



**Universidad de Valladolid**

**E.T.S Ingeniería Informática**

**TRABAJO FIN DE GRADO**

Grado en Ingeniería Informática  
Mención en Tecnologías de la Información

**AudiOS:  
ESTUDIOS DE ONDAS SONORAS Y  
AUDIOMETRÍAS EN ENTORNO IOS**

Autora:  
**María Merino de Vega**





**Universidad de Valladolid**

# **E.T.S Ingeniería Informática**

**TRABAJO FIN DE GRADO**

Grado en Ingeniería Informática  
Mención en Tecnologías de la Información

## **AudiOS: ESTUDIOS DE ONDAS SONORAS Y AUDIOMETRÍAS EN ENTORNO IOS**

Autora:  
**María Merino de Vega**

Tutores:  
**Joaquín Adiego Rodríguez  
Miguel Ángel González Rebollo**



## Resumen

---

El objetivo de este proyecto es el desarrollo y adaptación de una aplicación en entorno iOS que permita la utilización de un iPad o un iPhone para el estudio de las ondas sonoras. De esa manera se conseguirá disponer de un laboratorio móvil y de bajo coste para la realización de una serie de experimentos relacionados con las propiedades más importantes del sonido.

La aplicación permitirá la realización de test de barridos, el estudio de diferentes propiedades del sonido, superposición de ondas de igual frecuencia en tubos acústicos, tanto abiertos como cerrados, batidos, efecto Doppler, interferencia y difracción de ondas, medida de la velocidad del sonido, etc... y servirá de apoyo a los alumnos de la Universidad de Valladolid.

Además la aplicación integrará una serie de herramientas: sonómetro, grabadora de sonido, la posibilidad de visualizar el espectro de frecuencias de la señal percibida así como la de generar las diferentes ondas sonoras (sinusoidal, cuadrada, triangular, diente de sierra, ruido blanco, ruido rosa y ruido marrón), y una herramienta de calibración del propio dispositivo.

## Abstract

---

The objective of this project is the development and adaptation of an application in iOS environment that allows the use of an iPad or iPhone for the study of sound waves. Thus will be achieved have a mobile laboratory and inexpensive to carry out a series of experiments related to the most important properties of sound.

The application allows the realization of test scans, the study of various properties of sound, superposition of waves of the same frequency in acoustic tubes, both open and closed, shakes, Doppler, interference and diffraction of waves, measure the speed of sound effect etc ... and will support students of the University of Valladolid.

Also, the application will integrate a number of tools: sound meter, sound recorder, the ability to display the frequency spectrum of the perceived signal and to generate the different sound waves (sine, square, triangle, sawtooth, white noise, pink noise and brown noise), and a calibration tool device itself.



# Tabla de Contenidos

---

<b>1.- INTRODUCCIÓN .....</b>	<b>1</b>
1.1.- MOTIVACIÓN.....	1
1.2.- OBJETIVOS .....	2
1.3.- METODOLOGÍA.....	2
1.4.- RESUMEN MEMORIA .....	3
<b>2.- PLANIFICACIÓN Y GESTIÓN DE PROYECTO.....</b>	<b>5</b>
2.1.- PLANIFICACIÓN .....	6
2.1.1.- <i>Planificación Inicial</i> .....	7
2.1.1.- <i>Modificación Planificación Inicial</i> .....	9
2.1.3.- <i>Recursos</i> .....	10
2.1.4.- <i>Gestión de Riesgos</i> .....	10
2.2.- ANÁLISIS .....	12
2.2.1.- <i>Actores</i> .....	12
2.2.2.- <i>Requisitos</i> .....	12
2.2.2.1.- <i>Requisitos Funcionales</i> .....	12
2.2.2.2.- <i>Requisitos No Funcionales</i> .....	15
2.2.3.- <i>Diagrama de Casos de Uso</i> .....	16
2.2.4.- <i>Modelo de Dominio</i> .....	25
2.3.- DISEÑO .....	26
2.3.1.- <i>Diagramas de Secuencia</i> .....	26
2.3.2.- <i>Diagramas de Clases</i> .....	37
2.3.3.- <i>Clases Destalladas</i> .....	38
<b>3.- SISTEMA IOS .....</b>	<b>43</b>
3.1.- CARACTERÍSTICAS .....	43
3.2.- ARQUITECTURA .....	45
3.3.- DISPOSITIVOS.....	46
3.4.- SEGURIDAD .....	48
3.5.- IOS VS ANDROID.....	49
3.6.- XCODE .....	50
3.7.- LENGUAJES DE PROGRAMACIÓN IOS.....	52
3.8.- PERSONALIZACIÓN DE DISPOSITIVOS APPLE Y JAILBREAK.....	55
<b>4.- FUNDAMENTOS TEÓRICOS .....</b>	<b>57</b>
4.1.- ONDAS .....	57
4.1.1.- <i>Ondas Sonoras</i> .....	57
4.1.2.- <i>Ondas Estacionarias</i> .....	59
4.3.- DECIBELIOS.....	59
4.4.- TRANSFORMADA DE FOURIER.....	62
4.4.1.- <i>Ventanas</i> .....	62
4.5.- CALIBRACIÓN .....	67
4.6.- FORMATOS PARA GUARDAR .....	67
<b>5.- IMPLEMENTACIÓN.....</b>	<b>69</b>
<b>6.- PRUEBAS .....</b>	<b>71</b>

<b>7.- CONCLUSIÓN .....</b>	<b>75</b>
7.1.- CONCLUSIONES .....	75
7.2.- LÍNEAS DE TRABAJO FUTURO .....	75
<b>I ANEXO.- INSTALACIÓN .....</b>	<b>79</b>
<b>II ANEXO.- MANUAL DE USUARIO .....</b>	<b>81</b>
II.I.- HERRAMIENTAS .....	82
II.I.I- <i>Sonómetro</i> .....	83
II.I.II- <i>Espectro de Frecuencias</i> .....	83
II.I.III- <i>Generador Onda Sonora</i> .....	85
II.I.IV- <i>Grabadora de Sonido</i> .....	86
II.II.- EXPERIENCIAS .....	87
II.II.I- <i>Test Barrido de Frecuencias</i> .....	88
II.II.II- <i>Ondas Estacionarias</i> .....	90
II.II.III- <i>Efecto Doppler</i> .....	93
II.II.IV- <i>Batidos</i> .....	94
II.II.V- <i>Interferencias y Difracción</i> .....	95
II.III.- CALIBRACIÓN .....	95
II.III.I- <i>Calibración Micrófono</i> .....	96
II.III.II- <i>Calibración Altavoces</i> .....	98
II.IV.- CONFIGURACIÓN .....	99
II.V.- DATOS NUMÉRICOS .....	100
<b>BIBLIOGRAFÍA .....</b>	<b>103</b>





## Lista de Figuras

---

ILUSTRACIÓN 1: SISTEMAS OPERATIVOS DE MÓVILES (JUNIO 2015).....	1
ILUSTRACIÓN 2: SISTEMAS OPERATIVOS DE MÓVILES (JUNIO 2016).....	2
ILUSTRACIÓN 3: CICLO DE VIDA (VISIÓN ITERATIVA).....	5
ILUSTRACIÓN 4: ACTIVIDADES DE CICLO DE VIDA CON ITERACIONES.....	6
ILUSTRACIÓN 5: PLANIFICACIÓN FASE DE INICIO.....	7
ILUSTRACIÓN 6: PLANIFICACIÓN FASE DE ELABORACIÓN.....	7
ILUSTRACIÓN 7: PLANIFICACIÓN FASE DE CONSTRUCCIÓN.....	8
ILUSTRACIÓN 8: PLANIFICACIÓN FASE DE TRANSICIÓN.....	8
ILUSTRACIÓN 9: PLANIFICACIÓN FASE DE CONSTRUCCIÓN MODIFICADA.....	9
ILUSTRACIÓN 10: PLANIFICACIÓN FASE DE TRANSICIÓN NUEVA.....	9
ILUSTRACIÓN 11: DIAGRAMA DE CASOS DE USO.....	16
ILUSTRACIÓN 12: MODELO DE DOMINIO.....	25
ILUSTRACIÓN 13: DIAGRAMA DE SECUENCIA - VISUALIZAR SONÓMETRO.....	26
ILUSTRACIÓN 14: DIAGRAMA DE SECUENCIA - VISUALIZAR ESPECTRO FRECUENCIAS.....	27
ILUSTRACIÓN 15: DIAGRAMA DE SECUENCIA - GENERAR ONDA SONORA.....	27
ILUSTRACIÓN 16: DIAGRAMA DE SECUENCIA - GUARDAR ONDA SONORA.....	28
ILUSTRACIÓN 17: DIAGRAMA DE SECUENCIA - GRABAR SONIDO.....	28
ILUSTRACIÓN 18: DIAGRAMA DE SECUENCIA - REALIZAR TEST BARRIDO FRECUENCIAS.....	29
ILUSTRACIÓN 19: DIAGRAMA DE SECUENCIA - VISUALIZAR TEST BARRIDO FRECUENCIAS.....	30
ILUSTRACIÓN 20: DIAGRAMA DE SECUENCIA: MODIFICAR TEST BARRIDO FRECUENCIAS.....	30
ILUSTRACIÓN 21: DIAGRAMA DE SECUENCIA - BORRAR TEST BARRIDO FRECUENCIAS.....	31
ILUSTRACIÓN 22: DIAGRAMA DE SECUENCIA - REALIZAR ONDAS ESTACIONARIAS FIJO.....	31
ILUSTRACIÓN 23: DIAGRAMA DE SECUENCIA - REALIZAR ONDA ESTACIONARIA VARIABLE.....	32
ILUSTRACIÓN 24: DIAGRAMA DE SECUENCIA - REALIZAR EFECTO DOPPLER.....	32
ILUSTRACIÓN 25: DIAGRAMA DE SECUENCIA - REALIZAR BATIDOS.....	33
ILUSTRACIÓN 26: DIAGRAMA DE SECUENCIA - REALIZAR INTERFERENCIAS Y DIFRACCIÓN.....	33
ILUSTRACIÓN 27: DIAGRAMA DE SECUENCIA - REALIZAR CALIBRACIÓN MICRÓFONO.....	34
ILUSTRACIÓN 28: DIAGRAMA DE SECUENCIA - REALIZAR CALIBRACIÓN ALTAVOZ.....	35
ILUSTRACIÓN 29: DIAGRAMA DE SECUENCIA - MODIFICAR CONFIGURACIÓN.....	35
ILUSTRACIÓN 30: DIAGRAMA DE SECUENCIA - VISUALIZAR RESULTADO GUARDADO.....	36
ILUSTRACIÓN 31: DIAGRAMA DE SECUENCIA - BORRAR RESULTADO GUARDADO.....	36
ILUSTRACIÓN 32: DIAGRAMA DE PAQUETES.....	37
ILUSTRACIÓN 33: DIAGRAMA DE CLASES.....	37
ILUSTRACIÓN 34: CLASE SONÓMETRO.....	38
ILUSTRACIÓN 35: CLASE ESPECTRÓMETRO.....	38
ILUSTRACIÓN 36: CLASE GENERADOR ONDA SONORA.....	39
ILUSTRACIÓN 37: CLASE GENERADOR SONIDO.....	39
ILUSTRACIÓN 38: CLASE GRABADORA.....	39
ILUSTRACIÓN 39: CLASE TEST BARRIDO FRECUENCIAS.....	40
ILUSTRACIÓN 40: CLASE ONDAS ESTACIONARIAS FIJO.....	40
ILUSTRACIÓN 41: CLASE ONDAS ESTACIONARIAS VARIABLE.....	40
ILUSTRACIÓN 42: CLASE EFECTO DOPPLER.....	41
ILUSTRACIÓN 43: CLASE BATIDOS.....	41
ILUSTRACIÓN 44: CLASE INTERFERENCIAS Y DIFRACCIÓN.....	41
ILUSTRACIÓN 45: CLASE CALIBRACIÓN MICRÓFONO.....	41
ILUSTRACIÓN 46: CLASE CALIBRACIÓN ALTAVOZ.....	42
ILUSTRACIÓN 47: CLASE CONFIGURACIÓN.....	42
ILUSTRACIÓN 48: CLASE DATOS NUMÉRICOS.....	42

ILUSTRACIÓN 49: PANTALLA DE BLOQUEO/DESBLOQUEO	ILUSTRACIÓN 50: PANTALLA PRINCIPAL .....	44
ILUSTRACIÓN 51: AGRUPACIÓN DE ICONOS EN CARPETAS .....		44
ILUSTRACIÓN 52: CENTRO DE NOTIFICACIONES.....		44
ILUSTRACIÓN 53: CENTRO DE CONTROL .....		45
ILUSTRACIÓN 54: ARQUITECTURA IOS.....		45
ILUSTRACIÓN 55: ARQUITECTURA DE LA SEGURIDAD IOS .....		48
ILUSTRACIÓN 56: FRAGMENTACIÓN DE DISPOSITIVOS (AGOSTO 2015).....		50
ILUSTRACIÓN 57: VENTANA XCODE .....		51
ILUSTRACIÓN 58: RANKING LENGUAJES DE PROGRAMACIÓN .....		53
ILUSTRACIÓN 59: GRÁFICA EVOLUCIÓN OBJECTIVE-C .....		53
ILUSTRACIÓN 60: GRÁFICA EVOLUCIÓN SWIFT .....		54
ILUSTRACIÓN 61: CICLO DE VIDA DE UN VIEW CONTROLLER.....		54
ILUSTRACIÓN 62: ONDAS SONORAS .....		58
ILUSTRACIÓN 63: VALORES DE LA LONGITUD DE UNA ONDA ESTACIONARIA .....		59
ILUSTRACIÓN 64: PRIMERAS CURVAS ISOFÓNICAS ESTABLECIDAS POR FLETCHER Y MUNSON EN 1930 .....		61
ILUSTRACIÓN 65: CURVAS DE PONDERACIÓN A, B Y C .....		61
ILUSTRACIÓN 66: VENTANA RECTANGULAR .....		63
ILUSTRACIÓN 67: VENTANA BARTLETT O TRIANGULAR .....		63
ILUSTRACIÓN 68: VENTANA HAMMING.....		64
ILUSTRACIÓN 69: VENTANA HANNING .....		64
ILUSTRACIÓN 70: VENTANA BLACKMAN .....		65
ILUSTRACIÓN 71: VENTANA BLACKMAN-HARRIS .....		65
ILUSTRACIÓN 72: VENTANA GAUSS .....		66
ILUSTRACIÓN 73: VENTANA FLAT TOP .....		66
ILUSTRACIÓN 74: MUESTREO EN CÓDIGO PCM .....		68
ILUSTRACIÓN 75: APLICACIÓN APP STORE .....		79
ILUSTRACIÓN 76: DESCARGA DE XCODE DESDE APP STORE.....		79
ILUSTRACIÓN 77: PROYECTO AUDIOS .....		80
ILUSTRACIÓN 78: DETECCIÓN DEL DISPOSITIVO EN XCODE .....		80
ILUSTRACIÓN 79: SELECCIÓN DE DISPOSITIVO Y EJECUCIÓN EN EL MISMO.....		80
ILUSTRACIÓN 80: PANTALLA DE INICIO .....		81
ILUSTRACIÓN 81: PANTALLA DE AYUDA.....		81
ILUSTRACIÓN 82: PANTALLA PRINCIPAL .....		82
ILUSTRACIÓN 83: PANTALLA HERRAMIENTAS .....		82
ILUSTRACIÓN 84: PANTALLA SONÓMETRO.....		83
ILUSTRACIÓN 85: PANTALLA ESPECTRO DE FRECUENCIAS .....		84
ILUSTRACIÓN 86: PANTALLA OPCIONES ESPECTRO DE FRECUENCIAS .....		84
ILUSTRACIÓN 87: PANTALLA ESPECTRO DE FRECUENCIAS CON INFORMACIÓN .....		85
ILUSTRACIÓN 88: PANTALLA GENERADOR ONDA SONORA .....		85
ILUSTRACIÓN 89: PANTALLA GRABADORA DE SONIDO .....		86
ILUSTRACIÓN 90: PANTALLA SEÑAL CAPTURADA .....		86
ILUSTRACIÓN 91: PANTALLA EXPERIENCIAS.....		87
ILUSTRACIÓN 92: PANTALLA TEST BARRIDO DE FRECUENCIAS.....		88
ILUSTRACIÓN 93: PANTALLA INICIO TEST .....		88
ILUSTRACIÓN 94: PANTALLA EJECUCIÓN TEST.....		89
ILUSTRACIÓN 95: PANTALLA TEST GUARDADOS .....		89
ILUSTRACIÓN 96: PANTALLA ELIMINAR TEST GUARDADO .....		90
ILUSTRACIÓN 97: PANTALLA ONDAS ESTACIONARIAS .....		90
ILUSTRACIÓN 98: PANTALLA ONDA ESTACIONARIA DE LONGITUD DE TUBO FIJO .....		91
ILUSTRACIÓN 99: PANTALLA ONDA ESTACIONARIA DE LONGITUD DE TUBO VARIABLE .....		92
ILUSTRACIÓN 100: PANTALLA EFECTO DOPPLER .....		93
ILUSTRACIÓN 101: PANTALLA BATIDOS.....		94

ILUSTRACIÓN 102: PANTALLA INTERFERENCIAS Y DIFRACCIÓN. ....	95
ILUSTRACIÓN 103: PANTALLA CALIBRACIÓN .....	96
ILUSTRACIÓN 104: PANTALLA CALIBRACIÓN MICRÓFONO .....	96
ILUSTRACIÓN 105: PANTALLA PROCESO CALIBRACIÓN MICRÓFONO .....	97
ILUSTRACIÓN 106: PANTALLA RESULTADOS CALIBRACIÓN MICRÓFONO .....	97
ILUSTRACIÓN 107: PANTALLA CALIBRACIÓN ALTAVOCES.....	98
ILUSTRACIÓN 108: ASISTENTE DE LA CALIBRACIÓN DE ALTAVOCES.....	98
ILUSTRACIÓN 109: PANTALLA PROCESO CALIBRACIÓN ALTAVOZ.....	99
ILUSTRACIÓN 110: PANTALLA CONFIGURACIÓN.....	99
ILUSTRACIÓN 111: PANTALLA DATOS NUMÉRICOS .....	100
ILUSTRACIÓN 112: PANTALLA ELIMINAR DATOS NUMÉRICOS.....	101



## Lista de Tablas

---

TABLA 1: RIESGO - 0001.....	10
TABLA 2: RIESGO - 0002.....	10
TABLA 3: RIESGO - 0003.....	11
TABLA 4: RIESGO - 0004.....	11
TABLA 5: RIESGO - 0005.....	11
TABLA 6: RIESGO - 0006.....	11
TABLA 7: RIESGO - 0007.....	11
TABLA 8: REQUISITO FUNCIONAL - 0001.....	12
TABLA 9: REQUISITO FUNCIONAL - 0002.....	12
TABLA 10: REQUISITO FUNCIONAL - 0003.....	12
TABLA 11: REQUISITO FUNCIONAL - 0004.....	13
TABLA 12: REQUISITO FUNCIONAL - 0005.....	13
TABLA 13: REQUISITO FUNCIONAL - 0006.....	13
TABLA 14: REQUISITO FUNCIONAL - 0007.....	13
TABLA 15: REQUISITO FUNCIONAL - 0008.....	13
TABLA 16: REQUISITO FUNCIONAL - 0009.....	14
TABLA 17: REQUISITO FUNCIONAL - 0010.....	14
TABLA 18: REQUISITO FUNCIONAL - 0011.....	14
TABLA 19: REQUISITO FUNCIONAL - 0012.....	14
TABLA 20: REQUISITO FUNCIONAL - 0013.....	14
TABLA 21: REQUISITO FUNCIONAL - 0014.....	15
TABLA 22: REQUISITO NO FUNCIONAL - 0001.....	15
TABLA 23: REQUISITO NO FUNCIONAL - 0002.....	15
TABLA 24: REQUISITO NO FUNCIONAL - 0003.....	15
TABLA 25: CASO DE USO - 0001.....	17
TABLA 26: CASO DE USO - 0002.....	17
TABLA 27: CASO DE USO - 0003.....	17
TABLA 28: CASO DE USO - 0004.....	18
TABLA 29: CASO DE USO - 0005.....	18
TABLA 30: CASO DE USO - 0006.....	19
TABLA 31: CASO DE USO - 0007.....	19
TABLA 32: CASO DE USO - 0008.....	20
TABLA 33: CASO DE USO - 0009.....	20
TABLA 34: CASO DE USO - 0010.....	21
TABLA 35: CASO DE USO - 0011.....	21
TABLA 36: CASO DE USO - 0012.....	22
TABLA 37: CASO DE USO - 0013.....	22
TABLA 38: CASO DE USO - 0014.....	22
TABLA 39: CASO DE USO - 0015.....	23
TABLA 40: CASO DE USO - 0016.....	23
TABLA 41: CASO DE USO - 0017.....	24
TABLA 42: CASO DE USO - 0018.....	24
TABLA 43: CASO DE USO - 0019.....	25
TABLA 44: DISPOSITIVOS APPLE.....	47
TABLA 45: NIVEL DE INTENSIDAD DEL SONIDO.....	60
TABLA 46: CASO DE PRUEBA - 0001.....	71
TABLA 47: CASO DE PRUEBA - 0002.....	71
TABLA 48: CASO DE PRUEBA - 0003.....	71
TABLA 49: CASO DE PRUEBA - 0004.....	71

TABLA 50: CASO DE PRUEBA - 0005.....	71
TABLA 51: CASO DE PRUEBA - 0006.....	71
TABLA 52: CASO DE PRUEBA - 0007.....	72
TABLA 53: CASO DE PRUEBA - 0008.....	72
TABLA 54: CASO DE PRUEBA - 0009.....	72
TABLA 55: CASO DE PRUEBA - 0010.....	72
TABLA 56: CASO DE PRUEBA - 0011.....	72
TABLA 57: CASO DE PRUEBA - 0012.....	72
TABLA 58: CASO DE PRUEBA - 0013.....	72
TABLA 59: CASO DE PRUEBA - 0014.....	73
TABLA 60: CASO DE PRUEBA - 0015.....	73
TABLA 61: CASO DE PRUEBA - 0016.....	73
TABLA 62: CASO DE PRUEBA - 0017.....	73
TABLA 63: CASO DE PRUEBA - 0018.....	73
TABLA 64: CASO DE PRUEBA - 0019.....	73
TABLA 65: CASO DE PRUEBA - 0020.....	73
TABLA 66: CASO DE PRUEBA - 0021.....	74





# 1.- INTRODUCCIÓN

Este proyecto es generado por la diversidad de sistemas operativos que nos encontramos en los dispositivos móviles que tienen los estudiantes de la Universidad de Valladolid y por ello es necesario dar una igualdad de posibilidades a todos los usuarios sin discriminación por el Smartphone que se tenga. El proyecto tiene como objetivo crear un laboratorio móvil y de bajo coste utilizando su propio dispositivo móvil para estudiar las ondas sonoras. Ya existe la aplicación AudiA para dispositivos Android, con lo cual aquí se va a llevar a cabo la creación de la aplicación AudiOS (Audi de Audiometría e iOS del sistema operativo).

## 1.1.- Motivación

La motivación que he tenido a lo largo de este proyecto es la superación y aprender un nuevo lenguaje de manera autodidacta que me serviría para un futuro. Nunca había realizado una app para dispositivos de Apple ya que en las asignaturas del Grado se hicieron apps para dispositivos Android, lo cual esto suponía otro reto.

La programación para iOS se puede realizar en Objective-C o Swift, a pesar de haber programado en C tenía que aprender un lenguaje de programación y elegí Swift. Swift es un lenguaje nuevo creado por Apple que me parece más intuitivo y fácil de aprender. Aunque siendo un lenguaje tan joven surgía el inconveniente de que si me encontraba con alguna duda o problema fuese más difícil encontrar soluciones.

Hoy en día el uso del móvil está muy extendido y para la mayoría de personas su uso en el día a día es indispensable. Por ello es muy útil poder programar para los dos sistemas operativos principales en el mundo, a continuación se muestran dos gráficas, con diferencia de un año, de diferentes páginas web que verifican que Android e iOS son los principales sistemas operativos en dispositivos móviles.

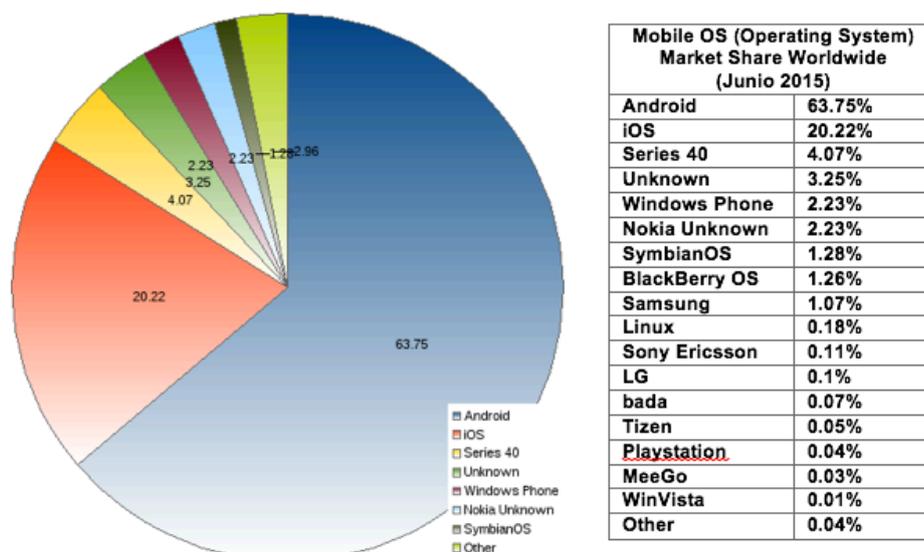


Ilustración 1: Sistemas Operativos de móviles (Junio 2015)

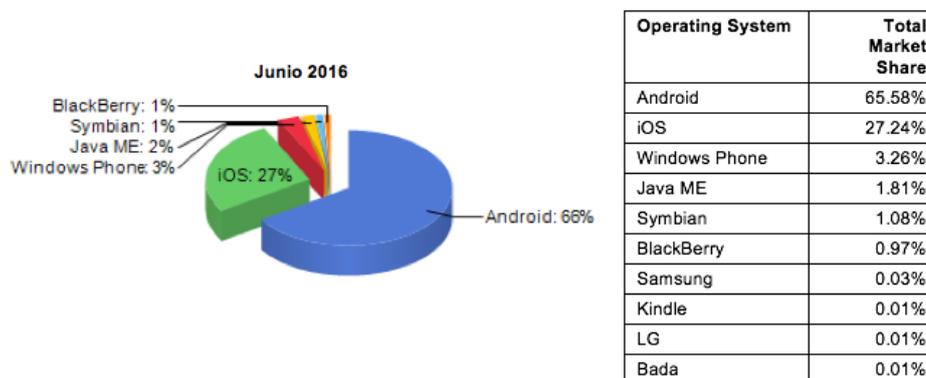


Ilustración 2: Sistemas Operativos de móviles (Junio 2016)

## 1.2.- Objetivos

El objetivo de este proyecto es que con recursos que la mayoría de la gente tiene y usa a diario, como el caso del Smartphone o Tablet, puedan tener un laboratorio fácil de transportar y de bajo coste.

El principal objetivo es crear una aplicación para dispositivos Apple como la aplicación AudiA, que ya existe pero sólo se puede utilizar en dispositivos Android.

Con esto se va a llevar a que estudiantes, principalmente porque es para quien está pensada esta aplicación, puedan realizar experiencias con el sonido y que eso les ayude en una mejora de sus conocimientos y una mayor facilidad del aprendizaje sin la necesidad de un gran laboratorio. A parte de experiencias, AudiOS les proporciona diferentes herramientas con las que podrán generar ondas y visualizar sus gráficas entre otras utilidades.

## 1.3.- Metodología

La ingeniería de software nos proporciona diferentes modelos que nos facilitan el desarrollo del software porque nos guía por diferentes etapas para conseguir el objetivo de conseguir el proyecto en el tiempo estipulado.

El proceso a utilizar va a ser el Proceso Unificado. Según sus desarrolladores es más que un proceso, es un marco de trabajo genérico que puede especializarse para gran cantidad de sistemas de software, diferentes áreas de aplicación, diferentes tipos de organizaciones, etc.

Me ha parecido más apropiado este proceso de desarrollo de software porque es iterativo e incremental, lo que conlleva a disminuir los riesgos más críticos y tener una mayor flexibilidad para posibles cambios. Otra ventaja de usar este proceso es que al usuario se le puede mostrar algo del proyecto con anterioridad y con ello se incrementa la implicación del usuario para un mejor resultado conociendo sus necesidades y opiniones del transcurso del proyecto.

## 1.4.- Resumen Memoria

A lo largo de esta memoria se va a mostrar con detalle la planificación y diseño que se ha llevado a cabo (Capítulo 2). En el capítulo posterior se va a información sobre el sistema operativo para el que se va a programar la aplicación, así como el entorno Xcode y los lenguajes de programación que podía elegir para realizar AudiOS. También expongo los datos e información teórica en el capítulo 4 que me han ayudado a desarrollar y entender la parte física.

En los capítulos 5 y 6 se detallará información de la implementación y las pruebas que se han realizado para determinar que la aplicación funciona correctamente. Y para finalizar con los capítulos se han añadido las conclusiones obtenidas, futuras mejoras y la bibliografía utilizada.

Por último se ha añadido dos anexos con la manera de instalar la App en nuestros dispositivos y el manual de usuario de la aplicación.



## 2.- PLANIFICACIÓN Y GESTIÓN DE PROYECTO

La metodología a usar como ya se ha explicado anteriormente va a ser el proceso unificado que está basado en componentes software que se conectan mediante interfaces, que es la manera que se va a trabajar con nuestra aplicación. El proceso unificado utiliza UML como lenguaje de modelado.

El proceso unificado se puede resumir en 3 elementos claves, que son:

**Dirigido por los casos de uso:** los casos de uso se utilizan para obtener los requisitos funcionales y para definir los contenidos de las iteraciones.

**Centrado en la arquitectura:** visualizar, especificar, construir y documentar un sistema software precisa una visión del sistema desde distintos puntos de vista. Cada actor implicado en el proceso mira al sistema desde una perspectiva diferente a lo largo de la vida del proyecto. La arquitectura del sistema es uno de los productos más importantes que se pueden utilizar para gestionar estos puntos de vista y, por tanto, para controlar el desarrollo del sistema a través de su ciclo de vida.

**Iterativo e incremental:** se compone de 4 fases a lo largo del ciclo de vida del proyecto. Inicio que define el ámbito del proyecto y desarrolla los casos de negocio. Elaboración que es donde se elabora el plan de proyecto, se especifican las características y el marco de referencia de la arquitectura. Construcción en la cual se construye el proyecto. Transición donde se obtiene el producto final y se pasa al usuario para que lo instale y lo pruebe.



Ilustración 3: Ciclo de vida (visión iterativa)

El ciclo de vida se divide en dos etapas, la etapa de ingeniería donde se realizan las actividades de diseño y síntesis, y la etapa de producción donde se realizan las actividades de construcción, prueba y lanzamiento. La etapa de ingeniería abarca la fase de inicio y elaboración, mientras que la etapa de producción contiene la fase de construcción y transición.

Al finalizar cada una de las fases del ciclo de vida hay un hito en el que se obtendrá un resultado que se añadirá al producto final. En el hito de la fase de inicio se obtiene el plan de proyecto, las características principales, los recursos y una identificación de los riesgos. En el hito de la fase de elaboración se definirán los actores, los requisitos funcionales y no funcionales, el modelo de casos de uso y se iniciará el proyecto. En el hito de la fase de construcción se obtiene el producto final. En el hito de la transición se deberá probar el producto final con manual para que el usuario pueda empezar a utilizarlo.

Durante el ciclo de vida se realizan una serie de actividades que en unas fases se realizan mayor porcentaje de unas u otras. A su vez cada fase puede estar dividida en iteraciones en las que se desarrollan diferentes trabajos.

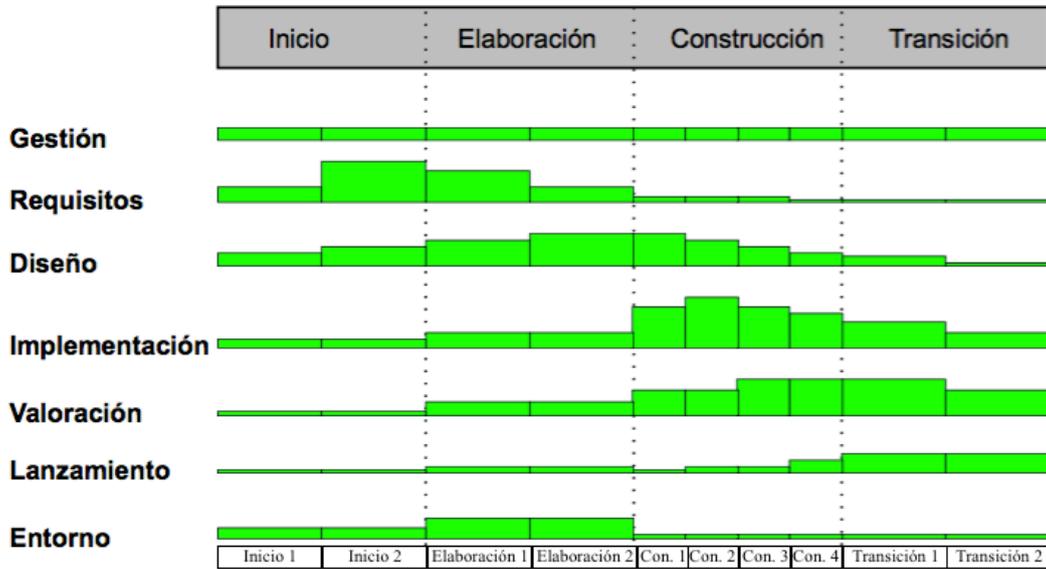


Ilustración 4: Actividades de ciclo de vida con iteraciones

Como se puede observar en la anterior imagen la repartición de actividades a lo largo del ciclo de vida corresponde con los resultados que se deben obtener en la finalización de cada hito en cada fase.

### 2.1.- Planificación

Los proyectos están desarrollados por varias personas con diferentes roles y responsabilidades a lo largo del proyecto. Los roles necesarios en este proyecto son:

**Gestor de Proyecto:** tiene como responsabilidad gestionar el proyecto, deberá comunicarse con el cliente, será el encargado de asignar tareas a los demás trabajadores y será el responsable de que el proyecto funcione como debería y en el tiempo estimado.

**Planificador:** es el responsable de planificar y documentar la planificación del proyecto para su buen desarrollo.

**Analista:** es el responsable de analizar el proyecto, sus requisitos y diseñar una solución.

**Programador:** es el responsable de realizar la implementación con un correcto funcionamiento y cumpliendo las necesidades del proyecto.

**Testeador:** es el responsable en probar el producto y verificar que todo está bien para poder entregar el producto al usuario.

En este proyecto hay una única persona que realiza el mismo y por eso deberá desempeñar todos los roles.

### 2.1.1.- Planificación Inicial

A continuación se muestran las planificaciones para cada fase del proyecto:

	<b>i</b> Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	<b>▲ Fase de Inicio</b>	<b>9 días</b>	<b>lun 09/02/15</b>	<b>jue 19/02/15</b>	
2	<b>▲ Iteración 1</b>	<b>9 días</b>	<b>lun 09/02/15</b>	<b>jue 19/02/15</b>	
3	Lectura del proyecto	1 día	lun 09/02/15	lun 09/02/15	
4	Determinar el alcance del proyecto	3 días	mar 10/02/15	jue 12/02/15	3
5	Elaboración de la planificación inicial	5 días	vie 13/02/15	jue 19/02/15	4
6	Identificación de los recursos	1 hora	vie 13/02/15	vie 13/02/15	4
7	Identificación y análisis de riesgos	3 días	vie 13/02/15	mar 17/02/15	4
8	Identificación de los requisitos	3 días	vie 13/02/15	mar 17/02/15	4
9	Identificación de casos de uso	4 días	vie 13/02/15	mié 18/02/15	4
10	Hito fase de inicio	0 días	jue 19/02/15	jue 19/02/15	2

Ilustración 5: Planificación Fase de Inicio

	<b>i</b> Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
11	<b>▲ Fase de Elaboración</b>	<b>31 días</b>	<b>vie 20/02/15</b>	<b>vie 03/04/15</b>	<b>1</b>
12	<b>▲ Iteración 1</b>	<b>16 días</b>	<b>vie 20/02/15</b>	<b>vie 13/03/15</b>	
13	Identificación de actores	1 día	vie 20/02/15	vie 20/02/15	
14	Documentación de requisitos	6 días	lun 23/02/15	lun 02/03/15	13
15	Documentación de los casos de uso	15 días	lun 23/02/15	vie 13/03/15	13
16	Realización del modelo de dominio	3 días	lun 23/02/15	mié 25/02/15	13
17	Hito iteración 1 de fase de elaboración	0 días	vie 13/03/15	vie 13/03/15	12
18	<b>▲ Iteración 2</b>	<b>15 días</b>	<b>lun 16/03/15</b>	<b>vie 03/04/15</b>	<b>17</b>
19	Realización y documentación de los diagramas de secuencia	15 días	lun 16/03/15	vie 03/04/15	
20	Realización y documentación de los diagramas de clases	8 días	lun 16/03/15	mié 25/03/15	
21	Hito iteración 2 de fase de elaboración	0 días	vie 03/04/15	vie 03/04/15	18

Ilustración 6: Planificación Fase de Elaboración

## 2.- PLANIFICACIÓN Y GESTIÓN DE PROYECTO | AudiOS

	 Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
22	<b>▲ Fase de Construcción</b>	<b>142 días</b>	<b>lun 06/04/15</b>	<b>mar 20/10/15</b>	<b>11</b>
23	<b>▲ Iteración 1</b>	<b>40 días</b>	<b>lun 06/04/15</b>	<b>vie 29/05/15</b>	
24	Estudio de los fundamentos teóricos	3 días	lun 06/04/15	mié 08/04/15	
25	Familiarización con el nuevo lenguaje	5 días	lun 06/04/15	vie 10/04/15	
26	Implementación del proyecto	35 días	lun 13/04/15	vie 29/05/15	25
27	Hito iteración 1 de fase de construcción	0 días	vie 29/05/15	vie 29/05/15	23
28	<b>▲ Iteración 2</b>	<b>43 días</b>	<b>lun 01/06/15</b>	<b>mié 29/07/15</b>	<b>27</b>
29	Revisión de requisitos y casos de uso	1 día	lun 01/06/15	lun 01/06/15	
30	Implementación del proyecto	40 días	mar 02/06/15	lun 27/07/15	29
31	Prueba de lo implementado	2 días	mar 28/07/15	mié 29/07/15	30
32	Hito iteración 2 de fase de construcción	0 días	mié 29/07/15	mié 29/07/15	28
33	<b>▲ Iteración 3</b>	<b>35 días</b>	<b>jue 30/07/15</b>	<b>mié 16/09/15</b>	<b>32</b>
34	Revisión de requisitos y casos de uso	1 día	jue 30/07/15	jue 30/07/15	
35	Implementación del proyecto	30 días	vie 31/07/15	jue 10/09/15	34
36	Prueba de lo implementado	4 días	vie 11/09/15	mié 16/09/15	35
37	Hito iteración 3 de fase de construcción	0 días	mié 16/09/15	mié 16/09/15	33
38	<b>▲ Iteración 4</b>	<b>24 días</b>	<b>jue 17/09/15</b>	<b>mar 20/10/15</b>	<b>37</b>
39	Revisión de requisitos y casos de uso	1 día	jue 17/09/15	jue 17/09/15	
40	Implementación del proyecto	20 días	vie 18/09/15	jue 15/10/15	39
41	Prueba de lo implementado	3 días	vie 16/10/15	mar 20/10/15	40
42	Hito iteración 4 de fase de construcción	0 días	mar 20/10/15	mar 20/10/15	38

Ilustración 7: Planificación Fase de Construcción

	 Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
43	<b>▲ Fase de Transición</b>	<b>35 días</b>	<b>mié 21/10/15</b>	<b>mar 08/12/15</b>	<b>22</b>
44	<b>▲ Iteración 1</b>	<b>13 días</b>	<b>mié 21/10/15</b>	<b>vie 06/11/15</b>	
45	Pruebas del proyecto	10 días	mié 21/10/15	mar 03/11/15	
46	Documentación de pruebas	3 días	mié 04/11/15	vie 06/11/15	45
47	Hito iteración 1 de fase de transición	0 días	vie 06/11/15	vie 06/11/15	44
48	<b>▲ Iteración 2</b>	<b>22 días</b>	<b>lun 09/11/15</b>	<b>mar 08/12/15</b>	
49	Realización de la documentación	20 días	lun 09/11/15	vie 04/12/15	46
50	Revisión de proyecto	1 día	lun 07/12/15	lun 07/12/15	49
51	Entrega de proyecto	1 día	mar 08/12/15	mar 08/12/15	50
52	Hito iteración 2 de fase de transición	0 días	mar 08/12/15	mar 08/12/15	48

Ilustración 8: Planificación Fase de Transición

### 2.1.1.- Modificación Planificación Inicial

Por motivos ajenos al proyecto, este tuvo que ser retrasado durante la fase de construcción y por ello la planificación se modificó obteniendo la siguiente:

	 Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
22	▲ Fase de Construcción	117 días	lun 11/01/16	mar 21/06/16	11
23	▲ Iteración 1	20 días	lun 11/01/16	vie 05/02/16	
24	Revisión del avance del proyecto	3 días	lun 11/01/16	mié 13/01/16	
25	Recordar el lenguaje de programación	5 días	lun 11/01/16	vie 15/01/16	
26	Implementación del proyecto	15 días	lun 18/01/16	vie 05/02/16	25
27	Hito iteración 1 de fase de construcción	0 días	vie 05/02/16	vie 05/02/16	23
28	▲ Iteración 2	38 días	lun 08/02/16	mié 30/03/16	27
29	Revisión de requisitos y casos de uso	1 día	lun 08/02/16	lun 08/02/16	
30	Implementación del proyecto	35 días	mar 09/02/16	lun 28/03/16	29
31	Prueba de lo implementado	2 días	mar 29/03/16	mié 30/03/16	30
32	Hito iteración 2 de fase de construcción	0 días	mié 30/03/16	mié 30/03/16	28
33	▲ Iteración 3	35 días	jue 31/03/16	mié 18/05/16	32
34	Revisión de requisitos y casos de uso	1 día	jue 31/03/16	jue 31/03/16	
35	Implementación del proyecto	30 días	vie 01/04/16	jue 12/05/16	34
36	Prueba de lo implementado	4 días	vie 13/05/16	mié 18/05/16	35
37	Hito iteración 3 de fase de construcción	0 días	mié 18/05/16	mié 18/05/16	33
38	▲ Iteración 4	24 días	jue 19/05/16	mar 21/06/16	37
39	Revisión de requisitos y casos de uso	1 día	jue 19/05/16	jue 19/05/16	
40	Implementación del proyecto	20 días	vie 20/05/16	jue 16/06/16	39
41	Prueba de lo implementado	3 días	vie 17/06/16	mar 21/06/16	40
42	Hito iteración 4 de fase de construcción	0 días	mar 21/06/16	mar 21/06/16	38

Ilustración 9: Planificación Fase de Construcción Modificada

	 Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
43	▲ Fase de Transición	53 días	mié 22/06/16	vie 02/09/16	22
44	▲ Iteración 1	12 días	mié 22/06/16	jue 07/07/16	
45	Pruebas del proyecto	8 días	mié 22/06/16	vie 01/07/16	
46	Documentación de pruebas	4 días	lun 04/07/16	jue 07/07/16	45
47	Hito iteración 1 de fase de transición	0 días	jue 07/07/16	jue 07/07/16	44
48	▲ Iteración 2	41 días	vie 08/07/16	vie 02/09/16	
49	Realización de la documentación	25 días	vie 08/07/16	jue 11/08/16	46
50	Revisión de proyecto	15 días	vie 12/08/16	jue 01/09/16	49
51	Entrega de proyecto	1 día	vie 02/09/16	vie 02/09/16	50
52	Hito iteración 2 de fase de transición	0 días	vie 02/09/16	vie 02/09/16	48

Ilustración 10: Planificación Fase de Transición Nueva

2.1.3.- Recursos

**Recursos Hardware:** para este proyecto se dispone de un ordenador Mac donde se realizará la mayor parte del proyecto y un iPhone que servirá para ir probando la aplicación.

**Recursos Software:** el software necesario será ASTAH para el análisis y diseño del proyecto y el Microsoft Project 2013 para la planificación y seguimiento del proyecto. Para la implementación del proyecto se ha usado la herramienta Xcode.

2.1.4.- Gestión de Riesgos

En todo proyecto siempre suele ocurrir algún imprevisto durante el mismo que nos retrasa el proyecto, por ello hay que detectar esos posibles riesgos e intentar que no ocurran. En caso de que el imprevisto llegue a ocurrir sería bueno tener preparado un plan de contingencia. Por ello a continuación se muestra una serie de riesgos que puede que se den a lo largo del proyecto que se intentarán evitar.

Riesgo - 0001	Mala planificación
Descripción	La planificación inicial es muy justa y no se pueden cumplir los plazos
Efecto	Retraso del proyecto.
Frecuencia	Alta
Gravedad	Alta
Detección	Alta
Acción Correctora	En la planificación inicial se pondrá más tiempo del que se calcule para tener un margen de tiempo.
Plan de contingencia	Aumentar las horas de trabajo.

Tabla 1: Riesgo - 0001

Riesgo - 0002	Poco conocimiento de los fundamentos teóricos
Descripción	Los fundamentos teóricos necesarios para desarrollar la aplicación no son conocidos por el desarrollador
Efecto	Retraso del proyecto.
Frecuencia	Alta
Gravedad	Alta
Detección	Alta
Acción Correctora	Preguntar a un especialista ante cualquier duda que surja.
Plan de contingencia	Una buena formación por parte de especialistas y un eficaz estudio de los fundamentos teóricos.

Tabla 2: Riesgo - 0002

Riesgo - 0003	Poca soltura con el lenguaje de programación
Descripción	Un proyecto nuevo que se realiza para un sistema operativo nuevo y se necesita un lenguaje de programación nuevo.
Efecto	Retraso en las tareas del proyecto.
Frecuencia	Media
Gravedad	Media
Detección	Alta
Acción Correctora	Consultar manuales y cursos de programación

Plan de contingencia	Un buen estudio del lenguaje de programación.
----------------------	---

Tabla 3: Riesgo - 0003

Riesgo - 0004	Cambio de requisitos
Descripción	No se entendieron las necesidades del usuario y hay que cambiar requisitos
Efecto	Retraso del proyecto.
Frecuencia	Media
Gravedad	Media
Detección	Alta
Acción Correctora	Poner atención en las reuniones con el usuario y apuntar todo. Ante cualquier duda por pequeña que sea preguntar para quedar todos los temas bien claros
Plan de contingencia	Aumentar las horas de trabajo.

Tabla 4: Riesgo - 0004

Riesgo - 0005	Acceso a Internet no disponible
Descripción	La red de internet está caída y no se pueden consultar información ni soluciones para posibles problemas de código.
Efecto	Retraso del proyecto.
Frecuencia	Baja
Gravedad	Baja
Detección	Alta
Acción Correctora	Contratar una línea de ADSL fiable.
Plan de contingencia	Usar los datos del móvil.

Tabla 5: Riesgo - 0005

Riesgo - 0006	Fallo en el entorno de desarrollo
Descripción	El entorno de desarrollo tuvo un fallo inesperado y se perdió las últimas modificaciones.
Efecto	Retraso del proyecto.
Frecuencia	Baja
Gravedad	Alta
Detección	Alta
Acción Correctora	En la planificación inicial se pondrá más tiempo del que se calcule para tener un margen de tiempo.
Plan de contingencia	Aumentar las horas de trabajo.

Tabla 6: Riesgo - 0006

Riesgo - 0007	Enfermedad
Descripción	El desarrollador cae enfermo sin poder continuar con el proyecto con normalidad.
Efecto	Retraso del proyecto.
Frecuencia	Baja
Gravedad	Baja
Detección	Baja
Acción Correctora	En la planificación inicial se pondrá más tiempo del que se calcule para tener un margen de tiempo.
Plan de contingencia	Aumentar las horas de trabajo.

Tabla 7: Riesgo - 0007

## 2.2.- Análisis

Se van a exponer los actores que participan en el sistema y que requisitos debe cumplir este. Al igual que como se organiza e interactúa el sistema.

### 2.2.1.- Actores

En la aplicación solo habrá un actor que será el usuario que disponga el dispositivo y maneje la aplicación.

### 2.2.2.- Requisitos

Los requisitos de un sistema son fundamentales para identificar que se puede hacer con la aplicación, como debe responder esta y que restricciones se tiene. Existen dos tipos de requisitos, los requisitos funcionales que son las funciones que realiza la aplicación y los requisitos no funcionales son aquellos que no tienen que ver con el funcionamiento, sino que son restricciones del sistema.

#### 2.2.2.1.- Requisitos Funcionales

Requisito Funcional - 0001	Acceso a la aplicación
Versión	1.0
Autores	María Merino de Vega
Descripción	El sistema deberá dejar acceder a la aplicación a cualquier usuario que la inicie.
Importancia	Vital

Tabla 8: Requisito Funcional - 0001

Requisito Funcional - 0002	Acceder a las diferentes partes de la aplicación
Versión	1.0
Autores	María Merino de Vega
Descripción	El sistema deberá recoger lo que el usuario le indica y llevarle a la parte de la aplicación que desee
Importancia	Vital

Tabla 9: Requisito Funcional - 0002

Requisito Funcional - 0003	Uso del sonómetro
Versión	1.0
Autores	María Merino de Vega
Descripción	El sistema permitirá usar al usuario sin ninguna dificultad el sonómetro, pudiendo variar parámetros, guardar valores y enviar por correo
Importancia	Vital

Tabla 10: Requisito Funcional - 0003

Requisito Funcional - 0004	Uso del espectro de frecuencias
Versión	1.0
Autores	María Merino de Vega
Descripción	El sistema permitirá visualizar el espectro de frecuencias generado, pudiendo variar parámetros, guardar valores y enviar por correo.
Importancia	Vital

Tabla 11: Requisito Funcional - 0004

Requisito Funcional - 0005	Generador de sonido
Versión	1.0
Autores	María Merino de Vega
Descripción	El sistema permitirá al usuario elegir los parámetros que desee, validándolos y generando el sonido elegido por el usuario. También permitirá el guardado y envío del sonido generado.
Importancia	Vital

Tabla 12: Requisito Funcional - 0005

Requisito Funcional - 0006	Grabar
Versión	1.0
Autores	María Merino de Vega
Descripción	El sistema permitirá grabar el sonido que el usuario desee y al terminar mostrará las gráficas dejando cambiar al usuario cambiar parámetros y dándole la posibilidad de guardado y envío.
Importancia	Vital

Tabla 13: Requisito Funcional - 0006

Requisito Funcional - 0007	Realizar test de barrido
Versión	1.0
Autores	María Merino de Vega
Descripción	El sistema deberá permitir que el usuario realice el test de barrido de frecuencias y que el resultado pueda ser guardado
Importancia	Vital

Tabla 14: Requisito Funcional - 0007

Requisito Funcional - 0008	Generar ondas estacionarias
Versión	1.0
Autores	María Merino de Vega
Descripción	El sistema permitirá al usuario elegir la onda y los parámetros que desee, validándolos y generando el sonido. También permitirá el guardado y envío del sonido generado.
Importancia	Vital

Tabla 15: Requisito Funcional - 0008

## 2.- PLANIFICACIÓN Y GESTIÓN DE PROYECTO | AudiOS

Requisito Funcional - 0009	Generar efecto doppler
Versión	1.0
Autores	María Merino de Vega
Descripción	El sistema permitirá al usuario elegir los parámetros que desee, validándolos y generando el sonido. También permitirá el guardado y envío del sonido generado.
Importancia	Vital

Tabla 16: Requisito Funcional - 0009

Requisito Funcional - 0010	Generar batidos
Versión	1.0
Autores	María Merino de Vega
Descripción	El sistema permitirá al usuario elegir los parámetros que desee, validándolos y generando el sonido. También permitirá el guardado y envío del sonido generado.
Importancia	Vital

Tabla 17: Requisito Funcional - 0010

Requisito Funcional - 0011	Generar interferencias y difracciones
Versión	1.0
Autores	María Merino de Vega
Descripción	El sistema permitirá al usuario elegir los parámetros que desee, validándolos y generando el sonido. También permitirá el guardado y envío del sonido generado.
Importancia	Vital

Tabla 18: Requisito Funcional - 0011

Requisito Funcional - 0012	Calibración del dispositivo
Versión	1.0
Autores	María Merino de Vega
Descripción	El sistema permitirá al usuario realizar las calibraciones de los altavoces y micrófono del dispositivo para un mejor funcionamiento de la aplicación.
Importancia	Vital

Tabla 19: Requisito Funcional - 0012

Requisito Funcional - 0013	Personalizar configuración
Versión	1.0
Autores	María Merino de Vega
Descripción	El sistema permitirá al usuario elegir los parámetros y que estos se puedan guardar para poder ser usados por las diferentes herramientas de la aplicación.
Importancia	Vital

Tabla 20: Requisito Funcional - 0013

Requisito Funcional - 0014	Visualizar datos
Versión	1.0
Autores	María Merino de Vega
Descripción	El sistema permitirá al usuario ver o escuchar los datos que se guarden en las diferentes experiencias o herramientas de la aplicación.
Importancia	Vital

Tabla 21: Requisito Funcional - 0014

#### 2.2.2.2.- Requisitos No Funcionales

Requisito No Funcional - 0001	Ejecución
Versión	1.0
Autores	María Merino de Vega
Descripción	El sistema deberá ejecutarse en sistema iOS
Importancia	Vital

Tabla 22: Requisito No Funcional - 0001

Requisito No Funcional - 0002	Facilidad de uso
Versión	1.0
Autores	María Merino de Vega
Descripción	El sistema deberá ser intuitivo y fácil de entender y manejar por el usuario.
Importancia	Vital

Tabla 23: Requisito No Funcional - 0002

Requisito No Funcional - 0003	Fiable y robusto
Versión	1.0
Autores	María Merino de Vega
Descripción	El sistema deberá funcionar correctamente y sin fallos.
Importancia	Vital

Tabla 24: Requisito No Funcional - 0003

2.2.3.- Diagrama de Casos de Uso

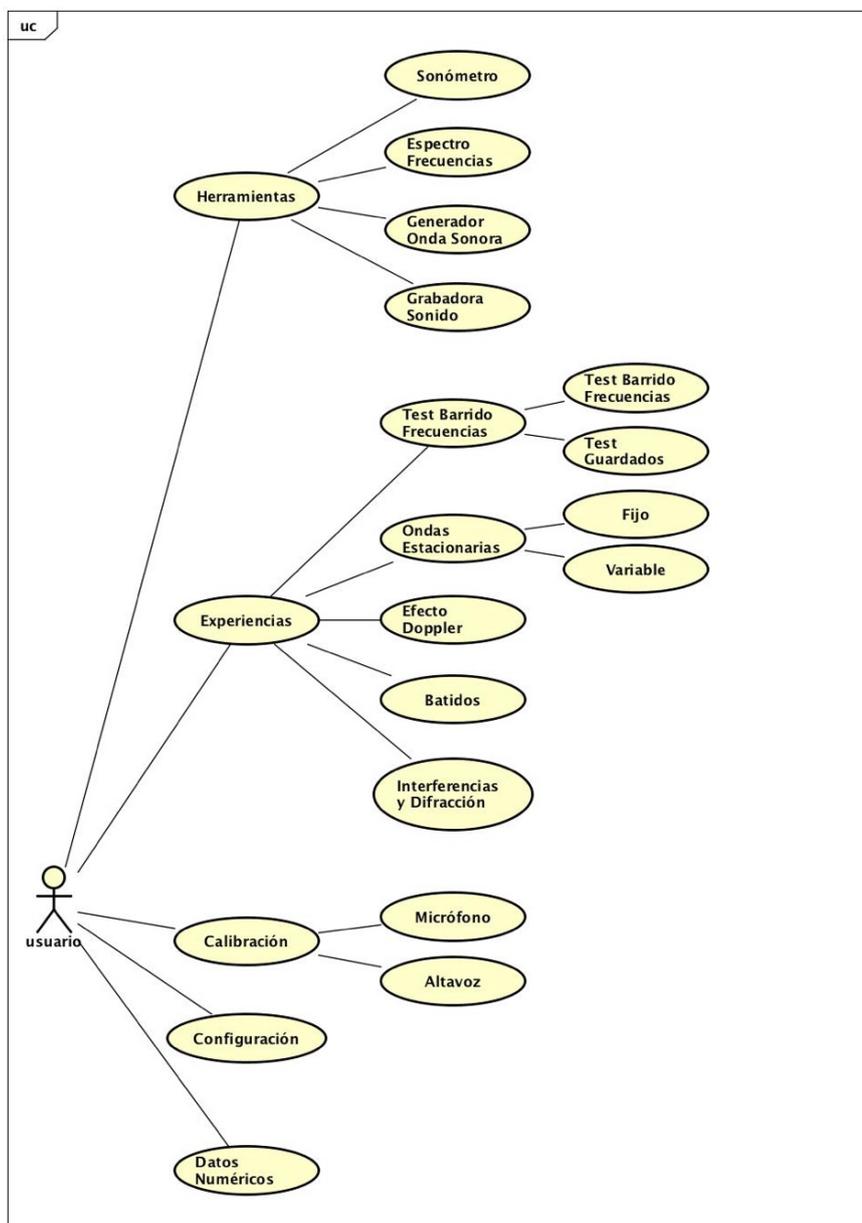


Ilustración 11: Diagrama de casos de uso

Caso de Uso - 0001	Visualizar sonómetro	
Versión	1.0	
Autores	María Merino de Vega	
Descripción	Mostrar RMS y Pico de la señal capturada.	
Precondición	<i>-ninguna-</i>	
Secuencia normal	Paso	Acción
	1	El usuario pulsará el botón herramientas de la pantalla principal de la aplicación.

	2	El sistema mostrará la pantalla de las herramientas.
	3	El usuario pulsará el botón sonómetro.
	4	El sistema mostrará la pantalla del sonómetro, empezará a capturar la señal e irá actualizando la vista.
Postcondición	<i>-ninguna-</i>	
Excepciones	Paso	Acción
	1	
Importancia	Vital	

Tabla 25: Caso de uso – 0001

Caso de Uso - 0002	Visualizar espectro frecuencias	
Versión	1.0	
Autores	María Merino de Vega	
Descripción	Mostrar el espectro de frecuencias de la señal capturada.	
Precondición	<i>-ninguna-</i>	
Secuencia normal	Paso	Acción
	1	El usuario pulsará el botón herramientas de la pantalla principal de la aplicación.
	2	El sistema mostrará la pantalla de las herramientas.
	3	El usuario pulsará el botón espectro frecuencias.
	4	El sistema mostrará la pantalla del espectro de frecuencias, empezará a capturar la señal e irá actualizando la vista.
Postcondición	<i>-ninguna-</i>	
Excepciones	Paso	Acción
	1	
Importancia	Vital	

Tabla 26: Caso de uso - 0002

Caso de Uso - 0003	Generar Onda Sonora	
Versión	1.0	
Autores	María Merino de Vega	
Descripción	Genera una onda sonora con los parámetros elegidos.	
Precondición	Sonido del dispositivo activado.	
Secuencia normal	Paso	Acción
	1	El usuario pulsará el botón herramientas de la pantalla principal de la aplicación.
	2	El sistema mostrará la pantalla de las herramientas.
	3	El usuario pulsará el botón generador de onda sonora.
	4	El sistema mostrará la pantalla del generador de onda sonora.
	5	El usuario elegirá los parámetros que desee y dará al botón de generar sonido
	6	El sistema reproducirá el sonido seleccionado y configurado.
Postcondición	<i>-ninguna-</i>	
Excepciones	Paso	Acción
	5	El usuario introduce mal los datos y el sistema les corrige por unos correctos.
Importancia	Vital	

Tabla 27: Caso de uso - 0003

## 2.- PLANIFICACIÓN Y GESTIÓN DE PROYECTO | AudiOS

Caso de Uso - 0004	Guardar Onda Sonora	
Versión	1.0	
Autores	María Merino de Vega	
Descripción	Guarda una onda sonora con los parámetros elegidos.	
Precondición	<i>-ninguna-</i>	
Secuencia normal	Paso	Acción
	1	El usuario pulsará el botón herramientas de la pantalla principal de la aplicación.
	2	El sistema mostrará la pantalla de las herramientas.
	3	El usuario pulsará el botón generador de onda sonora.
	4	El sistema mostrará la pantalla del generador de onda sonora.
	5	El usuario elegirá los parámetro que desee y dará al botón de guardar sonido.
6	El sistema guardará el sonido seleccionado y configurado, dando la opción de enviar.	
Postcondición	<i>-ninguna-</i>	
Excepciones	Paso	Acción
	5	El usuario introduce mal los datos y el sistema les corrige por unos correctos.
Importancia	Vital	

Tabla 28: Caso de uso - 0004

Caso de Uso - 0005	Grabar sonido	
Versión	1.0	
Autores	María Merino de Vega	
Descripción	Grabar sonido en diferentes formatos.	
Precondición	<i>-ninguna-</i>	
Secuencia normal	Paso	Acción
	1	El usuario pulsará el botón herramientas de la pantalla principal de la aplicación.
	2	El sistema mostrará la pantalla de las herramientas.
	3	El usuario pulsará el botón grabar sonido.
	4	El sistema mostrará la pantalla del grabador de sonido.
	5	El usuario elegirá el formato de grabación y dará a grabar.
	6	El sistema grabará en el formato indicado y esperará a la terminación.
	7	El usuario dará al botón parar cuando desee finalizar la grabación.
8	El sistema mostrará una gráfica de lo grabado.	
Postcondición	<i>-ninguna-</i>	
Excepciones	Paso	Acción
	1	
Importancia	Vital	

Tabla 29: Caso de uso - 0005

Caso de Uso - 0006	Realizar test de barrido de frecuencias	
Versión	1.0	
Autores	María Merino de Vega	
Descripción	Realizar un test de barrido de frecuencias para ver hasta que frecuencia somos capaces de escuchar.	
Precondición	Sonido del dispositivo activado y volumen al 100%.	
Secuencia normal	Paso	Acción
	1	El usuario pulsará el botón experiencias de la pantalla principal de la aplicación.
	2	El sistema mostrará la pantalla de las experiencias.

	3	El usuario pulsará el botón test barrido frecuencias.
	4	El sistema mostrará la pantalla de test barrido frecuencias.
	5	El usuario pulsará el botón test barrido frecuencias.
	6	El sistema mostrará la pantalla de altavoz test.
	7	El usuario elegirá sobre que altavoz o altavoces se realizará el test y pulsará el botón aceptar.
	8	El sistema mostrará la pantalla de inicio de test y comenzará a reproducir las frecuencias esperando una señal del usuario para que le indique si ya no oye la frecuencia.
	9	El usuario cuando no escuche deberá pulsar el altavoz para pasar al siguiente paso.
	10	El sistema mostrará los resultados del test.
Postcondición	<i>-ninguna-</i>	
Excepciones	Paso	Acción
	8 y 9	El sistema mostrará el siguiente paso y el usuario seguirá pulsando el botón dependiendo de los altavoces escogidos. Si se escoge altavoz izquierdo o derecho solo serán dos pasos, sin embargo si son ambos altavoces serán 4 pasos.
	11	El usuario podrá guardar los resultados con el nombre que elija.
Importancia	Vital	

Tabla 30: Caso de uso - 0006

Caso de Uso - 0007	Visualizar test de barrido de frecuencias	
Versión	1.0	
Autores	María Merino de Vega	
Descripción	Ver un test de barrido de frecuencias guardado.	
Precondición	El test debe existir.	
Secuencia normal	Paso	Acción
	1	El usuario pulsará el botón experiencias de la pantalla principal de la aplicación.
	2	El sistema mostrará la pantalla de las experiencias.
	3	El usuario pulsará el botón test barrido frecuencias.
	4	El sistema mostrará la pantalla de test barrido frecuencias.
	5	El usuario pulsará el botón test guardados
	6	El sistema mostrará una lista de test de barrido de frecuencias guardados.
	7	El usuario dará en un test.
8	El sistema mostrará la pantalla de resultados de test.	
Postcondición	<i>-ninguna-</i>	
Excepciones	Paso	Acción
	1, 2, 3, 4 y 5	El usuario pulsará el botón datos numéricos de la pantalla principal de la aplicación.
	6	El sistema mostrará una lista de resultados guardados.
Importancia	Vital	

Tabla 31: Caso de uso - 0007

Caso de Uso - 0008	Modificar test de barrido de frecuencias	
Versión	1.0	
Autores	María Merino de Vega	
Descripción	Modificar el nombre de un test de barrido de frecuencias guardado.	

## 2.- PLANIFICACIÓN Y GESTIÓN DE PROYECTO | AudiOS

Precondición	El test a modificar exista.	
Secuencia normal	Paso	Acción
	1	El usuario pulsará el botón experiencias de la pantalla principal de la aplicación.
	2	El sistema mostrará la pantalla de las experiencias.
	3	El usuario pulsará el botón test barrido frecuencias.
	4	El sistema mostrará la pantalla de test barrido frecuencias.
	5	El usuario pulsará el botón test guardados
	6	El sistema mostrará una lista de test de barrido de frecuencias guardados.
	7	El usuario dará en un test.
	8	El sistema mostrará la pantalla de resultados de test.
	9	El usuario cambiará el nombre del test y dará al botón editar.
10	El sistema mostrará el test con el nuevo nombre.	
Postcondición	<i>-ninguna-</i>	
Excepciones	Paso	Acción
	1, 2, 3, 4 y 5	El usuario pulsará el botón datos numéricos de la pantalla principal de la aplicación.
	6	El sistema mostrará una lista de resultados guardados.
	10	El sistema avisa al usuario y no realiza la edición si ya existe un test con el nombre al que se quiere cambiar.
Importancia	Vital	

Tabla 32: Caso de uso - 0008

Caso de Uso - 0009	Borrar test de barrido de frecuencias	
Versión	1.0	
Autores	María Merino de Vega	
Descripción	Borrar un test de barrido de frecuencias guardado.	
Precondición	El test a borrar exista.	
Secuencia normal	Paso	Acción
	1	El usuario pulsará el botón experiencias de la pantalla principal de la aplicación.
	2	El sistema mostrará la pantalla de las experiencias.
	3	El usuario pulsará el botón test barrido frecuencias.
	4	El sistema mostrará la pantalla de test barrido frecuencias.
	5	El usuario pulsará el botón test guardados
	6	El sistema mostrará una lista de test de barrido de frecuencias guardados.
	7	El usuario deslizará el dedo de derecha a izquierda sobre el test deseado.
	8	El sistema desplazará esa celda de la lista y mostrará un botón de eliminar.
	9	El usuario dará al botón eliminar.
10	El sistema eliminará el test.	
Postcondición	<i>-ninguna-</i>	
Excepciones	Paso	Acción
	1, 2, 3, 4 y 5	El usuario pulsará el botón datos numéricos de la pantalla principal de la aplicación.
	6	El sistema mostrará una lista de resultados guardados.
Importancia	Vital	

Tabla 33: Caso de uso - 0009

Caso de Uso - 0010	Realizar onda estacionaria fijo	
Versión	1.0	
Autores	María Merino de Vega	
Descripción	Realizar ondas estacionarias de longitud de tubo fijo.	
Precondición	Sonido del dispositivo activado.	
Secuencia normal	Paso	Acción
	1	El usuario pulsará el botón experiencias de la pantalla principal de la aplicación.
	2	El sistema mostrará la pantalla de las experiencias.
	3	El usuario pulsará el botón ondas estacionarias.
	4	El sistema mostrará la pantalla de ondas estacionarias.
	5	El usuario pulsará el botón fijo.
	6	El sistema mostrará la pantalla de ondas estacionarias de tubo fijo.
	7	El usuario elegirá los parámetro que desee y dará al botón generar sonido.
8	El sistema reproducirá la onda y al finalizar mostrará una gráfica de lo grabado.	
Postcondición	<i>-ninguna-</i>	
Excepciones	Paso	Acción
	7	El usuario introduce mal los datos y el sistema les corrige por unos correctos.
Importancia	Vital	

Tabla 34: Caso de uso - 0010

Caso de Uso - 0011	Realizar onda estacionaria variable	
Versión	1.0	
Autores	María Merino de Vega	
Descripción	Realizar ondas estacionarias de longitud de tubo variable.	
Precondición	Sonido del dispositivo activado.	
Secuencia normal	Paso	Acción
	1	El usuario pulsará el botón experiencias de la pantalla principal de la aplicación.
	2	El sistema mostrará la pantalla de las experiencias.
	3	El usuario pulsará el botón ondas estacionarias.
	4	El sistema mostrará la pantalla de ondas estacionarias.
	5	El usuario pulsará el botón variable.
	6	El sistema mostrará la pantalla de ondas estacionarias de tubo variable.
	7	El usuario elegirá los parámetro que desee y dará al botón generar sonido.
8	El sistema reproducirá la onda y al finalizar mostrará una gráfica de lo grabado.	
Postcondición	<i>-ninguna-</i>	
Excepciones	Paso	Acción
	7	El usuario introduce mal los datos y el sistema les corrige por unos correctos.
Importancia	Vital	

Tabla 35: Caso de uso - 0011

Caso de Uso - 0012	Realizar efecto doppler	
Versión	1.0	
Autores	María Merino de Vega	
Descripción	Realizar sonido del efecto doppler.	
Precondición	Sonido del dispositivo activado.	
Secuencia normal	Paso	Acción
	1	El usuario pulsará el botón experiencias de la pantalla principal de la aplicación.
	2	El sistema mostrará la pantalla de las experiencias.

## 2.- PLANIFICACIÓN Y GESTIÓN DE PROYECTO | AudiOS

	3	El usuario pulsará el botón efecto doppler.
	4	El sistema mostrará la pantalla de efecto doppler..
	5	El usuario elegirá los parámetro que desee y dará al botón generar sonido.
	6	El sistema reproducirá la onda y al finalizar mostrará una gráfica de lo grabado.
Postcondición	<i>-ninguna-</i>	
Excepciones	Paso	Acción
	5	El usuario introduce mal los datos y el sistema les corrige por unos correctos.
Importancia	Vital	

Tabla 36: Caso de uso - 0012

Caso de Uso - 0013	Realizar batidos	
Versión	1.0	
Autores	María Merino de Vega	
Descripción	Realizar sonido de batidos.	
Precondición	Sonido del dispositivo activado.	
Secuencia normal	Paso	Acción
	1	El usuario pulsará el botón experiencias de la pantalla principal de la aplicación.
	2	El sistema mostrará la pantalla de las experiencias.
	3	El usuario pulsará el botón batidos.
	4	El sistema mostrará la pantalla de batidos.
	5	El usuario elegirá los parámetro que desee y dará al botón generar sonido.
	6	El sistema reproducirá la onda y al finalizar mostrará una gráfica de lo grabado.
Postcondición	<i>-ninguna-</i>	
Excepciones	Paso	Acción
	5	El usuario introduce mal los datos y el sistema les corrige por unos correctos.
Importancia	Vital	

Tabla 37: Caso de uso - 0013

Caso de Uso - 0014	Realizar interferencias y difracción	
Versión	1.0	
Autores	María Merino de Vega	
Descripción	Realizar sonido de interferencias y difracción.	
Precondición	Sonido del dispositivo activado.	
Secuencia normal	Paso	Acción
	1	El usuario pulsará el botón experiencias de la pantalla principal de la aplicación.
	2	El sistema mostrará la pantalla de las experiencias.
	3	El usuario pulsará el botón interferencias y difracción.
	4	El sistema mostrará la pantalla de interferencias y difracción.
	5	El usuario elegirá los parámetro que desee y dará al botón generar sonido.
	6	El sistema reproducirá la onda y al finalizar mostrará una gráfica de lo grabado.
Postcondición	<i>-ninguna-</i>	
Excepciones	Paso	Acción
	5	El usuario introduce mal los datos y el sistema les corrige por unos correctos.
Importancia	Vital	

Tabla 38: Caso de uso - 0014

Caso de Uso - 0015	Realizar calibración de micrófono	
Versión	1.0	
Autores	María Merino de Vega	
Descripción	Calibrar el micrófono del dispositivo.	
Precondición	<i>-ninguna-</i>	
Secuencia normal	Paso	Acción
	1	El usuario pulsará el botón calibración de la pantalla principal de la aplicación.
	2	El sistema mostrará la pantalla de las calibraciones.
	3	El usuario pulsará el botón micrófono.
	4	El sistema mostrará la pantalla de calibración de micrófono.
	5	El usuario elegirá los parámetro que desee y dará al botón comenzar.
	6	El sistema comenzará a detectar decibelios esperando al usuario que pare la calibración.
	7	El usuario dará al botón parar para finalizar la calibración.
	8	El sistema mostrará la pantalla de datos calibración.
	9	El usuario pondrá los datos oportunos y dará al botón guardar.
10	El sistema guardará la nueva calibración.	
Postcondición	<i>-ninguna-</i>	
Excepciones	Paso	Acción
	8a	El sistema mostrará la pantalla de datos calibración vacía en caso de ser la primera vez que se calibra el micro.
	8b	El sistema mostrará la pantalla de datos calibración con los datos de la calibración ya existente.
Importancia	Vital	

Tabla 39: Caso de uso - 0015

Caso de Uso - 0016	Realizar calibración de altavoz	
Versión	1.0	
Autores	María Merino de Vega	
Descripción	Calibrar el altavoz del dispositivo.	
Precondición	Debe de existir calibración de micrófono.	
Secuencia normal	Paso	Acción
	1	El usuario pulsará el botón calibración de la pantalla principal de la aplicación.
	2	El sistema mostrará la pantalla de las calibraciones.
	3	El usuario pulsará el botón altavoz.
	4	El sistema mostrará la pantalla de calibración de altavoz.
	5	El usuario dará al botón comenzar asistente.
	6	El sistema comenzará a reproducir frecuencias y a detectar los decibelios.
	7	El usuario dará al botón siguiente cuando el sistema termine de calibrar el altavoz izquierdo
8	El sistema comenzará a reproducir frecuencias, a detectar los decibelios y finalizará la calibración.	
Postcondición	<i>-ninguna-</i>	
Excepciones	Paso	Acción
	1	
Importancia	Vital	

Tabla 40: Caso de uso - 0016

## 2.- PLANIFICACIÓN Y GESTIÓN DE PROYECTO | AudiOS

Caso de Uso - 0017	Modificar configuración	
Versión	1.0	
Autores	María Merino de Vega	
Descripción	Modificar la configuración de la aplicación.	
Precondición	<i>-ninguna-</i>	
Secuencia normal	Paso	Acción
	1	El usuario pulsará el botón configuración de la pantalla principal de la aplicación.
	2	El sistema mostrará la pantalla de la configuración.
	3	El usuario elegirá los parámetros que desee y dará al botón aceptar.
	4	El sistema guardará la nueva configuración.
Postcondición	<i>-ninguna-</i>	
Excepciones	Paso	Acción
	3	El usuario introduce mal los datos y el sistema les corrige por unos correctos.
Importancia	Vital	

Tabla 41: Caso de uso - 0017

Caso de Uso - 0018	Visualizar resultado guardado	
Versión	1.0	
Autores	María Merino de Vega	
Descripción	Ver un resultado guardado.	
Precondición	El resultado debe existir.	
Secuencia normal	Paso	Acción
	1	El usuario pulsará el botón datos numéricos de la pantalla principal de la aplicación.
	2	El sistema mostrará una lista de resultados guardados.
	7	El usuario dará en un test.
	8	El sistema mostrará la pantalla de resultados.
Postcondición	<i>-ninguna-</i>	
Excepciones	Paso	Acción
	8a	El sistema mostrará la pantalla de reproducción para ficheros .wav
	8b	El sistema mostrará la pantalla de datos para ficheros .csv o txt y dependiendo de que experiencia o herramienta se trate la pantalla variará.
Importancia	Vital	

Tabla 42: Caso de uso - 0018

Caso de Uso - 0019	Borrar resultado guardado	
Versión	1.0	
Autores	María Merino de Vega	
Descripción	Borrar un resultado guardado.	
Precondición	El resultado a borrar exista.	
Secuencia normal	Paso	Acción
	1	El usuario pulsará el botón datos numéricos de la pantalla principal de la aplicación.
	2	El sistema mostrará una lista de resultados guardados.
	7	El usuario deslizará el dedo de derecha a izquierda sobre el resultado deseado.
	8	El sistema desplazará esa celda de la lista y mostrará un botón de eliminar.
	9	El usuario dará al botón eliminar.
	10	El sistema eliminará el fichero.

Postcondición	<i>-ninguna-</i>	
Excepciones	Paso	Acción
	1	
Importancia	Vital	

Tabla 43: Caso de uso - 0019

### 2.2.4.- Modelo de Dominio

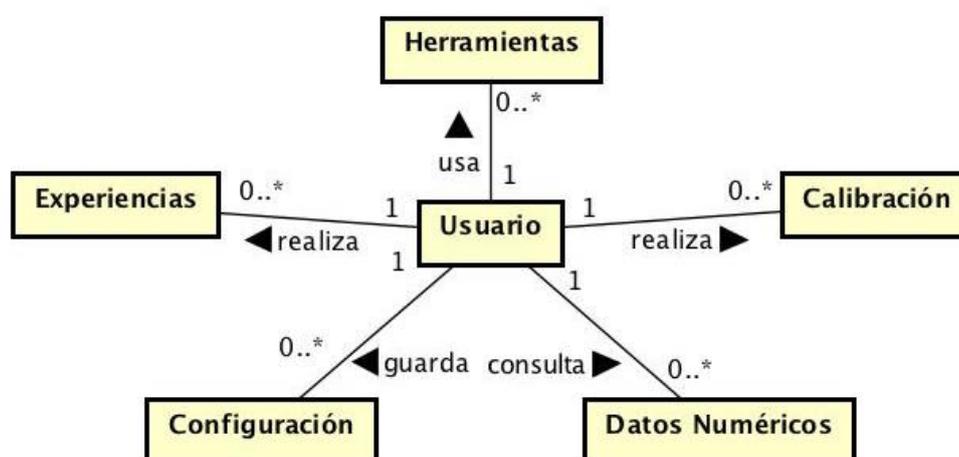


Ilustración 12: Modelo de dominio

El modelo de dominio arriba mostrado representa el dominio del usuario sobre la aplicación. Muestra la realización de las experiencias por parte del usuario, la realización de la calibración, el uso de las herramientas de la aplicación, el poder guardar la configuración y la consulta de los datos numéricos guardados.

Las experiencias representan las diferentes experiencias de las que dispone la aplicación, al igual que las herramientas representan al sonómetro, al espectro de frecuencias, al generador de ondas sonora y a la grabadora de sonido. Mientras que la calibración abarca las calibraciones del micrófono y altavoz.

La configuración puede ser modificada y guardada cuantas veces desee el usuario y lo mismo ocurre con la visualización de los datos numéricos.

## 2.3.- Diseño

### 2.3.1.- Diagramas de Secuencia

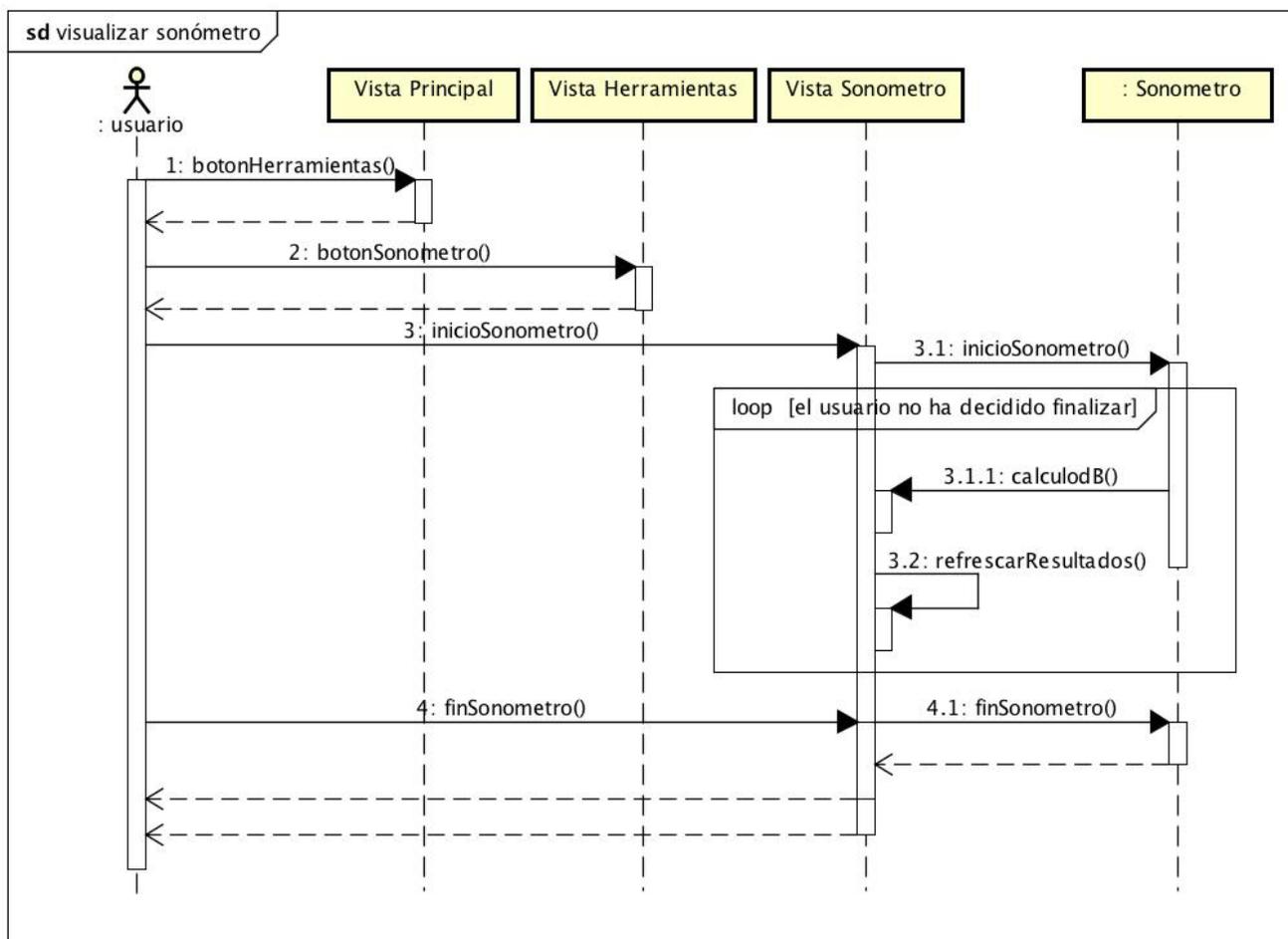


Ilustración 13: Diagrama de Secuencia - visualizar sonómetro

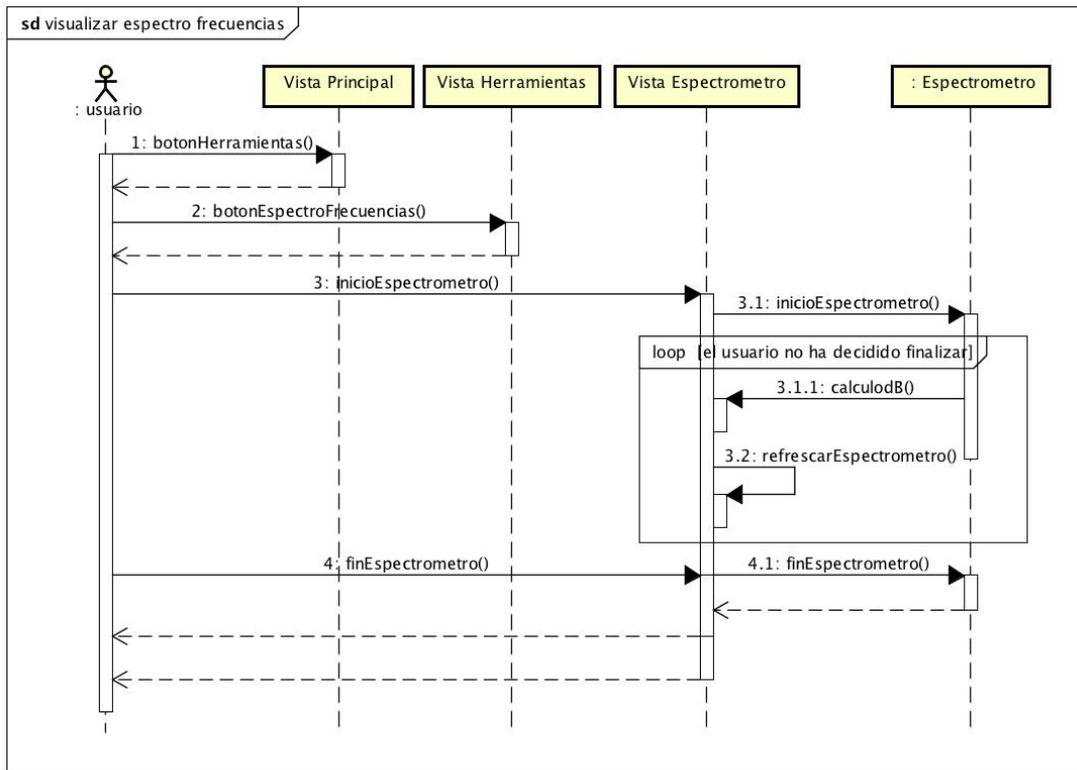


Ilustración 14: Diagrama de secuencia - visualizar espectro frecuencias

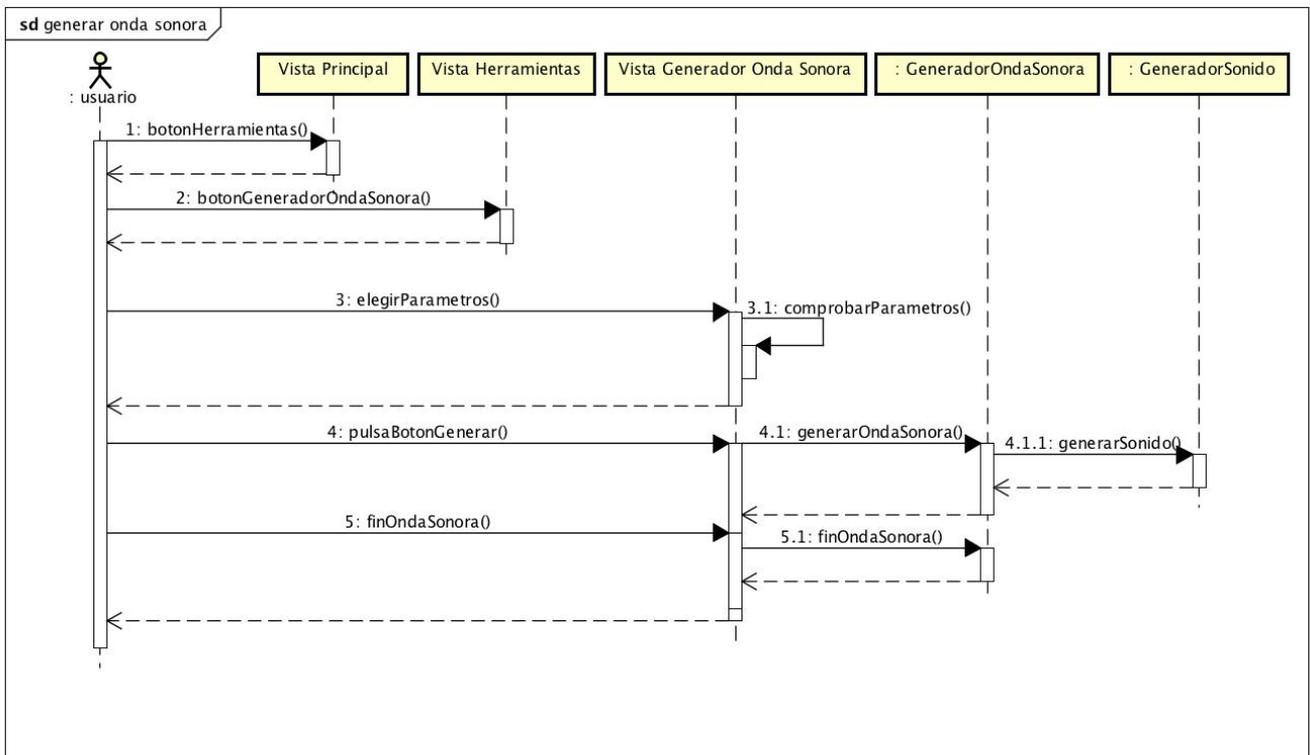


Ilustración 15: Diagrama de secuencia - generar onda sonora

## 2.- PLANIFICACIÓN Y GESTIÓN DE PROYECTO | AudiOS

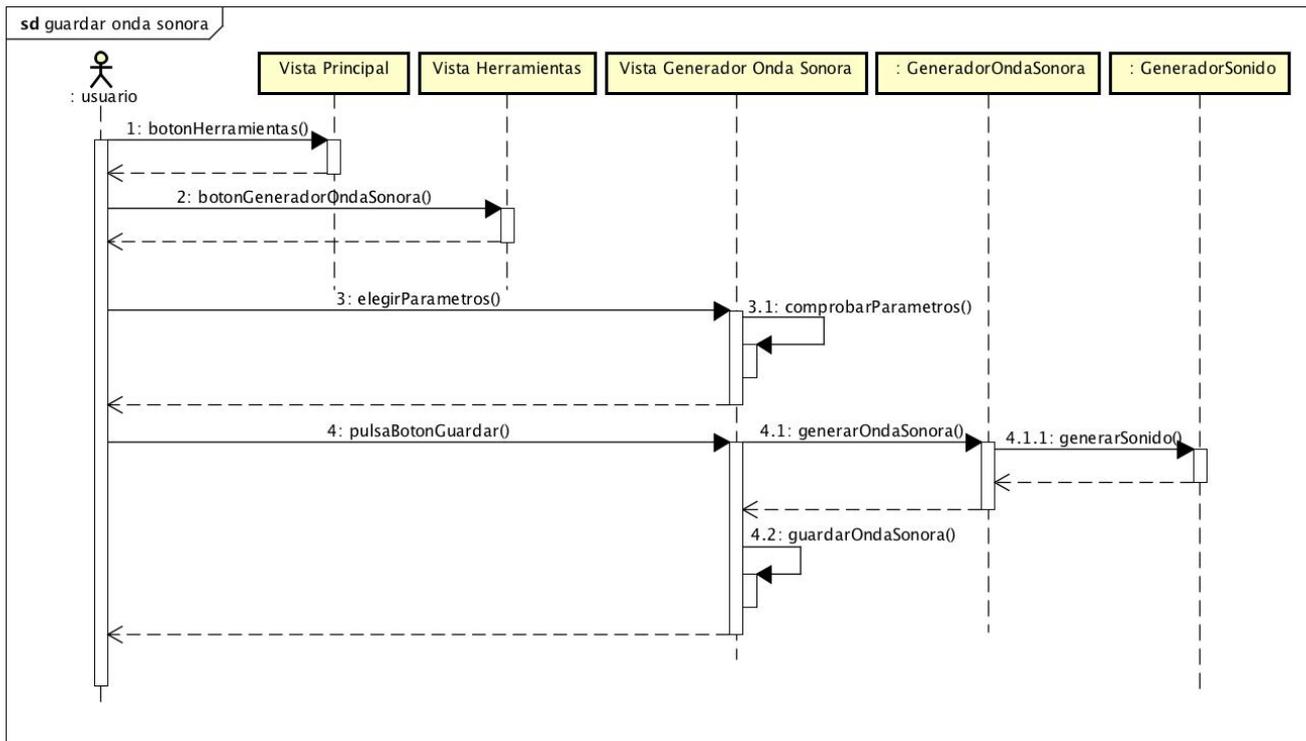


Ilustración 16: Diagrama de secuencia - guardar onda sonora

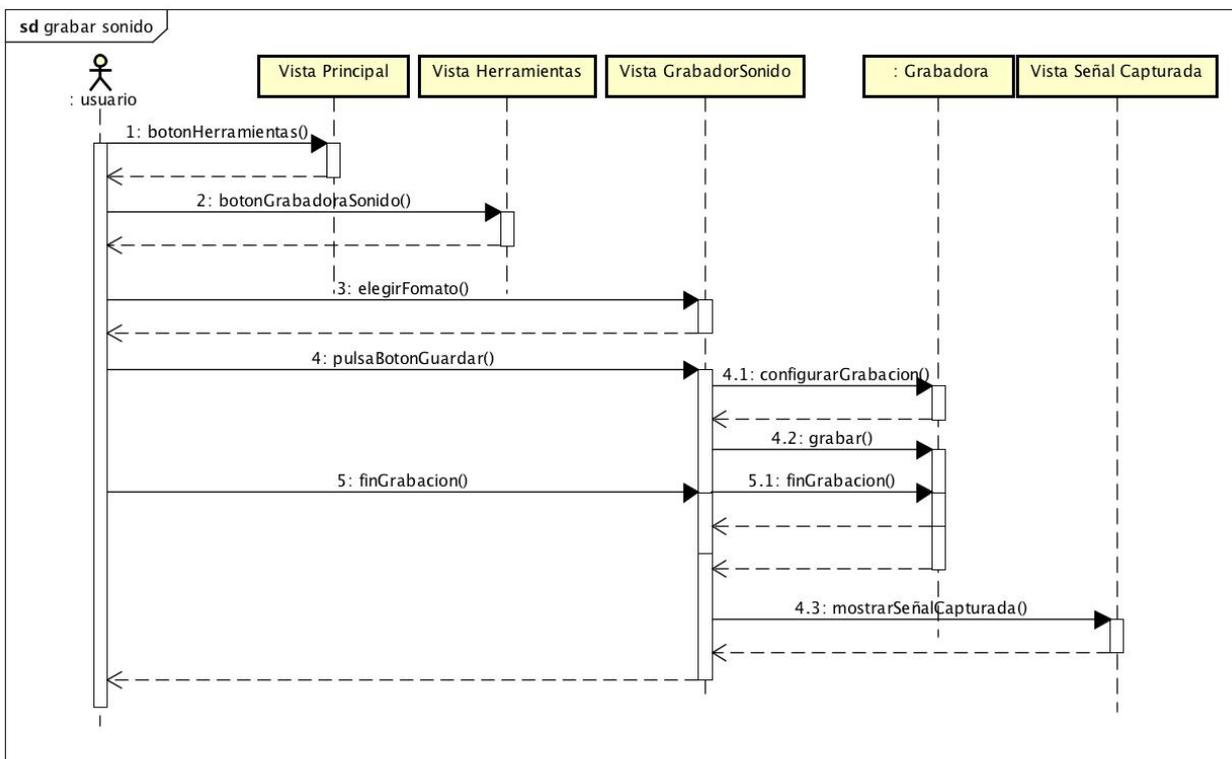


Ilustración 17: Diagrama de secuencia - grabar sonido

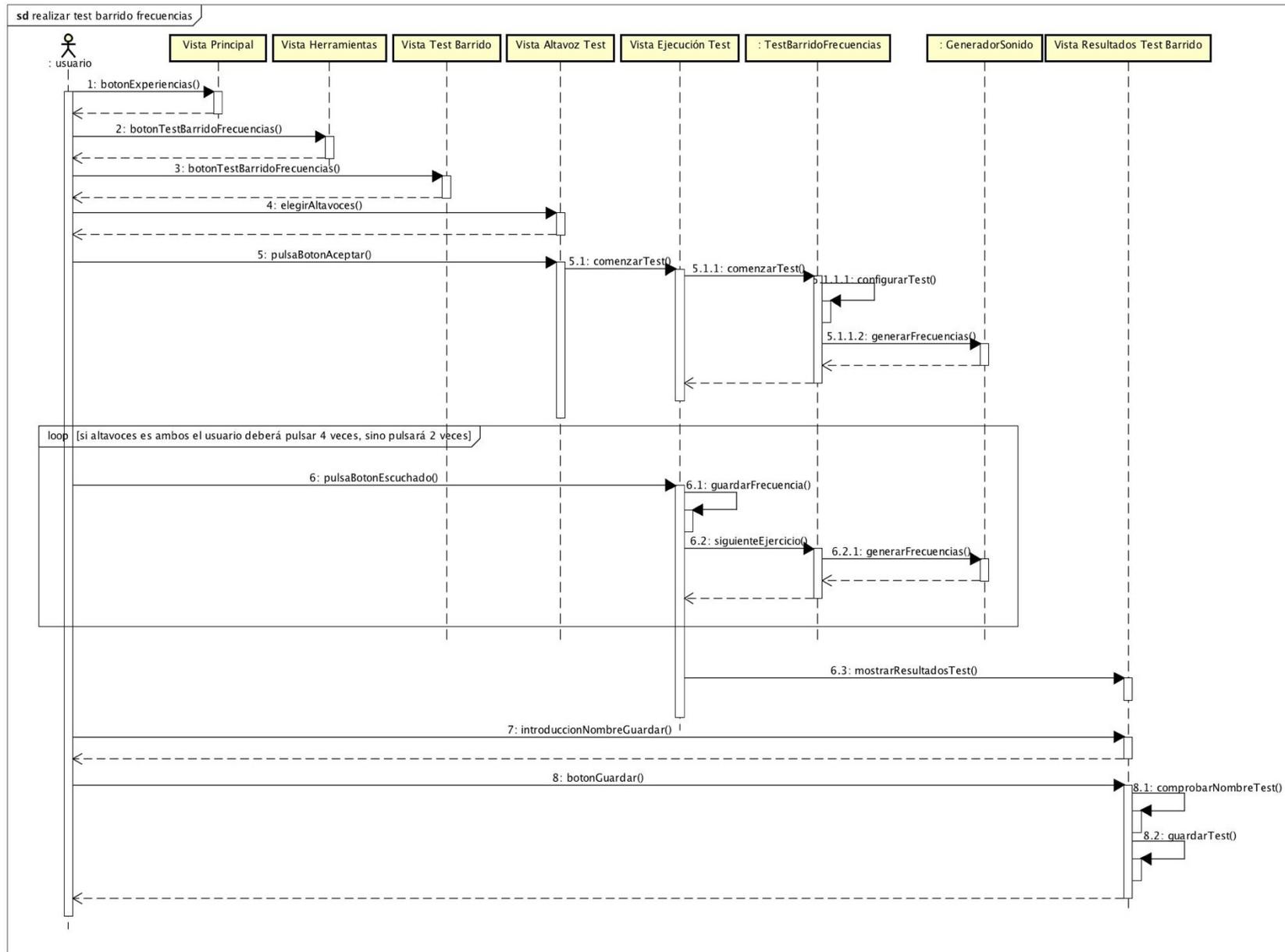


Ilustración 18: Diagrama de secuencia - realizar test barrido frecuencias

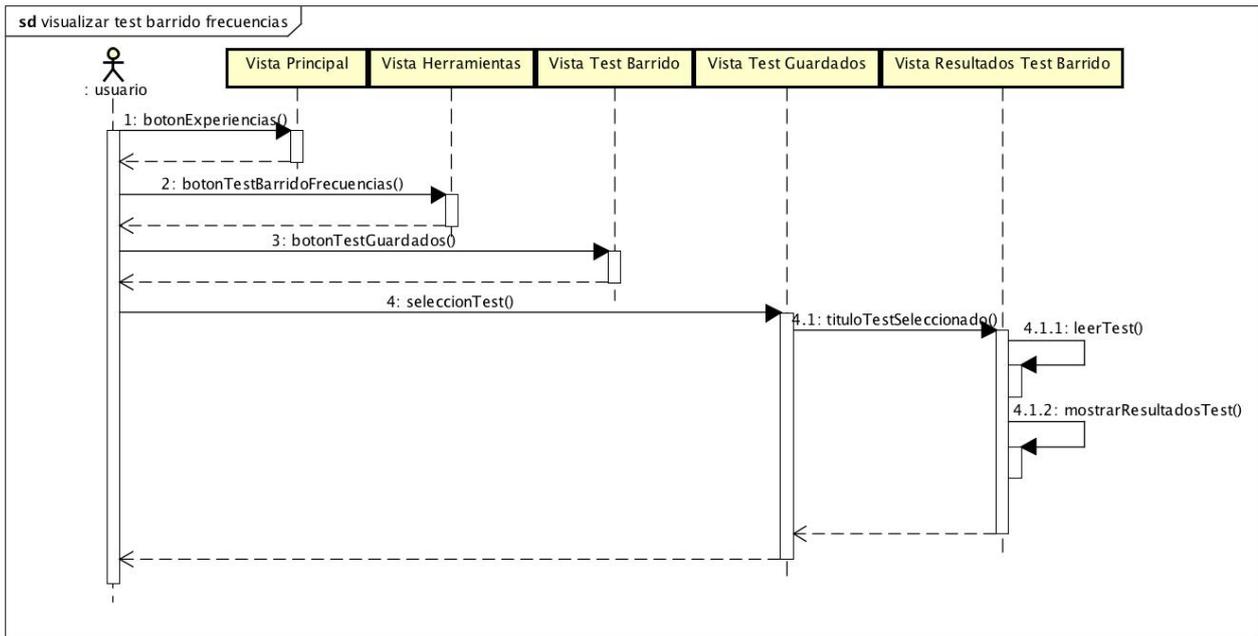


Ilustración 19: Diagrama de secuencia - visualizar test barrido frecuencias

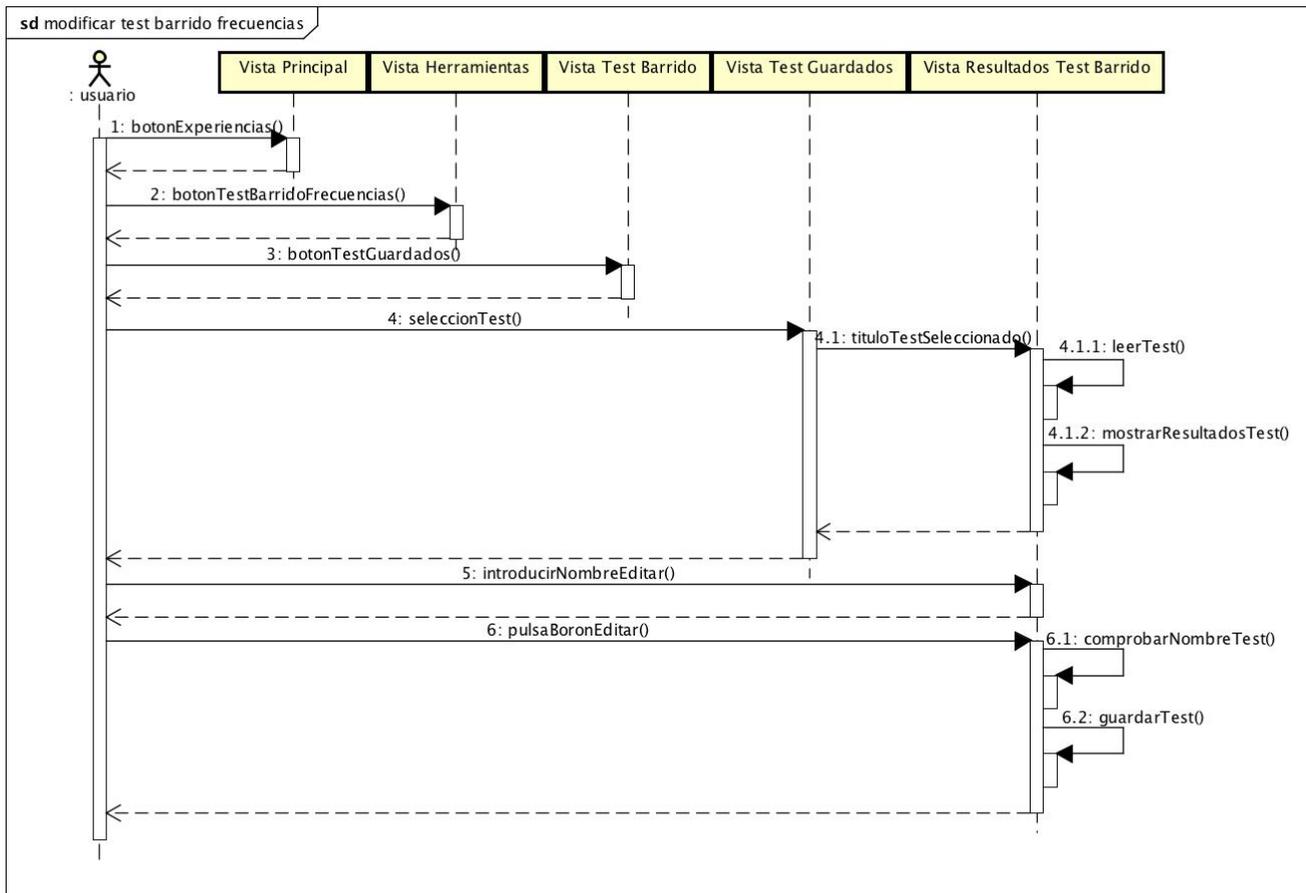


Ilustración 20: Diagrama de Secuencia: modificar test barrido frecuencias

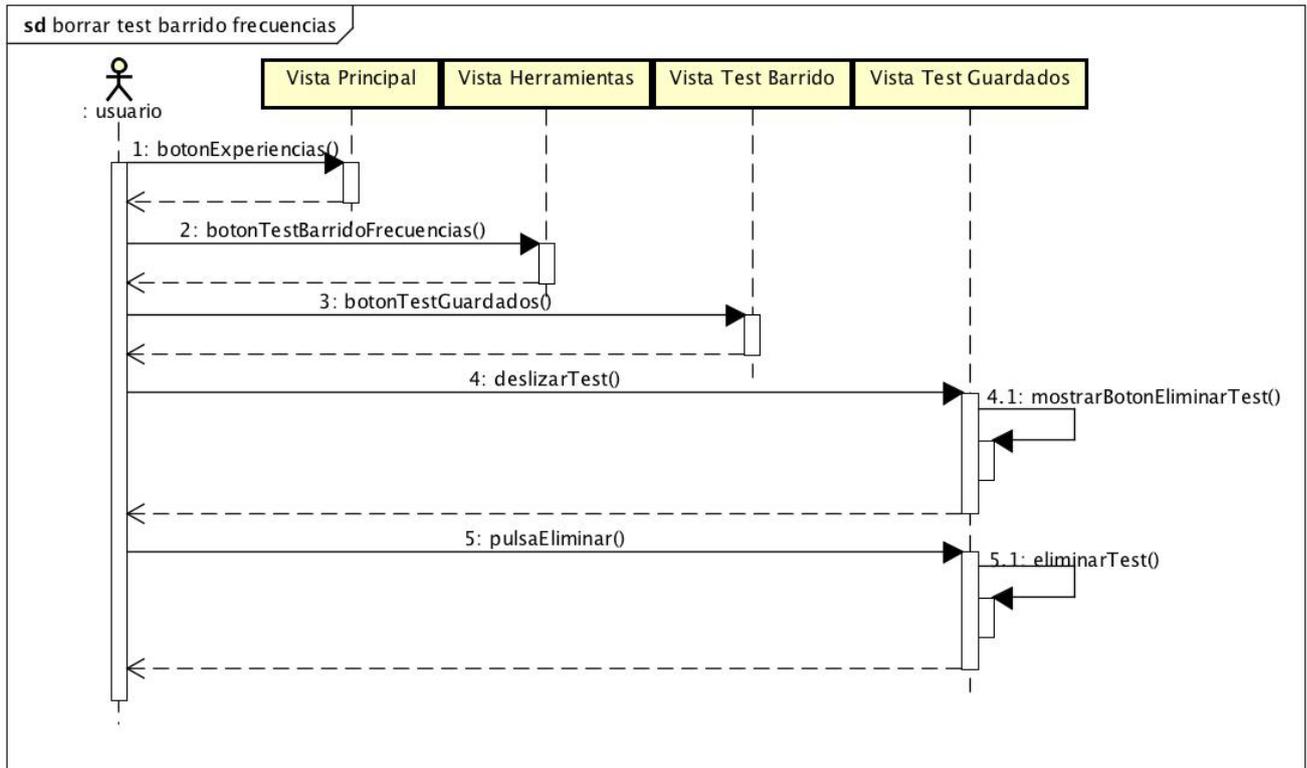


Ilustración 21: Diagrama de Secuencia - borrar test barrido frecuencias

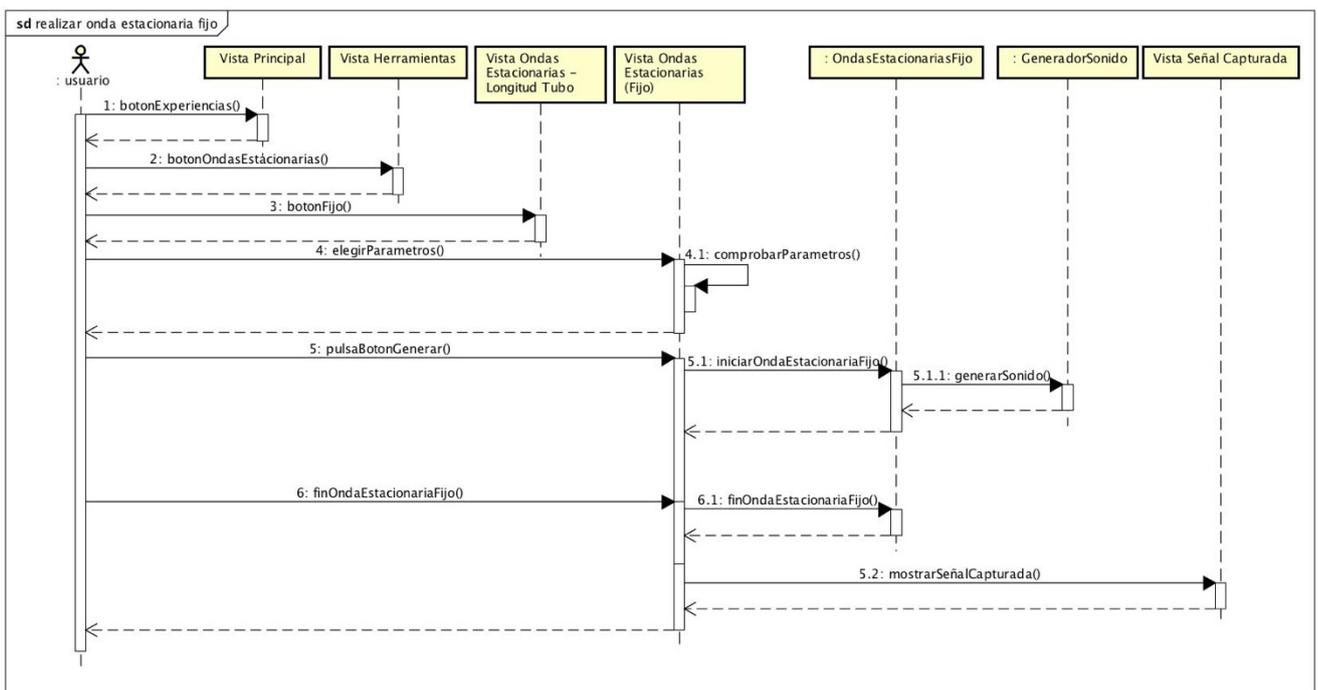


Ilustración 22: Diagrama de Secuencia - realizar ondas estacionarias fijo

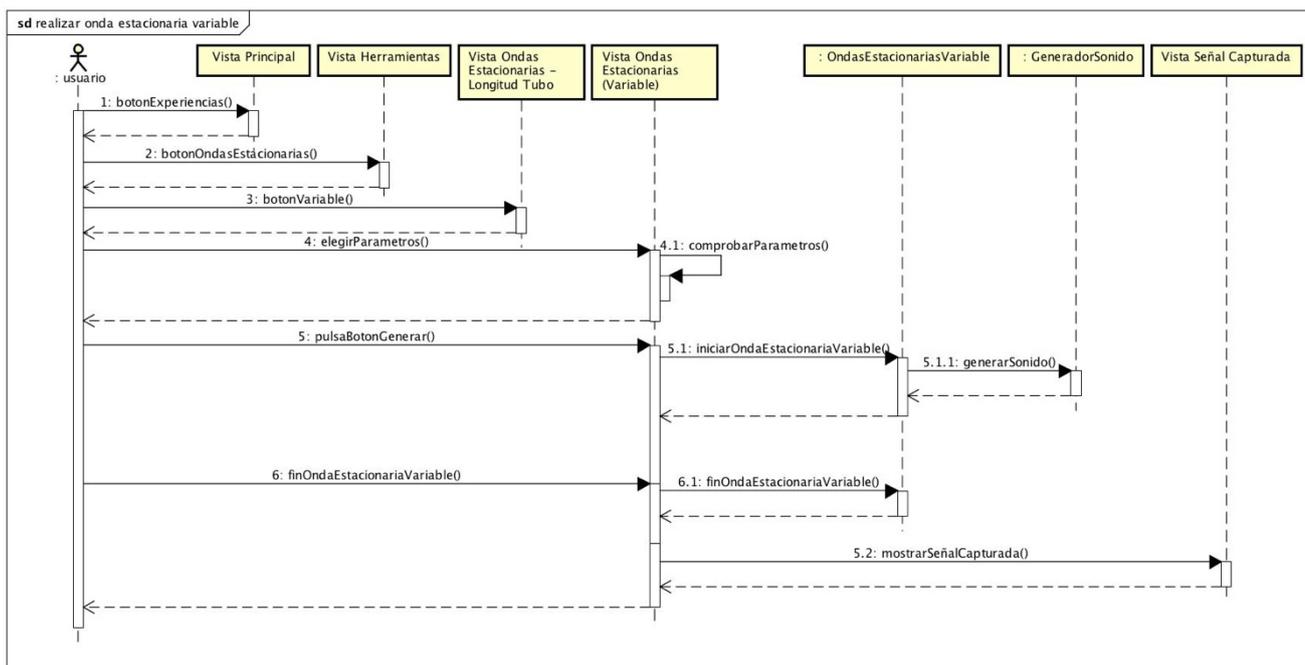


Ilustración 23: Diagrama de Secuencia - realizar onda estacionaria variable

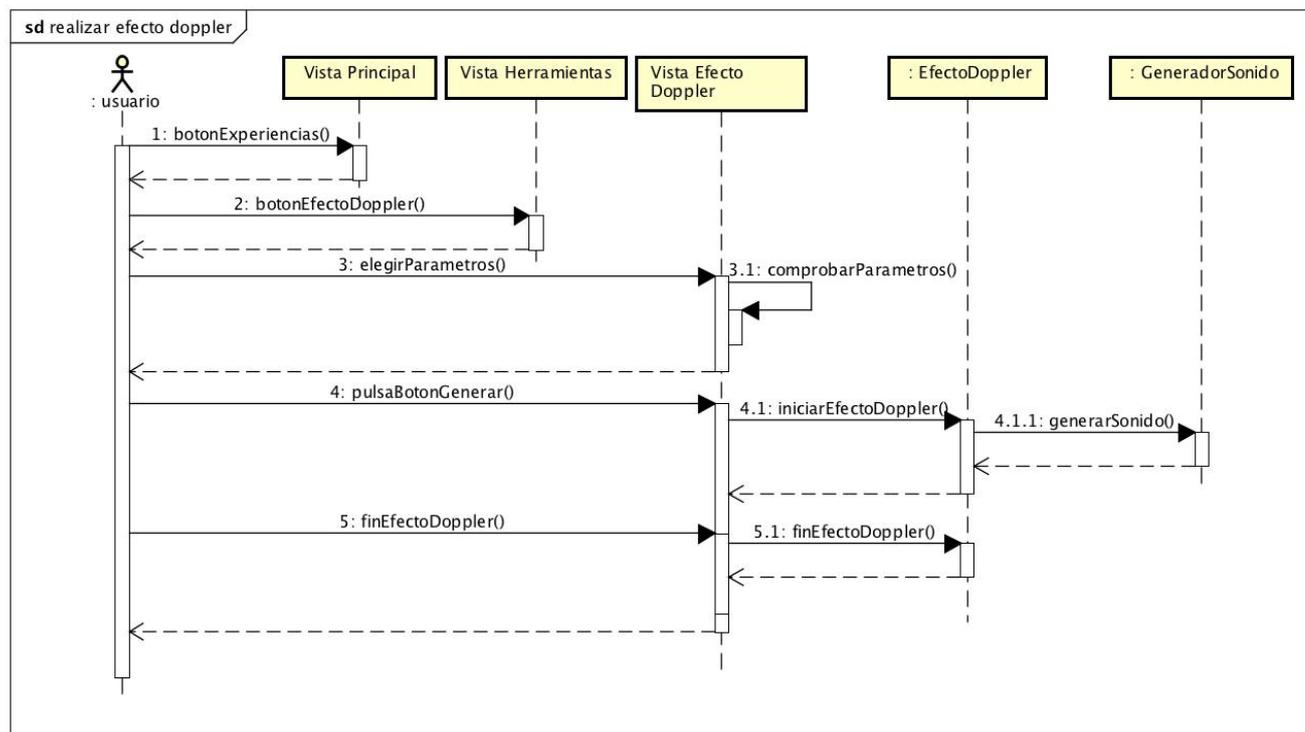


Ilustración 24: Diagrama de Secuencia - realizar efecto doppler

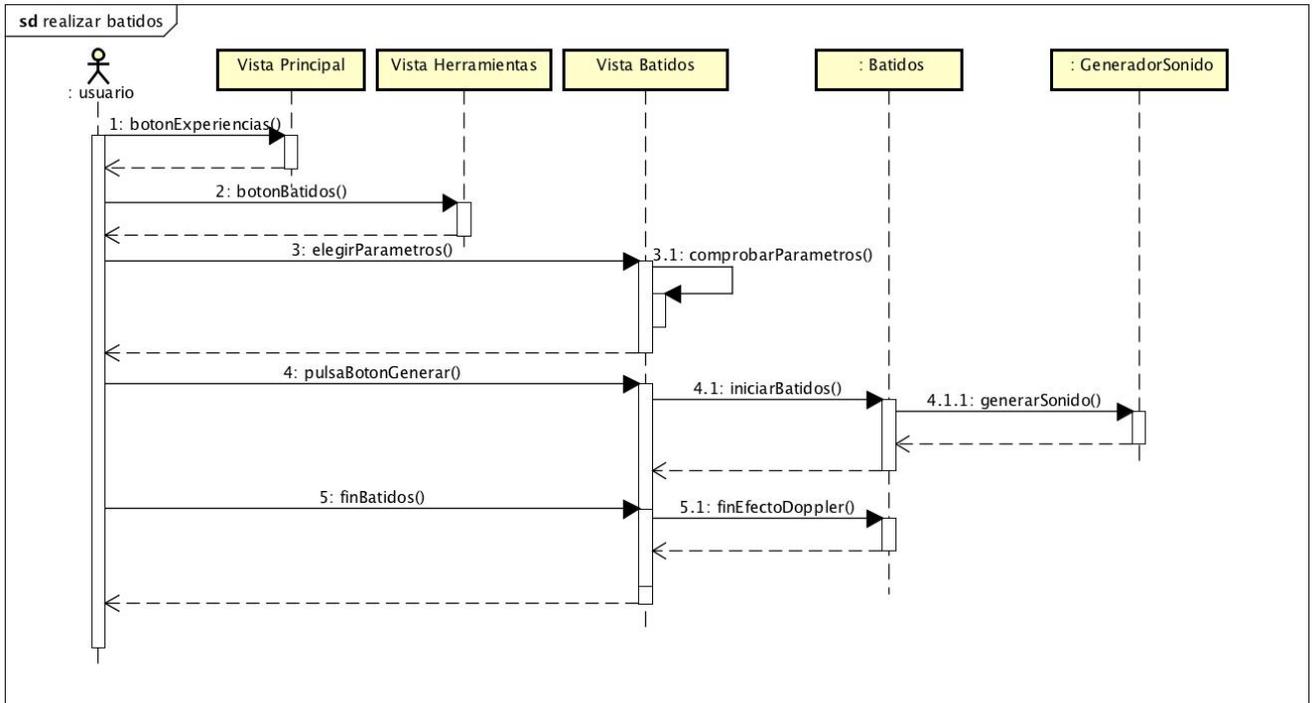


Ilustración 25: Diagrama de Secuencia - realizar batidos

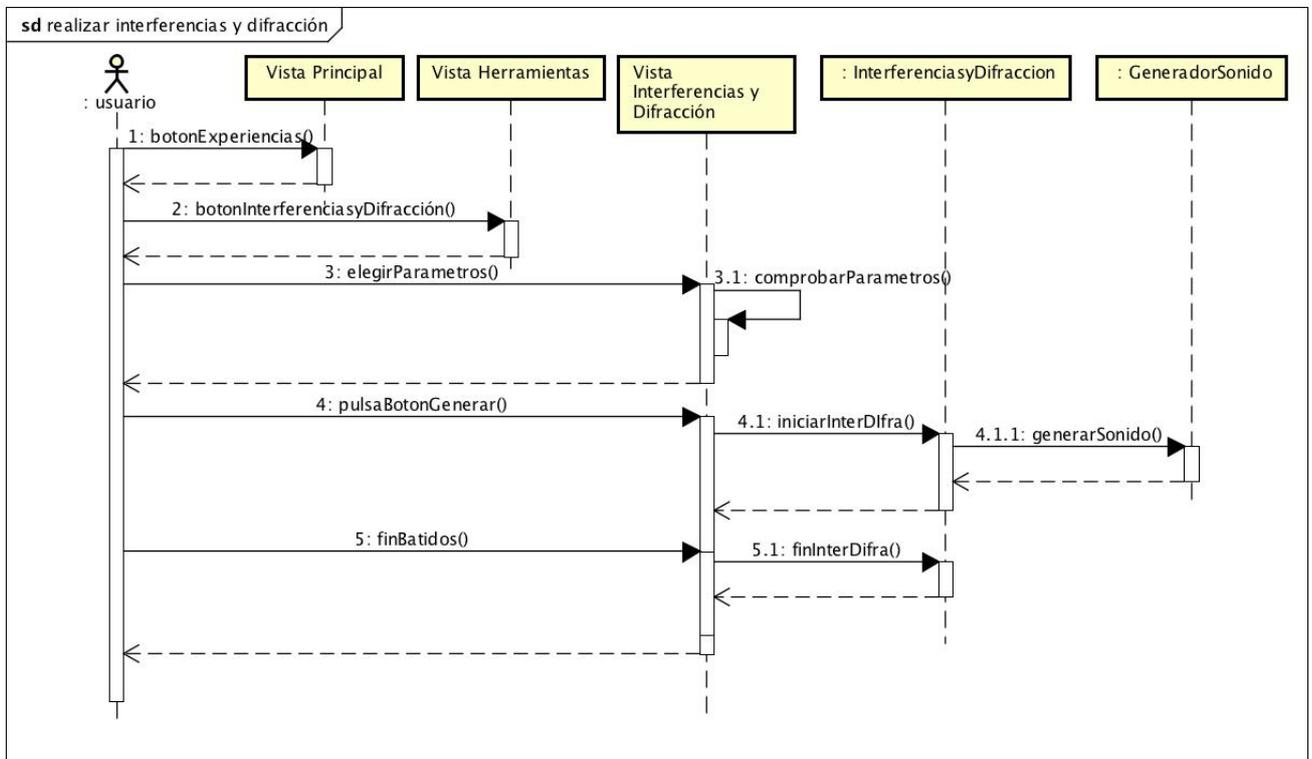


Ilustración 26: Diagrama de Secuencia - realizar interferencias y difracción

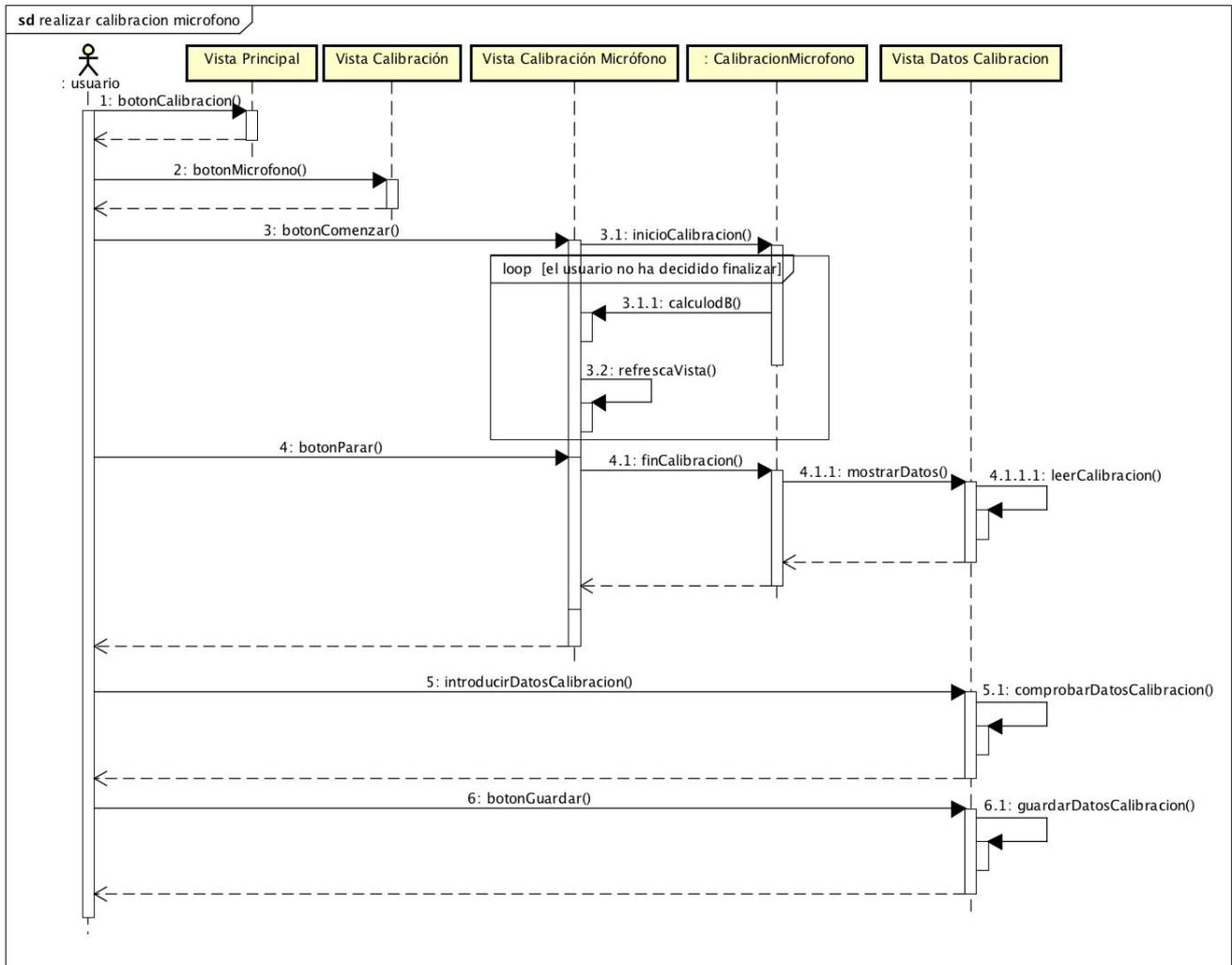


Ilustración 27: Diagrama de Secuencia - realizar calibración micrófono

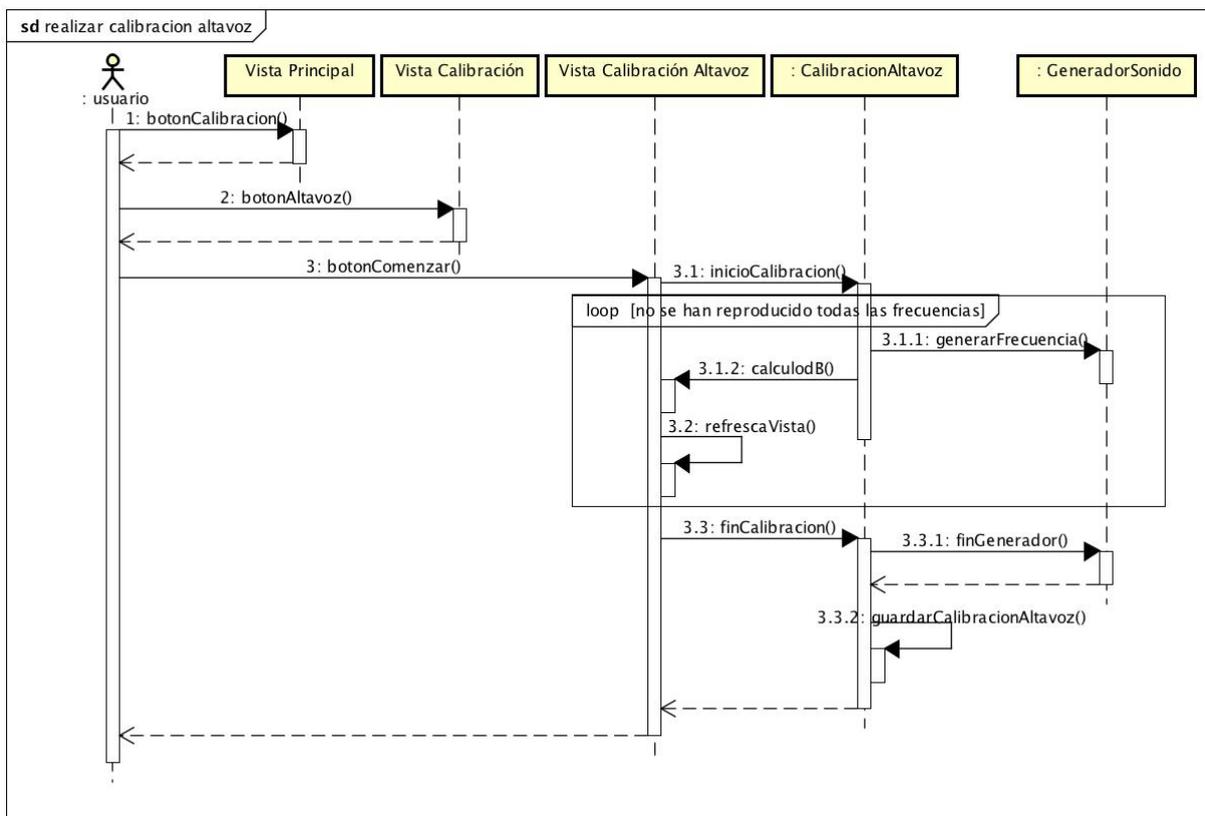


Ilustración 28: Diagrama de Secuencia - realizar calibración altavoz

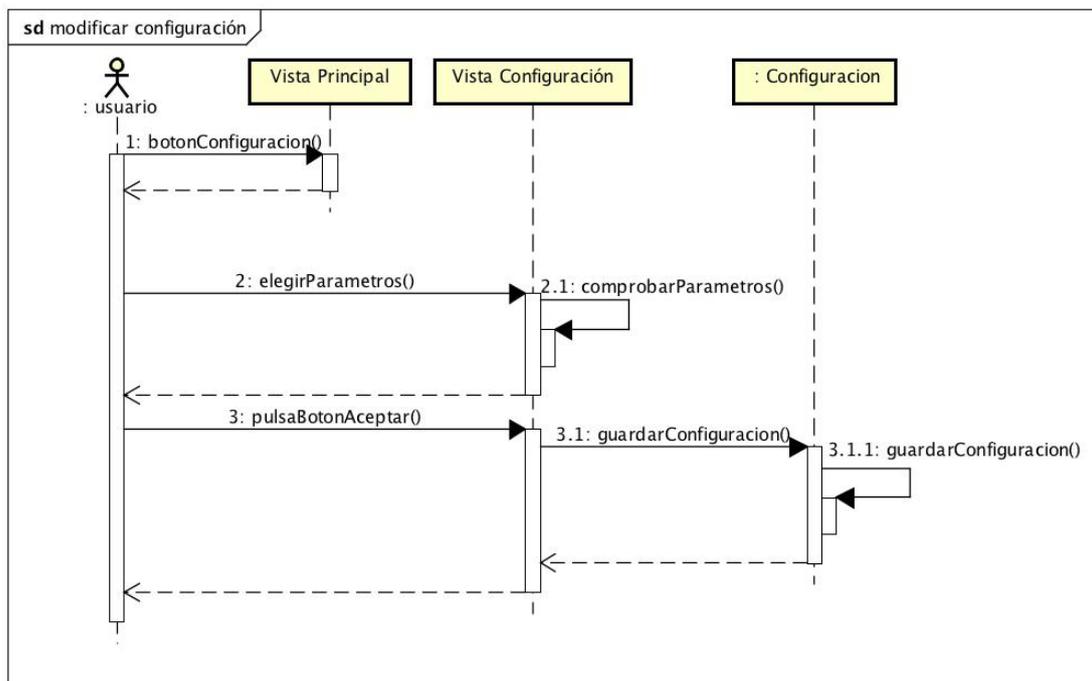


Ilustración 29: Diagrama de Secuencia - modificar configuración

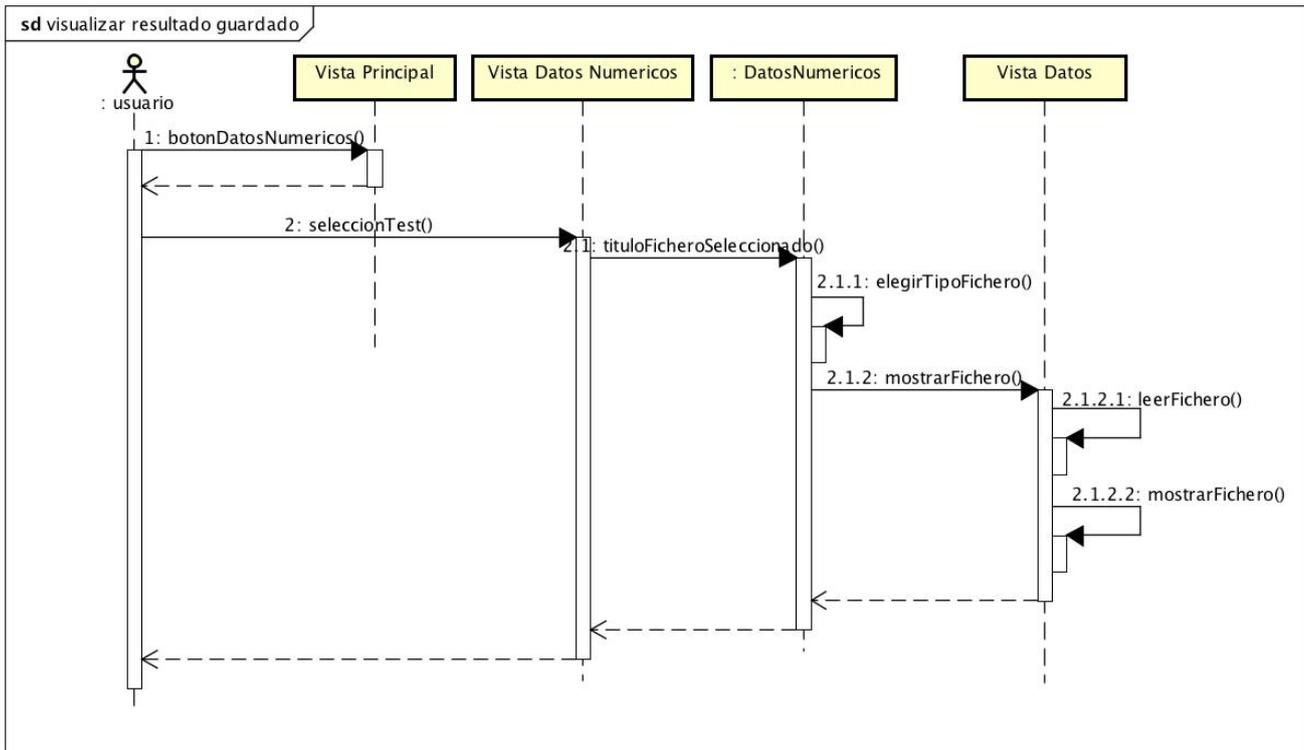


Ilustración 30: Diagrama de Secuencia - visualizar resultado guardado

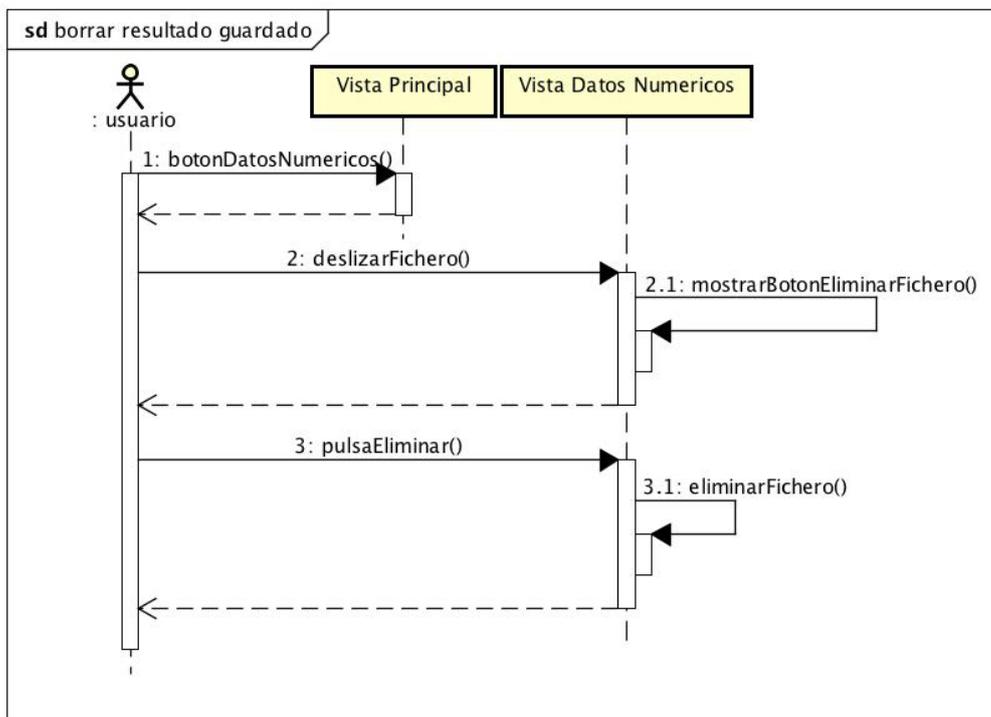


Ilustración 31: Diagrama de Secuencia - borrar resultado guardado

### 2.3.2.- Diagramas de Clases

Las diferentes clases de la aplicación se han agrupado en 3 paquetes diferentes llamados calibración, herramientas y experiencias.

Se han usado nombres muy intuitivos con lo que se puede intuir fácilmente que en el paquete de calibración se encontrarán las clases relacionadas con la calibración, en el paquete de herramientas se introducen las clases relacionadas con las herramientas y en el paquete experiencias se encontrarán todas las clases relacionadas con las diferentes experiencias que se pueden realizar en la aplicación.



Ilustración 32: Diagrama de paquetes

A continuación se muestra el diagrama de clases de la aplicación:

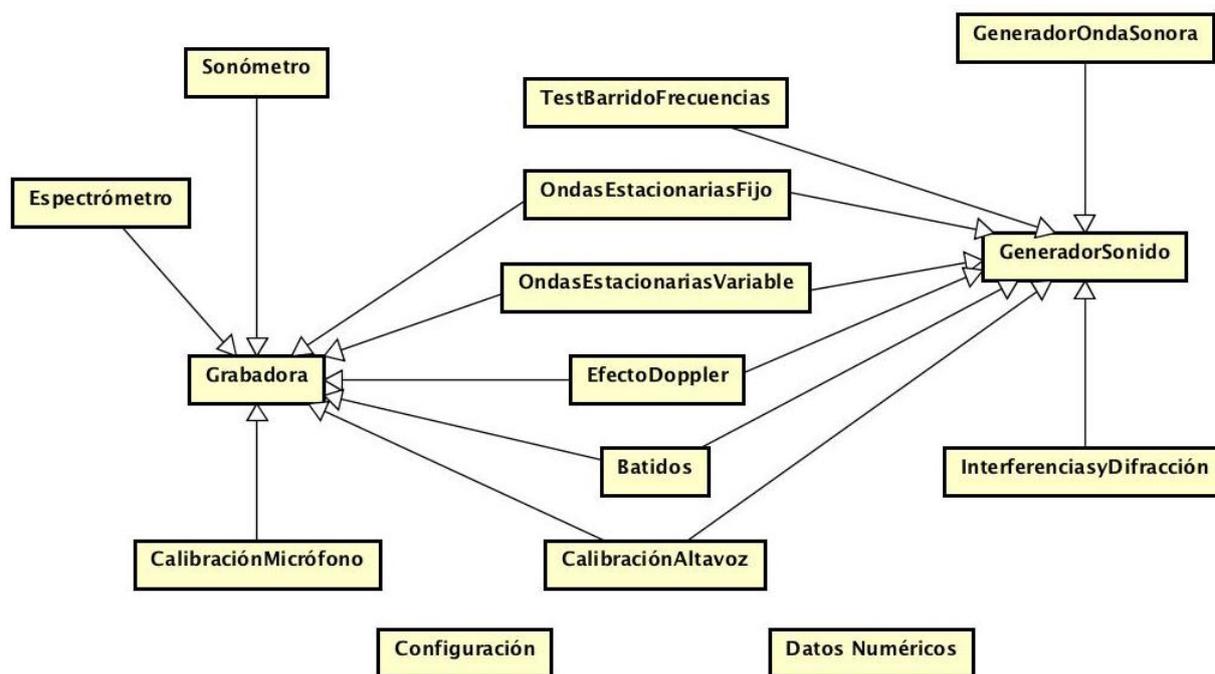


Ilustración 33: Diagrama de clases

2.3.3.- Clases Destalladas

La clase *Sonómetro* se encarga de la ejecución de la herramienta sonómetro de la aplicación.

<b>Sonómetro</b>
- intervaloMuestreo : int - frecuenciaMuestreo : float - cantidadEjeX : [String] - datosDibujar : [Double] - N : int - tipoVentana : String - x : DSPSplitComplex
+ pausarSonometro() : void + playSonometro() : void + reiniciar() : void + comenzarGrabacion() : void + terminarGrabacion() : void + pulsarTipoBoton() : void + setChart(cantidadEjeX : [String], datosDibujar : [Double]) : void + mideEspectrometro(x : DSPSplitComplex) : void

Ilustración 34: Clase sonómetro

La clase *Espectrómetro* se encarga de la ejecución de la herramienta espectro de frecuencias de la aplicación.

<b>Espectrómetro</b>
- x : DSPSplitComplex - resultado : [Float] - buffer : [Int] - inicioBuffer : int - escala : float - numFFT : int - frecuencias : [String] - decibelios : [Double]
+ pausarEspectro() : void + playEspectro() : void + pulsarBotonTipo() : void + darBotonCuadrícula() : void + guardarValores() : void + almacenaResultadosFFT(x : DSPSplitComplex, resultado : [Float], buffer : [Int], inicioBuffer : int) : void + calculaDBFS(resultado : [Float], escala : float, numFFT : int) : void + setChart(frecuencias : [String], decibelios : [Double]) : void

Ilustración 35: Clase espectrómetro

La clase *GeneradorOndaSonora* se encarga de la ejecución de la herramienta generador onda sonora de la aplicación.

<b>GeneradorOndaSonora</b>
- frecuenciaInicial : float - frecuenciaFinal : float - volumenIn : float - volumenFin : float - altavoz : String - formatoFichero : String - duracion : int
+ reproducirOnda() : void + terminar() : void + guardar() : void

Ilustración 36: Clase generador onda sonora

La clase *GeneradorSonido* se encarga de la creación y ejecución de las ondas o sonidos necesarios en las herramientas o experiencias de la aplicación.

<b>GeneradorSonido</b>
- buffer : AVAudioPCMBuffer
+ preparePlaying() : void + bufferSinusoidal(buffer : AVAudioPCMBuffer) : void + bufferCuadrada(buffer : AVAudioPCMBuffer) : void + bufferTriangular(buffer : AVAudioPCMBuffer) : void + bufferDienteSierra(buffer : AVAudioPCMBuffer) : void + bufferRuidoBlanco(buffer : AVAudioPCMBuffer) : void + bufferRuidoRosa(buffer : AVAudioPCMBuffer) : void + bufferRuidoMarron(buffer : AVAudioPCMBuffer) : void

Ilustración 37: Clase generador sonido

La clase *Grabadora* se encarga de la ejecución de la herramienta grabadora sonido de la aplicación.

<b>Grabadora</b>
- formatoFichero : String
+ pulsarGrabacion() : void + pulsarParar() : void + configuracionGrabacion() : void

Ilustración 38: Clase grabadora

La clase *TestBarridoFrecuencias* se encarga de la ejecución de la experiencia test barrido de frecuencias de la aplicación.

<b>TestBarridoFrecuencias</b>
- temporizador : NSTimer
+ cambiarFrecuencia(temporizador : NSTimer) : void + escuchadoSiguiente() : void + play() : void + pausa() : void + abortar() : void

*Ilustración 39: Clase test barrido frecuencias*

La clase *OndasEstacionariasFijo* se encarga de la ejecución de la experiencia onda estacionaria de tubo fijo de la aplicación.

<b>OndasEstacionariasFijo</b>
- frecuenciaInicial : float - frecuenciaFinal : float - volumen : float - altavoz : String - formatoFichero : String - duracion : int - lonTubo : int - diaTubo : int
+ reproducirGrabar() : void + terminar() : void + reproducirOnda() : void

*Ilustración 40: Clase ondas estacionarias fijo*

La clase *OndasEstacionariasVariable* se encarga de la ejecución de la experiencia onda estacionaria de tubo variable de la aplicación.

<b>OndasEstacionariasVariable</b>
- frecuencia : float - volumen : float - altavoz : String - formatoFichero : String - duracion : int - lonTuboIn : int - lonTuboFin : int - diaTubo : int
+ reproducirGrabar() : void + terminar() : void + reproducirOnda() : void

*Ilustración 41: Clase ondas estacionarias variable*

La clase *EfectoDoppler* se encarga de la ejecución de la experiencia efecto doppler de la aplicación.

<b>EfectoDoppler</b>
- frecuencia : float - volumen : float - altavoz : String - formatoFichero : String
+ reproducirGrabar() : void + terminar() : void + reproducir() : void

*Ilustración 42: Clase efecto doppler*

La clase *Batidos* se encarga de la ejecución de la experiencia batidos de la aplicación.

<b>Batidos</b>
- frecuenciaAltzq : float - frecuenciaAltDch : float - volumen : float - formatoFichero : String - duracion : int
+ reproducirBatido() : void + terminar() : void + guardar() : void + grabarOnda() : void

*Ilustración 43: Clase batidos*

La clase *InterferenciasyDifracción* se encarga de la ejecución de la experiencia interferencias y difracción de la aplicación.

<b>InterferenciasyDifracción</b>
- intervaloMuestreo : float - frecuencia : float - volumen : float - altavoz : String
+ reproducirInterDifra() : void + mostrardB() : void

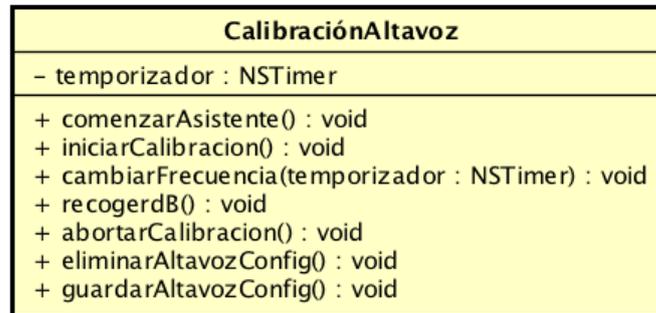
*Ilustración 44: Clase interferencias y difracción*

La clase *CalibraciónMicrófono* se encarga de la ejecución de la calibración del micrófono de la aplicación.

<b>CalibraciónMicrófono</b>
+ pararGrabacion() : void

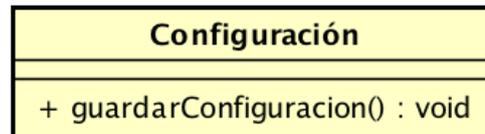
*Ilustración 45: Clase calibración micrófono*

La clase *CalibraciónAltavoz* se encarga de la ejecución de la calibración del altavoz de la aplicación.



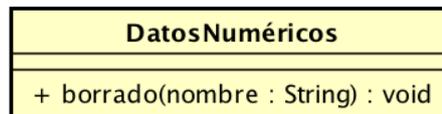
*Ilustración 46: Clase calibración altavoz*

La clase *Configuración* se encarga de la configuración de la aplicación.



*Ilustración 47: Clase configuración*

La clase *DatosNuméricos* se encarga de la visualización de datos guardados en herramientas o experiencias de la aplicación.



*Ilustración 48: Clase datos numéricos*

## 3.- SISTEMA iOS

---

El sistema iOS fue conocido en el 2007 con el nombre de iPhone OS por la empresa Apple pero no fue hasta unos meses más tarde cuando salió el primer iPhone. Al poco tiempo llegó el iPod Touch con el mismo sistema operativo y las mismas funcionalidades que el iPhone pero con la única diferencia de que no se permitían hacer llamadas con él. El iPod Touch tuvo una buena aceptación y por ello salió el iPad, que es igual que el iPod Touch pero con mayor tamaño y alguna característica nueva. Y finalmente en 2010 iPhone OS pasó a llamarse como lo conocemos hoy en día iOS y desde entonces se ha ido actualizando hasta tener, de momento iOS 9.

Los dispositivos que usa iOS siguen siendo el iPhone, iPod e iPad. Es más estricto que el sistema operativo Android por ello solo se puede encontrar en un pequeño porcentaje de terminales. Para programar y poder adaptar la aplicación al mayor número de dispositivos es más sencillo que en el caso de Android ya que se sabe las características, medidas y hardware de los dispositivos en los que se va a instalar la aplicación.

### 3.1.- Características

---

iOS es una variante de Mac OS X, la diferencia es que el primero está orientado a utilizarse en los dispositivos móviles con pantalla táctil mientras el otro está pensado para los ordenadores. Estos dos sistemas similares están basados en Unix.

Una de las principales características de este sistema es la multitarea que permite usar más de una tarea al mismo tiempo pudiendo alternarlas y volver al punto que te encontrabas en cada una de ellas. Además incluye la tecnología multitouch que es capaz de reconocer gestos sobre la pantalla para interactuar con ella, como hacer zoom, seleccionar texto, etc.....

Su interfaz gráfica es sencilla y fluida, con la que se comunica el usuario mediante gestos táctiles y movimientos del móvil que son captados por acelerómetros del dispositivo. Los elementos de control que usa son deslizadores, interruptores o botones.

Lo primero que nos encontramos en la ejecución de un sistema iOS es una pantalla de bloqueo/desbloqueo que nos da acceso a la pantalla principal. El desbloqueo se puede realizar mediante código o no y en algunos modelos con el reconocimiento de nuestra huella dactilar. Una vez desbloqueado el dispositivo nos encontramos con la pantalla principal que contiene los iconos de las aplicaciones y una barra de herramientas inferior llamada Dock, en la que se pueden poner aplicaciones que se encontrarán disponibles en las diferentes pantallas que tengamos. Se pueden tener múltiples pantallas con aplicaciones y movernos por ellas solo deslizando el dedo por la pantalla. En la parte superior de la pantalla se encuentra la barra de estado que nos indica la cobertura, en el caso de tener tarjeta telefónica, la conexión a internet, la hora, porcentaje de batería e incluso si tenemos activado el bluetooth.

### 3.- SISTEMA iOS | AudiOS



Ilustración 49: Pantalla de bloqueo/desbloqueo



Ilustración 50: Pantalla Principal

Tenemos la posibilidad de agrupar los iconos de las aplicaciones en carpetas según el criterio que se desee aunque con límite de número de aplicaciones por carpeta pero es una buena funcionalidad para tener las aplicaciones más usadas o del mismo tipo juntas.

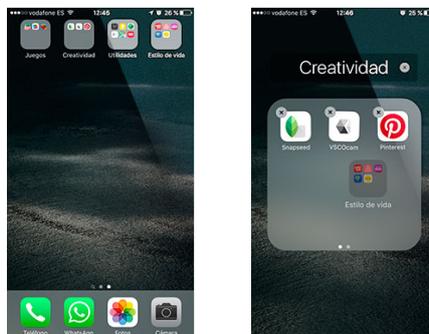


Ilustración 51: Agrupación de iconos en carpetas

Una característica muy útil y que es accesible desde cualquier pantalla o aplicación es el centro de notificaciones, que se encuentra de manera oculta en la parte superior y se accede deslizando el dedo desde el borde superior hacia abajo. Si se despliega el centro de notificaciones nos encontramos con dos pestañas, una nos informa del resumen del día de hoy con el día y eventos que se tienen apuntados en el calendario. La otra pestaña nos enseña las notificaciones que tenemos y hemos ignorado para que se pueda consultar más tarde.



Ilustración 52: Centro de notificaciones

Las notificaciones se pueden personalizar de que forma aparecen en pantalla, al igual que el centro de notificaciones también es personalizable y se pueden añadir/quitar widgets o decidir que notificaciones de que aplicaciones se desean recibir.

Al igual que hay un centro de notificaciones en la parte superior, en la parte inferior se encuentra el centro de control. El centro de control te permite el acceso directo a diferentes ajustes o aplicaciones que son interesantes tener un fácil acceso, como la linterna, la cámara, la calculadora, activar el wifi, activar el bluetooth, etc.....

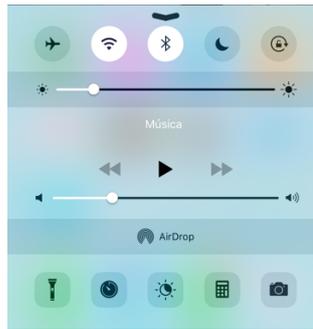


Ilustración 53: Centro de control

En iOS 9 se incluyó una utilidad muy efectiva que consiste en que cuando estás en una aplicación y te llega una notificación y la abres en ese mismo momento, arriba a la izquierda de la nueva pantalla aparecerá un botón para volver a la aplicación en la que estábamos. También ocurre con los enlaces, si damos a un enlace y este nos redirige a otra aplicación volverá a aparecer el botón.

Por último, una nueva característica de iOS 9 pero que sólo se encuentra en los iPhone es el modo de bajo consumo, que nos permite ahorrar batería cuando nos queda poca para que no nos quedemos incomunicados en el momento más inoportuno.

### 3.2.- Arquitectura

La arquitectura de iOS está basada en 4 capas, donde las capas más altas contienen servicios y funcionalidades para el desarrollo de las aplicaciones, mientras las capas más bajas se encargan de los servicios más básicos.

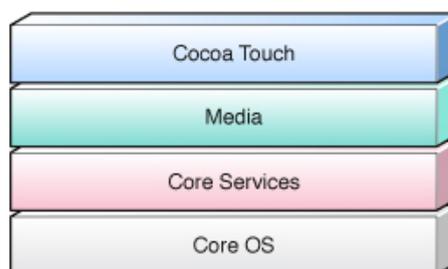


Ilustración 54: Arquitectura iOS

**Cocoa Touch:** es la capa visible que los usuarios utilizan para comunicarse con el sistema, donde se encuentra la interfaz del usuario. Esta capa es la que soporta las interacciones con el sistema mediante gestos, las notificaciones y la multitarea. Posee un conjunto de frameworks para el desarrollo de aplicaciones, los frameworks más importantes que contiene son UIKit y Foundation Framework.

UIKit es imprescindible para crear aplicaciones en iOS ya que sirve para gestionar la interfaz de usuario, la infraestructura para manejar eventos y la interacción con el sistema.

Foundation Framework permite acceder a las clases básicas que definen el comportamiento de los objetos, los tipos de datos, las colecciones y los servicios del sistema operativo. También Foundation está compuesto por patrones de diseño y mecanismos que hacen que las aplicaciones sean más eficientes y robustas.

**Media:** es la capa que da soporte a gráficos, audios, vídeos... todo lo que esté relacionado con ficheros multimedia. En esta capa se encuentran múltiples tecnologías para audio, como Media Player o Core Audio, para vídeo, como Media Player o Core Media, y para gráficos, como Core Graphics u Open GL.

**Core Services:** es la capa de servicios principales que permite acceder a todos los servicios básicos de iOS a las aplicaciones. Tecnologías como el almacenamiento iCloud o la compra de App se encuentran en este nivel. En esta capa se encuentra la biblioteca SQL que permite manejar y crear bases de datos en aplicaciones sin tener que ejecutar un proceso en un servidor, al igual que soporta lenguaje XML para escribir o leer de forma rápida datos XML.

**Core OS:** es la capa del núcleo del sistema operativo que como ya se explicó con anterioridad está basado en Mac OS X que a su vez está basado en Unix. Este nivel abarca el entorno del núcleo, los controladores y las interfaces de bajo nivel de Unix. Se encarga de las tareas básicas de bajo nivel como gestión de memoria virtual, hilos, sistema de archivos, acceso a la red, servicios de seguridad y procesos que interactúan directamente con el hardware. También contiene un gran número de interfaces que permiten el acceso a los servicios básicos de bajo nivel, a través de la biblioteca Libsystem.

### 3.3.- Dispositivos

Los dispositivos que usan el sistema iOS son los de la compañía Apple por ello son limitados y conocidos. Esto facilita el manejo de tamaños de los elementos de las vistas y la utilización de ciertas funciones que puedan ser soportados por el hardware de todos los dispositivos o por lo menos por la gran mayoría.

A continuación se van a exponer los dispositivos que se pueden adquirir de Apple y soportan iOS con sus características principales, es decir, los iPhone, los iPad y el iPod Touch:

iPhone 6s Plus	iPhone 6s	iPhone 6 Plus	iPhone 6	iPhone SE	iPad Pro de 12,9"	iPad Pro de 9,7"	iPad Air 2	iPad mini 4	iPad mini 2	iPod Touch
<b>Dimensiones</b>										
Alto 15,82 cm	Alto 13,83 cm	Alto 15,81 cm	Alto 13,81 cm	Alto 12,38 cm	Alto 30,57 cm	Alto 24 cm	Alto 24 cm	Alto 20,32 cm	Alto 20 cm	Alto 12,34 cm
Ancho 7,79 cm	Ancho 6,71 cm	Ancho 7,78 cm	Ancho 6,7 cm	Ancho 5,86 cm	Ancho 22,06 cm	Ancho 16,95 cm	Ancho 16,95 cm	Ancho 13,48 cm	Ancho 13,47 cm	Ancho 5,86 cm
<b>Pantalla</b>										
5,5 pulgadas	4,7 pulgadas	5,5 pulgadas	4,7 pulgadas	4 pulgadas	12,9 pulgadas	9,7 pulgadas	9,7 pulgadas	7,9 pulgadas	7,9 pulgadas	4 pulgadas
Resolución 1.920 x 1.080	Resolución 1.334 x 750	Resolución 1.920 x 1.080	Resolución 1.334 x 750	Resolución 1.136 x 640	Resolución 2.732 x 2.048	Resolución 2.048 x 1.536	Resolución 2.048 x 1.536	Resolución 2.048 x 1.536	Resolución 2.048 x 1.536	Resolución 1.136 x 640
401 p/p	326 p/p	401 p/p	326 p/p	326 p/p	264 p/p	264 p/p	264 p/p	326 p/p	326 p/p	326 p/p
<b>Chip</b>										
Chip A9 con arquitectura de 64 bits	Chip A9 con arquitectura de 64 bits	Chip A8 con arquitectura de 64 bits	Chip A8 con arquitectura de 64 bits	Chip A9 con arquitectura de 64 bits	Chip A9X de 3ª generación con arquitectura de 64 bits	Chip A9X de 3ª generación con arquitectura de 64 bits	Chip A8X de 2ª generación con arquitectura de 64 bits	Chip A8 de 2ª generación con arquitectura de 64 bits	Chip A7 con arquitectura de 64 bits	Chip A8 con arquitectura de 64 bits
<b>Audio</b>										
Un altavoz	4 altavoces	4 altavoces	2 altavoces	2 altavoces	2 altavoces	Un altavoz				
<b>Sensores</b>										
Giroscopio de 3 ejes	Giroscopio de 3 ejes	Giroscopio de 3 ejes	Giroscopio de 3 ejes	Giroscopio de 3 ejes	Giroscopio de 3 ejes					
Acelerómetro	Acelerómetro	Acelerómetro	Acelerómetro	Acelerómetro	Acelerómetro	Acelerómetro	Acelerómetro	Acelerómetro	Acelerómetro	Acelerómetro
Barómetro	Barómetro	Barómetro	Barómetro	—	Barómetro	Barómetro	Barómetro	Barómetro	—	—

Tabla 44: Dispositivos Apple

### 3.4.- Seguridad

Apple diseñó el sistema operativo iOS en torno a la seguridad pensando en los riesgos de seguridad del entorno de escritorio. Los dispositivos iOS combinan software, hardware y servicios para juntos conseguir la máxima seguridad protegiendo el dispositivo y los datos que contiene, así como las acciones realizan de forma local, en redes y con servicios clave de Internet.

En la imagen siguiente se puede ver la arquitectura de la seguridad de iOS pudiéndose apreciar las diferentes tecnologías usadas para conseguirlo.

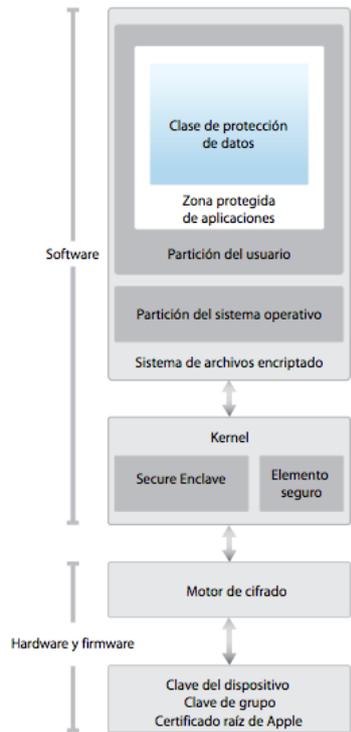


Ilustración 55: Arquitectura de la seguridad iOS

El software y el hardware del sistema iOS están protegidos en todos los componentes centrales de los dispositivos, por ello todos los componentes son seguros y de confianza. Todos los pasos que se realizan son analizados para cerciorarse de que el hardware y el software funcionan correctamente. Tanto está controlado el acceso al Kernel y drivers que solo un conjunto limitado de framework del sistema y aplicaciones pueden acceder a ellos.

La seguridad de los datos del usuario se garantizan con la encriptación y protección de los datos. Para que la encriptación no ralentice los dispositivos, iOS tiene un motor de cifrado AES de 256 bits integrado en la ruta de DMA, entre el almacenamiento flash y la memoria del sistema principal. Esto permite conseguir una encriptación de archivos muy eficiente que ningún software ni firmware puede leerlos directamente. La protección de datos consiste en la creación y la gestión de una jerarquía de claves, asignando a cada dato/archivo una clase.

La seguridad de las aplicaciones es muy importante y debe controlarse para garantizar la seguridad del sistema y la de los datos del usuario, por ello se añade capas de protección para garantizar la firma y verificación de las apps, además se consigue aislarlas para proteger los datos de usuario.

A parte de la seguridad del dispositivo y todo lo que contiene también hay que asegurar la seguridad que viene de fuera, la seguridad de red e internet. iOS utiliza protocolos de red estándar para comunicaciones autenticadas, autorizadas y encriptadas integrando tecnologías probadas y estándares nuevos de conexiones. Para una mayor seguridad limita los puertos de escucha y elimina utilidades de red innecesarias sin necesidad de usar un firewall. Las herramientas que proporciona Apple para un mayor aprovechamiento de sus dispositivos, como iCloud, FaceTime o iMessage, ya incluyen una gestión segura de los datos utilizando cada una su propia arquitectura de seguridad. Es importante que estas herramientas estén protegidas ya que manejan información y datos de los usuarios mediante conexiones inalámbricas.

### 3.5.- iOS vs Android

Se trata de los dos sistemas operativos en dispositivos móviles más usados en el mundo, se trata de una batalla constante y las comparaciones siempre están a la orden del día.

La primera diferencia que destacaría y que es en la que se suelen fijar los usuarios es la rigurosidad con la que Apple trabaja frente a los múltiples cambios y personalización que ofrece Android.

Respecto a la interfaz de usuario, iOS usa un diseño más limpio y moderno, mientras que Android opta por el colorido. Se programan en lenguajes de programación diferentes.

La gran mayoría de las aplicaciones se encuentran para los dos sistemas, y de no ser así suele haber una aplicación similar. Las aplicaciones no se pueden compartir ya que usan lenguajes de programación diferentes. Por ejemplo, la aplicación de asistencia Siri que se encuentra en los sistemas iOS no existe para Android pero este tiene su propio asistente llamado Google Now. No son exactamente iguales ya que Siri interactúa con el usuario mediante la voz para el uso del dispositivo mientras que Google Now también interactúa con la voz pero para dar información buscada en la red.

Siguiendo con el tema de aplicaciones y como ya se ha comentado anteriormente, en iOS es más sencillo de programar que en Android por tener menor número de dispositivos que lo utilizan. Por tener menos dispositivos a los que dar soporte tiene menos fragmentación que Android, que es el mayor problema con el que se encuentra este. La fragmentación son las diferentes versiones que existen del sistema operativo, ya que se tiene que adaptar a cada tipo de dispositivos en el que se encuentre. Los dispositivos que utilizan el sistema Android son miles, una gran diferencia con las decenas de dispositivos que usan iOS.

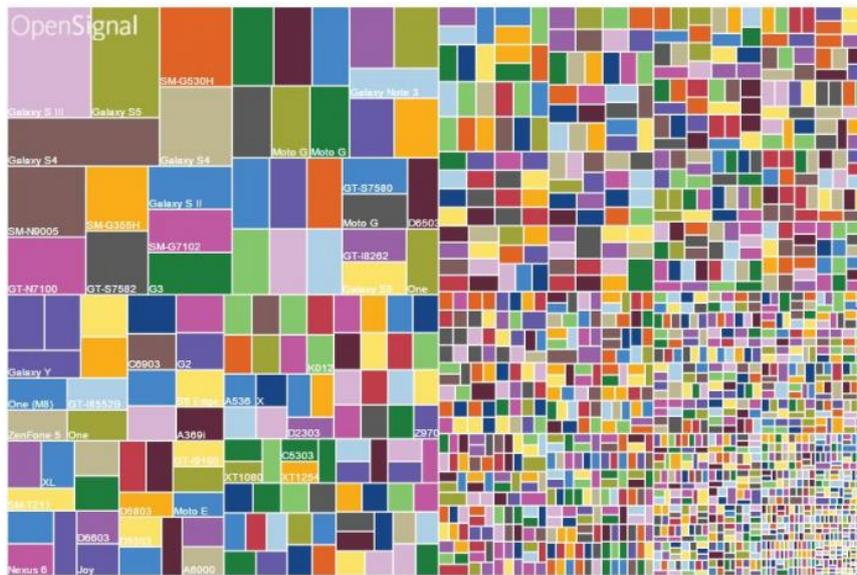


Ilustración 56: Fragmentación de dispositivos (Agosto 2015)

La seguridad de las aplicaciones es muy restrictiva en iOS por ello es más fácil probar las aplicaciones de terceros en Android. Sin embargo, las aplicaciones subidas a las plataformas de venta tanto de iOS como de Android garantizan la seguridad de nuestros dispositivos y no surge ningún problema para su instalación.

### 3.6.- Xcode

Es el entorno de desarrollo integrado con el que se va a crear la aplicación y el que proporciona Apple para desarrollar proyectos para sus dispositivos. Sustituyó a Interface Builder y le sirvió como base. Tiene una interfaz flexible que permite cambios para que se adapte a las necesidades del usuario y sea lo más cómodo para un buen trabajo. La ventana principal de Xcode está dividido en varias áreas de la siguiente manera:

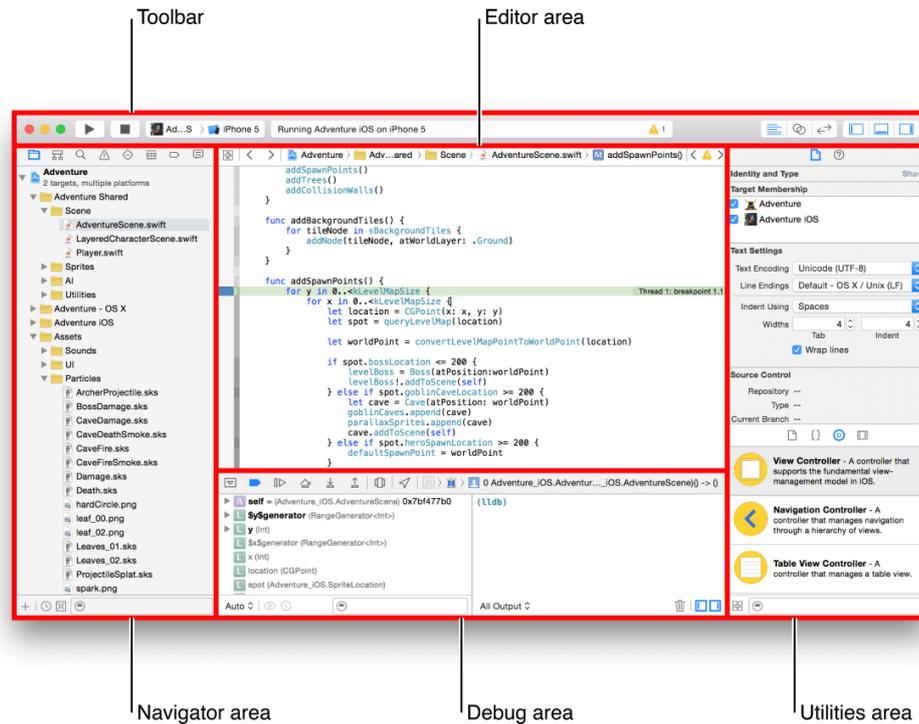


Ilustración 57: Ventana Xcode

**Toolbar:** es la barra de herramientas que permite la reproducción del proyecto y la configuración de las diferentes áreas de la ventana.

**Navigator area:** permite la visibilidad de los componentes del proyecto, así como facilita la selección o edición de cada una de ellas.

**Editor area:** es el área en el que se muestra el contenido del archivo que se haya seleccionado en el navigator area y de esa manera se podrá editar.

**Debug area:** se visualiza el estado del proyecto, los valores de las variables durante su ejecución para poder ir analizando.

**Utilities area:** proporciona información de los archivos y datos, también nos da recursos para poder añadir a nuestro código de una manera sencilla.

Un proyecto de Xcode sigue el patrón de arquitectura de software MVC (modelo-vista-controlador). La vista es la interfaz que se muestra al usuario, el modelo contiene lo que se muestra en la vista y le da funcionalidad a la misma y el controlador es el que comunica a la vista y al modelo.

Cuando se crea un proyecto nuevo en Xcode se genera el siguiente contenido:

**AppDelegate:** este fichero es el que delega el comportamiento básico de la aplicación y donde están las funciones que se lanzan cuando se arranca, se cierra, etc.... El fichero, en un principio, tiene las funciones básicas y una explicación de cuándo se ejecuta cada una de estas funciones, puede ser modificado para controlar el comportamiento de nuestra aplicación.

**Main.storyboard:** contiene todos los ficheros de la interfaz de la aplicación. Este fichero se muestra de manera gráfica pudiendo ver todas las vistas que componen la aplicación y la interacción entre ellas. Se puede editar e incluir elementos, vistas y todos los componentes que se deseen. También se pueden añadir las restricciones para adaptar los tamaños de los elementos a las pantallas de los dispositivos en los que se ejecutará la aplicación.

**ViewController:** este fichero representa al controlador de vista, que estará asociado a la vista principal. Según se van creando más vistas se tienen que crear sus correspondientes controladores de vista. Este primer controlador

de vista tiene creadas dos funciones vacíos *viewDidLoad()* y *didReceiveMemoryWarning()*. Estas y otras funciones de los controladores de vista serán explicados más adelante en este capítulo.

**Supporting files:** es una carpeta donde se encuentra un fichero llamado *info.plist* en el que viene determinado el comportamiento de la aplicación mediante diferentes claves y atributos. Estos valores pueden ser modificados desde diferentes lugares del proyecto y automáticamente se actualizan en el fichero.

**Images.xcassets:** aquí se almacenan las imágenes que se utilizan en la aplicación en diferentes tamaños y resoluciones para usar en los distintos dispositivos. También se incluye aquí el icono de la aplicación, que es el único obligatorio que debe estar aquí, las demás imágenes son opcionales.

**Launchscreen.xib:** es el fichero que almacena la vista estática que se muestra cuando se inicia la aplicación. Se visualiza igual que el *Main.storyboard* aunque no tiene tantas posibilidades ya que solo se trata de una única vista.

### 3.7.- Lenguajes de Programación iOS

Para programar aplicaciones para el sistema iOS se tienen 4 lenguajes de programación posibles para elegir, C, C++, Objective-C y Swift. Principalmente se solía programar en Objective-C y desde 2014 se disputa la programación de iOS junto con el nuevo lenguaje Swift. Swift nació como un lenguaje nuevo y definitivo que sustituiría a Objective-C, pero hasta que esto ocurra las dos conviven e incluso son compatibles.

Teniendo dos lenguajes para programar nos vemos en la situación de tener que elegir, si ya estamos familiarizados con uno de ellos la elección parece clara, pero si por el contrario hay que aprender un lenguaje se tendrán que analizar los pros y contras de cada uno de ellos.

Respecto a la potencia y rendimiento gana el lenguaje Swift, esto se debe a que este tiene menos capas que Objective-C y por ello se encuentra más cerca del hardware, lo que se traduce como mejor rendimiento.

En cuanto a la seguridad, Swift es muy estricto aunque eso implique que sea más complejo que Objective-C. Una diferencia que se destaca en este punto es que con Objective-C hubo un problema de certificación SSL por culpa de un `else` que se coló en el código en medio de `ifs`, que con Swift no hubiese ocurrido porque el lenguaje es más robusto y estricto y por ello en caso de error se hubiese avisado a la hora de compilar. Swift no permite ni punteros vacíos y no controlados, ni desbordamientos en buffer, ni variables no inicializadas, y otros muchos errores que con Objective-C se pueden cometer y no nos damos cuenta hasta que se ejecuta, sin embargo Swift nos avisaría en tiempo de compilación. Así que aunque Swift sea más estricto y complejo nos garantiza una mayor seguridad que Objective-C.

Hay diferentes opiniones respecto al tema de la facilidad de aprendizaje y de lectura, así que lo mejor será observar las guías de los lenguajes y valorar cuál nos parece más intuitivo y fácil de aprender para nosotros.

Hasta ahora la mayoría se decidiría por el nuevo lenguaje Swift ya que las comparaciones le favorecen, pero hay que tener en cuenta que es un lenguaje nuevo y que eso implica posibles cambios ya que está en constante evolución.

Al igual que al tener 2 años de vida los posibles problemas que nos encontremos sean más difícil solucionarlos que si nos les encontrásemos en Objective-C. Este último está más utilizado y probado con lo que conlleva que haya más probabilidad de que la gente se haya encontrado con los mismos problemas y ya se haya encontrado una o varias soluciones.

A continuación se puede observar una tabla con los lenguajes de programación más utilizados con la posición en la que se encuentran en estos momentos y la comparación de la posición con la de hace un año:

Jul 2016	Jul 2015	Change	Programming Language	Ratings	Change
1	1		Java	19.804%	+2.08%
2	2		C	12.238%	-3.91%
3	3		C++	6.311%	-2.33%
4	5	▲	Python	4.166%	-0.09%
5	4	▼	C#	3.920%	-1.73%
6	7	▲	PHP	3.272%	+0.38%
7	9	▲	JavaScript	2.643%	+0.45%
8	8		Visual Basic .NET	2.517%	+0.09%
9	11	▲	Perl	2.428%	+0.62%
10	12	▲	Assembly language	2.281%	+0.75%
11	15	▲	Ruby	2.122%	+0.74%
12	13	▲	Delphi/Object Pascal	2.045%	+0.57%
13	10	▼	Visual Basic	2.012%	+0.07%
14	16	▲	Swift	1.960%	+0.73%
15	6	▼	Objective-C	1.881%	-1.46%

Ilustración 58: Ranking lenguajes de programación

Objective-C ha sufrido una gran caída en cuanto a su utilización para programar, en 2015 llegó a ser el 3º lenguaje de programación más usado pero después empezó a descender y se encuentra en el 15º puesto en Julio del 2016. La gran crecida que obtuvo hasta 2015 se debió a la popularidad de desarrollar aplicaciones para Apple. Su descenso viene por la influencia del nuevo lenguaje anunciado por la compañía y de la popularidad de desarrollar aplicaciones que en su mayoría se realizan para los dispositivos Android.

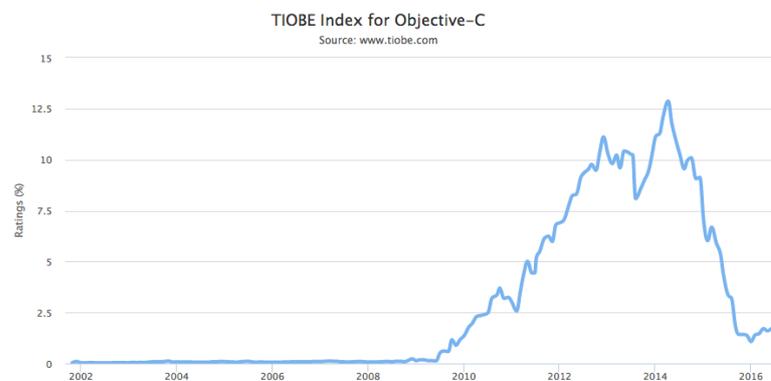


Ilustración 59: Gráfica evolución Objective-C

Swift ha sido un constante crecimiento y ahora se encuentra en un puesto por encima de Objective-C. Se prevé que sea el principal lenguaje de Apple y por ello su uso siga ascendiendo.

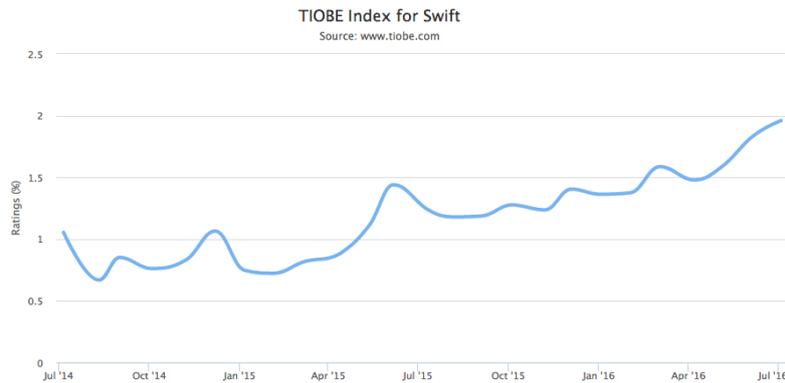


Ilustración 60: Gráfica evolución Swift

En mi caso tenía que aprender uno de los dos lenguajes de programación y observando las anteriores comparaciones opte por elegir Swift que es un lenguaje multiparadigma.

Las aplicaciones en Swift están formadas por una o varias vistas y cada una esta controlada por un controlador de vista. Durante la ejecución de una aplicación los controladores de vista se crean y eliminan como las vistas que se encienden y se apagan. Como esta práctica es muy común y hay que tenerlo bien controlado es esencial conocer como funcionan los controladores de vista, es decir su ciclo de vida. Ese ciclo de vida se muestra gráficamente a continuación:

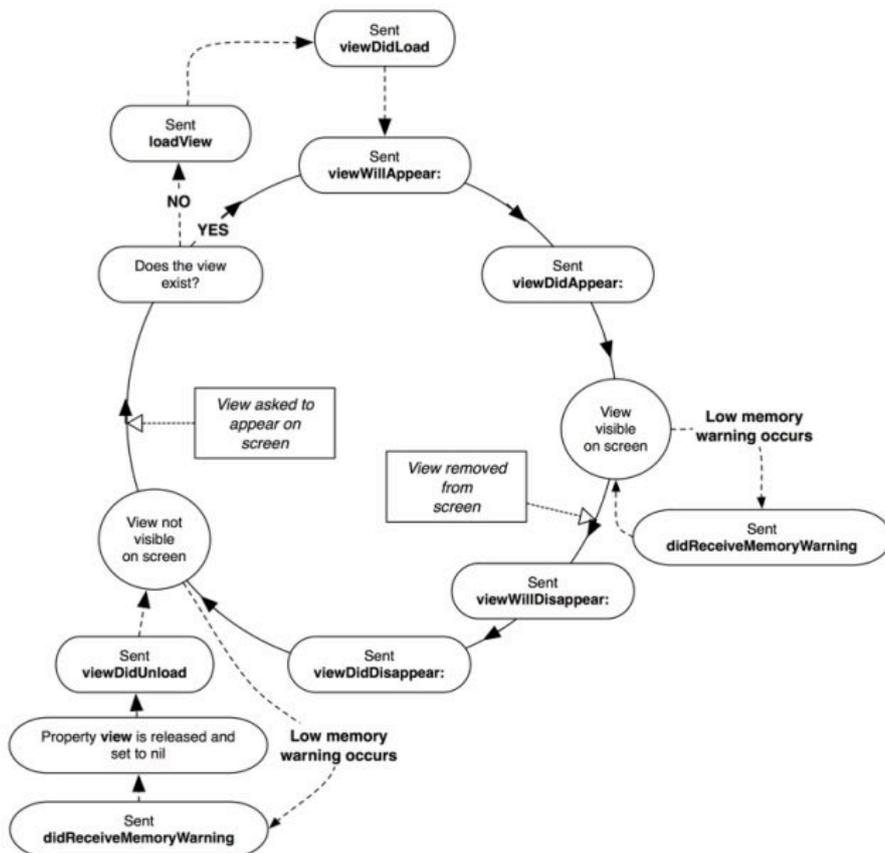


Ilustración 61: Ciclo de vida de un View Controller

***loadView()*** : este es el método que crea la vista del controlador de vista. Su implementación en **UIViewController** consiste en cargar la interfaz del archivo de interfaz y conectar todos los elementos y acciones que se encuentren en la vista.

***viewDidLoad()*** : antes de llamar a este método, la vista ya ha sido creada por ***loadView()***, esta función ya puede rellenar la vista con los datos que el usuario deberá ver en la misma. Se pueden realizar aquí las operaciones que solo se tengan que realizar una vez en la vista o se tengan que iniciar y posteriormente a lo largo de la vista necesiten actualización porque este método sólo se realiza una vez durante la vida del controlador de vista.

***viewWillAppear(\_ :)*** : este método se ejecuta antes de mostrar la vista y cada vez que se carga la vista por lo que se puede usar para actualizar la misma. También se pueden preparar datos o variables para cuando se inicie la vista.

***viewDidAppear(\_ :)*** : se llama cuando la vista ya ha aparecido y se han cargado todos los elementos. Es útil para iniciar tareas que necesiten empezar nada más abrir la vista, como la reproducción de un sonido, contabilizar un tiempo, etc....

***didReceiveMemoryWarning()*** : se realiza este método cuando hay una advertencia de memoria en el dispositivo. Cuando la memoria comienza a llenarse, iOS no utiliza su espacio de disco duro sino que mueve los datos de la memoria. Si la aplicación se inicia utilizando demasiada memoria, iOS lo notificará. Este método advierte al controlador de vista que es el responsable de la gestión de recursos.

***viewWillDisappear(\_ :)*** : este método se ejecuta antes de que la vista sea eliminada pero ya se sabe que va a desaparecer.

***viewDidDisappear(\_ :)*** : después de que la vista sea eliminada se llama a este método. Este método es apropiado para finalizar tareas que se realizaban constantemente durante la vista como una escucha. Si no se paran aquí se seguirán ejecutando en un segundo plano.

Los métodos anteriores se podrán configurar e incluir en cada controlador de vista que se tenga en la aplicación para un óptimo funcionamiento de su correspondiente vista.

En estos últimos meses se viene hablando de que Android podría usar Swift para programar, eso está por ver, no hay nada 100% seguro, pero sería un gran avance para que las aplicaciones entre Android e iOS fuesen compatibles.

### 3.8.- Personalización de dispositivos Apple y Jailbreak

Los dispositivos de la compañía Apple no son nada flexibles a la hora de poder personalizarlos, nos tendremos que conformar con cambiar el aspecto del fondo de pantalla y poco más. Esta es una característica que les distingue mucho de los dispositivos con sistema Android y que los usuarios suelen valorar ya que estos últimos pueden modificar la barra de notificaciones, los iconos, las animaciones,... vamos absolutamente todo. Lo mismo pasa o pasaba con las aplicaciones, antes de Xcode7 nuestras propias aplicaciones no se podían probar ni en nuestro teléfono si no era con una licencia. Ahora se puede probar pero con restricciones ya que se necesita tener un Mac y tener instalado Xcode.

Por todo esto la gente optaba por hacer jailbreak en sus dispositivos. Jailbreak consiste en saltarse o eliminar las limitaciones que Apple tiene en los dispositivos iOS mediante un kernel modificado. Esta modificación

### 3.- SISTEMA iOS | AudiOS

permite acceder al sistema operativo, descargar aplicaciones que no se encuentren en Apple Store o personalizar el dispositivo como se podría hacer en un dispositivo Android.

Hay dos maneras de tener el dispositivo con jailbreak uno es tethered el cual requiere que el dispositivo esté conectado a un ordenador cada vez que inicie y el otro es untethered que permite al dispositivo encender sin la asistencia de un ordenador.

Con el jailbreak tenemos el riesgo de que tengamos un dispositivo menos fiable, más propenso a errores y menos seguro. Las actualizaciones nuevas que salgan no se podrán realizar directamente en un dispositivo, si no se quiere perder el jailbreak, se debería hacer una copia de los datos del móvil, instalar la actualización y después volver a hacer jailbreak.

Según el Digital Millennium Copyright Act esta práctica es legal en Estados Unidos por ahora, pero en España con el nuevo código penal del año 2015 si podría ser ilegal. Apple dice que si tienes tu dispositivo con jailbreak no tendrías la garantía del mismo.

Así que cada usuario deberá valorar que prefiere o que cree que debe hacer con sus dispositivos.

## 4.- FUNDAMENTOS TEÓRICOS

---

### 4.1.- Ondas

Una onda es una perturbación que se propaga desde el punto en que se produjo hacia el medio que rodea ese punto. La teoría de ondas se conforma como una característica rama de la física que se ocupa de las propiedades de los fenómenos ondulatorios independientemente de cual sea su origen físico (Ostrovsky y Potapov, 1999). Una peculiaridad de estos fenómenos ondulatorios es que a pesar de que el estudio de sus características no depende del tipo de onda en cuestión, los distintos orígenes físicos que provocan su aparición les confieren propiedades muy particulares que las distinguen de unos fenómenos a otros. Las diferencias en el origen o naturaleza de las ondas producen ciertas propiedades que caracterizan cada onda, manifestando distintos efectos en el medio en que se propagan.

Todas las ondas se componen de unos elementos que indican el tipo de onda y su comportamiento. Esos elementos son:

**Cresta:** es el punto de máxima amplitud de la onda.

**Período ( $T$ ):** es el tiempo que tarda la onda en ir de una cresta a la siguiente.

**Amplitud ( $A$ ):** es la distancia entre la cresta y el punto medio de la onda.

**Frecuencia ( $f$ ):** es el número de oscilaciones que realiza la onda por unidad de tiempo.

**Valle:** es el punto más bajo de una onda.

**Longitud de onda ( $\lambda$ ):** es la distancia entre dos crestas consecutivas.

**Nodo:** es el punto donde la onda se corta con la línea de equilibrio.

**Elongación ( $x$ ):** es la distancia perpendicular entre un punto de la onda y la línea de equilibrio.

**Ciclo:** es una oscilación.

**Velocidad de propagación ( $v$ ):** es la velocidad a la que se propaga la onda.

Y todas las ondas pueden experimentar los fenómenos de difracción, que es cuando una onda choca contra un objeto y este deja de ir en línea recta y empieza a rodear el objeto, el efecto doppler, que es el aparente cambio de frecuencia de una onda provocado por el movimiento del emisor respecto al observador, la interferencia, que es la superposición de ondas formando una resultante, etc....

Las ondas se pueden clasificar de diferentes maneras. En función del medio en el que se propagan, de su dirección, del movimiento de sus partículas o de su periodicidad.

#### 4.1.1.- Ondas Sonoras

Una onda sonora es una onda longitudinal que transmite lo que se asocia con sonido. El sonido es cualquier fenómeno que involucre la propagación en forma de ondas elásticas, que sean audibles o no, generalmente a través de un fluido que esté generando el movimiento vibratorio de un cuerpo. La frecuencia de una onda se mide en hercios (Hz). Se considera que el oído humano puede percibir ondas sonoras de frecuencias entre los 20 y los 20.000 Hz, por ello las ondas con una frecuencia menor a la audible se denominan ondas infrasónicas y las ondas con frecuencias mayores ondas ultrasónicas.

Las ondas sonoras más simples y comunes son las ondas sinusoidales, que es una onda periódica que es creada por una oscilación. Esta onda representa gráficamente a la función seno, su forma más básica como una función del tiempo (t) es:

$$y(t) = A \sin(2\pi ft + \varphi) = A \sin(\omega t + \varphi)$$

Otras ondas sonoras más comunes son las ondas cuadradas, las cuales alternan su valor entre dos valores extremos sin pasar por los valores intermedios.

$$y = \begin{cases} A, & \text{frac}(ft)^* < 0.5 \\ -A, & \text{frac}(ft)^* > 0.5 \end{cases}$$

Las ondas triangulares que son periódicas y representan velocidades de bajada y subida constantes.

$$y = |2A \text{frac}(ft)^* - A|$$

Las ondas de diente sierra que gráficamente suben como una rampa y cuando llegan a su valor más alto descienden rápidamente hasta llegar al punto más bajo.

$$y = |2A \text{frac}(ft)^* - A|$$

\*frac(ft) es la parte fraccionaria de ft

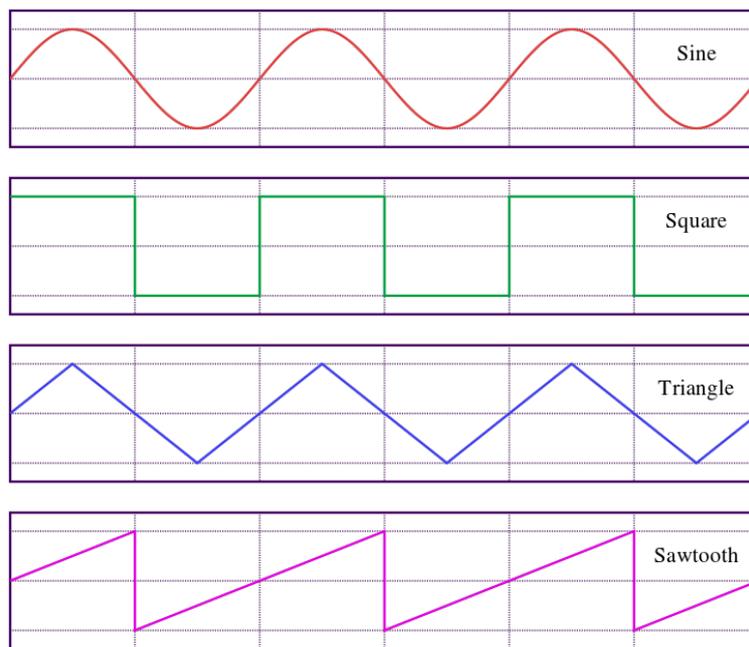


Ilustración 62: Ondas sonoras

Los ruidos también son sonidos pero, a diferencia de las ondas anteriores, se trata de combinaciones de todas las frecuencias sin ningún patrón ni fórmula a seguir. El ruido blanco es una señal aleatoria con densidad espectral de potencia constante, es decir, es una señal con todas las frecuencias con la misma potencia. El ruido rosa es una señal que tiene la densidad espectral inversamente proporcional a la frecuencia. El ruido marrón es una señal compuesto principalmente por frecuencias graves y medias.

### 4.1.2.- Ondas Estacionarias

Una onda estacionaria es el resultado de la superposición de dos movimientos ondulatorios armónicos de igual amplitud y frecuencia que se propagan en sentidos opuestos a través de un medio.

$$y(x, t) = 2A \cos(\omega * t) * \sin(k ** x)$$

$$* k = \frac{2\pi}{\lambda}$$

$$** \omega = 2\pi f = \frac{2\pi}{T}$$

Una de las principales características de las ondas estacionaria es que su longitud de onda no puede tener un valor aleatorio, sino sólo unos determinados valores que se relacionan con la longitud de la cuerda, mediante las siguientes expresiones:

$$\lambda_1 = 2L, \quad \lambda_2 = 2L/2, \quad \lambda_3 = 2L/3, \quad \lambda_4 = 2L/4, \dots \quad \lambda_n = 2L/n \quad (n = 1, 2, 3, \dots)$$

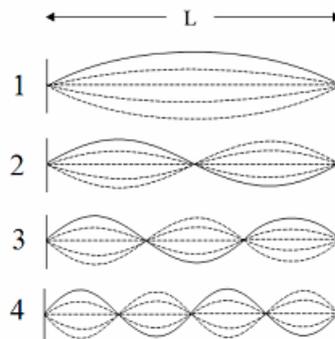


Ilustración 63: Valores de la longitud de una onda estacionaria

Y las frecuencias correspondientes son:

$$f_1 = v^*/2L, \quad f_2 = 2v^*/2L, \quad f_3 = 3v^*/2L, \quad f_4 = 4v^*/2L, \dots \quad f_n = nv^*/2L \quad (n = 1, 2, 3, \dots)$$

$$* v = \frac{\lambda}{T}$$

Estas frecuencias se llaman frecuencias de resonancia o frecuencias naturales del medio en el que se produce la onda. A la menor se la denomina frecuencia fundamental o "primer armónico" y las siguientes se designan sucesivamente como segundo armónico, tercer armónico, etc..

### 4.3.- Decibelios

Los decibelios (dB) son unidades de medidas relativas que se emplean en acústica, electricidad, telecomunicaciones y otras especialidades para expresar la relación entre dos magnitudes.

El decibelio es la medida utilizada para expresar el nivel de potencia y el nivel de intensidad del ruido. Se utiliza una escala logarítmica porque la sensibilidad que presenta el oído humano a las variaciones de intensidad sonora sigue una escala aproximadamente logarítmica, no lineal. Por ello el decibelio, resulta adecuado para valorar la percepción de los sonidos por un oyente. Se define como la comparación o relación entre dos sonidos porque en los estudios sobre acústica fisiológica se vio que un oyente, al que se le hace escuchar un solo sonido,

#### 4.- FUNDAMENTOS TEÓRICOS | AudiOS

no puede dar una indicación fiable de su intensidad, mientras que, si se le hace escuchar dos sonidos diferentes, es capaz de distinguir la diferencia de intensidad.

Como el decibelio es una unidad relativa, para las aplicaciones acústicas se asigna el valor de 0 dB al umbral de audición del ser humano, el verdadero umbral de audición varía entre distintas personas y para una misma persona, depende de la frecuencia del sonido. Se considera el umbral del dolor para el humano a partir de los 140 dB. Esta suele ser, aproximadamente, la medida máxima considerada en aplicaciones de acústica.

Nivel de intensidad del sonido	
200 dB	Bomba atómica similar a Hiroshima y Nagasaki
180 dB	Explosión del Volcán Krakatoa (a 160 km de distancia). Cohete en despegue
142.2 dB	Récord Guinness de ruido en un estadio
140 dB	Umbral del dolor. Coche de Fórmula 1
130 dB	Avión en despegue
120 dB	Motor de avión en marcha. Pirotecnia.
110 dB	Concierto. Acto cívico
100 dB	Perforadora eléctrica
90 dB	Tráfico
80 dB	Tren
70 dB	Aspiradora
50/60 dB	Aglomeración de gente / Lavaplatos
40 dB	Conversación
20 dB	Biblioteca

Tabla 45: Nivel de intensidad del sonido

El nivel de intensidad de sonido  $\beta$  de una onda sonora es una medida logarítmica de su intensidad y está definido como:

$$\beta = 10 \log \frac{I}{I_0^*}$$

$$* I_0 = 10^{-12} \text{ W/m}^2$$

Si la intensidad de una onda sonora es igual a  $10^{-12} \text{ W/m}^2$ , su nivel de intensidad de sonido es de 0 dB. Una intensidad de  $1 \text{ W/m}^2$  corresponde a 120 dB.

El oído no percibe igual las frecuencias y alcanza el máximo de percepción en las medias, por ello para aproximar más la unidad a la realidad auditiva, se ponderen las unidades utilizando las curvas isofónicas.

Las curvas isofónicas son curvas de igual sonoridad que calculan la relación entre la frecuencia y los decibelios de dos sonidos para que éstos sean percibidos como igual de fuertes por el oído, con lo que todos los puntos sobre una misma curva isofónica tienen la misma sonoridad.

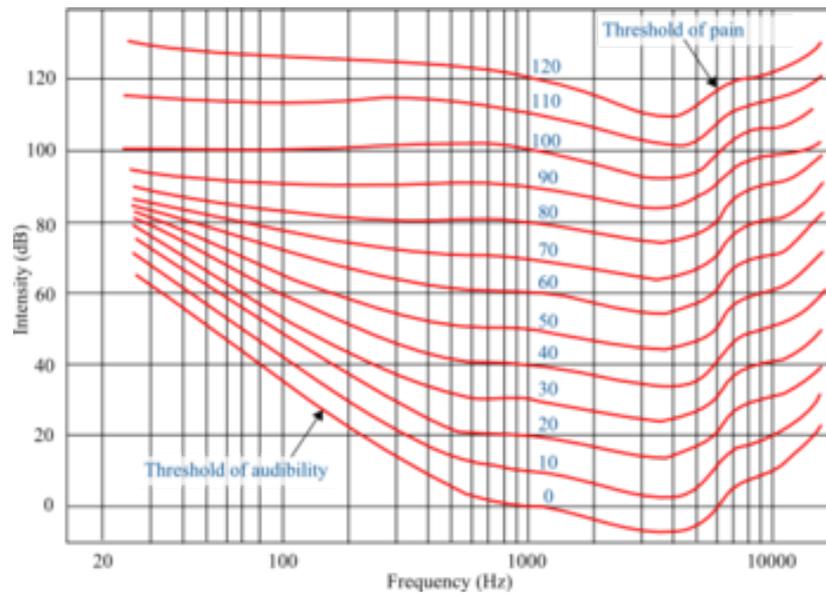


Ilustración 64: Primeras curvas isofónicas establecidas por Fletcher y Munson en 1930

Ya que el oído no percibe igual las frecuencias se definió el decibelio A (dBA), una unidad de nivel sonoro medido con un filtro previo que quita parte de las bajas y las muy altas frecuencias. De esta manera, después de la medición se filtra el sonido para conservar solamente las frecuencias más dañinas para el oído, razón por la cual la exposición medida en dBA es un buen indicador del riesgo auditivo y vital.

Hay además otras unidades ponderadas, como dBC, dBD, adecuadas para medir la reacción del oído ante distintos niveles de sonoridad.

Para estas unidades ponderadas también existen curvas isofónicas llamadas curva A, curva B y curva C para dBA, dBB y dBC, respectivamente.

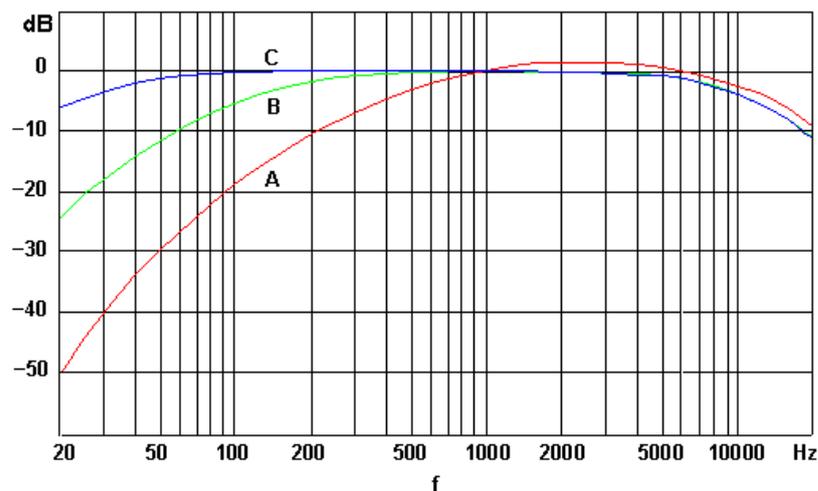


Ilustración 65: Curvas de ponderación A, B y C

Como el decibelio es adimensional y relativo, para medir valores absolutos se necesita especificar a qué unidades está referida la medida:

**dB SPL:** que hace referencia al nivel de presión sonora.

$$20 \log \frac{p^*}{p_0^{**}}$$

\**p* es la presión a calcular

\*\**p*<sub>0</sub> es la presión de referencia y se toma como referencia 20 μPa

**dBFS:** son los decibelios a escala completa, se trata de una abreviatura para los niveles de amplitud de decibelios en sistemas digitales que tienen un nivel máximo disponible. 0 dBFS se asigna al nivel máximo posible por lo que sus valores son todos negativos.

En algunos casos, al medir niveles relativos en decibelios, se da un nombre específico a la unidad, dependiendo del tipo de medida.

**dB C:** es el nivel relativo entre una señal portadora y alguno de sus armónicos.

**dB A:** son los decibelios ajustados.

#### 4.4.- Transformada de Fourier

La transformada de Fourier es una transformación matemática que transforma señales entre el dominio del tiempo (o espacial) y el dominio de la frecuencia, que tiene muchas aplicaciones en la física y la ingeniería. Es reversible, siendo capaz de transformarse en cualquiera de los dominios al otro. El propio término se refiere tanto a la operación de transformación como a la función que produce.

En el caso de una función periódica en el tiempo, la transformada de Fourier se puede simplificar para el cálculo de un conjunto discreto de amplitudes complejas, llamado coeficientes de las series de Fourier. Ellos representan el espectro de frecuencia de la señal del dominio-tiempo original.

La transformada de Fourier es una aplicación que hace corresponder a una función *f* de valores complejos y definida en la recta, con otra función *g* definida como:

$$g(\xi) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} f(x) e^{-i\xi x} dx$$

La transformada de Fourier así definida goza de una serie de propiedades de continuidad que garantizan que puede extenderse a espacios de funciones mayores e incluso a espacios de funciones generalizadas.

Para el procesamiento de señales hay que considerar que la señal de entrada es discreta y de duración finita. Y para ello se necesita utilizar la Transformada Discreta de Fourier (DFT), que requiere que la función de entrada sea una secuencia discreta y de duración finita.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} k n}$$

\**i* es la unidad imaginaria

\*\**k* = 0, ..., *N* - 1

##### 4.4.1.- Ventanas

Las ventanas son funciones matemáticas usadas con frecuencia en el análisis y el procesamiento de señales para evitar las discontinuidades al principio y al final de los bloques analizados.

En el procesamiento de señales, una ventana se utiliza cuando el análisis se centra en una señal de longitud voluntariamente limitada. En efecto, una señal real tiene que ser de tiempo finito; además, un cálculo sólo es posible a partir de un número finito de puntos. Para observar una señal en un tiempo finito, se multiplica por una función ventana.

La utilización de una ventana cambia el espectro en la frecuencia de la señal. Existen distintos tipos de ventana que permiten obtener distintos resultados en el dominio de las frecuencias.

Algunas ventanas en su forma discreta de tamaño  $N$ , donde  $0 \leq n \leq N - 1$  son las siguientes:

**Rectangular:** aplicar esta ventana como no utilizar ninguna ventana porque la función rectangular trunca la señal dentro de un intervalo finito de tiempo. La ventana rectangular es la que tiene mayor número de salidas espectrales.

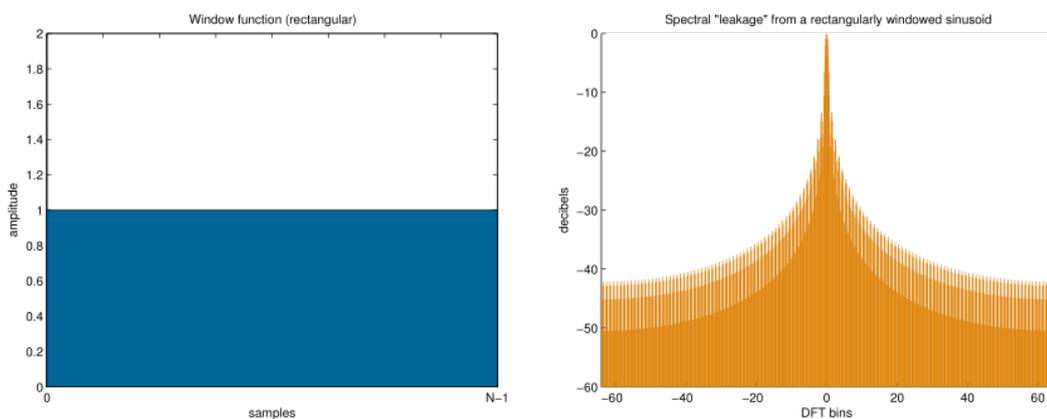


Ilustración 66: Ventana rectangular

$$v(n) = 1$$

**Bartlett o Triangular:** es una convolución de dos ventanas rectangulares, posee un pico espectral bueno y preciso similar al de la ventana Hamming. Hace un buen trabajo reduciendo pérdidas espectrales pero no tan bueno como el de la Hanning.

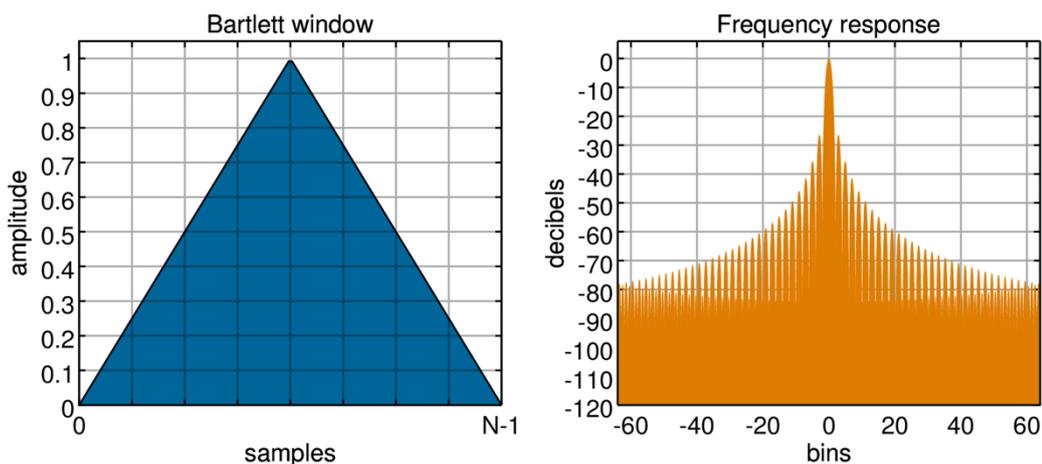


Ilustración 67: Ventana bartlett o triangular

$$v(n) = \frac{N - 1}{2} - \left| n - \frac{N - 1}{2} \right|$$

**Hamming:** tiene un comportamiento temporal de medio ciclo de una señal cosenoidal y normalizado en amplitud a la unidad. Es la ventana que tiene el lóbulo central más ancho pero la amplitud de los lóbulos laterales es prácticamente nula lo que minimiza la dispersión espectral.

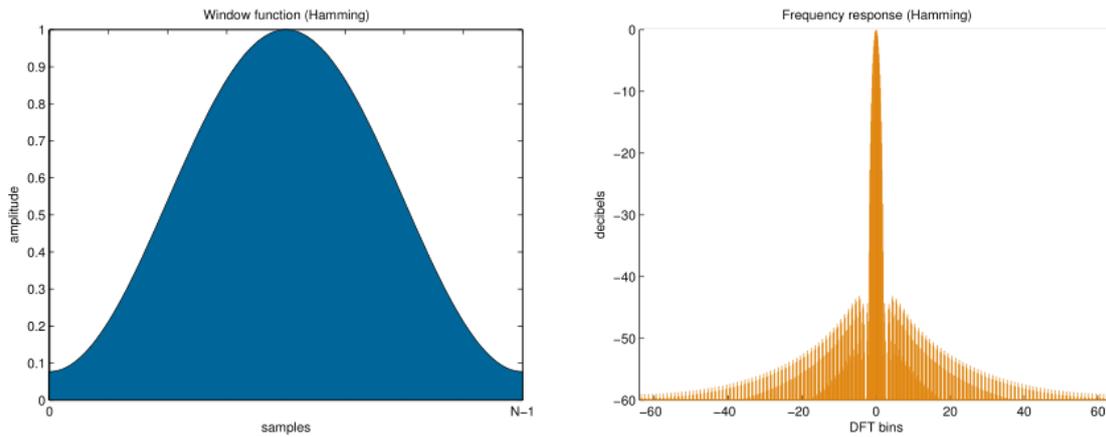


Ilustración 68: Ventana hamming

$$v(n) = 0,53836 - 0,46164 \cos\left(\frac{2\pi n}{N - 1}\right)$$

**Hanning:** tiene un comportamiento temporal de medio ciclo de una señal cosenoidal y normalizado en amplitud a la unidad. Posee un buen pico de lóbulo principal que permite tener un espectro de buena resolución, brinda además, muy buena reducción de pérdidas no deseadas en el espectro. La ventana Hanning se basa en una combinación de dos ventanas más simples; la suma de una rectangular y una coseno. Mejora el decaimiento de altas frecuencias, aunque posee un nivel máximo de lóbulo lateral grande.

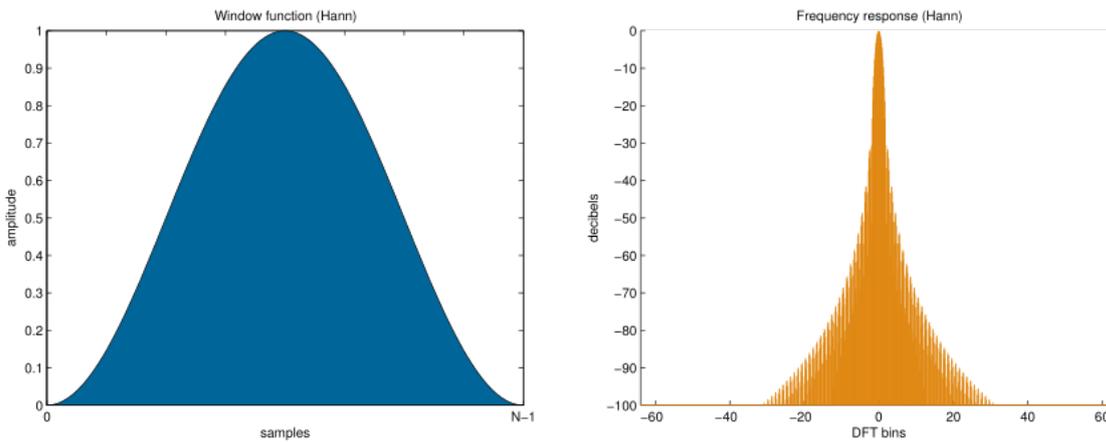


Ilustración 69: Ventana hanning

$$v(n) = 0,5 - 0,5 \cos\left(\frac{2\pi n}{N - 1}\right)$$

**Blackman:** tiene un comportamiento temporal definido por una serie trigonométrica de Fourier para aproximarse mejor a un espectro que carece de altas frecuencias. Es la suma de dos sinusoides, genera una gran mejora en términos de la amplitud de los lóbulos secundarios, pero tiene una mayor anchura del lóbulo principal.

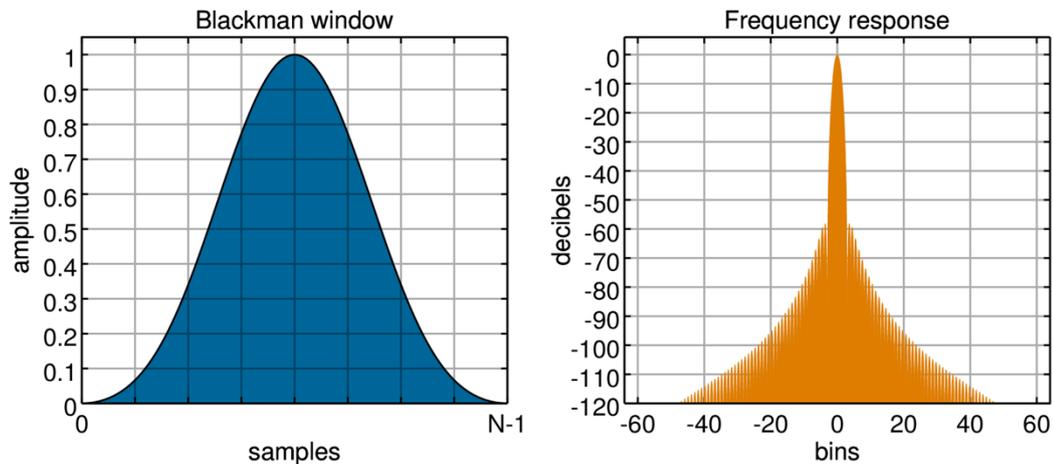


Ilustración 70: Ventana blackman

$$v(n) = 0,42 - 0,5 \cos\left(\frac{2\pi n}{N-1}\right) + 0,08 \cos\left(\frac{4\pi n}{N-1}\right)$$

**Blackman-Harris:** tiene un comportamiento temporal definido por una serie trigonométrica de Fourier para aproximarse mejor a un espectro que carece de altas frecuencias. Esta ventana tiene un espectro que requiere de un menor ancho de banda que la ventana Blackman.

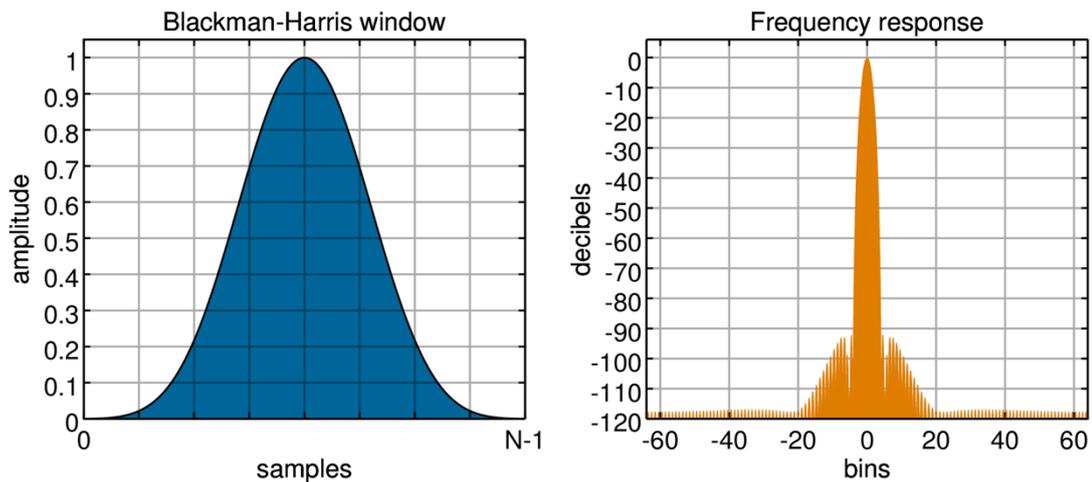


Ilustración 71: Ventana blackman-harris

$$v(n) = 0,35875 - 0,48829 \cos\left(\frac{2\pi n}{N-1}\right) + 0,14128 \cos\left(\frac{4\pi n}{N-1}\right) - 0,0118 \cos\left(\frac{6\pi n}{N-1}\right)$$

**Gauss:** dado que la función de Gauss se extiende hasta el infinito, tampoco se debe truncar en los extremos de la ventana, o sí con ventana de la otra ventana de composición cero. Puesto que el registro de un Gaussiana

produce una parábola , esto puede ser utilizado para la interpolación cuadrática casi exacta en la estimación de frecuencia.

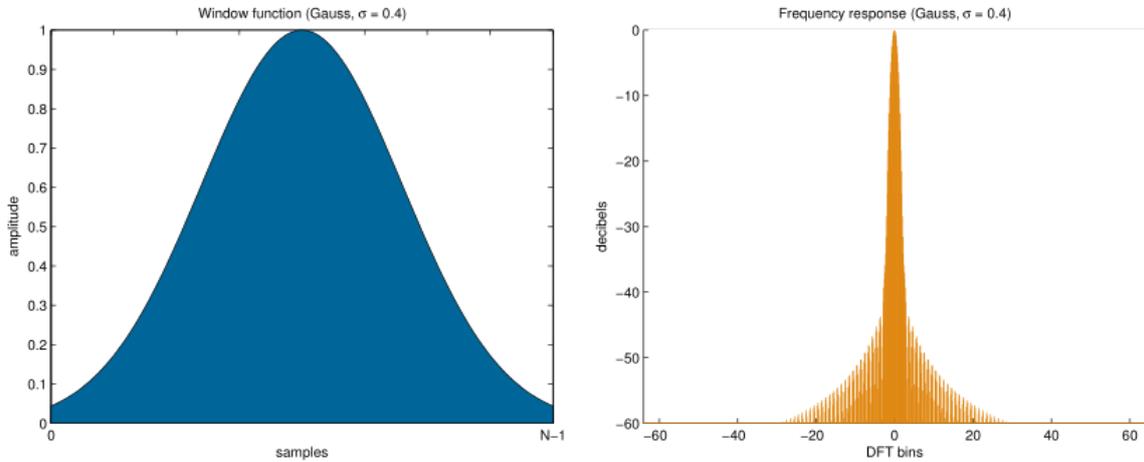


Ilustración 72: Ventana Gauss

$$v(n) = e^{-\frac{1}{2} \left( \frac{n - (N-1)/2}{\sigma \cdot (N-1)/2} \right)^2}$$

\*σ ≤ 0,5

**Flat Top:** es una ventana que tiene parcialmente valores negativos y que tiene una parte superior plana en el dominio de la frecuencia. Tales ventanas han sido puestas a disposición de los analizadores de espectro para la medición de las amplitudes de las componentes de frecuencia sinusoidales. Tienen un bajo error de medición de la amplitud adecuado para este propósito, logrado por la propagación de la energía de una onda sinusoidal a través de múltiples contenedores en el espectro. Esto asegura que la amplitud atenuada de la sinusoide se pueda encontrar en, al menos, una de los contenedores vecinos. El inconveniente es la anchura de banda ancha siendo pobre en la resolución de frecuencia. Para compensar, puede ser elegida una ventana de longitud más larga.

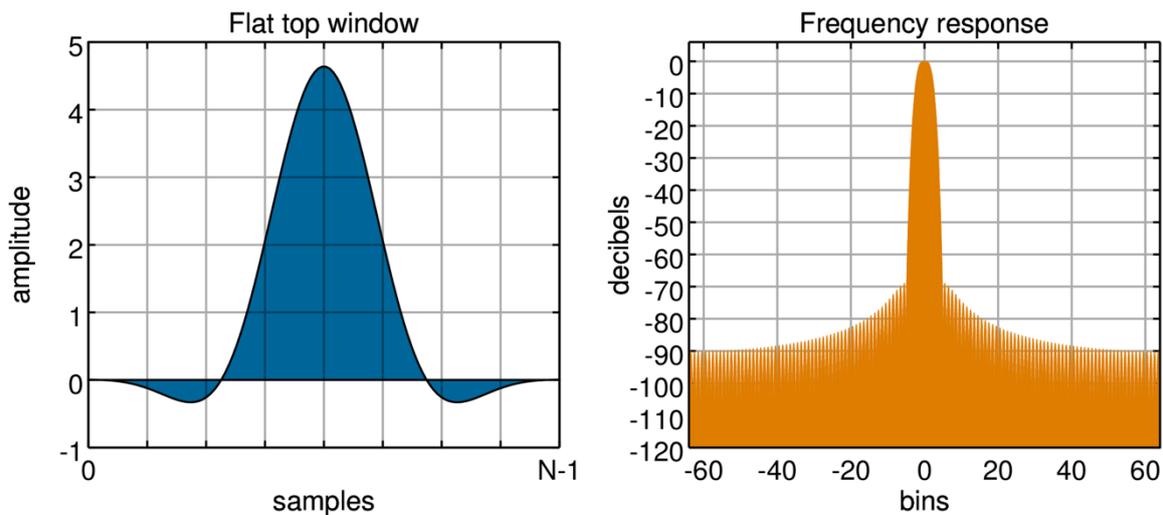


Ilustración 73: Ventana flat top

$$v(n) = 1 - 1,93 \cos\left(\frac{2\pi n}{N-1}\right) + 1,29 \cos\left(\frac{4\pi n}{N-1}\right) - 0,388 \cos\left(\frac{6\pi n}{N-1}\right) + 0,032 \cos\left(\frac{8\pi n}{N-1}\right)$$

Cuando la longitud de un conjunto de datos para ser transformado es más grande de lo necesario para proporcionar la resolución de frecuencia deseada, una práctica común consiste en subdividir en conjuntos más pequeños y ventana de ellos individualmente. Para mitigar la "pérdida" en los bordes de la ventana, los conjuntos individuales pueden solaparse en el tiempo.

El valor recomendado del solapamiento (ROV) depende del tipo de ventana, para la ventana rectangular es el 0% porque no se pierde nada al aplicarla, las ventanas bartlett, hanning y hamming es el 50%, para flat top es el 84%.

#### 4.5.- Calibración

La calibración es el proceso de comparar los valores obtenidos por un instrumento de medición con la medida correspondiente de un patrón de referencia.

De esta definición se puede deducir que para calibrar un instrumento se necesita disponer de uno de mayor precisión que proporcione el valor convencionalmente verificable, el cual se utilizará para compararlo con la indicación del instrumento que está siendo sometido a la calibración. Esto se realiza mediante una cadena ininterrumpida y completamente documentada de comparaciones hasta llegar al patrón primario, que constituye lo que se conoce como trazabilidad.

El objetivo de la calibración es mantener y verificar el buen funcionamiento de los equipos, responder los requisitos establecidos en las normas de calidad y garantizar la fiabilidad y la trazabilidad de las medidas. Los instrumentos de medida requieren ser calibrados con más frecuencia cuanto más exactas sean sus muestras, es decir, cuanto menor sean las tolerancias de error.

#### 4.6.- Formatos para Guardar

**PCM:** son las siglas de la modulación por impulsos codificado. Es un procedimiento de modulación utilizado para transformar una señal analógica en una secuencia de bits (señal digital), método inventado por el ingeniero británico Alec Reeves en 1937.

Una trama o flujo MIC es una representación digital de una señal analógica en donde la magnitud de la onda analógica es tomada en intervalos uniformes denominados "muestras", y cada muestra puede tomar un conjunto finito de valores, los cuales se encuentran codificados. Los flujos de MIC tienen dos propiedades básicas que determinan su fidelidad a la señal analógica original: la frecuencia de muestreo, es decir, el número de veces por segundo que se tomen las muestras; y la profundidad de bit, 1 que determina el número de posibles valores digitales que puede tomar cada muestra.

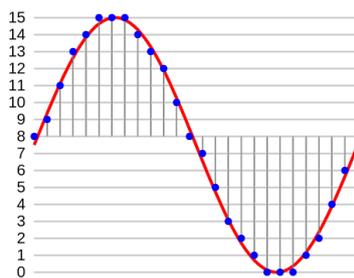


Ilustración 74: Muestreo en código PCM

**WAV o WAVE:** es un formato de audio digital normalmente sin compresión de datos desarrollado y propiedad de Microsoft y de IBM que se utiliza para almacenar sonidos en el PC, admite archivos mono y estéreo a diversas resoluciones y velocidades de muestreo. Es una variante del formato RIFF (Resource Interchange File Format, formato de fichero para intercambio de recursos), método para almacenamiento en "paquetes", y relativamente parecido al IFF y al formato AIFF usado por Macintosh.

A pesar de que el formato WAV es compatible con casi cualquier códec de audio, se utiliza principalmente con el formato PCM y, al no tener pérdida de calidad, es adecuado para uso profesional. Para tener calidad CD de audio se necesita que el sonido se grabe a 44100 Hz y a 16 bits. Por cada minuto de grabación de sonido se consumen unos 10 megabytes de espacio en disco. Una de sus grandes limitaciones es que solo se pueden grabar archivos de 4 gigabytes como máximo. Es una limitación propia del formato, independientemente del sistema operativo donde se utilice, y se debe a que en la cabecera del fichero se indica la longitud del mismo con un número entero de 32 bits, lo que limita el tamaño del fichero a un máximo de 4294967295 bytes.

**CSV:** son las siglas de comma-separated values. Es un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas o punto y coma en donde la coma es el Separador decimal y las filas por saltos de línea. Los campos que contengan una coma, un salto de línea o una comilla doble deben ser encerrados entre comillas dobles.

El formato CSV es muy sencillo y no indica un juego de caracteres concreto, ni cómo van situados los bytes, ni el formato para el salto de línea. Estos puntos deben indicarse muchas veces al abrir el archivo, por ejemplo, con una hoja de cálculo.

## 5.- IMPLEMENTACIÓN

---

El lenguaje que se ha usado para la implementación de la aplicación ha sido Swift por los motivos expuestos con anterioridad en los capítulos.

El entorno de desarrollo que se ha utilizado ha sido Xcode ya que es el que ofrece Apple para desarrollar código para su hardware. Esta plataforma es gratuita y se puede descargar e instalar en cualquier Mac.

La interfaz de usuario implementada para la aplicación es intuitiva y fácil de manejar para el usuario. En el caso de cualquier dificultad la mayoría de pantallas contienen un botón de ayuda para que el usuario pueda consultar y así poder resolver las posibles dudas o problemas que se encuentre durante la ejecución de la App. Para un buen funcionamiento de la aplicación se hacen las comprobaciones precisas, sobre todo en la introducción de datos por parte del usuario para mitigar los errores humanos con los que se pueda encontrar la aplicación.

La comunicación entre vistas se realizan mediante *segues* que son “flechas” que van de una vista a otra. Esas comunicaciones, a veces, se han complementado con código para pasar datos entre vistas, para decidir que vista mostrar dependiendo de un valor, etc....

***presentViewController(\_: animated: completion:)*** : esta función se llama cuando se quiere presentar un controlador de vista modal. Por lo general, no es necesario anularla, su aplicación estándar se encarga de todo. En la aplicación se ha usado principalmente para que las alertas creadas saliesen en pantalla.

***dismissViewControllerAnimated(\_: completion:)*** : al contrario que la función anterior, esta función se llama cuando se quiere quitar un controlador de vista que se ha presentado de forma modal. Se ha usado en la aplicación para que la alerta desapareciese de la pantalla cuando fuese necesario o cuando se pulsase un botón de la alerta.

***performSegueWithIdentifier(\_: sender:)*** : la presentación de un controlador de vista modal no es la única manera de pasar de un controlador de vista a otro. La forma más común es el uso de un controlador de vista de contenedor y la transición entre los controladores de vista utilizando *segues* de un guion. Un recipiente utilizado comúnmente es el controlador de navegación. Puede conectar los elementos de la interfaz de usuario de un controlador de vista directamente a *segues* que van a otros controladores de vista. Al hacer esto no es necesario que el usuario llame a esta función, ya que la *segue* se desencadena cuando interactúa con la interfaz de usuario. A veces, sin embargo, puede que tenga que realizar una transición suave después de un evento que no viene de la interfaz de usuario, por ejemplo, debido a la rotación del dispositivo o una devolución de llamada a la red. En estos casos es necesario llamar a este método con el identificador de la *segue* que desea realizar. Se configura el identificador de un *segue* en el guion gráfico.

Esta función se ha utilizado en la aplicación para, por ejemplo, cuando se tiene una lista de archivos y se elige uno, dependiendo del archivo que se haya elegido se tiene que ver una vista u otra.

***prepareForSegue(segue: sender:)*** : se activa después de un *segue* y sucede antes de la transición entre los dos controladores de vista involucrados. Esta función se puede variar para pasar datos de un controlador de vista al siguiente. El *segue* pasado como parámetro tiene referencias al controlador de vista de destino para poder acceder a él y así pasar los datos. Esta función ha sido usada en la aplicación para pasar datos, como pasar los resultados de un test de barrido a la vista de resultados o pasar el nombre de un archivo grabado a la ventana de señal capturada para que esta la muestre.

***popViewControllerAnimated(\_: completion:)*** : con esta función lo que se consigue es volver hacia atrás en la pila del control de navegación. Se ha introducido en la aplicación cuando se termina la ejecución de test de

## 5.- IMPLEMENTACIÓN | AudiOS

barrido se muestran los resultados de este y si se pulsa en botón de atrás en vez de ir a la pantalla de ejecución de test de barrido se va a la pantalla de test barrido, es decir, se salta 2 pantallas hacia atrás.

Para la implementación de algunas funciones de la aplicación se han usado funciones de terceros de código libre.

***ToneGenerator:*** se ha usado parte de código con modificaciones para conseguir la generación de las ondas en la aplicación.

***Charts:*** se ha usado para todo lo relacionado con las gráficas, como la creación, actualización o zoom en las mismas.

***AudiA:*** se han reutilizado fórmulas y funciones para los aspectos más relacionados con los fundamentos físicos.

## 6.- PRUEBAS

---

Caso de Prueba - 0001	
Descripción	Se pulsa un botón de la pantalla principal, herramientas, experiencias, etc...
Entrada	Pulsación botón.
Salida Esperada	Muestra la pantalla que corresponde.
Resultado	Correcto.

Tabla 46: Caso de prueba - 0001

Caso de Prueba - 0002	
Descripción	Se pulsa un botón de ayuda de cualquiera de las pantallas.
Entrada	Pulsación botón ayuda.
Salida Esperada	Muestra la pantalla de ayuda que corresponde.
Salida Obtenida	Correcto.

Tabla 47: Caso de prueba - 0002

Caso de Prueba - 0003	
Descripción	Se pulsa el botón de créditos de la pantalla principal.
Entrada	Pulsación botón créditos.
Salida Esperada	Muestra la pantalla de créditos.
Salida Obtenida	Correcto.

Tabla 48: Caso de prueba - 0003

Caso de Prueba - 0004	
Descripción	Play o pausar el sonómetro.
Entrada	Pulsación botón Play/Pausa.
Salida Esperada	El sonómetro se para o se ejecute.
Salida Obtenida	Correcto.

Tabla 49: Caso de prueba - 0004

Caso de Prueba - 0005	
Descripción	Grabar datos del sonómetro.
Entrada	Pulsación botón grabar.
Salida Esperada	Se graben los datos y aparezca el mensaje de enviar los resultados.
Salida Obtenida	Correcto.

Tabla 50: Caso de prueba - 0005

Caso de Prueba - 0006	
Descripción	Reiniciar el sonómetro.
Entrada	Pulsación botón reiniciar.
Salida Esperada	La gráfica se borra y comienza el trazado de una gráfica nueva.
Salida Obtenida	Correcto.

Tabla 51: Caso de prueba - 0006

Caso de Prueba - 0007	
Descripción	Variar el dB en pantallas que sea posible.
Entrada	Pulsación botón dB.
Salida Esperada	Los decibelios mostrados en pantalla varían al tipo mostrado en el botón.
Salida Obtenida	Correcto.

Tabla 52: Caso de prueba - 0007

Caso de Prueba - 0008	
Descripción	Play o pausar el espectrómetro.
Entrada	Pulsación botón Play/Pausa.
Salida Esperada	El espectrómetro se para o se ejecute.
Salida Obtenida	Correcto.

Tabla 53: Caso de prueba - 0008

Caso de Prueba - 0009	
Descripción	Cambiar las opciones del espectrómetro.
Entrada	Cambiar parámetros y pulsar botón aceptar.
Salida Esperada	Se ha cambiado la gráfica del espectrómetro.
Salida Obtenida	Correcto.

Tabla 54: Caso de prueba - 0009

Caso de Prueba - 0010	
Descripción	Generar el sonido de una onda sonora.
Entrada	Introducir parámetros y pulsar el botón de reproducir.
Salida Esperada	Sonido de la onda sonora.
Salida Obtenida	Correcto.

Tabla 55: Caso de prueba - 0010

Caso de Prueba - 0011	
Descripción	Grabar sonido.
Entrada	Elegir parámetro y pulsar el botón grabar.
Salida Esperada	Grabación del sonido escuchado por el altavoz del dispositivo.
Salida Obtenida	Correcto.

Tabla 56: Caso de prueba - 0011

Caso de Prueba - 0012	
Descripción	Realizar test de barrido de frecuencias.
Entrada	Pulsar botón siguiente.
Salida Esperada	Muestra resultados del test de barrido de frecuencias.
Salida Obtenida	Correcto.

Tabla 57: Caso de prueba - 0012

Caso de Prueba - 0013	
Descripción	Mostrar resultados de test de barrido de frecuencias guardados.
Entrada	Botón de test seleccionado.
Salida Esperada	Muestra los resultados del test seleccionado.
Salida Obtenida	Correcto.

Tabla 58: Caso de prueba - 0013

Caso de Prueba - 0014	
Descripción	Borrar resultado de test de barrido de frecuencias guardados.
Entrada	Desliz del test y pulsación de botón eliminar.
Salida Esperada	El test desaparece de la lista de test guardados.
Salida Obtenida	Correcto.

Tabla 59: Caso de prueba - 0014

Caso de Prueba - 0015	
Descripción	Realizar una de las 4 experiencias.
Entrada	Pulsación del botón reproducir experiencia.
Salida Esperada	Comenzará a sonar y generarse la experiencia.
Salida Obtenida	Correcto.

Tabla 60: Caso de prueba - 0015

Caso de Prueba - 0016	
Descripción	Realizar calibración del micrófono.
Entrada	Pulsación botón comenzar.
Salida Esperada	Se iniciará la calibración mostrando los dB.
Salida Obtenida	Correcto.

Tabla 61: Caso de prueba - 0016

Caso de Prueba - 0017	
Descripción	Guardar calibración del micrófono.
Entrada	Introducción de valores y pulsación botón guardar.
Salida Esperada	Se guarda los valores de calibración introducidos.
Salida Obtenida	Correcto.

Tabla 62: Caso de prueba - 0017

Caso de Prueba - 0018	
Descripción	Realizar calibración de los altavoces.
Entrada	Pulsación botón comenzar asistente.
Salida Esperada	Se iniciará la calibración mediante un asistente.
Salida Obtenida	Correcto.

Tabla 63: Caso de prueba - 0018

Caso de Prueba - 0019	
Descripción	Guardar nueva configuración.
Entrada	Cambiar parámetros y pulsación botón aceptar.
Salida Esperada	Guarda los nuevos valores de la configuración.
Salida Obtenida	Correcto.

Tabla 64: Caso de prueba - 0019

Caso de Prueba - 0020	
Descripción	Visualizar datos guardados en experiencias o herramientas.
Entrada	Seleccionar fichero.
Salida Esperada	Muestra los datos del fichero seleccionado.
Salida Obtenida	Correcto.

Tabla 65: Caso de prueba - 0020

Caso de Prueba - 0021	
Descripción	Borrar fichero guardado en experiencias o herramientas.
Entrada	Desliz del fichero y pulsación de botón eliminar.
Salida Esperada	El fichero desaparece de la lista de datos guardados.
Salida Obtenida	Correcto.

*Tabla 66: Caso de prueba - 0021*

## 7.- CONCLUSIÓN

---

### 7.1.- Conclusiones

La aplicación final obtenida es un gran recurso y ayuda para estudiantes de física por ser muy completa. También porque es de fácil usabilidad para cualquier usuario de dispositivos Apple.

Como este proyecto nació como necesidad de dar una aplicación similar a AudiA para dispositivos iOS creo que se ha cumplido con las expectativas en cuanto a similitud y funcionalidad. Ha habido dos modificaciones respecto a AudiA por petición de usuarios finales, una de las modificaciones es la eliminación del test de curvas isofónicas ya que servía para evaluar la respuesta del oído de las personas pero no era una herramienta o experiencia para realizar experimentos didácticos en el laboratorio. La otra modificación ha sido añadir la funcionalidad de poder ver los datos que se guardan de las experiencias y herramientas, en AudiA solo se podía ver los resultados de los test de barrido de frecuencias y ahora con AudiOS se podrá ver o escuchar los datos de los ficheros que se guarden.

Ha sido una satisfacción personal poder sacar adelante un proyecto, que a mi parecer, es de un buen nivel y también obtener nuevos conocimientos de manera autodidacta y que me pueden ser muy útiles en un futuro.

No todo ha sido fácil, me he encontrado con muchas dificultades a lo largo de todo el proyecto, sobretodo con los temas que quedaban más fuera de mi alcance como los físicos. Todos estos problemas han tenido su parte positivo pudiendo aprender cosas de todos ellos.

### 7.2.- Líneas de Trabajo Futuro

Para mejorar la aplicación se pueden añadir nuevas funcionalidades o mejorar las ya existentes.

Para trabajos futuros una nueva experiencia sería poder conectar una cuerda a un dispositivo móvil y que la vibración que se produce en las cuerdas se viese reflejado en una gráfica.

También se podría añadir una herramienta que calculase el tiempo de reverberación y otra que se comportase como un osciloscopio.

En cuanto a las mejoras se podría hacer un calibrado más preciso al igual que las demás medidas.

Y una buena mejora para llegar a un mayor número de usuarios sería añadir varios idiomas a la aplicación.



# ANEXOS



## I ANEXO.- INSTALACIÓN

Como ya se ha comentado en el Capítulo 3.- Sistema iOS Apple es muy estricto con la seguridad y para la instalación de la aplicación es un poco complejo si no se encuentra en el App Store. Aunque también hay que decir que con las versiones anteriores a Xcode no se podía probar nuestra aplicación en el móvil y ahora por lo menos tenemos esa posibilidad aunque sea un poco engorroso.

La primer opción como ya he dicho es descargándola de la App Store, como de momento no va a estar disponible en esa tienda online, hay que quedarse con la segunda opción.

La segunda opción consiste en, lo primero, aunque supongo que todo usuario de Apple lo tiene, es crear un ID de Apple desde aquí: [https://appleid.apple.com/es\\_ES](https://appleid.apple.com/es_ES)

Una vez creado se deberá instalar Xcode en el ordenador, aquí está el segundo inconveniente para las personas que no tengan en sus ordenadores instalado OS X que es el sistema operativo de los Mac.

Para descargar Xcode basta con entrar en AppStore, buscar Xcode y dar a Instalar.

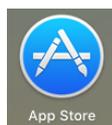


Ilustración 75: Aplicación App Store

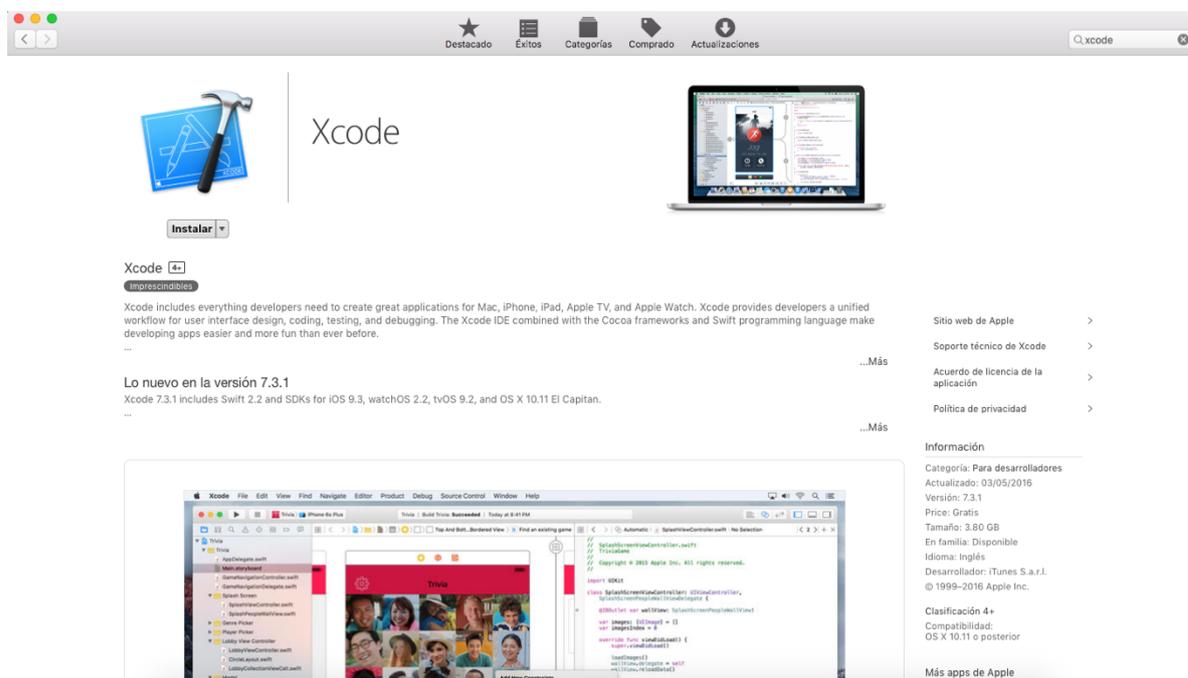


Ilustración 76: Descarga de Xcode desde App Store

Para abrir el proyecto de AudiOS solo habrá que hacer click en AudiOS.xcodeproj y se abrirá el proyecto en Xcode.



AudiOS.xcodeproj

Ilustración 77: Proyecto AudiOS

Una vez abierto se deberá conectar el móvil mediante un cable USB al ordenador y Xcode le reconocerá apareciendo en la barra de herramientas de la siguiente manera:

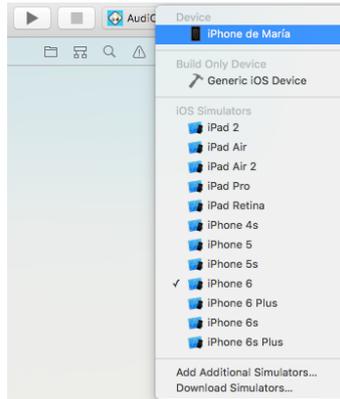


Ilustración 78: Detección del dispositivo en Xcode

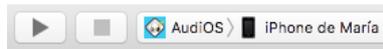
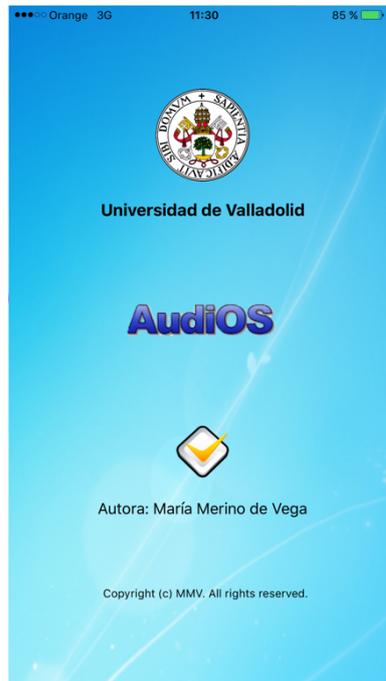


Ilustración 79: Selección de dispositivo y ejecución en el mismo

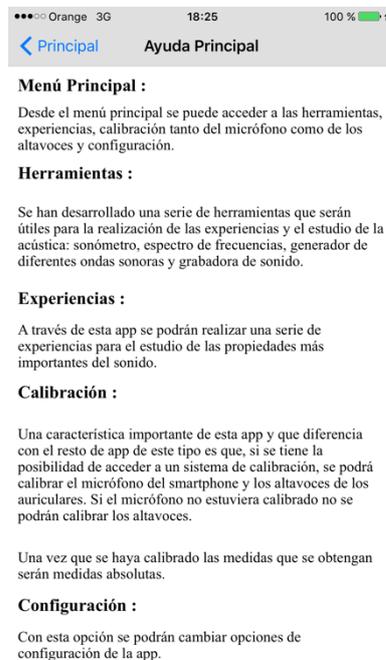
Selecciona tu dispositivo y lo único que queda es dar al play para que la aplicación se ejecute en él. El problema que nos encontramos con esta segunda opción es que transcurrida una semana no se puede acceder a ella en el dispositivo y se debería de volver a conectar al ordenador y volver a cargar la aplicación dando al botón play.

## II Anexo.- MANUAL DE USUARIO



*Ilustración 80: Pantalla de inicio*

Con este manual se pretende dar las pautas para la correcta utilización de la app AudiOS. En cada pantalla se dispone de una ayuda para podernos guiar por toda la aplicación.



*Ilustración 81: Pantalla de ayuda*

Desde el menú principal se puede acceder a las herramientas, experiencias, calibración tanto del micrófono como de los altavoces, configuración y a los datos numéricos guardados en las experiencias y herramientas.



Ilustración 82: Pantalla principal

## II.I.- Herramientas



Ilustración 83: Pantalla herramientas

Se han desarrollado una serie de herramientas que serán útiles para la realización de las experiencias y el estudio de la acústica: sonómetro, espectro de frecuencias, generador de diferentes ondas sonoras y grabadora de sonido.

### II.I.I- Sonómetro

Una característica importante de esta app es que puede funcionar como un sonómetro, de tal forma que si se ha podido calibrar el micrófono del dispositivo se tendrían medidas absolutas de los decibelios. En caso contrario, si no se tiene acceso a un sistema de calibración y el dispositivo no está calibrado todas las medidas obtenidas serán relativas.



Ilustración 84: Pantalla sonómetro

Se podrá cambiar el intervalo de muestreo, así como el tipo de decibelios que se muestran. Se podrán guardar los datos muestreados en un fichero.

En la pantalla, además de los valores de RMS y pico, se muestran los máximos. En la gráfica se van visualizando todos los valores RMS que va capturando. También existe la posibilidad de reiniciar los valores de máximos y los de la gráfica.

### II.I.II- Espectro de Frecuencias

A través de esta herramienta se puede ver el espectro de frecuencias de la señal de sonido que está capturando el dispositivo en el intervalo de muestreo seleccionado.

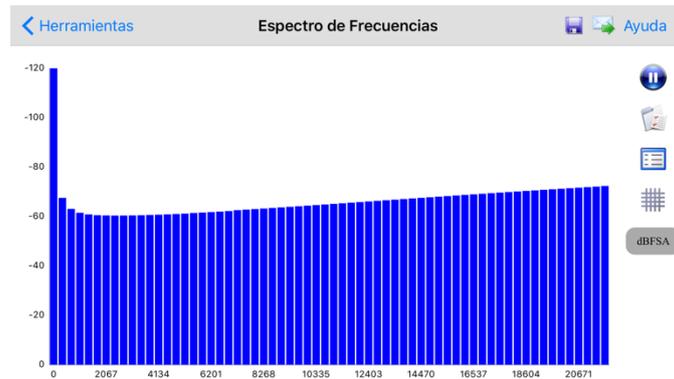


Ilustración 85: Pantalla espectro de frecuencias

Se puede guardar y enviar por correo los resultados en formato CSV.

Entre las opciones que se pueden modificar: intervalo de muestreo intervalos (35 ms, 125 ms, 1 s o 5 s), el tamaño de la ventana, el tipo de ventana, la escala de la gráfica (logarítmica o lineal), los dB mínimo a partir de los cuales se va a mostrar y el tipo de decibelios que se están mostrando.

Ilustración 86: Pantalla opciones espectro de frecuencias

Se puede hacer zoom en la gráfica y si se pulsa sobre la grafica aparece una línea vertical que indica los dB para esa frecuencia.

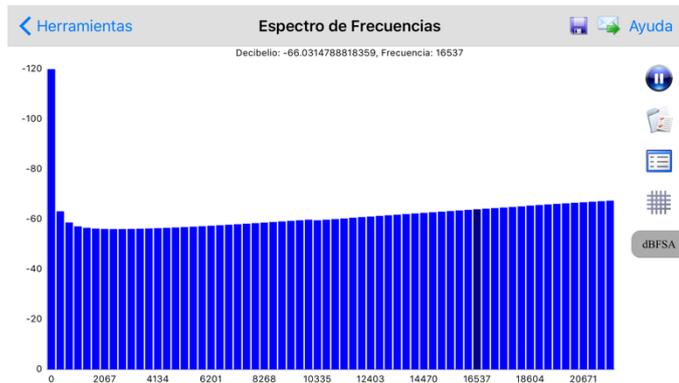


Ilustración 87: Pantalla espectro de frecuencias con información

También se pueden mostrar los valores del espectro en una tabla.

### II.I.III- Generador Onda Sonora

Ilustración 88: Pantalla generador onda sonora

Los datos que se solicitan son frecuencia inicial (Hz), frecuencia final (Hz), volumen inicial de generación del sonido (%), volumen final de generación del sonido (%), altavoz por el que saldrá el sonido, formato del fichero (WAV, PCM, CSV) y duración del sonido (s)

Con esta herramienta se puede generar o guardar en el formato seleccionado una onda sonora. Hay un desplegable para seleccionar entre onda sinusoidal, onda cuadrada, onda triangular, onda diente de sierra, ruido rosa, ruido blanco y ruido marrón. En el caso de guardar, cuando finaliza la grabación pregunta si se quiere enviar por correo.

### II.I.IV- Grabadora de Sonido



Ilustración 89: Pantalla grabadora de sonido

El dato que se solicita es el formato del fichero (WAV, PCM, CSV)

Se guardará un fichero con el sonido grabado en el formato seleccionado.

Cuando se para la grabación aparece una gráfica con los valores de la señal capturada. También se tendrá la opción de ver el espectro de frecuencias, así como poder enviar por correo los resultados.

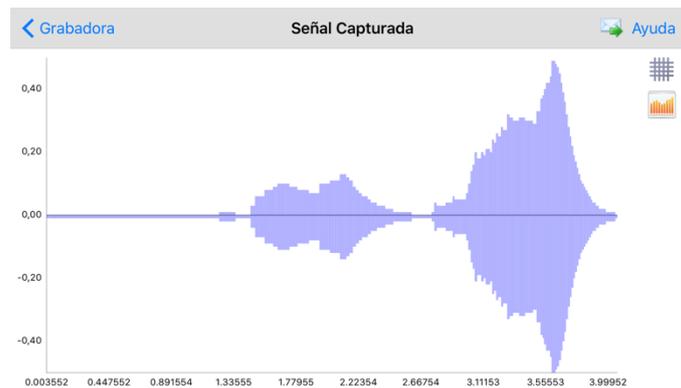


Ilustración 90: Pantalla señal capturada

Entre las opciones que se pueden modificar en el espectro de frecuencias: intervalo de muestreo intervalos (35 ms, 125 ms, 1 s o 5 s), el tamaño de la ventana, el tipo de ventana, la escala de la gráfica (logarítmica o lineal), los dB mínimo a partir de los cuales se va a mostrar y el tipo de decibelios que se están mostrando.

Se puede hacer zoom en la grafica y si se pulsa sobre la grafica aparece una línea vertical que indica los dB para esa frecuencia.

También se pueden mostrar, al igual que en el espectro de frecuencias, los valores del espectro en una tabla.

## II.II.- Experiencias



Ilustración 91: Pantalla experiencias

A través de esta app se podrán realizar una serie de experiencias para el estudio de las propiedades más importantes del sonido.

Las experiencias que se pueden realizar son el test de barrido de frecuencias, el estudio de las ondas estacionarias, efecto doppler, estudio de los batidos y de las interferencias y difracción.

Dependiendo de la experiencia habrá que introducir diferentes datos.

Para algunas de las experiencias solo tiene sentido realizarlas para ondas sinusoidales ya que son las que tiene la frecuencia completamente especificada.

Posteriormente y también dependiendo de la experiencia se tendrá la posibilidad de guardar en un archivo los datos obtenidos o grabar el fichero de audio con los resultados de la experiencia. En el caso de grabar el fichero de audio interesará estar lo más en silencio posible para que el micrófono no detecte ningún ruido ajeno a la experiencia.

### II.II.I- Test Barrido de Frecuencias



Ilustración 92: Pantalla test barrido de frecuencias

Primero se tendrá la opción de realizar el test de barrido de frecuencias o ver los resultados de test guardados.

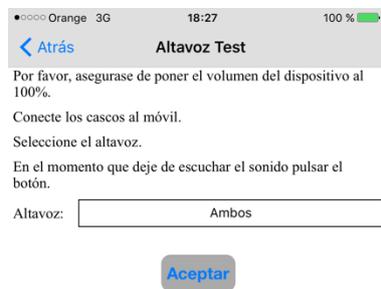


Ilustración 93: Pantalla inicio test

Si se desea realizar un test barrido de frecuencias se deberá elegir el altavoz por el que realizar el test.

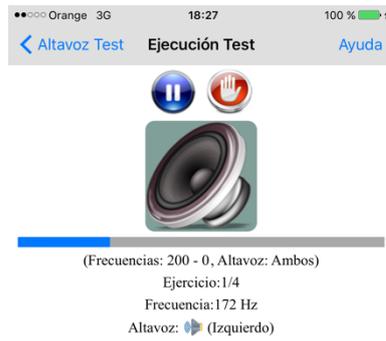


Ilustración 94: Pantalla ejecución test

Durante la ejecución del test cuando deje de escuchar el sonido pulsar el botón central. También se puede pausar el ejercicio que se está ejecutando y abortar el test.

Al finalizar el test se mostrará el resultado y se podrá guardar dicho test.

Si se desea ver un test ya guardado se mostrará una lista de los que hay.

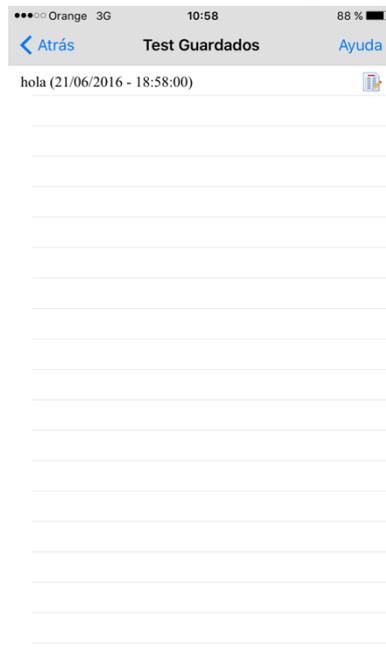


Ilustración 95: Pantalla test guardados

Para poder editar el test se deberá dar al botón que se encuentra a la derecha del nombre. Para poder eliminarlo solo hará falta deslizar la celda hacia la izquierda y dar al botón eliminar que aparecerá.



Ilustración 96: Pantalla eliminar test guardado

### II.II.II- Ondas Estacionarias



Ilustración 97: Pantalla ondas estacionarias

Se puede realizar el estudio de las ondas estacionarias con tubo de longitud fija o tubo de longitud variable.

Para las ondas estacionarias de longitud de tubo fijo los datos que se solicitan son frecuencia inicial (Hz), frecuencia final (Hz), volumen de generación del sonido (%), altavoz por el que saldrá el sonido, formato del fichero (WAV, PCM, CSV), duración que tardará en pasar de frecuencia inicial a la frecuencia final (s), longitud del tubo (mm) y diámetro del tubo (mm).

Al pulsar el botón de iniciar la experiencia se hará un barrido desde la frecuencia inicial a la final con la duración que se ha puesto. Al mismo tiempo que genera el sonido lo va grabando. Se podrá parar en cualquier momento y los datos que se tengan serán hasta ese momento.

Orange 3G 18:27 100 %

< Atrás Ondas Estacionarias (Fijo) Ayuda

Frecuencia Inicial (Hz): 400

Frecuencia Final (Hz): 3000

Volumen (%): 100

Altavoz: Ambos

Formato Fichero: WAV

Duración Sonido (s): 60

Longitud Tubo (mm): 150

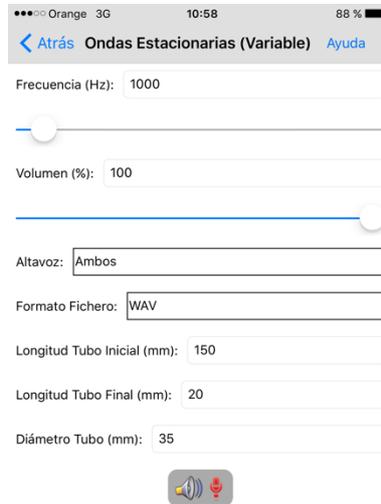
Diámetro Tubo (mm): 35

Speaker icon

Ilustración 98: Pantalla onda estacionaria de longitud de tubo fijo

Al finalizar aparece una gráfica con el resultado. También se tendrá la opción de ver el espectro de frecuencias y de enviar por correo los resultados. Se pueden realizar las mismas acciones que con el espectro de frecuencias.

Para las ondas estacionarias de longitud de tubo variable los datos que se solicitan son frecuencia (Hz), volumen de generación del sonido (%), altavoz por el que saldrá el sonido, formato del fichero (WAV, PCM, CSV), duración que tardará en pasar de frecuencia inicial a la frecuencia final (s), longitud del tubo inicial (mm), longitud del tubo final (mm) y diámetro del tubo (mm).



*Ilustración 99: Pantalla onda estacionaria de longitud de tubo variable*

Al pulsar el botón de iniciar la experiencia se hará un barrido desde la frecuencia inicial a la final con la duración que se ha puesto. Al mismo tiempo que genera el sonido lo va grabando. Se podrá parar en cualquier momento y los datos que se tengan serán hasta ese momento.

Al finalizar aparece una gráfica con el resultado. También se tendrá la opción de ver el espectro de frecuencias y de enviar por correo los resultados. Se pueden realizar las mismas acciones que con el espectro de frecuencias.

### II.II.III- Efecto Doppler

---

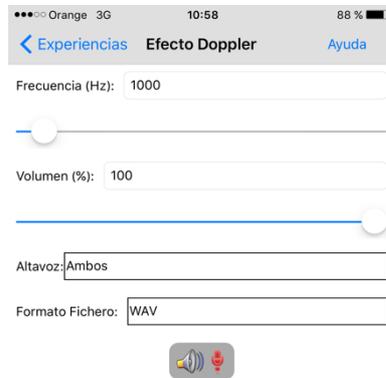


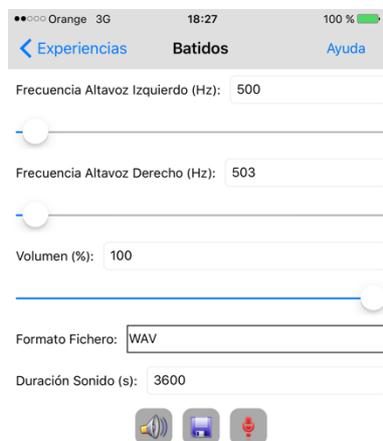
Ilustración 100: Pantalla efecto doppler

Los datos que se solicitan son frecuencia (Hz), volumen de generación del sonido (%), altavoz por el que saldrá el sonido y formato del fichero (WAV, PCM, CSV)

Al pulsar el botón de iniciar la experiencia empieza a emitirse un sonido a la frecuencia indicada y a la vez se está grabando.

Al finalizar aparece una gráfica con el resultado. También se tendrá la opción de ver el espectro de frecuencias y de enviar por correo los resultados. Se pueden realizar las mismas acciones que con el espectro de frecuencias.

## II.II.IV- Batidos



*Ilustración 101: Pantalla batidos*

Los datos que se solicitan son frecuencia altavoz izquierdo (Hz), frecuencia altavoz derecho (Hz), volumen de generación del sonido (%), formato del fichero (WAV, PCM, CSV) y duración del sonido (s).

Al poner dos frecuencias muy próximas, una por cada altavoz se produce el efecto de los batidos.

Si se pulsa el botón de generar se emitirá el sonido y se podrá comprobar el efecto de los batidos.

Si se pulsa el botón de guardar se guardará un fichero con los datos del sonido en el formato seleccionado. Cuando finaliza la grabación pregunta si se quiere enviar por correo.

Si se pulsa el botón grabar se grabará el sonido de la señal capturada. Al finalizar aparece una gráfica con el resultado. También se tendrá la opción de ver el espectro de frecuencias y de enviar por correo los resultados. Se pueden realizar las mismas acciones que con el espectro de frecuencias.

## II.II.V- Interferencias y Difracción

---



*Ilustración 102: Pantalla interferencias y difracción.*

Los datos que se solicitan son intervalo de muestreo (ms), frecuencia (Hz), volumen de generación del sonido (%) y altavoz por el que saldrá el sonido.

Si se pretenden estudiar las interferencias habrá que seleccionar los dos altavoces. En el caso de la difracción sólo habrá que seleccionar uno, el izquierdo o el derecho.

Cuando se pulsa el botón de generar empieza a emitirse el sonido y se mostrará los decibelios que está capturando. Se irá tomando nota de la distancia a la que está el Smartphone y los decibelios que está midiendo.

## II.III.- Calibración

---

Una característica importante de esta app y que diferencia con el resto de app de este tipo es que, si se tiene la posibilidad de acceder a un sistema de calibración, se podrá calibrar el micrófono del Smartphone y los altavoces de los auriculares.

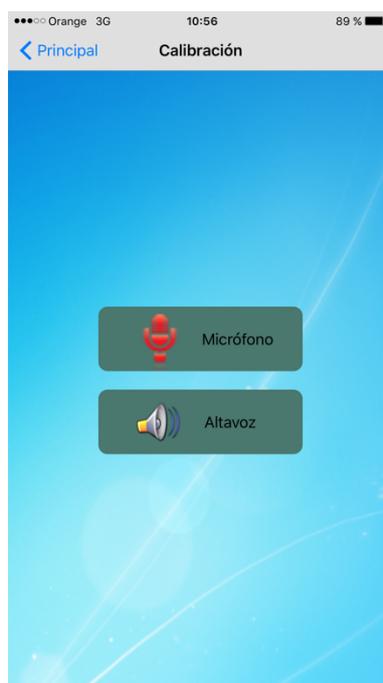


Ilustración 103: Pantalla calibración

Si el micrófono no estuviera calibrado no se podrán calibrar los altavoces. Una vez que se haya calibrado las medidas que se obtengan serán medidas absolutas.

### II.III.I- Calibración Micrófono

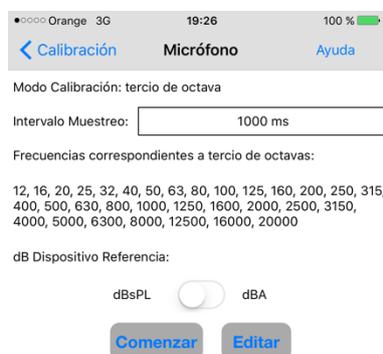


Ilustración 104: Pantalla calibración micrófono

Dependiendo del tipo de decibelios (dB SPL o dBA) que mide el dispositivo de referencia habrá que seleccionar uno u otro para que el Smartphone a calibrar nos de dBFS o dBFSa respectivamente.



Ilustración 105: Pantalla proceso calibración micrófono

Con una fuente externa se irán emitiendo sonidos a las frecuencias indicadas. Se anotará el valor obtenido con el dispositivo de referencia y el obtenido con el Smartphone. Cuando se pulse “Parar” finalizará de tomar medidas y aparecerá un formulario donde habrá que introducir los dos valores y pulsar “Guardar”.

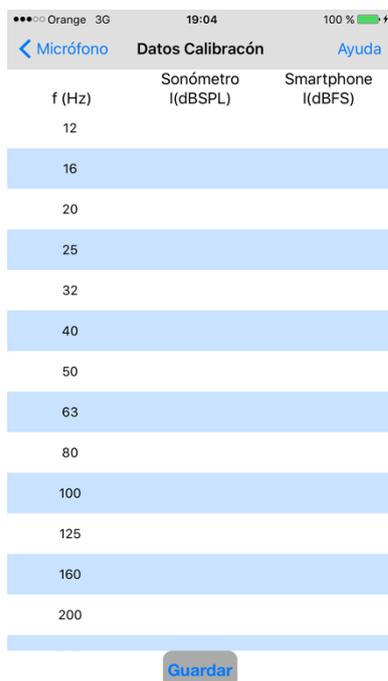


Ilustración 106: Pantalla resultados calibración micrófono

## II.III.II- Calibración Altavoces



Ilustración 107: Pantalla calibración altavoces

La calibración de los altavoces se realiza de forma automática. Se dispone de un asistente que le irá guiando por toda la calibración.

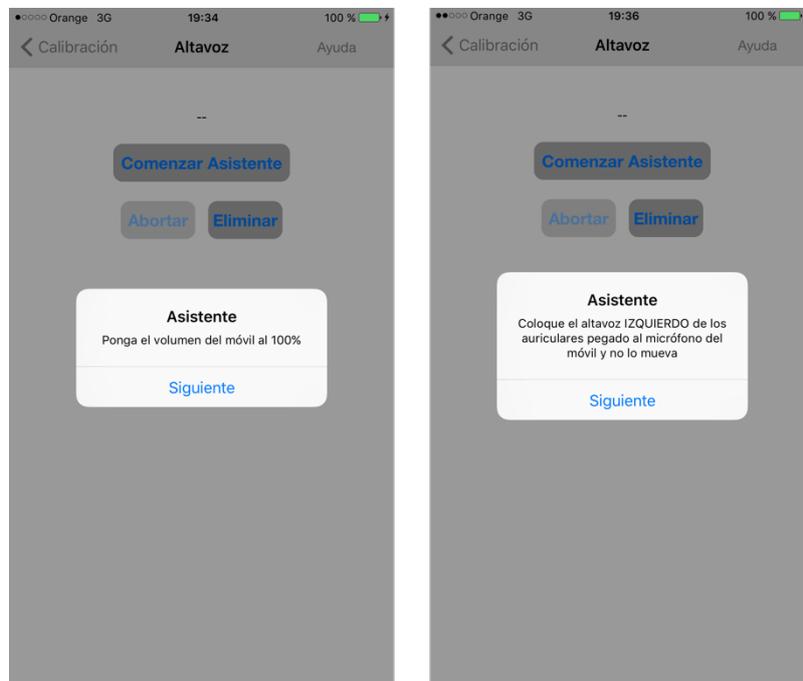


Ilustración 108: Asistente de la calibración de altavoces

Durante la ejecución de la calibración de los micrófonos se mostrará la frecuencia generada y los decibelios capturados por el dispositivo pudiendo abortar la calibración.

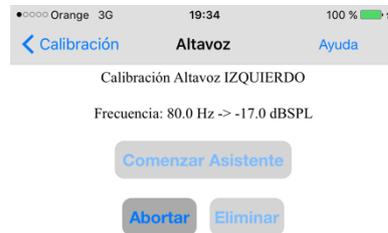


Ilustración 109: Pantalla proceso calibración altavoz

## II.IV.- Configuración

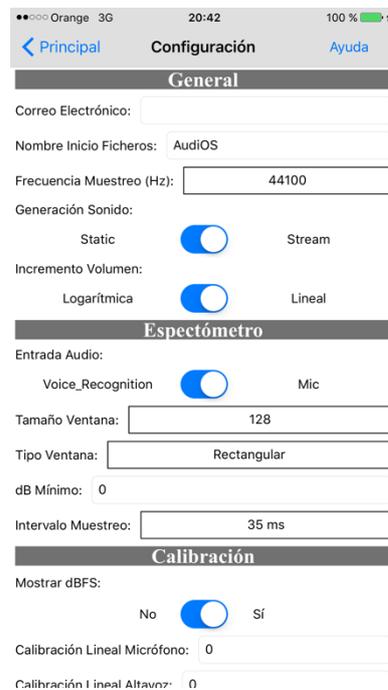


Ilustración 110: Pantalla configuración

Desde esta pantalla se podrán cambiar opciones de configuración de la app, como: cambiar el nombre con el que empiezan los resultados guardados tanto de las herramientas como de las experiencias, un correo electrónico para mandar los resultados, frecuencia de muestreo, tipo de generación del sonido, opciones referentes al espectrómetro (entrada de audio, tamaño de ventana, tipo de ventana, dB mínimo a mostrar, intervalo de muestreo), y opciones referentes a la calibración (si mostrar dBFS o no, el valor de la calibración lineal para el micrófono, el valor de la calibración lineal para los altavoces).

El valor de calibración lineal para el micrófono y para los altavoces es para el caso de que no se tenga la posibilidad de calibrar el dispositivo y así poder tener valores en positivo de dBSPL, dBA y dBC ya que los únicos valores de los que se disponen son dBFS que son negativos (0 dB es el valor máximo).

## II.V.- Datos Numéricos



Ilustración 111: Pantalla datos numéricos

Desde esta pantalla se pueden ver los diferentes ficheros guardados de las herramientas y experiencias para poder oírles o verles. Si se pulsa sobre ellos se visualizarán y si se deslizan hacia la izquierda se podrá dar la opción de borrarle.

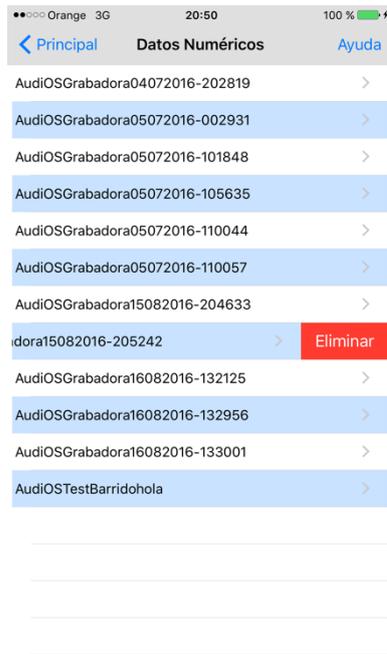


Ilustración 112: Pantalla eliminar datos numéricos



## BIBLIOGRAFÍA

---

- 📍 Apple. <http://www.apple.com/es/>
- 📍 Apple Developer. <https://developer.apple.com/>
- 📍 Mobile OS (Operating System) Percent Market Share Worldwide (As of June 2015) - areppim AG. [http://stats.areppim.com/stats/stats\\_mobiosxsnapshot.htm](http://stats.areppim.com/stats/stats_mobiosxsnapshot.htm)
- 📍 Mobile/Tablet Operating System Market Share - Net Applications.com. <https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8&qpcustomd=1>
- 📍 iOS - Wikipedia. <https://es.wikipedia.org/wiki/IOS>
- 📍 iOS: El sistema operativo móvil de Apple - Culturacion. <http://culturacion.com/ios-el-sistema-operativo-movil-de-apple/>
- 📍 Arquitectura iOS - Tecnología iOS. <https://sites.google.com/site/tecnologiaiostm/desarrollo-de-aplicaciones/arquitectura-ios>
- 📍 Sistemas operativos móviles: iOS - Eve Porras. <http://eve-ingsistemas-u.blogspot.com.es/2012/04/sistemas-operativos-moviles-ios.html>
- 📍 Taller de introducción a iOS development - rafaechve. <http://es.slideshare.net/rafaechve/taller-de-introduccion-a-i-os-development>
- 📍 Sistema operativo iOS características en iPhone y iPad - Toni Mateu. <http://iphonedigital.es/sistema-operativo-ios-caracteristicas-del-iphone/>
- 📍 iOS vs Android: así se comparan en 2015 - David Nield. <http://es.gizmodo.com/ios-versus-android-asi-se-comparan-en-2015-1700516382>
- 📍 La fragmentación, uno de los mayores problemas del ecosistema Android - Xavi Santamaria. <http://andro4all.com/2015/08/android-fragmentacion-2015>
- 📍 Seguridad de iOS - Apple. [https://www.apple.com/es/business/docs/iOS\\_Security\\_Guide\\_es\\_Oct14.pdf](https://www.apple.com/es/business/docs/iOS_Security_Guide_es_Oct14.pdf)
- 📍 Xcode - Wikipedia. <https://es.wikipedia.org/wiki/Xcode>
- 📍 Swift Vs Objective-C: Comparamos ambos Lenguajes de Programación – Luis Roca. <http://www.ipadizate.es/2014/06/30/swift-vs-objective-c-94499/>
- 📍 Swift y Objective-C, analizamos cuál es el mejor de ambos - Julio César Fernández. <https://applecoding.com/analisis/swift-objectivec-analizamos-cual-es-mejor>
- 📍 Por qué Swift es mejor que Objective-C - Cesar Anton. <https://platzi.com/blog/swift-vs-objective-c/>
- 📍 Swift vs. Objective-C - Xavi. <http://ameu8.com/sin-categoria/swift-vs-objective-c>
- 📍 TIOBE Index for July 2016 - TIOBE. <http://www.tiobe.com/tiobe-index/>
- 📍 Swift – Apple. <https://swift.org/>
- 📍 Curso de Apps (I): estructura básica y primera app - Julio César Fernández. <https://applecoding.com/cursos/curso-apps-estructura-basica>
- 📍 Curso de iOS7: Ciclo de vida de un View Controller - Cursos de Desarrollo. <http://cursosdedesarrollo.com/2015/12/curso-de-ios7-ciclo-de-vida-de-un-view-controller/>
- 📍 Work with View Controllers - Apple. <https://developer.apple.com/library/prerelease/content/referencelibrary/GettingStarted/DevelopiOSAppsSwift/Lesson4.html>

- 🍏 UIViewController - Apple. [https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIViewController\\_Class/index.html](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIViewController_Class/index.html)
- 🍏 THE COMMON LIFECYCLE OF A VIEW CONTROLLER – Matteo Manferdini. <http://matteomanferdini.com/the-common-lifecycle-of-a-view-controller/>
- 🍏 Con el nuevo código penal... ¿Es legal el Jailbreak? - Carlos Burges Ruiz de Gopegui. <http://www.faq-mac.com/2015/07/con-el-nuevo-codigo-penal-es-legal-el-jailbreak/>
- 🍏 Onda - Wikipedia. <https://es.wikipedia.org/wiki/Onda>
- 🍏 Sonido - Wikipedia. <https://es.wikipedia.org/wiki/Sonido>
- 🍏 Ondas - Explicación y definición de una onda - quees.info. <http://www.quees.info/que-es-una-onda.html>
- 🍏 LAS ONDAS Y SUS CARACTERÍSTICAS - José Villasuso Gato . [http://teleformacion.edu.aytolacoruna.es/FISICA/document/fisicaInteractiva/Ondasbachillerato/ondasCaract/ondas-Caract\\_indice.htm#tipos](http://teleformacion.edu.aytolacoruna.es/FISICA/document/fisicaInteractiva/Ondasbachillerato/ondasCaract/ondas-Caract_indice.htm#tipos)
- 🍏 Onda Sonora - Wikipedia. [https://es.wikipedia.org/wiki/Onda\\_sonora](https://es.wikipedia.org/wiki/Onda_sonora)
- 🍏 Onda senoidal - EcuRed, [http://www.ecured.cu/Onda\\_senoidal](http://www.ecured.cu/Onda_senoidal)
- 🍏 Tipos de ondas sonoras - Equipo de Redacción MasTiposde.com. [http://www.mastiposde.com/ondas\\_sonoras.html](http://www.mastiposde.com/ondas_sonoras.html)
- 🍏 Root mean square - Wikipedia. [https://en.wikipedia.org/wiki/Root\\_mean\\_square](https://en.wikipedia.org/wiki/Root_mean_square)
- 🍏 Colors of noise - Wikipedia. [https://en.wikipedia.org/wiki/Colors\\_of\\_noise](https://en.wikipedia.org/wiki/Colors_of_noise)
- 🍏 PROPIEDADES DE LAS ONDAS - Ondas estacionarias - Manuel Alonso Sánchez, Ana Lozano del Águila, Alberto Vallecillo. <http://intercentres.edu.gva.es/iesleonardodavinci/Fisica/Ondas/Ondas12.htm>
- 🍏 ONDAS ESTACIONARIAS - Antonio Juan Barbero García. <https://www.uclm.es/profesorado/ajbarbero/CursoAB2007/OndasEstacionarias06.pdf>
- 🍏 Decibelio - Wikipedia. <https://es.wikipedia.org/wiki/Decibelio>
- 🍏 Curvas Isofónicas - Wikipedia. [https://es.wikipedia.org/wiki/Curva\\_isof%C3%B3nica](https://es.wikipedia.org/wiki/Curva_isof%C3%B3nica)
- 🍏 Decibelios Ponderados - Marco Juan de Dios. <http://www.orbitasonica.com/2011/10/decibelios-ponderados.html>
- 🍏 Transformada de Fourier - Wikipedia. [https://es.wikipedia.org/wiki/Transformada\\_de\\_Fourier](https://es.wikipedia.org/wiki/Transformada_de_Fourier)
- 🍏 Transformada de Fourier Discreta - Wikipedia. [https://es.wikipedia.org/wiki/Transformada\\_de\\_Fourier\\_discreta](https://es.wikipedia.org/wiki/Transformada_de_Fourier_discreta)
- 🍏 Ventana (función) - Wikipedia. [https://es.wikipedia.org/wiki/Ventana\\_\(funci%C3%B3n\)](https://es.wikipedia.org/wiki/Ventana_(funci%C3%B3n))
- 🍏 Window Function - Wikipedia. [https://en.wikipedia.org/wiki/Window\\_function#Gaussian\\_window](https://en.wikipedia.org/wiki/Window_function#Gaussian_window)
- 🍏 Trabajo Ventanas - geocities. [http://www.geocities.ws/fg\\_audio/Archivos/Trabajo\\_Ventanas.pdf](http://www.geocities.ws/fg_audio/Archivos/Trabajo_Ventanas.pdf)
- 🍏 La FFT y las ventanas - Universidad del País Vasco. <http://www.ehu.eus/Procesadodesenales/tema8/tz2y.html>
- 🍏 Analysis Window (Part 1) - educasound. <http://educasound.com/analysis-window-part-i/>
- 🍏 Ventanas. [http://profesores.fi-b.unam.mx/maixx/Biblioteca/Librero\\_Telecom/Libro\\_ProcDigitSeniales\\_Ibarra/DSP\\_Cap\\_18\\_ventanas.pdf](http://profesores.fi-b.unam.mx/maixx/Biblioteca/Librero_Telecom/Libro_ProcDigitSeniales_Ibarra/DSP_Cap_18_ventanas.pdf)
- 🍏 Spectrum and spectral density estimation by the Discrete Fourier transform (DFT), including a comprehensive list of window functions and some new flat-top windows - G. Heinzel, A. Rüdiger and R. Schilling. [https://holometer.fnal.gov/GH\\_FFT.pdf](https://holometer.fnal.gov/GH_FFT.pdf)
- 🍏 Calibración - Wikipedia. <https://es.wikipedia.org/wiki/Calibraci%C3%B3n>
- 🍏 Modulación por impulsos codificados - Wikipedia. [https://es.wikipedia.org/wiki/Modulaci%C3%B3n\\_por\\_impulsos\\_codificados](https://es.wikipedia.org/wiki/Modulaci%C3%B3n_por_impulsos_codificados)
- 🍏 Waveform Audio Format - Wikipedia. [https://es.wikipedia.org/wiki/Waveform\\_Audio\\_Format](https://es.wikipedia.org/wiki/Waveform_Audio_Format)
- 🍏 CSV - Wikipedia. <https://es.wikipedia.org/wiki/CSV>

- 🍏 ToneGenerator - OOPer in cooperation with shlab.jp. <https://github.com/ooper-shlab/ToneGenerator1.0-Swift>
- 🍏 Charts - Daniel Cohen Gindi. <https://github.com/danielgindi/Charts/tree/master/Charts>
- 🍏 Apuntes de la carrera
- 🍏 stackoverflow

