioValorEficaz.b.Value

If ValorMedioValorEficaz.Valor_Ganancia.Value = 1 Then

    GraficaSchmitt.Vo_med_rms.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("c3"), 2)
    GraficaSchmitt.vmed_vrms.Caption = "Vmed"

ElseIf ValorMedioValorEficaz.Valor_Ganancia.Value = 1.11 Then

    GraficaSchmitt.Vo_med_rms.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("c3"), 2)
    GraficaSchmitt.vmed_vrms.Caption = "Vrms"

Else

    GraficaSchmitt.Vo_med_rms.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("c3"), 2)
    GraficaSchmitt.vmed_vrms.Visible = False
    GraficaSchmitt.flecha3.Visible = False

End If

'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-t"
Set Grafico = Sheets("Gráfica Vo-t").ChartObjects(1).Chart
```



```
'Definimos dicho gráfico como un GIF'  
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &  
"temp.gif"  
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"  
  
'En la imagen de nuestro formulario mostramos el GIF capturado,  
que representa la gráfica'  
Image1.Picture = LoadPicture(NombreArchivo)  
  
'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica CosSen"  
Set Grafico = Sheets("Gráfica CosSen").ChartObjects(1).Chart  
  
'Definimos dicho gráfico como un GIF'  
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &  
"temp.gif"  
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"  
  
'En la imagen de nuestro formulario mostramos el GIF capturado,  
que representa la gráfica'  
Image2.Picture = LoadPicture(NombreArchivo)  
  
End If  
  
End Sub
```

7.2.2.23. GraficaVot

```
Private Sub UserForm_Initialize()  
  
'Para poder mostrar las gráficas Vi-t Vo-t, antes tiene que estar  
visible la pestaña del circuito integrador inversor'  
If IntegradorInversor.Visible = True And  
IntegradorInversor.Valor_Vo_1.Caption <> Empty Then  
  
    'Cada valor lo obtenemos a partir de las hojas de cálculo "Gráfica  
    CosSen" y Gráfica Vo-t, definidos en sus respectivas celdas'  
    GraficaVot.Ganancia.Caption =  
    Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
    Vo-t").Range("d4"), 2)  
    GraficaVot.n_2.Caption =  
    Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
    Vo-t").Range("d10"), 2)  
  
    GraficaVot.a.Caption =  
    Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
    CosSen").Range("ah14"), 2)  
    GraficaVot.n_1.Caption =  
    Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
    CosSen").Range("ah15"), 2)  
  
    'Si hay tensión inicial la definimos en nuestro formulario'  
    If IntegradorInversor.Valor_VoInicial_1 > 0 Then  
  
        GraficaVot.Vo_Inicial.Caption =  
        Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
        Vo-t").Range("c11"), 2)  
  
    End If  
  
End If  
  
End Sub
```



```
GraficaVot.signo.Caption = "+"

ElseIf IntegradorInversor.Valor_VoInicial_1 < 0 Then

    GraficaVot.Vo_Inicial.Caption =
Abs(Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Grá
fica Vo-t").Range("c11"), 2))
    GraficaVot.signo.Caption = "-"

ElseIf IntegradorInversor.Valor_VoInicial_1 = 0 Then

    GraficaVot.Vo_Inicial.Visible = False
    GraficaVot.Vo_Inicial.Caption = ""
    GraficaVot.signo.Caption = ""

End If

'Analizamos si alcanzamos la tensión de saturación'
If IntegradorInversor.Valor_Vi_1 < 0 Then

    GraficaVot.Vsat.Caption = "+Vsat"
    GraficaVot.Valor_Vsat.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-t").Range("c13"), 2)

ElseIf IntegradorInversor.Valor_Vi_1 > 0 Then

    GraficaVot.Vsat.Caption = "-Vsat"
    GraficaVot.Valor_Vsat.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-t").Range("c14"), 2)

End If

GraficaVot.t_final.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-t").Range("c16"), 2)
GraficaVot.t_final2.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-t").Range("c16"), 2)

'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica CosSen"
Set Grafico = Sheets("Gráfica CosSen").ChartObjects(1).Chart

'Definimos dicho gráfico como un GIF'
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &
"temp.gif"
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"

'En la imagen de nuestro formulario mostramos el GIF capturado,
que representa la gráfica'
Image1.Picture = LoadPicture(NombreArchivo)

'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-t"
Set Grafico = Sheets("Gráfica Vo-t").ChartObjects(1).Chart

'Definimos dicho gráfico como un GIF'
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &
"temp.gif"
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"
```



```
'En la imagen de nuestro formulario mostramos el GIF capturado,
que representa la gráfica'
Image2.Picture = LoadPicture(NombreArchivo)

'Para poder mostrar las gráficas Vi-t Vo-t, antes tiene que estar
visible la pestaña del circuito integrador no inversor'
ElseIf IntegradorNoInversor.Visible = True And
IntegradorNoInversor.Valor_Vo_1.Caption <> Empty Then

    'Cada valor lo obtenemos a partir de las hojas de cálculo "Gráfica
    CosSen" y Gráfica Vo-t, definidos en sus respectivas celdas'
    GraficaVot.Ganancia.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-t").Range("d5"), 2)
    GraficaVot.n_2.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-t").Range("d10"), 2)

    GraficaVot.a.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("ah14"), 2)
    GraficaVot.n_1.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("ah15"), 2)

    'Si hay tensión inicial la definimos en nuestro formulario'
    If IntegradorNoInversor.Valor_VoInicial_1 > 0 Then

        GraficaVot.Vo_Inicial.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-t").Range("c12"), 2)
        GraficaVot.signo.Caption = "+"

    ElseIf IntegradorNoInversor.Valor_VoInicial_1 < 0 Then

        GraficaVot.Vo_Inicial.Caption =
Abs(Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Grá
fica Vo-t").Range("c12"), 2))
        GraficaVot.signo.Caption = "-"

    ElseIf IntegradorNoInversor.Valor_VoInicial_1 = 0 Then

        GraficaVot.Vo_Inicial.Visible = False
        GraficaVot.Vo_Inicial.Caption = ""
        GraficaVot.signo.Caption = ""

    End If

    'Analizamos si alcanzamos la tensión de saturación'
    If IntegradorNoInversor.Valor_Vi_1 > 0 Then

        GraficaVot.Vsat.Caption = "+Vsat"
        GraficaVot.Valor_Vsat.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-t").Range("c13"), 2)

    ElseIf IntegradorNoInversor.Valor_Vi_1 < 0 Then

        GraficaVot.Vsat.Caption = "-Vsat"
```



```
GraficaVot.Valor_Vsat.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-t").Range("c14"), 2)  
  
End If  
  
GraficaVot.t_final.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-t").Range("c17"), 2)  
GraficaVot.t_final2.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-t").Range("c17"), 2)  
  
'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica CosSen"  
Set Grafico = Sheets("Gráfica CosSen").ChartObjects(1).Chart  
  
'Definimos dicho gráfico como un GIF'  
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &  
"temp.gif"  
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"  
  
'En la imagen de nuestro formulario mostramos el GIF capturado,  
que representa la gráfica'  
Image1.Picture = LoadPicture(NombreArchivo)  
  
'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-t"  
Set Grafico = Sheets("Gráfica Vo-t").ChartObjects(1).Chart  
  
'Definimos dicho gráfico como un GIF'  
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &  
"temp.gif"  
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"  
  
'En la imagen de nuestro formulario mostramos el GIF capturado,  
que representa la gráfica'  
Image2.Picture = LoadPicture(NombreArchivo)  
  
End If  
  
End Sub
```

7.2.2.24. GraficaVot_2

```
Private Sub UserForm_Initialize()  
  
'Para poder mostrar la gráfica Vo-t, antes tiene que estar visible la  
pestaña del circuito generador de onda cuadrada'  
If OndaCuadrada.Visible = True Then  
  
'Recogemos el valor del periodo de nuestro formulario'  
GraficaVot_2.t.Caption = OndaCuadrada.Periodo.Caption  
GraficaVot_2.f.Caption = OndaCuadrada.Frecuencia.Caption  
  
'Recogemos la gráfica de nuestra hoja de cálculo "Ondas"  
Set Grafico = Sheets("Ondas").ChartObjects(1).Chart
```




```
'Definimos dicho gráfico como un GIF'
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &
"temp.gif"
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"

'En la imagen de nuestro formulario mostramos el GIF capturado,
que representa la gráfica'
Image.Picture = LoadPicture(NombreArchivo)

'Para poder mostrar la gráfica Vo-t, antes tiene que estar visible la
pestaña del circuito generador de onda triangular'
ElseIf OndaTriangular.Visible = True Then

'Recogemos el valor del periodo de nuestro formulario'
GraficaVot_2.t.Caption = OndaTriangular.Periodo.Caption
GraficaVot_2.f.Caption = OndaTriangular.Frecuencia.Caption

'Recogemos la gráfica de nuestra hoja de cálculo "Ondas"'
Set Grafico = Sheets("Ondas").ChartObjects(1).Chart

'Definimos dicho gráfico como un GIF'
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &
"temp.gif"
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"

'En la imagen de nuestro formulario mostramos el GIF capturado,
que representa la gráfica'
Image.Picture = LoadPicture(NombreArchivo)

End If

End Sub
```

7.2.2.25. GraficaVoVi

```
Private Sub UserForm_Initialize()

'Para poder mostrar la gráfica Vo-Vi, antes tiene que estar visible la
pestaña del circuito inversor'
If Inversor.Visible = True Then

'Cada valor lo obtenemos a partir de la hoja de cálculo "Gráfica
Vo-Vi" definidos en sus respectivas celdas'
GraficaVoVi.Ganancia.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("c5"), 2)
GraficaVoVi.VsatPos.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("c10"), 2)
GraficaVoVi.VsatNeg.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("c11"), 2)
GraficaVoVi.Vi_VsatPos.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("c7"), 2)
```



```
GraficaVoVi.Vi_VsatPos2.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c7"), 2)  
GraficaVoVi.Vi_VsatNeg.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c8"), 2)  
GraficaVoVi.Vi_VsatNeg2.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c8"), 2)  
  
'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-Vi"  
Set Grafico = Sheets("Gráfica Vo-Vi").ChartObjects(1).Chart  
  
'Definimos dicho gráfico como un GIF'  
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &  
"temp.gif"  
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"  
  
'En la imagen de nuestro formulario mostramos el GIF capturado,  
que representa la gráfica'  
Image.Picture = LoadPicture(NombreArchivo)  
  
'Para poder mostrar la gráfica Vo-Vi, antes tiene que estar visible la  
pestaña del circuito no inversor'  
ElseIf NoInversor.Visible = True Then  
  
'Cada valor lo obtenemos a partir de la hoja de cálculo "Gráfica  
Vo-Vi" definidos en sus respectivas celdas'  
GraficaVoVi.Ganancia.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c5"), 2)  
GraficaVoVi.VsatPos.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c10"), 2)  
GraficaVoVi.VsatNeg.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c11"), 2)  
GraficaVoVi.Vi_VsatPos.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c13"), 2)  
GraficaVoVi.Vi_VsatNeg.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c14"), 2)  
GraficaVoVi.Vi_VsatPos2.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c13"), 2)  
GraficaVoVi.Vi_VsatNeg2.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c14"), 2)  
  
'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-Vi"  
Set Grafico = Sheets("Gráfica Vo-Vi").ChartObjects(1).Chart  
  
'Definimos dicho gráfico como un GIF'  
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &  
"temp.gif"  
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"  
  
'En la imagen de nuestro formulario mostramos el GIF capturado,  
que representa la gráfica'
```



```
Image.Picture = LoadPicture(NombreArchivo)

End If

End Sub
```

7.2.2.26. GraficaVoVi_2

```
Private Sub UserForm_Initialize ()

'Para poder mostrar la gráfica Vo-Vi, antes tiene que estar visible la
pestaña del ejemplo 1 del circuito limitador con diodo normal'
If LimitadorDiodoNormal.Visible = True And
LimitadorDiodoNormal.Valor_Vo_1.Caption <> Empty Then

    'Establecemos qué valores queremos mostrar en la pestaña de la
gráfica'
    GraficaVoVi_2.Vi_1.Visible = True
    GraficaVoVi_2.Vi_2.Visible = True
    GraficaVoVi_2.Vi_Pos.Visible = True
    GraficaVoVi_2.Vi_Neg.Visible = True
    GraficaVoVi_2.Vo.Visible = True
    GraficaVoVi_2.Ganancia.Visible = True
    GraficaVoVi_2.Mult.Visible = True
    GraficaVoVi_2.Vi_Ec.Visible = True
    GraficaVoVi_2.Vref.Visible = False
    GraficaVoVi_2.Mult.Visible = False
    GraficaVoVi_2.Vi_Resta.Visible = False
    GraficaVoVi_2.ProductoVref.Visible = False
    GraficaVoVi_2.ProductoVi.Visible = False
    GraficaVoVi_2.Suma.Visible = False
    GraficaVoVi_2.Vi_Mult.Visible = False
    GraficaVoVi_2.Vref.Visible = False

    'Cada valor lo obtenemos a partir de la hoja de cálculo "Gráfica
Vo-Vi" definidos en sus respectivas celdas'
    GraficaVoVi_2.Ganancia.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("c5"), 2)
    GraficaVoVi_2.Vi_Pos.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("c17"), 2)
    GraficaVoVi_2.Vi_Neg.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("c16"), 2)
    GraficaVoVi_2.Vi_Pos_1.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("c17"), 2)
    GraficaVoVi_2.Vi_Neg_1.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("c16"), 2)
    GraficaVoVi_2.Vo_1.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("c19"), 2)
    GraficaVoVi_2.Vo_2.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("c20"), 2)
```



```
'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-Vi"'
Set Grafico = Sheets("Gráfica Vo-Vi").ChartObjects(1).Chart

'Definimos dicho gráfico como un GIF'
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &
"temp.gif"
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"

'En la imagen de nuestro formulario mostramos el GIF capturado,
que representa la gráfica'
Image.Picture = LoadPicture(NombreArchivo)

'Para poder mostrar la gráfica Vo-Vi, antes tiene que estar visible la
pestaña del ejemplo 2 del circuito limitador con diodo normal'
ElseIf LimitadorDiodoNormal.Visible = True And
LimitadorDiodoNormal.Valor_Vo_2.Caption <> Empty Then

'Establecemos qué valores queremos mostrar en la pestaña de la
gráfica'
GraficaVoVi_2.Vi_1.Visible = True
GraficaVoVi_2.Vi_2.Visible = True
GraficaVoVi_2.Vi_Pos.Visible = True
GraficaVoVi_2.Vi_Neg.Visible = True
GraficaVoVi_2.Vo.Visible = True
GraficaVoVi_2.Ganancia.Visible = True
GraficaVoVi_2.Mult.Visible = True
GraficaVoVi_2.Vi_Ec.Visible = True
GraficaVoVi_2.Vref.Visible = False
GraficaVoVi_2.Mult.Visible = False
GraficaVoVi_2.Vi_Resta.Visible = False
GraficaVoVi_2.ProductoVref.Visible = False
GraficaVoVi_2.ProductoVi.Visible = False
GraficaVoVi_2.Suma.Visible = False
GraficaVoVi_2.Vi_Mult.Visible = False
GraficaVoVi_2.Vref.Visible = False

'Cada valor lo obtenemos a partir de la hoja de cálculo "Gráfica
Vo-Vi" definidos en sus respectivas celdas'
GraficaVoVi_2.Ganancia.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("c5"), 2)
GraficaVoVi_2.Vi_Pos.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("c25"), 2)
GraficaVoVi_2.Vi_Neg.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("c24"), 2)
GraficaVoVi_2.Vi_Pos_1.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("c25"), 2)
GraficaVoVi_2.Vi_Neg_1.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("c24"), 2)
GraficaVoVi_2.Vo_1.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("c21"), 2)
GraficaVoVi_2.Vo_2.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("c22"), 2)
```



```
'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-Vi"  
Set Grafico = Sheets("Gráfica Vo-Vi").ChartObjects(1).Chart  
  
'Definimos dicho gráfico como un GIF'  
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &  
"temp.gif"  
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"  
  
'En la imagen de nuestro formulario mostramos el GIF capturado,  
que representa la gráfica'  
Image.Picture = LoadPicture(NombreArchivo)  
  
'Para poder mostrar la gráfica Vo-Vi, antes tiene que estar visible la  
pestaña del ejemplo 1 del circuito limitador con diodo normal y diodo  
zener en serie'  
ElseIf LimitadorDiodoNormalDiodoZener.Visible = True And  
LimitadorDiodoNormalDiodoZener.Valor_Vo_1.Caption <> Empty Then  
  
'Establecemos qué valores queremos mostrar en la pestaña de la  
gráfica'  
GraficaVoVi_2.Vi_1.Visible = True  
GraficaVoVi_2.Vi_2.Visible = True  
GraficaVoVi_2.Vi_Pos.Visible = True  
GraficaVoVi_2.Vi_Neg.Visible = True  
GraficaVoVi_2.Vo.Visible = True  
GraficaVoVi_2.Ganancia.Visible = True  
GraficaVoVi_2.Mult.Visible = True  
GraficaVoVi_2.Vi_Ec.Visible = True  
GraficaVoVi_2.Vref.Visible = False  
GraficaVoVi_2.Mult.Visible = False  
GraficaVoVi_2.Vi_Resta.Visible = False  
GraficaVoVi_2.ProductoVref.Visible = False  
GraficaVoVi_2.ProductoVi.Visible = False  
GraficaVoVi_2.Suma.Visible = False  
GraficaVoVi_2.Vi_Mult.Visible = False  
GraficaVoVi_2.Vref.Visible = False  
  
'Cada valor lo obtenemos a partir de la hoja de cálculo "Gráfica  
Vo-Vi" definidos en sus respectivas celdas'  
GraficaVoVi_2.Ganancia.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c5"), 2)  
GraficaVoVi_2.Vi_Pos.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c28"), 2)  
GraficaVoVi_2.Vi_Neg.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c27"), 2)  
GraficaVoVi_2.Vi_Pos_1.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c28"), 2)  
GraficaVoVi_2.Vi_Neg_1.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c27"), 2)  
GraficaVoVi_2.Vo_1.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c30"), 2)  
GraficaVoVi_2.Vo_2.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c31"), 2)
```



```
'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-Vi"  
Set Grafico = Sheets("Gráfica Vo-Vi").ChartObjects(1).Chart  
  
'Definimos dicho gráfico como un GIF'  
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &  
"temp.gif"  
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"  
  
'En la imagen de nuestro formulario mostramos el GIF capturado,  
que representa la gráfica'  
Image.Picture = LoadPicture(NombreArchivo)  
  
'Para poder mostrar la gráfica Vo-Vi, antes tiene que estar visible la  
pestaña del ejemplo 2 del circuito limitador con diodo normal y diodo  
zener en serie'  
ElseIf LimitadorDiodoNormalDiodoZener.Visible = True And  
LimitadorDiodoNormalDiodoZener.Valor_Vo_2.Caption <> Empty Then  
  
'Establecemos qué valores queremos mostrar en la pestaña de la  
gráfica'  
GraficaVoVi_2.Vi_1.Visible = True  
GraficaVoVi_2.Vi_2.Visible = True  
GraficaVoVi_2.Vi_Pos.Visible = True  
GraficaVoVi_2.Vi_Neg.Visible = True  
GraficaVoVi_2.Vo.Visible = True  
GraficaVoVi_2.Ganancia.Visible = True  
GraficaVoVi_2.Mult.Visible = True  
GraficaVoVi_2.Vi_Ec.Visible = True  
GraficaVoVi_2.Vref.Visible = False  
GraficaVoVi_2.Mult.Visible = False  
GraficaVoVi_2.Vi_Resta.Visible = False  
GraficaVoVi_2.ProductoVref.Visible = False  
GraficaVoVi_2.ProductoVi.Visible = False  
GraficaVoVi_2.Suma.Visible = False  
GraficaVoVi_2.Vi_Mult.Visible = False  
GraficaVoVi_2.Vref.Visible = False  
  
'Cada valor lo obtenemos a partir de la hoja de cálculo "Gráfica  
Vo-Vi" definidos en sus respectivas celdas'  
GraficaVoVi_2.Ganancia.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c5"), 2)  
GraficaVoVi_2.Vi_Pos.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c36"), 2)  
GraficaVoVi_2.Vi_Neg.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c35"), 2)  
GraficaVoVi_2.Vi_Pos_1.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c36"), 2)  
GraficaVoVi_2.Vi_Neg_1.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c35"), 2)  
GraficaVoVi_2.Vo_1.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c32"), 2)  
GraficaVoVi_2.Vo_2.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c33"), 2)
```



```
'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-Vi"  
Set Grafico = Sheets("Gráfica Vo-Vi").ChartObjects(1).Chart  
  
'Definimos dicho gráfico como un GIF'  
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &  
"temp.gif"  
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"  
  
'En la imagen de nuestro formulario mostramos el GIF capturado,  
que representa la gráfica'  
Image.Picture = LoadPicture(NombreArchivo)  
  
'Para poder mostrar la gráfica Vo-Vi, antes tiene que estar visible la  
pestaña del circuito limitador con dos diodos zener en serie'  
ElseIf LimitadorDosZener.Visible = True Then  
  
'Establecemos qué valores queremos mostrar en la pestaña de la  
gráfica'  
GraficaVoVi_2.Vi_1.Visible = True  
GraficaVoVi_2.Vi_2.Visible = True  
GraficaVoVi_2.Vi_Pos.Visible = True  
GraficaVoVi_2.Vi_Neg.Visible = True  
GraficaVoVi_2.Vo.Visible = True  
GraficaVoVi_2.Ganancia.Visible = True  
GraficaVoVi_2.Mult.Visible = True  
GraficaVoVi_2.Vi_Ec.Visible = True  
GraficaVoVi_2.Vref.Visible = False  
GraficaVoVi_2.Mult.Visible = False  
GraficaVoVi_2.Vi_Resta.Visible = False  
GraficaVoVi_2.ProductoVref.Visible = False  
GraficaVoVi_2.ProductoVi.Visible = False  
GraficaVoVi_2.Suma.Visible = False  
GraficaVoVi_2.Vi_Mult.Visible = False  
GraficaVoVi_2.Vref.Visible = False  
  
'Cada valor lo obtenemos a partir de la hoja de cálculo "Gráfica  
Vo-Vi" definidos en sus respectivas celdas'  
GraficaVoVi_2.Ganancia.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c5"), 2)  
GraficaVoVi_2.Vi_Pos.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c39"), 2)  
GraficaVoVi_2.Vi_Neg.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c38"), 2)  
GraficaVoVi_2.Vi_Pos_1.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c39"), 2)  
GraficaVoVi_2.Vi_Neg_1.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c38"), 2)  
GraficaVoVi_2.Vo_1.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c41"), 2)  
GraficaVoVi_2.Vo_2.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c42"), 2)  
  
'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-Vi"  
Set Grafico = Sheets("Gráfica Vo-Vi").ChartObjects(1).Chart
```



```
'Definimos dicho gráfico como un GIF'  
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &  
"temp.gif"  
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"  
  
'En la imagen de nuestro formulario mostramos el GIF capturado,  
que representa la gráfica'  
Image.Picture = LoadPicture(NombreArchivo)  
  
'Para poder mostrar la gráfica Vo-Vi, antes tiene que estar visible la  
pestaña del circuito comparador simple'  
ElseIf CompSimple.Visible = True Then  
  
'Establecemos qué valores queremos mostrar en la pestaña de la  
gráfica'  
GraficaVoVi_2.Vi_1.Visible = False  
GraficaVoVi_2.Vi_2.Visible = False  
GraficaVoVi_2.Vi_Pos.Visible = False  
GraficaVoVi_2.Vi_Neg.Visible = False  
GraficaVoVi_2.ProductoVref.Visible = False  
GraficaVoVi_2.ProductoVi.Visible = False  
GraficaVoVi_2.Suma.Visible = False  
GraficaVoVi_2.Vi_Mult.Visible = False  
GraficaVoVi_2.Vo.Visible = False  
GraficaVoVi_2.Ganancia.Visible = False  
GraficaVoVi_2.Mult.Visible = False  
GraficaVoVi_2.Vref.Visible = False  
GraficaVoVi_2.Vi_Resta.Visible = False  
GraficaVoVi_2.Vi_Ec.Visible = False  
  
'Cada valor lo obtenemos a partir de la hoja de cálculo "Gráfica  
Vo-Vi" definidos en sus respectivas celdas'  
GraficaVoVi_2.Ganancia.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c5"), 2)  
GraficaVoVi_2.Vref.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("s6"), 2)  
GraficaVoVi_2.Vi_Pos_1.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("s6"), 2)  
GraficaVoVi_2.Vi_Neg_1.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("s6"), 2)  
GraficaVoVi_2.Vo_1.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("s9"), 2)  
GraficaVoVi_2.Vo_2.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("s10"), 2)  
  
'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-Vi"  
Set Grafico = Sheets("Gráfica Vo-Vi").ChartObjects(1).Chart  
  
'Definimos dicho gráfico como un GIF'  
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &  
"temp.gif"  
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"
```




```
'En la imagen de nuestro formulario mostramos el GIF capturado,  
que representa la gráfica'  
Image.Picture = LoadPicture(NombreArchivo)  
  
'Para poder mostrar la gráfica Vo-Vi, antes tiene que estar visible la  
pestaña del circuito comparador con salida saturación-saturación'  
ElseIf CompSatSat.Visible = True Then  
  
  'Establecemos qué valores queremos mostrar en la pestaña de la  
  gráfica'  
  GraficaVoVi_2.Vi_1.Visible = False  
  GraficaVoVi_2.Vi_2.Visible = False  
  GraficaVoVi_2.Vi_Pos.Visible = False  
  GraficaVoVi_2.Vi_Neg.Visible = False  
  GraficaVoVi_2.ProductoVref.Visible = False  
  GraficaVoVi_2.ProductoVi.Visible = False  
  GraficaVoVi_2.Suma.Visible = False  
  GraficaVoVi_2.Vi_Mult.Visible = False  
  GraficaVoVi_2.Vo.Visible = False  
  GraficaVoVi_2.Ganancia.Visible = False  
  GraficaVoVi_2.Mult.Visible = False  
  GraficaVoVi_2.Vref.Visible = False  
  GraficaVoVi_2.Vi_Resta.Visible = False  
  GraficaVoVi_2.Vi_Ec.Visible = False  
  
  'Cada valor lo obtenemos a partir de la hoja de cálculo "Gráfica  
  Vo-Vi" definidos en sus respectivas celdas'  
  GraficaVoVi_2.Ganancia.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("c5"), 2)  
  GraficaVoVi_2.ProductoVref.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("q3"), 2)  
  GraficaVoVi_2.ProductoVi.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("r3"), 2)  
  GraficaVoVi_2.Vi_Pos_1.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("s6"), 2)  
  GraficaVoVi_2.Vi_Neg_1.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("s6"), 2)  
  GraficaVoVi_2.Vo_1.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("s9"), 2)  
  GraficaVoVi_2.Vo_2.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("s10"), 2)  
  
  'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-Vi"  
  Set Grafico = Sheets("Gráfica Vo-Vi").ChartObjects(1).Chart  
  
  'Definimos dicho gráfico como un GIF'  
  NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &  
"temp.gif"  
  Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"  
  
  'En la imagen de nuestro formulario mostramos el GIF capturado,  
  que representa la gráfica'  
  Image.Picture = LoadPicture(NombreArchivo)
```



'Para poder mostrar la gráfica Vo-Vi, antes tiene que estar visible la pestaña del ejemplo 1 del circuito comparador con salida cero-saturación'

```
ElseIf CompCeroSat.Visible = True And CompCeroSat.Valor_Vo_1.Caption  
<> Empty Then
```

'Establecemos que valores queremos mostrar en la pestaña de la gráfica'

```
GraficaVoVi_2.Vi_1.Visible = False  
GraficaVoVi_2.Vi_2.Visible = False  
GraficaVoVi_2.Vi_Pos.Visible = False  
GraficaVoVi_2.Vi_Neg.Visible = False  
GraficaVoVi_2.Vo.Visible = False  
GraficaVoVi_2.Ganancia.Visible = False  
GraficaVoVi_2.Mult.Visible = False  
GraficaVoVi_2.ProductoVref.Visible = False  
GraficaVoVi_2.ProductoVi.Visible = False  
GraficaVoVi_2.Suma.Visible = False  
GraficaVoVi_2.Vi_Resta.Visible = False  
GraficaVoVi_2.Vi_Mult.Visible = False  
GraficaVoVi_2.Vi_Ec.Visible = False  
GraficaVoVi_2.Vref.Visible = False
```

'Cada valor lo obtenemos a partir de la hoja de cálculo "Gráfica Vo-Vi" definidos en sus respectivas celdas'

```
GraficaVoVi_2.Vi_Pos_1.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("s6"), 2)
```

```
GraficaVoVi_2.Vi_Neg_1.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("s6"), 2)
```

```
GraficaVoVi_2.Vo_1.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("s9"), 2)
```

```
GraficaVoVi_2.Vo_2.Caption =  
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica  
Vo-Vi").Range("s10"), 2)
```

'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-Vi"
Set Grafico = Sheets("Gráfica Vo-Vi").ChartObjects(1).Chart

'Definimos dicho gráfico como un GIF'

```
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &  
"temp.gif"
```

```
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"
```

'En la imagen de nuestro formulario mostramos el GIF capturado,
que representa la gráfica'

```
Image.Picture = LoadPicture(NombreArchivo)
```

'Para poder mostrar la gráfica Vo-Vi, antes tiene que estar visible la pestaña del ejemplo 2 del circuito comparador con salida cero-saturación'

```
ElseIf CompCeroSat.Visible = True And CompCeroSat.Valor_Vo_2.Caption  
<> Empty Then
```

'Establecemos qué valores queremos mostrar en la pestaña de la gráfica'

```
GraficaVoVi_2.Vi_1.Visible = False  
GraficaVoVi_2.Vi_2.Visible = False  
GraficaVoVi_2.Vi_Pos.Visible = False
```



```
GraficaVoVi_2.Vi_Neg.Visible = False
GraficaVoVi_2.Vo.Visible = False
GraficaVoVi_2.Ganancia.Visible = False
GraficaVoVi_2.Mult.Visible = False
GraficaVoVi_2.ProductoVref.Visible = False
GraficaVoVi_2.ProductoVi.Visible = False
GraficaVoVi_2.Suma.Visible = False
GraficaVoVi_2.Vi_Resta.Visible = False
GraficaVoVi_2.Vi_Mult.Visible = False
GraficaVoVi_2.Vi_Ec.Visible = False
GraficaVoVi_2.Vref.Visible = False

'Cada valor lo obtenemos a partir de la hoja de cálculo "Gráfica
Vo-Vi" definidos en sus respectivas celdas'
GraficaVoVi_2.Vi_Pos_1.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("s6"), 2)
GraficaVoVi_2.Vi_Neg_1.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("s6"), 2)
GraficaVoVi_2.Vo_1.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("s9"), 2)
GraficaVoVi_2.Vo_2.Caption =
Application.WorksheetFunction.Round(ActiveWorkbook.Worksheets("Gráfica
Vo-Vi").Range("s10"), 2)

'Recogemos la gráfica de nuestra hoja de cálculo "Gráfica Vo-Vi"
Set Grafico = Sheets("Gráfica Vo-Vi").ChartObjects(1).Chart

'Definimos dicho gráfico como un GIF'
NombreArchivo = ThisWorkbook.Path & Application.PathSeparator &
"temp.gif"
Grafico.Export Filename:=NombreArchivo, FilterName:="GIF"

'En la imagen de nuestro formulario mostramos el GIF capturado,
que representa la gráfica'
Image.Picture = LoadPicture(NombreArchivo)

End If

End Sub
```

7.2.2.27. Comparador con histéresis (Histeresis1)

```
Private Sub Teoria_Click() 'Botón "Teoría"'
ModHisteresis.MostrarTeoriaHisteresis
End Sub

Private Sub Calcular_Click() 'Botón "Calcular"'
ModHisteresis.CalcularValoresHisteresis
```



```
End Sub

Private Sub Restaurar_Click() 'Botón "Restaurar valores"

ModHisteresis.RestaurarValoresHisteresis

End Sub

Private Sub Informacion_Click() 'Botón "Información"

ModHisteresis.MostrarInformacionHisteresis

End Sub

Private Sub Vo_Vi_Click() 'Botón "Representación de Vo-Vi"

'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If Histeresis1.Valor_Vo_1.Caption = Empty Then

MsgBox "Debe obtener primero un valor de Vo para poder calcular la
gráfica Vo-Vi", vbExclamation + vbOKOnly, "AVISO"

Else

'Cargamos la gráfica'
Load GraficaHisteresis

'Refrescamos la gráfica'
GraficaHisteresis.Repaint

'Mostramos la gráfica'
GraficaHisteresis.Show

End If

End Sub

Private Sub UserForm_terminate()

'Vaciamos las celdas de la hoja de cálculo "Gráfica Vo-Vi"
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s14") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("v14") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s23") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("v23") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("w17") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s16") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t16") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s17") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t17") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s19") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t19") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s20") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t20") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s21") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t21") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s30") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("v30") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("w17") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s28") = Empty
```



```
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t28") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s26") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t26") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s27") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t27") = Empty
```

```
End Sub
```

7.2.2.28. Comparador con histéresis y limitación a la salida (Histeresis2)

```
Private Sub Teoria1_Click() 'Botón "Teoría" del ejemplo 1'
```

```
ModHisteresis.MostrarTeoriaHisteresis1
```

```
End Sub
```

```
Private Sub Teoria2_Click() 'Botón "Teoría" del ejemplo 2'
```

```
ModHisteresis.MostrarTeoriaHisteresis2
```

```
End Sub
```

```
Private Sub Calcular1_Click() 'Botón "Calcular" del ejemplo 1'
```

```
ModHisteresis.CalcularValoresHisteresis1
```

```
End Sub
```

```
Private Sub Calcular2_Click() 'Botón "Calcular" del ejemplo 2'
```

```
ModHisteresis.CalcularValoresHisteresis2
```

```
End Sub
```

```
Private Sub Restaurar1_Click() 'Botón "Restaurar valores" del  
ejemplo 1'
```

```
ModHisteresis.RestaurarValoresHisteresis1
```

```
End Sub
```

```
Private Sub Restaurar2_Click() 'Botón "Restaurar valores" del  
ejemplo 2'
```

```
ModHisteresis.RestaurarValoresHisteresis2
```

```
End Sub
```

```
Private Sub Informacion1_Click() 'Botón "Información" del ejemplo 1'
```

```
ModHisteresis.MostrarInformacionHisteresis1
```



```
End Sub

Private Sub Informacion2_Click() 'Botón "Información" del ejemplo 2'
ModHisteresis.MostrarInformacionHisteresis2
End Sub

Private Sub Vo_Vi_1_Click() 'Botón "Representación Vo-Vi" del
ejemplo 1'

'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If Histeresis2.Valor_Vo_1_1.Caption = Empty Then

    MsgBox "Debe obtener primero un valor de Vo para poder calcular la
gráfica Vo-Vi", vbExclamation + vbOKOnly, "AVISO"

Else

    'Cargamos la gráfica'
    Load GraficaHisteresis

    'Refrescamos la gráfica'
    GraficaHisteresis.Repaint

    'Mostramos la gráfica'
    GraficaHisteresis.Show

End If

End Sub

Private Sub Vo_Vi_2_Click() 'Botón "Representación Vo-Vi" del
ejemplo 2'

'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If Histeresis2.Valor_Vo_2_2.Caption = Empty Then

    MsgBox "Debe obtener primero un valor de Vo para poder calcular la
gráfica Vo-Vi", vbExclamation + vbOKOnly, "AVISO"

Else

    'Cargamos la gráfica'
    Load GraficaHisteresis

    'Refrescamos la gráfica'
    GraficaHisteresis.Repaint

    'Mostramos la gráfica'
    GraficaHisteresis.Show

End If

End Sub

Private Sub UserForm_terminate()
```



```
'Vaciamos las celdas de la hoja de cálculo "Gráfica Vo-Vi"'
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s14") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("v14") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s23") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("v23") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("w17") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s16") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t16") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s17") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t17") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s19") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t19") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s20") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t20") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s21") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t21") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s30") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("v30") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s28") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t28") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s26") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t26") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s27") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t27") = Empty
```

End Sub

7.2.2.29. Integrador inversor (IntegradorInversor)

```
Private Sub Teoria1_Click() 'Botón "Teoría" del tipo 1'
```

```
ModIntInv.MostrarTeoriaIntInv
```

End Sub

```
Private Sub Teoria2_Click() 'Botón "Teoría" del tipo 2'
```

```
ModIntInv.MostrarTeoriaIntInv
```

End Sub

```
Private Sub Calcular1_Click() 'Botón "Calcular" del tipo 1'
```

```
ModIntInv.CalcularValores1
```

End Sub

```
Private Sub Calcular2_Click() 'Botón "Calcular" del tipo 2'
```

```
ModIntInv.CalcularValores2
```

End Sub



```
Private Sub Restaurar1_Click() 'Botón "Restaurar valores" del tipo
1'

ModIntInv.RestaurarValores1

End Sub

Private Sub Restaurar2_Click() 'Botón "Restaurar valores" del tipo
2'

ModIntInv.RestaurarValores2

End Sub

Private Sub Informacion1_Click() 'Botón "Información" del tipo 1'

ModIntInv.MostrarInformacion1

End Sub

Private Sub Informacion2_Click() 'Botón "Información" del tipo 2'

ModIntInv.MostrarInformacion2

End Sub

Private Sub UserForm_Activate() 'Definimos para el tipo 2 si la señal
de entrada es un seno o un coseno'

Sen_Cos.AddItem "sen"
Sen_Cos.AddItem "cos"

End Sub

Private Sub ViVo_t_1_Click() 'Botón "Representación de Vi y Vo
frente a t" del tipo 1'

'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If IntegradorInversor.Valor_Vo_1.Caption = Empty Then

MsgBox "Debe obtener primero un valor de Vo para poder calcular
las gráficas Vi-t Vo-t", vbExclamation + vbOKOnly, "AVISO"

Else

'Cargamos la gráfica'
Load GraficaVot

'Refrescamos la gráfica'
GraficaVot.Repaint

'Mostramos la gráfica'
GraficaVot.Show

End If

End Sub

Private Sub ViVo_t_2_Click() 'Botón "Representación de Vi y Vo
frente a t" del tipo 2'
```




```
'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If IntegradorInversor.Valor_Vo_2.Caption = Empty Then
```

```
MsgBox "Debe obtener primero un valor de Vo para poder calcular
las gráficas Vi-t Vo-t", vbExclamation + vbOKOnly, "AVISO"
```

```
Else
```

```
  'Cargamos la gráfica'
  Load GraficaCosSen
```

```
  'Refrescamos la gráfica'
  GraficaCosSen.Repaint
```

```
  'Mostramos la gráfica'
  GraficaCosSen.Show
```

```
End If
```

```
End Sub
```

```
Private Sub UserForm_terminate()
```

```
  'Vaciamos las celdas de la hoja de cálculo "Gráfica Vo-t"
```

```
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c7") = Empty
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c8") = Empty
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c9") = Empty
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c10") = Empty
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c11") = Empty
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c13") = Empty
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c14") = Empty
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c16") = Empty
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("d16") = Empty
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c15") = Empty
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("d15") = Empty
```

```
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("u5") = Empty
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v5") = Empty
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("w5") = Empty
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("x5") = Empty
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("s8:s728") = Empty
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("y8:y728") = Empty
  ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("z8:z728") = Empty
```

```
  'Vaciamos las celdas de la hoja de cálculo "Gráfica CosSen"
```

```
  ActiveWorkbook.Worksheets("Gráfica CosSen").Range("p5") = Empty
  ActiveWorkbook.Worksheets("Gráfica CosSen").Range("q5") = Empty
  ActiveWorkbook.Worksheets("Gráfica CosSen").Range("p6") = Empty
  ActiveWorkbook.Worksheets("Gráfica CosSen").Range("q6") = Empty
  ActiveWorkbook.Worksheets("Gráfica CosSen").Range("o6") = Empty
  ActiveWorkbook.Worksheets("Gráfica CosSen").Range("o8") = Empty
  ActiveWorkbook.Worksheets("Gráfica CosSen").Range("o9") = Empty
  ActiveWorkbook.Worksheets("Gráfica CosSen").Range("p8") = Empty
  ActiveWorkbook.Worksheets("Gráfica CosSen").Range("p9") = Empty
  ActiveWorkbook.Worksheets("Gráfica CosSen").Range("q8") = Empty
  ActiveWorkbook.Worksheets("Gráfica CosSen").Range("q9") = Empty
  ActiveWorkbook.Worksheets("Gráfica CosSen").Range("r8") = Empty
  ActiveWorkbook.Worksheets("Gráfica CosSen").Range("r9") = Empty
  ActiveWorkbook.Worksheets("Gráfica CosSen").Range("s8") = Empty
```



```
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("s9") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n2") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n3") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("a8:a728") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("e8:e728") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("f8:f728") = Empty
```

```
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah14") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah15") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag17") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah17") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag16") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah16") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag19") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah19") = Empty
```

End Sub

7.2.2.30. Integrador no inversor (IntegradorNoInversor)

```
Private Sub Teoria1_Click() 'Botón "Teoría" del tipo 1'
```

```
ModIntNoInv.MostrarTeoriaIntNoInv
```

End Sub

```
Private Sub Teoria2_Click() 'Botón "Teoría" del tipo 2'
```

```
ModIntNoInv.MostrarTeoriaIntNoInv
```

End Sub

```
Private Sub Calcular1_Click() 'Botón "Calcular" del tipo 1'
```

```
ModIntNoInv.CalcularValores1
```

End Sub

```
Private Sub Calcular2_Click() 'Botón "Calcular" del tipo 2'
```

```
ModIntNoInv.CalcularValores2
```

End Sub

```
Private Sub Restaurar1_Click() 'Botón "Restaurar valores" del tipo  
1'
```

```
ModIntNoInv.RestaurarValores1
```

End Sub

```
Private Sub Restaurar2_Click() 'Botón "Restaurar valores" del tipo  
2'
```



```
ModIntNoInv.RestaurarValores2

End Sub

Private Sub Informacion1_Click() 'Botón "Información" del tipo 1'
ModIntNoInv.MostrarInformacion1
End Sub

Private Sub Informacion2_Click() 'Botón "Información" del tipo 2'
ModIntNoInv.MostrarInformacion2
End Sub

Private Sub UserForm_Activate() 'Definimos para el tipo 2 si la se de
entrada es un seno o un coseno'

Sen_Cos.AddItem "sen"
Sen_Cos.AddItem "cos"

End Sub

Private Sub ViVo_t_1_Click() 'Botón "Representación de Vi y Vo
frente a t" del tipo 1'

'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If IntegradorNoInversor.Valor_Vo_1.Caption = Empty Then

    MsgBox "Debe obtener primero un valor de Vo para poder calcular
las gráficas Vi-t Vo-t", vbExclamation + vbOKOnly, "AVISO"

Else

    'Cargamos la gráfica'
    Load GraficaVot

    'Refrescamos la gráfica'
    GraficaVot.Repaint

    'Mostramos la gráfica'
    GraficaVot.Show

End If

End Sub

Private Sub ViVo_t_2_Click() 'Botón "Representación de Vi y Vo
frente a t" del tipo 2'

'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If IntegradorNoInversor.Valor_Vo_2 = Empty Then

    MsgBox "Debe obtener primero un valor de Vo para poder calcular
las gráficas Vi-t Vo-t", vbExclamation + vbOKOnly, "AVISO"

Else
```



```
'Cargamos la gráfica'  
Load GraficaCosSen  
  
'Refrescamos la gráfica'  
GraficaCosSen.Repaint  
  
'Mostramos la gráfica'  
GraficaCosSen.Show  
  
End If  
  
End Sub  
  
Private Sub UserForm_terminate()  
  
'Vaciamos las celdas de la hoja de cálculo "Gráfica Vo-t"  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c7") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c8") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c9") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c10") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c12") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c13") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c14") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c17") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("d17") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c15") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("d15") = Empty  
  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("y5") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("z5") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("aa5") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("ab5") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("s8:s728") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("aa8:aa728") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("ab8:ab728") = Empty  
  
'Vaciamos las celdas de la hoja de cálculo "Gráfica CosSen"  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("p5") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("q5") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("p6") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("q6") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("o6") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("o10") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("o11") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("p10") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("p11") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("q10") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("q11") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("r10") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("r11") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("s10") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("s11") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n4") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n5") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("a8:a728") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("i8:i728") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("j8:j728") = Empty  
  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah14") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah15") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag17") = Empty
```



```
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah17") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag16") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah16") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag19") = Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah19") = Empty
```

```
End Sub
```

7.2.2.31. Integrador restador (IntegradorRestador)

```
Private Sub Teoria1_Click() 'Botón "Teoría" del tipo 1'
```

```
ModIntRes.MostrarTeoriaIntRes
```

```
End Sub
```

```
Private Sub Teoria2_Click() 'Botón "Teoría" del tipo 2'
```

```
ModIntRes.MostrarTeoriaIntRes
```

```
End Sub
```

```
Private Sub Calcular1_Click() 'Botón "Calcular" del tipo 1'
```

```
ModIntRes.CalcularValores1
```

```
End Sub
```

```
Private Sub Calcular2_Click() 'Botón "Calcular" del tipo 2'
```

```
ModIntRes.CalcularValores2
```

```
End Sub
```

```
Private Sub Restaurar1_Click() 'Botón "Restaurar valores" del tipo  
1'
```

```
ModIntRes.RestaurarValores1
```

```
End Sub
```

```
Private Sub Restaurar2_Click() 'Botón "Restaurar valores" del tipo  
2'
```

```
ModIntRes.RestaurarValores2
```

```
End Sub
```

```
Private Sub Informacion1_Click() 'Botón "Información" del tipo 1'
```

```
ModIntRes.MostrarInformacion1
```

```
End Sub
```



```
Private Sub Informacion2_Click() 'Botón "Información" del tipo 2'  
  
ModIntRes.MostrarInformacion2  
  
End Sub  
  
Private Sub UserForm_Activate() 'Definimos para el tipo 2 si la seña de  
entrada es un seno o un coseno'  
  
Sen_Cos1.AddItem "sen"  
Sen_Cos1.AddItem "cos"  
Sen_Cos2.AddItem "sen"  
Sen_Cos2.AddItem "cos"  
  
End Sub
```

7.2.2.32. Integrador sumador (IntegradorSumador)

```
Private Sub Teoria1_Click() 'Botón "Teoría" del tipo 1'  
  
ModIntSum.MostrarTeoriaIntSum  
  
End Sub  
  
Private Sub Teoria2_Click() 'Botón "Teoría" del tipo 2'  
  
ModIntSum.MostrarTeoriaIntSum  
  
End Sub  
  
Private Sub Calcular1_Click() 'Botón "Calcular" del tipo 1'  
  
ModIntSum.CalcularValores1  
  
End Sub  
  
Private Sub Calcular2_Click() 'Botón "Calcular" del tipo 2'  
  
ModIntSum.CalcularValores2  
  
End Sub  
  
Private Sub Restaurar1_Click() 'Botón "Restaurar valores" del tipo  
1'  
  
ModIntSum.RestaurarValores1  
  
End Sub  
  
Private Sub Restaurar2_Click() 'Botón "Restaurar valores" del tipo  
2'  
  
ModIntSum.RestaurarValores2
```



```
End Sub

Private Sub Informacion1_Click() 'Botón "Información" del tipo 1'
ModIntSum.MostrarInformacion1
End Sub

Private Sub Informacion2_Click() 'Botón "Información" del tipo 2'
ModIntSum.MostrarInformacion2
End Sub

Private Sub UserForm_Activate() 'Definimos para el tipo 2 si la se de
entrada es un seno o un coseno'

Sen_Cos1.AddItem "sen"
Sen_Cos1.AddItem "cos"
Sen_Cos2.AddItem "sen"
Sen_Cos2.AddItem "cos"
Sen_Cos3.AddItem "sen"
Sen_Cos3.AddItem "cos"

End Sub
```

7.2.2.33. Circuito inversor (Inversor)

```
Private Sub Teoria_Click() 'Botón "Teoría"'
ModInversor.MostrarTeoriaInversor
End Sub

Private Sub Calcular_Click() 'Botón "Calcular"'
ModInversor.CalcularValores
End Sub

Private Sub Restaurar_Click() 'Botón "Restaurar valores"'
ModInversor.RestaurarValores
End Sub

Private Sub Informacion_Click() 'Botón "Información"'
ModInversor.MostrarInformacion
End Sub

Private Sub Vo_Vi_Click() 'Botón "Representación Vo-Vi"'

'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If Inversor.Valor_Vo.Caption = Empty Then
```



```
MsgBox "Debe obtener primero un valor de Vo para poder calcular la  
gráfica Vo-Vi", vbExclamation + vbOKOnly, "AVISO"
```

```
Else
```

```
'Cargamos la gráfica'  
Load GraficaVoVi  
  
'Refrescamos la gráfica'  
GraficaVoVi.Repaint  
  
'Mostramos la gráfica'  
GraficaVoVi.Show
```

```
End If
```

```
End Sub
```

```
Private Sub UserForm_terminate()
```

```
'Vaciamos las celdas de la hoja de cálculo "Gráfica Vo-Vi"  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c5") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c7") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c8") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c10") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d10") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c11") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d11") = Empty
```

```
End Sub
```

7.2.2.34. Limitador con diodo normal (LimitadorDiodoNormal)

```
Private Sub Teoria1_Click() 'Botón "Teoría" del ejemplo 1'
```

```
ModCircLim.MostrarTeoriaLimDiodoNormal1
```

```
End Sub
```

```
Private Sub Teoria2_Click() 'Botón "Teoría" del ejemplo 2'
```

```
ModCircLim.MostrarTeoriaLimDiodoNormal2
```

```
End Sub
```

```
Private Sub Calcular1_Click() 'Botón "Calcular" del ejemplo 1'
```

```
ModCircLim.CalcularValoresDiodoNormal1
```

```
End Sub
```

```
Private Sub Calcular2_Click() 'Botón "Calcular" del ejemplo 2'
```




```
ModCircLim.CalcularValoresDiodoNormal2

End Sub

Private Sub Restaurar1_Click() 'Botón "Restaurar valores" del
ejemplo 1'

ModCircLim.RestaurarValoresDiodoNormal1

End Sub

Private Sub Restaurar2_Click() 'Botón "Restaurar valores" del
ejemplo 2'

ModCircLim.RestaurarValoresDiodoNormal2

End Sub

Private Sub Informacion1_Click() 'Botón "Información" del ejemplo 1'

ModCircLim.MostrarInformacionLimDiodoNormal

End Sub

Private Sub Informacion2_Click() 'Botón "Información" del ejemplo 2'

ModCircLim.MostrarInformacionLimDiodoNormal

End Sub

Private Sub Vo_Vi_1_Click() 'Botón "Representación Vo-Vi" del
ejemplo 1'

'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If LimitadorDiodoNormal.Valor_Vo_1.Caption = Empty Then

    MsgBox "Debe obtener primero un valor de Vo para poder calcular la
gráfica Vo-Vi", vbExclamation + vbOKOnly, "AVISO"

Else

    'Cargamos la gráfica'
    Load GraficaVoVi_2

    'Refrescamos la gráfica'
    GraficaVoVi_2.Repaint

    'Mostramos la gráfica'
    GraficaVoVi_2.Show

End If

End Sub

Private Sub Vo_Vi_2_Click() 'Botón "Representación Vo-Vi" del
ejemplo 2'

'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If LimitadorDiodoNormal.Valor_Vo_2.Caption = Empty Then
```



```
MsgBox "Debe obtener primero un valor de Vo para poder calcular la  
gráfica Vo-Vi", vbExclamation + vbOKOnly, "AVISO"
```

```
Else
```

```
'Cargamos la gráfica'  
Load GraficaVoVi_2  
  
'Refrescamos la gráfica'  
GraficaVoVi_2.Repaint  
  
'Mostramos la gráfica'  
GraficaVoVi_2.Show
```

```
End If
```

```
End Sub
```

```
Private Sub UserForm_terminate()
```

```
'Vaciamos las celdas de la hoja de cálculo "Gráfica Vo-Vi"  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c5") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c16") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c17") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c19") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d19") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c20") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d20") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c21") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d21") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c22") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d22") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c24") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c25") = Empty
```

```
End Sub
```

7.2.2.35. Limitador con diodo normal y diodo Zener en serie (LimitadorDiodoNormalDiodoZener)

```
Private Sub Teoria1_Click() 'Botón "Teoría" del ejemplo 1'
```

```
ModCircLim.MostrarTeoriaLimDiodoNormalDiodoZener1
```

```
End Sub
```

```
Private Sub Teoria2_Click() 'Botón "Teoría" del ejemplo 2'
```

```
ModCircLim.MostrarTeoriaLimDiodoNormalDiodoZener2
```

```
End Sub
```



```
Private Sub Calcular1_Click()      'Botón "Calcular" del ejemplo 1'
ModCircLim.CalcularValoresDiodoNormalDiodoZener1
End Sub

Private Sub Calcular2_Click()      'Botón "Calcular" del ejemplo 2'
ModCircLim.CalcularValoresDiodoNormalDiodoZener2
End Sub

Private Sub Restaurar1_Click()     'Botón "Restaurar valores" del
ejemplo 1'
ModCircLim.RestaurarValoresDiodoNormalDiodoZener1
End Sub

Private Sub Restaurar2_Click()     'Botón "Restaurar valores" del
ejemplo 2'
ModCircLim.RestaurarValoresDiodoNormalDiodoZener2
End Sub

Private Sub Informacion1_Click()   'Botón "Información" del ejemplo 1'
ModCircLim.MostrarInformacionLimDiodoNormalDiodoZener
End Sub

Private Sub Informacion2_Click()   'Botón "Información" del ejemplo 2'
ModCircLim.MostrarInformacionLimDiodoNormalDiodoZener
End Sub

Private Sub Vo_Vi_1_Click()        'Botón "Representación Vo-Vi" del
ejemplo 1'

'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If LimitadorDiodoNormalDiodoZener.Valor_Vo_1.Caption = Empty Then

    MsgBox "Debe obtener primero un valor de Vo para poder calcular la
gráfica Vo-Vi", vbExclamation + vbOKOnly, "AVISO"

Else

    'Cargamos la gráfica'
    Load GraficaVoVi_2

    'Refrescamos la gráfica'
    GraficaVoVi_2.Repaint

    'Mostramos la gráfica'
    GraficaVoVi_2.Show

End If
```



```
End Sub

Private Sub Vo_Vi_2_Click() 'Botón "Representación Vo-Vi" del
ejemplo 2'

'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If LimitadorDiodoNormalDiodoZener.Valor_Vo_2.Caption = Empty Then

    MsgBox "Debe obtener primero un valor de Vo para poder calcular la
gráfica Vo-Vi", vbExclamation + vbOKOnly, "AVISO"

Else

    'Cargamos la gráfica'
    Load GraficaVoVi_2

    'Refrescamos la gráfica'
    GraficaVoVi_2.Repaint

    'Mostramos la gráfica'
    GraficaVoVi_2.Show

End If

End Sub

Private Sub UserForm_terminate()

'Vaciamos las celdas de la hoja de cálculo "Gráfica Vo-Vi"'
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c5") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c27") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c28") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c30") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d30") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c31") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d31") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c32") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d32") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c33") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d33") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c35") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c36") = Empty

End Sub
```

7.2.2.36. Limitador con dos diodos Zener (LimitadorDosZener)

```
Private Sub Teoria_Click() 'Botón "Teoría"'

ModCircLim.MostrarTeoriaLimDosZener

End Sub
```



```
Private Sub Calcular_Click() 'Botón "Calcular"'
ModCircLim.CalcularValoresDosZener
End Sub

Private Sub Restaurar_Click() 'Botón "Restaurar valores"'
ModCircLim.RestaurarValoresDosZener
End Sub

Private Sub Informacion_Click() 'Botón "Información"'
ModCircLim.MostrarInformacionLimDosZener
End Sub

Private Sub Vo_Vi_Click() 'Botón "Representación Vo-Vi"'

'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If LimitadorDosZener.Valor_Vo.Caption = Empty Then

MsgBox "Debe obtener primero un valor de Vo para poder calcular la
gráfica Vo-Vi", vbExclamation + vbOKOnly, "AVISO"

Else

'Cargamos la gráfica'
Load GraficaVoVi_2

'Refrescamos la gráfica'
GraficaVoVi_2.Repaint

'Mostramos la gráfica'
GraficaVoVi_2.Show

End If

End Sub

Private Sub UserForm_terminate()

'Vaciamos las celdas de la hoja de cálculo "Gráfica Vo-Vi"'
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c5") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c38") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c39") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c41") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d41") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c42") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d42") = Empty

End Sub
```



7.2.2.37. Circuito no inversor (NoInversor)

```
Private Sub Teoria_Click() 'Botón "Teoría"
ModNoInversor.MostrarTeoriaNoInversor
End Sub

Private Sub Calcular_Click() 'Botón "Calcular"
ModNoInversor.CalcularValores
End Sub

Private Sub Restaurar_Click() 'Botón "Restaurar valores"
ModNoInversor.RestaurarValores
End Sub

Private Sub Informacion_Click() 'Botón "Información"
ModNoInversor.MostrarInformacion
End Sub

Private Sub Vo_Vi_Click() 'Botón "Representación Vo-Vi"
'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If NoInversor.Valor_Vo.Caption = Empty Then

MsgBox "Debe obtener primero un valor de Vo para poder calcular la
gráfica Vo-Vi", vbExclamation + vbOKOnly, "AVISO"

Else

'Cargamos la gráfica'
Load GraficaVoVi

'Refrescamos la gráfica'
GraficaVoVi.Repaint

'Mostramos la gráfica'
GraficaVoVi.Show

End If
End Sub

Private Sub UserForm_terminate()

'Vaciamos las celdas de la hoja de cálculo "Gráfica Vo-Vi"
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c5") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c10") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d10") = Empty
```



```
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c11") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d11") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c13") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c14") = Empty
```

```
End Sub
```

7.2.2.38. Generador de onda cuadrada (OndaCuadrada)

```
Private Sub Teoria_Click() 'Botón "Teoría"  
  
ModOndaCua.MostrarTeoriaOndaCuadrada  
  
End Sub  
  
Private Sub Calcular_Click() 'Botón "Calcular"  
  
ModOndaCua.CalcularValores  
  
End Sub  
  
Private Sub Restaurar_Click() 'Botón "Restaurar valores"  
  
ModOndaCua.RestaurarValores  
  
End Sub  
  
Private Sub Informacion_Click() 'Botón "Información"  
  
ModOndaCua.MostrarInformacion  
  
End Sub  
  
Private Sub Vo_t_Click() 'Botón "Representación Vo-t"  
  
'Si no hemos obtenido el periodo, no se mostrará la gráfica y saltará  
un mensaje indicando que debemos calcular el periodo de la onda'  
If OndaCuadrada.Periodo.Caption = Empty Then  
  
MsgBox "Debe calcular el periodo para poder calcular la gráfica  
Vo-t", vbExclamation + vbOKOnly, "AVISO"  
  
Else  
  
'Cargamos la gráfica'  
Load GraficaVot_2  
  
'Refrescamos la gráfica'  
GraficaVot_2.Repaint  
  
'Mostramos la gráfica'  
GraficaVot_2.Show  
  
End If
```



```
End Sub
```

```
Private Sub UserForm_terminate()
```

```
'Vaciamos las celdas de la hoja de cálculo "Ondas"  
ActiveWorkbook.Worksheets("Ondas").Range("b2") = Empty  
ActiveWorkbook.Worksheets("Ondas").Range("c2") = Empty  
ActiveWorkbook.Worksheets("Ondas").Range("b9:b400") = Empty  
ActiveWorkbook.Worksheets("Ondas").Range("c8:c400") = Empty  
ActiveWorkbook.Worksheets("Ondas").Range("d8:d400") = Empty
```

```
End Sub
```

7.2.2.39. Generador de onda triangular (OndaTriangular)

```
Private Sub Teoria_Click() 'Botón "Teoría"
```

```
ModOndaTri.MostrarTeoriaOndaTriangular
```

```
End Sub
```

```
Private Sub Calcular_Click() 'Botón "Calcular"
```

```
ModOndaTri.CalcularValores
```

```
End Sub
```

```
Private Sub Restaurar_Click() 'Botón "Restaurar valores"
```

```
ModOndaTri.RestaurarValores
```

```
End Sub
```

```
Private Sub Informacion_Click() 'Botón "Información"
```

```
ModOndaTri.MostrarInformacion
```

```
End Sub
```

```
Private Sub Vo_t_Click() 'Botón "Representación Vo-t"
```

```
'Si no hemos obtenido el periodo, no se mostrará la gráfica y saltará  
un mensaje indicando que debemos calcular el periodo de la onda'  
If OndaTriangular.Periodo.Caption = Empty Then
```

```
MsgBox "Debe calcular el periodo para poder calcular la gráfica  
Vo-t", vbExclamation + vbOKOnly, "AVISO"
```

```
Else
```

```
'Cargamos la gráfica'  
Load GraficaVot_2
```




```
'Refrescamos la gráfica'  
GraficaVot_2.Repaint  
  
'Mostramos la gráfica'  
GraficaVot_2.Show  
  
End If  
  
End Sub  
  
Private Sub UserForm_terminate()  
  
'Vaciamos las celdas de la hoja de cálculo "Ondas"  
ActiveWorkbook.Worksheets("Ondas").Range("b2") = Empty  
ActiveWorkbook.Worksheets("Ondas").Range("c2") = Empty  
ActiveWorkbook.Worksheets("Ondas").Range("q9:q400") = Empty  
ActiveWorkbook.Worksheets("Ondas").Range("r8:r400") = Empty  
ActiveWorkbook.Worksheets("Ondas").Range("s8:s400") = Empty  
  
End Sub
```

7.2.2.40. Rectificador de media onda (RectMediaOnda)

```
Private Sub Teoria_Click() 'Botón "Teoría"  
  
ModCircRectPre.MostrarTeoriaMediaOnda  
  
End Sub  
  
Private Sub Calcular_Click() 'Botón "Calcular"  
  
ModCircRectPre.CalcularValoresMediaOnda  
  
End Sub  
  
Private Sub Restaurar_Click() 'Botón "Restaurar valores"  
  
ModCircRectPre.RestaurarValoresMediaOnda  
  
End Sub  
  
Private Sub Informacion_Click() 'Botón "Información"  
  
ModCircRectPre.MostrarInformacionMediaOnda  
  
End Sub  
  
Private Sub ViVo_t_Click() 'Botón "Representación de Vi y Vo frente  
a t"  
  
'Si no hemos obtenido previamente un valor de Vo, no se mostrará la  
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'  
If RectMediaOnda.Valor_Vo.Caption = Empty Then
```



```
MsgBox "Debe obtener primero un valor de Vo para poder calcular  
las gráficas Vi-t Vo-t", vbExclamation + vbOKOnly, "AVISO"
```

```
Else
```

```
'Cargamos la gráfica'  
Load GraficaSchmitt  
  
'Refrescamos la gráfica'  
GraficaSchmitt.Repaint  
  
'Mostramos la gráfica'  
GraficaSchmitt.Show
```

```
End If
```

```
End Sub
```

```
Private Sub UserForm_terminate()
```

```
'Vaciamos las celdas de la hoja de cálculo "Gráfica Vo-t"  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("u5") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v5") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("s8:s728") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v8:v728") = Empty  
  
'Vaciamos las celdas de la hoja de cálculo "Gráfica Vo-t 2"  
ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("b5") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("d8:d410") = Empty
```

```
End Sub
```

7.2.2.41. Rectificador de onda completa (RectOndaCompleta)

```
Private Sub Teoria1_Click() 'Botón "Teoría" del ejemplo 1'
```

```
ModCircRectPre.MostrarTeoriaOndaCompleta1
```

```
End Sub
```

```
Private Sub Teoria2_Click() 'Botón "Teoría" del ejemplo 2'
```

```
ModCircRectPre.MostrarTeoriaOndaCompleta2
```

```
End Sub
```

```
Private Sub Calcular1_Click() 'Botón "Calcular" del ejemplo 1'
```

```
ModCircRectPre.CalcularValoresOndaCompleta1
```

```
End Sub
```

```
Private Sub Calcular2_Click() 'Botón "Calcular" del ejemplo 2'
```



```
ModCircRectPre.CalcularValoresOndaCompleta2

End Sub

Private Sub Restaurar1_Click() 'Botón "Restaurar valores" del
ejemplo 1'

ModCircRectPre.RestaurarValoresOndaCompleta1

End Sub

Private Sub Restaurar2_Click() 'Botón "Restaurar valores" del
ejemplo 2'

ModCircRectPre.RestaurarValoresOndaCompleta2

End Sub

Private Sub Informacion1_Click() 'Botón "Información" del ejemplo 1'

ModCircRectPre.MostrarInformacionOndaCompleta

End Sub

Private Sub Informacion2_Click() 'Botón "Información" del ejemplo 2'

ModCircRectPre.MostrarInformacionOndaCompleta

End Sub

Private Sub ViVo_t_1_Click() 'Botón "Representación de Vi y Vo
frente a t" del ejemplo 1'

'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If RectOndaCompleta.Valor_Vo1_1.Caption = Empty Then

    MsgBox "Debe obtener primero un valor de Vo para poder calcular la
gráficas Vi-t Vo-t", vbExclamation + vbOKOnly, "AVISO"

Else

    'Cargamos la gráfica'
    Load GraficaSchmitt

    'Refrescamos la gráfica'
    GraficaSchmitt.Repaint

    'Mostramos la gráfica'
    GraficaSchmitt.Show

End If

End Sub

Private Sub ViVo_t_2_Click() 'Botón "Representación de Vi y Vo
frente a t" del ejemplo 2'

'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If RectOndaCompleta.Valor_Vo_2.Caption = Empty Then
```



```
MsgBox "Debe obtener primero un valor de Vo para poder calcular  
las gráficas Vi-t Vo-t", vbExclamation + vbOKOnly, "AVISO"
```

```
Else
```

```
'Cargamos la gráfica'  
Load GraficaSchmitt  
  
'Refrescamos la gráfica'  
GraficaSchmitt.Repaint  
  
'Mostramos la gráfica'  
GraficaSchmitt.Show
```

```
End If
```

```
End Sub
```

```
Private Sub UserForm_terminate()
```

```
'Vaciamos las celdas de la hoja de cálculo "Gráfica Vo-t"  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("u5") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v5") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("s8:s728") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v8:v728") = Empty
```

```
'Vaciamos las celdas de la hoja de cálculo "Gráfica Vo-t 2"  
ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("b5") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("c5") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("d8:d410") = Empty
```

```
End Sub
```

7.2.2.42. Restador de alta impedancia a la entrada (RestadorAltaImp)

```
Private Sub Teoria_Click() 'Botón "Teoría"
```

```
ModRestador.MostrarTeoria2
```

```
End Sub
```

```
Private Sub Calcular_Click() 'Botón "Calcular"
```

```
ModRestador.CalcularValores2
```

```
End Sub
```

```
Private Sub Restaurar_Click() 'Botón "Restaurar valores"
```

```
ModRestador.RestaurarValores2
```

```
End Sub
```



```
Private Sub Informacion_Click() 'Botón "Información"  
ModRestador.MostrarInformacion2  
End Sub
```

7.2.2.43. Restador simple (RestadorSimple)

```
Private Sub Teoria_Click() 'Botón "Teoría"  
ModRestador.MostrarTeoria1  
End Sub  
Private Sub Calcular_Click() 'Botón "Calcular"  
ModRestador.CalcularValores1  
End Sub  
Private Sub Restaurar_Click() 'Botón "Restaurar valores"  
ModRestador.RestaurarValores1  
End Sub  
Private Sub Informacion_Click() 'Botón "Información"  
ModRestador.MostrarInformacion1  
End Sub
```

7.2.2.44. Sumador inversor (SumadorInversor)

```
Private Sub Teoria_Click() 'Botón "Teoría"  
ModSumador.MostrarTeoria1  
End Sub  
Private Sub Calcular_Click() 'Botón "Calcular"  
ModSumador.CalcularValores1  
End Sub  
Private Sub Restaurar_Click() 'Botón "Restaurar valores"
```



```
ModSumador.RestaurarValores1  
  
End Sub  
  
Private Sub Informacion_Click() 'Botón "Información"  
  
ModSumador.MostrarInformacion1  
  
End Sub
```

7.2.2.45. Sumador no inversor (SumadorNoInversor)

```
Private Sub Teoria_Click() 'Botón "Teoría"  
  
ModSumador.MostrarTeoria2  
  
End Sub  
  
Private Sub Calcular_Click() 'Botón "Calcular"  
  
ModSumador.CalcularValores2  
  
End Sub  
  
Private Sub Restaurar_Click() 'Botón "Restaurar valores"  
  
ModSumador.RestaurarValores2  
  
End Sub  
  
Private Sub Informacion_Click() 'Botón "Información"  
  
ModSumador.MostrarInformacion2  
  
End Sub
```

7.2.2.46. Circuitos de valor medio y valor eficaz (ValorMedioValorEficaz)

```
Private Sub Teoria_Click() 'Botón "Teoría"  
  
ModCircRectPre.MostrarTeoriaValorMedioEficaz  
  
End Sub  
  
Private Sub Calcular_Click() 'Botón "Calcular"
```



```
ModCircRectPre.CalcularValoresMedioEficaz

End Sub

Private Sub Restaurar_Click() 'Botón "Restaurar valores"

ModCircRectPre.RestaurarValorMedioEficaz

End Sub

Private Sub Informacion_Click() 'Botón "Información"

ModCircRectPre.MostrarInformacionValorMedioEficaz

End Sub

Private Sub ViVo_t_Click() 'Botón "Representación de Vi y Vo frente
a t"

'Si no hemos obtenido previamente un valor de Vo, no se mostrará la
gráfica y saltará un mensaje indicando que calculemos el valor de Vo'
If ValorMedioValorEficaz.Valor_Vo.Caption = Empty Then

MsgBox "Debe obtener primero un valor de Vo para poder calcular
las gráficas Vi-t Vo-t", vbExclamation + vbOKOnly, "AVISO"

Else

'Cargamos la gráfica'
Load GraficaSchmitt

'Refrescamos la gráfica'
GraficaSchmitt.Repaint

'Mostramos la gráfica'
GraficaSchmitt.Show

End If

End Sub

Private Sub UserForm_terminate()

'Vaciamos las celdas de la hoja de cálculo de la "Gráfica Vo-t"
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("u5") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v5") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("s8:s410") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v8:v410") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("x8:x410") = Empty

'Vaciamos las celdas de la hoja de cálculo "Gráfica CosSen"
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("c2") = Empty
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("d2") = Empty
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("c3") = Empty
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("d3") = Empty

End Sub
```



7.2.3. Código de módulos

7.2.3.1. Botones de la hoja principal (Circuitos)

```
Sub MostrarCircSum()  
  
'Mostramos la pestaña de la aplicación de los circuitos sumadores'  
CircSum.Show  
  
End Sub  
  
Sub MostrarCircRes()  
  
'Mostramos la pestaña de la aplicación de los circuitos restadores'  
CircRes.Show  
  
End Sub  
  
Sub MostrarCircInt()  
  
'Mostramos la pestaña de la aplicación de los circuitos integradores'  
CircInt.Show  
  
End Sub  
  
Sub MostrarCircLim()  
  
'Mostramos la pestaña de la aplicación de los circuitos limitadores'  
CircLim.Show  
  
End Sub  
  
Sub MostrarCircCom()  
  
'Mostramos la pestaña de la aplicación de los circuitos comparadores'  
CircCom.Show  
  
End Sub  
  
Sub MostrarCircHis()  
  
'Mostramos la pestaña de la aplicación de los circuitos comparadores  
con histéresis'  
CircHis.Show  
  
End Sub  
  
Sub MostrarCircRectPre()  
  
'Mostramos la pestaña de la aplicación de los circuitos rectificadores  
de precisión'  
CircRectPre.Show
```




End Sub

7.2.3.2. Detectores de tensión o comparadores (ModCircComp)

```
Sub MostrarSimple ()
```

```
'Mostramos la pestaña de la aplicación del circuito comparador simple'  
CompSimple.Show
```

```
End Sub
```

```
Sub MostrarTeoriaCompSimple ()
```

```
'Mostramos la pestaña de la teoría del circuito comparador simple'  
TeoriaCompSimple.Show
```

```
End Sub
```

```
Sub MostrarInformacionCompSimple ()
```

```
'Mostramos la pestaña de la información del circuito comparador  
simple'  
Informacion.Show
```

```
End Sub
```

```
Sub CalcularValoresCompSimple ()
```

```
'Declaramos las variables de los parámetros de un circuito comparador  
simple'
```

```
Dim Vi As Single  
Dim Vref As Single  
Dim VsatPos As Single  
Dim VsatNeg As Single  
Dim Vo As Single
```

```
'Mediante esta declaración conseguimos que nuestra macro no pare de  
ejecutarse aunque se encuentre un error'  
On Error Resume Next
```

```
'Nos aseguramos de que se introduzca un valor en las casillas de las  
tensiones, pudiendo ser cero o negativo'
```

```
If CompSimple.Valor_Vi.Text = Empty Or CompSimple.Valor_Vref.Text =  
Empty Or CompSimple.Valor_VsatPos.Text = Empty Or  
CompSimple.Valor_VsatNeg.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas del Voltaje",  
vbExclamation + vbOKOnly, "AVISO"
```

```
CompSimple.Valor_Vo.Caption = ""  
CompSimple.Valor_Saturacion.Caption = ""
```

```
Else
```



```
'Asignamos las variables a los correspondientes parámetros de un
circuito comparador simple'
Vi = CompSimple.Valor_Vi.Text
Vref = CompSimple.Valor_Vref.Text
VsatPos = CompSimple.Valor_VsatPos.Text
VsatNeg = CompSimple.Valor_VsatNeg.Text

'Introducimos la condición de que -Vsat NO puede ser mayor que
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
de Vo'
If VsatPos < VsatNeg Then

    MsgBox "La tensión de saturación -Vsat no puede ser mayor que
+Vsat", vbInformation + vbOKOnly, "AVISO"

    CompSimple.Valor_VsatPos.Text = ""
    CompSimple.Valor_VsatNeg.Text = ""
    CompSimple.Valor_Vo.Caption = ""
    CompSimple.Valor_Saturacion.Caption = ""

Else

    'Definimos las ecuaciones de los parametros a calcular y
mostramos el resultado'
    If Vi < Vref Then

        Vo = VsatPos

    ElseIf Vi > Vref Then

        Vo = VsatNeg

    ElseIf Vi = Vref Then

        Vo = 0

    End If

    CompSimple.Valor_Vo.Caption = Vo

    'Analizamos si la tensión de salida está saturada'
    If Vo >= VsatPos Then

        CompSimple.Valor_Vo.Caption = VsatPos
        CompSimple.Valor_Saturacion.Caption = "Salida saturada"

    ElseIf Vo <= VsatNeg Then

        CompSimple.Valor_Vo.Caption = VsatNeg
        CompSimple.Valor_Saturacion.Caption = "Salida saturada"

    Else

        CompSimple.Valor_Saturacion.Caption = ""
        CompSimple.Valor_Vo.Caption = Vo

    End If

    'Mostramos la representación Vo-Vi'
    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s6") = Vref
```



```
        ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t6") = Vref
        ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s9") =
VsatPos
        ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t9") =
VsatPos
        ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s10") =
VsatNeg
        ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t10") =
VsatNeg

    End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub RestaurarValoresCompSimple ()

'Vaciamos los valores de cada uno de los parámetros'
CompSimple.Valor_Vi.Text = ""
CompSimple.Valor_Vref.Text = ""
CompSimple.Valor_VsatPos.Text = ""
CompSimple.Valor_VsatNeg.Text = ""
CompSimple.Valor_Vo.Caption = ""
CompSimple.Valor_Saturacion.Caption = ""

End Sub

Sub MostrarSatSat ()

'Mostramos la pestaña de la aplicación del circuito detector con
salida saturación-saturación'
CompSatSat.Show

End Sub

Sub MostrarTeoriaCompSatSat ()

'Mostramos la pestaña de la teoría del circuito detector con salida
saturación-saturación'
TeoriaCompSatSat.Show

End Sub

Sub MostrarInformacionCompSatSat ()

'Mostramos la pestaña de la información del circuito detector con
salida saturación-saturación'
Informacion.Show

End Sub

Sub CalcularValoresCompSatSat ()

'Declaremos las variables de los parámetros de un circuito detector
con salida saturación-saturación'
Dim R1 As Single
Dim R2 As Single
```



```
Dim Vi As Single
Dim Vref As Single
Dim Vx As Single
Dim VsatPos As Single
Dim VsatNeg As Single
Dim R3 As Single
Dim Vo As Single

'Mediante esta declaración, conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de las resistencias, su valor no puede ser cero o
negativo'
'En el caso de las tensiones, su valor sí puede ser cero o negativo'
If CompSatSat.Valor_R1.Text <= 0 Or CompSatSat.Valor_R2.Text <= 0 Or
CompSatSat.Valor_R1.Text = Empty Or CompSatSat.Valor_R2.Text = Empty
Then

    MsgBox "Debe introducir un valor en las casillas de las
resistencias, y que no sea ni nulo ni negativo", vbExclamation +
vbOKOnly, "AVISO"

    CompSatSat.Valor_R3.Caption = ""
    CompSatSat.Valor_Vo.Caption = ""
    CompSatSat.Valor_Saturacion.Caption = ""

ElseIf CompSatSat.Valor_Vi.Text = Empty Or CompSatSat.Valor_Vref.Text
= Empty Or CompSatSat.Valor_VsatPos.Text = Empty Or
CompSatSat.Valor_VsatNeg.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas del Voltaje",
vbExclamation + vbOKOnly, "AVISO"

    CompSatSat.Valor_R3.Caption = ""
    CompSatSat.Valor_Vo.Caption = ""
    CompSatSat.Valor_Saturacion.Caption = ""

Else

    'Asignamos las variables a los correspondientes parámetros de un
circuito detector con salida saturación-saturación'
    R1 = CompSatSat.Valor_R1.Text * 1000
    R2 = CompSatSat.Valor_R2.Text * 1000
    Vi = CompSatSat.Valor_Vi.Text
    Vref = CompSatSat.Valor_Vref.Text
    VsatPos = CompSatSat.Valor_VsatPos.Text
    VsatNeg = CompSatSat.Valor_VsatNeg.Text

    'Introducimos la condición de que -Vsat NO puede ser mayor que
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
de Vo'
    If VsatPos < VsatNeg Then

        MsgBox "La tensión de saturación -Vsat no puede ser mayor que
+Vsat", vbInformation + vbOKOnly, "AVISO"

        CompSatSat.Valor_VsatPos.Text = ""
        CompSatSat.Valor_VsatNeg.Text = ""
        CompSatSat.Valor_R3.Caption = ""
```



```
CompSatSat.Valor_Vo.Caption = ""
CompSatSat.Valor_Saturacion.Caption = ""

Else

    'Definimos las ecuaciones de los parametros a calcular y
    mostramos el resultado'
    
$$V_x = (V_i * R_2 + V_{ref} * R_1) / (R_1 + R_2)$$


    If Vi > -Vref * (R1 / R2) Then

        Vo = VsatNeg

    ElseIf Vi < -Vref * (R1 / R2) Then

        Vo = VsatPos

    ElseIf Vi = -Vref * (R1 / R2) Then

        Vo = 0

    End If

    
$$R_3 = ((R_1 * R_2) / (R_1 + R_2)) / 1000$$


    CompSatSat.Valor_R3.Caption = R3
    CompSatSat.Valor_Vo.Caption = Vo

    'Analizamos si la tensión de salida está saturada'
    If Vo >= VsatPos Then

        CompSatSat.Valor_Vo.Caption = VsatPos
        CompSatSat.Valor_Saturacion.Caption = "Salida saturada"

    ElseIf Vo <= VsatNeg Then

        CompSatSat.Valor_Vo.Caption = VsatNeg
        CompSatSat.Valor_Saturacion.Caption = "Salida saturada"

    Else

        CompSatSat.Valor_Saturacion.Caption = ""
        CompSatSat.Valor_Vo.Caption = Vo

    End If

    'Mostramos la representación Vo-Vi'
    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s6") = -Vref
    * (R1 / R2)
    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t6") = -Vref
    * (R1 / R2)
    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s9") =
    VsatPos
    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t9") =
    VsatPos
    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s10") =
    VsatNeg
    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t10") =
    VsatNeg
    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("q3") = (Vref
    * R1) / (R1 + R2)
```



```
        ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("r3") = R1 /  
(R1 + R2)  
  
        End If  
  
End If  
  
'Desactivamos el detector de errores'  
On Error GoTo 0  
  
End Sub  
  
Sub RestaurarValoresCompSatSat()  
  
'Vaciamos los valores de cada uno de los parámetros'  
CompSatSat.Valor_R1.Text = ""  
CompSatSat.Valor_R2.Text = ""  
CompSatSat.Valor_Vi.Text = ""  
CompSatSat.Valor_Vref.Text = ""  
CompSatSat.Valor_VsatPos.Text = ""  
CompSatSat.Valor_VsatNeg.Text = ""  
CompSatSat.Valor_R3.Caption = ""  
CompSatSat.Valor_Vo.Caption = ""  
CompSatSat.Valor_Saturacion.Caption = ""  
  
End Sub  
  
Sub MostrarCeroSat()  
  
'Mostramos la pestaña de la aplicación del circuito detector de salida  
cero-saturación'  
CompCeroSat.Show  
  
End Sub  
  
Sub MostrarTeoriaCompCeroSat1()  
  
'Mostramos la pestaña de la teoría del ejemplo 1 del circuito detector  
de salida cero-saturación'  
TeoriaCompCeroSat1.Show  
  
End Sub  
  
Sub MostrarTeoriaCompCeroSat2()  
  
'Mostramos la pestaña de la teoría del ejemplo 2 del circuito detector  
de salida cero-saturación'  
TeoriaCompCeroSat2.Show  
  
End Sub  
  
Sub MostrarInformacionCompCeroSat()  
  
'Mostramos la pestaña de la información del circuito detector de  
salida cero-saturación'  
Informacion.Show  
  
End Sub  
  
Sub CalcularValoresCompCeroSat1()
```



```
'Declaramos las variables de los parámetros del ejemplo 1 de un
circuito detector con salida cero-saturación'
Dim R1_1 As Single
Dim R2_1 As Single
Dim Vi_1 As Single
Dim Vref_1 As Single
Dim Vu_1 As Single
Dim VsatPos_1 As Single
Dim VsatNeg_1 As Single
Dim R3_1 As Single
Dim Vo_1 As Single

'Mediante esta declaración, conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de las resistencias, su valor no puede ser cero o
negativo'
'En el caso de las tensiones, su valor sí puede ser cero o negativo'
If CompCeroSat.Valor_R1_1.Text <= 0 Or CompCeroSat.Valor_R2_1.Text <=
0 Or CompCeroSat.Valor_R1_1.Text = Empty Or
CompCeroSat.Valor_R2_1.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas de las
resistencias, y que no sea ni nulo ni negativo", vbExclamation +
vbOKOnly, "AVISO"

    CompCeroSat.Valor_R3_1.Caption = ""
    CompCeroSat.Valor_Vo_1.Caption = ""
    CompCeroSat.Valor_Saturacion_1.Caption = ""
    CompCeroSat.Conduccion_1.Caption = ""

ElseIf CompCeroSat.Valor_Vi_1.Text = Empty Or
CompCeroSat.Valor_Vref_1.Text = Empty Or CompCeroSat.Valor_Vu_1.Text =
Empty Or CompCeroSat.Valor_VsatPos_1.Text = Empty Or
CompCeroSat.Valor_VsatNeg_1.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas del Voltaje",
vbExclamation + vbOKOnly, "AVISO"

    CompCeroSat.Valor_R3_1.Caption = ""
    CompCeroSat.Valor_Vo_1.Caption = ""
    CompCeroSat.Valor_Saturacion_1.Caption = ""
    CompCeroSat.Conduccion_1.Caption = ""

Else

    'Asignamos las variables a los correspondientes parámetros del
ejemplo 1 de un circuito detector con salida cero-saturación'
    R1_1 = CompCeroSat.Valor_R1_1.Text * 1000
    R2_1 = CompCeroSat.Valor_R2_1.Text * 1000
    Vi_1 = CompCeroSat.Valor_Vi_1.Text
    Vref_1 = CompCeroSat.Valor_Vref_1.Text
    Vu_1 = CompCeroSat.Valor_Vu_1.Text
    VsatPos_1 = CompCeroSat.Valor_VsatPos_1.Text
    VsatNeg_1 = CompCeroSat.Valor_VsatNeg_1.Text

    'Introducimos la condición de que -Vsat NO puede ser mayor que
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
de Vo'
```



```
If VsatPos_1 < VsatNeg_1 Then

    MsgBox "La tensión de saturación -Vsat no puede ser mayor que
+Vsat", vbInformation + vbOKOnly, "AVISO"

    CompCeroSat.Valor_VsatPos_1.Text = ""
    CompCeroSat.Valor_VsatNeg_1.Text = ""
    CompCeroSat.Valor_R3_1.Caption = ""
    CompCeroSat.Valor_Vo_1.Caption = ""
    CompCeroSat.Valor_Saturacion_1.Caption = ""
    CompCeroSat.Conduccion_1.Caption = ""

Else

    'Definimos las ecuaciones de los parametros a calcular y
mostramos el resultado'
    If Vi_1 > -Vref_1 * (R1_1 / R2_1) Then

        Vo_1 = -Vu_1
        CompCeroSat.Conduccion_1.Caption = "D conduce"

    ElseIf Vi_1 < -Vref_1 * (R1_1 / R2_1) Then

        Vo_1 = VsatPos_1
        CompCeroSat.Conduccion_1.Caption = "D no conduce"

    ElseIf Vi_1 = -Vref_1 * (R1_1 / R2_1) Then

        Vo_1 = 0
        CompCeroSat.Conduccion_1.Caption = "D no conduce"

    End If

    R3_1 = ((R1_1 * R2_1) / (R1_1 + R2_1)) / 1000

    CompCeroSat.Valor_R3_1.Caption = R3_1
    CompCeroSat.Valor_Vo_1.Caption = Vo_1

    'Analizamos si la tensión de salida está saturada'
    If Vo_1 >= VsatPos_1 Then

        CompCeroSat.Valor_Vo_1.Caption = VsatPos_1
        CompCeroSat.Valor_Saturacion_1.Caption = "Salida saturada"

    ElseIf Vo_1 <= VsatNeg_1 Then

        CompCeroSat.Valor_Vo_1.Caption = VsatNeg_1
        CompCeroSat.Valor_Saturacion_1.Caption = "Salida saturada"

    Else

        CompCeroSat.Valor_Saturacion_1.Caption = ""
        CompCeroSat.Valor_Vo_1.Caption = Vo_1

    End If

    'Mostramos la representación Vo-Vi'
    If -Vu_1 >= VsatPos_1 Then

        ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s10") =
VsatPos_1
```




```
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t10") =  
VsatPos_1  
  
ElseIf -Vi_1 <= VsatNeg_1 Then  
  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s10") =  
VsatNeg_1  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t10") =  
VsatNeg_1  
  
Else  
  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s10") =  
-Vu_1  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t10") =  
-Vu_1  
  
End If  
  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s6") = -  
Vref_1 * (R1_1 / R2_1)  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t6") = -  
Vref_1 * (R1_1 / R2_1)  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s9") =  
VsatPos_1  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t9") =  
VsatPos_1  
  
End If  
  
End If  
  
'Desactivamos el detector de errores'  
On Error GoTo 0  
  
End Sub  
  
Sub CalcularValoresCompCeroSat2()  
  
'Declaramos las variables de los parámetros del ejemplo 2 de un  
circuito detector con salida cero-saturación'  
Dim R1_2 As Single  
Dim R2_2 As Single  
Dim Vi_2 As Single  
Dim Vref_2 As Single  
Dim Vu_2 As Single  
Dim VsatPos_2 As Single  
Dim VsatNeg_2 As Single  
Dim R3_2 As Single  
Dim Vo_2 As Single  
  
'Mediante esta declaración, conseguimos que nuestra macro no pare de  
ejecutarse aunque se encuentre un error'  
On Error Resume Next  
  
'Nos aseguramos de que se introduzca un valor en todas las casillas'  
'En el caso de las resistencias, su valor no puede ser cero o  
negativo'  
'En el caso de las tensiones, su valor sí puede ser cero o negativo'
```



```
If CompCeroSat.Valor_R1_2.Text <= 0 Or CompCeroSat.Valor_R2_2.Text <= 0 Or CompCeroSat.Valor_R1_2.Text = Empty Or CompCeroSat.Valor_R2_2.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas de las resistencias, y que no sea ni nulo ni negativo", vbExclamation + vbOKOnly, "AVISO"
```

```
CompCeroSat.Valor_R3_2.Caption = ""  
CompCeroSat.Valor_Vo_2.Caption = ""  
CompCeroSat.Valor_Saturacion_2.Caption = ""  
CompCeroSat.Conduccion_2.Caption = ""
```

```
ElseIf CompCeroSat.Valor_Vi_2.Text = Empty Or CompCeroSat.Valor_Vref_2.Text = Empty Or CompCeroSat.Valor_Vu_2.Text = Empty Or CompCeroSat.Valor_VsatPos_2.Text = Empty Or CompCeroSat.Valor_VsatNeg_2.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas del Voltaje", vbExclamation + vbOKOnly, "AVISO"
```

```
CompCeroSat.Valor_R3_2.Caption = ""  
CompCeroSat.Valor_Vo_2.Caption = ""  
CompCeroSat.Valor_Saturacion_2.Caption = ""  
CompCeroSat.Conduccion_2.Caption = ""
```

```
Else
```

```
'Asignamos las variables a los correspondientes parámetros del ejemplo 2 de un circuito detector con salida cero-saturación'
```

```
R1_2 = CompCeroSat.Valor_R1_2.Text * 1000  
R2_2 = CompCeroSat.Valor_R2_2.Text * 1000  
Vi_2 = CompCeroSat.Valor_Vi_2.Text  
Vref_2 = CompCeroSat.Valor_Vref_2.Text  
Vu_2 = CompCeroSat.Valor_Vu_2.Text  
VsatPos_2 = CompCeroSat.Valor_VsatPos_2.Text  
VsatNeg_2 = CompCeroSat.Valor_VsatNeg_2.Text
```

```
'Introducimos la condición de que -Vsat NO puede ser mayor que +Vsat, ya que si se pudiese dar el caso, podría alterar la saturación de Vo'
```

```
If VsatPos_2 < VsatNeg_2 Then
```

```
MsgBox "La tensión de saturación -Vsat no puede ser mayor que +Vsat", vbInformation + vbOKOnly, "AVISO"
```

```
CompCeroSat.Valor_VsatPos_2.Text = ""  
CompCeroSat.Valor_VsatNeg_2.Text = ""  
CompCeroSat.Valor_R3_2.Caption = ""  
CompCeroSat.Valor_Vo_2.Caption = ""  
CompCeroSat.Valor_Saturacion_2.Caption = ""  
CompCeroSat.Conduccion_2.Caption = ""
```

```
Else
```

```
'Definimos las ecuaciones de los parametros a calcular y mostramos el resultado'
```

```
If Vi_2 < -Vref_2 * (R1_2 / R2_2) Then
```

```
Vo_2 = Vu_2  
CompCeroSat.Conduccion_2.Caption = "D conduce"
```



```
ElseIf Vi_2 > -Vref_2 * (R1_2 / R2_2) Then

    Vo_2 = VsatNeg_2
    CompCeroSat.Conduccion_2.Caption = "D no conduce"

ElseIf Vi_2 = -Vref_2 * (R1_2 / R2_2) Then

    Vo_2 = 0
    CompCeroSat.Conduccion_2.Caption = "D no conduce"

End If

R3_2 = ((R1_2 * R2_2) / (R1_2 + R2_2)) / 1000

CompCeroSat.Valor_R3_2.Caption = R3_2
CompCeroSat.Valor_Vo_2.Caption = Vo_2

'Analizamos si la tensión de salida está saturada'
If Vo_2 >= VsatPos_2 Then

    CompCeroSat.Valor_Vo_2.Caption = VsatPos_2
    CompCeroSat.Valor_Saturacion_2.Caption = "Salida saturada"

ElseIf Vo_2 <= VsatNeg_2 Then

    CompCeroSat.Valor_Vo_2.Caption = VsatNeg_2
    CompCeroSat.Valor_Saturacion_2.Caption = "Salida saturada"

Else

    CompCeroSat.Valor_Saturacion_2.Caption = ""
    CompCeroSat.Valor_Vo_2.Caption = Vo_2

End If

'Mostramos la representación Vo-Vi'
If Vu_2 >= VsatPos_2 Then

    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s9") =
VsatPos_2
    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t9") =
VsatPos_2

ElseIf Vu_2 <= VsatNeg_2 Then

    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s9") =
VsatNeg_2
    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t9") =
VsatNeg_2

Else

    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s9") =
Vu_2
    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t9") =
Vu_2

End If
```



```
        ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s6") = -  
Vref_2 * (R1_2 / R2_2)  
        ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t6") = -  
Vref_2 * (R1_2 / R2_2)  
        ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s10") =  
VsatNeg_2  
        ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t10") =  
VsatNeg_2  
  
    End If  
  
End If  
  
'Desactivamos el detector de errores'  
On Error GoTo 0  
  
End Sub  
  
Sub RestaurarValoresCompCeroSat1()  
  
'Vaciamos los valores de cada uno de los parámetros'  
CompCeroSat.Valor_R1_1.Text = ""  
CompCeroSat.Valor_R2_1.Text = ""  
CompCeroSat.Valor_Vi_1.Text = ""  
CompCeroSat.Valor_Vref_1.Text = ""  
CompCeroSat.Valor_Vu_1.Text = ""  
CompCeroSat.Valor_VsatPos_1.Text = ""  
CompCeroSat.Valor_VsatNeg_1.Text = ""  
CompCeroSat.Valor_R3_1.Caption = ""  
CompCeroSat.Valor_Vo_1.Caption = ""  
CompCeroSat.Valor_Saturacion_1.Caption = ""  
CompCeroSat.Conduccion_1.Caption = ""  
  
End Sub  
  
Sub RestaurarValoresCompCeroSat2()  
  
'Vaciamos los valores de cada uno de los parámetros'  
CompCeroSat.Valor_R1_2.Text = ""  
CompCeroSat.Valor_R2_2.Text = ""  
CompCeroSat.Valor_Vi_2.Text = ""  
CompCeroSat.Valor_Vref_2.Text = ""  
CompCeroSat.Valor_Vu_2.Text = ""  
CompCeroSat.Valor_VsatPos_2.Text = ""  
CompCeroSat.Valor_VsatNeg_2.Text = ""  
CompCeroSat.Valor_R3_2.Caption = ""  
CompCeroSat.Valor_Vo_2.Caption = ""  
CompCeroSat.Valor_Saturacion_2.Caption = ""  
CompCeroSat.Conduccion_2.Caption = ""  
  
End Sub  
  
Sub MostrarSchmitt()  
  
'Mostramos la pestaña de la aplicación del circuito disipador de  
Schmitt'  
CompSchmitt.Show  
  
End Sub  
  
Sub MostrarTeoriaCompSchmitt()
```



```
'Mostramos la pestaña de la teoría del circuito disipador de Schmitt'  
TeoriaCompSchmitt.Show  
  
End Sub  
  
Sub MostrarInformacionCompSchmitt ()  
  
'Mostramos la pestaña de la información del circuito disipador de  
Schmitt'  
Informacion3.Show  
  
End Sub  
  
Sub CalcularValoresCompSchmitt ()  
  
'Declaramos las variables de los parámetros de un circuito disipador  
de Schmitt'  
Dim Vi As Single  
Dim a As Single  
Dim b As Single  
Dim VsatPos As Single  
Dim VsatNeg As Single  
Dim t_final As Single  
Dim Vo As Single  
  
'Mediante esta declaración conseguimos que nuestra macro no pare de  
ejecutarse aunque se encuentre un error'  
On Error Resume Next  
  
'Nos aseguramos de que se introduzca un valor en las casillas de las  
tensiones, pudiendo ser cero o negativo'  
'En el caso del valor de t final, su valor no puede ser cero o  
negativo'  
If CompSchmitt.a.Text = Empty Or CompSchmitt.b.Text = Empty Or  
CompSchmitt.Valor_VsatPos.Text = Empty Or  
CompSchmitt.Valor_VsatNeg.Text = Empty Then  
  
    MsgBox "Debe introducir un valor en las casillas del Voltaje",  
vbExclamation + vbOKOnly, "AVISO"  
  
    CompSchmitt.Valor_Vo.Caption = ""  
    CompSchmitt.Valor_Saturacion.Caption = ""  
  
ElseIf CompSchmitt.t_final <= 0 Or CompSchmitt.t_final = Empty Then  
  
    MsgBox "Debe introducir un valor en la casilla del tiempo referido  
a la condición final y que no sea nulo o negativo", vbExclamation +  
vbOKOnly, "AVISO"  
  
    CompSchmitt.Valor_Vo.Caption = ""  
    CompSchmitt.Valor_Saturacion.Caption = ""  
  
Else  
  
'Asignamos las variables a los correspondientes parámetros de un  
circuito disipador de Schmitt'  
a = CompSchmitt.a.Text  
b = CompSchmitt.b.Text  
VsatPos = CompSchmitt.Valor_VsatPos.Text  
VsatNeg = CompSchmitt.Valor_VsatNeg.Text
```



```
t_final = CompSchmitt.t_final.Text

'Introducimos la condición de que -Vsat NO puede ser mayor que
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
de Vo'
If VsatPos < VsatNeg Then

    MsgBox "La tensión de saturación -Vsat no puede ser mayor que
+Vsat", vbInformation + vbOKOnly, "AVISO"

    CompSchmitt.Valor_VsatPos.Text = ""
    CompSchmitt.Valor_VsatNeg.Text = ""
    CompSchmitt.Valor_Vo.Caption = ""
    CompSchmitt.Valor_Saturacion.Caption = ""

Else

    'Definimos las ecuaciones de los parametros a calcular y
mostramos el resultado'
    Vi = a * Sin(t_final * b)

    If Vi < 0 Then

        Vo = VsatPos

    ElseIf Vi > 0 Then

        Vo = VsatNeg

    End If

    CompSchmitt.Valor_Vo.Caption = Vo

    'Analizamos si la tensión de salida está saturada'
    If Vo >= VsatPos Then

        CompSchmitt.Valor_Vo.Caption = VsatPos
        CompSchmitt.Valor_Saturacion.Caption = "Salida saturada"

    ElseIf Vo <= VsatNeg Then

        CompSchmitt.Valor_Vo.Caption = VsatNeg
        CompSchmitt.Valor_Saturacion.Caption = "Salida saturada"

    Else

        CompSchmitt.Valor_Saturacion.Caption = ""
        CompSchmitt.Valor_Vo.Caption = Vo

    End If

    'Mostramos la representación Vi-t y Vo-t'
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("u5") = a
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v5") = b
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("s8:s728") =
Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v8:v728") =
Empty
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("a8:a728") =
Empty
```



```
Empty ActiveWorkbook.Worksheets("Gráfica CosSen").Range("w8:w728") =  
  
Dim j As Integer  
Dim t_max As Integer  
Dim grados1 As Integer  
Dim grados2 As Integer  
  
t_max = (t_final / 0.017453293) + 200  
grados1 = 0  
grados2 = 0  
  
'Definimos el valor de Vi y su instante de tiempo  
correspondiente'  
For j = 8 To t_max  
  
    Worksheets("Gráfica Vo-t").Cells(j, 19).Value = grados1  
    grados1 = grados1 + 1  
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Cells(j,  
22).Value = ActiveWorkbook.Worksheets("Gráfica Vo-t").Cells(j,  
23).Value  
  
Next j  
  
Dim i As Integer  
  
'Definimos también la salida en función del valor de la seña de  
entrada'  
For i = 8 To t_max  
  
    Worksheets("Gráfica CosSen").Cells(i, 1).Value = grados2  
    grados2 = grados2 + 1  
  
    If Worksheets("Gráfica Vo-t").Cells(i, 22).Value > 0 Then  
  
        Worksheets("Gráfica CosSen").Cells(i, 23).Value =  
CompSchmitt.Valor_VsatNeg.Text  
  
    ElseIf Worksheets("Gráfica Vo-t").Cells(i, 22).Value < 0  
Then  
  
        Worksheets("Gráfica CosSen").Cells(i, 23).Value =  
CompSchmitt.Valor_VsatPos.Text  
  
    Else  
  
        Worksheets("Gráfica CosSen").Cells(i, 23).Value = 0  
  
    End If  
  
Next i  
  
End If  
  
End If  
  
'Desactivamos el detector de errores'  
On Error GoTo 0  
  
End Sub
```



```
Sub RestaurarValoresCompSchmitt ()  
  
'Vaciamos los valores de cada uno de los parámetros'  
CompSchmitt.a.Text = ""  
CompSchmitt.b.Text = ""  
CompSchmitt.Valor_VsatPos.Text = ""  
CompSchmitt.Valor_VsatNeg.Text = ""  
CompSchmitt.t_final.Text = ""  
CompSchmitt.Valor_Vo.Caption = ""  
CompSchmitt.Valor_Saturacion.Caption = ""  
  
End Sub
```

7.2.3.3. Circuitos limitadores (ModCircLim)

```
Sub MostrarDiodoNormal ()  
  
'Mostramos la pestaña de la aplicación del circuito limitador con  
diodo normal'  
LimitadorDiodoNormal.Show  
  
End Sub  
  
Sub MostrarTeoriaLimDiodoNormal1 ()  
  
'Mostramos la pestaña de la teoría del ejemplo 1 del circuito  
limitador con diodo normal'  
TeoriaLimSimple1.Show  
  
End Sub  
  
Sub MostrarTeoriaLimDiodoNormal2 ()  
  
'Mostramos la pestaña de la teoría del ejemplo 2 del circuito  
limitador con diodo normal'  
TeoriaLimSimple2.Show  
  
End Sub  
  
Sub MostrarInformacionLimDiodoNormal ()  
  
'Mostramos la pestaña de la información del circuito limitador con  
diodo normal'  
Informacion4.Show  
  
End Sub  
  
Sub CalcularValoresDiodoNormal1 ()  
  
'Declaramos las variables de los parámetros del ejemplo 1 de un  
circuito limitador con diodo normal'  
Dim R1_1 As Single  
Dim R2_1 As Single
```




```
Dim Vi_1 As Single
Dim Vu_1 As Single
Dim VsatPos_1 As Single
Dim VsatNeg_1 As Single
Dim Ganancia_1 As Single
Dim Vo_1 As Single
```

```
'Mediante esta declaración, conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next
```

```
'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de las resistencias, su valor no puede ser cero o
negativo'
```

```
'En el caso de las tensiones, su valor sí puede ser cero o negativo'
If LimitadorDiodoNormal.Valor_R1_1.Text <= 0 Or
LimitadorDiodoNormal.Valor_R2_1.Text <= 0 Or
LimitadorDiodoNormal.Valor_R1_1.Text = Empty Or
LimitadorDiodoNormal.Valor_R2_1.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas de las
resistencias, y que no sea ni nulo ni negativo", vbExclamation +
vbOKOnly, "AVISO"
```

```
LimitadorDiodoNormal.Valor_Ganancia_1.Caption = ""
LimitadorDiodoNormal.Valor_Vo_1.Caption = ""
LimitadorDiodoNormal.Valor_Saturacion_1.Caption = ""
LimitadorDiodoNormal.Polarizacion_1.Caption = ""
LimitadorDiodoNormal.Conduccion_1.Caption = ""
LimitadorDiodoNormal.lineal_limitada_1.Caption = ""
```

```
ElseIf LimitadorDiodoNormal.Valor_Vi_1.Text = Empty Or
LimitadorDiodoNormal.Valor_Vu_1.Text = Empty Or
LimitadorDiodoNormal.Valor_VsatPos_1.Text = Empty Or
LimitadorDiodoNormal.Valor_VsatNeg_1.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas del Voltaje",
vbExclamation + vbOKOnly, "AVISO"
```

```
LimitadorDiodoNormal.Valor_Ganancia_1.Caption = ""
LimitadorDiodoNormal.Valor_Vo_1.Caption = ""
LimitadorDiodoNormal.Valor_Saturacion_1.Caption = ""
LimitadorDiodoNormal.Polarizacion_1.Caption = ""
LimitadorDiodoNormal.Conduccion_1.Caption = ""
LimitadorDiodoNormal.lineal_limitada_1.Caption = ""
```

```
Else
```

```
'Asignamos las variables a los correspondientes parámetros del
ejemplo 1 de un circuito limitador con diodo normal'
```

```
R1_1 = LimitadorDiodoNormal.Valor_R1_1.Text * 1000
R2_1 = LimitadorDiodoNormal.Valor_R2_1.Text * 1000
Vi_1 = LimitadorDiodoNormal.Valor_Vi_1.Text
Vu_1 = LimitadorDiodoNormal.Valor_Vu_1.Text
VsatPos_1 = LimitadorDiodoNormal.Valor_VsatPos_1.Text
VsatNeg_1 = LimitadorDiodoNormal.Valor_VsatNeg_1.Text
```

```
'Introducimos la condición de que -Vsat NO puede ser mayor que
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
de Vo'
```

```
If VsatPos_1 < VsatNeg_1 Then
```



```
MsgBox "La tensión de saturación -Vsat no puede ser mayor que  
+Vsat", vbInformation + vbOKOnly, "AVISO"
```

```
LimitadorDiodoNormal.Valor_VsatPos_1.Text = ""  
LimitadorDiodoNormal.Valor_VsatNeg_1.Text = ""  
LimitadorDiodoNormal.Valor_Ganancia_1.Caption = ""  
LimitadorDiodoNormal.Valor_Vo_1.Caption = ""  
LimitadorDiodoNormal.Valor_Saturacion_1.Caption = ""  
LimitadorDiodoNormal.Polarizacion_1.Caption = ""  
LimitadorDiodoNormal.Conduccion_1.Caption = ""  
LimitadorDiodoNormal.lineal_limitada_1.Caption = ""
```

```
Else
```

```
'Definimos las ecuaciones de los parametros a calcular y  
mostramos el resultado'
```

```
If Vi_1 > 0 Then
```

```
LimitadorDiodoNormal.Polarizacion_1.Caption = "D  
polarizado en inversa"  
LimitadorDiodoNormal.Conduccion_1.Caption = ""  
LimitadorDiodoNormal.lineal_limitada_1.Caption = "Salida  
lineal"  
Vo_1 = -Vi_1 * (R2_1 / R1_1)  
Ganancia_1 = -(R2_1 / R1_1)
```

```
ElseIf Vi_1 < 0 Then
```

```
LimitadorDiodoNormal.Polarizacion_1.Caption = "D  
polarizado en directa"  
Vo_1 = -Vi_1 * (R2_1 / R1_1)  
Ganancia_1 = -(R2_1 / R1_1)
```

```
If Vo_1 < Vu_1 Then
```

```
LimitadorDiodoNormal.Conduccion_1.Caption = "No  
conduce"  
LimitadorDiodoNormal.lineal_limitada_1.Caption =  
"Salida lineal"
```

```
ElseIf Vo_1 = Vu_1 Then
```

```
LimitadorDiodoNormal.Conduccion_1.Caption = "Conduce"  
LimitadorDiodoNormal.lineal_limitada_1.Caption =  
"Salida limitada"
```

```
ElseIf Vo_1 >= Vu_1 Then
```

```
LimitadorDiodoNormal.Conduccion_1.Caption = "Conduce"  
LimitadorDiodoNormal.lineal_limitada_1.Caption =  
"Salida limitada"  
Vo_1 = Vu_1
```

```
End If
```

```
ElseIf Vi_1 = 0 Then
```

```
LimitadorDiodoNormal.Polarizacion_1.Caption = ""  
LimitadorDiodoNormal.Conduccion_1.Caption = ""  
LimitadorDiodoNormal.lineal_limitada_1.Caption = ""
```



```
End If

LimitadorDiodoNormal.Valor_Vo_1.Caption = Vo_1
LimitadorDiodoNormal.Valor_Ganancia_1.Caption = Ganancia_1

'Analizamos si la tensión de salida está saturada'
If Vo_1 >= VsatPos_1 Then

    LimitadorDiodoNormal.Valor_Vo_1.Caption = VsatPos_1
    LimitadorDiodoNormal.Valor_Saturacion_1.Caption = "Salida
saturada"
    LimitadorDiodoNormal.lineal_limitada_1.Caption = ""

ElseIf Vo_1 <= VsatNeg_1 Then

    LimitadorDiodoNormal.Valor_Vo_1.Caption = VsatNeg_1
    LimitadorDiodoNormal.Valor_Saturacion_1.Caption = "Salida
saturada"
    LimitadorDiodoNormal.lineal_limitada_1.Caption = ""

Else

    LimitadorDiodoNormal.Valor_Saturacion_1.Caption = ""
    LimitadorDiodoNormal.Valor_Vo_1.Caption = Vo_1

End If

'Mostramos la representación Vo-Vi'

ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c5") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c16") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c17") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c19") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d19") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c20") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d20") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c21") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d21") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c24") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c25") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c27") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c28") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c30") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d30") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c31") =
```



```
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d31") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c32") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d32") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c35") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c36") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c33") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d33") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c38") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c39") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c41") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d41") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c42") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d42") =  
Empty  
  
'Calculamos los valores de Vi para los que Vo satura, ya que  
serán necesarios para obtener la gráfica'  
Vi_Neg_1 = -Vu_1 * (R1_1 / R2_1)  
Vi_Pos_1 = VsatNeg_1 / Ganancia_1  
  
If Vu_1 >= VsatPos_1 Then  
  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c19") =  
VsatPos_1  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d19") =  
VsatPos_1  
  
ElseIf Vu_1 <= VsatNeg_1 Then  
  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c19") =  
VsatNeg_1  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d19") =  
VsatNeg_1  
  
Else  
  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c19") =  
Vu_1  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d19") =  
Vu_1  
  
End If  
  
'Definimos los puntos necesarios para obtener nuestra gráfica'  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c5") =  
Ganancia_1  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c16") =  
Vi_Neg_1
```



```
        ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c17") =  
Vi_Pos_1  
        ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c20") =  
VsatNeg_1  
        ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d20") =  
VsatNeg_1  
        ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c21") =  
VsatPos_1  
        ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d21") =  
VsatPos_1  
  
    End If  
  
End If  
  
'Desactivamos el detector de errores'  
On Error GoTo 0  
  
End Sub  
  
Sub CalcularValoresDiodoNormal2()  
  
'Declaramos las variables de los parámetros del ejemplo 2 de un  
circuito limitador con diodo normal'  
Dim R1_2 As Single  
Dim R2_2 As Single  
Dim Vi_2 As Single  
Dim Vu_2 As Single  
Dim VsatPos_2 As Single  
Dim VsatNeg_2 As Single  
Dim Ganancia_2 As Single  
Dim Vo_2 As Single  
  
'Mediante esta declaración, conseguimos que nuestra macro no pare de  
ejecutarse aunque se encuentre un error'  
On Error Resume Next  
  
'Nos aseguramos de que se introduzca un valor en todas las casillas'  
'En el caso de las resistencias, su valor no puede ser cero o  
negativo'  
'En el caso de las tensiones, su valor sí puede ser cero o negativo'  
If LimitadorDiodoNormal.Valor_R1_2.Text <= 0 Or  
LimitadorDiodoNormal.Valor_R2_2.Text <= 0 Or  
LimitadorDiodoNormal.Valor_R1_2.Text = Empty Or  
LimitadorDiodoNormal.Valor_R2_2.Text = Empty Then  
  
    MsgBox "Debe introducir un valor en las casillas de las  
resistencias, y que no sea ni nulo ni negativo", vbExclamation +  
vbOKOnly, "AVISO"  
  
    LimitadorDiodoNormal.Valor_Ganancia_2.Caption = ""  
    LimitadorDiodoNormal.Valor_Vo_2.Caption = ""  
    LimitadorDiodoNormal.Valor_Saturacion_2.Caption = ""  
    LimitadorDiodoNormal.Polarizacion_2.Caption = ""  
    LimitadorDiodoNormal.Conduccion_2.Caption = ""  
    LimitadorDiodoNormal.lineal_limitada_2.Caption = ""  
  
ElseIf LimitadorDiodoNormal.Valor_Vi_2.Text = Empty Or  
LimitadorDiodoNormal.Valor_Vu_2.Text = Empty Or  
LimitadorDiodoNormal.Valor_VsatPos_2.Text = Empty Or  
LimitadorDiodoNormal.Valor_VsatNeg_2.Text = Empty Then
```



```
MsgBox "Debe introducir un valor en las casillas del Voltaje",  
vbExclamation + vbOKOnly, "AVISO"
```

```
LimitadorDiodoNormal.Valor_Ganancia_2.Caption = ""  
LimitadorDiodoNormal.Valor_Vo_2.Caption = ""  
LimitadorDiodoNormal.Valor_Saturacion_2.Caption = ""  
LimitadorDiodoNormal.Polarizacion_2.Caption = ""  
LimitadorDiodoNormal.Conduccion_2.Caption = ""  
LimitadorDiodoNormal.lineal_limitada_2.Caption = ""
```

Else

```
'Asignamos las variables a los correspondientes parámetros del  
ejemplo 2 de un circuito limitador con diodo normal'
```

```
R1_2 = LimitadorDiodoNormal.Valor_R1_2.Text * 1000  
R2_2 = LimitadorDiodoNormal.Valor_R2_2.Text * 1000  
Vi_2 = LimitadorDiodoNormal.Valor_Vi_2.Text  
Vu_2 = LimitadorDiodoNormal.Valor_Vu_2.Text  
VsatPos_2 = LimitadorDiodoNormal.Valor_VsatPos_2.Text  
VsatNeg_2 = LimitadorDiodoNormal.Valor_VsatNeg_2.Text
```

```
'Introducimos la condición de que -Vsat NO puede ser mayor que  
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación  
de Vo'
```

```
If VsatPos_2 < VsatNeg_2 Then
```

```
MsgBox "La tensión de saturación -Vsat no puede ser mayor que  
+Vsat", vbInformation + vbOKOnly, "AVISO"
```

```
LimitadorDiodoNormal.Valor_VsatPos_2.Text = ""  
LimitadorDiodoNormal.Valor_VsatNeg_2.Text = ""  
LimitadorDiodoNormal.Valor_Ganancia_2.Caption = ""  
LimitadorDiodoNormal.Valor_Vo_2.Caption = ""  
LimitadorDiodoNormal.Valor_Saturacion_2.Caption = ""  
LimitadorDiodoNormal.Polarizacion_2.Caption = ""  
LimitadorDiodoNormal.Conduccion_2.Caption = ""  
LimitadorDiodoNormal.lineal_limitada_2.Caption = ""
```

Else

```
'Definimos las ecuaciones de los parametros a calcular y  
mostramos el resultado'
```

```
If Vi_2 < 0 Then
```

```
LimitadorDiodoNormal.Polarizacion_2.Caption = "D  
polarizado en inversa"  
LimitadorDiodoNormal.Conduccion_2.Caption = ""  
LimitadorDiodoNormal.lineal_limitada_2.Caption = "Salida  
lineal"
```

```
Vo_2 = -Vi_2 * (R2_2 / R1_2)  
Ganancia_2 = -(R2_2 / R1_2)
```

```
ElseIf Vi_2 > 0 Then
```

```
LimitadorDiodoNormal.Polarizacion_2.Caption = "D  
polarizado en directa"  
Vo_2 = -Vi_2 * (R2_2 / R1_2)  
Ganancia_2 = -(R2_2 / R1_2)
```

```
If Vo_2 > -Vu_2 Then
```



```
LimitadorDiodoNormal.Conduccion_2.Caption = "No  
conduce"  
LimitadorDiodoNormal.lineal_limitada_2.Caption =  
"Salida lineal"  
  
ElseIf Vo_2 = -Vu_2 Then  
  
LimitadorDiodoNormal.Conduccion_2.Caption = "Conduce"  
LimitadorDiodoNormal.lineal_limitada_2.Caption =  
"Salida limitada"  
  
ElseIf Vo_2 <= -Vu_2 Then  
  
LimitadorDiodoNormal.Conduccion_2.Caption = "Conduce"  
LimitadorDiodoNormal.lineal_limitada_2.Caption =  
"Salida limitada"  
Vo_2 = -Vu_2  
  
End If  
  
ElseIf Vi_2 = 0 Then  
  
LimitadorDiodoNormal.Polarizacion_2.Caption = ""  
LimitadorDiodoNormal.Conduccion_2.Caption = ""  
LimitadorDiodoNormal.lineal_limitada_2.Caption = ""  
  
End If  
  
LimitadorDiodoNormal.Valor_Vo_2.Caption = Vo_2  
LimitadorDiodoNormal.Valor_Ganancia_2.Caption = Ganancia_2  
  
'Analizamos si la tensión de salida está saturada'  
If Vo_2 >= VsatPos_2 Then  
  
LimitadorDiodoNormal.Valor_Vo_2.Caption = VsatPos_2  
LimitadorDiodoNormal.Valor_Saturacion_2.Caption = "Salida  
saturada"  
LimitadorDiodoNormal.lineal_limitada_2.Caption = ""  
  
ElseIf Vo_2 <= VsatNeg_2 Then  
  
LimitadorDiodoNormal.Valor_Vo_2.Caption = VsatNeg_2  
LimitadorDiodoNormal.Valor_Saturacion_2.Caption = "Salida  
saturada"  
LimitadorDiodoNormal.lineal_limitada_2.Caption = ""  
  
Else  
  
LimitadorDiodoNormal.Valor_Saturacion_2.Caption = ""  
LimitadorDiodoNormal.Valor_Vo_2.Caption = Vo_2  
  
End If  
  
'Mostramos la representación Vo-Vi'  
  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c5") = Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c16") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c17") =  
Empty
```



```
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c19") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d19") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c20") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d20") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c21") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d21") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c24") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c25") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c27") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c28") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c30") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d30") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c31") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d31") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c32") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d32") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c35") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c36") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c33") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d33") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c38") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c39") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c41") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d41") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c42") =
Empty ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d42") =
```

'Calculamos los valores de Vi para los que Vo satura, ya que serán necesarios para obtener la gráfica'

```
Vi_Neg_2 = (VsatPos_2 / Ganancia_2)
```

```
Vi_Pos_2 = Vu_2 * (R1_2 / R2_2)
```

```
If -Vu_2 >= VsatPos_2 Then
```




```
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c22") =  
VsatPos_2  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d22") =  
VsatPos_2  
  
ElseIf -Vu_2 <= VsatNeg_2 Then  
  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c22") =  
VsatNeg_2  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d22") =  
VsatNeg_2  
  
Else  
  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c22") =  
-Vu_2  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d22") =  
-Vu_2  
  
End If  
  
'Definimos los puntos necesarios para obtener nuestra gráfica'  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c5") =  
Ganancia_2  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c24") =  
Vi_Neg_2  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c25") =  
Vi_Pos_2  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c20") =  
VsatNeg_2  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d20") =  
VsatNeg_2  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c21") =  
VsatPos_2  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d21") =  
VsatPos_2  
  
End If  
  
End If  
  
'Desactivamos el detector de errores'  
On Error GoTo 0  
  
End Sub  
  
Sub RestaurarValoresDiodoNormal1()  
  
'Vaciamos los valores de cada uno de los parámetros'  
LimitadorDiodoNormal.Valor_R1_1.Text = ""  
LimitadorDiodoNormal.Valor_R2_1.Text = ""  
LimitadorDiodoNormal.Valor_Vi_1.Text = ""  
LimitadorDiodoNormal.Valor_Vu_1.Text = ""  
LimitadorDiodoNormal.Polarizacion_1.Caption = ""  
LimitadorDiodoNormal.Conduccion_1.Caption = ""  
LimitadorDiodoNormal.lineal_limitada_1.Caption = ""  
LimitadorDiodoNormal.Valor_VsatPos_1.Text = ""  
LimitadorDiodoNormal.Valor_VsatNeg_1.Text = ""  
LimitadorDiodoNormal.Valor_Ganancia_1.Caption = ""  
LimitadorDiodoNormal.Valor_Vo_1.Caption = ""
```



```
LimitadorDiodoNormal.Valor_Saturacion_1.Caption = ""
```

```
End Sub
```

```
Sub RestaurarValoresDiodoNormal2 ()
```

```
'Vaciamos los valores de cada uno de los parámetros'
```

```
LimitadorDiodoNormal.Valor_R1_2.Text = ""
```

```
LimitadorDiodoNormal.Valor_R2_2.Text = ""
```

```
LimitadorDiodoNormal.Valor_Vi_2.Text = ""
```

```
LimitadorDiodoNormal.Valor_Vu_2.Text = ""
```

```
LimitadorDiodoNormal.Polarizacion_2.Caption = ""
```

```
LimitadorDiodoNormal.Conduccion_2.Caption = ""
```

```
LimitadorDiodoNormal.lineal_limitada_2.Caption = ""
```

```
LimitadorDiodoNormal.Valor_VsatPos_2.Text = ""
```

```
LimitadorDiodoNormal.Valor_VsatNeg_2.Text = ""
```

```
LimitadorDiodoNormal.Valor_Ganancia_2.Caption = ""
```

```
LimitadorDiodoNormal.Valor_Vo_2.Caption = ""
```

```
LimitadorDiodoNormal.Valor_Saturacion_2.Caption = ""
```

```
End Sub
```

```
Sub MostrarDiodoNormalDiodoZener ()
```

```
'Mostramos la pestaña de la aplicación del circuito limitador con  
diodo normal y diodo zener en serie'
```

```
LimitadorDiodoNormalDiodoZener.Show
```

```
End Sub
```

```
Sub MostrarTeoriaLimDiodoNormalDiodoZener1 ()
```

```
'Mostramos la pestaña de la teoría del ejemplo 1 del circuito  
limitador con diodo normal y diodo zener en serie'
```

```
TeoriaLimDiodoNormalDiodoZener1.Show
```

```
End Sub
```

```
Sub MostrarTeoriaLimDiodoNormalDiodoZener2 ()
```

```
'Mostramos la pestaña de la teoría del ejemplo 2 del circuito  
limitador con diodo normal y diodo zener en serie'
```

```
TeoriaLimDiodoNormalDiodoZener2.Show
```

```
End Sub
```

```
Sub MostrarInformacionLimDiodoNormalDiodoZener ()
```

```
'Mostramos la pestaña de la información del circuito limitador con  
diodo normal y diodo zener en serie'
```

```
Informacion4.Show
```

```
End Sub
```

```
Sub CalcularValoresDiodoNormalDiodoZener1 ()
```

```
'Declaramos las variables de los parámetros del ejemplo 1 de un  
circuito limitador con diodo normal y diodo zener en serie'
```

```
Dim R1_1 As Single
```

```
Dim R2_1 As Single
```

```
Dim Vi_1 As Single
```



```
Dim Vu_1 As Single
Dim Vz_1 As Single
Dim VsatPos_1 As Single
Dim VsatNeg_1 As Single
Dim Ganancia_1 As Single
Dim Vo_1 As Single
```

```
'Mediante esta declaración conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next
```

```
'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de las resistencias, su valor no puede ser cero o
negativo'
```

```
'En el caso de las tensiones, su valor sí puede ser cero o negativo'
If LimitadorDiodoNormalDiodoZener.Valor_R1_1.Text <= 0 Or
LimitadorDiodoNormalDiodoZener.Valor_R2_1.Text <= 0 Or
LimitadorDiodoNormalDiodoZener.Valor_R1_1.Text = Empty Or
LimitadorDiodoNormalDiodoZener.Valor_R2_1.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas de las
resistencias, y que no sea ni nulo ni negativo", vbExclamation +
vbOKOnly, "AVISO"
```

```
LimitadorDiodoNormalDiodoZener.Valor_Ganancia_1.Caption = ""
LimitadorDiodoNormalDiodoZener.Valor_Vo_1.Caption = ""
LimitadorDiodoNormalDiodoZener.Valor_Saturacion_1.Caption = ""
LimitadorDiodoNormalDiodoZener.PolarizacionD_1.Caption = ""
LimitadorDiodoNormalDiodoZener.PolarizacionDz_1.Caption = ""
LimitadorDiodoNormalDiodoZener.Conduccion_1.Caption = ""
LimitadorDiodoNormalDiodoZener.lineal_limitada_1.Caption = ""
```

```
ElseIf LimitadorDiodoNormalDiodoZener.Valor_Vi_1.Text = Empty Or
LimitadorDiodoNormalDiodoZener.Valor_Vu_1.Text = Empty Or
LimitadorDiodoNormalDiodoZener.Valor_Vz_1.Text = Empty Or
LimitadorDiodoNormalDiodoZener.Valor_VsatPos_1.Text = Empty Or
LimitadorDiodoNormalDiodoZener.Valor_VsatNeg_1.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas del Voltaje",
vbExclamation + vbOKOnly, "AVISO"
```

```
LimitadorDiodoNormalDiodoZener.Valor_Ganancia_1.Caption = ""
LimitadorDiodoNormalDiodoZener.Valor_Vo_1.Caption = ""
LimitadorDiodoNormalDiodoZener.Valor_Saturacion_1.Caption = ""
LimitadorDiodoNormalDiodoZener.PolarizacionD_1.Caption = ""
LimitadorDiodoNormalDiodoZener.PolarizacionDz_1.Caption = ""
LimitadorDiodoNormalDiodoZener.Conduccion_1.Caption = ""
LimitadorDiodoNormalDiodoZener.lineal_limitada_1.Caption = ""
```

```
Else
```

```
'Asignamos las variables a los correspondientes parámetros del
ejemplo 1 de un circuito limitador con diodo normal y diodo zener en
serie'
```

```
R1_1 = LimitadorDiodoNormalDiodoZener.Valor_R1_1.Text * 1000
R2_1 = LimitadorDiodoNormalDiodoZener.Valor_R2_1.Text * 1000
Vi_1 = LimitadorDiodoNormalDiodoZener.Valor_Vi_1.Text
Vu_1 = LimitadorDiodoNormalDiodoZener.Valor_Vu_1.Text
Vz_1 = LimitadorDiodoNormalDiodoZener.Valor_Vz_1.Text
VsatPos_1 = LimitadorDiodoNormalDiodoZener.Valor_VsatPos_1.Text
VsatNeg_1 = LimitadorDiodoNormalDiodoZener.Valor_VsatNeg_1.Text
```



```
'Introducimos la condición de que -Vsat NO puede ser mayor que
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
de Vo'
If VsatPos_1 < VsatNeg_1 Then

    MsgBox "La tensión de saturación -Vsat no puede ser mayor que
+Vsat", vbInformation + vbOKOnly, "AVISO"

    LimitadorDiodoNormalDiodoZener.Valor_VsatPos_1.Text = ""
    LimitadorDiodoNormalDiodoZener.Valor_VsatNeg_1.Text = ""
    LimitadorDiodoNormalDiodoZener.Valor_Ganancia_1.Caption = ""
    LimitadorDiodoNormalDiodoZener.Valor_Vo_1.Caption = ""
    LimitadorDiodoNormalDiodoZener.Valor_Saturacion_1.Caption = ""
    LimitadorDiodoNormalDiodoZener.PolarizacionD_1.Caption = ""
    LimitadorDiodoNormalDiodoZener.PolarizacionDz_1.Caption = ""
    LimitadorDiodoNormalDiodoZener.Conduccion_1.Caption = ""
    LimitadorDiodoNormalDiodoZener.lineal_limitada_1.Caption = ""

ElseIf Vz_1 <= VsatNeg_1 Or Vz_1 >= VsatPos_1 Then

    MsgBox "La tensión Zener no puede superar a la de saturación",
vbInformation + vbOKOnly, "AVISO"

    LimitadorDiodoNormalDiodoZener.Valor_Vz_1.Text = ""
    LimitadorDiodoNormalDiodoZener.Valor_VsatPos_1.Text = ""
    LimitadorDiodoNormalDiodoZener.Valor_VsatNeg_1.Text = ""
    LimitadorDiodoNormalDiodoZener.Valor_Ganancia_1.Caption = ""
    LimitadorDiodoNormalDiodoZener.Valor_Vo_1.Caption = ""
    LimitadorDiodoNormalDiodoZener.Valor_Saturacion_1.Caption = ""
    LimitadorDiodoNormalDiodoZener.PolarizacionD_1.Caption = ""
    LimitadorDiodoNormalDiodoZener.PolarizacionDz_1.Caption = ""
    LimitadorDiodoNormalDiodoZener.Conduccion_1.Caption = ""
    LimitadorDiodoNormalDiodoZener.lineal_limitada_1.Caption = ""

Else

    'Definimos las ecuaciones de los parametros a calcular y
mostramos el resultado'
    If Vi_1 > 0 Then

        LimitadorDiodoNormalDiodoZener.PolarizacionD_1.Caption =
"D polarizado en inversa"
        LimitadorDiodoNormalDiodoZener.PolarizacionDz_1.Caption =
"Dz polarizado en directa"
        LimitadorDiodoNormalDiodoZener.Conduccion_1.Caption = "No
conducción"
        LimitadorDiodoNormalDiodoZener.lineal_limitada_1.Caption =
"Salida lineal"
        Vo_1 = -Vi_1 * (R2_1 / R1_1)
        Ganancia_1 = -(R2_1 / R1_1)

    ElseIf Vi_1 < 0 Then

        LimitadorDiodoNormalDiodoZener.PolarizacionD_1.Caption =
"D polarizado en directa"
        LimitadorDiodoNormalDiodoZener.PolarizacionDz_1.Caption =
"Dz polarizado en inversa"
        Vo_1 = -Vi_1 * (R2_1 / R1_1)
        Ganancia_1 = -(R2_1 / R1_1)

    End If

End If
```



```

    If Vo_1 < (Vu_1 + Vz_1) Then

        LimitadorDiodoNormalDiodoZener.PolarizacionD_1.Caption
        = "D polarizado en directa"

        LimitadorDiodoNormalDiodoZener.PolarizacionDz_1.Caption = "Dz no en
        zona Zener"

        LimitadorDiodoNormalDiodoZener.Conduccion_1.Caption =
        "No conducción"

        LimitadorDiodoNormalDiodoZener.lineal_limitada_1.Caption = "Salida
        lineal"

        ElseIf Vo_1 = (Vu_1 + Vz_1) Then

            LimitadorDiodoNormalDiodoZener.PolarizacionD_1.Caption
            = "D polarizado en directa"

            LimitadorDiodoNormalDiodoZener.PolarizacionDz_1.Caption = "Dz en zona
            Zener"

            LimitadorDiodoNormalDiodoZener.Conduccion_1.Caption =
            "Conducción"

            LimitadorDiodoNormalDiodoZener.lineal_limitada_1.Caption = "Salida
            limitada"

            ElseIf Vo_1 > (Vu_1 + Vz_1) Then

                LimitadorDiodoNormalDiodoZener.PolarizacionD_1.Caption
                = "D polarizado en directa"

                LimitadorDiodoNormalDiodoZener.PolarizacionDz_1.Caption = "Dz en zona
                Zener"

                LimitadorDiodoNormalDiodoZener.Conduccion_1.Caption =
                "Conducción"

                LimitadorDiodoNormalDiodoZener.lineal_limitada_1.Caption = "Salida
                limitada"

                Vo_1 = Vu_1 + Vz_1

            End If

        ElseIf Vi_1 = 0 Then

            LimitadorDiodoNormalDiodoZener.PolarizacionD_1.Caption =
            ""

            LimitadorDiodoNormalDiodoZener.PolarizacionDz_1.Caption =
            ""

            LimitadorDiodoNormalDiodoZener.Conduccion_1.Caption = ""
            LimitadorDiodoNormalDiodoZener.lineal_limitada_1.Caption =
            ""

        End If

        LimitadorDiodoNormalDiodoZener.Valor_Vo_1.Caption = Vo_1
        LimitadorDiodoNormalDiodoZener.Valor_Ganancia_1.Caption =
        Ganancia_1

        'Analizamos si la tensión de salida está saturada'
        If Vo_1 >= VsatPos_1 Then

```



```
LimitadorDiodoNormalDiodoZener.Valor_Vo_1.Caption =
VsatPos_1
LimitadorDiodoNormalDiodoZener.Valor_Saturacion_1.Caption
= "Salida saturada"
LimitadorDiodoNormalDiodoZener.lineal_limitada_1.Caption =
""

ElseIf Vo_1 <= VsatNeg_1 Then

LimitadorDiodoNormalDiodoZener.Valor_Vo_1.Caption =
VsatNeg_1
LimitadorDiodoNormalDiodoZener.Valor_Saturacion_1.Caption
= "Salida saturada"
LimitadorDiodoNormalDiodoZener.lineal_limitada_1.Caption =
""

Else

LimitadorDiodoNormalDiodoZener.Valor_Saturacion_1.Caption
= ""
LimitadorDiodoNormalDiodoZener.Valor_Vo_1.Caption = Vo_1

End If

'Mostramos la representación Vo-Vi'

ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c5") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c16") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c17") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c19") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d19") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c20") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d20") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c21") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d21") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c24") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c25") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c27") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c28") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c30") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d30") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c31") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d31") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c32") =
Empty
```



```
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d32") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c35") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c36") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c33") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d33") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c38") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c39") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c41") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d41") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c42") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d42") =  
Empty  
  
'Calculamos los valores de Vi para los que Vo satura, ya que  
serán necesarios para obtener la gráfica'  
Vi_Neg_1 = -(Vu_1 + Vz_1) * (R1_1 / R2_1)  
Vi_Pos_1 = VsatNeg_1 / Ganancia_1  
  
If Vu_1 + Vz_1 >= VsatPos_1 Then  
  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c30") =  
VsatPos_1  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d30") =  
VsatPos_1  
  
ElseIf Vu_1 + Vz_1 <= VsatNeg_1 Then  
  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c30") =  
VsatNeg_1  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d30") =  
VsatNeg_1  
Else  
  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c30") =  
Vu_1 + Vz_1  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d30") =  
Vu_1 + Vz_1  
  
End If  
  
'Definimos los puntos necesarios para obtener nuestra gráfica'  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c5") =  
Ganancia_1  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c27") =  
Vi_Neg_1  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c28") =  
Vi_Pos_1  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c31") =  
VsatNeg_1  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d31") =  
VsatNeg_1
```



```
        ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c32") =  
VsatPos_1  
        ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d32") =  
VsatPos_1  
  
        End If  
  
End If  
  
'Desactivamos el detector de errores'  
On Error GoTo 0  
  
End Sub  
  
Sub CalcularValoresDiodoNormalDiodoZener2()  
  
'Declaramos las variables de los parámetros del ejemplo 2 de un  
circuito limitador con diodo normal y diodo zener en serie'  
Dim R1_2 As Single  
Dim R2_2 As Single  
Dim Vi_2 As Single  
Dim Vu_2 As Single  
Dim Vz_2 As Single  
Dim VsatPos_2 As Single  
Dim VsatNeg_2 As Single  
Dim Ganancia_2 As Single  
Dim Vo_2 As Single  
  
'Mediante esta declaración, conseguimos que nuestra macro no pare de  
ejecutarse aunque se encuentre un error'  
On Error Resume Next  
  
'Nos aseguramos de que se introduzca un valor en todas las casillas'  
'En el caso de las resistencias, su valor no puede ser cero o  
negativo'  
'En el caso de las tensiones, su valor sí puede ser cero o negativo'  
If LimitadorDiodoNormalDiodoZener.Valor_R1_2.Text <= 0 Or  
LimitadorDiodoNormalDiodoZener.Valor_R2_2.Text <= 0 Or  
LimitadorDiodoNormalDiodoZener.Valor_R1_2.Text = Empty Or  
LimitadorDiodoNormalDiodoZener.Valor_R2_2.Text = Empty Then  
  
    MsgBox "Debe introducir un valor en las casillas de las  
resistencias, y que no sea ni nulo ni negativo", vbExclamation +  
vbOKOnly, "AVISO"  
  
    LimitadorDiodoNormalDiodoZener.Valor_Ganancia_2.Caption = ""  
    LimitadorDiodoNormalDiodoZener.Valor_Vo_2.Caption = ""  
    LimitadorDiodoNormalDiodoZener.Valor_Saturacion_2.Caption = ""  
    LimitadorDiodoNormalDiodoZener.PolarizacionD_2.Caption = ""  
    LimitadorDiodoNormalDiodoZener.PolarizacionDz_2.Caption = ""  
    LimitadorDiodoNormalDiodoZener.Conduccion_2.Caption = ""  
    LimitadorDiodoNormalDiodoZener.lineal_limitada_2.Caption = ""  
  
ElseIf LimitadorDiodoNormalDiodoZener.Valor_Vi_2.Text = Empty Or  
LimitadorDiodoNormalDiodoZener.Valor_Vu_2.Text = Empty Or  
LimitadorDiodoNormalDiodoZener.Valor_Vz_2.Text = Empty Or  
LimitadorDiodoNormalDiodoZener.Valor_VsatPos_2.Text = Empty Or  
LimitadorDiodoNormalDiodoZener.Valor_VsatNeg_2.Text = Empty Then  
  
    MsgBox "Debe introducir un valor en las casillas del Voltaje",  
vbExclamation + vbOKOnly, "AVISO"
```




```
LimitadorDiodoNormalDiodoZener.Valor_Ganancia_2.Caption = ""
LimitadorDiodoNormalDiodoZener.Valor_Vo_2.Caption = ""
LimitadorDiodoNormalDiodoZener.Valor_Saturacion_2.Caption = ""
LimitadorDiodoNormalDiodoZener.PolarizacionD_2.Caption = ""
LimitadorDiodoNormalDiodoZener.PolarizacionDz_2.Caption = ""
LimitadorDiodoNormalDiodoZener.Conduccion_2.Caption = ""
LimitadorDiodoNormalDiodoZener.lineal_limitada_2.Caption = ""

Else

    'Asignamos las variables a los correspondientes parámetros del
ejemplo 2 de un circuito limitador con diodo normal y diodo zener en
serie'
    R1_2 = LimitadorDiodoNormalDiodoZener.Valor_R1_2.Text * 1000
    R2_2 = LimitadorDiodoNormalDiodoZener.Valor_R2_2.Text * 1000
    Vi_2 = LimitadorDiodoNormalDiodoZener.Valor_Vi_2.Text
    Vu_2 = LimitadorDiodoNormalDiodoZener.Valor_Vu_2.Text
    Vz_2 = LimitadorDiodoNormalDiodoZener.Valor_Vz_2.Text
    VsatPos_2 = LimitadorDiodoNormalDiodoZener.Valor_VsatPos_2.Text
    VsatNeg_2 = LimitadorDiodoNormalDiodoZener.Valor_VsatNeg_2.Text

    'Introducimos la condición de que -Vsat NO puede ser mayor que
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
de Vo'
    If VsatPos_2 < VsatNeg_2 Then

        MsgBox "La tensión de saturación -Vsat no puede ser mayor que
+Vsat", vbInformation + vbOKOnly, "AVISO"

        LimitadorDiodoNormalDiodoZener.Valor_VsatPos_2.Text = ""
        LimitadorDiodoNormalDiodoZener.Valor_VsatNeg_2.Text = ""
        LimitadorDiodoNormalDiodoZener.Valor_Ganancia_2.Caption = ""
        LimitadorDiodoNormalDiodoZener.Valor_Vo_2.Caption = ""
        LimitadorDiodoNormalDiodoZener.Valor_Saturacion_2.Caption = ""
        LimitadorDiodoNormalDiodoZener.PolarizacionD_2.Caption = ""
        LimitadorDiodoNormalDiodoZener.PolarizacionDz_2.Caption = ""
        LimitadorDiodoNormalDiodoZener.Conduccion_2.Caption = ""
        LimitadorDiodoNormalDiodoZener.lineal_limitada_2.Caption = ""

    ElseIf Vz_2 <= VsatNeg_2 Or Vz_2 >= VsatPos_2 Then

        MsgBox "La tensión zener no puede superar a la de saturación",
vbInformation + vbOKOnly, "AVISO"

        LimitadorDiodoNormalDiodoZener.Valor_Vz_2.Text = ""
        LimitadorDiodoNormalDiodoZener.Valor_VsatPos_2.Text = ""
        LimitadorDiodoNormalDiodoZener.Valor_VsatNeg_2.Text = ""
        LimitadorDiodoNormalDiodoZener.Valor_Ganancia_2.Caption = ""
        LimitadorDiodoNormalDiodoZener.Valor_Vo_2.Caption = ""
        LimitadorDiodoNormalDiodoZener.Valor_Saturacion_2.Caption = ""
        LimitadorDiodoNormalDiodoZener.PolarizacionD_2.Caption = ""
        LimitadorDiodoNormalDiodoZener.PolarizacionDz_2.Caption = ""
        LimitadorDiodoNormalDiodoZener.Conduccion_2.Caption = ""
        LimitadorDiodoNormalDiodoZener.lineal_limitada_2.Caption = ""

    Else

        'Definimos las ecuaciones de los parametros a calcular y
mostramos el resultado'
        If Vi_2 < 0 Then
```



```
LimitadorDiodoNormalDiodoZener.PolarizacionD_2.Caption =  
"D polarizado en inversa"  
LimitadorDiodoNormalDiodoZener.PolarizacionDz_2.Caption =  
"Dz polarizado en directa"  
LimitadorDiodoNormalDiodoZener.Conduccion_2.Caption = "No  
conducción"  
LimitadorDiodoNormalDiodoZener.lineal_limitada_2.Caption =  
"Salida lineal"  
Vo_2 = -Vi_2 * (R2_2 / R1_2)  
Ganancia_2 = -(R2_2 / R1_2)  
  
ElseIf Vi_2 > 0 Then  
  
LimitadorDiodoNormalDiodoZener.PolarizacionD_2.Caption =  
"D polarizado en directa"  
LimitadorDiodoNormalDiodoZener.PolarizacionDz_2.Caption =  
"Dz polarizado en inversa"  
Vo_2 = -Vi_2 * (R2_2 / R1_2)  
Ganancia_2 = -(R2_2 / R1_2)  
  
If Vo_2 > -(Vu_2 + Vz_2) Then  
  
LimitadorDiodoNormalDiodoZener.PolarizacionD_2.Caption  
= "D polarizado en directa"  
  
LimitadorDiodoNormalDiodoZener.PolarizacionDz_2.Caption = "Dz no en  
zona Zener"  
LimitadorDiodoNormalDiodoZener.Conduccion_2.Caption =  
"No conducción"  
  
LimitadorDiodoNormalDiodoZener.lineal_limitada_2.Caption = "Salida  
lineal"  
  
ElseIf Vo_2 = -(Vu_2 + Vz_2) Then  
  
LimitadorDiodoNormalDiodoZener.PolarizacionD_2.Caption  
= "D polarizado en directa"  
  
LimitadorDiodoNormalDiodoZener.PolarizacionDz_2.Caption = "Dz en zona  
Zener"  
LimitadorDiodoNormalDiodoZener.Conduccion_2.Caption =  
"Conducción"  
  
LimitadorDiodoNormalDiodoZener.lineal_limitada_2.Caption = "Salida  
limitada"  
  
ElseIf Vo_2 < -(Vu_2 + Vz_2) Then  
  
LimitadorDiodoNormalDiodoZener.PolarizacionD_2.Caption  
= "D polarizado en directa"  
  
LimitadorDiodoNormalDiodoZener.PolarizacionDz_2.Caption = "Dz en zona  
Zener"  
LimitadorDiodoNormalDiodoZener.Conduccion_2.Caption =  
"Conducción"  
  
LimitadorDiodoNormalDiodoZener.lineal_limitada_2.Caption = "Salida  
limitada"  
Vo_2 = -(Vu_2 + Vz_2)
```



```
End If

ElseIf Vi_2 = 0 Then

    LimitadorDiodoNormalDiodoZener.PolarizacionD_2.Caption =
""
    LimitadorDiodoNormalDiodoZener.PolarizacionDz_2.Caption =
""
    LimitadorDiodoNormalDiodoZener.Conduccion_2.Caption = ""
    LimitadorDiodoNormalDiodoZener.lineal_limitada_2.Caption =
""

End If

LimitadorDiodoNormalDiodoZener.Valor_Vo_2.Caption = Vo_2
LimitadorDiodoNormalDiodoZener.Valor_Ganancia_2.Caption =
Ganancia_2

'Analizamos si la tensión de salida está saturada'
If Vo_2 >= VsatPos_2 Then

    LimitadorDiodoNormalDiodoZener.Valor_Vo_2.Caption =
VsatPos_2
    LimitadorDiodoNormalDiodoZener.Valor_Saturacion_2.Caption
= "Salida saturada"
    LimitadorDiodoNormalDiodoZener.lineal_limitada_2.Caption =
""

ElseIf Vo_2 <= VsatNeg_2 Then

    LimitadorDiodoNormalDiodoZener.Valor_Vo_2.Caption =
VsatNeg_2
    LimitadorDiodoNormalDiodoZener.Valor_Saturacion_2.Caption
= "Salida saturada"
    LimitadorDiodoNormalDiodoZener.lineal_limitada_2.Caption =
""

Else

    LimitadorDiodoNormalDiodoZener.Valor_Saturacion_2.Caption
= ""
    LimitadorDiodoNormalDiodoZener.Valor_Vo_2.Caption = Vo_2

End If

'Mostramos la representacio-Vi'

ActiveWorkbook.Worksheets("Grfica Vo-Vi").Range("c5") = Empty
ActiveWorkbook.Worksheets("Grfica Vo-Vi").Range("c16") =
Empty
ActiveWorkbook.Worksheets("Grfica Vo-Vi").Range("c17") =
Empty
ActiveWorkbook.Worksheets("Grfica Vo-Vi").Range("c19") =
Empty
ActiveWorkbook.Worksheets("Grfica Vo-Vi").Range("d19") =
Empty
ActiveWorkbook.Worksheets("Grfica Vo-Vi").Range("c20") =
Empty
ActiveWorkbook.Worksheets("Grfica Vo-Vi").Range("d20") =
Empty
```



```
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c21") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d21") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c24") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c25") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c27") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c28") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c30") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d30") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c31") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d31") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c32") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d32") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c35") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c36") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c33") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d33") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c38") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c39") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c41") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d41") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c42") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d42") =  
Empty  
  
'Calculamos los valores de Vi para los que Vo satura, ya que  
serán necesarios para obtener la gráfica'  
Vi_Neg_2 = VsatPos_2 / Ganancia_2  
Vi_Pos_2 = (Vu_2 + Vz_2) * (R1_2 / R2_2)  
  
If -(Vu_2 + Vz_2) >= VsatPos_2 Then  
  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c33") =  
VsatPos_2  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d33") =  
VsatPos_2  
  
ElseIf -(Vu_2 + Vz_2) <= VsatNeg_2 Then  
  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c33") =  
VsatNeg_2
```



```
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d33") =  
VsatNeg_2  
  
Else  
  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c33") =  
-(Vu_2 + Vz_2)  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d33") =  
-(Vu_2 + Vz_2)  
  
End If  
  
'Definimos los puntos necesarios para obtener nuestra gráfica'  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c5") =  
Ganancia_2  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c35") =  
Vi_Neg_2  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c36") =  
Vi_Pos_2  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c31") =  
VsatNeg_2  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d31") =  
VsatNeg_2  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c32") =  
VsatPos_2  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d32") =  
VsatPos_2  
  
End If  
  
End If  
  
'Desactivamos el detector de errores'  
On Error GoTo 0  
  
End Sub  
  
Sub RestaurarValoresDiodoNormalDiodoZener1()  
  
'Vaciamos los valores de cada uno de los parámetros'  
LimitadorDiodoNormalDiodoZener.Valor_R1_1.Text = ""  
LimitadorDiodoNormalDiodoZener.Valor_R2_1.Text = ""  
LimitadorDiodoNormalDiodoZener.Valor_Vi_1.Text = ""  
LimitadorDiodoNormalDiodoZener.Valor_Vu_1.Text = ""  
LimitadorDiodoNormalDiodoZener.Valor_Vz_1.Text = ""  
LimitadorDiodoNormalDiodoZener.PolarizacionD_1.Caption = ""  
LimitadorDiodoNormalDiodoZener.PolarizacionDz_1.Caption = ""  
LimitadorDiodoNormalDiodoZener.Conduccion_1.Caption = ""  
LimitadorDiodoNormalDiodoZener.lineal_limitada_1.Caption = ""  
LimitadorDiodoNormalDiodoZener.Valor_VsatPos_1.Text = ""  
LimitadorDiodoNormalDiodoZener.Valor_VsatNeg_1.Text = ""  
LimitadorDiodoNormalDiodoZener.Valor_Ganancia_1.Caption = ""  
LimitadorDiodoNormalDiodoZener.Valor_Vo_1.Caption = ""  
LimitadorDiodoNormalDiodoZener.Valor_Saturacion_1.Caption = ""  
  
End Sub  
  
Sub RestaurarValoresDiodoNormalDiodoZener2()  
  
'Vaciamos los valores de cada uno de los parámetros'
```



```
LimitadorDiodoNormalDiodoZener.Valor_R1_2.Text = ""  
LimitadorDiodoNormalDiodoZener.Valor_R2_2.Text = ""  
LimitadorDiodoNormalDiodoZener.Valor_Vi_2.Text = ""  
LimitadorDiodoNormalDiodoZener.Valor_Vu_2.Text = ""  
LimitadorDiodoNormalDiodoZener.Valor_Vz_2.Text = ""  
LimitadorDiodoNormalDiodoZener.PolarizacionD_2.Caption = ""  
LimitadorDiodoNormalDiodoZener.PolarizacionDz_2.Caption = ""  
LimitadorDiodoNormalDiodoZener.Conduccion_2.Caption = ""  
LimitadorDiodoNormalDiodoZener.lineal_limitada_2.Caption = ""  
LimitadorDiodoNormalDiodoZener.Valor_VsatPos_2.Text = ""  
LimitadorDiodoNormalDiodoZener.Valor_VsatNeg_2.Text = ""  
LimitadorDiodoNormalDiodoZener.Valor_Ganancia_2.Caption = ""  
LimitadorDiodoNormalDiodoZener.Valor_Vo_2.Caption = ""  
LimitadorDiodoNormalDiodoZener.Valor_Saturacion_2.Caption = ""
```

End Sub

Sub MostrarDosZener ()

```
'Mostramos la pestaña de la aplicación del circuito limitador con dos  
diodos zener'
```

```
LimitadorDosZener.Show
```

End Sub

Sub MostrarTeoriaLimDosZener ()

```
'Mostramos la pestaña de la teoría del circuito limitador con dos  
diodos zener'
```

```
TeoriaLimDosZener.Show
```

End Sub

Sub MostrarInformacionLimDosZener ()

```
'Mostramos la pestaña de la información del circuito limitador con dos  
diodos zener'
```

```
Informacion.Show
```

End Sub

Sub CalcularValoresDosZener ()

```
'Declaramos las variables de los parámetros de un circuito limitador  
con diodo normal y diodo zener en serie'
```

```
Dim R1 As Single
```

```
Dim R2 As Single
```

```
Dim Vi As Single
```

```
Dim Vu As Single
```

```
Dim Vz1 As Single
```

```
Dim Vz2 As Single
```

```
Dim VsatPos As Single
```

```
Dim VsatNeg As Single
```

```
Dim Ganancia As Single
```

```
Dim Vo As Single
```

```
'Mediante esta declaración conseguimos que nuestra macro no pare de  
ejecutarse aunque se encuentre un error'
```

```
On Error Resume Next
```

```
'Nos aseguramos de que se introduzca un valor en todas las casillas'
```



```
'En el caso de las resistencias, su valor no puede ser cero o  
negativo'  
'En el caso de las tensiones, su valor sí puede ser cero o negativo'  
If LimitadorDosZener.Valor_R1.Text <= 0 Or  
LimitadorDosZener.Valor_R2.Text <= 0 Or  
LimitadorDosZener.Valor_R1.Text = Empty Or  
LimitadorDosZener.Valor_R2.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas de las  
resistencias, y que no sea ni nulo ni negativo", vbExclamation +  
vbOKOnly, "AVISO"
```

```
LimitadorDosZener.Valor_Ganancia.Caption = ""  
LimitadorDosZener.Valor_Vo.Caption = ""  
LimitadorDosZener.Valor_Saturacion.Caption = ""  
LimitadorDosZener.PolarizacionDz1.Caption = ""  
LimitadorDosZener.PolarizacionDz2.Caption = ""  
LimitadorDosZener.Conduccion.Caption = ""  
LimitadorDosZener.lineal_limitada.Caption = ""
```

```
ElseIf LimitadorDosZener.Valor_Vi.Text = Empty Or  
LimitadorDosZener.Valor_Vu.Text = Empty Or  
LimitadorDosZener.Valor_Vz1.Text = Empty Or  
LimitadorDosZener.Valor_Vz2.Text = Empty Or  
LimitadorDosZener.Valor_VsatPos.Text = Empty Or  
LimitadorDosZener.Valor_VsatNeg.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas del Voltaje",  
vbExclamation + vbOKOnly, "AVISO"
```

```
LimitadorDosZener.Valor_Ganancia.Caption = ""  
LimitadorDosZener.Valor_Vo.Caption = ""  
LimitadorDosZener.Valor_Saturacion.Caption = ""  
LimitadorDosZener.PolarizacionDz1.Caption = ""  
LimitadorDosZener.PolarizacionDz2.Caption = ""  
LimitadorDosZener.Conduccion.Caption = ""  
LimitadorDosZener.lineal_limitada.Caption = ""
```

Else

```
'Asignamos las variables a los correspondientes parámetros de un  
circuito limitador con diodo normal y diodo zener en serie'
```

```
R1 = LimitadorDosZener.Valor_R1.Text * 1000  
R2 = LimitadorDosZener.Valor_R2.Text * 1000  
Vi = LimitadorDosZener.Valor_Vi.Text  
Vu = LimitadorDosZener.Valor_Vu.Text  
Vz1 = LimitadorDosZener.Valor_Vz1.Text  
Vz2 = LimitadorDosZener.Valor_Vz2.Text  
VsatPos = LimitadorDosZener.Valor_VsatPos.Text  
VsatNeg = LimitadorDosZener.Valor_VsatNeg.Text
```

```
'Introducimos la condición de que -Vsat NO puede ser mayor que  
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación  
de Vo'
```

```
If VsatPos < VsatNeg Then
```

```
MsgBox "La tensión de saturación -Vsat no puede ser mayor que  
+Vsat", vbInformation + vbOKOnly, "AVISO"
```

```
LimitadorDosZener.Valor_VsatPos.Text = ""  
LimitadorDosZener.Valor_VsatNeg.Text = ""
```



```
LimitadorDosZener.Valor_Ganancia.Caption = ""
LimitadorDosZener.Valor_Vo.Caption = ""
LimitadorDosZener.Valor_Saturacion.Caption = ""
LimitadorDosZener.PolarizacionDz1.Caption = ""
LimitadorDosZener.PolarizacionDz2.Caption = ""
LimitadorDosZener.Conduccion.Caption = ""
LimitadorDosZener.lineal_limitada.Caption = ""

    ElseIf (Vz1 <= VsatNeg And Vz2 <= VsatNeg) Or (Vz1 >= VsatPos And
Vz2 >= VsatPos) Then

        MsgBox "La tensión zener no puede superar a la de saturación",
vbInformation + vbOKOnly, "AVISO"

        LimitadorDosZener.Valor_Vz1.Text = ""
        LimitadorDosZener.Valor_Vz2.Text = ""
        LimitadorDosZener.Valor_VsatPos.Text = ""
        LimitadorDosZener.Valor_VsatNeg.Text = ""
        LimitadorDosZener.Valor_Ganancia.Caption = ""
        LimitadorDosZener.Valor_Vo.Caption = ""
        LimitadorDosZener.Valor_Saturacion.Caption = ""
        LimitadorDosZener.PolarizacionDz1.Caption = ""
        LimitadorDosZener.PolarizacionDz2.Caption = ""
        LimitadorDosZener.Conduccion.Caption = ""
        LimitadorDosZener.lineal_limitada.Caption = ""

        ElseIf Vz1 <= VsatNeg Or Vz1 >= VsatPos Then

            MsgBox "La tensión zener no puede superar a la de saturación",
vbInformation + vbOKOnly, "AVISO"

            LimitadorDosZener.Valor_Vz1.Text = ""
            LimitadorDosZener.Valor_VsatPos.Text = ""
            LimitadorDosZener.Valor_VsatNeg.Text = ""
            LimitadorDosZener.Valor_Ganancia.Caption = ""
            LimitadorDosZener.Valor_Vo.Caption = ""
            LimitadorDosZener.Valor_Saturacion.Caption = ""
            LimitadorDosZener.PolarizacionDz1.Caption = ""
            LimitadorDosZener.PolarizacionDz2.Caption = ""
            LimitadorDosZener.Conduccion.Caption = ""
            LimitadorDosZener.lineal_limitada.Caption = ""

            ElseIf Vz2 <= VsatNeg Or Vz2 >= VsatPos Then

                MsgBox "La tensión zener no puede superar a la de saturación",
vbInformation + vbOKOnly, "AVISO"

                LimitadorDosZener.Valor_Vz2.Text = ""
                LimitadorDosZener.Valor_VsatPos.Text = ""
                LimitadorDosZener.Valor_VsatNeg.Text = ""
                LimitadorDosZener.Valor_Ganancia.Caption = ""
                LimitadorDosZener.Valor_Vo.Caption = ""
                LimitadorDosZener.Valor_Saturacion.Caption = ""
                LimitadorDosZener.PolarizacionDz1.Caption = ""
                LimitadorDosZener.PolarizacionDz2.Caption = ""
                LimitadorDosZener.Conduccion.Caption = ""
                LimitadorDosZener.lineal_limitada.Caption = ""

            Else
```




```
'Definimos las ecuaciones de los parametros a calcular y
mostramos el resultado'
If Vi > 0 Then

    LimitadorDosZener.PolarizacionDz1.Caption = "Dz1
polarizado en directa"
    LimitadorDosZener.PolarizacionDz2.Caption = "Dz2
polarizado en inversa"
    LimitadorDosZener.Conduccion.Caption = "No conducción"
    Vo = -Vi * (R2 / R1)
    Ganancia = -(R2 / R1)

    If Vo > -(Vu + Vz2) Then

        LimitadorDosZener.PolarizacionDz1.Caption = "Dz1
polarizado en directa"
        LimitadorDosZener.PolarizacionDz2.Caption = "Dz2 no en
zona Zener"
        LimitadorDosZener.Conduccion.Caption = "No conducción"
        LimitadorDosZener.lineal_limitada.Caption = "Salida
lineal"

        ElseIf Vo = -(Vu + Vz2) Then

            LimitadorDosZener.PolarizacionDz1.Caption = "Dz1
polarizado en directa"
            LimitadorDosZener.PolarizacionDz2.Caption = "Dz2 en
zona Zener"
            LimitadorDosZener.Conduccion.Caption = "Conducción"
            LimitadorDosZener.lineal_limitada.Caption = "Salida
limitada"

            ElseIf Vo < -(Vu + Vz2) Then

                LimitadorDosZener.PolarizacionDz1.Caption = "Dz1
polarizado en directa"
                LimitadorDosZener.PolarizacionDz2.Caption = "Dz2 en
zona Zener"
                LimitadorDosZener.Conduccion.Caption = "Conducción"
                LimitadorDosZener.lineal_limitada.Caption = "Salida
limitada"
                Vo = -(Vu + Vz2)

            End If

        ElseIf Vi < 0 Then

            LimitadorDosZener.PolarizacionDz1.Caption = "Dz1
polarizado en inversa"
            LimitadorDosZener.PolarizacionDz2.Caption = "Dz2
polarizado en directa"
            Vo = -Vi * (R2 / R1)
            Ganancia = -(R2 / R1)

            If Vo < (Vu + Vz1) Then

                LimitadorDosZener.PolarizacionDz1.Caption = "Dz1 no en
zona Zener"
                LimitadorDosZener.PolarizacionDz2.Caption = "Dz2
polarizado en directa "
                LimitadorDosZener.Conduccion.Caption = "No conducción"
```



```
LimitadorDosZener.lineal_limitada.Caption = "Salida
lineal"

ElseIf Vo = (Vu + Vz1) Then

LimitadorDosZener.PolarizacionDz1.Caption = "Dz1 en
zona Zener"
LimitadorDosZener.PolarizacionDz2.Caption = "Dz2
polarizado en directa"
LimitadorDosZener.Conduccion.Caption = "Conducción"
LimitadorDosZener.lineal_limitada.Caption = "Salida
limitada"

ElseIf Vo > (Vu + Vz1) Then

LimitadorDosZener.PolarizacionDz1.Caption = "Dz1 en
zona Zener"
LimitadorDosZener.PolarizacionDz2.Caption = "Dz2
polarizado en directa"
LimitadorDosZener.Conduccion.Caption = "Conducción"
LimitadorDosZener.lineal_limitada.Caption = "Salida
limitada"

Vo = Vu + Vz1

End If

ElseIf Vi = 0 Then

LimitadorDosZener.PolarizacionDz1.Caption = ""
LimitadorDosZener.PolarizacionDz2.Caption = ""
LimitadorDosZener.Conduccion.Caption = ""
LimitadorDosZener.lineal_limitada.Caption = ""

End If

LimitadorDosZener.Valor_Vo.Caption = Vo
LimitadorDosZener.Valor_Ganancia.Caption = Ganancia

'Analizamos si la tensión de salida está saturada'
If Vo >= VsatPos Then

LimitadorDosZener.Valor_Vo.Caption = VsatPos
LimitadorDosZener.Valor_Saturacion.Caption = "Salida
saturada"
LimitadorDosZener.lineal_limitada.Caption = ""

ElseIf Vo <= VsatNeg Then

LimitadorDosZener.Valor_Vo.Caption = VsatNeg
LimitadorDosZener.Valor_Saturacion.Caption = "Salida
saturada"
LimitadorDosZener.lineal_limitada.Caption = ""

Else

LimitadorDosZener.Valor_Saturacion.Caption = ""
LimitadorDosZener.Valor_Vo.Caption = Vo

End If

'Mostramos la representación Vo-Vi'
```



```
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c5") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c16") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c17") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c19") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d19") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c20") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d20") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c21") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d21") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c24") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c25") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c27") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c28") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c30") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d30") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c31") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d31") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c32") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d32") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c35") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c36") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c33") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d33") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c38") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c39") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c41") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d41") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c42") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d42") =
```

'Calculamos los valores de Vi para los que Vo satura, ya que serán necesarios para obtener la gráfica'



```
Vi_Neg = -(Vu + Vz1) * (R1 / R2)
Vi_Pos = (Vu + Vz2) * (R1 / R2)

If Vu + Vz1 >= VsatPos Then

    VsatPos    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c41") =
    VsatPos    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d41") =
    VsatPos

ElseIf Vu + Vz1 <= VsatNeg Then

    VsatNeg    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c41") =
    VsatNeg    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d41") =
    VsatNeg

Else

    (Vu + Vz1) ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c41") =
    (Vu + Vz1) ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d41") =

End If

If -(Vu + Vz2) >= VsatPos Then

    VsatPos    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c42") =
    VsatPos    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d42") =
    VsatPos

ElseIf -(Vu + Vz2) <= VsatNeg Then

    VsatNeg    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c42") =
    VsatNeg    ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d42") =
    VsatNeg

Else

    -(Vu + Vz2) ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c42") =
    -(Vu + Vz2) ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d42") =

End If

'Definimos los puntos necesarios para obtener nuestra gráfica'
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c5") =
Ganancia
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c38") =
Vi_Neg
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c39") =
Vi_Pos

End If

End If
```



```
'Desactivamos el detector de errores'  
On Error GoTo 0  
  
End Sub  
  
Sub RestaurarValoresDosZener ()  
  
'Vaciamos los valores de cada uno de los parámetros'  
LimitadorDosZener.Valor_R1.Text = ""  
LimitadorDosZener.Valor_R2.Text = ""  
LimitadorDosZener.Valor_Vi.Text = ""  
LimitadorDosZener.Valor_Vz1.Text = ""  
LimitadorDosZener.Valor_Vu.Text = ""  
LimitadorDosZener.Valor_Vz2.Text = ""  
LimitadorDosZener.PolarizacionDz1.Caption = ""  
LimitadorDosZener.PolarizacionDz2.Caption = ""  
LimitadorDosZener.Conduccion.Caption = ""  
LimitadorDosZener.lineal_limitada.Caption = ""  
LimitadorDosZener.Valor_VsatPos.Text = ""  
LimitadorDosZener.Valor_VsatNeg.Text = ""  
LimitadorDosZener.Valor_Ganancia.Caption = ""  
LimitadorDosZener.Valor_Vo.Caption = ""  
LimitadorDosZener.Valor_Saturacion.Caption = ""  
  
End Sub
```

7.2.3.4. Rectificadores de precisión (ModCircRectPre)

```
Sub MostrarMediaOnda ()  
  
'Mostramos la pestaña de la aplicación del circuito rectificador de  
media onda'  
RectMediaOnda.Show  
  
End Sub  
  
Sub MostrarTeoriaMediaOnda ()  
  
'Mostramos la pestaña de la teoría del circuito rectificador de media  
onda'  
TeoriaMediaOnda.Show  
  
End Sub  
  
Sub MostrarInformacionMediaOnda ()  
  
'Mostramos la pestaña de la información del circuito rectificador de  
media onda'  
Informacion3.Show  
  
End Sub  
  
Sub CalcularValoresMediaOnda ()
```



```
'Declaramos las variables de los parámetros del circuito rectificador  
de media onda'  
Dim R1 As Single  
Dim R2 As Single  
Dim Vi As Single  
Dim a As Single  
Dim b As Single  
Dim Vu As Single  
Dim VsatPos As Single  
Dim VsatNeg As Single  
Dim Vo As Single
```

```
'Mediante esta declaración conseguimos que nuestra macro no pare de  
ejecutarse aunque se encuentre un error'  
On Error Resume Next
```

```
'Nos aseguramos de que se introduzca un valor en todas las casillas'  
'En el caso de las resistencias, su valor no puede ser cero o  
negativo'  
'En el caso de las tensiones, su valor si puede ser cero o negativo'  
If RectMediaOnda.Valor_R1.Text <= 0 Or RectMediaOnda.Valor_R2.Text <=  
0 Or RectMediaOnda.Valor_R1.Text = Empty Or  
RectMediaOnda.Valor_R2.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas de las  
resistencias, y que no sea ni nulo ni negativo", vbExclamation +  
vbOKOnly, "AVISO"
```

```
RectMediaOnda.Conduccion.Caption = ""  
RectMediaOnda.semionda.Caption = ""  
RectMediaOnda.Valor_Vo.Caption = ""  
RectMediaOnda.Valor_Saturacion.Caption = ""
```

```
ElseIf RectMediaOnda.a.Text = Empty Or RectMediaOnda.b.Text = Empty Or  
RectMediaOnda.Valor_Vu.Text = Empty Or  
RectMediaOnda.Valor_VsatPos.Text = Empty Or  
RectMediaOnda.Valor_VsatNeg.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas del Voltaje",  
vbExclamation + vbOKOnly, "AVISO"
```

```
RectMediaOnda.Conduccion.Caption = ""  
RectMediaOnda.semionda.Caption = ""  
RectMediaOnda.Valor_Vo.Caption = ""  
RectMediaOnda.Valor_Saturacion.Caption = ""
```

```
Else
```

```
'Asignamos las variables a los correspondientes parámetros del  
circuito rectificador de media onda'
```

```
R1 = RectMediaOnda.Valor_R1.Text * 1000  
R2 = RectMediaOnda.Valor_R2.Text * 1000  
a = RectMediaOnda.a.Text  
b = RectMediaOnda.b.Text  
Vu = RectMediaOnda.Valor_Vu.Text  
VsatPos = RectMediaOnda.Valor_VsatPos.Text  
VsatNeg = RectMediaOnda.Valor_VsatNeg.Text
```

```
'Introducimos la condición de que -Vsat NO puede ser mayor que  
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación  
Vo'
```



```
If VsatPos < VsatNeg Then

    MsgBox "La tensión de saturación -Vsats no puede ser mayor que
+Vsats", vbInformation + vbOKOnly, "AVISO"

    RectMediaOnda.Valor_VsatPos.Text = ""
    RectMediaOnda.Valor_VsatNeg.Text = ""
    RectMediaOnda.Conduccion.Caption = ""
    RectMediaOnda.semionda.Caption = ""
    RectMediaOnda.Valor_Vo.Caption = ""
    RectMediaOnda.Valor_Saturacion.Caption = ""

Else

    'Definimos las ecuaciones de los parametros a calcular y
mostramos el resultado'
    If a > 0 Then

        Vo = -a * (R2 / R1)
        RectMediaOnda.Conduccion.Caption = "D1 OFF y D2 ON"
        RectMediaOnda.semionda.Caption = "por la semi onda
positiva"

    ElseIf a <= 0 Then

        Vo = a * (R2 / R1)
        RectMediaOnda.Conduccion.Caption = "D1 ON y D2 OFF"
        RectMediaOnda.semionda.Caption = "por la semi onda
negativa"

    End If

    RectMediaOnda.Valor_Vo.Caption = Vo

    'Analizamos si la tensión de salida está saturada'
    If Vo >= Vsats Then

        RectMediaOnda.Valor_Vo.Caption = Vsats
        RectMediaOnda.Valor_Saturacion.Caption = "Salida saturada"

    ElseIf Vo <= VsatsNeg Then

        RectMediaOnda.Valor_Vo.Caption = VsatsNeg
        RectMediaOnda.Valor_Saturacion.Caption = "Salida saturada"

    Else

        RectMediaOnda.Valor_Saturacion.Caption = ""
        RectMediaOnda.Valor_Vo.Caption = Vo

    End If

    'Mostramos la representación Vi-t y Vo-t'
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("u5") = a
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v5") = b
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("s8:s728") =
Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v8:v728") =
Empty

    Dim i As Integer
```



```
Dim k As Integer
Dim j As Single
Dim j1 As Integer

j = 0
j1 = 0

'Pasamos los valores de Vi y t a nuestra hoja de cálculo'
For k = 8 To 410

    Worksheets("Gráfica Vo-t").Cells(k, 19).Value = j1
    j1 = j1 + 1

    ActiveWorkbook.Worksheets("Gráfica Vo-t").Cells(k,
22).Value = ActiveWorkbook.Worksheets("Gráfica Vo-t").Cells(k,
23).Value

Next k

'Obtenemos el valor de Vo para cada instante de tiempo y lo
almacenamos en nuestra hoja de cálculo'
For i = 8 To 410

    Vi = a * Sin(b * j)

    If Vi > 0 Then

        Vo = -a * Sin(b * j) * (R2 / R1)

    ElseIf Vi <= 0 Then

        Vo = 0

    End If

    If Vo >= VsatPos Then

        Vo = VsatPos

    ElseIf Vo <= VsatNeg Then

        Vo = VsatNeg

    End If

    Worksheets("Gráfica Vo-t 2").Cells(i, 4).Value = Vo
    j = j + 0.017453293

Next i

If a > 0 Then

    ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("b5") =
-(R2 / R1) * a

ElseIf a <= 0 Then

    ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("b5") =
(R2 / R1) * a

End If
```




```
'Comprobamos si la salida satura o no'  
If ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("b5") >=  
VsatPos Then  
  
    ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("b5") =  
VsatPos  
  
    ElseIf ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("b5")  
<= VsatNeg Then  
  
        ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("b5") =  
VsatNeg  
  
    End If  
  
End If  
  
End If  
  
'Desactivamos el detector de errores'  
On Error GoTo 0  
  
End Sub  
  
Sub RestaurarValoresMediaOnda ()  
  
'Vaciamos los valores de cada uno de los parámetros'  
RectMediaOnda.Valor_R1.Text = ""  
RectMediaOnda.Valor_R2.Text = ""  
RectMediaOnda.a.Text = ""  
RectMediaOnda.b.Text = ""  
RectMediaOnda.Valor_Vu.Text = ""  
RectMediaOnda.Valor_VsatPos.Text = ""  
RectMediaOnda.Valor_VsatNeg.Text = ""  
RectMediaOnda.Conduccion.Caption = ""  
RectMediaOnda.semionda.Caption = ""  
RectMediaOnda.Valor_Vo.Caption = ""  
RectMediaOnda.Valor_Saturacion.Caption = ""  
  
End Sub  
  
Sub MostrarOndaCompleta ()  
  
'Mostramos la pestaña de la aplicación del circuito rectificador de  
onda completa'  
RectOndaCompleta.Show  
  
End Sub  
  
Sub MostrarTeoriaOndaCompleta1 ()  
  
'Mostramos la pestaña de la teoria del circuito rectificador de onda  
completa'  
TeoriaOndaCompleta1.Show  
  
End Sub  
  
Sub MostrarTeoriaOndaCompleta2 ()
```



```
'Mostramos la pestaña de la teoria del circuito rectificador de onda
completa'
TeoriaOndaCompleta2.Show

End Sub

Sub MostrarInformacionOndaCompleta ()

'Mostramos la pestaña de la información del circuito rectificador de
onda completa'
Informacion5.Show

End Sub

Sub CalcularValoresOndaCompleta1 ()

'Declaramos las variables de los parámetros del ejemplo 1 de un
circuito rectificador de onda completa'
Dim R1_1 As Single
Dim R2_1 As Single
Dim R3_1 As Single
Dim R4_1 As Single
Dim R5_1 As Single
Dim Vi_1 As Single
Dim a1 As Single
Dim b1 As Single
Dim Vu_1 As Single
Dim VsatPos_1 As Single
Dim VsatNeg_1 As Single
Dim Vo_1 As Single
Dim Vo1_1 As Single
Dim Vo2_1 As Single

'Mediante esta declaración conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de las resistencias, su valor no puede ser cero o
negativo'
'En el caso de las tensiones, su valor si puede ser cero o negativo'
If RectOndaCompleta.Valor_R1_1.Text <= 0 Or
RectOndaCompleta.Valor_R2_1.Text <= 0 Or
RectOndaCompleta.Valor_R3_1.Text <= 0 Or
RectOndaCompleta.Valor_R4_1.Text <= 0 Or
RectOndaCompleta.Valor_R5_1.Text <= 0 Or
RectOndaCompleta.Valor_R1_1.Text = Empty Or
RectOndaCompleta.Valor_R2_1.Text = Empty Or
RectOndaCompleta.Valor_R3_1.Text = Empty Or
RectOndaCompleta.Valor_R4_1.Text = Empty Or
RectOndaCompleta.Valor_R5_1.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas de las
resistencias, y que no sea ni nulo ni negativo", vbExclamation +
vbOKOnly, "AVISO"

    RectOndaCompleta.Conduccion_1.Caption = ""
    RectOndaCompleta.Valor_Vo1_1.Caption = ""
    RectOndaCompleta.Valor_Saturacion_Vo1_1.Caption = ""
    RectOndaCompleta.Valor_Vo2_1.Caption = ""
    RectOndaCompleta.Valor_Saturacion_Vo2_1.Caption = ""
```



```
ElseIf RectOndaCompleta.a1.Text = Empty Or RectOndaCompleta.b1.Text =  
Empty Or RectOndaCompleta.Valor_Vu_1.Text = Empty Or  
RectOndaCompleta.Valor_VsatPos_1.Text = Empty Or  
RectOndaCompleta.Valor_VsatNeg_1.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas del Voltaje",  
vbExclamation + vbOKOnly, "AVISO"
```

```
RectOndaCompleta.Conduccion_1.Caption = ""  
RectOndaCompleta.Valor_Vol_1.Caption = ""  
RectOndaCompleta.Valor_Saturacion_Vol_1.Caption = ""  
RectOndaCompleta.Valor_Vo2_1.Caption = ""  
RectOndaCompleta.Valor_Saturacion_Vo2_1.Caption = ""
```

```
Else
```

```
'Asignamos las variables a los correspondientes parámetros del  
ejemplo 1 de un circuito rectificador de onda completa'
```

```
R1_1 = RectOndaCompleta.Valor_R1_1.Text * 1000  
R2_1 = RectOndaCompleta.Valor_R2_1.Text * 1000  
R3_1 = RectOndaCompleta.Valor_R3_1.Text * 1000  
R4_1 = RectOndaCompleta.Valor_R4_1.Text * 1000  
R5_1 = RectOndaCompleta.Valor_R5_1.Text * 1000
```

```
a1 = RectOndaCompleta.a1.Text  
b1 = RectOndaCompleta.b1.Text  
Vu_1 = RectOndaCompleta.Valor_Vu_1.Text  
VsatPos_1 = RectOndaCompleta.Valor_VsatPos_1.Text  
VsatNeg_1 = RectOndaCompleta.Valor_VsatNeg_1.Text
```

```
'Introducimos la condición de que -Vsat NO puede ser mayor que  
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación  
de Vo'
```

```
If VsatPos_1 < VsatNeg_1 Then
```

```
MsgBox "La tensión de saturación -Vsat no puede ser mayor que  
+Vsat", vbInformation + vbOKOnly, "AVISO"
```

```
RectOndaCompleta.Valor_VsatPos_1.Text = ""  
RectOndaCompleta.Valor_VsatNeg_1.Text = ""  
RectOndaCompleta.Conduccion_1.Caption = ""  
RectOndaCompleta.Valor_Vol_1.Caption = ""  
RectOndaCompleta.Valor_Saturacion_Vol_1.Caption = ""  
RectOndaCompleta.Valor_Vo2_1.Caption = ""  
RectOndaCompleta.Valor_Saturacion_Vo2_1.Caption = ""
```

```
Else
```

```
'Definimos las ecuaciones de los parametros a calcular y  
mostramos el resultado'
```

```
If a1 > 0 Then
```

```
RectOndaCompleta.Conduccion_1.Caption = "D1 OFF y D2 ON"
```

```
ElseIf a1 <= 0 Then
```

```
RectOndaCompleta.Conduccion_1.Caption = "D1 ON y D2 OFF"
```

```
End If
```



```
Vo1_1 = Abs(a1 * ((R2_1 * R5_1) / (R1_1 * R3_1) - (R5_1 /  
R4_1)))  
RectOndaCompleta.Valor_Vo1_1.Caption = Vo1_1  
Vo2_1 = Abs(a1 * (R5_1 / R4_1))  
RectOndaCompleta.Valor_Vo2_1.Caption = Vo2_1  
  
'Analizamos si la tensión de salida está saturada'  
If Vo1_1 >= VsatPos_1 Then  
  
    RectOndaCompleta.Valor_Vo1_1.Caption = VsatPos_1  
    RectOndaCompleta.Valor_Saturacion_Vo1_1.Caption = "Salida  
saturada"  
  
    ElseIf Vo1_1 <= VsatNeg_1 Then  
  
        RectOndaCompleta.Valor_Vo1_1.Caption = VsatNeg_1  
        RectOndaCompleta.Valor_Saturacion_Vo1_1.Caption = "Salida  
saturada"  
  
    Else  
  
        RectOndaCompleta.Valor_Saturacion_Vo1_1.Caption = ""  
        RectOndaCompleta.Valor_Vo1_1.Caption = Vo1_1  
  
    End If  
  
    If Vo2_1 >= VsatPos_1 Then  
  
        RectOndaCompleta.Valor_Vo2_1.Caption = VsatPos_1  
        RectOndaCompleta.Valor_Saturacion_Vo2_1.Caption = "Salida  
saturada"  
  
        ElseIf Vo2_1 <= VsatNeg_1 Then  
  
            RectOndaCompleta.Valor_Vo2_1.Caption = VsatNeg_1  
            RectOndaCompleta.Valor_Saturacion_Vo2_1.Caption = "Salida  
saturada"  
  
        Else  
  
            RectOndaCompleta.Valor_Saturacion_Vo2_1.Caption = ""  
            RectOndaCompleta.Valor_Vo2_1.Caption = Vo2_1  
  
        End If  
  
'Mostramos la representación Vi-t y Vo-t'  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("u5") = a1  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v5") = b1  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("s8:s728") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v8:v728") =  
Empty  
  
Dim i As Integer  
Dim k As Integer  
Dim j As Single  
Dim j1 As Integer  
  
j = 0  
j1 = 0
```



```
'Pasamos los valores de Vi y t a nuestra hoja de cálculo'  
For k = 8 To 410  
  
    Worksheets("Gráfica Vo-t").Cells(k, 19).Value = j1  
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Cells(k,  
22).Value = ActiveWorkbook.Worksheets("Gráfica Vo-t").Cells(k,  
23).Value  
    j1 = j1 + 1  
  
Next k  
  
'Obtenemos el valor de Vo para cada instante de tiempo y lo  
almacenamos en nuestra hoja de cálculo'  
For i = 8 To 410  
  
    Vi_1 = a1 * Sin(b1 * j)  
  
    If Vi_1 > 0 Then  
  
        Vo_1 = Abs(a1 * Sin(b1 * j) * ((R2_1 * R5_1) / (R1_1 *  
R3_1) - (R5_1 / R4_1)))  
  
    ElseIf Vi_1 <= 0 Then  
  
        Vo_1 = Abs(a1 * Sin(b1 * j) * (R5_1 / R4_1))  
  
    End If  
  
    If Vo_1 >= VsatPos_1 Then  
  
        Vo_1 = VsatPos_1  
  
    ElseIf Vo_1 <= VsatNeg_1 Then  
  
        Vo_1 = VsatNeg_1  
  
    End If  
  
    Worksheets("Gráfica Vo-t 2").Cells(i, 4).Value = Vo_1  
    j = j + 0.017453293  
  
Next i  
  
    ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("b5") =  
((R2_1 * R5_1) / (R1_1 * R3_1) - (R5_1 / R4_1)) * a1  
    ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("c5") =  
(R5_1 / R4_1) * a1  
  
    'Comprobamos si la salida satura o no'  
    If ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("b5") >=  
VsatPos_1 Then  
  
        ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("b5") =  
VsatPos_1  
  
    ElseIf ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("b5")  
<= VsatNeg_1 Then  
  
        ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("b5") =  
VsatNeg_1
```



```
End If

    If ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("c5") >=
VsatPos_1 Then

        ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("c5") =
VsatPos_1

    ElseIf ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("c5")
<= VsatNeg_1 Then

        ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("c5") =
VsatNeg_1

    End If

End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub CalcularValoresOndaCompleta2()

'Declaramos las variables de los parámetros del ejemplo 1 de un
circuito rectificador de onda completa'
Dim R_2 As Single
Dim R3_2 As Single
Dim Vi_2 As Single
Dim a2 As Single
Dim b2 As Single
Dim VsatPos_2 As Single
Dim VsatNeg_2 As Single
Dim Vo_2 As Single

'Mediante esta declaración conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de las resistencias, su valor no puede ser cero o
negativo'
'En el caso de las tensiones, su valor si puede ser cero o negativo'
If RectOndaCompleta.Valor_R_2.Text <= 0 Or
RectOndaCompleta.Valor_R3_2.Text <= 0 Or
RectOndaCompleta.Valor_R_2.Text = Empty Or
RectOndaCompleta.Valor_R3_2.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas de las
resistencias, y que no sea ni nulo ni negativo", vbExclamation +
vbOKOnly, "AVISO"

    RectOndaCompleta.Conduccion_2.Caption = ""
    RectOndaCompleta.Valor_Vo_2.Caption = ""
    RectOndaCompleta.Valor_Saturacion_Vo_2.Caption = ""
```



```
ElseIf RectOndaCompleta.a2.Text = Empty Or RectOndaCompleta.b2.Text =  
Empty Or RectOndaCompleta.Valor_VsatPos_2.Text = Empty Or  
RectOndaCompleta.Valor_VsatNeg_2.Text = Empty Then
```

```
    MsgBox "Debe introducir un valor en las casillas del Voltaje",  
vbExclamation + vbOKOnly, "AVISO"
```

```
    RectOndaCompleta.Conduccion_2.Caption = ""  
    RectOndaCompleta.Valor_Vo_2.Caption = ""  
    RectOndaCompleta.Valor_Saturacion_Vo_2.Caption = ""
```

```
Else
```

```
    'Asignamos las variables a los correspondientes parámetros del  
ejemplo 1 de un circuito rectificador de onda completa'
```

```
    R_2 = RectOndaCompleta.Valor_R_2.Text * 1000  
    R3_2 = RectOndaCompleta.Valor_R3_2.Text * 1000  
    a2 = RectOndaCompleta.a2.Text  
    b2 = RectOndaCompleta.b2.Text  
    VsatPos_2 = RectOndaCompleta.Valor_VsatPos_2.Text  
    VsatNeg_2 = RectOndaCompleta.Valor_VsatNeg_2.Text
```

```
    'Introducimos la condición de que -Vsat NO puede ser mayor que  
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación  
de Vo'
```

```
    If VsatPos_2 < VsatNeg_2 Then
```

```
        MsgBox "La tensión de saturación -Vsat no puede ser mayor que  
+Vsat", vbInformation + vbOKOnly, "AVISO"
```

```
        RectOndaCompleta.Valor_VsatPos_2.Text = ""  
        RectOndaCompleta.Valor_VsatNeg_2.Text = ""  
        RectOndaCompleta.Conduccion_2.Caption = ""  
        RectOndaCompleta.Valor_Vo_2.Caption = ""  
        RectOndaCompleta.Valor_Saturacion_Vo_2.Caption = ""
```

```
Else
```

```
    'Definimos las ecuaciones de los parametros a calcular y  
mostramos el resultado'
```

```
    If a2 > 0 Then
```

```
        Vo_2 = Abs(a2 * ((R_2 + R3_2) / R_2))  
        RectOndaCompleta.Conduccion_2.Caption = "D1 ON y D2 OFF"
```

```
    ElseIf a2 <= 0 Then
```

```
        Vo_2 = Abs(a2 * ((R_2 + R3_2) / R_2))  
        RectOndaCompleta.Conduccion_2.Caption = "D1 OFF y D2 ON"
```

```
    End If
```

```
    RectOndaCompleta.Valor_Vo_2.Caption = Vo_2
```

```
    'Analizamos si la tensión de salida está saturada'
```

```
    If Vo_2 >= VsatPos_2 Then
```

```
        RectOndaCompleta.Valor_Vo_2.Caption = VsatPos_2  
        RectOndaCompleta.Valor_Saturacion_Vo_2.Caption = "Salida  
saturada"
```



```
ElseIf Vo_2 <= VsatNeg_2 Then

    RectOndaCompleta.Valor_Vo_2.Caption = VsatNeg_2
    RectOndaCompleta.Valor_Saturacion_Vo_2.Caption = "Salida
saturada"

Else

    RectOndaCompleta.Valor_Saturacion_Vo_2.Caption = ""
    RectOndaCompleta.Valor_Vo_2.Caption = Vo_2

End If

'Mostramos la representación Vi-t y Vo-t'
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("u5") = a2
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v5") = b2
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("s8:s728") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v8:v728") =
Empty

Dim i As Integer
Dim k As Integer
Dim j As Single
Dim j1 As Integer

j = 0
j1 = 0

'Pasamos los valores de Vi y t a nuestra hoja de cálculo'
For k = 8 To 410

    Worksheets("Gráfica Vo-t").Cells(k, 19).Value = j1
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Cells(k,
22).Value = ActiveWorkbook.Worksheets("Gráfica Vo-t").Cells(k,
23).Value
    j1 = j1 + 1

Next k

'Obtenemos el valor de Vo para cada instante de tiempo y lo
almacenamos en nuestra hoja de cálculo'
For i = 8 To 410

    Vi_2 = a2 * Sin(b2 * j)

    If Vi_2 > 0 Then

        Vo_2 = Abs(a2 * Sin(b2 * j) * ((R_2 + R3_2) / R_2))

    ElseIf Vi_2 <= 0 Then

        Vo_2 = Abs(a2 * Sin(b2 * j) * ((R_2 + R3_2) / R_2))

    End If

    If Vo_2 >= VsatPos_2 Then

        Vo_2 = VsatPos_2

    ElseIf Vo_2 <= VsatNeg_2 Then
```




```
Vo_2 = VsatNeg_2

End If

Worksheets("Gráfica Vo-t 2").Cells(i, 4).Value = Vo_2
j = j + 0.017453293

Next i

ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("b5") =
((R_2 + R3_2) / R_2) * a2
ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("c5") =
((R_2 + R3_2) / R_2) * a2

'Comprobamos si la salida satura o no'
If ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("b5") >=
VsatPos_2 Then

ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("b5") =
VsatPos_2

ElseIf ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("b5")
<= VsatNeg_2 Then

ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("b5") =
VsatNeg_2

End If

If ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("c5") >=
VsatPos_2 Then

ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("c5") =
VsatPos_2

ElseIf ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("c5")
<= VsatNeg_2 Then

ActiveWorkbook.Worksheets("Gráfica Vo-t 2").Range("c5") =
VsatNeg_2

End If

End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub RestaurarValoresOndaCompleta()

'Vaciamos los valores de cada uno de los parámetros'
RectOndaCompleta.Valor_R1_1.Text = ""
RectOndaCompleta.Valor_R2_1.Text = ""
RectOndaCompleta.Valor_R3_1.Text = ""
RectOndaCompleta.Valor_R4_1.Text = ""
RectOndaCompleta.Valor_R5_1.Text = ""
```



```
RectOndaCompleta.a1.Text = ""  
RectOndaCompleta.b1.Text = ""  
RectOndaCompleta.Valor_Vu_1.Text = ""  
RectOndaCompleta.Conduccion_1.Caption = ""  
RectOndaCompleta.Valor_VsatPos_1.Text = ""  
RectOndaCompleta.Valor_VsatNeg_1.Text = ""  
RectOndaCompleta.Valor_Vo1_1.Caption = ""  
RectOndaCompleta.Valor_Saturacion_Vo1_1.Caption = ""  
RectOndaCompleta.Valor_Vo2_1.Caption = ""  
RectOndaCompleta.Valor_Saturacion_Vo2_1.Caption = ""
```

End Sub

Sub RestaurarValoresOndaCompleta2 ()

```
'Vaciamos los valores de cada uno de los parámetros'  
RectOndaCompleta.Valor_R_2.Text = ""  
RectOndaCompleta.Valor_R3_2.Text = ""  
RectOndaCompleta.a2.Text = ""  
RectOndaCompleta.b2.Text = ""  
RectOndaCompleta.Conduccion_2.Caption = ""  
RectOndaCompleta.Valor_VsatPos_2.Text = ""  
RectOndaCompleta.Valor_VsatNeg_2.Text = ""  
RectOndaCompleta.Valor_Vo_2.Caption = ""  
RectOndaCompleta.Valor_Saturacion_Vo_2.Caption = ""
```

End Sub

Sub MostrarValorMedioValorEficaz ()

```
'Mostramos la pestaña de la aplicación del circuito para obtener el  
valor medio y el valor eficaz'  
ValorMedioValorEficaz.Show
```

End Sub

Sub MostrarTeoriaValorMedioEficaz ()

```
'Mostramos la pestaña de la teoría del circuito para obtener el valor  
medio y el valor eficaz'  
TeoriaMedioEficaz.Show
```

End Sub

Sub MostrarInformacionValorMedioEficaz ()

```
'Mostramos la pestaña de la información del circuito para obtener el  
valor medio y el valor eficaz'  
Informacion3.Show
```

End Sub

Sub CalcularValoresMedioEficaz ()

```
'Declaramos las variables de los parámetros del circuito para obtener  
el valor medio y el valor eficaz'  
Dim R As Single  
Dim max_rizado As Single  
Dim Ganancia As Single  
Dim f_min As Single  
Dim Vi As Single
```



```
Dim a As Single
Dim b As Single
Dim PI As Single
Dim VsatPos As Single
Dim VsatNeg As Single
Dim R5 As Single
Dim C As Single
Dim Vo As Single
Dim Vmed As Single
Dim Vrms As Single

ValorMedioValorEficaz.vmed_vrms.Visible = True
ValorMedioValorEficaz.flecha.Visible = True

'Mediante esta declaración conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de la resistencia, el rizado mínimo, la frecuencia mínima
y la ganancia, su valor no puede ser cero o negativo'
'En el caso de las tensiones, su valor si puede ser cero o negativo'
If ValorMedioValorEficaz.Valor_R.Text <= 0 Or
ValorMedioValorEficaz.Valor_R.Text = Empty Or
ValorMedioValorEficaz.Valor_Ganancia.Text <= 0 Or
ValorMedioValorEficaz.Valor_Ganancia.Text = Empty Or
ValorMedioValorEficaz.max_rizado.Text <= 0 Or
ValorMedioValorEficaz.max_rizado.Text = Empty Or
ValorMedioValorEficaz.Valor_fmin.Text <= 0 Or
ValorMedioValorEficaz.Valor_fmin.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas, y que no sea ni
nulo ni negativo", vbExclamation + vbOKOnly, "AVISO"

    ValorMedioValorEficaz.Valor_R5.Caption = ""
    ValorMedioValorEficaz.Valor_C.Caption = ""
    ValorMedioValorEficaz.Valor_Saturacion.Caption = ""
    ValorMedioValorEficaz.Valor_Vo.Caption = ""
    ValorMedioValorEficaz.vmed_vrms.Caption = ""

ElseIf ValorMedioValorEficaz.a.Text = Empty Or
ValorMedioValorEficaz.b.Text = Empty Or
ValorMedioValorEficaz.Valor_VsatPos.Text = Empty Or
ValorMedioValorEficaz.Valor_VsatNeg.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas del Voltaje",
vbExclamation + vbOKOnly, "AVISO"

    ValorMedioValorEficaz.Valor_R5.Caption = ""
    ValorMedioValorEficaz.Valor_C.Caption = ""
    ValorMedioValorEficaz.Valor_Saturacion.Caption = ""
    ValorMedioValorEficaz.Valor_Vo.Caption = ""
    ValorMedioValorEficaz.vmed_vrms.Caption = ""

Else

    'Asignamos las variables a los correspondientes parámetros del
circuito para obtener el valor medio y el valor eficaz'
    R = ValorMedioValorEficaz.Valor_R.Text * 1000
    max_rizado = ValorMedioValorEficaz.max_rizado.Text / 100
    Ganancia = ValorMedioValorEficaz.Valor_Ganancia.Text
```



```
f_min = ValorMedioValorEficaz.Valor_fmin.Text
a = ValorMedioValorEficaz.a.Text
b = ValorMedioValorEficaz.b.Text
VsatPos = ValorMedioValorEficaz.Valor_VsatPos.Text
VsatNeg = ValorMedioValorEficaz.Valor_VsatNeg.Text

'Introducimos la condición de que -Vsat NO puede ser mayor que
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
de Vo'
If VsatPos < VsatNeg Then

    MsgBox "La tensión de saturación -Vsat no puede ser mayor que
+Vsat", vbInformation + vbOKOnly, "AVISO"

    ValorMedioValorEficaz.Valor_VsatPos.Text = ""
    ValorMedioValorEficaz.Valor_VsatNeg.Text = ""
    ValorMedioValorEficaz.Valor_R5.Caption = ""
    ValorMedioValorEficaz.Valor_C.Caption = ""
    ValorMedioValorEficaz.Valor_Saturacion.Caption = ""
    ValorMedioValorEficaz.Valor_Vo.Caption = ""
    ValorMedioValorEficaz.vmed_vrms.Caption = ""

Else

    'Definimos las ecuaciones de los parametros a calcular y
mostramos el resultado'
    PI = 3.141592654

    Vmed = Abs(0.637 * a)
    Vrms = Abs(0.707 * a)
    R5 = (Ganancia * R) / 1000
    C = (1 / max_rizado) * (1 / (4 * PI * R5 * 1000 * f_min)) *
1000000

    If Ganancia = 1.11 Then

        Vo = Vrms
        ValorMedioValorEficaz.vmed_vrms.Visible = True
        ValorMedioValorEficaz.flecha.Visible = True
        ValorMedioValorEficaz.vmed_vrms.Caption = "Vrms"

    ElseIf Ganancia = 1 Then

        Vo = Vmed
        ValorMedioValorEficaz.vmed_vrms.Visible = True
        ValorMedioValorEficaz.flecha.Visible = True
        ValorMedioValorEficaz.vmed_vrms.Caption = "Vmed"

    Else

        Vo = a
        ValorMedioValorEficaz.vmed_vrms.Visible = False
        ValorMedioValorEficaz.flecha.Visible = False
        ValorMedioValorEficaz.vmed_vrms.Caption = ""

    End If

    ValorMedioValorEficaz.Valor_Vo.Caption = Vo
    ValorMedioValorEficaz.Valor_R5.Caption = R5
    ValorMedioValorEficaz.Valor_C.Caption = C
```



```
'Analizamos si la tensión de salida está saturada'  
If Vo >= VsatPos Then  
  
    ValorMedioValorEficaz.Valor_Vo.Caption = VsatPos  
    ValorMedioValorEficaz.Valor_Saturacion.Caption = "Salida  
saturada"  
  
ElseIf Vo <= VsatNeg Then  
  
    ValorMedioValorEficaz.Valor_Vo.Caption = VsatNeg  
    ValorMedioValorEficaz.Valor_Saturacion.Caption = "Salida  
saturada"  
  
Else  
  
    ValorMedioValorEficaz.Valor_Saturacion.Caption = ""  
    ValorMedioValorEficaz.Valor_Vo.Caption = Vo  
  
End If  
  
'Nos aseguramos de que los valores medio y eficaz no  
sobrepasen los valores de saturación'  
If Vrms >= VsatPos Then  
  
    Vrms = VsatPos  
  
ElseIf Vrms <= VsatNeg Then  
  
    Vrms = VsatNeg  
  
End If  
  
If Vmed >= VsatPos Then  
  
    Vmed = VsatPos  
  
ElseIf Vmed <= VsatNeg Then  
  
    Vmed = VsatNeg  
  
End If  
  
'Mostramos la representación Vi-t y Vo-t'  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("u5") = a  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v5") = b  
  
Dim j1 As Integer  
Dim grados As Integer  
grados = 0  
  
'Pasamos los valores de Vi y t a nuestra hoja de cálculo'  
For j1 = 8 To 410  
  
    Worksheets("Gráfica Vo-t").Cells(j1, 19).Value = grados  
    grados = grados + 1  
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Cells(j1,  
22).Value = ActiveWorkbook.Worksheets("Gráfica Vo-t").Cells(j1,  
23).Value  
  
Next j1
```



```
'Definimos cuando obtenemos el valor eficaz y cuando el valor
medio'
If Vo = Vmed Then

    Vmed
        ActiveWorkbook.Worksheets("Gráfica CosSen").Range("c3") =
    Vmed
        ActiveWorkbook.Worksheets("Gráfica CosSen").Range("d3") =

    ElseIf Vo = Vrms Then

        Vrms
            ActiveWorkbook.Worksheets("Gráfica CosSen").Range("c3") =
        Vrms
            ActiveWorkbook.Worksheets("Gráfica CosSen").Range("d3") =

    Else

        Vo
            ActiveWorkbook.Worksheets("Gráfica CosSen").Range("c3") =
        Vo
            ActiveWorkbook.Worksheets("Gráfica CosSen").Range("d3") =

    End If

    0
        ActiveWorkbook.Worksheets("Gráfica CosSen").Range("c2") =
    7
        ActiveWorkbook.Worksheets("Gráfica CosSen").Range("d2") =

    End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub RestaurarValorMedioEficaz ()

'Vaciamos los valores de cada uno de los parámetros'
ValorMedioValorEficaz.Valor_R.Text = ""
ValorMedioValorEficaz.Valor_Ganancia.Text = ""
ValorMedioValorEficaz.max_rizado.Text = ""
ValorMedioValorEficaz.Valor_fmin.Text = ""
ValorMedioValorEficaz.a.Text = ""
ValorMedioValorEficaz.b.Text = ""
ValorMedioValorEficaz.Valor_VsatPos.Text = ""
ValorMedioValorEficaz.Valor_VsatNeg.Text = ""
ValorMedioValorEficaz.Valor_R5.Caption = ""
ValorMedioValorEficaz.Valor_C.Caption = ""
ValorMedioValorEficaz.Valor_Saturacion.Caption = ""
ValorMedioValorEficaz.Valor_Vo.Caption = ""
ValorMedioValorEficaz.flecha.Caption = ""
ValorMedioValorEficaz.vmed_vrms.Caption = ""

End Sub
```



7.2.3.5. Convertidores corriente-tensión (ModConvCorrTen)

```
Sub MostrarConvCorrTen ()

'Mostramos la pestaña de la aplicación del convertidor corriente-
tensión'
ConvCorrTen.Show

End Sub

Sub MostrarTeoriaConvCorrTen ()

'Mostramos la pestaña de la teoría del convertidor corriente-tensión'
TeoriaConvCorrTen.Show

End Sub

Sub MostrarInformacion ()

'Mostramos la pestaña de la información del convertidor corriente-
tensión'
Informacion1.Show

End Sub

Sub CalcularValores1 ()

'Declaremos las variables de los parámetros del convertidor corriente-
tensión'
Dim R_1 As Single
Dim Ii_1 As Single
Dim VsatPos_1 As Single
Dim VsatNeg_1 As Single
Dim Ganancia_1 As Single
Dim Vo_1 As Single

'Mediante esta declaración conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de la resistencia, su valor no puede ser cero o negativo'
'En el caso de la corriente y las tensiones, su valor si puede ser
cero o negativo'
If ConvCorrTen.Valor_R_1.Text <= 0 Or ConvCorrTen.Valor_R_1.Text =
Empty Then

    MsgBox "Debe introducir un valor en la casilla de la resistencia y
que no sea ni nulo ni negativo", vbExclamation + vbOKOnly, "AVISO"

    ConvCorrTen.Valor_Ganancia_1.Caption = ""
    ConvCorrTen.Valor_Vo_1.Caption = ""
    ConvCorrTen.Valor_k_1.Caption = ""
    ConvCorrTen.Valor_Saturacion_1.Caption = ""

ElseIf ConvCorrTen.Valor_Ii_1.Text = Empty Then
```



```
MsgBox "Debe introducir un valor en la casilla de la corriente",  
vbExclamation + vbOKOnly, "AVISO"  
  
ConvCorrTen.Valor_Ganancia_1.Caption = ""  
ConvCorrTen.Valor_Vo_1.Caption = ""  
ConvCorrTen.Valor_k_1.Caption = ""  
ConvCorrTen.Valor_Saturacion_1.Caption = ""  
  
ElseIf ConvCorrTen.Valor_VsatPos_1.Text = Empty Or  
ConvCorrTen.Valor_VsatNeg_1.Text = Empty Then  
  
MsgBox "Debe introducir un valor en las casillas del Voltaje",  
vbExclamation + vbOKOnly, "AVISO"  
  
ConvCorrTen.Valor_Ganancia_1.Caption = ""  
ConvCorrTen.Valor_Vo_1.Caption = ""  
ConvCorrTen.Valor_k_1.Caption = ""  
ConvCorrTen.Valor_Saturacion_1.Caption = ""  
  
Else  
  
'Asignamos las variables a los correspondientes parámetros del  
convertidor corriente-tensión'  
R_1 = ConvCorrTen.Valor_R_1.Text * 1000  
Ii_1 = ConvCorrTen.Valor_Ii_1.Text / 1000  
VsatPos_1 = ConvCorrTen.Valor_VsatPos_1.Text  
VsatNeg_1 = ConvCorrTen.Valor_VsatNeg_1.Text  
  
'Introducimos la condición de que -Vsat NO puede ser mayor que  
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación  
de Vo'  
If VsatPos_1 < VsatNeg_1 Then  
  
MsgBox "La tensión de saturación -Vsat no puede ser mayor que  
+Vsat", vbInformation + vbOKOnly, "AVISO"  
  
ConvCorrTen.Valor_VsatPos_1.Text = ""  
ConvCorrTen.Valor_VsatNeg_1.Text = ""  
ConvCorrTen.Valor_Ganancia_1.Caption = ""  
ConvCorrTen.Valor_Vo_1.Caption = ""  
ConvCorrTen.Valor_k_1.Caption = ""  
ConvCorrTen.Valor_Saturacion_1.Caption = ""  
  
Else  
  
'Definimos las ecuaciones de los parametros a calcular y  
mostramos el resultado'  
Ganancia_1 = -R_1  
Vo_1 = (Ganancia_1 * Ii_1)  
  
ConvCorrTen.Valor_Ganancia_1.Caption = Ganancia_1  
ConvCorrTen.Valor_Vo_1.Caption = Vo_1  
ConvCorrTen.Valor_k_1.Caption = "K < 0"  
  
'Analizamos si la tensión de salida está saturada'  
If Vo_1 >= VsatPos_1 Then  
  
ConvCorrTen.Valor_Vo_1.Caption = VsatPos_1  
ConvCorrTen.Valor_Saturacion_1.Caption = "Salida saturada"
```




```
ElseIf Vo_1 <= VsatNeg_1 Then

    ConvCorrTen.Valor_Vo_1.Caption = VsatNeg_1
    ConvCorrTen.Valor_Saturacion_1.Caption = "Salida saturada"

Else

    ConvCorrTen.Valor_Saturacion_1.Caption = ""
    ConvCorrTen.Valor_Vo_1.Caption = Vo_1

End If

End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub CalcularValores2 ()

'Declaramos las variables de los parámetros del convertidor corriente-
tensión'
Dim R_2 As Single
Dim R1_2 As Single
Dim R2_2 As Single
Dim Ii_2 As Single
Dim VsatPos_2 As Single
Dim VsatNeg_2 As Single
Dim Ganancia_2 As Single
Dim Vo1_2 As Single
Dim Vo_2 As Single

'Mediante esta declaración conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de las resistencias, su valor no puede ser cero o
negativo'
'En el caso de la corriente y las tensiones, su valor si puede ser
cero o negativo'
If ConvCorrTen.Valor_R_2.Text <= 0 Or ConvCorrTen.Valor_R1_2.Text <= 0
Or ConvCorrTen.Valor_R2_2.Text <= 0 Or ConvCorrTen.Valor_R_2.Text =
Empty Or ConvCorrTen.Valor_R1_2.Text = Empty Or
ConvCorrTen.Valor_R2_2.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas de las
resistencias y que no sea ni nulo ni negativo", vbExclamation +
vbOKOnly, "AVISO"

    ConvCorrTen.Valor_Ganancia_2.Caption = ""
    ConvCorrTen.Valor_Vo1_2.Caption = ""
    ConvCorrTen.Valor_Vo_2.Caption = ""
    ConvCorrTen.Valor_k_2.Caption = ""
    ConvCorrTen.Valor_Saturacion_Vo1_2.Caption = ""
    ConvCorrTen.Valor_Saturacion_2.Caption = ""

ElseIf ConvCorrTen.Valor_Ii_2.Text = Empty Then
```



```
MsgBox "Debe introducir un valor en la casilla de la corriente",  
vbExclamation + vbOKOnly, "AVISO"
```

```
ConvCorrTen.Valor_Ganancia_1.Caption = ""  
ConvCorrTen.Valor_Vo_1.Caption = ""  
ConvCorrTen.Valor_k_1.Caption = ""  
ConvCorrTen.Valor_Saturacion_1.Caption = ""
```

```
ElseIf ConvCorrTen.Valor_VsatPos_2.Text = Empty Or  
ConvCorrTen.Valor_VsatNeg_2.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas del Voltaje",  
vbExclamation + vbOKOnly, "AVISO"
```

```
ConvCorrTen.Valor_Ganancia_2.Caption = ""  
ConvCorrTen.Valor_Vol_2.Caption = ""  
ConvCorrTen.Valor_Vo_2.Caption = ""  
ConvCorrTen.Valor_k_2.Caption = ""  
ConvCorrTen.Valor_Saturacion_Vol_2.Caption = ""  
ConvCorrTen.Valor_Saturacion_2.Caption = ""
```

```
Else
```

```
'Asignamos las variables a los correspondientes parámetros del  
convertidor corriente-tensión'
```

```
R_2 = ConvCorrTen.Valor_R_2.Text * 1000  
R1_2 = ConvCorrTen.Valor_R1_2.Text * 1000  
R2_2 = ConvCorrTen.Valor_R2_2.Text * 1000  
Ii_2 = ConvCorrTen.Valor_Ii_2.Text / 1000  
VsatPos_2 = ConvCorrTen.Valor_VsatPos_2.Text  
VsatNeg_2 = ConvCorrTen.Valor_VsatNeg_2.Text
```

```
'Introducimos la condición de que -Vsat NO puede ser mayor que  
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación  
de Vo'
```

```
If VsatPos_2 < VsatNeg_2 Then
```

```
MsgBox "La tensión de saturación -Vsat no puede ser mayor que  
+Vsat", vbInformation + vbOKOnly, "AVISO"
```

```
ConvCorrTen.Valor_VsatPos_2.Text = ""  
ConvCorrTen.Valor_VsatNeg_2.Text = ""  
ConvCorrTen.Valor_Ganancia_2.Caption = ""  
ConvCorrTen.Valor_Vol_2.Caption = ""  
ConvCorrTen.Valor_Vo_2.Caption = ""  
ConvCorrTen.Valor_k_2.Caption = ""  
ConvCorrTen.Valor_Saturacion_Vol_2.Caption = ""  
ConvCorrTen.Valor_Saturacion_2.Caption = ""
```

```
Else
```

```
'Definimos las ecuaciones de los parametros a calcular  
mostramos el resultado'
```

```
Ganancia_2 = (R_2 * (R2_2 / R1_2))  
Vol_2 = -(R_2 * Ii_2)  
Vo_2 = (Ganancia_2 * Ii_2)
```

```
ConvCorrTen.Valor_Ganancia_2.Caption = Ganancia_2  
ConvCorrTen.Valor_Vol_2.Caption = Vol_2  
ConvCorrTen.Valor_Vo_2.Caption = Vo_2
```



```
ConvCorrTen.Valor_k_2.Caption = "K > 0"

'Analizamos si la tensión de salida está saturada'
If Vo1_2 >= VsatPos_2 Then

    ConvCorrTen.Valor_Vo1_2.Caption = VsatPos_2
    ConvCorrTen.Valor_Saturacion_Vo1_2.Caption = "Vo1 está
saturada"

ElseIf Vo1_2 <= VsatNeg_2 Then

    ConvCorrTen.Valor_Vo1_2.Caption = VsatNeg_2
    ConvCorrTen.Valor_Saturacion_Vo1_2.Caption = "Vo1 está
saturada"

Else

    ConvCorrTen.Valor_Saturacion_Vo1_2.Caption = ""
    ConvCorrTen.Valor_Vo1_2.Caption = Vo1_2

End If

If Vo_2 >= VsatPos_2 Then

    ConvCorrTen.Valor_Vo_2.Caption = VsatPos_2
    ConvCorrTen.Valor_Saturacion_2.Caption = "Salida saturada"

ElseIf Vo_2 <= VsatNeg_2 Then

    ConvCorrTen.Valor_Vo_2.Caption = VsatNeg_2
    ConvCorrTen.Valor_Saturacion_2.Caption = "Salida saturada"

Else

    ConvCorrTen.Valor_Saturacion_2.Caption = ""
    ConvCorrTen.Valor_Vo_2.Caption = Vo_2

End If

End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub RestaurarValores1 ()

'Vaciamos los valores de cada uno de los parámetros del ejemplo 1'
ConvCorrTen.Valor_R_1.Text = ""
ConvCorrTen.Valor_Ii_1.Text = ""
ConvCorrTen.Valor_VsatPos_1.Text = ""
ConvCorrTen.Valor_VsatNeg_1.Text = ""
ConvCorrTen.Valor_k_1.Caption = ""
ConvCorrTen.Valor_Ganancia_1.Caption = ""
ConvCorrTen.Valor_Vo_1.Caption = ""
ConvCorrTen.Valor_Saturacion_1.Caption = ""

End Sub
```



```
Sub RestaurarValores2 ()

'Vaciamos los valores de cada uno de los parámetros del ejemplo 2'
ConvCorrTen.Valor_R_2.Text = ""
ConvCorrTen.Valor_R1_2.Text = ""
ConvCorrTen.Valor_R2_2.Text = ""
ConvCorrTen.Valor_Ii_2.Text = ""
ConvCorrTen.Valor_VsatPos_2.Text = ""
ConvCorrTen.Valor_VsatPos_2.Text = ""
ConvCorrTen.Valor_Ganancia_2.Caption = ""
ConvCorrTen.Valor_k_2.Caption = ""
ConvCorrTen.Valor_Vo1_2.Caption = ""
ConvCorrTen.Valor_Saturacion_Vo1_2.Caption = ""
ConvCorrTen.Valor_Vo_2.Caption = ""
ConvCorrTen.Valor_Saturacion_2.Caption = ""

End Sub
```

7.2.3.6. Convertidor tensión-corriente (ModConvTenCorr)

```
Sub MostrarConvTenCorr ()

'Mostramos la pestaña de la aplicación del convertidor tensión-
corriente'
ConvTenCorr.Show

End Sub

Sub MostrarTeoriaConvTenCorr ()

'Mostramos la pestaña de la teoría del convertidor tensión-corriente'
TeoriaConvTenCorr.Show

End Sub

Sub MostrarInformacion ()

'Mostramos la pestaña de la información del convertidor tensión-
corriente'
Informacion1.Show

End Sub

Sub CalcularValores ()

'Declaramos las variables de los parámetros del convertidor tensión-
corriente'
Dim R1 As Single
Dim R2 As Single
Dim Vi1 As Single
Dim Vi2 As Single
Dim VsatPos As Single
Dim VsatNeg As Single
```



```
Dim Ganancia As Single
Dim Io As Single

'Mediante esta declaración conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de las resistencias, su valor no puede ser cero o
negativo'
'En el caso de las tensiones, su valor si puede ser cero o negativo'
If ConvTenCorr.Valor_R1.Text <= 0 Or ConvTenCorr.Valor_R2.Text <= 0 Or
ConvTenCorr.Valor_R1.Text = Empty Or ConvTenCorr.Valor_R2.Text = Empty
Then

    MsgBox "Debe introducir un valor en las casillas de las
resistencias y que no sea ni nulo ni negativo", vbExclamation +
vbOKOnly, "AVISO"

    ConvTenCorr.Valor_Ganancia.Caption = ""
    ConvTenCorr.Valor_Io.Caption = ""
    ConvTenCorr.Valor_k.Caption = ""

ElseIf ConvTenCorr.Valor_Vi1.Text = Empty Or
ConvTenCorr.Valor_Vi2.Text = Empty Or ConvTenCorr.Valor_VsatPos.Text =
Empty Or ConvTenCorr.Valor_VsatNeg.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas del Voltaje",
vbExclamation + vbOKOnly, "AVISO"

    ConvTenCorr.Valor_Ganancia.Caption = ""
    ConvTenCorr.Valor_Io.Caption = ""
    ConvTenCorr.Valor_k.Caption = ""

Else

    'Asignamos las variables a los correspondientes parámetros del del
convertidor tensión-corriente'
    R1 = ConvTenCorr.Valor_R1.Text * 1000
    R2 = ConvTenCorr.Valor_R2.Text * 1000
    Vi1 = ConvTenCorr.Valor_Vi1.Text
    Vi2 = ConvTenCorr.Valor_Vi2.Text
    VsatPos = ConvTenCorr.Valor_VsatPos.Text
    VsatNeg = ConvTenCorr.Valor_VsatNeg.Text

    'Introducimos la condición de que -Vsat NO puede ser mayor que
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
de Vo'
    If VsatPos < VsatNeg Then

        MsgBox "La tensión de saturación -Vsat no puede ser mayor que
+Vsat", vbInformation + vbOKOnly, "AVISO"

        ConvTenCorr.Valor_VsatPos.Text = ""
        ConvTenCorr.Valor_VsatNeg.Text = ""
        ConvTenCorr.Valor_Ganancia.Caption = ""
        ConvTenCorr.Valor_Io.Caption = ""
        ConvTenCorr.Valor_k.Caption = ""

    Else
```



```
'Definimos las ecuaciones de los parametros a calcular y
mostramos el resultado'
If (Vi2 - Vi1) < 0 Then

    Ganancia = -(1 / R1)

ElseIf (Vi2 - Vi1) > 0 Then

    Ganancia = (1 / R1)

End If

Io = (Ganancia * Abs(Vi2 - Vi1))

ConvTenCorr.Valor_Ganancia.Caption = Ganancia
ConvTenCorr.Valor_Io.Caption = Io

'Analizamos si la constante proporcional es positiva o
negativa'
If Io > 0 Then

    ConvTenCorr.Valor_k.Caption = "K > 0"

ElseIf Io < 0 Then

    ConvTenCorr.Valor_k.Caption = "K < 0"

Else

    MsgBox "La constante proporcional (K) no puede ser nula.
Introduzca unos nuevos valores de Vi1 y Vi2", vbExclamation +
vbOKOnly, "AVISO"

    ConvTenCorr.Valor_Vi1.Text = ""
    ConvTenCorr.Valor_Vi2.Text = ""
    ConvTenCorr.Valor_Ganancia.Caption = ""
    ConvTenCorr.Valor_Io.Caption = ""
    ConvTenCorr.Valor_k.Caption = ""

End If

End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub RestaurarValores ()

'Vaciamos los valores de cada uno de los parámetros'
ConvTenCorr.Valor_R1.Text = ""
ConvTenCorr.Valor_R2.Text = ""
ConvTenCorr.Valor_Vi1.Text = ""
ConvTenCorr.Valor_Vi2.Text = ""
ConvTenCorr.Valor_VsatPos.Text = ""
ConvTenCorr.Valor_VsatNeg.Text = ""
ConvTenCorr.Valor_Ganancia.Caption = ""
ConvTenCorr.Valor_Io.Caption = ""
```



```
ConvTenCorr.Valor_k.Caption = ""
```

```
End Sub
```

7.2.3.7. Detector de pico (ModDetectPico)

```
Sub MostrarDetectPico()
```

```
'Mostramos la pestaña de la aplicación del circuito detector de pico'  
DetectPico.Show
```

```
End Sub
```

```
Sub MostrarTeoriaDetectorPico()
```

```
'Mostramos la pestaña de la aplicación del circuito detector de pico'  
TeoriaDetectorPico.Show
```

```
End Sub
```

```
Sub MostrarInformacion()
```

```
'Mostramos la pestaña de la información del circuito detector de pico'  
Informacion2.Show
```

```
End Sub
```

```
Sub CalcularValores()
```

```
'Declaramos las variables de los parámetros del circuito detector de  
pico'
```

```
Dim Vi As Single
```

```
Dim PI As Single
```

```
Dim a As Single
```

```
Dim b As Single
```

```
Dim VsatPos As Single
```

```
Dim VsatNeg As Single
```

```
Dim Vp As Single
```

```
Dim Vp_Max As Single
```

```
Dim i As Integer
```

```
Dim j As Integer
```

```
Dim k As Integer
```

```
Dim k1 As Integer
```

```
Dim k2 As Integer
```

```
Dim k3 As Integer
```

```
Dim maximo As Integer
```

```
PI = 3.14159265358979
```

```
maximo = 0
```

```
'Mediante esta declaración conseguimos que nuestra macro no pare de  
ejecutarse aunque se encuentre un error'
```

```
On Error Resume Next
```

```
'Nos aseguramos de que se introduzca un valor en todas las casillas'
```



```
'El valor de las tensiones puede ser negativo'  
If DetectPico.a.Text = Empty Or DetectPico.b.Text = Empty Or  
DetectPico.a.Text = 0 Or DetectPico.b.Text = 0 Or  
DetectPico.Valor_VsatPos.Text = Empty Or DetectPico.Valor_VsatNeg.Text  
= Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas del Voltaje",  
vbExclamation + vbOKOnly, "AVISO"
```

```
DetectPico.Valor_Saturacion_Vp.Caption = ""  
DetectPico.Valor_Vp.Caption = ""  
DetectPico.Valor_Saturacion_Vp_Max.Caption = ""  
DetectPico.Valor_Vp_Max.Caption = ""
```

Else

```
'Asignamos las variables a los correspondientes parámetros del  
circuito para obtener el valor medio y el valor eficaz'
```

```
a = DetectPico.a.Text  
b = DetectPico.b.Text  
VsatPos = DetectPico.Valor_VsatPos.Text  
VsatNeg = DetectPico.Valor_VsatNeg.Text  
i = DetectPico.i.Caption  
DetectPico.i.Visible = False  
k1 = DetectPico.k1.Caption  
DetectPico.k1.Visible = False  
k2 = DetectPico.k2.Caption  
DetectPico.k2.Visible = False  
maximo = DetectPico.maximo.Caption  
DetectPico.maximo.Visible = False
```

```
'Introducimos la condición de que -Vsat NO puede ser mayor que  
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación  
de Vo'
```

```
If VsatPos < VsatNeg Then
```

```
MsgBox "La tensión de saturación -Vsat no puede ser mayor que  
+Vsat", vbInformation + vbOKOnly, "AVISO"
```

```
DetectPico.Valor_Vp.Caption = ""  
DetectPico.Valor_Vp_Max.Caption = ""
```

Else

```
'Definimos las ecuaciones de los parametros a calcular y  
mostramos el resultado'
```

```
Vi = Abs(a)  
Vp = Vi  
DetectPico.Valor_Vp.Caption = Vp  
Worksheets("DetectPico").Cells(i, 18).Value = Vp  
k3 = k1 + (360 / b)
```

```
'Analizamos el resultado la primera vez que damos a calcular'
```

```
If i = 5 Then  
ActiveWorkbook.Worksheets("DetectPico").Range("b2") = a  
ActiveWorkbook.Worksheets("DetectPico").Range("c2") = b
```

```
For k = k1 To k3
```

```
Worksheets("DetectPico").Cells(k1, 1).Value = k2
```




```
Worksheets("DetectPico").Cells(k1, 3).Value = a *  
Sin(b * Worksheets("DetectPico").Cells(k1, 2).Value)  
  
If Worksheets("DetectPico").Cells(k1, 3).Value >=  
VsatPos Then  
  
Worksheets("DetectPico").Cells(k1, 3).Value =  
VsatPos  
  
ElseIf Worksheets("DetectPico").Cells(k1, 3).Value <=  
VsatNeg Then  
  
Worksheets("DetectPico").Cells(k1, 3).Value =  
VsatNeg  
  
End If  
  
If k1 >= 5 + (90 / b) Then  
  
Worksheets("DetectPico").Cells(k1, 4).Value = a  
  
If Worksheets("DetectPico").Cells(k1, 4).Value >=  
VsatPos Then  
  
Worksheets("DetectPico").Cells(k1, 4).Value =  
VsatPos  
  
ElseIf Worksheets("DetectPico").Cells(k1, 4).Value  
<= VsatNeg Then  
  
Worksheets("DetectPico").Cells(k1, 4).Value =  
VsatNeg  
  
End If  
  
Else  
  
Worksheets("DetectPico").Cells(k1, 4).Value =  
Worksheets("DetectPico").Cells(k1, 3).Value  
  
End If  
  
k1 = k1 + 1  
k2 = k2 + 1  
  
Next k  
  
Vp = a  
DetectPico.Valor_Vp.Caption = Vp  
Worksheets("DetectPico").Cells(i, 18).Value = Vp  
Vp_Max = Vp  
DetectPico.Valor_Vp_Max.Caption = Vp_Max  
Worksheets("DetectPico").Cells(i, 19).Value = Vp_Max  
  
'A partir de dos o mas veces que queramos calcular el valor de  
pico, se ejecuta el siguiente c□o'  
Else  
  
ActiveWorkbook.Worksheets("DetectPico").Range("b2") = a  
ActiveWorkbook.Worksheets("DetectPico").Range("c2") = b
```



```
        For k = k1 To k3

            Worksheets("DetectPico").Cells(k1, 1).Value = k2
            Worksheets("DetectPico").Cells(k1, 3).Value = a *
Sin(b * Worksheets("DetectPico").Cells(k1, 2).Value)

            If Worksheets("DetectPico").Cells(k1, 3).Value >=
VsatPos Then

                Worksheets("DetectPico").Cells(k1, 3).Value =
VsatPos

            ElseIf Worksheets("DetectPico").Cells(k1, 3).Value <=
VsatNeg Then

                Worksheets("DetectPico").Cells(k1, 3).Value =
VsatNeg

            End If

            If Worksheets("DetectPico").Cells(k1, 3).Value >
Worksheets("DetectPico").Cells(k1 - 1, 4).Value Then

                Worksheets("DetectPico").Cells(k1, 4).Value =
Worksheets("DetectPico").Cells(k1, 3).Value

            ElseIf Worksheets("DetectPico").Cells(k1, 3).Value
< Worksheets("DetectPico").Cells(k1 - 1, 4).Value Then

                Worksheets("DetectPico").Cells(k1, 4).Value =
Worksheets("DetectPico").Cells(k1 - 1, 4).Value

            End If

            k1 = k1 + 1
            k2 = k2 + 1

        Next k

        'Almacenamos todos los valores de pico que hemos obtenido
en nuestra hoja de cálculo, y analizamos cual es el mayor de todos
ellos'
        For j = 5 To i

            If maximo < Worksheets("DetectPico").Cells(j,
18).Value Then

                maximo = Worksheets("DetectPico").Cells(j,
18).Value

            End If

        Next j

        If Vi < maximo Then

            Vp_Max = maximo
            DetectPico.Valor_Vp_Max.Caption = Vp_Max
            Worksheets("DetectPico").Cells(i, 19).Value = Vp_Max
```



```
ElseIf Vi >= maximo Then

    Vp_Max = Vp
    DetectPico.Valor_Vp_Max.Caption = Vp_Max
    Worksheets("DetectPico").Cells(i, 19).Value = Vp_Max

End If

End If

'Analizamos si la tensión de salida está saturada'
If Vp >= VsatPos Then

    DetectPico.Valor_Vp.Caption = VsatPos
    Worksheets("DetectPico").Cells(i, 18).Value = VsatPos
    DetectPico.Valor_Saturacion_Vp.Caption = "Salida saturada"

ElseIf Vp <= VsatNeg Then

    DetectPico.Valor_Vp.Caption = VsatNeg
    Worksheets("DetectPico").Cells(i, 18).Value = VsatNeg
    DetectPico.Valor_Saturacion_Vp.Caption = "Salida saturada"

Else

    DetectPico.Valor_Saturacion_Vp.Caption = ""
    DetectPico.Valor_Vp.Caption = Vp
    Worksheets("DetectPico").Cells(i, 18).Value = Vp

End If

If Vp_Max >= VsatPos Then

    DetectPico.Valor_Vp_Max.Caption = VsatPos
    Worksheets("DetectPico").Cells(i, 19).Value = VsatPos
    DetectPico.Valor_Saturacion_Vp_Max.Caption = "Salida
saturada"

ElseIf Vp_Max <= VsatNeg Then

    DetectPico.Valor_Vp_Max.Caption = VsatNeg
    Worksheets("DetectPico").Cells(i, 19).Value = VsatNeg
    DetectPico.Valor_Saturacion_Vp_Max.Caption = "Salida
saturada"

Else

    DetectPico.Valor_Saturacion_Vp_Max.Caption = ""
    DetectPico.Valor_Vp_Max.Caption = Vp_Max

End If

'Mostramos la representación Vo-t'

'Definimos los puntos necesarios para obtener nuestra gráfica'
Dim j1 As Integer
Dim j2 As Integer
j2 = 0

For j1 = 5 To i
```



```
Worksheets("DetectPico").Cells(j1, 17).Value = j2
j2 = j2 + 1

Next j1

End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub Interruptor()

'Si no hemos obtenido un valor de pico a la salida, no podemos pulsar
el interruptor'
If DetectPico.Valor_Vp.Caption = Empty Then

    MsgBox "No hay valores almacenados", vbExclamation + vbOKOnly,
"AVISO"

'Si ya hemos obtenido valores de pico, podemos pulsar el interruptor y
así vaciar los valores almacenados'
Else

    DetectPico.Valor_Saturacion_Vp.Caption = ""
    DetectPico.Valor_Vp.Caption = 0
    DetectPico.Valor_Saturacion_Vp_Max.Caption = ""
    DetectPico.Valor_Vp_Max.Caption = 0
    DetectPico.maximo.Caption = 0
    DetectPico.i.Caption = 5
    DetectPico.k1.Caption = 5
    DetectPico.k2.Caption = 0
    DetectPico.maximo.Visible = False
    DetectPico.i.Visible = False
    DetectPico.k1.Visible = False
    DetectPico.k2.Visible = False
    ActiveWorkbook.Worksheets("DetectPico").Range("b2") = Empty
    ActiveWorkbook.Worksheets("DetectPico").Range("c2") = Empty
    ActiveWorkbook.Worksheets("DetectPico").Range("a5:a1200") = Empty
    ActiveWorkbook.Worksheets("DetectPico").Range("c5:c1200") = Empty
    ActiveWorkbook.Worksheets("DetectPico").Range("d5:d1200") = Empty
    ActiveWorkbook.Worksheets("DetectPico").Range("q5:q1200") = Empty
    ActiveWorkbook.Worksheets("DetectPico").Range("r5:r1200") = Empty
    ActiveWorkbook.Worksheets("DetectPico").Range("s5:s1200") = Empty

End If

End Sub

Sub RestaurarValores()

'Vaciamos los valores de cada uno de los parámetros'
DetectPico.a.Text = ""
DetectPico.b.Text = ""
DetectPico.Valor_VsatPos.Text = ""
DetectPico.Valor_VsatNeg.Text = ""
DetectPico.Valor_Saturacion_Vp.Caption = ""
DetectPico.Valor_Vp.Caption = ""
```



```
DetectPico.Valor_Saturacion_Vp_Max.Caption = ""  
DetectPico.Valor_Vp_Max.Caption = ""
```

```
End Sub
```

7.2.3.8. Diferenciador (ModDiferenciador)

```
Sub MostrarDiferenciador()
```

```
'Mostramos la pestaña de la aplicación del circuito diferenciador'  
Diferenciador.Show
```

```
End Sub
```

```
Sub MostrarTeoriaDiferenciador()
```

```
'Mostramos la pestaña de la teoría del circuito integrador  
diferenciador'  
TeoriaDiferenciador.Show
```

```
End Sub
```

```
Sub MostrarInformacion()
```

```
'Mostramos la pestaña de la información del circuito integrador  
diferenciador'  
InformacionDiferencial.Show
```

```
End Sub
```

```
Sub CalcularValores()
```

```
'Declaramos las variables de los parámetros de un circuito  
diferenciador'
```

```
Dim R As Single  
Dim C As Single  
Dim Vi As Single  
Dim n As Single  
Dim Derivada As Single  
Dim VsatPos As Single  
Dim VsatNeg As Single  
Dim t_final As Single  
Dim t_sat As Single  
Dim Vo As Single
```

```
'Mediante esta declaración conseguimos que nuestra macro no pare de  
ejecutarse aunque se encuentre un error'  
On Error Resume Next
```

```
'Nos aseguramos de que se introduzca un valor en todas las casillas'  
'En el caso de la resistencia, el condensador y el valor de t final,  
su valor no puede ser cero o negativo'  
'La potencia de la tensión de entrada no puede ser nula o negativa'  
'En el caso de las tensiones, su valor si puede ser cero o negativo'
```



```
If Diferenciador.Valor_R.Text <= 0 Or Diferenciador.Valor_C.Text <= 0
Or Diferenciador.Valor_R.Text = Empty Or Diferenciador.Valor_C.Text =
Empty Then

    MsgBox "Debe introducir un valor en las casillas de la resistencia
y el condensador, y que no sea ni nulo ni negativo", vbExclamation +
vbOKOnly, "AVISO"

    Diferenciador.Valor_Vo.Caption = ""
    Diferenciador.Valor_Saturacion.Caption = ""

ElseIf Diferenciador.Valor_Vi.Text = Empty Or
Diferenciador.Valor_VsatPos.Text = Empty Or
Diferenciador.Valor_VsatNeg.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas del Voltaje",
vbExclamation + vbOKOnly, "AVISO"

    Diferenciador.Valor_Vo.Caption = ""
    Diferenciador.Valor_Saturacion.Caption = ""

ElseIf Diferenciador.n.Text <= 0 Or Diferenciador.n.Text = Empty Then

    MsgBox "Debe introducir un valor en la casilla de la potencia del
tiempo y que no sea nulo", vbExclamation + vbOKOnly, "AVISO"

    Diferenciador.Valor_Vo.Caption = ""
    Diferenciador.Valor_Saturacion.Caption = ""

ElseIf Diferenciador.t_final.Text <= 0 Or Diferenciador.t_final.Text =
Empty Then

    MsgBox "Debe introducir un valor en la casilla del tiempo referido
a la condición final y que no sea nulo o negativo", vbExclamation +
vbOKOnly, "AVISO"

    Diferenciador.Valor_Vo.Caption = ""
    Diferenciador.Valor_Saturacion.Caption = ""

Else

    'Asignamos las variables a los correspondientes parámetros del
circuito diferenciador'
    R = Diferenciador.Valor_R.Text * 1000
    C = Diferenciador.Valor_C.Text / 1000000
    Vi = Diferenciador.Valor_Vi.Text
    n = Diferenciador.n.Text
    VsatPos = Diferenciador.Valor_VsatPos.Text
    VsatNeg = Diferenciador.Valor_VsatNeg.Text
    t_final = Diferenciador.t_final.Text

    'Introducimos la condición de que -Vsat NO puede ser mayor que
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
de Vo'
    If VsatPos < VsatNeg Then

        MsgBox "La tensión de saturación -Vsat no puede ser mayor que
+Vsat", vbInformation + vbOKOnly, "AVISO"

        Diferenciador.Valor_VsatPos.Text = ""
        Diferenciador.Valor_VsatNeg.Text = ""
```



```
Diferenciador.Valor_Vo.Caption = ""
Diferenciador.Valor_Saturacion.Caption = ""

Else

    'Definimos las ecuaciones de los parametros a calcular y
    mostramos el resultado'
    Derivada = n * (t_final) ^ (n - 1)
    Vo = -R * C * Vi * Derivada

    Diferenciador.Valor_Vo.Caption = Vo

    'Analizamos si la tensión de salida está saturada'
    If Vo >= VsatPos Then

        Diferenciador.Valor_Vo.Caption = VsatPos
        Diferenciador.Valor_Saturacion.Caption = "Salida saturada"

    ElseIf Vo <= VsatNeg Then

        Diferenciador.Valor_Vo.Caption = VsatNeg
        Diferenciador.Valor_Saturacion.Caption = "Salida saturada"

    Else

        Diferenciador.Valor_Saturacion.Caption = ""
        Diferenciador.Valor_Vo.Caption = Vo

    End If

    'Mostramos la representacion Vi-t Vo-t'

    'Calculamos el valor de t en el que Vo satura, ya que serán
    necesarios para obtener la gráfica'
    If Vi > 0 Then

        t_sat = ((Abs(VsatNeg) / (R * C * Vi * n)) ^ (1 / (n -
1)))

    ElseIf Vi < 0 Then

        t_sat = ((Abs(VsatPos) / (R * C * Abs(Vi) * n)) ^ (1 / (n
- 1)))

    End If

    'Definimos los puntos necesarios para obtener nuestra gráfica'

    'En función de la potencia de la entrada, pasamos unos datos u
    otros'
    If n > 1 Then

        Vi
        ActiveWorkbook.Worksheets("Diferenciador").Range("c9") =
        ActiveWorkbook.Worksheets("Diferenciador").Range("c6") = R
        ActiveWorkbook.Worksheets("Diferenciador").Range("c7") = C
        ActiveWorkbook.Worksheets("Diferenciador").Range("c15") =
        VsatPos
        ActiveWorkbook.Worksheets("Diferenciador").Range("c16") =
        VsatNeg
        ActiveWorkbook.Worksheets("Diferenciador").Range("c8") = n
```



```
ActiveWorkbook.Worksheets("Diferenciador").Range("c18") =  
t_sat  
ActiveWorkbook.Worksheets("Diferenciador").Range("d18") =  
t_sat + 3  
ActiveWorkbook.Worksheets("Diferenciador").Range("c19") =  
Empty  
ActiveWorkbook.Worksheets("Diferenciador").Range("d19") =  
Empty  
ActiveWorkbook.Worksheets("Diferenciador").Range("c21") =  
Empty  
ActiveWorkbook.Worksheets("Diferenciador").Range("d21") =  
Empty  
  
ElseIf n = 1 Then  
  
ActiveWorkbook.Worksheets("Diferenciador").Range("c19") =  
0  
ActiveWorkbook.Worksheets("Diferenciador").Range("d19") =  
3  
ActiveWorkbook.Worksheets("Diferenciador").Range("c9") =  
Vi  
ActiveWorkbook.Worksheets("Diferenciador").Range("c6") = R  
ActiveWorkbook.Worksheets("Diferenciador").Range("c7") = C  
ActiveWorkbook.Worksheets("Diferenciador").Range("c15") =  
VsatPos  
ActiveWorkbook.Worksheets("Diferenciador").Range("c16") =  
VsatNeg  
ActiveWorkbook.Worksheets("Diferenciador").Range("c8") = n  
ActiveWorkbook.Worksheets("Diferenciador").Range("c18") =  
Empty  
ActiveWorkbook.Worksheets("Diferenciador").Range("d18") =  
Empty  
  
If -R * C * Vi > VsatPos Then  
  
ActiveWorkbook.Worksheets("Diferenciador").Range("c21") = VsatPos  
ActiveWorkbook.Worksheets("Diferenciador").Range("d21") = VsatPos  
  
ElseIf -R * C * Vi < VsatNeg Then  
  
ActiveWorkbook.Worksheets("Diferenciador").Range("c21") = VsatNeg  
ActiveWorkbook.Worksheets("Diferenciador").Range("d21") = VsatNeg  
  
Else  
  
ActiveWorkbook.Worksheets("Diferenciador").Range("c21") = -R * C * Vi  
ActiveWorkbook.Worksheets("Diferenciador").Range("d21") = -R * C * Vi  
  
End If  
  
End If  
  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c36") = Vi  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c37") = n  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c47") = t_sat
```




```
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("d47") = t_sat
+ 3

End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub RestaurarValores()

'Vaciamos los valores de cada uno de los parámetros'
Diferenciador.Valor_R.Text = ""
Diferenciador.Valor_C.Text = ""
Diferenciador.Valor_Vi.Text = ""
Diferenciador.n.Text = ""
Diferenciador.Valor_VsatPos.Text = ""
Diferenciador.Valor_VsatNeg.Text = ""
Diferenciador.t_final.Text = ""
Diferenciador.Valor_Vo.Caption = ""
Diferenciador.Valor_Saturacion.Caption = ""

End Sub
```

7.2.3.9. Fuentes de corriente (ModFuenteCorriente)

```
Sub MostrarFuenteCorriente()

'Mostramos la pestaña de la aplicación de dos ejemplos de fuente de corriente'
FuenteCorriente.Show

End Sub

Sub MostrarTeoriaFuenteCorriente()

'Mostramos la pestaña de la teoría de dos ejemplos de fuente de corriente'
TeoriaFuenteCorriente.Show

End Sub

Sub MostrarInformacion()

'Mostramos la pestaña de la información de dos ejemplos de fuente de corriente'
Informacion1.Show

End Sub

Sub CalcularValores1()
```



```
'Declaramos las variables de los parámetros del ejemplo 1 de fuente de corriente'  
Dim R1_1 As Single  
Dim R2_1 As Single  
Dim RL_1 As Single  
Dim Vi_1 As Single  
Dim VsatPos_1 As Single  
Dim VsatNeg_1 As Single  
Dim IL_1 As Single  
  
'Mediante esta declaración conseguimos que nuestra macro no pare de ejecutarse aunque se encuentre un error'  
On Error Resume Next  
  
'Nos aseguramos de que se introduzca un valor en todas las casillas'  
'En el caso de las resistencias, su valor no puede ser cero o negativo'  
'En el caso de las tensiones, su valor si puede ser cero o negativo'  
If FuenteCorriente.Valor_R1_1.Text <= 0 Or  
FuenteCorriente.Valor_R2_1.Text <= 0 Or  
FuenteCorriente.Valor_RL_1.Text <= 0 Or  
FuenteCorriente.Valor_R1_1.Text = Empty Or  
FuenteCorriente.Valor_R2_1.Text = Empty Or  
FuenteCorriente.Valor_RL_1.Text = Empty Then  
  
    MsgBox "Debe introducir un valor en las casillas de las resistencias y que no sea ni nulo ni negativo", vbExclamation + vbOKOnly, "AVISO"  
  
    FuenteCorriente.Valor_IL_1.Caption = ""  
  
ElseIf FuenteCorriente.Valor_Vi_1.Text = Empty Or  
FuenteCorriente.Valor_VsatPos_1.Text = Empty Or  
FuenteCorriente.Valor_VsatNeg_1.Text = Empty Then  
  
    MsgBox "Debe introducir un valor en las casillas del Voltaje", vbExclamation + vbOKOnly, "AVISO"  
  
    FuenteCorriente.Valor_IL_1.Caption = ""  
  
Else  
  
    'Asignamos las variables a los correspondientes parámetros del ejemplo 1 de fuente de corriente'  
    R1_1 = FuenteCorriente.Valor_R1_1.Text * 1000  
    R2_1 = FuenteCorriente.Valor_R2_1.Text * 1000  
    RL_1 = FuenteCorriente.Valor_RL_1.Text * 1000  
    Vi_1 = FuenteCorriente.Valor_Vi_1.Text  
    VsatPos_1 = FuenteCorriente.Valor_VsatPos_1.Text  
    VsatNeg_1 = FuenteCorriente.Valor_VsatNeg_1.Text  
  
    'Introducimos la condición de que -Vsat NO puede ser mayor que +Vsat, ya que si se pudiese dar el caso, podría alterar la saturación de Vo'  
    If VsatPos_1 < VsatNeg_1 Then  
  
        MsgBox "La tensión de saturación -Vsat no puede ser mayor que +Vsat", vbInformation + vbOKOnly, "AVISO"  
  
        FuenteCorriente.Valor_VsatPos_1.Text = ""
```



```
FuenteCorriente.Valor_VsatNeg_1.Text = ""
FuenteCorriente.Valor_IL_1.Caption = ""

Else

    'Definimos las ecuaciones de los parametros a calcular y
    mostramos el resultado'
    IL_1 = (Vi_1 / R1_1)

    FuenteCorriente.Valor_IL_1.Caption = IL_1

End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub CalcularValores2()

'Declaremos las variables de los parámetros del ejemplo 2 de fuente de
corriente'
Dim R1_2 As Single
Dim R2_2 As Single
Dim R3_2 As Single
Dim RL_2 As Single
Dim Vi_2 As Single
Dim VsatPos_2 As Single
Dim VsatNeg_2 As Single
Dim Ganancia_2 As Single
Dim Vo_2 As Single
Dim IL_2 As Single

'Mediante esta declaración conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de las resistencias, su valor no puede ser cero o
negativo'
'En el caso de las tensiones, su valor si puede ser cero o negativo'
If FuenteCorriente.Valor_R1_2.Text <= 0 Or
FuenteCorriente.Valor_R2_2.Text <= 0 Or
FuenteCorriente.Valor_R3_2.Text <= 0 Or
FuenteCorriente.Valor_RL_2.Text <= 0 Or
FuenteCorriente.Valor_R1_2.Text = Empty Or
FuenteCorriente.Valor_R2_2.Text = Empty Or
FuenteCorriente.Valor_R3_2.Text = Empty Or
FuenteCorriente.Valor_RL_2.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas de las
    resistencias y que no sea ni nulo ni negativo", vbExclamation +
    vbOKOnly, "AVISO"

    FuenteCorriente.Valor_Ganancia_2.Caption = ""
    FuenteCorriente.Valor_Vo_2.Caption = ""
    FuenteCorriente.Valor_IL_2.Caption = ""
    FuenteCorriente.Valor_Saturacion_2.Caption = ""
```



```
ElseIf FuenteCorriente.Valor_Vi_2.Text = Empty Or
FuenteCorriente.Valor_VsatPos_2.Text = Empty Or
FuenteCorriente.Valor_VsatNeg_2.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas del Voltaje",
vbExclamation + vbOKOnly, "AVISO"

    FuenteCorriente.Valor_Ganancia_2.Caption = ""
    FuenteCorriente.Valor_Vo_2.Caption = ""
    FuenteCorriente.Valor_IL_2.Caption = ""
    FuenteCorriente.Valor_Saturacion_2.Caption = ""

Else

    'Asignamos las variables a los correspondientes parámetros del
ejemplo 2 de fuente de corriente'
    R1_2 = FuenteCorriente.Valor_R1_2.Text * 1000
    R2_2 = FuenteCorriente.Valor_R2_2.Text * 1000
    R3_2 = FuenteCorriente.Valor_R3_2.Text * 1000
    RL_2 = FuenteCorriente.Valor_RL_2.Text * 1000
    Vi_2 = FuenteCorriente.Valor_Vi_2.Text
    VsatPos_2 = FuenteCorriente.Valor_VsatPos_2.Text
    VsatNeg_2 = FuenteCorriente.Valor_VsatNeg_2.Text

    'Introducimos la condición de que -Vsat NO puede ser mayor que
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
de Vo'
    If VsatPos_2 < VsatNeg_2 Then

        MsgBox "La tensión de saturación -Vsat no puede ser mayor que
+Vsat", vbInformation + vbOKOnly, "AVISO"

        FuenteCorriente.Valor_VsatPos_2.Text = ""
        FuenteCorriente.Valor_VsatNeg_2.Text = ""
        FuenteCorriente.Valor_Ganancia_2.Caption = ""
        FuenteCorriente.Valor_Vo_2.Caption = ""
        FuenteCorriente.Valor_IL_2.Caption = ""
        FuenteCorriente.Valor_Saturacion_2.Caption = ""

    Else

        'Definimos las ecuaciones de los parametros a calcular y
mostramos el resultado'
        Ganancia_2 = (-(R2_2 / R1_2))
        Vo_2 = (Ganancia_2 * Vi_2)
        IL_2 = (-(Vi_2 / R1_2) * ((R3_2 + R2_2) / R3_2))

        FuenteCorriente.Valor_Ganancia_2.Caption = Ganancia_2
        FuenteCorriente.Valor_Vo_2.Caption = Vo_2
        FuenteCorriente.Valor_IL_2.Caption = IL_2

        'Analizamos si la tensión de salida está saturada'
        If Vo_2 >= VsatPos_2 Then

            FuenteCorriente.Valor_Vo_2.Caption = VsatPos_2
            FuenteCorriente.Valor_Saturacion_2.Caption = "Salida
saturada"

        ElseIf Vo_2 <= VsatNeg_2 Then

            FuenteCorriente.Valor_Vo_2.Caption = VsatNeg_2

        End If

    End If

End If
```



```
FuenteCorriente.Valor_Saturacion_2.Caption = "Salida
saturada"

Else

FuenteCorriente.Valor_Saturacion_2.Caption = ""
FuenteCorriente.Valor_Vo_2.Caption = Vo_2

End If

End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub RestaurarValores1 ()

'Vaciamos los valores de cada uno de los parámetros del ejemplo 1'
FuenteCorriente.Valor_R1_1.Text = ""
FuenteCorriente.Valor_R2_1.Text = ""
FuenteCorriente.Valor_RL_1.Text = ""
FuenteCorriente.Valor_Vi_1.Text = ""
FuenteCorriente.Valor_VsatPos_1.Text = ""
FuenteCorriente.Valor_VsatNeg_1.Text = ""
FuenteCorriente.Valor_IL_1.Caption = ""

End Sub

Sub RestaurarValores2 ()

'Vaciamos los valores de cada uno de los parámetros del ejemplo 2'
FuenteCorriente.Valor_R1_2.Text = ""
FuenteCorriente.Valor_R2_2.Text = ""
FuenteCorriente.Valor_R3_2.Text = ""
FuenteCorriente.Valor_RL_2.Text = ""
FuenteCorriente.Valor_Vi_2.Text = ""
FuenteCorriente.Valor_VsatPos_2.Text = ""
FuenteCorriente.Valor_VsatNeg_2.Text = ""
FuenteCorriente.Valor_Ganancia_2.Caption = ""
FuenteCorriente.Valor_Vo_2.Caption = ""
FuenteCorriente.Valor_Saturacion_2.Caption = ""
FuenteCorriente.Valor_IL_2.Caption = ""

End Sub
```

7.2.3.10. Fuente de tensión de referencia (ModFuenteTension)

```
Sub MostrarFuenteTension ()
```



```
'Mostramos la pestaña de la aplicación de una fuente de tensión con
diode zener'
FuenteTension.Show

End Sub

Sub MostrarTeoriaFuenteTension ()

'Mostramos la pestaña de la teoría de una fuente de tensión con diode
zener'
TeoriaFuenteTension.Show

End Sub

Sub MostrarInformacion ()

'Mostramos la pestaña de la información de una fuente de tensión con
diode zener'
Informacion1.Show

End Sub

Sub CalcularValores ()

'Declaramos las variables de los parámetros de una fuente de tensión
con diode zener'
Dim R1 As Single
Dim R2 As Single
Dim Vz As Single
Dim VsatPos As Single
Dim VsatNeg As Single
Dim R3 As Single
Dim Ganancia As Single
Dim Vref As Single

'Mediante esta declaración conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de las resistencias, su valor no puede ser cero o
negativo'
'En el caso de las tensiones, su valor si puede ser cero o negativo'
If FuenteTension.Valor_R1.Text <= 0 Or FuenteTension.Valor_R2.Text <=
0 Or FuenteTension.Valor_R1.Text = Empty Or
FuenteTension.Valor_R2.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas de las
resistencias y que no sea ni nulo ni negativo", vbExclamation +
vbOKOnly, "AVISO"

    FuenteTension.Valor_R3.Caption = ""
    FuenteTension.Valor_Ganancia.Caption = ""
    FuenteTension.Valor_Vref.Caption = ""
    FuenteTension.Valor_Saturacion.Caption = ""

ElseIf FuenteTension.Valor_Vz.Text = Empty Or
FuenteTension.Valor_VsatPos.Text = Empty Or
FuenteTension.Valor_VsatNeg.Text = Empty Then
```



```
MsgBox "Debe introducir un valor en las casillas del Voltaje",  
vbExclamation + vbOKOnly, "AVISO"  
  
FuenteTension.Valor_R3.Caption = ""  
FuenteTension.Valor_Ganancia.Caption = ""  
FuenteTension.Valor_Vref.Caption = ""  
FuenteTension.Valor_Saturacion.Caption = ""  
  
Else  
  
    'Asignamos las variables a los correspondientes parámetros de una  
    fuente de tensión con diodo zener'  
    R1 = FuenteTension.Valor_R1.Text * 1000  
    R2 = FuenteTension.Valor_R2.Text * 1000  
    Vz = FuenteTension.Valor_Vz.Text  
    VsatPos = FuenteTension.Valor_VsatPos.Text  
    VsatNeg = FuenteTension.Valor_VsatNeg.Text  
  
    'Introducimos la condición de que -Vsat NO puede ser mayor que  
    +Vsat, ya que si se pudiese dar el caso, podría alterar la saturación de  
    Vo'  
    If VsatPos < VsatNeg Then  
  
        MsgBox "La tensión de saturación -Vsat no puede ser mayor que  
        +Vsat", vbInformation + vbOKOnly, "AVISO"  
  
        FuenteTension.Valor_VsatPos.Text = ""  
        FuenteTension.Valor_VsatNeg.Text = ""  
        FuenteTension.Valor_Ganancia.Caption = ""  
        FuenteTension.Valor_Vref.Caption = ""  
        FuenteTension.Valor_Saturacion.Caption = ""  
  
    Else  
  
        'Definimos las ecuaciones de los parametros a calcular y  
        mostramos el resultado'  
        R3 = ((R2 * R1) / (R2 + R1)) / 1000  
        Ganancia = (-(R2 / R1))  
        Vref = (Ganancia * Vz)  
  
        FuenteTension.Valor_R3.Caption = R3  
        FuenteTension.Valor_Ganancia.Caption = Ganancia  
        FuenteTension.Valor_Vref.Caption = Vref  
  
        'Analizamos si la tensión de salida está saturada'  
        If Vref >= VsatPos Then  
  
            FuenteTension.Valor_Vref.Caption = VsatPos  
            FuenteTension.Valor_Saturacion.Caption = "Salida saturada"  
  
        ElseIf Vref <= VsatNeg Then  
  
            FuenteTension.Valor_Vref.Caption = VsatNeg  
            FuenteTension.Valor_Saturacion.Caption = "Salida saturada"  
  
        Else  
  
            FuenteTension.Valor_Saturacion.Caption = ""  
            FuenteTension.Valor_Vref.Caption = Vref  
  
        End If  
  
    End If
```



```
        End If

    End If

    'Desactivamos el detector de errores'
    On Error GoTo 0

    End Sub

    Sub RestaurarValores ()

        'Vaciamos los valores de cada uno de los parámetros'
        FuenteTension.Valor_R1.Text = ""
        FuenteTension.Valor_R2.Text = ""
        FuenteTension.Valor_Vz.Text = ""
        FuenteTension.Valor_VsatNeg.Text = ""
        FuenteTension.Valor_VsatPos.Text = ""
        FuenteTension.Valor_R3.Caption = ""
        FuenteTension.Valor_Ganancia.Caption = ""
        FuenteTension.Valor_Vref.Caption = ""
        FuenteTension.Valor_Saturacion.Caption = ""

    End Sub
```

7.2.3.11. Comparadores con histéresis (ModHisteresis)

```
Sub MostrarHisteresis1 ()

    'Mostramos la pestaña de la aplicación del circuito comparador con
    histéresis'
    Histeresis1.Show

End Sub

Sub MostrarTeoriaHisteresis ()

    'Mostramos la pestaña de la teoría del circuito comparador con
    histéresis'
    TeoriaHisteresis.Show

End Sub

Sub MostrarInformacionHisteresis ()

    'Mostramos la pestaña de la información del circuito comparador con
    histéresis'
    Informacion.Show

End Sub

Sub CalcularValoresHisteresis ()
```




```
'Declaramos las variables de los parámetros de un circuito comparador
con histéresis'
Dim R1 As Single
Dim R2 As Single
Dim Vi As Single
Dim Vref As Single
Dim a As Single
Dim b As Single
Dim VsatPos As Single
Dim VsatNeg As Single
Dim Vo_1 As Single
Dim Vo_2 As Single

'Mediante esta declaración conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de las resistencias, su valor no puede ser cero o
negativo'
'En el caso de las tensiones, su valor si puede ser cero o negativo'
If Histeresis1.Valor_R1.Text <= 0 Or Histeresis1.Valor_R2.Text <= 0 Or
Histeresis1.Valor_R1.Text = Empty Or Histeresis1.Valor_R2.Text = Empty
Then

    MsgBox "Debe introducir un valor en las casillas de las
resistencias, y que no sea ni nulo ni negativo", vbExclamation +
vbOKOnly, "AVISO"

    Histeresis1.Valor_A.Caption = ""
    Histeresis1.Valor_B.Caption = ""
    Histeresis1.Valor_Vo_1.Caption = ""
    Histeresis1.Valor_Vo_2.Caption = ""

ElseIf Histeresis1.Valor_Vi.Text = Empty Or
Histeresis1.Valor_Vref.Text = Empty Or Histeresis1.Valor_VsatPos.Text
= Empty Or Histeresis1.Valor_VsatNeg.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas del Voltaje",
vbExclamation + vbOKOnly, "AVISO"

    Histeresis1.Valor_A.Caption = ""
    Histeresis1.Valor_B.Caption = ""
    Histeresis1.Valor_Vo_1.Caption = ""
    Histeresis1.Valor_Vo_2.Caption = ""

Else

    'Asignamos las variables a los correspondientes parámetros de un
circuito comparador con histéresis'
    R1 = Histeresis1.Valor_R1.Text
    R2 = Histeresis1.Valor_R2.Text
    Vi = Histeresis1.Valor_Vi.Text
    Vref = Histeresis1.Valor_Vref.Text
    VsatPos = Histeresis1.Valor_VsatPos.Text
    VsatNeg = Histeresis1.Valor_VsatNeg.Text

    'Introducimos la condición de que -Vsat NO puede ser mayor que
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
de Vo'
    If VsatPos < VsatNeg Then
```



```
MsgBox "La tensión de saturación -Vsat no puede ser mayor que  
+Vsat", vbInformation + vbOKOnly, "AVISO"
```

```
Histeresis1.Valor_A.Caption = ""  
Histeresis1.Valor_B.Caption = ""  
Histeresis1.Valor_VsatPos.Text = ""  
Histeresis1.Valor_VsatNeg.Text = ""  
Histeresis1.Valor_Vo_1.Caption = ""  
Histeresis1.Valor_Vo_2.Caption = ""
```

```
Else
```

```
'Definimos las ecuaciones de los parametros a calcular y  
mostramos el resultado'
```

```
a = (Vref * R2 + VsatPos * R1) / (R1 + R2)  
b = (Vref * R2 + VsatNeg * R1) / (R1 + R2)
```

```
If Vi > a Then
```

```
Vo_1 = VsatNeg
```

```
ElseIf Vi <= a Then
```

```
Vo_1 = VsatPos
```

```
End If
```

```
If Vi >= b Then
```

```
Vo_2 = VsatNeg
```

```
ElseIf Vi < b Then
```

```
Vo_2 = VsatPos
```

```
End If
```

```
Histeresis1.Valor_Vo_1.Caption = Vo_1  
Histeresis1.Valor_Vo_2.Caption = Vo_2  
Histeresis1.Valor_A.Caption = a  
Histeresis1.Valor_B.Caption = b
```

```
'Analizamos si la tensión de salida está saturada'
```

```
If Vo_1 >= VsatPos Then
```

```
Histeresis1.Valor_Vo_1.Caption = VsatPos
```

```
ElseIf Vo_1 <= VsatNeg Then
```

```
Histeresis1.Valor_Vo_1.Caption = VsatNeg
```

```
Else
```

```
Histeresis1.Valor_Vo_1.Caption = Vo_1
```

```
End If
```

```
If Vo_2 >= VsatPos Then
```

```
Histeresis1.Valor_Vo_2.Caption = VsatPos
```



```
ElseIf Vo_2 <= VsatNeg Then

    Histeresis1.Valor_Vo_2.Caption = VsatNeg

Else

    Histeresis1.Valor_Vo_2.Caption = Vo_2

End If

'Mostramos la representación Vo-Vi'
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s14") = a
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("v14") = b
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("w17") = a -
b
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s16") =
VsatPos
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t16") =
VsatPos
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s17") =
VsatNeg
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t17") =
VsatNeg
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s23") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("v23") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s19") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t19") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s20") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t20") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s21") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t21") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s30") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("v30") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s28") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t28") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s26") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t26") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s27") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t27") = ""

End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub RestaurarValoresHisteresis()

'Maciamos los valores de cada uno de los parámetros'
Histeresis1.Valor_R1.Text = ""
Histeresis1.Valor_R2.Text = ""
Histeresis1.Valor_Vi.Text = ""
Histeresis1.Valor_Vref.Text = ""
Histeresis1.Valor_VsatPos.Text = ""
Histeresis1.Valor_VsatNeg.Text = ""
Histeresis1.Valor_A.Caption = ""
Histeresis1.Valor_B.Caption = ""
```



```
Histeresis1.Valor_Vo_1.Caption = ""
Histeresis1.Valor_Vo_2.Caption = ""

End Sub

Sub MostrarHisteresis2 ()

'Mostramos la pestaña de la aplicación del circuito comparador con
histéresis y limitación en la salida'
Histeresis2.Show

End Sub

Sub MostrarTeoriaHisteresis1 ()

'Mostramos la pestaña de la teoría del ejemplo 1 circuito comparador
con histéresis y limitación en la salida'
TeoriaHisteresis1.Show

End Sub

Sub MostrarTeoriaHisteresis2 ()

'Mostramos la pestaña de la teoría del ejemplo 2 circuito comparador
con histéresis y limitación en la salida'
TeoriaHisteresis2.Show

End Sub

Sub MostrarInformacionHisteresis1 ()

'Mostramos la pestaña de la información del ejemplo 1 circuito
comparador con histéresis y limitación en la salida'
Informacion4.Show

End Sub

Sub MostrarInformacionHisteresis2 ()

'Mostramos la pestaña de la información del ejemplo 2 circuito
comparador con histéresis y limitación en la salida'
Informacion4.Show

End Sub

Sub CalcularValoresHisteresis1 ()

'Declaramos las variables de los parámetros del ejemplo 1 de un
circuito comparador con histéresis y limitación en la salida'
Dim R1_1 As Single
Dim R2_1 As Single
Dim Vi_1 As Single
Dim Vref_1 As Single
Dim Vu_1 As Single
Dim a_1 As Single
Dim b_1 As Single
Dim VsatPos_1 As Single
Dim VsatNeg_1 As Single
Dim Vlo_1_1 As Single
Dim Vo_1_1 As Single
Dim Vlo_2_1 As Single
```



```
Dim Vo_2_1 As Single
```

```
'Mediante esta declaración conseguimos que nuestra macro no pare de  
ejecutarse aunque se encuentre un error'
```

```
On Error Resume Next
```

```
'Nos aseguramos de que se introduzca un valor en todas las casillas'  
'En el caso de las resistencias, su valor no puede ser cero o  
negativo'
```

```
'En el caso de las tensiones, su valor si puede ser cero o negativo'  
If Histeresis2.Valor_R1_1.Text <= 0 Or Histeresis2.Valor_R2_1.Text <=  
0 Or Histeresis2.Valor_R1_1.Text = Empty Or  
Histeresis2.Valor_R2_1.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas de las  
resistencias, y que no sea ni nulo ni negativo", vbExclamation +  
vbOKOnly, "AVISO"
```

```
Histeresis2.Valor_A_1.Caption = ""  
Histeresis2.Valor_B_1.Caption = ""  
Histeresis2.Valor_Vlo_1_1.Caption = ""  
Histeresis2.Valor_Vo_1_1.Caption = ""  
Histeresis2.Valor_Vlo_2_1.Caption = ""  
Histeresis2.Valor_Vo_2_1.Caption = ""  
Histeresis2.Conduccion_1_1.Caption = ""  
Histeresis2.Conduccion_2_1.Caption = ""
```

```
ElseIf Histeresis2.Valor_Vi_1.Text = Empty Or  
Histeresis2.Valor_Vref_1.Text = Empty Or Histeresis2.Valor_Vu_1.Text =  
Empty Or Histeresis2.Valor_VsatPos_1.Text = Empty Or  
Histeresis2.Valor_VsatNeg_1.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas del Voltaje",  
vbExclamation + vbOKOnly, "AVISO"
```

```
Histeresis2.Valor_A_1.Caption = ""  
Histeresis2.Valor_B_1.Caption = ""  
Histeresis2.Valor_Vlo_1_1.Caption = ""  
Histeresis2.Valor_Vo_1_1.Caption = ""  
Histeresis2.Valor_Vlo_2_1.Caption = ""  
Histeresis2.Valor_Vo_2_1.Caption = ""  
Histeresis2.Conduccion_1_1.Caption = ""  
Histeresis2.Conduccion_2_1.Caption = ""
```

```
Else
```

```
'Asignamos las variables a los correspondientes parámetros del  
ejemplo 1 de un circuito comparador con histéresis y limitación en la  
salida'
```

```
R1_1 = Histeresis2.Valor_R1_1.Text  
R2_1 = Histeresis2.Valor_R2_1.Text  
Vi_1 = Histeresis2.Valor_Vi_1.Text  
Vref_1 = Histeresis2.Valor_Vref_1.Text  
Vu_1 = Histeresis2.Valor_Vu_1.Text  
VsatPos_1 = Histeresis2.Valor_VsatPos_1.Text  
VsatNeg_1 = Histeresis2.Valor_VsatNeg_1.Text
```

```
'Introducimos la condición de que -Vsat NO puede ser mayor que  
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación  
de Vo'
```

```
If VsatPos_1 < VsatNeg_1 Then
```



```
MsgBox "La tensión de saturación -Vsat no puede ser mayor que  
+Vsat", vbInformation + vbOKOnly, "AVISO"
```

```
Histeresis2.Valor_A_1.Caption = ""  
Histeresis2.Valor_B_1.Caption = ""  
Histeresis2.Valor_VsatPos_1.Text = ""  
Histeresis2.Valor_VsatNeg_1.Text = ""  
Histeresis2.Valor_Vlo_1_1.Caption = ""  
Histeresis2.Valor_Vo_1_1.Caption = ""  
Histeresis2.Valor_Vlo_2_1.Caption = ""  
Histeresis2.Valor_Vo_2_1.Caption = ""  
Histeresis2.Conduccion_1_1.Caption = ""  
Histeresis2.Conduccion_2_1.Caption = ""
```

```
Else
```

```
'Definimos las ecuaciones de los parametros a calcular y  
mostramos el resultado'
```

```
a_1 = (Vref_1 * R2_1 + VsatPos_1 * R1_1) / (R1_1 + R2_1)  
b_1 = (Vref_1 * R2_1 + VsatNeg_1 * R1_1) / (R1_1 + R2_1)
```

```
If Vi_1 > a_1 Then
```

```
Vlo_1_1 = VsatNeg_1  
Vo_1_1 = VsatNeg_1  
Histeresis2.Conduccion_1_1.Caption = "D no conduce"
```

```
ElseIf Vi_1 <= a_1 Then
```

```
Vlo_1_1 = VsatPos_1  
Vo_1_1 = Vu_1  
Histeresis2.Conduccion_1_1.Caption = "D conduce"
```

```
End If
```

```
If Vi_1 >= b_1 Then
```

```
Vlo_2_1 = VsatNeg_1  
Vo_2_1 = VsatNeg_1  
Histeresis2.Conduccion_2_1.Caption = "D no conduce"
```

```
ElseIf Vi_1 < b_1 Then
```

```
Vlo_2_1 = VsatPos_1  
Vo_2_1 = Vu_1  
Histeresis2.Conduccion_2_1.Caption = "D conduce"
```

```
End If
```

```
Histeresis2.Valor_A_1.Caption = a_1  
Histeresis2.Valor_B_1.Caption = b_1  
Histeresis2.Valor_Vlo_1_1.Caption = Vlo_1_1  
Histeresis2.Valor_Vo_1_1.Caption = Vo_1_1  
Histeresis2.Valor_Vlo_2_1.Caption = Vlo_2_1  
Histeresis2.Valor_Vo_2_1.Caption = Vo_2_1
```

```
'Analizamos si la tensión de salida está saturada'
```

```
If Vlo_1_1 >= VsatPos_1 Then
```

```
Histeresis2.Valor_Vlo_1_1.Caption = VsatPos_1
```



```
ElseIf Vlo_1_1 <= VsatNeg_1 Then
    Histeresis2.Valor_Vlo_1_1.Caption = VsatNeg_1
Else
    Histeresis2.Valor_Vlo_1_1.Caption = Vlo_1_1
End If

If Vlo_2_1 >= VsatPos_1 Then
    Histeresis2.Valor_Vlo_2_1.Caption = VsatPos_1
ElseIf Vlo_2_1 <= VsatNeg_1 Then
    Histeresis2.Valor_Vlo_2_1.Caption = VsatNeg_1
Else
    Histeresis2.Valor_Vlo_2_1.Caption = Vlo_2_1
End If

'Mostramos la representación Vo-Vi'
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s14") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("v14") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s16") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t16") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s17") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t17") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s23") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("v23") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s19") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t19") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s20") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t20") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s21") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t21") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s30") = a_1
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("v30") = b_1
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("w17") = a_1
- b_1
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s28") = Vu_1
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t28") = Vu_1
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s26") =
VsatsNeg_1
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t26") =
VsatsNeg_1
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s27") =
VsatsPos_1
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t27") =
VsatsPos_1

End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0
```



```
End Sub
```

```
Sub CalcularValoresHisteresis2()
```

```
'Declaramos las variables de los parámetros del ejemplo 2 de un  
circuito comparador con histéresis y limitación en la salida'
```

```
Dim R1_2 As Single
```

```
Dim R2_2 As Single
```

```
Dim Vi_2 As Single
```

```
Dim Vref_2 As Single
```

```
Dim Vu_2 As Single
```

```
Dim a_2 As Single
```

```
Dim b_2 As Single
```

```
Dim VsatPos_2 As Single
```

```
Dim VsatNeg_2 As Single
```

```
Dim Vlo_1_2 As Single
```

```
Dim Vo_1_2 As Single
```

```
Dim Vlo_2_2 As Single
```

```
Dim Vo_2_2 As Single
```

```
'Mediante esta declaración conseguimos que nuestra macro no pare de  
ejecutarse aunque se encuentre un error'
```

```
On Error Resume Next
```

```
'Nos aseguramos de que se introduzca un valor en todas las casillas'
```

```
'En el caso de las resistencias, su valor no puede ser cero o  
negativo'
```

```
'En el caso de las tensiones, su valor si puede ser cero o negativo'
```

```
If Histeresis2.Valor_R1_2.Text <= 0 Or Histeresis2.Valor_R2_2.Text <=  
0 Or Histeresis2.Valor_R1_2.Text = Empty Or  
Histeresis2.Valor_R2_2.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas de las  
resistencias, y que no sea ni nulo ni negativo", vbExclamation +  
vbOKOnly, "AVISO"
```

```
Histeresis2.Valor_A_2.Caption = ""
```

```
Histeresis2.Valor_B_2.Caption = ""
```

```
Histeresis2.Valor_Vlo_1_2.Caption = ""
```

```
Histeresis2.Valor_Vo_1_2.Caption = ""
```

```
Histeresis2.Valor_Vlo_2_2.Caption = ""
```

```
Histeresis2.Valor_Vo_2_2.Caption = ""
```

```
Histeresis2.Conduccion_1_2.Caption = ""
```

```
Histeresis2.Conduccion_2_2.Caption = ""
```

```
ElseIf Histeresis2.Valor_Vi_2.Text = Empty Or
```

```
Histeresis2.Valor_Vref_2.Text = Empty Or Histeresis2.Valor_Vu_2.Text =
```

```
Empty Or Histeresis2.Valor_VsatPos_2.Text = Empty Or
```

```
Histeresis2.Valor_VsatNeg_2.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas del Voltaje",  
vbExclamation + vbOKOnly, "AVISO"
```

```
Histeresis2.Valor_A_2.Caption = ""
```

```
Histeresis2.Valor_B_2.Caption = ""
```

```
Histeresis2.Valor_Vlo_1_2.Caption = ""
```

```
Histeresis2.Valor_Vo_1_2.Caption = ""
```

```
Histeresis2.Valor_Vlo_2_2.Caption = ""
```

```
Histeresis2.Valor_Vo_2_2.Caption = ""
```

```
Histeresis2.Conduccion_1_2.Caption = ""
```




```
Histeresis2.Conduccion_2_2.Caption = ""

Else

    'Asignamos las variables a los correspondientes parámetros del
ejemplo 2 de un circuito comparador con histéresis y limitación en la
salida'
    R1_2 = Histeresis2.Valor_R1_2.Text
    R2_2 = Histeresis2.Valor_R2_2.Text
    Vi_2 = Histeresis2.Valor_Vi_2.Text
    Vref_2 = Histeresis2.Valor_Vref_2.Text
    Vu_2 = Histeresis2.Valor_Vu_2.Text
    VsatPos_2 = Histeresis2.Valor_VsatPos_2.Text
    VsatNeg_2 = Histeresis2.Valor_VsatNeg_2.Text

    'Introducimos la condición de que -Vsat NO puede ser mayor que
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
de Vo'
    If VsatPos_2 < VsatNeg_2 Then

        MsgBox "La tensión de saturación -Vsat no puede ser mayor que
+Vsat", vbInformation + vbOKOnly, "AVISO"

        Histeresis2.Valor_VsatPos_2.Text = ""
        Histeresis2.Valor_VsatNeg_2.Text = ""
        Histeresis2.Valor_A_2.Caption = ""
        Histeresis2.Valor_B_2.Caption = ""
        Histeresis2.Valor_Vlo_1_2.Caption = ""
        Histeresis2.Valor_Vo_1_2.Caption = ""
        Histeresis2.Valor_Vlo_2_2.Caption = ""
        Histeresis2.Valor_Vo_2_2.Caption = ""
        Histeresis2.Conduccion_1_2.Caption = ""
        Histeresis2.Conduccion_2_2.Caption = ""

    Else

        'Definimos las ecuaciones de los parametros a calcular y
mostramos el resultado'
        a_2 = (Vref_2 * R2_2 + VsatPos_2 * R1_2) / (R1_2 + R2_2)
        b_2 = (Vref_2 * R2_2 + VsatNeg_2 * R1_2) / (R1_2 + R2_2)

        If Vi_2 > a_2 Then

            Vlo_1_2 = VsatNeg_2
            Vo_1_2 = -Vu_2
            Histeresis2.Conduccion_1_2.Caption = "D conduce"

        ElseIf Vi_2 <= a_2 Then

            Vlo_1_2 = VsatPos_2
            Vo_1_2 = VsatPos_2
            Histeresis2.Conduccion_1_2.Caption = "D no conduce"

        End If

        If Vi_2 >= b_2 Then

            Vlo_2_2 = VsatNeg_2
            Vo_2_2 = -Vu_2
            Histeresis2.Conduccion_2_2.Caption = "D conduce"
```



```
ElseIf Vi_2 < b_2 Then

    Vlo_2_2 = VsatPos_2
    Vo_2_2 = VsatPos_2
    Histeresis2.Conduccion_2_2.Caption = "D no conduce"

End If

Histeresis2.Valor_A_2.Caption = a_2
Histeresis2.Valor_B_2.Caption = b_2
Histeresis2.Valor_Vlo_1_2.Caption = Vlo_1_2
Histeresis2.Valor_Vo_1_2.Caption = Vo_1_2
Histeresis2.Valor_Vlo_2_2.Caption = Vlo_2_2
Histeresis2.Valor_Vo_2_2.Caption = Vo_2_2

'Analizamos si la tensión de salida está saturada'
If Vlo_1_2 >= VsatPos_2 Then

    Histeresis2.Valor_Vlo_1_2.Caption = VsatPos_2

ElseIf Vlo_1_2 <= VsatNeg_2 Then

    Histeresis2.Valor_Vlo_1_2.Caption = VsatNeg_2

Else

    Histeresis2.Valor_Vlo_1_2.Caption = Vlo_1_2

End If

If Vlo_2_2 >= VsatPos_2 Then

    Histeresis2.Valor_Vlo_2_2.Caption = VsatPos_2

ElseIf Vlo_2_2 <= VsatNeg_2 Then

    Histeresis2.Valor_Vlo_2_2.Caption = VsatNeg_2

Else

    Histeresis2.Valor_Vlo_2_2.Caption = Vlo_2_2

End If

'Mostramos la representación Vo-Vi'
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s14") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("v14") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s16") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t16") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s17") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t17") = ""
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s23") = a_2
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("v23") = b_2
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("w17") = a_2
- b_2
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s19") =
VsatNeg_2
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t19") =
VsatNeg_2
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s20") =
VsatPos_2
```



```
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t20") =  
VsatPos_2  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s21") = -  
Vu_2  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t21") = -  
Vu_2  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s30") = ""  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("v30") = ""  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s28") = ""  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t28") = ""  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s26") = ""  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t26") = ""  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("s27") = ""  
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("t27") = ""
```

End If

End If

```
'Desactivamos el detector de errores'  
On Error GoTo 0
```

End Sub

Sub RestaurarValoresHisteresis1()

```
'Vaciamos los valores de cada uno de los parámetros'
```

```
Histeresis2.Valor_R1_1.Text = ""  
Histeresis2.Valor_R2_1.Text = ""  
Histeresis2.Valor_Vi_1.Text = ""  
Histeresis2.Valor_Vref_1.Text = ""  
Histeresis2.Valor_Vu_1.Text = ""  
Histeresis2.Valor_VsatPos_1.Text = ""  
Histeresis2.Valor_VsatNeg_1.Text = ""  
Histeresis2.Valor_A_1.Caption = ""  
Histeresis2.Valor_B_1.Caption = ""  
Histeresis2.Valor_Vlo_1_1.Caption = ""  
Histeresis2.Valor_Vo_1_1.Caption = ""  
Histeresis2.Valor_Vlo_2_1.Caption = ""  
Histeresis2.Valor_Vo_2_1.Caption = ""  
Histeresis2.Conduccion_1_1.Caption = ""  
Histeresis2.Conduccion_2_1.Caption = ""
```

End Sub

Sub RestaurarValoresHisteresis2()

```
'Vaciamos los valores de cada uno de los parámetros'
```

```
Histeresis2.Valor_R1_2.Text = ""  
Histeresis2.Valor_R2_2.Text = ""  
Histeresis2.Valor_Vi_2.Text = ""  
Histeresis2.Valor_Vref_2.Text = ""  
Histeresis2.Valor_Vu_2.Text = ""  
Histeresis2.Valor_VsatPos_2.Text = ""  
Histeresis2.Valor_VsatNeg_2.Text = ""  
Histeresis2.Valor_A_2.Caption = ""  
Histeresis2.Valor_B_2.Caption = ""  
Histeresis2.Valor_Vlo_1_2.Caption = ""  
Histeresis2.Valor_Vo_1_2.Caption = ""  
Histeresis2.Valor_Vlo_2_2.Caption = ""  
Histeresis2.Valor_Vo_2_2.Caption = ""
```



```
Histeresis2.Conduccion_1_2.Caption = ""  
Histeresis2.Conduccion_2_2.Caption = ""
```

```
End Sub
```

7.2.3.12. Integrador inversor (ModIntInv)

```
Sub MostrarIntegradorInversor()
```

```
'Mostramos la pestaña de la aplicación del circuito integrador  
inversor'  
IntegradorInversor.Show
```

```
End Sub
```

```
Sub MostrarTeoriaIntInv()
```

```
'Mostramos la pestaña de la teoría del circuito integrador inversor'  
TeoriaIntInv.Show
```

```
End Sub
```

```
Sub MostrarInformacion1()
```

```
'Mostramos la pestaña de la información del circuito integrador  
inversor'  
InformacionInt1.Show
```

```
End Sub
```

```
Sub MostrarInformacion2()
```

```
'Mostramos la pestaña de la información del circuito integrador  
inversor'  
InformacionInt1.Show
```

```
End Sub
```

```
Sub CalcularValores1()
```

```
'Declaramos las variables de los parámetros de un circuito integrador  
inversor'
```

```
Dim R_1 As Single  
Dim C_1 As Single  
Dim Vi_1 As Single  
Dim n_1 As Single  
Dim Integral_1 As Single  
Dim VsatPos_1 As Single  
Dim VsatNeg_1 As Single  
Dim Vo_Inicial_1 As Single  
Dim t_final_1 As Single  
Dim t_sat_1 As Single  
Dim Vo_1 As Single
```

```
'Mediante esta declaración conseguimos que nuestra macro no pare de  
ejecutarse aunque se encuentre un error'
```

```
On Error Resume Next
```



```
'Nos aseguramos de que se introduzca un valor en todas las casillas'  
'En el caso de la resistencia, el condensador y el valor de t final,  
su valor no puede ser cero o negativo'  
'La potencia de la tensión de entrada no puede ser menor o igual que -  
1'  
'En el caso de las tensiones, su valor si puede ser cero o negativo'  
If IntegradorInversor.Valor_R_1.Text <= 0 Or  
IntegradorInversor.Valor_C_1.Text <= 0 Or  
IntegradorInversor.Valor_R_1.Text = Empty Or  
IntegradorInversor.Valor_C_1.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas de la resistencia  
y el condensador, y que no sea ni nulo ni negativo", vbExclamation +  
vbOKOnly, "AVISO"
```

```
IntegradorInversor.Valor_Vo_1.Caption = ""  
IntegradorInversor.Valor_Saturacion_1.Caption = ""
```

```
ElseIf IntegradorInversor.Valor_Vi_1.Text = Empty Or  
IntegradorInversor.Valor_VsatPos_1.Text = Empty Or  
IntegradorInversor.Valor_VsatNeg_1.Text = Empty Or  
IntegradorInversor.Valor_VoInicial_1.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas del Voltaje",  
vbExclamation + vbOKOnly, "AVISO"
```

```
IntegradorInversor.Valor_Vo_1.Caption = ""  
IntegradorInversor.Valor_Saturacion_1.Caption = ""
```

```
ElseIf IntegradorInversor.n_1.Text <= -1 Or  
IntegradorInversor.n_1.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en la casilla de la potencia del  
tiempo y que no sea nulo", vbExclamation + vbOKOnly, "AVISO"
```

```
IntegradorInversor.Valor_Vo_1.Caption = ""  
IntegradorInversor.Valor_Saturacion_1.Caption = ""
```

```
ElseIf IntegradorInversor.t_final_1.Text <= 0 Or  
IntegradorInversor.t_final_1.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en la casilla del tiempo referido  
a la condición final y que no sea nulo o negativo", vbExclamation +  
vbOKOnly, "AVISO"
```

```
IntegradorInversor.Valor_Vo_1.Caption = ""  
IntegradorInversor.Valor_Saturacion_1.Caption = ""
```

```
Else
```

```
'Asignamos las variables a los correspondientes parámetros del  
circuito integrador inversor'
```

```
R_1 = IntegradorInversor.Valor_R_1.Text * 1000  
C_1 = IntegradorInversor.Valor_C_1.Text / 1000000  
Vi_1 = IntegradorInversor.Valor_Vi_1.Text  
n_1 = IntegradorInversor.n_1.Text  
VsatPos_1 = IntegradorInversor.Valor_VsatPos_1.Text  
VsatNeg_1 = IntegradorInversor.Valor_VsatNeg_1.Text  
Vo_Inicial_1 = IntegradorInversor.Valor_VoInicial_1.Text  
t_final_1 = IntegradorInversor.t_final_1.Text
```



```
'Introducimos la condición de que -Vsat NO puede ser mayor que
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
de Vo'
If VsatPos_1 < VsatNeg_1 Then

    MsgBox "La tensión de saturación -Vsat no puede ser mayor que
+Vsat", vbInformation + vbOKOnly, "AVISO"

    IntegradorInversor.Valor_VsatPos_1.Text = ""
    IntegradorInversor.Valor_VsatNeg_1.Text = ""
    IntegradorInversor.Valor_Vo_1.Caption = ""
    IntegradorInversor.Valor_Saturacion_1.Caption = ""

Else

    'Definimos las ecuaciones de los parametros a calcular y
mostramos el resultado'
    Integral_1 = ((1 / (n_1 + 1)) * t_final_1 ^ (n_1 + 1))
    Vo_1 = (-(Vi_1 / (R_1 * C_1)) * Integral_1) + Vo_Inicial_1

    IntegradorInversor.Valor_Vo_1.Caption = Vo_1

    'Analizamos si la tensión de salida está saturada'
    If Vo_1 >= VsatPos_1 Then

        IntegradorInversor.Valor_Vo_1.Caption = VsatPos_1
        IntegradorInversor.Valor_Saturacion_1.Caption = "Salida
saturada"

    ElseIf Vo_1 <= VsatNeg_1 Then

        IntegradorInversor.Valor_Vo_1.Caption = VsatNeg_1
        IntegradorInversor.Valor_Saturacion_1.Caption = "Salida
saturada"

    Else

        IntegradorInversor.Valor_Saturacion_1.Caption = ""
        IntegradorInversor.Valor_Vo_1.Caption = Vo_1

    End If

    'Mostramos la representación Vi-t y Vo-t'

    'Calculamos el valor de t en el que Vo satura, ya que serán
necesarios para obtener la gráfica'
    If Vi_1 > 0 Then

        t_sat_1 = ((Abs(VsatNeg_1 - Vo_Inicial_1) * R_1 * C_1 *
Abs((n_1 + 1))) / Vi_1) ^ (1 / (n_1 + 1))

    ElseIf Vi_1 < 0 Then

        t_sat_1 = ((Abs(VsatPos_1 - Vo_Inicial_1) * R_1 * C_1 *
Abs((n_1 + 1))) / Abs(Vi_1)) ^ (1 / (n_1 + 1))

    End If

    'Definimos los puntos necesarios para obtener nuestra gráfica'
```



```
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("a8:a728") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("e8:e728") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("f8:f728") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("p5") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("q5") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("p6") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("q6") =  
Empty  
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("o6") =  
Empty  
  
If n_1 = 0 Then  
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah14")  
= Vi_1  
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah15")  
= n_1  
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag17")  
= Empty  
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah17")  
= Empty  
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag16")  
= Vi_1  
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah16")  
= Vi_1  
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag19")  
= 0  
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah19")  
= t_final_1 + 3  
  
ElseIf n_1 > 0 Then  
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah14")  
= Vi_1  
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah15")  
= n_1  
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag17")  
= t_final_1  
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah17")  
= t_final_1 + 3  
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag16")  
= Empty  
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah16")  
= Empty  
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag19")  
= Empty  
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah19")  
= Empty  
  
End If  
  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c7") = Vi_1  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c8") = R_1  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c9") = C_1  
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c10") = n_1
```



```
        ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c11") =  
Vo_Inicial_1  
        ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c14") =  
VsatNeg_1  
        ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c13") =  
VsatPos_1  
        ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c16") =  
t_sat_1  
        ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("d16") =  
t_sat_1 + 3  
        ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c15") = 0  
        ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("d15") = -3  
        ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("s8:s728") =  
Empty  
        ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("y8:y728") =  
Empty  
        ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("z8:z728") =  
Empty  
  
    End If  
  
End If  
  
'Desactivamos el detector de errores'  
On Error GoTo 0  
  
End Sub  
  
Sub CalcularValores2()  
  
'Declaramos las variables de los parámetros de un circuito integrador  
inversor'  
Dim R_2 As Single  
Dim C_2 As Single  
Dim a As Single  
Dim Sen_Cos As String  
Dim b As Single  
Dim Integral_2 As Single  
Dim VsatPos_2 As Single  
Dim VsatNeg_2 As Single  
Dim Vo_Inicial_2 As Single  
Dim t_final_2 As Single  
Dim Vo_2 As Single  
  
'Mediante esta declaración conseguimos que nuestra macro no pare de  
ejecutarse aunque se encuentre un error'  
On Error Resume Next  
  
'Nos aseguramos de que se introduzca un valor en todas las casillas'  
'En el caso de la resistencia, el condensador y el valor de t final,  
su valor no puede ser cero o negativo'  
'En el caso de las tensiones, su valor si puede ser cero o negativo'  
If IntegradorInversor.Valor_R_2.Text <= 0 Or  
IntegradorInversor.Valor_C_2.Text <= 0 Or  
IntegradorInversor.Valor_R_2.Text = Empty Or  
IntegradorInversor.Valor_C_2.Text = Empty Then  
  
    MsgBox "Debe introducir un valor en las casillas de la resistencia  
y el condensador, y que no sea ni nulo ni negativo", vbExclamation +  
vbOKOnly, "AVISO"
```




```
IntegradorInversor.Valor_Vo_2.Caption = ""
IntegradorInversor.Valor_Saturacion_2.Caption = ""

ElseIf IntegradorInversor.a.Text = Empty Or IntegradorInversor.b.Text
= Empty Or IntegradorInversor.Valor_VsatPos_2.Text = Empty Or
IntegradorInversor.Valor_VsatNeg_2.Text = Empty Or
IntegradorInversor.Valor_VoInicial_2.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas del Voltaje",
vbExclamation + vbOKOnly, "AVISO"

    IntegradorInversor.Valor_Vo_2.Caption = ""
    IntegradorInversor.Valor_Saturacion_2.Caption = ""

ElseIf IntegradorInversor.t_final_2.Text <= 0 Or
IntegradorInversor.t_final_2.Text = Empty Then

    MsgBox "Debe introducir un valor en la casilla del tiempo referido
a la condición final y que no sea nulo o negativo", vbExclamation +
vbOKOnly, "AVISO"

    IntegradorInversor.Valor_Vo_2.Caption = ""
    IntegradorInversor.Valor_Saturacion_2.Caption = ""

Else

    'Asignamos las variables a los correspondientes parámetros del
circuito integrador inversor'
    R_2 = IntegradorInversor.Valor_R_2.Text * 1000
    C_2 = IntegradorInversor.Valor_C_2.Text / 1000000
    a = IntegradorInversor.a.Text
    b = IntegradorInversor.b.Text
    Sen_Cos = IntegradorInversor.Sen_Cos.Text
    VsatPos_2 = IntegradorInversor.Valor_VsatPos_2.Text
    VsatNeg_2 = IntegradorInversor.Valor_VsatNeg_2.Text
    Vo_Inicial_2 = IntegradorInversor.Valor_VoInicial_2.Text
    t_final_2 = IntegradorInversor.t_final_2.Text

    'Introducimos la condición de que -Vsat NO puede ser mayor que
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
de Vo'
    If VsatPos_2 < VsatNeg_2 Then

        MsgBox "La tensión de saturación -Vsat no puede ser mayor que
+Vsat", vbInformation + vbOKOnly, "AVISO"

        IntegradorInversor.Valor_VsatPos_2.Text = ""
        IntegradorInversor.Valor_VsatNeg_2.Text = ""
        IntegradorInversor.Valor_Vo_2.Caption = ""
        IntegradorInversor.Valor_Saturacion_2.Caption = ""

    Else

        'Definimos las ecuaciones de los parametros a calcular y
mostramos el resultado'

        If IntegradorInversor.Sen_Cos.Text = "sen" Then

            Integral_2 = -(a / b) * Cos(b * t_final_2)
            Vo_2 = -(1 / (R_2 * C_2)) * Integral_2 + Vo_Inicial_2
```



```
ElseIf IntegradorInversor.Sen_Cos.Text = "cos" Then

    Integral_2 = (a / b) * Sin(b * t_final_2)
    Vo_2 = -(1 / (R_2 * C_2)) * Integral_2 + Vo_Inicial_2

Else

    Integral_2 = ""
    Vo_2 = ""

End If

IntegradorInversor.Valor_Vo_2.Caption = Vo_2

'Analizamos si la tensión de salida está saturada'
If Vo_2 >= VsatPos_2 Then

    IntegradorInversor.Valor_Vo_2.Caption = VsatPos_2
    IntegradorInversor.Valor_Saturacion_2.Caption = "Salida
saturada"

ElseIf Vo_2 <= VsatNeg_2 Then

    IntegradorInversor.Valor_Vo_2.Caption = VsatNeg_2
    IntegradorInversor.Valor_Saturacion_2.Caption = "Salida
saturada"

Else

    IntegradorInversor.Valor_Saturacion_2.Caption = ""
    IntegradorInversor.Valor_Vo_2.Caption = Vo_2

End If

'Mostramos la representación Vi-t y Vo-t'

'Definimos los puntos necesarios para obtener nuestra gráfica'
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("q8") = R_2
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("q9") = R_2
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("r8") = C_2
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("r9") = C_2
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("p5") = 0
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("q5") = -2
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("p6") =
Vo_Inicial_2
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("q6") =
Vo_Inicial_2
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("p8") =
Vo_Inicial_2
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("p9") =
Vo_Inicial_2
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n2") =
VsatPos_2
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n3") =
VsatNeg_2
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag17") =
Empty
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah17") =
Empty
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag16") =
Empty
```



```

ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah16") =
Empty
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag19") =
Empty
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah19") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c16") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c15") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("d15") = Empty

Dim t_max As Integer
Dim grados1 As Integer
Dim grados2 As Integer

t_max = (t_final_2 / 0.017453293) + 200
grados1 = 0
grados2 = 0

If IntegradorInversor.Sen_Cos.Text = "sen" Then

    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("s8") =
a
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("o8") =
b
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("s9") =
Empty
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("o9") =
Empty
    ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("a8:a728") = Empty
    ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("e8:e728") = Empty
    ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("f8:f728") = Empty

    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("s8:s728")
= Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("y8:y728")
= Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("z8:z728")
= Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-
t").Range("aa8:aa728") = Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-
t").Range("ab8:ab728") = Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("w5") =
Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("x5") =
Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("y5") =
Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("z5") =
Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("aa5") =
Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("ab5") =
Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("u5") = a
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v5") = b

```



```
'Desarrollamos el cñico con el que la gráfica muestra si la
tensión de salida está saturada'
Dim i1 As Integer

For i1 = 8 To t_max

    Worksheets("Gráfica Vo-t").Cells(i1, 19).Value =
grados1
    Worksheets("Gráfica CosSen").Cells(i1, 1).Value =
grados1
    grados1 = grados1 + 1

    Worksheets("Gráfica Vo-t").Cells(i1, 25).Value =
Worksheets("Gráfica Vo-t").Cells(i1, 29).Value
    Worksheets("Gráfica CosSen").Cells(i1, 6).Value =
Worksheets("Gráfica CosSen").Cells(i1, 26).Value

    If Worksheets("Gráfica CosSen").Cells(i1, 6).Value >=
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n2").Value Then

        Worksheets("Gráfica CosSen").Cells(i1, 6).Value =
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n2").Value

    ElseIf Worksheets("Gráfica CosSen").Cells(i1, 6).Value
<= ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n3").Value Then

        Worksheets("Gráfica CosSen").Cells(i1, 6).Value =
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n3").Value

    Else

        Worksheets("Gráfica CosSen").Cells(i1, 6).Value =
Worksheets("Gráfica CosSen").Cells(i1, 26).Value

    End If

Next i1

ActiveWorkbook.Worksheets("Gráfica CosSen").Range("o6") =
Worksheets("Gráfica CosSen").Cells(8, 6).Value

ElseIf IntegradorInversor.Sen_Cos.Text = "cos" Then

    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("s9") =
a
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("o9") =
b
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("s8") =
Empty
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("o8") =
Empty
    ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("a8:a728") = Empty
    ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("e8:e728") = Empty
    ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("f8:f728") = Empty

    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("s8:s728")
= Empty
```



```
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("y8:y728")
= Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("z8:z728")
= Empty
ActiveWorkbook.Worksheets("Gráfica Vo-
t").Range("aa8:aa728") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-
t").Range("ab8:ab728") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("u5") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v5") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("y5") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("z5") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("aa5") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("ab5") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("w5") = a
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("x5") = b

'Desarrollamos el código con el que la gráfica muestra si
la tensión de salida está saturada'
Dim i2 As Integer

For i2 = 8 To t_max

    Worksheets("Gráfica Vo-t").Cells(i2, 19).Value =
grados2
    Worksheets("Gráfica CosSen").Cells(i2, 1).Value =
grados2
    grados2 = grados2 + 1

    Worksheets("Gráfica Vo-t").Cells(i2, 26).Value =
Worksheets("Gráfica Vo-t").Cells(i2, 30).Value
    Worksheets("Gráfica CosSen").Cells(i2, 5).Value =
Worksheets("Gráfica CosSen").Cells(i2, 25).Value

    If Worksheets("Gráfica CosSen").Cells(i2, 5).Value >=
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n2").Value Then

        Worksheets("Gráfica CosSen").Cells(i2, 5) =
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n2").Value

    ElseIf Worksheets("Gráfica CosSen").Cells(i2, 5).Value
<= ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n3").Value Then

        Worksheets("Gráfica CosSen").Cells(i2, 5).Value =
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n3").Value

    Else

        Worksheets("Gráfica CosSen").Cells(i2, 5).Value =
Worksheets("Gráfica CosSen").Cells(i2, 25).Value

    End If

Next i2
```



```
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("o6") =  
Worksheets("Gráfica CosSen").Cells(8, 5).Value  
  
End If  
  
End If  
  
End If  
  
'Desactivamos el detector de errores'  
On Error GoTo 0  
  
End Sub  
  
Sub RestaurarValores1()  
  
'Vaciamos los valores de cada uno de los parámetros'  
IntegradorInversor.Valor_R_1.Text = ""  
IntegradorInversor.Valor_C_1.Text = ""  
IntegradorInversor.Valor_Vi_1.Text = ""  
IntegradorInversor.n_1.Text = ""  
IntegradorInversor.Valor_VsatPos_1.Text = ""  
IntegradorInversor.Valor_VsatNeg_1.Text = ""  
IntegradorInversor.Valor_VoInicial_1.Text = ""  
IntegradorInversor.t_final_1.Text = ""  
IntegradorInversor.Valor_Vo_1.Caption = ""  
IntegradorInversor.Valor_Saturacion_1.Caption = ""  
  
End Sub  
  
Sub RestaurarValores2()  
  
'Vaciamos los valores de cada uno de los parámetros'  
IntegradorInversor.Valor_R_2.Text = ""  
IntegradorInversor.Valor_C_2.Text = ""  
IntegradorInversor.a.Text = ""  
IntegradorInversor.b.Text = ""  
IntegradorInversor.Valor_VsatPos_2.Text = ""  
IntegradorInversor.Valor_VsatNeg_2.Text = ""  
IntegradorInversor.Valor_VoInicial_2.Text = ""  
IntegradorInversor.t_final_2.Text = ""  
IntegradorInversor.Valor_Vo_2.Caption = ""  
IntegradorInversor.Valor_Saturacion_2.Caption = ""  
  
End Sub
```

7.2.3.13. Integrador no inversor (ModIntNoInv)

```
Sub MostrarIntegradorNoInversor()  
  
'Mostramos la pestaña de la aplicación del circuito integrador no  
inversor'  
IntegradorNoInversor.Show
```



```
End Sub

Sub MostrarTeoriaIntNoInv()

'Mostramos la pestaña de la teoría del circuito integrador no
inversor'
TeoriaIntNoInv.Show

End Sub

Sub MostrarInformacion1()

'Mostramos la pestaña de la información del circuito integrador no
inversor'
InformacionInt1.Show

End Sub

Sub MostrarInformacion2()

'Mostramos la pestaña de la información del circuito integrador no
inversor'
InformacionInt1.Show

End Sub

Sub CalcularValores1()

'Declaramos las variables de los parámetros de un circuito integrador
no inversor'
Dim R_1 As Single
Dim C_1 As Single
Dim Vi_1 As Single
Dim n_1 As Single
Dim Integral_1 As Single
Dim VsatPos_1 As Single
Dim VsatNeg_1 As Single
Dim Vo_Inicial_1 As Single
Dim t_final_1 As Single
Dim t_sat_1 As Single
Dim Vo_1 As Single

'Mediante esta declaración conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de la resistencia, el condensador y el valor de t final,
su valor no puede ser cero o negativo'
'La potencia de la tensión de entrada no puede ser menor o igual que -
1'
'En el caso de las tensiones, su valor si puede ser cero o negativo'
If IntegradorNoInversor.Valor_R_1.Text <= 0 Or
IntegradorNoInversor.Valor_C_1.Text <= 0 Or
IntegradorNoInversor.Valor_R_1.Text = Empty Or
IntegradorNoInversor.Valor_C_1.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas de la resistencia
y el condensador, y que no sea ni nulo ni negativo", vbExclamation +
vbOKOnly, "AVISO"
```



```
IntegradorNoInversor.Valor_Vo_1.Caption = ""
IntegradorNoInversor.Valor_Saturacion_1.Caption = ""

ElseIf IntegradorNoInversor.Valor_Vi_1.Text = Empty Or
IntegradorNoInversor.Valor_VsatPos_1.Text = Empty Or
IntegradorNoInversor.Valor_VsatNeg_1.Text = Empty Or
IntegradorNoInversor.Valor_VoInicial_1.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas del Voltaje",
vbExclamation + vbOKOnly, "AVISO"

    IntegradorNoInversor.Valor_Vo_1.Caption = ""
    IntegradorNoInversor.Valor_Saturacion_1.Caption = ""

ElseIf IntegradorNoInversor.n_1.Text <= -1 Or
IntegradorNoInversor.n_1.Text = Empty Then

    MsgBox "Debe introducir un valor en la casilla de la potencia del
tiempo y que no sea nulo", vbExclamation + vbOKOnly, "AVISO"

    IntegradorNoInversor.Valor_Vo_1.Caption = ""
    IntegradorNoInversor.Valor_Saturacion_1.Caption = ""

ElseIf IntegradorNoInversor.t_final_1.Text <= 0 Or
IntegradorNoInversor.t_final_1.Text = Empty Then

    MsgBox "Debe introducir un valor en la casilla del tiempo referido
a la condición final y que no sea nulo o negativo", vbExclamation +
vbOKOnly, "AVISO"

    IntegradorNoInversor.Valor_Vo_1.Caption = ""
    IntegradorNoInversor.Valor_Saturacion_1.Caption = ""

Else

    'Asignamos las variables a los correspondientes parámetros del
circuito integrador no inversor'
    R_1 = IntegradorNoInversor.Valor_R_1.Text * 1000
    C_1 = IntegradorNoInversor.Valor_C_1.Text / 1000000
    Vi_1 = IntegradorNoInversor.Valor_Vi_1.Text
    n_1 = IntegradorNoInversor.n_1.Text
    VsatPos_1 = IntegradorNoInversor.Valor_VsatPos_1.Text
    VsatNeg_1 = IntegradorNoInversor.Valor_VsatNeg_1.Text
    Vo_Inicial_1 = IntegradorNoInversor.Valor_VoInicial_1.Text
    t_final_1 = IntegradorNoInversor.t_final_1.Text

    'Introducimos la condición de que -Vsat NO puede ser mayor que
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
de Vo'
    If VsatPos_1 < VsatNeg_1 Then

        MsgBox "La tensión de saturación -Vsat no puede ser mayor que
+Vsat", vbInformation + vbOKOnly, "AVISO"

        IntegradorNoInversor.Valor_VsatPos_1.Text = ""
        IntegradorNoInversor.Valor_VsatNeg_1.Text = ""
        IntegradorNoInversor.Valor_Vo_1.Caption = ""
        IntegradorNoInversor.Valor_Saturacion_1.Caption = ""

    Else
```




```
'Definimos las ecuaciones de los parametros a calcular y
mostramos el resultado'
Integral_1 = ((1 / (n_1 + 1)) * t_final_1 ^ (n_1 + 1))
Vo_1 = ((2 * Vi_1) / (R_1 * C_1) * Integral_1) + Vo_Inicial_1

IntegradorNoInversor.Valor_Vo_1.Caption = Vo_1

'Analizamos si la tensión de salida está saturada'
If Vo_1 >= VsatPos_1 Then

    IntegradorNoInversor.Valor_Vo_1.Caption = VsatPos_1
    IntegradorNoInversor.Valor_Saturacion_1.Caption = "Salida
saturada"

ElseIf Vo_1 <= VsatNeg_1 Then

    IntegradorNoInversor.Valor_Vo_1.Caption = VsatNeg_1
    IntegradorNoInversor.Valor_Saturacion_1.Caption = "Salida
saturada"

Else

    IntegradorNoInversor.Valor_Saturacion_1.Caption = ""
    IntegradorNoInversor.Valor_Vo_1.Caption = Vo_1

End If

'Mostramos la representación Vi-t y Vo-t'

'Calculamos el valor de t en el que Vo satura, ya que serán
necesarios para obtener la gráfica'
If Vi_1 > 0 Then

    t_sat_1 = ((Abs(VsatPos_1 - Vo_Inicial_1) * R_1 * C_1 *
Abs((n_1 + 1))) / (2 * Vi_1)) ^ (1 / (n_1 + 1))

ElseIf Vi_1 < 0 Then

    t_sat_1 = ((Abs(VsatNeg_1 - Vo_Inicial_1) * R_1 * C_1 *
Abs((n_1 + 1))) / (2 * Abs(Vi_1))) ^ (1 / (n_1 + 1))

End If

'Definimos los puntos necesarios para obtener nuestra gráfica'
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("a8:a728") =
Empty
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("i8:i728") =
Empty
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("j8:j728") =
Empty
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("p5") =
Empty
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("q5") =
Empty
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("p6") =
Empty
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("q6") =
Empty
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("o6") =
Empty
```



```
    If n_1 = 0 Then
        ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah14")
= Vi_1
        ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah15")
= n_1
        ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag17")
= Empty
        ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah17")
= Empty
        ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag16")
= Vi_1
        ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah16")
= Vi_1
        ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag19")
= 0
        ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah19")
= t_final_1 + 3
    ElseIf n_1 > 0 Then
        ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah14")
= Vi_1
        ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah15")
= n_1
        ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag17")
= t_final_1
        ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah17")
= t_final_1 + 3
        ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag16")
= Empty
        ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah16")
= Empty
        ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag19")
= Empty
        ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah19")
= Empty
    End If

    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c7") = Vi_1
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c8") = R_1
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c9") = C_1
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c10") = n_1
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c12") =
Vo_Inicial_1
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c14") =
VsatNeg_1
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c13") =
VsatPos_1
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c17") =
t_sat_1
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("d17") =
t_sat_1 + 3
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c15") = 0
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("d15") = -3
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("s8:s728") =
Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("aa8:aa728") =
Empty
```



```
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("ab8:ab728") =  
Empty  
  
End If  
  
End If  
  
'Desactivamos el detector de errores'  
On Error GoTo 0  
  
End Sub  
  
Sub CalcularValores2()  
  
'Declaramos las variables de los parámetros de un circuito integrador  
inversor'  
Dim R_2 As Single  
Dim C_2 As Single  
Dim a As Single  
Dim Sen_Cos As String  
Dim b As Single  
Dim Integral_2 As Single  
Dim VsatPos_2 As Single  
Dim VsatNeg_2 As Single  
Dim Vo_Inicial_2 As Single  
Dim t_final_2 As Single  
Dim Vo_2 As Single  
  
'Mediante esta declaración conseguimos que nuestra macro no pare de  
ejecutarse aunque se encuentre un error'  
On Error Resume Next  
  
'Nos aseguramos de que se introduzca un valor en todas las casillas'  
'En el caso de la resistencia, el condensador y el valor de t final,  
su valor no puede ser cero o negativo'  
'En el caso de las tensiones, su valor si puede ser cero o negativo'  
If IntegradorNoInversor.Valor_R_2.Text <= 0 Or  
IntegradorNoInversor.Valor_C_2.Text <= 0 Or  
IntegradorNoInversor.Valor_R_2.Text = Empty Or  
IntegradorNoInversor.Valor_C_2.Text = Empty Then  
  
MsgBox "Debe introducir un valor en las casillas de la resistencia  
y el condensador, y que no sea ni nulo ni negativo", vbExclamation +  
vbOKOnly, "AVISO"  
  
IntegradorNoInversor.Valor_Vo_2.Caption = ""  
IntegradorNoInversor.Valor_Saturacion_2.Caption = ""  
  
ElseIf IntegradorNoInversor.a.Text = Empty Or  
IntegradorNoInversor.b.Text = Empty Or  
IntegradorNoInversor.Valor_VsatPos_2.Text = Empty Or  
IntegradorNoInversor.Valor_VsatNeg_2.Text = Empty Or  
IntegradorNoInversor.Valor_VoInicial_2.Text = Empty Then  
  
MsgBox "Debe introducir un valor en las casillas del Voltaje",  
vbExclamation + vbOKOnly, "AVISO"  
  
IntegradorNoInversor.Valor_Vo_2.Caption = ""  
IntegradorNoInversor.Valor_Saturacion_2.Caption = ""
```



```
ElseIf IntegradorNoInversor.t_final_2.Text <= 0 Or  
IntegradorNoInversor.t_final_2.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en la casilla del tiempo referido  
a la condición final y que no sea nulo o negativo", vbExclamation +  
vbOKOnly, "AVISO"
```

```
IntegradorNoInversor.Valor_Vo_2.Caption = ""  
IntegradorNoInversor.Valor_Saturacion_2.Caption = ""
```

```
Else
```

```
'Asignamos las variables a los correspondientes parámetros del  
circuito integrador inversor'
```

```
R_2 = IntegradorNoInversor.Valor_R_2.Text * 1000  
C_2 = IntegradorNoInversor.Valor_C_2.Text / 1000000  
a = IntegradorNoInversor.a.Text  
b = IntegradorNoInversor.b.Text  
Sen_Cos = IntegradorNoInversor.Sen_Cos.Text  
VsatPos_2 = IntegradorNoInversor.Valor_VsatPos_2.Text  
VsatNeg_2 = IntegradorNoInversor.Valor_VsatNeg_2.Text  
Vo_Inicial_2 = IntegradorNoInversor.Valor_VoInicial_2.Text  
t_final_2 = IntegradorNoInversor.t_final_2.Text
```

```
'Introducimos la condición de que -Vsat NO puede ser mayor que  
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación  
de Vo'
```

```
If VsatPos_2 < VsatNeg_2 Then
```

```
MsgBox "La tensión de saturación -Vsat no puede ser mayor que  
+Vsat", vbInformation + vbOKOnly, "AVISO"
```

```
IntegradorNoInversor.Valor_VsatPos_2.Text = ""  
IntegradorNoInversor.Valor_VsatNeg_2.Text = ""  
IntegradorNoInversor.Valor_Vo_2.Caption = ""  
IntegradorNoInversor.Valor_Saturacion_2.Caption = ""
```

```
Else
```

```
'Definimos las ecuaciones de los parametros a calcular y  
mostramos el resultado'
```

```
If IntegradorNoInversor.Sen_Cos.Text = "sen" Then
```

```
Integral_2 = -(a / b) * Cos(b * t_final_2)  
Vo_2 = (2 / (R_2 * C_2)) * Integral_2 + Vo_Inicial_2
```

```
ElseIf IntegradorNoInversor.Sen_Cos.Text = "cos" Then
```

```
Integral_2 = (a / b) * Sin(b * t_final_2)  
Vo_2 = (2 / (R_2 * C_2)) * Integral_2 + Vo_Inicial_2
```

```
Else
```

```
Integral_2 = ""  
Vo_2 = ""
```

```
End If
```

```
IntegradorNoInversor.Valor_Vo_2.Caption = Vo_2
```

```
'Analizamos si la tensión de salida está saturada'
```



```
If Vo_2 >= VsatPos_2 Then

    IntegradorNoInversor.Valor_Vo_2.Caption = VsatPos_2
    IntegradorNoInversor.Valor_Saturacion_2.Caption = "Salida
saturada"

ElseIf Vo_2 <= VsatNeg_2 Then

    IntegradorNoInversor.Valor_Vo_2.Caption = VsatNeg_2
    IntegradorNoInversor.Valor_Saturacion_2.Caption = "Salida
saturada"

Else

    IntegradorNoInversor.Valor_Saturacion_2.Caption = ""
    IntegradorNoInversor.Valor_Vo_2.Caption = Vo_2

End If

'Mostramos la representación Vi-t y Vo-t'

'Definimos los puntos necesarios para obtener nuestra gráfica'
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("q10") = R_2
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("q11") = R_2
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("r10") = C_2
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("r11") = C_2
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("p5") = 0
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("q5") = -3
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("p6") =
Vo_Inicial_2
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("q6") =
Vo_Inicial_2
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("p10") =
Vo_Inicial_2
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("p11") =
Vo_Inicial_2
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n4") =
VsatPos_2
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n5") =
VsatNeg_2
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag17") =
Empty
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah17") =
Empty
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag16") =
Empty
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah16") =
Empty
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ag19") =
Empty
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("ah19") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c17") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("c15") = Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("d15") = Empty

Dim t_max As Integer
Dim grados1 As Integer
Dim grados2 As Integer

t_max = (t_final_2 / 0.017453293) + 200
```



```
grados1 = 0
grados2 = 0

If IntegradorNoInversor.Sen_Cos.Text = "sen" Then

    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("s10") =
a
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("o10") =
b
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("s11") =
Empty
    ActiveWorkbook.Worksheets("Gráfica CosSen").Range("o11") =
Empty
    ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("a8:a728") = Empty
    ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("i8:i728") = Empty
    ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("j8:j728") = Empty

    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("s8:s728")
= Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("y8:y728")
= Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("z8:z728")
= Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-
t").Range("aa8:aa728") = Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-
t").Range("ab8:ab728") = Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("u5") =
Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v5") =
Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("w5") =
Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("x5") =
Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("aa5") =
Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("ab5") =
Empty
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("y5") = a
    ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("z5") = b

    'Desarrollamos el c[]o con el que la gráfica muestra si la
tensión de salida está saturada'
    Dim i1 As Integer

    For i1 = 8 To t_max

        Worksheets("Gráfica Vo-t").Cells(i1, 19).Value =
grados1
        Worksheets("Gráfica CosSen").Cells(i1, 1).Value =
grados1
        grados1 = grados1 + 1

        Worksheets("Gráfica Vo-t").Cells(i1, 27).Value =
Worksheets("Gráfica Vo-t").Cells(i1, 31).Value
        Worksheets("Gráfica CosSen").Cells(i1, 10).Value =
Worksheets("Gráfica CosSen").Cells(i1, 28).Value
```



```
        If Worksheets("Gráfica CosSen").Cells(i1, 10).Value >=
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n4").Value Then

            Worksheets("Gráfica CosSen").Cells(i1, 10).Value =
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n4").Value

            ElseIf Worksheets("Gráfica CosSen").Cells(i1,
10).Value <= ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("n5").Value Then

                Worksheets("Gráfica CosSen").Cells(i1, 10).Value =
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n5").Value

            Else

                Worksheets("Gráfica CosSen").Cells(i1, 10).Value =
Worksheets("Gráfica CosSen").Cells(i1, 28).Value

            End If

        Next i1

        ActiveWorkbook.Worksheets("Gráfica CosSen").Range("o6") =
Worksheets("Gráfica CosSen").Cells(8, 10).Value

        ElseIf IntegradorNoInversor.Sen_Cos.Text = "cos" Then

            ActiveWorkbook.Worksheets("Gráfica CosSen").Range("s11") =
a
            ActiveWorkbook.Worksheets("Gráfica CosSen").Range("o11") =
b
            ActiveWorkbook.Worksheets("Gráfica CosSen").Range("s10") =
Empty
            ActiveWorkbook.Worksheets("Gráfica CosSen").Range("o10") =
Empty
            ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("a8:a728") = Empty
            ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("i8:i728") = Empty
            ActiveWorkbook.Worksheets("Gráfica
CosSen").Range("j8:j728") = Empty

            ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("s8:s728")
= Empty
            ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("y8:y728")
= Empty
            ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("z8:z728")
= Empty
            ActiveWorkbook.Worksheets("Gráfica Vo-
t").Range("aa8:aa728") = Empty
            ActiveWorkbook.Worksheets("Gráfica Vo-
t").Range("ab8:ab728") = Empty
            ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("u5") =
Empty
            ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("v5") =
Empty
            ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("w5") =
Empty
            ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("x5") =
Empty
```



```
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("y5") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("z5") =
Empty
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("aa5") = a
ActiveWorkbook.Worksheets("Gráfica Vo-t").Range("ab5") = b

'Desarrollamos el cño con el que la gráfica muestra si la
tensión de salida está saturada'
Dim i2 As Integer

For i2 = 8 To t_max

Worksheets("Gráfica Vo-t").Cells(i2, 19).Value =
grados2
Worksheets("Gráfica CosSen").Cells(i2, 1).Value =
grados2
grados2 = grados2 + 1

Worksheets("Gráfica Vo-t").Cells(i2, 28).Value =
Worksheets("Gráfica Vo-t").Cells(i2, 32).Value
Worksheets("Gráfica CosSen").Cells(i2, 9).Value =
Worksheets("Gráfica CosSen").Cells(i2, 27).Value

If Worksheets("Gráfica CosSen").Cells(i2, 9).Value >=
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n4").Value Then

Worksheets("Gráfica CosSen").Cells(i2, 9) =
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n4").Value

ElseIf Worksheets("Gráfica CosSen").Cells(i2, 9).Value
<= ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n5").Value Then

Worksheets("Gráfica CosSen").Cells(i2, 9).Value =
ActiveWorkbook.Worksheets("Gráfica CosSen").Range("n5").Value

Else

Worksheets("Gráfica CosSen").Cells(i2, 9).Value =
Worksheets("Gráfica CosSen").Cells(i2, 27).Value

End If

Next i2

ActiveWorkbook.Worksheets("Gráfica CosSen").Range("o6") =
Worksheets("Gráfica CosSen").Cells(8, 9).Value

End If

End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub RestaurarValores1()
```




```
'Vaciamos los valores de cada uno de los parámetros'  
IntegradorNoInversor.Valor_R_1.Text = ""  
IntegradorNoInversor.Valor_C_1.Text = ""  
IntegradorNoInversor.Valor_Vi_1.Text = ""  
IntegradorNoInversor.n_1.Text = ""  
IntegradorNoInversor.Valor_VsatPos_1.Text = ""  
IntegradorNoInversor.Valor_VsatNeg_1.Text = ""  
IntegradorNoInversor.Valor_VoInicial_1.Text = ""  
IntegradorNoInversor.t_final_1.Text = ""  
IntegradorNoInversor.Valor_Vo_1.Caption = ""  
IntegradorNoInversor.Valor_Saturacion_1.Caption = ""
```

End Sub

```
Sub RestaurarValores2 ()
```

```
'Vaciamos los valores de cada uno de los parámetros'  
IntegradorNoInversor.Valor_R_2.Text = ""  
IntegradorNoInversor.Valor_C_2.Text = ""  
IntegradorNoInversor.a.Text = ""  
IntegradorNoInversor.b.Text = ""  
IntegradorNoInversor.Valor_VsatPos_2.Text = ""  
IntegradorNoInversor.Valor_VsatNeg_2.Text = ""  
IntegradorNoInversor.Valor_VoInicial_2.Text = ""  
IntegradorNoInversor.t_final_2.Text = ""  
IntegradorNoInversor.Valor_Vo_2.Caption = ""  
IntegradorNoInversor.Valor_Saturacion_2.Caption = ""
```

End Sub

7.2.3.14. Integrador restador (ModIntRes)

```
Sub MostrarIntegradorRestador ()
```

```
'Mostramos la pestaña de la aplicación del circuito integrador  
restador'  
IntegradorRestador.Show
```

End Sub

```
Sub MostrarTeoriaIntRes ()
```

```
'Mostramos la pestaña de la teoría del circuito integrador restador'  
TeoriaIntRes.Show
```

End Sub

```
Sub MostrarInformacion1 ()
```

```
'Mostramos la pestaña de la información del circuito integrador  
restador'  
InformacionInt2.Show
```

End Sub



```
Sub MostrarInformacion2 ()

'Mostramos la pestaña de la información del circuito integrador
restador'
InformacionInt2.Show

End Sub

Sub CalcularValores1 ()

'Declaremos las variables de los parámetros de un circuito integrador
restador'
Dim R_1 As Single
Dim C_1 As Single
Dim Vi1 As Single
Dim Vi2 As Single
Dim n1 As Single
Dim n2 As Single
Dim Integrall_1 As Single
Dim Integral2_1 As Single
Dim VsatPos_1 As Single
Dim VsatNeg_1 As Single
Dim Vo_Inicial_1 As Single
Dim t_final_1 As Single
Dim Vo_1 As Single

'Mediante esta declaración conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de la resistencia, el condensador y el valor de t final,
su valor no puede ser cero o negativo'
'La potencia de la tensión de entrada no puede ser menor o igual que -
1'
'En el caso de las tensiones, su valor si puede ser cero o negativo'
If IntegradorRestador.Valor_R_1.Text <= 0 Or
IntegradorRestador.Valor_C_1.Text <= 0 Or
IntegradorRestador.Valor_R_1.Text = Empty Or
IntegradorRestador.Valor_C_1.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas de la resistencia
y el condensador, y que no sea ni nulo ni negativo", vbExclamation +
vbOKOnly, "AVISO"

    IntegradorRestador.Valor_Vo_1.Caption = ""
    IntegradorRestador.Valor_Saturacion_1.Caption = ""

ElseIf IntegradorRestador.Valor_Vi1.Text = Empty Or
IntegradorRestador.Valor_Vi2.Text = Empty Or
IntegradorRestador.Valor_VsatPos_1.Text = Empty Or
IntegradorRestador.Valor_VsatNeg_1.Text = Empty Or
IntegradorRestador.Valor_VoInicial_1.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas del Voltaje",
vbExclamation + vbOKOnly, "AVISO"

    IntegradorRestador.Valor_Vo_1.Caption = ""
    IntegradorRestador.Valor_Saturacion_1.Caption = ""
```



```
ElseIf IntegradorRestador.n1.Text <= -1 Or IntegradorRestador.n1.Text  
= Empty Or IntegradorRestador.n2.Text <= -1 Or  
IntegradorRestador.n2.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en la casilla de la potencia del  
tiempo y que no sea nulo", vbExclamation + vbOKOnly, "AVISO"
```

```
IntegradorRestador.Valor_Vo_1.Caption = ""  
IntegradorRestador.Valor_Saturacion_1.Caption = ""
```

```
ElseIf IntegradorRestador.t_final_1.Text <= 0 Or  
IntegradorRestador.t_final_1.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en la casilla del tiempo referido  
a la condición final y que no sea nulo o negativo", vbExclamation +  
vbOKOnly, "AVISO"
```

```
IntegradorRestador.Valor_Vo_1.Caption = ""  
IntegradorRestador.Valor_Saturacion_1.Caption = ""
```

```
Else
```

```
'Asignamos las variables a los correspondientes parámetros del  
circuito integrador restador'
```

```
R_1 = IntegradorRestador.Valor_R_1.Text * 1000  
C_1 = IntegradorRestador.Valor_C_1.Text / 1000000  
Vi1 = IntegradorRestador.Valor_Vi1.Text  
Vi2 = IntegradorRestador.Valor_Vi2.Text  
n1 = IntegradorRestador.n1.Text  
n2 = IntegradorRestador.n2.Text  
VsatPos_1 = IntegradorRestador.Valor_VsatPos_1.Text  
VsatNeg_1 = IntegradorRestador.Valor_VsatNeg_1.Text  
Vo_Inicial_1 = IntegradorRestador.Valor_VoInicial_1.Text  
t_final_1 = IntegradorRestador.t_final_1.Text
```

```
'Introducimos la condición de que -Vsat NO puede ser mayor que  
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación  
de Vo'
```

```
If VsatPos_1 < VsatNeg_1 Then
```

```
MsgBox "La tensión de saturación -Vsat no puede ser mayor que  
+Vsat", vbInformation + vbOKOnly, "AVISO"
```

```
IntegradorRestador.Valor_VsatPos_1.Text = ""  
IntegradorRestador.Valor_VsatNeg_1.Text = ""  
IntegradorRestador.Valor_Vo_1.Caption = ""  
IntegradorRestador.Valor_Saturacion_1.Caption = ""
```

```
Else
```

```
'Definimos las ecuaciones de los parametros a calcular y  
mostramos el resultado'
```

```
Integrall_1 = ((1 / (n1 + 1)) * t_final_1 ^ (n1 + 1))  
Integral2_1 = ((1 / (n2 + 1)) * t_final_1 ^ (n2 + 1))  
Vo_1 = ((1 / (R_1 * C_1)) * (Vi2 * Integral2_1 - Vi1 *  
Integrall_1)) + Vo_Inicial_1
```

```
IntegradorRestador.Valor_Vo_1.Caption = Vo_1
```

```
'Analizamos si la tensión de salida está saturada'
```

```
If Vo_1 >= VsatPos_1 Then
```



```
IntegradorRestador.Valor_Vo_1.Caption = VsatPos_1
IntegradorRestador.Valor_Saturacion_1.Caption = "Salida
saturada"

ElseIf Vo_1 <= VsatNeg_1 Then

IntegradorRestador.Valor_Vo_1.Caption = VsatNeg_1
IntegradorRestador.Valor_Saturacion_1.Caption = "Salida
saturada"

Else

IntegradorRestador.Valor_Saturacion_1.Caption = ""
IntegradorRestador.Valor_Vo_1.Caption = Vo_1

End If

End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub CalcularValores2()

'Declaremos las variables de los parámetros de un circuito integrador
inversor'
Dim R_2 As Single
Dim C_2 As Single
Dim a1 As Single
Dim a2 As Single
Dim Sen_Cos1 As String
Dim Sen_Cos2 As String
Dim b1 As Single
Dim b2 As Single
Dim Integrall_2 As Single
Dim Integral2_2 As Single
Dim VsatPos_2 As Single
Dim VsatNeg_2 As Single
Dim Vo_Inicial_2 As Single
Dim t_final_2 As Single
Dim Vo_2 As Single

'Mediante esta declaración conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de la resistencia, el condensador y el valor de t final,
su valor no puede ser cero o negativo'
'En el caso de las tensiones, su valor si puede ser cero o negativo'
If IntegradorRestador.Valor_R_2.Text <= 0 Or
IntegradorRestador.Valor_C_2.Text <= 0 Or
IntegradorRestador.Valor_R_2.Text = Empty Or
IntegradorRestador.Valor_C_2.Text = Empty Then
```



```
MsgBox "Debe introducir un valor en las casillas de la resistencia  
y el condensador, y que no sea ni nulo ni negativo", vbExclamation +  
vbOKOnly, "AVISO"
```

```
IntegradorRestador.Valor_Vo_2.Caption = ""  
IntegradorRestador.Valor_Saturacion_2.Caption = ""
```

```
ElseIf IntegradorRestador.a1.Text = Empty Or  
IntegradorRestador.a2.Text = Empty Or IntegradorRestador.b1.Text =  
Empty Or IntegradorRestador.b2.Text = Empty Or  
IntegradorRestador.Valor_VsatPos_2.Text = Empty Or  
IntegradorRestador.Valor_VsatNeg_2.Text = Empty Or  
IntegradorRestador.Valor_VoInicial_2.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas del Voltaje",  
vbExclamation + vbOKOnly, "AVISO"
```

```
IntegradorRestador.Valor_Vo_2.Caption = ""  
IntegradorRestador.Valor_Saturacion_2.Caption = ""
```

```
ElseIf IntegradorRestador.t_final_2.Text <= 0 Or  
IntegradorRestador.t_final_2.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en la casilla del tiempo referido  
a la condición final y que no sea nulo o negativo", vbExclamation +  
vbOKOnly, "AVISO"
```

```
IntegradorRestador.Valor_Vo_2.Caption = ""  
IntegradorRestador.Valor_Saturacion_2.Caption = ""
```

Else

```
'Asignamos las variables a los correspondientes parámetros del  
circuito integrador inversor'
```

```
R_2 = IntegradorRestador.Valor_R_2.Text * 1000  
C_2 = IntegradorRestador.Valor_C_2.Text / 1000000  
a1 = IntegradorRestador.a1.Text  
a2 = IntegradorRestador.a2.Text  
b1 = IntegradorRestador.b1.Text  
b2 = IntegradorRestador.b2.Text  
Sen_Cos1 = IntegradorRestador.Sen_Cos1.Text  
Sen_Cos2 = IntegradorRestador.Sen_Cos2.Text  
VsatPos_2 = IntegradorRestador.Valor_VsatPos_2.Text  
VsatNeg_2 = IntegradorRestador.Valor_VsatNeg_2.Text  
Vo_Inicial_2 = IntegradorRestador.Valor_VoInicial_2.Text  
t_final_2 = IntegradorRestador.t_final_2.Text
```

```
'Introducimos la condición de que -Vsat NO puede ser mayor que  
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación  
de Vo'
```

```
If VsatPos_2 < VsatNeg_2 Then
```

```
MsgBox "La tensión de saturación -Vsat no puede ser mayor que  
+Vsat", vbInformation + vbOKOnly, "AVISO"
```

```
IntegradorRestador.Valor_VsatPos_2.Text = ""  
IntegradorRestador.Valor_VsatNeg_2.Text = ""  
IntegradorRestador.Valor_Vo_2.Caption = ""  
IntegradorRestador.Valor_Saturacion_2.Caption = ""
```

Else



```
'Definimos las ecuaciones de los parametros a calcular y
mostramos el resultado'
If IntegradorRestador.Sen_Cos1.Text = "sen" And
IntegradorRestador.Sen_Cos2.Text = "sen" Then

    Integrall_2 = -(a1 / b1) * Cos(b1 * t_final_2)
    Integral2_2 = -(a2 / b2) * Cos(b2 * t_final_2)
    Vo_2 = (1 / (R_2 * C_2)) * (Integral2_2 - Integrall_2) +
Vo_Inicial_2

    ElseIf IntegradorRestador.Sen_Cos1.Text = "sen" And
IntegradorRestador.Sen_Cos2.Text = "cos" Then

    Integrall_2 = -(a1 / b1) * Cos(b1 * t_final_2)
    Integral2_2 = (a2 / b2) * Sin(b2 * t_final_2)
    Vo_2 = (1 / (R_2 * C_2)) * (Integral2_2 - Integrall_2) +
Vo_Inicial_2

    ElseIf IntegradorRestador.Sen_Cos1.Text = "cos" And
IntegradorRestador.Sen_Cos2.Text = "sen" Then

    Integrall_2 = (a1 / b1) * Sin(b1 * t_final_2)
    Integral2_2 = -(a2 / b2) * Cos(b2 * t_final_2)
    Vo_2 = (1 / (R_2 * C_2)) * (Integral2_2 - Integrall_2) +
Vo_Inicial_2

    ElseIf IntegradorRestador.Sen_Cos1.Text = "cos" And
IntegradorRestador.Sen_Cos2.Text = "cos" Then

    Integrall_2 = (a1 / b1) * Sin(b1 * t_final_2)
    Integral2_2 = (a2 / b2) * Sin(b2 * t_final_2)
    Vo_2 = (1 / (R_2 * C_2)) * (Integral2_2 - Integrall_2) +
Vo_Inicial_2

Else

    Integrall_2 = ""
    Integral2_2 = ""
    Vo_2 = ""

End If

IntegradorRestador.Valor_Vo_2.Caption = Vo_2

'Analizamos si la tensión de salida está saturada'
If Vo_2 >= VsatPos_2 Then

    IntegradorRestador.Valor_Vo_2.Caption = VsatPos_2
    IntegradorRestador.Valor_Saturacion_2.Caption = "Salida
saturada"

ElseIf Vo_2 <= VsatNeg_2 Then

    IntegradorRestador.Valor_Vo_2.Caption = VsatNeg_2
    IntegradorRestador.Valor_Saturacion_2.Caption = "Salida
saturada"

Else

    IntegradorRestador.Valor_Saturacion_2.Caption = ""
```



```
IntegradorRestador.Valor_Vo_2.Caption = Vo_2

End If

End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub RestaurarValores1 ()

'Vaciamos los valores de cada uno de los parámetros'
IntegradorRestador.Valor_R_1.Text = ""
IntegradorRestador.Valor_C_1.Text = ""
IntegradorRestador.Valor_Vi1.Text = ""
IntegradorRestador.Valor_Vi2.Text = ""
IntegradorRestador.n1.Text = ""
IntegradorRestador.n2.Text = ""
IntegradorRestador.Valor_VsatPos_1.Text = ""
IntegradorRestador.Valor_VsatNeg_1.Text = ""
IntegradorRestador.Valor_VoInicial_1.Text = ""
IntegradorRestador.t_final_1.Text = ""
IntegradorRestador.Valor_Vo_1.Caption = ""
IntegradorRestador.Valor_Saturacion_1.Caption = ""

End Sub

Sub RestaurarValores2 ()

'Vaciamos los valores de cada uno de los parámetros'
IntegradorRestador.Valor_R_2.Text = ""
IntegradorRestador.Valor_C_2.Text = ""
IntegradorRestador.a1.Text = ""
IntegradorRestador.a2.Text = ""
IntegradorRestador.b1.Text = ""
IntegradorRestador.b2.Text = ""
IntegradorRestador.Valor_VsatPos_2.Text = ""
IntegradorRestador.Valor_VsatNeg_2.Text = ""
IntegradorRestador.Valor_VoInicial_2.Text = ""
IntegradorRestador.t_final_2.Text = ""
IntegradorRestador.Valor_Vo_2.Caption = ""
IntegradorRestador.Valor_Saturacion_2.Caption = ""

End Sub
```

7.2.3.15. Integrador sumador (ModIntSum)

```
Sub MostrarIntegradorSumador ()
```



```
'Mostramos la pestaña de la aplicación del circuito integrador
sumador'
IntegradorSumador.Show

End Sub

Sub MostrarTeoriaIntSum()

'Mostramos la pestaña de la teoría del circuito integrador sumador'
TeoriaIntSum.Show

End Sub

Sub MostrarInformacion1()

'Mostramos la pestaña de la información del circuito integrador
sumador'
InformacionInt2.Show

End Sub

Sub MostrarInformacion2()

'Mostramos la pestaña de la información del circuito integrador
sumador'
InformacionInt2.Show

End Sub

Sub CalcularValores1()

'Declaremos las variables de los parámetros de un circuito integrador
sumador'
Dim R1_1 As Single
Dim R2_1 As Single
Dim R3_1 As Single
Dim C_1 As Single
Dim Vi1 As Single
Dim Vi2 As Single
Dim Vi3 As Single
Dim n1 As Single
Dim n2 As Single
Dim n3 As Single
Dim Integrall_1 As Single
Dim Integral2_1 As Single
Dim Integral3_1 As Single
Dim VsatPos_1 As Single
Dim VsatNeg_1 As Single
Dim Vo_Inicial_1 As Single
Dim t_final_1 As Single
Dim R4_1 As Single
Dim Vo_1 As Single

'Mediante esta declaración conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de la resistencia, el condensador y el valor de t final,
su valor no puede ser cero o negativo'
```




```
'La potencia de la tensión de entrada no puede ser menor o igual que -  
1'  
'En el caso de las tensiones, su valor si puede ser cero o negativo'  
If IntegradorSumador.Valor_R1_1.Text <= 0 Or  
IntegradorSumador.Valor_R2_1.Text <= 0 Or  
IntegradorSumador.Valor_R3_1.Text <= 0 Or  
IntegradorSumador.Valor_C_1.Text <= 0 Or  
IntegradorSumador.Valor_R1_1.Text = Empty Or  
IntegradorSumador.Valor_R2_1.Text = Empty Or  
IntegradorSumador.Valor_R3_1.Text = Empty Or  
IntegradorSumador.Valor_C_1.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas de la resistencia  
y el condensador, y que no sea ni nulo ni negativo", vbExclamation +  
vbOKOnly, "AVISO"
```

```
IntegradorSumador.Valor_R4_1.Caption = ""  
IntegradorSumador.Valor_Vo_1.Caption = ""  
IntegradorSumador.Valor_Saturacion_1.Caption = ""
```

```
ElseIf IntegradorSumador.Valor_Vi1.Text = Empty Or  
IntegradorSumador.Valor_Vi2.Text = Empty Or  
IntegradorSumador.Valor_Vi3.Text = Empty Or  
IntegradorSumador.Valor_VsatPos_1.Text = Empty Or  
IntegradorSumador.Valor_VsatNeg_1.Text = Empty Or  
IntegradorSumador.Valor_VoInicial_1.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas del Voltaje",  
vbExclamation + vbOKOnly, "AVISO"
```

```
IntegradorSumador.Valor_R4_1.Caption = ""  
IntegradorSumador.Valor_Vo_1.Caption = ""  
IntegradorSumador.Valor_Saturacion_1.Caption = ""
```

```
ElseIf IntegradorSumador.n1.Text <= -1 Or IntegradorSumador.n1.Text =  
Empty Or IntegradorSumador.n2.Text <= -1 Or IntegradorSumador.n2.Text  
= Empty Or IntegradorSumador.n3.Text <= -1 Or  
IntegradorSumador.n3.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en la casilla de la potencia del  
tiempo y que no sea nulo", vbExclamation + vbOKOnly, "AVISO"
```

```
IntegradorSumador.Valor_R4_1.Caption = ""  
IntegradorSumador.Valor_Vo_1.Caption = ""  
IntegradorSumador.Valor_Saturacion_1.Caption = ""
```

```
ElseIf IntegradorSumador.t_final_1.Text <= 0 Or  
IntegradorSumador.t_final_1.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en la casilla del tiempo referido  
a la condición final y que no sea nulo o negativo", vbExclamation +  
vbOKOnly, "AVISO"
```

```
IntegradorSumador.Valor_R4_1.Caption = ""  
IntegradorSumador.Valor_Vo_1.Caption = ""  
IntegradorSumador.Valor_Saturacion_1.Caption = ""
```

```
Else
```

```
'Asignamos las variables a los correspondientes parámetros del  
circuito integrador inversor'
```



```
R1_1 = IntegradorSumador.Valor_R1_1.Text * 1000
R2_1 = IntegradorSumador.Valor_R2_1.Text * 1000
R3_1 = IntegradorSumador.Valor_R3_1.Text * 1000
C_1 = IntegradorSumador.Valor_C_1.Text / 1000000
Vi1 = IntegradorSumador.Valor_Vi1.Text
Vi2 = IntegradorSumador.Valor_Vi2.Text
Vi3 = IntegradorSumador.Valor_Vi3.Text
n1 = IntegradorSumador.n1.Text
n2 = IntegradorSumador.n2.Text
n3 = IntegradorSumador.n3.Text
VsatPos_1 = IntegradorSumador.Valor_VsatPos_1.Text
VsatNeg_1 = IntegradorSumador.Valor_VsatNeg_1.Text
Vo_Inicial_1 = IntegradorSumador.Valor_VoInicial_1.Text
t_final_1 = IntegradorSumador.t_final_1.Text

'Introducimos la condición de que -Vsat NO puede ser mayor que
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
de Vo'
If VsatPos_1 < VsatNeg_1 Then

    MsgBox "La tensión de saturación -Vsat no puede ser mayor que
+Vsat", vbInformation + vbOKOnly, "AVISO"

    IntegradorSumador.Valor_VsatPos_1.Text = ""
    IntegradorSumador.Valor_VsatNeg_1.Text = ""
    IntegradorSumador.Valor_Vo_1.Caption = ""
    IntegradorSumador.Valor_Saturacion_1.Caption = ""

Else

    'Definimos las ecuaciones de los parametros a calcular y
mostramos el resultado'
    R4_1 = ((R1_1 * R2_1 * R3_1) / ((R2_1 + R1_1) * R3_1 + R1_1 *
R2_1)) / 1000
    Integrall_1 = ((1 / (n1 + 1)) * t_final_1 ^ (n1 + 1))
    Integral2_1 = ((1 / (n2 + 1)) * t_final_1 ^ (n2 + 1))
    Integral3_1 = ((1 / (n3 + 1)) * t_final_1 ^ (n3 + 1))

    Vo_1 = (-1 / C_1) * ((Vi1 / R1_1) * Integrall_1 + (Vi2 / R2_1)
* Integral2_1 + (Vi3 / R3_1) * Integral3_1) + Vo_Inicial_1

    IntegradorSumador.Valor_R4_1.Caption = R4_1
    IntegradorSumador.Valor_Vo_1.Caption = Vo_1

    'Analizamos si la tensión de salida está saturada'
    If Vo_1 >= VsatPos_1 Then

        IntegradorSumador.Valor_Vo_1.Caption = VsatPos_1
        IntegradorSumador.Valor_Saturacion_1.Caption = "Salida
saturada"

    ElseIf Vo_1 <= VsatNeg_1 Then

        IntegradorSumador.Valor_Vo_1.Caption = VsatNeg_1
        IntegradorSumador.Valor_Saturacion_1.Caption = "Salida
saturada"

    Else

        IntegradorSumador.Valor_Saturacion_1.Caption = ""
        IntegradorSumador.Valor_Vo_1.Caption = Vo_1
```



```
End If

End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub CalcularValores2 ()

'Declaremos las variables de los parámetros de un circuito integrador
inversor'
Dim R1_2 As Single
Dim R2_2 As Single
Dim R3_2 As Single
Dim C_2 As Single
Dim a1 As Single
Dim a2 As Single
Dim a3 As Single
Dim Sen_Cos1 As String
Dim Sen_Cos2 As String
Dim Sen_Cos3 As String
Dim b1 As Single
Dim b2 As Single
Dim b3 As Single
Dim Integrall_2 As Single
Dim Integral2_2 As Single
Dim Integral3_2 As Single
Dim VsatPos_2 As Single
Dim VsatNeg_2 As Single
Dim Vo_Inicial_2 As Single
Dim t_final_2 As Single
Dim Vo_2 As Single

'Mediante esta declaración conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de la resistencia, el condensador y el valor de t final,
su valor no puede ser cero o negativo'
'En el caso de las tensiones, su valor si puede ser cero o negativo'
If IntegradorSumador.Valor_R1_2.Text <= 0 Or
IntegradorSumador.Valor_R2_2.Text <= 0 Or
IntegradorSumador.Valor_R3_2.Text <= 0 Or
IntegradorSumador.Valor_C_2.Text <= 0 Or
IntegradorSumador.Valor_R1_2.Text = Empty Or
IntegradorSumador.Valor_R2_2.Text = Empty Or
IntegradorSumador.Valor_R3_2.Text = Empty Or
IntegradorSumador.Valor_C_2.Text = Empty Then

MsgBox "Debe introducir un valor en las casillas de la resistencia
y el condensador, y que no sea ni nulo ni negativo", vbExclamation +
vbOKOnly, "AVISO"

IntegradorSumador.Valor_R4_2.Caption = ""
IntegradorSumador.Valor_Vo_2.Caption = ""
```



```
IntegradorSumador.Valor_Saturacion_2.Caption = ""

ElseIf IntegradorSumador.a1.Text = Empty Or IntegradorSumador.a2.Text
= Empty Or IntegradorSumador.a3.Text = Empty Or
IntegradorSumador.b1.Text = Empty Or IntegradorSumador.b2.Text = Empty
Or IntegradorSumador.b3.Text = Empty Or
IntegradorSumador.Valor_VsatPos_2.Text = Empty Or
IntegradorSumador.Valor_VsatNeg_2.Text = Empty Or
IntegradorSumador.Valor_VoInicial_2.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas del Voltaje",
vbExclamation + vbOKOnly, "AVISO"

    IntegradorSumador.Valor_R4_2.Caption = ""
    IntegradorSumador.Valor_Vo_2.Caption = ""
    IntegradorSumador.Valor_Saturacion_2.Caption = ""

ElseIf IntegradorSumador.t_final_2.Text <= 0 Or
IntegradorSumador.t_final_2.Text = Empty Then

    MsgBox "Debe introducir un valor en la casilla del tiempo referido
a la condición final y que no sea nulo o negativo", vbExclamation +
vbOKOnly, "AVISO"

    IntegradorSumador.Valor_R4_2.Caption = ""
    IntegradorSumador.Valor_Vo_2.Caption = ""
    IntegradorSumador.Valor_Saturacion_2.Caption = ""

Else

    'Asignamos las variables a los correspondientes parámetros del
circuito integrador inversor'
    R1_2 = IntegradorSumador.Valor_R1_2.Text * 1000
    R2_2 = IntegradorSumador.Valor_R1_2.Text * 1000
    R3_2 = IntegradorSumador.Valor_R1_2.Text * 1000
    C_2 = IntegradorSumador.Valor_C_2.Text / 1000000
    a1 = IntegradorSumador.a1.Text
    a2 = IntegradorSumador.a2.Text
    a3 = IntegradorSumador.a3.Text
    b1 = IntegradorSumador.b1.Text
    b2 = IntegradorSumador.b2.Text
    b3 = IntegradorSumador.b3.Text
    Sen_Cos1 = IntegradorSumador.Sen_Cos1.Text
    Sen_Cos2 = IntegradorSumador.Sen_Cos2.Text
    Sen_Cos3 = IntegradorSumador.Sen_Cos3.Text
    VsatPos_2 = IntegradorSumador.Valor_VsatPos_2.Text
    VsatNeg_2 = IntegradorSumador.Valor_VsatNeg_2.Text
    Vo_Inicial_2 = IntegradorSumador.Valor_VoInicial_2.Text
    t_final_2 = IntegradorSumador.t_final_2.Text

    'Introducimos la condición de que -Vsat NO puede ser mayor que
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
de Vo'
    If VsatPos_2 < VsatNeg_2 Then

        MsgBox "La tensión de saturación -Vsat no puede ser mayor que
+Vsat", vbInformation + vbOKOnly, "AVISO"

        IntegradorSumador.Valor_VsatPos_2.Text = ""
        IntegradorSumador.Valor_VsatNeg_2.Text = ""
        IntegradorSumador.Valor_Vo_2.Caption = ""
```



```
IntegradorSumador.Valor_Saturacion_2.Caption = ""

Else

    'Definimos las ecuaciones de los parametros a calcular y
    mostramos el resultado'
    R4_2 = ((R1_2 * R2_2 * R3_2) / ((R2_2 + R1_2) * R3_2 + R1_2 *
R2_2)) / 1000

    If IntegradorSumador.Sen_Cos1.Text = "sen" And
IntegradorSumador.Sen_Cos2.Text = "sen" And
IntegradorSumador.Sen_Cos3.Text = "sen" Then

        Integral1_2 = -(a1 / b1) * Cos(b1 * t_final_2)
        Integral2_2 = -(a2 / b2) * Cos(b2 * t_final_2)
        Integral3_2 = -(a3 / b3) * Cos(b3 * t_final_2)
        Vo_2 = (-1 / C_2) * (Integral1_2 / R1_2 + Integral2_2 /
R2_2 + Integral3_2 / R3_2) + Vo_Inicial_2

    ElseIf IntegradorSumador.Sen_Cos1.Text = "sen" And
IntegradorSumador.Sen_Cos2.Text = "cos" And
IntegradorSumador.Sen_Cos3.Text = "sen" Then

        Integral1_2 = -(a1 / b1) * Cos(b1 * t_final_2)
        Integral2_2 = (a2 / b2) * Sin(b2 * t_final_2)
        Integral3_2 = -(a3 / b3) * Cos(b3 * t_final_2)
        Vo_2 = (-1 / C_2) * (Integral1_2 / R1_2 + Integral2_2 /
R2_2 + Integral3_2 / R3_2) + Vo_Inicial_2

    ElseIf IntegradorSumador.Sen_Cos1.Text = "sen" And
IntegradorSumador.Sen_Cos2.Text = "sen" And
IntegradorSumador.Sen_Cos3.Text = "cos" Then

        Integral1_2 = -(a1 / b1) * Cos(b1 * t_final_2)
        Integral2_2 = -(a2 / b2) * Cos(b2 * t_final_2)
        Integral3_2 = (a3 / b3) * Sin(b3 * t_final_2)
        Vo_2 = (-1 / C_2) * (Integral1_2 / R1_2 + Integral2_2 /
R2_2 + Integral3_2 / R3_2) + Vo_Inicial_2

    ElseIf IntegradorSumador.Sen_Cos1.Text = "sen" And
IntegradorSumador.Sen_Cos2.Text = "cos" And
IntegradorSumador.Sen_Cos3.Text = "cos" Then

        Integral1_2 = -(a1 / b1) * Cos(b1 * t_final_2)
        Integral2_2 = (a2 / b2) * Sin(b2 * t_final_2)
        Integral3_2 = (a3 / b3) * Sin(b3 * t_final_2)
        Vo_2 = (-1 / C_2) * (Integral1_2 / R1_2 + Integral2_2 /
R2_2 + Integral3_2 / R3_2) + Vo_Inicial_2

    ElseIf IntegradorSumador.Sen_Cos1.Text = "cos" And
IntegradorSumador.Sen_Cos2.Text = "cos" And
IntegradorSumador.Sen_Cos3.Text = "cos" Then

        Integral1_2 = (a1 / b1) * Sin(b1 * t_final_2)
        Integral2_2 = (a2 / b2) * Sin(b2 * t_final_2)
        Integral3_2 = (a3 / b3) * Sin(b3 * t_final_2)
        Vo_2 = (-1 / C_2) * (Integral1_2 / R1_2 + Integral2_2 /
R2_2 + Integral3_2 / R3_2) + Vo_Inicial_2
```



```
ElseIf IntegradorSumador.Sen_Cos1.Text = "cos" And
IntegradorSumador.Sen_Cos2.Text = "sen" And
IntegradorSumador.Sen_Cos3.Text = "cos" Then

    Integral1_2 = (a1 / b1) * Sin(b1 * t_final_2)
    Integral2_2 = -(a2 / b2) * Cos(b2 * t_final_2)
    Integral3_2 = (a3 / b3) * Sin(b3 * t_final_2)
    Vo_2 = (-1 / C_2) * (Integral1_2 / R1_2 + Integral2_2 /
R2_2 + Integral3_2 / R3_2) + Vo_Inicial_2

ElseIf IntegradorSumador.Sen_Cos1.Text = "cos" And
IntegradorSumador.Sen_Cos2.Text = "cos" And
IntegradorSumador.Sen_Cos3.Text = "sen" Then

    Integral1_2 = (a1 / b1) * Sin(b1 * t_final_2)
    Integral2_2 = (a2 / b2) * Sin(b2 * t_final_2)
    Integral3_2 = -(a3 / b3) * Cos(b3 * t_final_2)
    Vo_2 = (-1 / C_2) * (Integral1_2 / R1_2 + Integral2_2 /
R2_2 + Integral3_2 / R3_2) + Vo_Inicial_2

ElseIf IntegradorSumador.Sen_Cos1.Text = "cos" And
IntegradorSumador.Sen_Cos2.Text = "sen" And
IntegradorSumador.Sen_Cos3.Text = "sen" Then

    Integral1_2 = (a1 / b1) * Sin(b1 * t_final_2)
    Integral2_2 = -(a2 / b2) * Cos(b2 * t_final_2)
    Integral3_2 = -(a3 / b3) * Cos(b3 * t_final_2)
    Vo_2 = (-1 / C_2) * (Integral1_2 / R1_2 + Integral2_2 /
R2_2 + Integral3_2 / R3_2) + Vo_Inicial_2

Else

    Integral1_2 = ""
    Integral2_2 = ""
    Vo_2 = ""

End If

IntegradorSumador.Valor_Vo_2.Caption = Vo_2
IntegradorSumador.Valor_R4_2.Caption = R4_2

'Analizamos si la tensión de salida está saturada'
If Vo_2 >= VsatPos_2 Then

    IntegradorSumador.Valor_Vo_2.Caption = VsatPos_2
    IntegradorSumador.Valor_Saturacion_2.Caption = "Salida
saturada"

ElseIf Vo_2 <= VsatNeg_2 Then

    IntegradorSumador.Valor_Vo_2.Caption = VsatNeg_2
    IntegradorSumador.Valor_Saturacion_2.Caption = "Salida
saturada"

Else

    IntegradorSumador.Valor_Saturacion_2.Caption = ""
    IntegradorSumador.Valor_Vo_2.Caption = Vo_2

End If
```



```
End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub RestaurarValores1 ()

'Vaciamos los valores de cada uno de los parámetros'
IntegradorSumador.Valor_R1_1.Text = ""
IntegradorSumador.Valor_R2_1.Text = ""
IntegradorSumador.Valor_R3_1.Text = ""
IntegradorSumador.Valor_C_1.Text = ""
IntegradorSumador.Valor_Vi1.Text = ""
IntegradorSumador.Valor_Vi2.Text = ""
IntegradorSumador.Valor_Vi3.Text = ""
IntegradorSumador.n1.Text = ""
IntegradorSumador.n2.Text = ""
IntegradorSumador.n3.Text = ""
IntegradorSumador.Valor_VsatPos_1.Text = ""
IntegradorSumador.Valor_VsatNeg_1.Text = ""
IntegradorSumador.Valor_VoInicial_1.Text = ""
IntegradorSumador.t_final_1.Text = ""
IntegradorSumador.Valor_R4_1.Caption = ""
IntegradorSumador.Valor_Vo_1.Caption = ""
IntegradorSumador.Valor_Saturacion_1.Caption = ""

End Sub

Sub RestaurarValores2 ()

'Vaciamos los valores de cada uno de los parámetros'
IntegradorSumador.Valor_R1_2.Text = ""
IntegradorSumador.Valor_R2_2.Text = ""
IntegradorSumador.Valor_R3_2.Text = ""
IntegradorSumador.Valor_C_2.Text = ""
IntegradorSumador.a1.Text = ""
IntegradorSumador.a2.Text = ""
IntegradorSumador.a3.Text = ""
IntegradorSumador.b1.Text = ""
IntegradorSumador.b2.Text = ""
IntegradorSumador.b3.Text = ""
IntegradorSumador.Valor_VsatPos_2.Text = ""
IntegradorSumador.Valor_VsatNeg_2.Text = ""
IntegradorSumador.Valor_VoInicial_2.Text = ""
IntegradorSumador.t_final_2.Text = ""
IntegradorSumador.Valor_R4_2.Caption = ""
IntegradorSumador.Valor_Vo_2.Caption = ""
IntegradorSumador.Valor_Saturacion_2.Caption = ""

End Sub
```

7.2.3.16. Circuito inversor (ModInversor)



```
Sub MostrarInversor ()

'Mostramos la pestaña de la aplicación del circuito inversor'
Inversor.Show

End Sub

Sub MostrarTeoriaInversor ()

'Mostramos la pestaña de la teoría de un circuito inversor'
TeoriaInv.Show

End Sub

Sub MostrarInformacion ()

'Mostramos la pestaña de la información del circuito inversor'
Informacion.Show

End Sub

Sub CalcularValores ()

'Declaramos las variables de los parámetros de un circuito inversor'
Dim R1 As Single
Dim R2 As Single
Dim Vi As Single
Dim VsatPos As Single
Dim VsatNeg As Single
Dim Vi_Pos As Single
Dim Vi_Neg As Single
Dim R3 As Single
Dim Ganancia As Single
Dim Vo As Single

'Mediante esta declaración conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de las resistencias, su valor no puede ser cero o
negativo'
'En el caso de las tensiones, su valor si puede ser cero o negativo'
If Inversor.Valor_R1.Text <= 0 Or Inversor.Valor_R2.Text <= 0 Or
Inversor.Valor_R1.Text = Empty Or Inversor.Valor_R2.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas de las
resistencias y que no sea ni nulo ni negativo", vbExclamation +
vbOKOnly, "AVISO"

    Inversor.Valor_R3.Caption = ""
    Inversor.Valor_Ganancia.Caption = ""
    Inversor.Valor_Vo.Caption = ""
    Inversor.Valor_Saturacion.Caption = ""

ElseIf Inversor.Valor_Vi.Text = Empty Or Inversor.Valor_VsatPos.Text =
Empty Or Inversor.Valor_VsatNeg.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas del Voltaje",
vbExclamation + vbOKOnly, "AVISO"
```




```
Inversor.Valor_R3.Caption = ""
Inversor.Valor_Ganancia.Caption = ""
Inversor.Valor_Vo.Caption = ""
Inversor.Valor_Saturacion.Caption = ""

Else

    'Asignamos las variables a los correspondientes parámetros del
    circuito inversor'
    R1 = Inversor.Valor_R1.Text * 1000
    R2 = Inversor.Valor_R2.Text * 1000
    Vi = Inversor.Valor_Vi.Text
    VsatPos = Inversor.Valor_VsatPos.Text
    VsatNeg = Inversor.Valor_VsatNeg.Text

    'Introducimos la condición de que -Vsat NO puede ser mayor que
    +Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
    de Vo'
    If VsatPos < VsatNeg Then

        MsgBox "La tensión de saturación -Vsat no puede ser mayor que
        +Vsat", vbInformation + vbOKOnly, "AVISO"

        Inversor.Valor_VsatPos.Text = ""
        Inversor.Valor_VsatNeg.Text = ""
        Inversor.Valor_R3.Caption = ""
        Inversor.Valor_Ganancia.Caption = ""
        Inversor.Valor_Vo.Caption = ""
        Inversor.Valor_Saturacion.Caption = ""

    Else

        'Definimos las ecuaciones de los parametros a calcular y
        mostramos el resultado'
        R3 = ((R2 * R1) / (R2 + R1)) / 1000
        Ganancia = (-(R2 / R1))
        Vo = (Ganancia * Vi)

        Inversor.Valor_R3.Caption = R3
        Inversor.Valor_Ganancia.Caption = Ganancia
        Inversor.Valor_Vo.Caption = Vo

        'Analizamos si la tensión de salida está saturada'
        If Vo >= VsatPos Then

            Inversor.Valor_Vo.Caption = VsatPos
            Inversor.Valor_Saturacion.Caption = "Salida saturada"

        ElseIf Vo <= VsatNeg Then

            Inversor.Valor_Vo.Caption = VsatNeg
            Inversor.Valor_Saturacion.Caption = "Salida saturada"

        Else

            Inversor.Valor_Saturacion.Caption = ""
            Inversor.Valor_Vo.Caption = Vo

        End If

        'Mostramos la representación Vo-Vi'
```



```
'Calculamos los valores de Vi para los que Vo satura, ya que
será necesarios para obtener la gráfica'
Vi_Neg = VsatPos / Ganancia
Vi_Pos = VsatNeg / Ganancia

'Definimos los puntos necesarios para obtener nuestra gráfica'
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c5") =
Ganancia
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c7") =
Vi_Neg
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c8") =
Vi_Pos
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c10") =
VsatPos
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d10") =
VsatPos
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c11") =
VsatNeg
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d11") =
VsatNeg

End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub RestaurarValores ()

'Vaciamos los valores de cada uno de los parámetros'
Inversor.Valor_R1.Text = ""
Inversor.Valor_R2.Text = ""
Inversor.Valor_Vi.Text = ""
Inversor.Valor_VsatPos.Text = ""
Inversor.Valor_VsatNeg.Text = ""
Inversor.Valor_R3.Caption = ""
Inversor.Valor_Ganancia.Caption = ""
Inversor.Valor_Vo.Caption = ""
Inversor.Valor_Saturacion.Caption = ""

End Sub
```

7.2.3.17. Circuito no inversor (ModNoInversor)

```
Sub MostrarNoInversor ()

'Mostramos la pestaña de la aplicación del circuito no inversor'
NoInversor.Show

End Sub
```



```
Sub MostrarTeoriaNoInversor ()

'Mostramos la pestaña de la teoría de un circuito no inversor'
TeoriaNoInv.Show

End Sub

Sub MostrarInformacion ()

'Mostramos la pestaña de la información del circuito no inversor'
Informacion.Show

End Sub

Sub CalcularValores ()

'Declaramos las variables de los parámetros de un circuito no
inversor'
Dim R1 As Single
Dim R2 As Single
Dim Vi As Single
Dim VsatPos As Single
Dim VsatNeg As Single
Dim Vi_Pos As Single
Dim Vi_Neg As Single
Dim R3 As Single
Dim Ganancia As Single
Dim Vo As Single

'Mediante esta declaración conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de las resistencias, su valor no puede ser cero o
negativo'
'En el caso de las tensiones, su valor si puede ser cero o negativo'
If NoInversor.Valor_R1.Text <= 0 Or NoInversor.Valor_R2.Text <= 0 Or
NoInversor.Valor_R1.Text = Empty Or NoInversor.Valor_R2.Text = Empty
Then

    MsgBox "Debe introducir un valor en las casillas de las
resistencias y que no sea ni nulo ni negativo", vbExclamation +
vbOKOnly, "AVISO"

    NoInversor.Valor_R3.Caption = ""
    NoInversor.Valor_Ganancia.Caption = ""
    NoInversor.Valor_Vo.Caption = ""
    NoInversor.Valor_Saturacion.Caption = ""

ElseIf NoInversor.Valor_Vi.Text = Empty Or
NoInversor.Valor_VsatPos.Text = Empty Or NoInversor.Valor_VsatNeg.Text
= Empty Then

    MsgBox "Debe introducir un valor en las casillas del Voltaje",
vbExclamation + vbOKOnly, "AVISO"

    NoInversor.Valor_R3.Caption = ""
    NoInversor.Valor_Ganancia.Caption = ""
    NoInversor.Valor_Vo.Caption = ""
```



```
NoInversor.Valor_Saturacion.Caption = ""

Else

    'Asignamos las variables a los correspondientes parámetros del
    circuito no inversor'
    R1 = NoInversor.Valor_R1.Text * 1000
    R2 = NoInversor.Valor_R2.Text * 1000
    Vi = NoInversor.Valor_Vi.Text
    VsatPos = NoInversor.Valor_VsatPos.Text
    VsatNeg = NoInversor.Valor_VsatNeg.Text

    'Introducimos la condición de que -Vsat NO puede ser mayor que
    +Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
    de Vo'
    If VsatPos < VsatNeg Then

        MsgBox "La tensión de saturación -Vsat no puede ser mayor que
        +Vsat", vbInformation + vbOKOnly, "AVISO"

        NoInversor.Valor_VsatPos.Text = ""
        NoInversor.Valor_VsatNeg.Text = ""
        NoInversor.Valor_R3.Caption = ""
        NoInversor.Valor_Ganancia.Caption = ""
        NoInversor.Valor_Vo.Caption = ""
        NoInversor.Valor_Saturacion.Caption = ""

    Else

        'Definimos las ecuaciones de los parametros a calcular y
        mostramos el resultado'
        R3 = ((R2 * R1) / (R2 + R1)) / 1000
        Ganancia = ((R1 + R2) / R1)
        Vo = (Ganancia * Vi)

        NoInversor.Valor_R3.Caption = R3
        NoInversor.Valor_Ganancia.Caption = Ganancia
        NoInversor.Valor_Vo.Caption = Vo

        'Analizamos si la tensión de salida está saturada'
        If Vo >= VsatPos Then

            NoInversor.Valor_Vo.Caption = VsatPos
            NoInversor.Valor_Saturacion.Caption = "Salida saturada"

        ElseIf Vo <= VsatNeg Then

            NoInversor.Valor_Vo.Caption = VsatNeg
            NoInversor.Valor_Saturacion.Caption = "Salida saturada"

        Else

            NoInversor.Valor_Saturacion.Caption = ""
            NoInversor.Valor_Vo.Caption = Vo

        End If

        'Mostramos la representación Vo-Vi'

        'Calculamos los valores de Vi para los que Vo satura, ya que
        serán necesarios para obtener la grafica'
```



```
Vi_Neg = VsatNeg / Ganancia
Vi_Pos = VsatPos / Ganancia

'Definimos los puntos necesarios para obtener nuestra grafica'
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c5") =
Ganancia
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c10") =
VsatPos
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d10") =
VsatPos
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c11") =
VsatNeg
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("d11") =
VsatNeg
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c13") =
Vi_Pos
ActiveWorkbook.Worksheets("Gráfica Vo-Vi").Range("c14") =
Vi_Neg

End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub RestaurarValores ()

'Vaciamos los valores de cada uno de los parámetros'
NoInversor.Valor_R1.Text = ""
NoInversor.Valor_R2.Text = ""
NoInversor.Valor_Vi.Text = ""
NoInversor.Valor_VsatPos.Text = ""
NoInversor.Valor_VsatNeg.Text = ""
NoInversor.Valor_R3.Caption = ""
NoInversor.Valor_Ganancia.Caption = ""
NoInversor.Valor_Vo.Caption = ""
NoInversor.Valor_Saturacion.Caption = ""

End Sub
```

7.2.3.18. Generador de onda cuadrada (ModOndaCua)

```
Sub MostrarOndaCuadrada ()

'Mostramos la pestaña de la aplicación del circuito generador de onda
cuadrada'
OndaCuadrada.Show

End Sub

Sub MostrarTeoriaOndaCuadrada ()
```



```
'Mostramos la pestaña de la aplicación del circuito generador de onda
cuadrada'
TeoriaOndaCuadrada.Show

End Sub

Sub MostrarInformacion ()

'Mostramos la pestaña de la información del circuito generador de onda
cuadrada'
InformacionOndas.Show

End Sub

Sub CalcularValores ()

'Declaramos las variables de los parámetros del circuito generador de
onda cuadrada'
Dim R As Single
Dim R1 As Single
Dim R2 As Single
Dim C As Single
Dim Beta As Single
Dim VsatPos As Single
Dim VsatNeg As Single
Dim Periodo As Single
Dim Frecuencia As Single
Dim Vc As Single
Dim Vo As Single

'Mediante esta declaración conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de las resistencias y el condensador, su valor no puede
ser cero o negativo'
'En el caso de las tensiones, su valor si puede ser cero o negativo'
If OndaCuadrada.Valor_R.Text <= 0 Or OndaCuadrada.Valor_R1.Text <= 0
Or OndaCuadrada.Valor_R2.Text <= 0 Or OndaCuadrada.Valor_C.Text <= 0
Or OndaCuadrada.Valor_R.Text = Empty Or OndaCuadrada.Valor_R1.Text =
Empty Or OndaCuadrada.Valor_R2.Text = Empty Or
OndaCuadrada.Valor_C.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas de las
resistencias y el condensador, y que no sea ni nulo ni negativo",
vbExclamation + vbOKOnly, "AVISO"

    OndaCuadrada.Periodo.Caption = ""
    OndaCuadrada.Frecuencia.Caption = ""

ElseIf OndaCuadrada.Valor_VsatPos.Text = Empty Or
OndaCuadrada.Valor_VsatNeg.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas del Voltaje",
vbExclamation + vbOKOnly, "AVISO"

    OndaCuadrada.Periodo.Caption = ""
    OndaCuadrada.Frecuencia.Caption = ""
```



```
Else

    'Asignamos las variables a los correspondientes parámetros del
    circuito generador de onda cuadrada'
    R = OndaCuadrada.Valor_R.Text * 1000
    R1 = OndaCuadrada.Valor_R1.Text * 1000
    R2 = OndaCuadrada.Valor_R2.Text * 1000
    C = OndaCuadrada.Valor_C.Text / 1000000
    VsatPos = OndaCuadrada.Valor_VsatPos.Text
    VsatNeg = OndaCuadrada.Valor_VsatNeg.Text

    'Introducimos la condición de que -Vsat NO puede ser mayor que
    +Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
    de Vo'
    If VsatPos < VsatNeg Then

        MsgBox "La tensión de saturación -Vsat no puede ser mayor que
        +Vsat", vbInformation + vbOKOnly, "AVISO"

        OndaCuadrada.Valor_VsatPos.Text = ""
        OndaCuadrada.Valor_VsatNeg.Text = ""
        OndaCuadrada.Periodo.Caption = ""
        OndaCuadrada.Frecuencia.Caption = ""

    Else

        'Definimos las ecuaciones de los parametros a calcular y
        mostramos el resultado'
        Beta = R1 / (R1 + R2)
        Periodo = 2 * R * C * Log((1 + Beta) / (1 - Beta))
        Frecuencia = (1 / Periodo) / 1000

        OndaCuadrada.Periodo.Caption = Periodo
        OndaCuadrada.Frecuencia.Caption = Frecuencia

        'Mostramos la representación Vo-t'
        ActiveWorkbook.Worksheets("Ondas").Range("b9:b400") = Empty
        ActiveWorkbook.Worksheets("Ondas").Range("c8:c400") = Empty
        ActiveWorkbook.Worksheets("Ondas").Range("d8:d400") = Empty

        'Definimos todas las variables necesarias para realizar el
        bucle'
        Dim i As Single
        Dim i1 As Single
        Dim i2 As Single
        Dim i3 As Single
        Dim j As Single
        Dim j1 As Single
        Dim j2 As Single
        Dim j3 As Single
        Dim j4 As Single
        Dim j5 As Single
        Dim j6 As Single
        Dim j7 As Single
        Dim j8 As Single
        Dim j9 As Single
        Dim j10 As Single
        Dim j11 As Single
        Dim j12 As Single
        Dim j13 As Single
        Dim j14 As Single
```



```
Dim j15 As Single
Dim j16 As Single
Dim j17 As Single
Dim j18 As Single
Dim n As Double
Dim n1 As Double
Dim n2 As Double

n = 0
n1 = 0
n2 = 0
i = 8
i3 = 9
j = 0
j1 = 0
j2 = 0
j3 = 0
j4 = 0
j5 = 0
j6 = 0
j7 = 0
j8 = 0
j9 = 0
j10 = 0
j11 = 0
j12 = 0
j13 = 0
j14 = 0
j15 = 0
j16 = 0
j17 = 0
j18 = 0
j19 = 0
j20 = 0

n = ((2 * R * C * Log((1 + Beta) / (1 - Beta))) / 2) * 10
n1 = n / 100
n2 = n / 100

'Cada bucle genera una onda de Vc, y por tanto de Vo'
For i1 = 0 To n Step n1

    C))
        Vc = VsatPos + (VsatNeg * Beta - VsatPos) * Exp(-j / (R *
        Vo = VsatPos
        j = j + n1

        If Vc > Beta * VsatPos Then

            (R * C))
                Vo = VsatNeg
                Vc = VsatNeg + (VsatPos * Beta - VsatNeg) * Exp(-j1 /
                j1 = j1 + n1

                If Vc < Beta * VsatNeg Then

                    j2 / (R * C))
                        Vo = VsatPos
                        Vc = VsatPos + (VsatNeg * Beta - VsatPos) * Exp(-
                        j2 = j2 + n1
```




```

    If Vc > Beta * VsatPos Then
        Vo = VsatNeg
        Vc = VsatNeg + (VsatPos * Beta - VsatNeg) *
Exp(-j3 / (R * C))
        j3 = j3 + n1
    If Vc < Beta * VsatNeg Then
        Vo = VsatPos
        Vc = VsatPos + (VsatNeg * Beta - VsatPos)
* Exp(-j4 / (R * C))
        j4 = j4 + n1
    If Vc > Beta * VsatPos Then
        Vo = VsatNeg
        Vc = VsatNeg + (VsatPos * Beta -
VsatNeg) * Exp(-j5 / (R * C))
        j5 = j5 + n1
    If Vc < Beta * VsatNeg Then
        Vo = VsatPos
        Vc = VsatPos + (VsatNeg * Beta -
VsatPos) * Exp(-j6 / (R * C))
        j6 = j6 + n1
    If Vc > Beta * VsatPos Then
        Vo = VsatNeg
        Vc = VsatNeg + (VsatPos * Beta
- VsatNeg) * Exp(-j7 / (R * C))
        j7 = j7 + n1
    If Vc < Beta * VsatNeg Then
        Vo = VsatPos
        Vc = VsatPos + (VsatNeg *
Beta - VsatPos) * Exp(-j8 / (R * C))
        j8 = j8 + n1
Then
    If Vc > Beta * VsatPos
        Vo = VsatNeg
        Vc = VsatNeg +
(VsatPos * Beta - VsatNeg) * Exp(-j9 / (R * C))
        j9 = j9 + n1
    If Vc < Beta * VsatNeg
Then
        Vo = VsatPos
        Vc = VsatPos +
(VsatNeg * Beta - VsatPos) * Exp(-j10 / (R * C))
        j10 = j10 + n1
    If Vc > Beta *
VsatPos Then

```



```

(VsatPos * Beta - VsatNeg) * Exp(-j11 / (R * C))
VsatNeg Then
VsatPos
VsatPos + (VsatNeg * Beta - VsatPos) * Exp(-j12 / (R * C))
+ n1
Beta * VsatPos Then
VsatNeg
VsatNeg + (VsatPos * Beta - VsatNeg) * Exp(-j13 / (R * C))
j13 + n1
< Beta * VsatNeg Then
= VsatPos
= VsatPos + (VsatNeg * Beta - VsatPos) * Exp(-j14 / (R * C))
j14 = j14 + n1
Vc > Beta * VsatPos Then
Vo = VsatNeg
Vc = VsatNeg + (VsatPos * Beta - VsatNeg) * Exp(-j15 / (R * C))
j15 = j15 + n1
If Vc < Beta * VsatNeg Then
Vo = VsatPos
Vc = VsatPos + (VsatNeg * Beta - VsatPos) * Exp(-j16 / (R * C))
j16 = j16 + n1
If Vc > Beta * VsatPos Then
Vo = VsatNeg

```

```

Vo = VsatNeg
Vc = VsatNeg +
j11 = j11 + n1
If Vc < Beta *

```

```

Vo =
Vc =
j12 = j12

```

```

If Vc >

```

```

Vo =
Vc =
j13 =

```

```

If Vc

```

```

Vo

```

```

Vc

```

```

If

```



```
Vc = VsatNeg + (VsatPos * Beta - VsatNeg) * Exp(-j17 / (R * C))
```

```
j17 = j17 + n1
```

```
If Vc < Beta * VsatNeg Then
```

```
Vo = VsatPos
```

```
Vc = VsatPos + (VsatNeg * Beta - VsatPos) * Exp(-j18 / (R * C))
```

```
j18 = j18 + n1
```

```
If Vc > Beta * VsatPos Then
```

```
Vo = VsatNeg
```

```
Vc = VsatNeg + (VsatPos * Beta - VsatNeg) * Exp(-j19 / (R * C))
```

```
j19 = j19 + n1
```

```
If Vc < Beta * VsatNeg Then
```

```
Vo = VsatPos
```

```
Vc = VsatPos + (VsatNeg * Beta - VsatPos) * Exp(-j20 / (R * C))
```

```
j20 = j20 + n1
```

```
End If
```

```
End If
```

```
End If
```

```
End If
```

```
End If
```

```
End If
```

```
End If
```

```
End If
```

```
End If
```

```
End If
```




```
End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub RestaurarValores ()

'Vaciamos los valores de cada uno de los parámetros'
OndaCuadrada.Valor_R.Text = ""
OndaCuadrada.Valor_R1.Text = ""
OndaCuadrada.Valor_R2.Text = ""
OndaCuadrada.Valor_C.Text = ""
OndaCuadrada.Valor_VsatPos.Text = ""
OndaCuadrada.Valor_VsatNeg.Text = ""
OndaCuadrada.Periodo.Caption = ""
OndaCuadrada.Frecuencia.Caption = ""

End Sub
```

7.2.3.19. Generador de onda triangular (ModOndaTri)

```
Sub MostrarOndaTriangular ()

'Mostramos la pestaña de la aplicación del circuito generador de onda triangular'
OndaTriangular.Show

End Sub

Sub MostrarTeoriaOndaTriangular ()

'Mostramos la pestaña de la aplicación del circuito generador de onda triangular'
TeoriaOndaTriangular.Show

End Sub

Sub MostrarInformacion ()

'Mostramos la pestaña de la información del circuito generador de onda triangular'
InformacionOndas.Show

End Sub

Sub CalcularValores ()

'Declaremos las variables de los parámetros del circuito generador de onda triangular'
Dim R As Single
```



```
Dim R1 As Single
Dim R2 As Single
Dim C As Single
Dim VsatPos As Single
Dim VsatNeg As Single
Dim Periodo As Single
Dim Frecuencia As Single
Dim Vo1 As Single
Dim Vo As Single

'Mediante esta declaración conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de las resistencias y el condensador, su valor no puede
ser cero o negativo'
'En el caso de las tensiones, su valor si puede ser cero o negativo'
If OndaTriangular.Valor_R.Text <= 0 Or OndaTriangular.Valor_R1.Text <=
0 Or OndaTriangular.Valor_R2.Text <= 0 Or OndaTriangular.Valor_C.Text
<= 0 Or OndaTriangular.Valor_R.Text = Empty Or
OndaTriangular.Valor_R1.Text = Empty Or OndaTriangular.Valor_R2.Text =
Empty Or OndaTriangular.Valor_C.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas de las
resistencias y el condensador, y que no sea ni nulo ni negativo",
vbExclamation + vbOKOnly, "AVISO"

    OndaTriangular.Periodo.Caption = ""
    OndaTriangular.Frecuencia.Caption = ""

ElseIf OndaTriangular.Valor_VsatPos.Text = Empty Or
OndaTriangular.Valor_VsatNeg.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas del Voltaje",
vbExclamation + vbOKOnly, "AVISO"

    OndaTriangular.Periodo.Caption = ""

    OndaTriangular.Periodo.Caption = ""
    OndaTriangular.Frecuencia.Caption = ""

Else

    'Asignamos las variables a los correspondientes parámetros del
circuito generador de onda triangular'
    R = OndaTriangular.Valor_R.Text * 1000
    R1 = OndaTriangular.Valor_R1.Text * 1000
    R2 = OndaTriangular.Valor_R2.Text * 1000
    C = OndaTriangular.Valor_C.Text / 1000000
    VsatPos = OndaTriangular.Valor_VsatPos.Text
    VsatNeg = OndaTriangular.Valor_VsatNeg.Text

    'Introducimos la condición de que -Vsat NO puede ser mayor que
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
de Vo'
    If VsatPos < VsatNeg Then

        MsgBox "La tensión de saturación -Vsat no puede ser mayor que
+Vsat", vbInformation + vbOKOnly, "AVISO"
```



```
OndaTriangular.Valor_VsatPos.Text = ""
OndaTriangular.Valor_VsatNeg.Text = ""
OndaTriangular.Periodo.Caption = ""
OndaTriangular.Frecuencia.Caption = ""

Else

    'Definimos las ecuaciones de los parametros a calcular y
    mostramos el resultado'
    Periodo = 4 * R * C * (R1 / R2)
    Frecuencia = (1 / Periodo) / 1000

    OndaTriangular.Periodo.Caption = Periodo
    OndaTriangular.Frecuencia.Caption = Frecuencia

    'Mostramos la representación Vo-t'
    ActiveWorkbook.Worksheets("Ondas").Range("q9:q400") = Empty
    ActiveWorkbook.Worksheets("Ondas").Range("r8:r400") = Empty
    ActiveWorkbook.Worksheets("Ondas").Range("s8:s400") = Empty

    'Definimos todas las variables necesarias para realizar el
    bucle'
    Dim i As Single
    Dim i1 As Single
    Dim i2 As Single
    Dim i3 As Single
    Dim j As Single
    Dim j1 As Single
    Dim j2 As Single
    Dim j3 As Single
    Dim j4 As Single
    Dim j5 As Single
    Dim j6 As Single
    Dim j7 As Single
    Dim j8 As Single
    Dim j9 As Single
    Dim j10 As Single
    Dim j11 As Single
    Dim j12 As Single
    Dim j13 As Single
    Dim j14 As Single
    Dim j15 As Single
    Dim j16 As Single
    Dim j17 As Single
    Dim j18 As Single
    Dim j19 As Single
    Dim j20 As Single
    Dim n As Double
    Dim n1 As Double
    Dim n2 As Double

    n = 0
    n1 = 0
    n2 = 0
    i = 8
    i3 = 9
    j = 0
    j1 = 0
    j2 = 0
    j3 = 0
    j4 = 0
```



```
j5 = 0
j6 = 0
j7 = 0
j8 = 0
j9 = 0
j10 = 0
j11 = 0
j12 = 0
j13 = 0
j14 = 0
j15 = 0
j16 = 0
j17 = 0
j18 = 0
j19 = 0
j20 = 0

n = ((4 * R * C * (R1 / R2)) / 2) * 10
n1 = n / 100
n2 = n / 100

'Cada bucle genera una onda de Vo, y por consiguiente, de Vo1
For i1 = 0 To n Step n1

    Vo = (VsatNeg * j) / (R * C) + VsatPos * (R1 / R2)
    Vo1 = VsatPos
    j = j + n1

    If Vo < VsatNeg * (R1 / R2) Then

        Vo = (VsatPos * j1) / (R * C) + VsatNeg * (R1 / R2)
        Vo1 = VsatNeg
        j1 = j1 + n1

    If Vo > VsatPos * (R1 / R2) Then

        Vo = (VsatNeg * j2) / (R * C) + VsatPos * (R1 /
R2)

        Vo1 = VsatPos
        j2 = j2 + n1

    If Vo < VsatNeg * (R1 / R2) Then

        Vo = (VsatPos * j3) / (R * C) + VsatNeg * (R1
/ R2)

        Vo1 = VsatNeg
        j3 = j3 + n1

    If Vo > VsatPos * (R1 / R2) Then

        Vo = (VsatNeg * j4) / (R * C) + VsatPos *
(R1 / R2)

        Vo1 = VsatPos
        j4 = j4 + n1

    If Vo < VsatNeg * (R1 / R2) Then

        Vo = (VsatPos * j5) / (R * C) +
VsatNeg * (R1 / R2)

        Vo1 = VsatNeg
        j5 = j5 + n1
```




```

VsatPos * (R1 / R2)

+ VsatNeg * (R1 / R2)

Then

C) + VsatPos * (R1 / R2)

R2) Then

(R * C) + VsatNeg * (R1 / R2)

/ R2) Then

j10) / (R * C) + VsatPos * (R1 / R2)

(R1 / R2) Then

* j11) / (R * C) + VsatNeg * (R1 / R2)

VsatPos * (R1 / R2) Then

(VsatNeg * j12) / (R * C) + VsatPos * (R1 / R2)

VsatPos

+ n1

VsatNeg * (R1 / R2) Then

If Vo > VsatPos * (R1 / R2) Then

Vo = (VsatNeg * j6) / (R * C) +

Vo1 = VsatPos
j6 = j6 + n1

If Vo < VsatNeg * (R1 / R2) Then

Vo = (VsatPos * j7) / (R * C)

Vo1 = VsatNeg
j7 = j7 + n1

If Vo > VsatPos * (R1 / R2)

Vo = (VsatNeg * j8) / (R *

Vo1 = VsatPos
j8 = j8 + n1

If Vo < VsatNeg * (R1 /

Vo = (VsatPos * j9) /

Vo1 = VsatNeg
j9 = j9 + n1

If Vo > VsatPos * (R1

Vo = (VsatNeg *

Vo1 = VsatPos
j10 = j10 + n1

If Vo < VsatNeg *

Vo = (VsatPos

Vo1 = VsatNeg
j11 = j11 + n1

If Vo >

Vo =

Vo1 =

j12 = j12

If Vo <

```



$$(V_{satPos} * j_{13}) / (R * C) + V_{satNeg} * (R1 / R2)$$

V_{satNeg}

$$j_{13} + n1$$

> $V_{satPos} * (R1 / R2)$ Then

$$= (V_{satNeg} * j_{14}) / (R * C) + V_{satPos} * (R1 / R2)$$

$V_{o1} = V_{satPos}$

$$j_{14} = j_{14} + n1$$

$V_o < V_{satNeg} * (R1 / R2)$ Then

$$V_o = (V_{satPos} * j_{15}) / (R * C) + V_{satNeg} * (R1 / R2)$$

$V_{o1} = V_{satNeg}$

$$j_{15} = j_{15} + n1$$

If $V_o > V_{satPos} * (R1 / R2)$ Then

$$V_o = (V_{satNeg} * j_{16}) / (R * C) + V_{satPos} * (R1 / R2)$$

$V_{o1} = V_{satPos}$

$$j_{16} = j_{16} + n1$$

If $V_o < V_{satNeg} * (R1 / R2)$ Then

$$V_o = (V_{satPos} * j_{17}) / (R * C) + V_{satNeg} * (R1 / R2)$$

$V_{o1} = V_{satNeg}$

$$j_{17} = j_{17} + n1$$

If $V_o > V_{satPos} * (R1 / R2)$ Then

$$V_o = (V_{satNeg} * j_{18}) / (R * C) + V_{satPos} * (R1 / R2)$$

$V_{o1} = V_{satPos}$

$$j_{18} = j_{18} + n1$$

If $V_o < V_{satNeg} * (R1 / R2)$ Then

$V_o =$

$V_{o1} =$

$j_{13} =$

If V_o

V_o

If



$V_o = (V_{satPos} * j_{19}) / (R * C) + V_{satNeg} * (R_1 / R_2)$

$V_{o1} = V_{satNeg}$

$j_{19} = j_{19} + n_1$

If $V_o > V_{satPos} * (R_1 / R_2)$ Then

$V_o = (V_{satNeg} * j_{20}) / (R * C) + V_{satPos} * (R_1 / R_2)$

$V_{o1} = V_{satPos}$

$j_{20} = j_{20} + n_1$

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If



```
End If

'Nos aseguramos de que la salida no satura'
If Vo >= VsatPos Then

    Vo = VsatPos

ElseIf Vo <= VsatNeg Then

    Vo = VsatNeg

End If

If Vol >= VsatPos Then

    Vol = VsatPos

ElseIf Vol <= VsatNeg Then

    Vol = VsatNeg

End If

Worksheets("Ondas").Cells(i, 18).Value = Vo
Worksheets("Ondas").Cells(i, 19).Value = Vol
i = i + 1

Next i1

'Definimos los valores de t de nuestra gráfica'
For i2 = 0 To n Step n1

    Worksheets("Ondas").Cells(i3, 17).Value = n1
    n1 = n1 + n2
    i3 = i3 + 1

Next i2

'Definimos los puntos necesarios para obtener nuestra gráfica'
ActiveWorkbook.Worksheets("Ondas").Range("q8") = 0
ActiveWorkbook.Worksheets("Ondas").Range("b2") = Periodo
ActiveWorkbook.Worksheets("Ondas").Range("c2") = Frecuencia

End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub RestaurarValores()

'Vaciamos los valores de cada uno de los parámetros'
OndaTriangular.Valor_R.Text = ""
OndaTriangular.Valor_R1.Text = ""
OndaTriangular.Valor_R2.Text = ""
OndaTriangular.Valor_C.Text = ""
OndaTriangular.Valor_VsatPos.Text = ""
OndaTriangular.Valor_VsatNeg.Text = ""
```



```
OndaTriangular.Periodo.Caption = ""  
OndaTriangular.Frecuencia.Caption = ""
```

```
End Sub
```

7.2.3.20. Circuitos restadores (ModRestador)

```
Sub MostrarRestadorSimple ()
```

```
'Mostramos la pestaña de la aplicación del circuito restador simple'  
RestadorSimple.Show
```

```
End Sub
```

```
Sub MostrarTeoria1 ()
```

```
'Mostramos la pestaña de la teoría del circuito restador simple'  
TeoriaResSimple.Show
```

```
End Sub
```

```
Sub MostrarInformacion1 ()
```

```
'Mostramos la pestaña de la información del circuito restador simple'  
Informacion1.Show
```

```
End Sub
```

```
Sub CalcularValores1 ()
```

```
'Declaramos las variables de los parámetros de un circuito restador  
simple'
```

```
Dim R1 As Single
```

```
Dim R2 As Single
```

```
Dim R3 As Single
```

```
Dim R4 As Single
```

```
Dim Vi1 As Single
```

```
Dim Vi2 As Single
```

```
Dim VsatPos As Single
```

```
Dim VsatNeg As Single
```

```
Dim Ganancia_Vi1 As Single
```

```
Dim Ganancia_Vi2 As Single
```

```
Dim Vo As Single
```

```
'Mediante esta declaración conseguimos que nuestra macro no pare de  
ejecutarse aunque se encuentre un error'
```

```
On Error Resume Next
```

```
'Nos aseguramos de que se introduzca un valor en todas las casillas'
```

```
'En el caso de las resistencias, su valor no puede ser cero o  
negativo'
```

```
'En el caso de las tensiones, su valor si puede ser cero o negativo'
```

```
If RestadorSimple.Valor_R1.Text <= 0 Or RestadorSimple.Valor_R2.Text  
<= 0 Or RestadorSimple.Valor_R3.Text <= 0 Or
```



```
RestadorSimple.Valor_R4.Text <= 0 Or RestadorSimple.Valor_R1.Text =  
Empty Or RestadorSimple.Valor_R2.Text = Empty Or  
RestadorSimple.Valor_R3.Text = Empty Or RestadorSimple.Valor_R4.Text =  
Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas de las  
resistencias y que no sea ni nulo ni negativo", vbExclamation +  
vbOKOnly, "AVISO"
```

```
RestadorSimple.Valor_Ganancia_Vi1.Caption = ""  
RestadorSimple.Valor_Ganancia_Vi2.Caption = ""  
RestadorSimple.Valor_Vo.Caption = ""  
RestadorSimple.Valor_Saturacion.Caption = ""
```

```
ElseIf RestadorSimple.Valor_Vi1.Text = Empty Or  
RestadorSimple.Valor_Vi2.Text = Empty Or  
RestadorSimple.Valor_VsatPos.Text = Empty Or  
RestadorSimple.Valor_VsatNeg.Text = Empty Then
```

```
MsgBox "Debe introducir un valor en las casillas del Voltaje",  
vbExclamation + vbOKOnly, "AVISO"
```

```
RestadorSimple.Valor_Ganancia_Vi1.Caption = ""  
RestadorSimple.Valor_Ganancia_Vi2.Caption = ""  
RestadorSimple.Valor_Vo.Caption = ""  
RestadorSimple.Valor_Saturacion.Caption = ""
```

Else

```
'Asignamos las variables a los correspondientes parámetros del  
circuito restador simple'
```

```
R1 = RestadorSimple.Valor_R1.Text * 1000  
R2 = RestadorSimple.Valor_R2.Text * 1000  
R3 = RestadorSimple.Valor_R3.Text * 1000  
R4 = RestadorSimple.Valor_R4.Text * 1000  
Vi1 = RestadorSimple.Valor_Vi1.Text  
Vi2 = RestadorSimple.Valor_Vi2.Text  
VsatPos = RestadorSimple.Valor_VsatPos.Text  
VsatNeg = RestadorSimple.Valor_VsatNeg.Text
```

```
'Introducimos la condición de que -Vsat NO puede ser mayor que  
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación  
de Vo'
```

```
If VsatPos < VsatNeg Then
```

```
MsgBox "La tensión de saturación -Vsat no puede ser mayor que  
+Vsat", vbInformation + vbOKOnly, "AVISO"
```

```
RestadorSimple.Valor_VsatPos.Text = ""  
RestadorSimple.Valor_VsatNeg.Text = ""  
RestadorSimple.Valor_Ganancia_Vi1.Caption = ""  
RestadorSimple.Valor_Ganancia_Vi2.Caption = ""  
RestadorSimple.Valor_Vo.Caption = ""  
RestadorSimple.Valor_Saturacion.Caption = ""
```

Else

```
'Definimos las ecuaciones de los parametros a calcular y  
mostramos el resultado'
```

```
Ganancia_Vi1 = (-(R2 / R1))  
Ganancia_Vi2 = (R4 * (R1 + R2)) / ((R3 + R4) * R1)
```



```
Vo = (Ganancia_Vi1 * Vi1 + Ganancia_Vi2 * Vi2)

RestadorSimple.Valor_Ganancia_Vi1.Caption = Ganancia_Vi1
RestadorSimple.Valor_Ganancia_Vi2.Caption = Ganancia_Vi2
RestadorSimple.Valor_Vo.Caption = Vo

'Analizamos si la tensión de salida está saturada'
If Vo >= VsatPos Then

    RestadorSimple.Valor_Vo.Caption = VsatPos
    RestadorSimple.Valor_Saturacion.Caption = "Salida
saturada"

ElseIf Vo <= VsatNeg Then

    RestadorSimple.Valor_Vo.Caption = VsatNeg
    RestadorSimple.Valor_Saturacion.Caption = "Salida
saturada"

Else

    RestadorSimple.Valor_Saturacion.Caption = ""
    RestadorSimple.Valor_Vo.Caption = Vo

End If

End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub RestaurarValores1 ()

'Vaciamos los valores de cada uno de los parámetros'
RestadorSimple.Valor_R1.Text = ""
RestadorSimple.Valor_R2.Text = ""
RestadorSimple.Valor_R3.Text = ""
RestadorSimple.Valor_R4.Text = ""
RestadorSimple.Valor_Vi1.Text = ""
RestadorSimple.Valor_Vi2.Text = ""
RestadorSimple.Valor_VsatPos.Text = ""
RestadorSimple.Valor_VsatNeg.Text = ""
RestadorSimple.Valor_Ganancia_Vi1.Caption = ""
RestadorSimple.Valor_Ganancia_Vi2.Caption = ""
RestadorSimple.Valor_Vo.Caption = ""
RestadorSimple.Valor_Saturacion.Caption = ""

End Sub

Sub MostrarRestadorAltaImp ()

'Mostramos la pestaña de la aplicación del circuito restador de alta
impedancia a la entrada'
RestadorAltaImp.Show

End Sub
```



```
Sub MostrarTeoria2 ()

'Mostramos la pestaña de la teoría del circuito restador de alta
impedancia a la entrada'
TeoriaResAltaImp.Show

End Sub

Sub MostrarInformacion2 ()

'Mostramos la pestaña de la información del circuito restador de alta
impedancia a la entrada'
Informacion1.Show

End Sub

Sub CalcularValores2 ()

'Declaramos las variables de los parámetros de un circuito de alta
impedancia a la entrada'
Dim R1 As Single
Dim R2 As Single
Dim R4 As Single
Dim R5 As Single
Dim Vi1 As Single
Dim Vi2 As Single
Dim VsatPos As Single
Dim VsatNeg As Single
Dim R3 As Single
Dim R6 As Single
Dim Ganancia_Vo1 As Single
Dim Ganancia_Vi1 As Single
Dim Ganancia_Vi2 As Single
Dim Vo1 As Single
Dim Vo As Single

'Mediante esta declaración conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de las resistencias, su valor no puede ser cero o
negativo'
'En el caso de las tensiones, su valor si puede ser cero o negativo'
If RestadorAltaImp.Valor_R1.Text <= 0 Or RestadorAltaImp.Valor_R2.Text
<= 0 Or RestadorAltaImp.Valor_R5.Text <= 0 Or
RestadorAltaImp.Valor_R4.Text <= 0 Or RestadorAltaImp.Valor_R1.Text =
Empty Or RestadorAltaImp.Valor_R2.Text = Empty Or
RestadorAltaImp.Valor_R5.Text = Empty Or RestadorAltaImp.Valor_R4.Text
= Empty Then

    MsgBox "Debe introducir un valor en las casillas de las
resistencias y que no sea ni nulo ni negativo", vbExclamation +
vbOKOnly, "AVISO"

    RestadorAltaImp.Valor_R3.Caption = ""
    RestadorAltaImp.Valor_R6.Caption = ""
    RestadorAltaImp.Valor_Ganancia_Vol.Caption = ""
    RestadorAltaImp.Valor_Ganancia_Vi1.Caption = ""
    RestadorAltaImp.Valor_Ganancia_Vi2.Caption = ""
    RestadorAltaImp.Valor_Vol.Caption = ""
```




```
RestadorAltaImp.Valor_Vo.Caption = ""
RestadorAltaImp.Valor_Saturacion_Vol.Caption = ""
RestadorAltaImp.Valor_Saturacion_Vo.Caption = ""

ElseIf RestadorAltaImp.Valor_Vi1.Text = Empty Or
RestadorAltaImp.Valor_Vi2.Text = Empty Or
RestadorAltaImp.Valor_VsatPos.Text = Empty Or
RestadorAltaImp.Valor_VsatNeg.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas del Voltaje",
vbExclamation + vbOKOnly, "AVISO"

    RestadorAltaImp.Valor_R3.Caption = ""
    RestadorAltaImp.Valor_R6.Caption = ""
    RestadorAltaImp.Valor_Ganancia_Vol.Caption = ""
    RestadorAltaImp.Valor_Ganancia_Vi1.Caption = ""
    RestadorAltaImp.Valor_Ganancia_Vi2.Caption = ""
    RestadorAltaImp.Valor_Vol.Caption = ""
    RestadorAltaImp.Valor_Vo.Caption = ""
    RestadorAltaImp.Valor_Saturacion_Vol.Caption = ""
    RestadorAltaImp.Valor_Saturacion_Vo.Caption = ""

Else

    'Asignamos las variables a los correspondientes parámetros del
circuito restador de alta impedancia a la entrada'
    R1 = RestadorAltaImp.Valor_R1.Text * 1000
    R2 = RestadorAltaImp.Valor_R2.Text * 1000
    R4 = RestadorAltaImp.Valor_R4.Text * 1000
    R5 = RestadorAltaImp.Valor_R5.Text * 1000
    Vi1 = RestadorAltaImp.Valor_Vi1.Text
    Vi2 = RestadorAltaImp.Valor_Vi2.Text
    VsatPos = RestadorAltaImp.Valor_VsatPos.Text
    VsatNeg = RestadorAltaImp.Valor_VsatNeg.Text

    'Introducimos la condición de que -Vsat NO puede ser mayor que
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación
de Vo'
    If VsatPos < VsatNeg Then

        MsgBox "La tensión de saturación -Vsat no puede ser mayor que
+Vsat", vbInformation + vbOKOnly, "AVISO"

        RestadorAltaImp.Valor_VsatPos.Text = ""
        RestadorAltaImp.Valor_VsatNeg.Text = ""
        RestadorAltaImp.Valor_R3.Caption = ""
        RestadorAltaImp.Valor_R6.Caption = ""
        RestadorAltaImp.Valor_Ganancia_Vol.Caption = ""
        RestadorAltaImp.Valor_Ganancia_Vi1.Caption = ""
        RestadorAltaImp.Valor_Ganancia_Vi2.Caption = ""
        RestadorAltaImp.Valor_Vol.Caption = ""
        RestadorAltaImp.Valor_Vo.Caption = ""
        RestadorAltaImp.Valor_Saturacion_Vol.Caption = ""
        RestadorAltaImp.Valor_Saturacion_Vo.Caption = ""

    Else

        'Definimos las ecuaciones de los parametros a calcular y
mostramos el resultado'
        R3 = ((R1 * R2) / (R1 + R2)) / 1000
        R6 = ((R4 * R5) / (R4 + R5)) / 1000
```



```
Ganancia_Vo1 = (R1 + R2) / R1
Ganancia_Vi1 = -((1 + (R2 / R1)) * (R5 / R4))
Ganancia_Vi2 = 1 + (R5 / R4)
Vo1 = Ganancia_Vo1 * Vi1
Vo = Ganancia_Vi1 * Vi1 + Ganancia_Vi2 * Vi2

RestadorAltaImp.Valor_R3.Caption = R3
RestadorAltaImp.Valor_R6.Caption = R6
RestadorAltaImp.Valor_Ganancia_Vo1.Caption = Ganancia_Vo1
RestadorAltaImp.Valor_Ganancia_Vi1.Caption = Ganancia_Vi1
RestadorAltaImp.Valor_Ganancia_Vi2.Caption = Ganancia_Vi2
RestadorAltaImp.Valor_Vo1.Caption = Vo1
RestadorAltaImp.Valor_Vo.Caption = Vo

'Analizamos si la tensión Vo1 está saturada'
If Vo1 >= VsatPos Then

    RestadorAltaImp.Valor_Vo1.Caption = VsatPos
    RestadorAltaImp.Valor_Saturacion_Vo1.Caption = "Vo1 está
saturada"

ElseIf Vo1 <= VsatNeg Then

    RestadorAltaImp.Valor_Vo1.Caption = VsatNeg
    RestadorAltaImp.Valor_Saturacion_Vo1.Caption = "Vo1 está
saturada"

Else

    RestadorAltaImp.Valor_Saturacion_Vo1.Caption = ""
    RestadorAltaImp.Valor_Vo1.Caption = Vo1

End If

'Analizamos si la tensión de salida Vo está saturada'
If Vo >= VsatPos Then

    RestadorAltaImp.Valor_Vo.Caption = VsatPos
    RestadorAltaImp.Valor_Saturacion_Vo.Caption = "Salida
saturada"

ElseIf Vo <= VsatNeg Then

    RestadorAltaImp.Valor_Vo.Caption = VsatNeg
    RestadorAltaImp.Valor_Saturacion_Vo.Caption = "Salida
saturada"

Else

    RestadorAltaImp.Valor_Saturacion_Vo.Caption = ""
    RestadorAltaImp.Valor_Vo.Caption = Vo

End If

End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0
```



End Sub

Sub RestaurarValores2 ()

```
'Vaciamos los valores de cada uno de los parámetros'  
RestadorAltaImp.Valor_R1.Text = ""  
RestadorAltaImp.Valor_R2.Text = ""  
RestadorAltaImp.Valor_R4.Text = ""  
RestadorAltaImp.Valor_R5.Text = ""  
RestadorAltaImp.Valor_Vi1.Text = ""  
RestadorAltaImp.Valor_Vi2.Text = ""  
RestadorAltaImp.Valor_VsatPos.Text = ""  
RestadorAltaImp.Valor_VsatNeg.Text = ""  
RestadorAltaImp.Valor_R3.Caption = ""  
RestadorAltaImp.Valor_R6.Caption = ""  
RestadorAltaImp.Valor_Ganancia_Vol.Caption = ""  
RestadorAltaImp.Valor_Ganancia_Vil.Caption = ""  
RestadorAltaImp.Valor_Ganancia_Vi2.Caption = ""  
RestadorAltaImp.Valor_Vo1.Caption = ""  
RestadorAltaImp.Valor_Vo.Caption = ""  
RestadorAltaImp.Valor_Saturacion_Vol.Caption = ""  
RestadorAltaImp.Valor_Saturacion_Vo.Caption = ""
```

End Sub

7.2.3.21. Circuitos sumadores (ModSumador)

Sub MostrarSumadorInversor ()

```
'Mostramos la pestaña de la aplicación del circuito sumador inversor'  
SumadorInversor.Show
```

End Sub

Sub MostrarTeorial ()

```
'Mostramos la pestaña de la teoría del circuito sumador inversor'  
TeoriaSumInv.Show
```

End Sub

Sub MostrarInformacion1 ()

```
'Mostramos la pestaña de la información del circuito sumador inversor'  
Informacion1.Show
```

End Sub

Sub CalcularValores1 ()

```
'Declaramos las variables de los parámetros del circuito sumador  
inversor'  
Dim R1 As Single  
Dim R2 As Single
```



```
Dim R3 As Single
Dim Vi1 As Single
Dim Vi2 As Single
Dim VsatPos As Single
Dim VsatNeg As Single
Dim R4 As Single
Dim Ganancia_Vi1 As Single
Dim Ganancia_Vi2 As Single
Dim Vo As Single

'Mediante esta declaración conseguimos que nuestra macro no pare de
ejecutarse aunque se encuentre un error'
On Error Resume Next

'Nos aseguramos de que se introduzca un valor en todas las casillas'
'En el caso de las resistencias, su valor no puede ser cero o
negativo'
'En el caso de las tensiones, su valor si puede ser cero o negativo'
If SumadorInversor.Valor_R1.Text <= 0 Or SumadorInversor.Valor_R2.Text
<= 0 Or SumadorInversor.Valor_R3.Text <= 0 Or
SumadorInversor.Valor_R1.Text = Empty Or SumadorInversor.Valor_R2.Text
= Empty Or SumadorInversor.Valor_R3.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas de las
resistencias y que no sea ni nulo ni negativo", vbExclamation +
vbOKOnly, "AVISO"

    SumadorInversor.Valor_R4.Caption = ""
    SumadorInversor.Valor_Ganancia_Vi1.Caption = ""
    SumadorInversor.Valor_Ganancia_Vi2.Caption = ""
    SumadorInversor.Valor_Vo.Caption = ""
    SumadorInversor.Valor_Saturacion.Caption = ""

ElseIf SumadorInversor.Valor_Vi1.Text = Empty Or
SumadorInversor.Valor_Vi2.Text = Empty Or
SumadorInversor.Valor_VsatPos.Text = Empty Or
SumadorInversor.Valor_VsatNeg.Text = Empty Then

    MsgBox "Debe introducir un valor en las casillas del Voltaje",
vbExclamation + vbOKOnly, "AVISO"

    SumadorInversor.Valor_R4.Caption = ""
    SumadorInversor.Valor_Ganancia_Vi1.Caption = ""
    SumadorInversor.Valor_Ganancia_Vi2.Caption = ""
    SumadorInversor.Valor_Vo.Caption = ""
    SumadorInversor.Valor_Saturacion.Caption = ""

Else

    'Asignamos las variables a los correspondientes parámetros del
circuito sumador inversor'
    R1 = SumadorInversor.Valor_R1.Text * 1000
    R2 = SumadorInversor.Valor_R2.Text * 1000
    R3 = SumadorInversor.Valor_R3.Text * 1000
    Vi1 = SumadorInversor.Valor_Vi1.Text
    Vi2 = SumadorInversor.Valor_Vi2.Text
    VsatPos = SumadorInversor.Valor_VsatPos.Text
    VsatNeg = SumadorInversor.Valor_VsatNeg.Text
```



```
'Introducimos la condición de que -Vsat NO puede ser mayor que  
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación  
de Vo'
```

```
If VsatPos < VsatNeg Then
```

```
MsgBox "La tensión de saturación Vsat no puede ser mayor que  
+Vsat", vbInformation + vbOKOnly, "AVISO"
```

```
SumadorInversor.Valor_VsatPos.Text = ""  
SumadorInversor.Valor_VsatNeg.Text = ""  
SumadorInversor.Valor_R4.Caption = ""  
SumadorInversor.Valor_Ganancia_Vi1.Caption = ""  
SumadorInversor.Valor_Ganancia_Vi2.Caption = ""  
SumadorInversor.Valor_Vo.Caption = ""  
SumadorInversor.Valor_Saturacion.Caption = ""
```

```
Else
```

```
'Definimos las ecuaciones de los parametros a calcular y  
mostramos el resultado'
```

```
R4 = ((R1 * R2 * R3) / (R3 * (R2 + R1) + (R1 * R2))) / 1000  
Ganancia_Vi1 = -(R3 / R1)  
Ganancia_Vi2 = -(R3 / R2)  
Vo = (Ganancia_Vi1 * Vi1 + Ganancia_Vi2 * Vi2)
```

```
SumadorInversor.Valor_R4.Caption = R4  
SumadorInversor.Valor_Ganancia_Vi1.Caption = Ganancia_Vi1  
SumadorInversor.Valor_Ganancia_Vi2.Caption = Ganancia_Vi2  
SumadorInversor.Valor_Vo.Caption = Vo
```

```
'Analizamos si la tensión de salida está saturada'
```

```
If Vo >= VsatPos Then
```

```
SumadorInversor.Valor_Vo.Caption = VsatPos  
SumadorInversor.Valor_Saturacion.Caption = "Salida  
saturada"
```

```
ElseIf Vo <= VsatNeg Then
```

```
SumadorInversor.Valor_Vo.Caption = VsatNeg  
SumadorInversor.Valor_Saturacion.Caption = "Salida  
saturada"
```

```
Else
```

```
SumadorInversor.Valor_Saturacion.Caption = ""  
SumadorInversor.Valor_Vo.Caption = Vo
```

```
End If
```

```
End If
```

```
End If
```

```
'Desactivamos el detector de errores'
```

```
On Error GoTo 0
```

```
End Sub
```

```
Sub RestaurarValores1 ()
```



```
'Vaciamos los valores de cada uno de los parámetros'  
SumadorInversor.Valor_R1.Text = ""  
SumadorInversor.Valor_R2.Text = ""  
SumadorInversor.Valor_R3.Text = ""  
SumadorInversor.Valor_Vi1.Text = ""  
SumadorInversor.Valor_Vi2.Text = ""  
SumadorInversor.Valor_VsatPos.Text = ""  
SumadorInversor.Valor_VsatNeg.Text = ""  
SumadorInversor.Valor_R4.Caption = ""  
SumadorInversor.Valor_Ganancia_Vi1.Caption = ""  
SumadorInversor.Valor_Ganancia_Vi2.Caption = ""  
SumadorInversor.Valor_Vo.Caption = ""  
SumadorInversor.Valor_Saturacion.Caption = ""  
  
End Sub  
  
Sub MostrarSumadorNoInversor ()  
  
'Mostramos la pestaña de la aplicación del circuito sumador inversor'  
SumadorNoInversor.Show  
  
End Sub  
  
Sub MostrarTeoria2 ()  
  
'Mostramos la pestaña de la teoría del circuito sumador inversor'  
TeoriaSumNoInv.Show  
  
End Sub  
  
Sub MostrarInformacion2 ()  
  
'Mostramos la pestaña de la información del circuito sumador no  
inversor'  
Informacion1.Show  
  
End Sub  
  
Sub CalcularValores2 ()  
  
'Declaramos las variables de los parámetros del circuito sumador no  
inversor'  
Dim R1 As Single  
Dim R2 As Single  
Dim R3 As Single  
Dim R4 As Single  
Dim Vi1 As Single  
Dim Vi2 As Single  
Dim VccPos As Single  
Dim VccNeg As Single  
Dim Ganancia_Vi1 As Single  
Dim Ganancia_Vi2 As Single  
Dim Vo As Single  
  
'Mediante esta declaración conseguimos que nuestra macro no pare de  
ejecutarse aunque se encuentre un error'  
On Error Resume Next  
  
'Nos aseguramos de que se introduzca un valor en todas las casillas'  
'En el caso de las resistencias, su valor no puede ser cero o  
negativo'
```



```
'En el caso de las tensiones, su valor si puede ser cero o negativo'  
If SumadorNoInversor.Valor_R1.Text <= 0 Or  
SumadorNoInversor.Valor_R2.Text <= 0 Or  
SumadorNoInversor.Valor_R3.Text <= 0 Or  
SumadorNoInversor.Valor_R4.Text <= 0 Or  
SumadorNoInversor.Valor_R1.Text = Empty Or  
SumadorNoInversor.Valor_R2.Text = Empty Or  
SumadorNoInversor.Valor_R3.Text = Empty Or  
SumadorNoInversor.Valor_R4.Text = Empty Then    MsgBox "Debe introducir un valor en las casillas de las  
resistencias y que no sea ni nulo ni negativo", vbExclamation +  
vbOKOnly, "AVISO"    SumadorNoInversor.Valor_Ganancia_Vi1.Caption = ""  
    SumadorNoInversor.Valor_Ganancia_Vi2.Caption = ""  
    SumadorNoInversor.Valor_Vo.Caption = ""  
    SumadorNoInversor.Valor_Saturacion.Caption = ""ElseIf SumadorNoInversor.Valor_Vi1.Text = Empty Or  
SumadorNoInversor.Valor_Vi2.Text = Empty Or  
SumadorNoInversor.Valor_VsatPos.Text = Empty Or  
SumadorNoInversor.Valor_VsatNeg.Text = Empty Then    MsgBox "Debe introducir un valor en las casillas del Voltaje",  
vbExclamation + vbOKOnly, "AVISO"    SumadorNoInversor.Valor_Ganancia_Vi1.Caption = ""  
    SumadorNoInversor.Valor_Ganancia_Vi2.Caption = ""  
    SumadorNoInversor.Valor_Vo.Caption = ""  
    SumadorNoInversor.Valor_Saturacion.Caption = ""Else    'Asignamos las variables a los correspondientes parámetros del  
circuito sumador no inversor'  
    R1 = SumadorNoInversor.Valor_R1.Text * 1000  
    R2 = SumadorNoInversor.Valor_R2.Text * 1000  
    R3 = SumadorNoInversor.Valor_R3.Text * 1000  
    R4 = SumadorNoInversor.Valor_R4.Text * 1000  
    Vi1 = SumadorNoInversor.Valor_Vi1.Text  
    Vi2 = SumadorNoInversor.Valor_Vi2.Text  
    VsatPos = SumadorNoInversor.Valor_VsatPos.Text  
    VsatNeg = SumadorNoInversor.Valor_VsatNeg.Text    'Introducimos la condición de que -Vsat NO puede ser mayor que  
+Vsat, ya que si se pudiese dar el caso, podría alterar la saturación  
de Vo'  
    If VsatPos < VsatNeg Then        MsgBox "La tensión de saturación -Vsat no puede ser mayor que  
+Vsat", vbInformation + vbOKOnly, "AVISO"        SumadorNoInversor.Valor_VsatPos.Text = ""  
        SumadorNoInversor.Valor_VsatNeg.Text = ""  
        SumadorNoInversor.Valor_Ganancia_Vi1.Caption = ""  
        SumadorNoInversor.Valor_Ganancia_Vi2.Caption = ""  
        SumadorNoInversor.Valor_Vo.Caption = ""  
        SumadorNoInversor.Valor_Saturacion.Caption = ""    Else
```



```
'Definimos las ecuaciones de los parametros a calcular y
mostramos el resultado'
Ganancia_Vi1 = ((R1 + R2) / R1) * (R4 / (R3 + R4))
Ganancia_Vi2 = ((R1 + R2) / R1) * (R3 / (R3 + R4))
Vo = (Ganancia_Vi1 * Vi1 + Ganancia_Vi2 * Vi2)

SumadorNoInversor.Valor_Ganancia_Vi1.Caption = Ganancia_Vi1
SumadorNoInversor.Valor_Ganancia_Vi2.Caption = Ganancia_Vi2
SumadorNoInversor.Valor_Vo.Caption = Vo

'Analizamos si la tensión de salida está saturada'
If Vo >= VsatPos Then

    SumadorNoInversor.Valor_Vo.Caption = VsatPos
    SumadorNoInversor.Valor_Saturacion.Caption = "Salida
saturada"

ElseIf Vo <= VsatNeg Then

    SumadorNoInversor.Valor_Vo.Caption = VsatNeg
    SumadorNoInversor.Valor_Saturacion.Caption = "Salida
saturada"

Else

    SumadorNoInversor.Valor_Saturacion.Caption = ""
    SumadorNoInversor.Valor_Vo.Caption = Vo

End If

End If

End If

'Desactivamos el detector de errores'
On Error GoTo 0

End Sub

Sub RestaurarValores2 ()

'Vaciamos los valores de cada uno de los parámetros'
SumadorNoInversor.Valor_R1.Text = ""
SumadorNoInversor.Valor_R2.Text = ""
SumadorNoInversor.Valor_R3.Text = ""
SumadorNoInversor.Valor_R4.Text = ""
SumadorNoInversor.Valor_Vi1.Text = ""
SumadorNoInversor.Valor_Vi2.Text = ""
SumadorNoInversor.Valor_VsatPos.Text = ""
SumadorNoInversor.Valor_VsatNeg.Text = ""
SumadorNoInversor.Valor_Ganancia_Vi1.Caption = ""
SumadorNoInversor.Valor_Ganancia_Vi2.Caption = ""
SumadorNoInversor.Valor_Vo.Caption = ""
SumadorNoInversor.Valor_Saturacion.Caption = ""

End Sub
```