



Universidad de Valladolid

Facultad de Ciencias

TRABAJO FIN DE GRADO

Grado en Física

Determinación de Dynamic-Route-Maps en
Memristores RRAM a partir de medidas
experimentales de parámetros de admitancia

Autor: *Carlos Díez García*

Tutor/es: *Salvador Dueñas Carazo*

Agradecimientos

Agradecer a mi pareja todo el ánimo y apoyo que me ha brindado para abordar este trabajo de fin de grado. Del mismo modo, mencionar a mi tutor Salvador Dueñas Carazo por los buenos comentarios y correcciones durante toda la elaboración del proyecto.

Índice de contenidos

MARCO TEÓRICO	7
1.1 INTRODUCCIÓN.....	7
1.1.1 <i>Situación actual de los sistemas de aprendizaje automático.</i>	7
1.2 CONMUTACIÓN RESISTIVA.....	7
1.2.1 <i>Tipos de conmutación resistiva</i>	8
1.2.2 <i>Mecanismos de formación del filamento de corriente en óxidos</i>	9
1.3 MEMORIAS RRAM	9
1.4 PROPIEDADES DE LA RRAM.....	10
1.5 MULTIPLICIDAD DE ESTADOS INTERMEDIOS EN UNA RRAM BASADA EN <i>HfO₂</i>	11
DYNAMIC-ROUTE-MAPS	12
2.1 INTRODUCCIÓN	12
2.2 FASE I. DEFINICIÓN DE REQUISITOS FUNCIONALES DE DRM	12
2.3 FASE II. DISEÑO DE DRM	13
2.3.1 <i>Diseño de la pantalla de inicio.</i>	13
2.3.2 <i>Diseño de pantallas de configuración</i>	14
2.3.3 <i>Diseño de pantallas de simulación</i>	14
2.3.4 <i>Diseño de pantallas de 4 gráficos</i>	16
2.4 FASE III. DESARROLLO DE DRM	16
2.4.1 <i>Arquitectura empleada</i>	17
2.4.2 <i>Desarrollo de la capa visual (interfaz gráfica) de DRM.</i>	18
2.4.3 <i>Planteamiento del núcleo de DRM: Simulación</i>	20
2.4.4 <i>Planteamiento de la función triangular.</i>	25
2.4.5 <i>Desarrollo de la capa lógica de DRM</i>	27
2.5 FASE IV. PRUEBAS DE DRM	30
2.5.1 <i>Pruebas generales funcionales</i>	30
2.5.2 <i>Pruebas generales de calibración</i>	32
2.6 FASE V. EXPORTACIÓN DEL EJECUTABLE DRM	34
2.7 LIMITACIONES.....	34
2.8 MEJORAS O FUNCIONALIDADES FUTURAS	35
SIMULACIÓN DE CURVAS I(V) CON DRM	36
3.1 INTRODUCCIÓN	36
3.2 CONFIGURACIÓN INICIAL.....	36
3.3 BARRIDO SINUSOIDAL EXPONENCIAL	36
3.4 BARRIDO TRIANGULAR.....	39
3.5 FUNCIÓN <i>V(t)</i> AMORTIGUADA.....	41
3.6 FUNCIÓN <i>Vt</i> ARBITRARIA.....	41
ANEXO I. MANUAL DE USUARIO DE DRM	45

Índice de ilustraciones

ILUSTRACIÓN 1 - TIPOS DE CONMUTACIÓN RESISTIVA.....	8
ILUSTRACIÓN 2 - ESTRUCTURA DE RRAM	10
ILUSTRACIÓN 3 – ESTADOS INTERMEDIOS DEL $TiN/Ti/a - HfO_2/TiN$	11
ILUSTRACIÓN 4 - DISEÑO PRIMARIO DE PANTALLA DE INICIO	14
ILUSTRACIÓN 5 - DISEÑO PRIMARIO DE PANTALLAS DE CONFIGURACIÓN.....	14
ILUSTRACIÓN 6 - DISEÑO PRIMARIO PANTALLA SIMULACIÓN BARRIDO PRESTABLECIDO	15
ILUSTRACIÓN 7 - DISEÑO PRIMARIO PANTALLA BARRIDO MANUAL	15
ILUSTRACIÓN 8 - DISEÑO PRIMARIO PANTALLAS DE 4 GRÁFICOS	16
ILUSTRACIÓN 9 - ESQUEMA DE ARQUITECTURA DRM	18
ILUSTRACIÓN 10 - MANEJO DE EVENTOS EN JAVA.....	19
ILUSTRACIÓN 11 - DESARROLLO DE LA INTERFAZ DE DRM	20
ILUSTRACIÓN 12 - REPRESENTACIÓN DEL COMPORTAMIENTO MULTINIVEL DEL MATERIAL	21
ILUSTRACIÓN 13 - OBTENCIÓN DE TENSIONES PARA CONDICIÓN DE RESET COMPLETO	21
ILUSTRACIÓN 14 - R(V)	22
ILUSTRACIÓN 15 - G(V).....	22
ILUSTRACIÓN 16 - DETERMINACIÓN DE CICLOS SUPERIOR E INFERIOR	23
ILUSTRACIÓN 17 - INTERPOLACIÓN DE LAS CURVAS DE CONDUCTANCIA	24
ILUSTRACIÓN 18 – OBTENCIÓN DE PUNTOS VIRTUALES DE CURVAS DE CONDUCTANCIA.....	24
ILUSTRACIÓN 19 - OBTENCIÓN DE EXPRESIÓN PARA UNA FUNCIÓN TRIANGULAR CRECIENTE	26
ILUSTRACIÓN 20 - CICLO DE VIDA DE EXTRACCIONES DATOS EXPERIMENTALES.JAVA	28
ILUSTRACIÓN 21 - CICLO DE VIDA GENERADORA TENSIONES.JAVA	28
ILUSTRACIÓN 22 - DIAGRAMA DE FLUJO DEL CORE DE DRM	29
ILUSTRACIÓN 23 - COMPORTAMIENTO DE CAPA LÓGICA A ALTO NIVEL	30
ILUSTRACIÓN 24 - PANTALLA DE INICIO.....	31
ILUSTRACIÓN 25 - PANTALLA DE CONFIGURACIÓN.....	31
ILUSTRACIÓN 26 - PANTALLA DE DATOS BASE	32
ILUSTRACIÓN 27 - PANTALLA DE SIMULACIÓN DE BARRIDO PREDEFINIDO	32
ILUSTRACIÓN 28 - VISUALIZACIÓN DE DATOS DE ENTRADA	33
ILUSTRACIÓN 29 - VALIDACIÓN DEL ALGORITMO.....	33
ILUSTRACIÓN 30- COMPARATIVA DATOS REALES CON DATOS GENERADOS POR DRM	34
ILUSTRACIÓN 31 - SELECCIÓN DE FICHEROS DE ENTRADA	36
ILUSTRACIÓN 32 - ANÁLISIS V(T) E I(V) DE BARRIDO SINUSOIDAL EXPONENCIAL	37
ILUSTRACIÓN 33 - ANÁLISIS CONJUNTO PARA V(T) SINUSOIDAL EXPONENCIAL	37
ILUSTRACIÓN 34 - MAPA COMPLETO MULTINIVEL DEL OXIDO DE HAFNIO	38
ILUSTRACIÓN 35 - TENDENCIA DE LOS PUNTOS DE RESET	39
ILUSTRACIÓN 36 - BARRIDO TRIANGULAR DONDE NO SE PRODUCE NINGÚN RESET	40
ILUSTRACIÓN 37 - GRÁFICO CONJUNTO PARA SIMULACIÓN TRIANGULAR.	40
ILUSTRACIÓN 38 - V(T) AMORTIGUADA	41
ILUSTRACIÓN 39 - FUNCIÓN V(T) ARBITRARIA	42

Resumen

La creciente demanda por parte de los sofisticados algoritmos basados en sistemas neuronales implementados por las inteligencias artificiales y el aprendizaje automático, han promovido numerosas investigaciones y caracterizaciones de materiales en busca de propiedades que suplan con las exigencias tecnológicas que dichos sistemas demandan. En este contexto adquieren el foco de atención las memorias emergentes RRAM, ya que además de presentar mejor escalabilidad, durabilidad y eficiencia energética, hay presencia de estados intermedios.

Con el propósito de servir como herramienta de apoyo a la caracterización de materiales con esta propiedad, y obtener estados intermedios no obtenidos en mediciones experimentales, se ha desarrollado el software DRM. Este software permite simular las curvas características del material ante variaciones arbitrarias del campo eléctrico al que se someta al dispositivo.

Abstract

The growing demand for sophisticated algorithms based on neural systems implemented by artificial intelligences and machine learning, have promoted numerous investigations and characterizations of materials in search of properties that meet the technological demands that these systems demand. In this context, the RRAM emergent memories acquire the focus of attention, since in addition to presenting better scalability, durability and energy efficiency, there is the presence of intermediate states.

In order to serve as a support tool for the characterization of materials with this property, and to obtain intermediate states not obtained in experimental measurements, the DRM software has been developed. This software allows simulating the characteristic curves of the material in the event of arbitrary variations in the electric field to which the device is subjected.

Introducción

Se ha estructurado el trabajo en 3 capítulos:

- En el capítulo 1 se da una breve introducción al fenómeno de la conmutación resistiva y se repasan las propiedades de las RRAM que motivan el desarrollo de DRM.
- En el capítulo 2 se detalla de forma exhaustiva el proceso de elaboración del software, así como el planteamiento de un método para simular las curvas características del material a analizar, en base a las consideraciones físicas que se deben imponer al modelo. También se especifican las limitaciones y fallos del modelo, así como futuras mejoras.
- Por último, en el capítulo 3 se expone el funcionamiento de DRM mediante el análisis de 4 funciones $V(t)$ de entrada.

Capítulo 1.

Marco teórico

1.1 Introducción

1.1.1 Situación actual de los sistemas de aprendizaje automático.

En la actualidad, las ramas que se valen del aprendizaje automático en los sistemas tecnológicos abarcan desde el reconocimiento de imágenes, super resolución y super fluidez aplicados a vídeos antiguos, hasta la elaboración de sistemas robóticos capaces de analizar y desenvolverse completamente en su entorno. .

La rápida y continua evolución de algoritmos matemáticos implementados en estos sistemas demanda cada vez componentes electrónicos más sofisticados, que permitan responder eficientemente en términos computacionales y energéticos (1). Estos sistemas de aprendizaje trabajan con enormes cantidades de datos, por lo que son necesarias memorias que sean capaces de responder en tiempo prudencial a la gran cantidad de cálculos que se deben efectuar. De esta demanda surge la motivación de investigar nuevas propiedades que emergen de la combinación de materiales conformados en estructuras Metal-Aislante-Metal (MIM).

En este contexto, aparecen las RRAM¹ como memorias no volátiles² que se fundamentan en el fenómeno físico de la conmutación resistiva, y que presentan excelentes propiedades que cumplen los requisitos exigidos anteriormente (2).

1.2 Conmutación resistiva

Las RRAM están gobernadas por el fenómeno de la conmutación resistiva, que se entiende como a capacidad que presenta un determinado material aislante para alterar su estado de resistencia desde un estado de alta resistencia (HRS³) hasta un estado de baja resistencia (LRS⁴) o viceversa, lo que se conoce como

¹ Memoria de acceso aleatorio resistivo

² Se entiende por memoria no volátil a aquella que preserva la información cuando deja de recibir el estímulo de corriente.

³ High Resistance State

⁴ Low Resistance State

SET o RESET respectivamente (3). Esta capacidad puede aprovecharse en computación asociando cada estado del sistema con los estados binarios 0 y 1, lo que se consigue estudiando las tensiones en las que se produce un cambio de estado, V_{reset}, V_{set} . Este fenómeno se observa comúnmente en estructuras metal-aislante-metal, en las que la transición entre estados se produce mediante la formación en el material aislante de un filamento conductor al alcanzarse la tensión de set.

1.2.1 Tipos de conmutación resistiva

Existen 2 tipos de conmutación resistiva (4):

Conmutación Unipolar:

Se caracteriza porque la transición entre los estados *HRS* y *LRS* puede realizarse indistintamente tanto si la tensión BIAS⁵ aplicada es directa (>0) o inversa (<0), por lo que las características I-V serán simétricas respecto de dicha tensión BIAS (Ilustración 1 a). Es por este motivo que se denomina unipolar, ya que realmente solo requiere de una polaridad para poder observar la transición de estado.

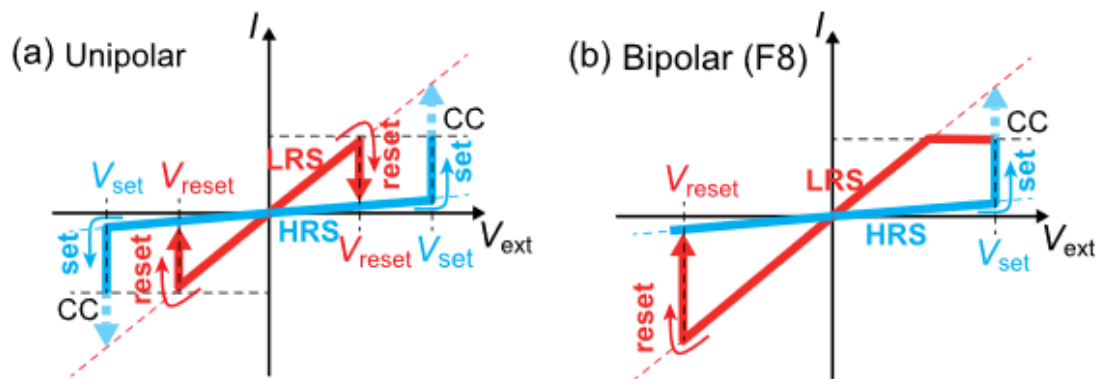


Ilustración 1 - Tipos de conmutación resistiva

Conmutación Bipolar:

En este caso, son requeridas ambas polaridades para completar un ciclo completo de transiciones de estado. Así, si la tensión suministrada es mayor que 0, se produce el Set al estado *LRS*. Una vez en este estado, no se produce de nuevo una transición a *HRS* hasta que no se alcance cierta tensión en polaridad inversa (Ilustración 1 b), o Reset del material.

⁵ Se refiere a la tensión de la fuente.

1.2.2 Mecanismos de formación del filamento de corriente en óxidos

Tal como se ha mencionado anteriormente, el mecanismo de conmutación resistiva viene asociado a la formación de un filamento conductor en el material aislante. En función del material aislante que se coloque en la capa funcional de la estructura MIM, la formación del filamento se producirá de una u otra forma. En el caso de que dicho material sea un óxido, el mecanismo que interviene es la migración de vacantes de oxígeno (4). Esto puede entenderse de varias formas:

- a. Al aplicar un campo eléctrico externo suficientemente elevado, se promueve el movimiento de electrones dando lugar a la formación de vacantes de oxígeno en el material aislante, que tienden a concentrarse en una región del espacio. Cuando se generan varios de estos agregados de vacantes, localmente se produce una caída de la resistencia, y se dice que se ha formado un filamento conductor (CF⁶). En estos términos, se produce una transición de estado *HRS* a *LRS* cuando un determinado número de estos CF han sido formados.
- b. La generación en el material de vacantes de oxígeno puede generar una barrera de Schottky modulable aplicando una tensión externa, modificando el estado del material.
- c. Las vacantes de oxígeno generadas pueden confinar los electrones libres que quedan atrapados en la barrera de Schottky, pero esto puede contrarrestarse con un campo eléctrico que los libere, promoviendo así una nueva transición.

1.3 Memorias RRAM

Las memorias de acceso aleatorio resistivo son un tipo de memoria no volátil están formadas por 2 delgadas capas externas (electrodos), y una capa interna aislante o semiconductor, conformando una estructura tipo sándwich (5), como la que figura en la Ilustración 2.

⁶ Filamento conductor.

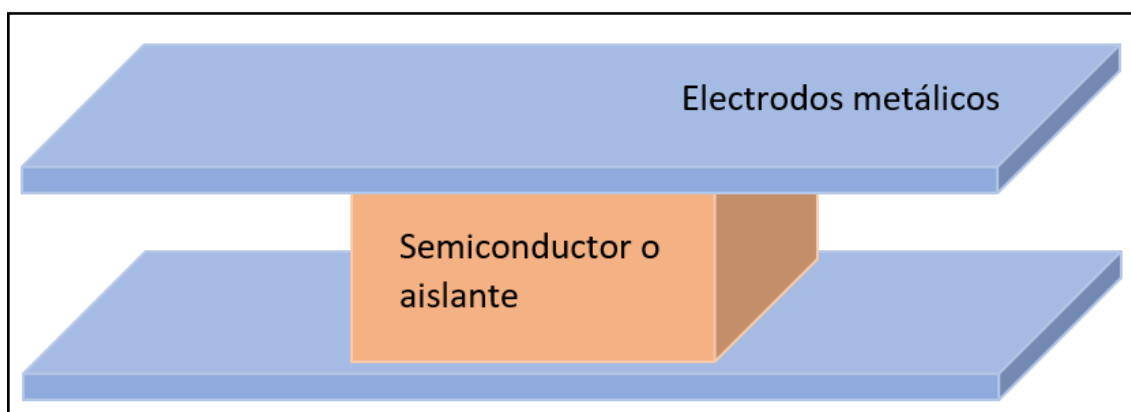


Ilustración 2 - Estructura de RRAM

A diferencia de otro tipo de memorias, están basadas en el fenómeno físico de la conmutación resistiva expuesto en el apartado anterior.

Existen varios materiales que pueden emplearse como base de fabricación de una RRAM, no obstante, los que están basados en óxidos inorgánicos y combustibles fósiles son los que conceden a las RRAM las propiedades óptimas para operar como elementos sinápticos de sistemas neuromórficos (3).

1.4 Propiedades de la RRAM.

Los avances en investigación han demostrado que las RRAM podrían ser la solución a los problemas de escalabilidad, durabilidad, suficiencia energética y necesidad de operar con grandes cantidades de datos. Las propiedades que se infieren de las mediciones de los parámetros de admitancia de estos sistemas (2) son las siguientes:

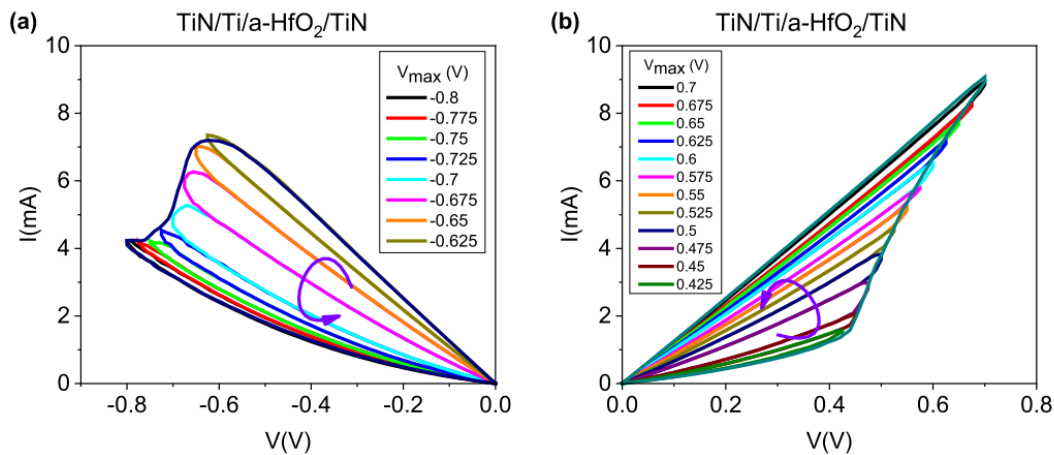
- ✓ Bajo consumo de potencia: en ambas polaridades las tensiones requeridas para conseguir ciclos de set y reset completos son muy bajas, lo que reduce significativamente el consumo de potencia de estos sistemas.
- ✓ Multiplicidad de estados intermedios: se observa que para cada tensión máxima que se aplique en el material, se obtiene un ciclo diferente, que solo depende de dicha tensión máxima.
- ✓ Alta durabilidad: se comprueba que el material continúa siendo estable tras aplicar numerosos ciclos de lectura/escritura consecutivos.
- ✓ Miniaturización: estas estructuras están compuestas por delgadas capas conductoras y una fina capa de material dieléctrico, lo que supone una solución al exigente problema de escalabilidad de los componentes electrónicos.

1.5 Multiplicidad de estados intermedios en una RRAM basada en HfO_2

El gran interés mostrado en la detección de materiales que presenten este fenómeno multinivel entre estados primarios de set y reset puros (0 y 1 lógicos), radica principalmente en que tanto el aprendizaje automático, como los algoritmos que implementan las inteligencias artificiales depende de un elevadísimo número de cálculos para lograr estabilizar los parámetros óptimos del modelo que se esté realizando.

Estos algoritmos están compuestos por multitud de neuronas que realizan diversas tareas, transformando unas determinadas entradas de datos en una salida de diferentes datos, que son a su vez inputs de otras neuronas. A medida que se entrena al algoritmo con diferentes entradas, este es capaz de ir regulando el peso que tiene cada neurona en el resultado final. Este proceso es, en general, costoso, y requiere de una gran cantidad de ciclos obtener los parámetros adecuados que logren que el método funcione.

Si se piensa ahora en los estados intermedios detectados en las RRAM, por ejemplo, aquellas en las que el material aislante es un óxido de hafnio, y se asocia cada uno de estos estados con los pesos que gobiernan cada neurona, se consigue reducir significativamente el tiempo de computo del algoritmo además de mejorar la precisión del mismo(6), ya que no se dispone únicamente de 2 niveles de tensión (ceros y unos), sino de múltiples valores intermedios.



Capítulo 2.

Dynamic-Route-Maps

2.1 Introducción

Los recientes avances científicos acerca del fenómeno de conmutación resistiva, así como los estudios de caracterización de memorias RRAM que presentan estados intermedios entre set y reset completos (2), como el HfO_2 , hacen necesario el desarrollo de un software cuya finalidad sea simular el comportamiento del material ante variaciones arbitrarias de la tensión a la que se le someta.

De esta motivación surge el proyecto DRM, creado como una herramienta de apoyo al análisis y caracterización de materiales que presenten estos estados intermedios, mediante la visualización de los datos experimentales así como la predicción de nuevos estados no obtenidos experimentalmente.

En las siguientes secciones se detalla el proceso de creación de DRM, que se ha dividido en 5 fases diferenciadas: fase de definición de requisitos, fase de diseño, fase de desarrollo, fase de pruebas, y fase de exportación del software.

2.2. Fase I. Definición de requisitos funcionales de DRM

Al igual que en todo nuevo desarrollo, antes de comenzar a preparar DRM, ha sido necesario realizar una toma inicial de requisitos en los que se recojan todas las especificaciones funcionales que debe cumplir, o, en otras palabras, cómo debe comportarse la aplicación de cara a un usuario final (7).

Esta serie de requisitos se han tomado pensando en las necesidades de DRM:

- I. Pantalla de inicio que cuente con acceso al resto de las pantallas a través de una botonera central.
- II. Debe permitir visualizar los datos de entrada.
- III. Pantalla específica con barrido predefinido sinusoidal exponencial.
- IV. Pantalla específica con barrido predefinido triangular.
- V. La simulación de curvas predefinidas se hará en tiempo real.
- VI. Es necesario que se permita importar una $V(t)$ arbitraria.
- VII. Todas las pantallas en las que se realice un barrido deben permitir generar cada gráfico resultante individualmente en la interfaz.
- VIII. Tiene que poder visualizarse la curva $V(t)$.

- IX. Todas las pantallas en las que se realice un barrido deben contar con un botón que permita visualizar los gráficos $I(V)$, $R(V)$, $G(V)$, $G(P)$ simultáneamente.
- X. Todos los gráficos deben poder exportarse en formato PNG.
- XI. Todas las pantallas de simulación deben permitir exportar los datos.
- XII. Pantalla de configuración en la que se puedan regular los parámetros de cada barrido predefinido.
- XIII. Para editar una configuración, se tiene que solicitar confirmación previa.
- XIV. Solo será posible modificar los valores de las propiedades, no sus descripciones.
- XV. Se debe permitir modificar los ficheros base en los que se encuentran los datos experimentales.
- XVI. El formato de los ficheros csv admitidos por DRM será separados por el carácter “;” y no estarán entrecomillados.
- XVII. Menú superior de acceso rápido a cada pantalla.
- XVIII. Botón de inicio en todas las pantallas para volver a la pantalla inicial.
- XIX. Botón salir en cada pantalla que permita abandonar la aplicación.
- XX. El botón salir debe solicitar confirmación al usuario.

Esta especificación inicial de requisitos ha facilitado enormemente el desarrollo de DRM, ya que se ha partido de un plan de actuación prefijado impidiendo así la desviación del objetivo principal.

2.3 Fase II. Diseño de DRM

Tras haber sido especificados los requisitos funcionales de la aplicación, se ha procedido con la elaboración del diseño de la interfaz gráfica (7) de DRM. Dicho diseño se estructura en 4 pantallas base, siendo el resto de las que se pudieran presentar similares a alguna de las presentadas aquí, pero con ligeras variaciones.

2.3.1 Diseño de la pantalla de inicio.

De acuerdo con los requisitos definidos anteriormente (Req. I), se ha preferido una disposición centrada con iconos de acceso a cada pantalla.

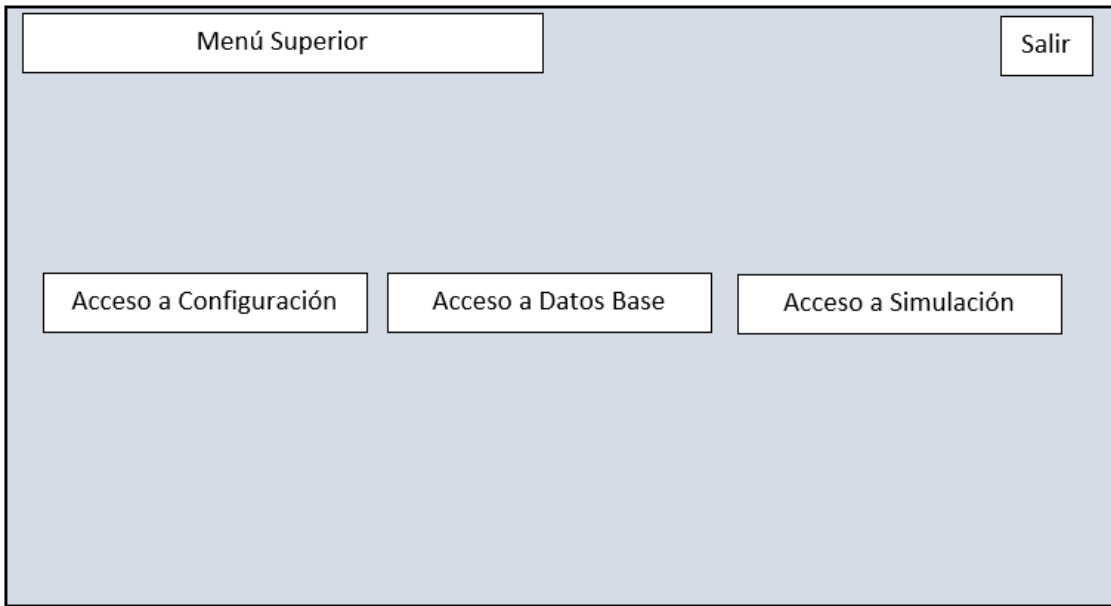


Ilustración 4 - Diseño primario de pantalla de inicio

2.3.2 Diseño de pantallas de configuración

Para esta pantalla se ha escogido una disposición de pantalla dividida horizontalmente en 2, conteniendo la parte superior una tabla editable, y en la inferior un botón para confirmar la edición de los parámetros que sean configurables (Req. XI – XV).

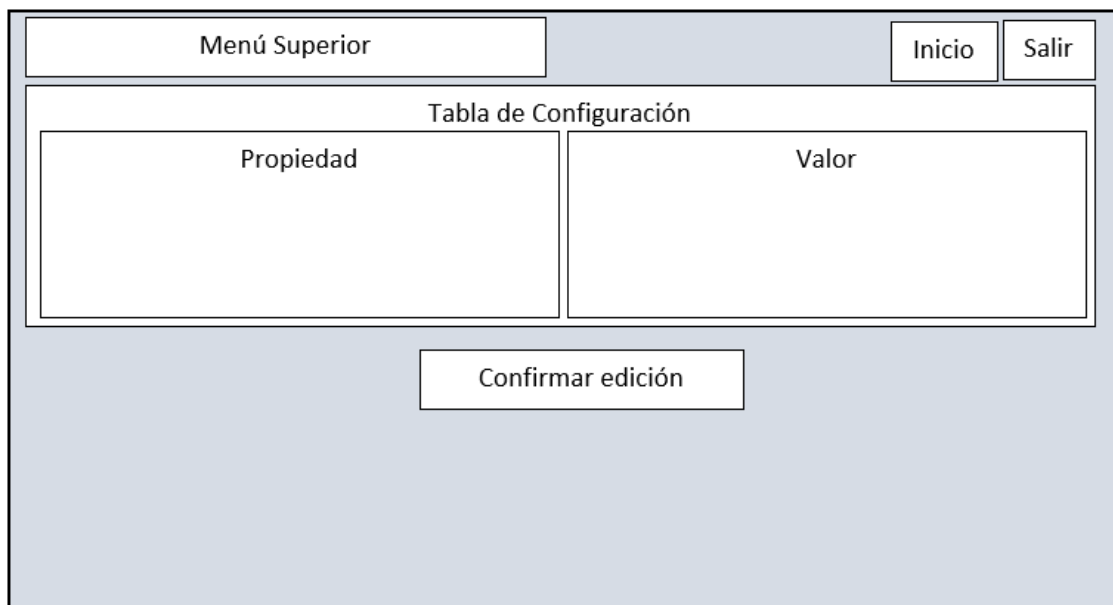


Ilustración 5 - Diseño primario de pantallas de configuración.

2.3.3 Diseño de pantallas de simulación

Para elaborar este diseño, siguiendo la tabla de requisitos (Req. III – X), se ha optado por una configuración de pantalla dividida en 2 de forma vertical, de

modo que a la izquierda se puedan elegir los datos del barrido, y a la derecha se muestren los gráficos generados. Así pues, se han distinguido 2 casos:

a. Barridos predefinidos:

La zona de la izquierda contendrá deslizadores y manipuladores que permitan al usuario modificar el valor de los parámetros del barrido. Debajo de estos, se ha colocado la botonera de generación de cada gráfico.

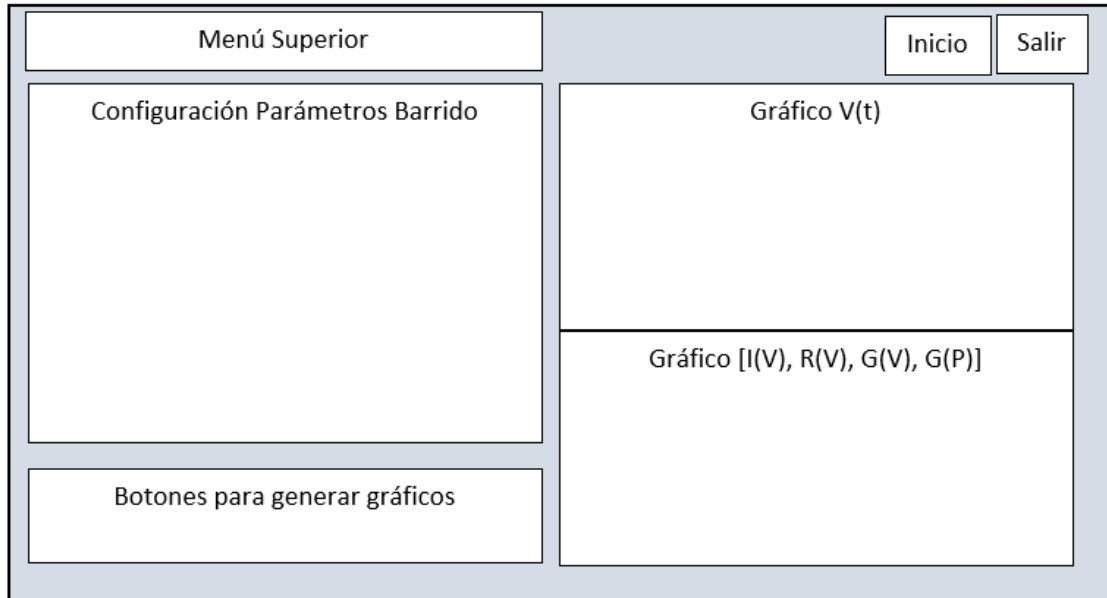


Ilustración 6 - Diseño primario pantalla simulación barrido preestablecido

b. Barrido manual (importación de datos):

Para este caso no aplica la configuración de parámetros, por lo que se ha elegido una disposición en cascada para cada uno de los botones de generación.

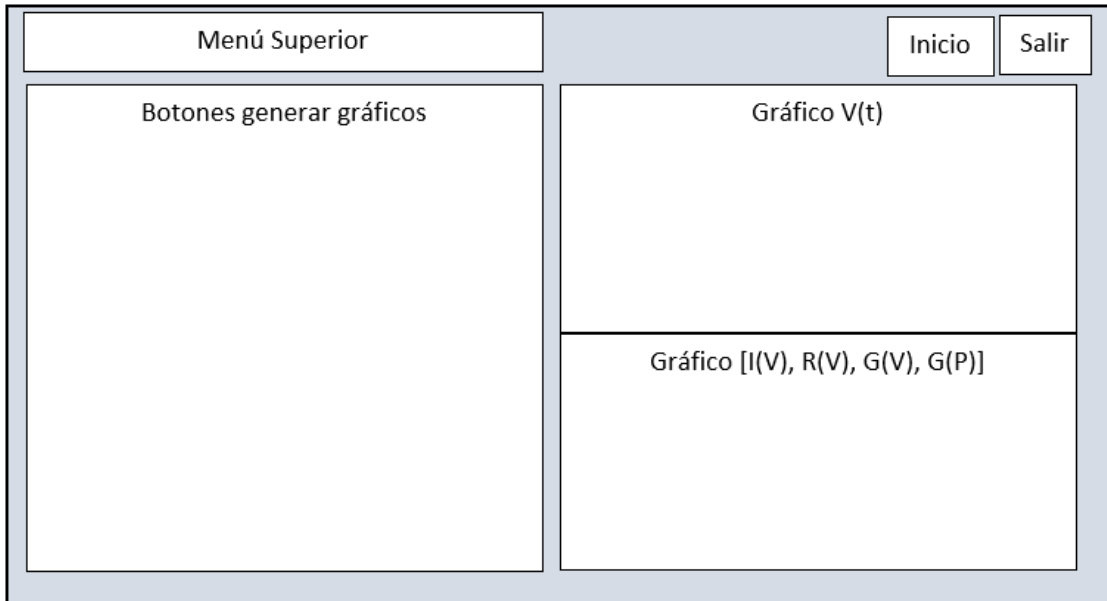


Ilustración 7 - Diseño primario pantalla barrido manual

2.3.4 Diseño de pantallas de 4 gráficos

Cumpliendo con los requisitos funcionales II y VIII, se ha diseñado una pantalla que alberga simultáneamente los 4 gráficos característicos del material.

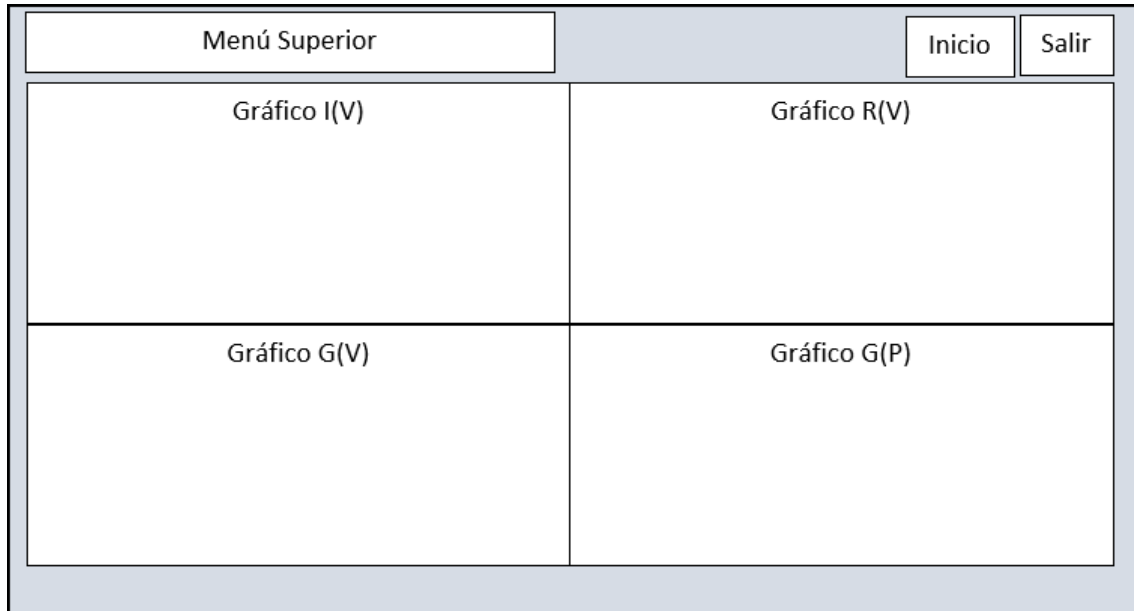


Ilustración 8 - Diseño primario pantallas de 4 gráficos

2.4 Fase III. Desarrollo de DRM

Sabidos ya los requerimientos funcionales del software, y especificado ya el diseño de la interfaz gráfica de usuario, se ha procedido a realizar un diseño técnico de cara a su implantación. Este diseño técnico (arquitectura), elaborado previo al comienzo del desarrollo, ha permitido detectar y solucionar los errores de forma rápida.

Respecto al lenguaje elegido para la codificación, se ha tenido preferencia por JAVA, ya que por sus características de ser un lenguaje orientado a objetos facilita mucho el planteamiento de una arquitectura limpia, mantenible y analizable en caso de error. En concreto, se ha requerido de las siguientes librerías para el desarrollo:

- ✓ JRE System Library [JavaSE-1.8] (librerías estándar de java).
- ✓ Librería jfreechart [versión 1.5.3] (para la generación de gráficos).

Los paquetes que han sido necesarios son:

- ✓ Paquete java.awt* para el manejo de disposiciones en pantalla y para el control de eventos.
- ✓ Paquete javax.swing.* para la generación de una interfaz de usuario.
- ✓ Paquete java.util.* que permite manejar colecciones y diccionarios de datos.
- ✓ Paquete org.jfree.* empleado en la generación de gráficos.

- ✓ Paquete java.io.* utilizado para la comunicación con los ficheros externos de datos.

Como entorno de desarrollo para el proyecto, se ha utilizado el IDE eclipse, ya que su interfaz facilita el mantenimiento de una correcta estructura del código en todo momento. Igualmente, su editor de código muestra diferentes niveles de advertencia y error en el momento de la codificación, así como predicción y sugerencia de métodos y variables.

2.4.1 Arquitectura empleada

Antes de exponer la arquitectura de software elegida para el diseño de DRM se puede definir como tal al conjunto de todos los elementos del código, así como las interrelaciones que mantienen entre ellos, con los datos, y con el usuario de la aplicación (8).

La arquitectura diseñada para este software ha sido la siguiente:

- a. Se dispone de un paquete “configuracion” que se encarga de regular el funcionamiento interno de la aplicación en base a los datos que se encuentren en los ficheros de configuración.
- b. Existe un paquete “metodos” que se encarga de realizar todas las operaciones lógicas con los datos. Se encarga de procesar los datos de los barridos así como de proporcionar los mismos como inputs del modelo y generar los datos de salida.
- c. Se ha definido un paquete “utilidades” cuya función es albergar procedimientos y algoritmos comunes a toda la aplicación, como son la lectura de ficheros, la generación de gráficos, etc.
- d. La paquetería “pantallas” se encarga de la configuración visual de la aplicación, es decir, almacena todo lo que el usuario ve, modifica...
- e. Por último, el paquete “listeners” se encarga de escuchar las acciones que el usuario ejecute en la aplicación, y transformarlas en eventos que se ejecuten. Es el encargado del comportamiento dinámico de la misma.

En un ejemplo, lo que se pretende con esta arquitectura es lo siguiente:

El usuario arranca la aplicación, y el paquete de configuración se carga con los valores que se toman de input. A través del paquete transmisor utilidades, se envía esta configuración a las pantallas para que al cargarlas, aparezcan con la información adecuada. También se envía dicha información a los métodos. Una vez cargadas, se puede, a través del paquete listeners:

- ✓ Cambiar a otra pantalla.
- ✓ Modificar la configuración.
- ✓ Realizar una simulación y dar lugar a un output.

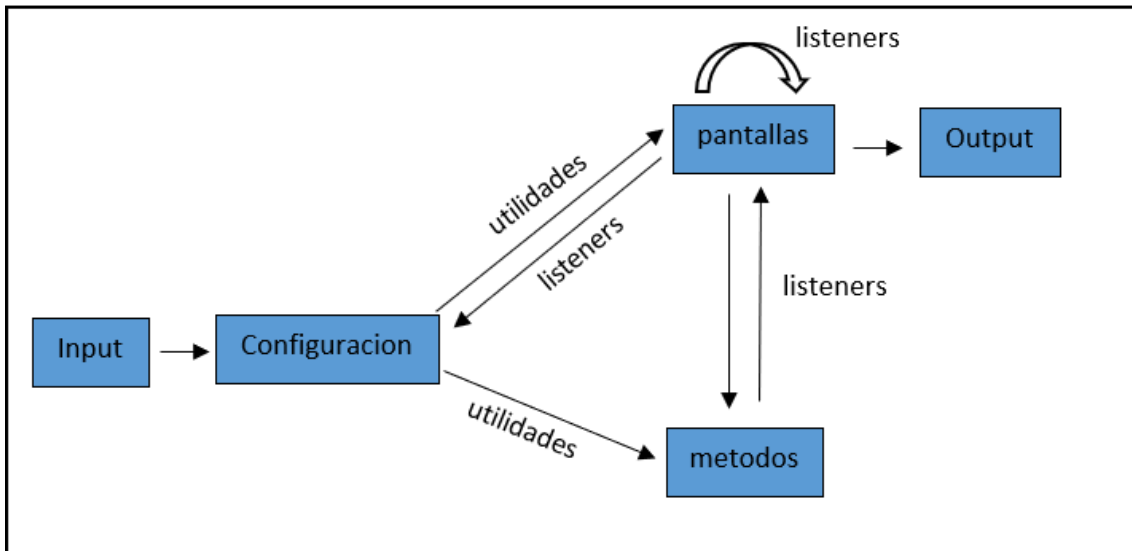


Ilustración 9 - Esquema de arquitectura DRM

2.4.2 Desarrollo de la capa visual (interfaz gráfica) de DRM.

Como se ha explicado en la sección anterior, el análisis previo de la arquitectura óptima para la consecución de los objetivos de DRM, ha facilitado enormemente la estructuración de cada uno de los componentes que se integran en la aplicación. En este punto se especifica de un modo global, el desarrollo de la interfaz de usuario.

Combinando la toma inicial de requisitos en la fase I, el diseño gráfico en la fase II, y el planteamiento de la arquitectura general en la fase III, ha sido necesaria la elaboración de 7 pantallas diferentes. Dichas pantallas están relacionadas a través de un menú superior, que permite la conexión y la navegación entre cada una de ellas por medio del control de eventos correspondiente.

Antes de continuar, es necesario explicar cómo JAVA es capaz de manejar e interpretar estos eventos (9). Digamos que existe un objeto⁷ "A" que debe realizar la acción " a_A " cuando este objeto es modificado de cualquier manera (bien sea un botón que se pulsa, un deslizador que se mueve, un área de la pantalla por la que se pasa el puntero del ratón...). Para ello, es necesario indicarle a "A" que debe estar pendiente en todo momento de una posible modificación en alguna de sus propiedades. Se dice entonces, que "A" es un escuchador de una acción sobre la propiedad P_A . Si " P_A " se ve modificado por dicha acción, se genera un evento que no es otra cosa que la acción " a_A " que debe realizar. De este modo, escuchador y evento son 2 entidades independientes, de forma que A podría a su vez escuchar otra propiedad P'_A para generar otro evento a'_A .

⁷ En java, todo es un objetos: Una variable, una clase, un evento...

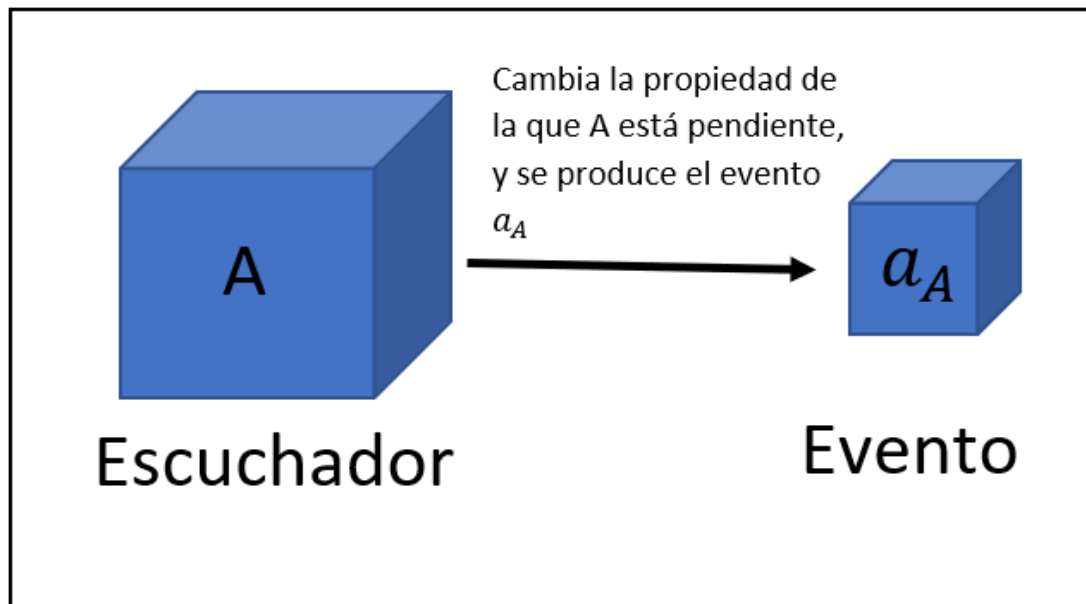


Ilustración 10 - Manejo de eventos en JAVA

Visto el concepto de escuchador-evento, se ha analizado el desarrollo de la interfaz de DRM (paquetes pantallas y listeners).

- ✓ PantallaInicio.java: conformada por los botones que redirigen a cada módulo particular de DRM, mediante la escucha a EscuchadoresAccion.java.
- ✓ MenuSuperior.java: alberga cada uno de los submenús que dan acceso a las diferentes pantallas del aplicativo. Los elementos de esta clase están preparados para realizar un evento de la clase EscuchadoresMenuSuperior.java y así poder navegar a través de la aplicación.
- ✓ PantallaConfiguración.java: se encarga de crear la pantalla desde la que se podrán modificar las variables de configuración de DRM. Dispone de un botón modificar que está a la escucha de la clase EscuchadoresAccion.java, para generar un evento que cambie dicha configuración.
- ✓ PantallaDatosEntrada.java: se encarga de crear la pantalla en la que se visualizan los gráficos en base a los datos de los ficheros de entrada para la simulación.
- ✓ PantallaSinExp.java y PantallaTriangular.java: formalmente son idénticas. Sus elementos están escuchando constantemente posibles modificaciones de los deslizadores e incrementadores, para generar un evento que realice cambios dinámicamente sobre los gráficos de la simulación (EscuchadoresCambio.java). Así mismo, la botonera inferior está preparada para ser pulsada y generar un evento de EscuchadoresAcción.java que genere los diferentes gráficos de forma estática.
- ✓ Pantalla Manual.java: solo está preparada para recibir eventos de EscuchadoresAcción.java, y generar los gráficos de la simulación de manera estática.
- ✓ PantallaTodosGraficos.java: se trata de una pantalla particular que se genera como evento al haber pulsado el botón “Grafico Conjunto”. Se implementa como una pantalla particular que no tiene asignado ningún

elemento escuchador, por lo que se sale de ella pulsando en el aspa de la ventana.

A modo de esquema ilustrativo, se incluye una representación del desarrollo de la capa visual, que incluye las diferentes pantallas y las interrelaciones entre ellas:

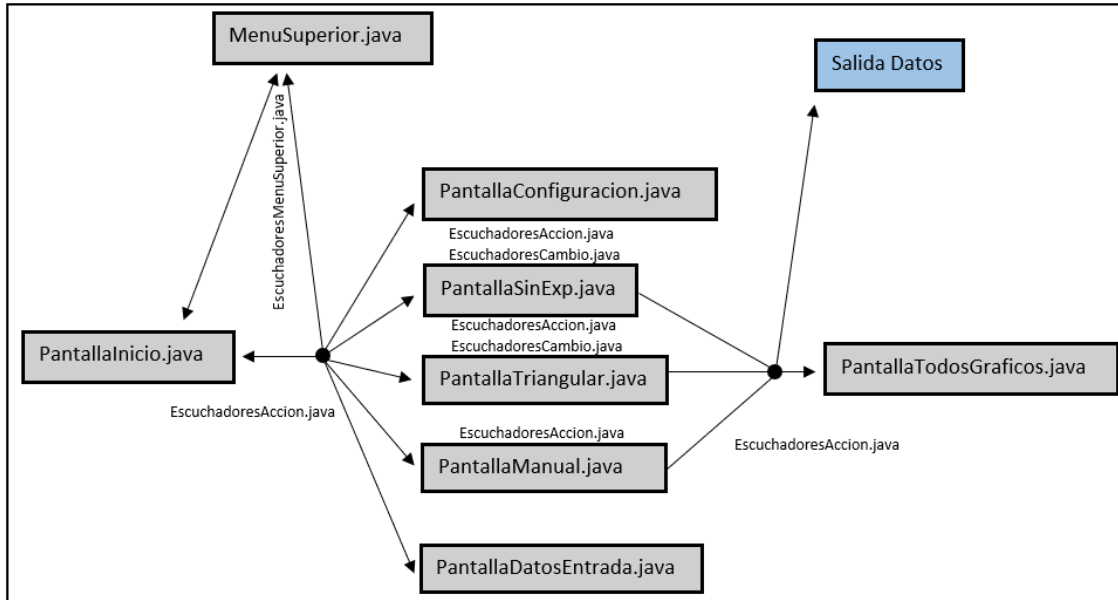


Ilustración 11 - Desarrollo de la interfaz de DRM

2.4.3 Planteamiento del núcleo de DRM: Simulación

Para ser capaces de implementar una lógica que pueda efectuar los cálculos oportunos, se han planteado física y matemáticamente las necesidades del núcleo, es decir, que datos tomará como entrada y cómo procesará los mismos para obtener unos datos de salida coherentes.

De lo expuesto en el capítulo 1, se revela un comportamiento multinivel del material, en el que el ciclo set-reset dependen únicamente de la tensión máxima que se haya aplicado sobre el dispositivo. De este modo, si se aplica una tensión máxima V_a en directa (set a estado LRS), al disminuir el voltaje seguirá una determinada curva c_a hasta que se produzca el reset completo (HRS). Pero si en lugar de aplicar V_a , se aplica V_b , el estado LRS será diferente al anterior, y por tanto el ciclo se consolidará con una curva c_b .

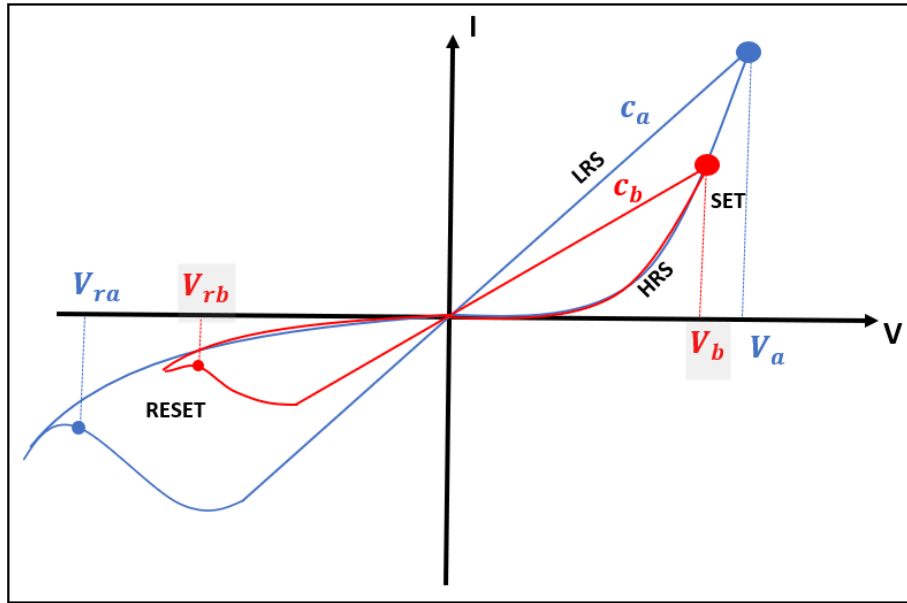


Ilustración 12 - Representación del comportamiento multinivel del material

Considerando que la corriente no puede cambiar espontáneamente del estado LRS a HRS si no es mediante un reset completo, es necesario imponer que si $\Delta V < 0$ pero no se ha alcanzado V_r , la corriente continúe por la curva definida LRS. Por el contrario, si se ha superado (en valor absoluto) V_r , al comenzar a incrementarse V , la corriente seguirá la curva correspondiente definida por HRS, completando así el ciclo set-reset.

De las observaciones experimentales se puede considerar en primera aproximación que $V_r \approx -V_M + \Delta V$, es decir, que dado el barrido en tensiones elegido, podemos establecer como condición de reset completo que se alcance, en tensión inversa, el valor del máximo de tensión del que se obtiene, cambiado de signo, y sumado a una pequeña cantidad que dependerá en cada caso del material analizado.

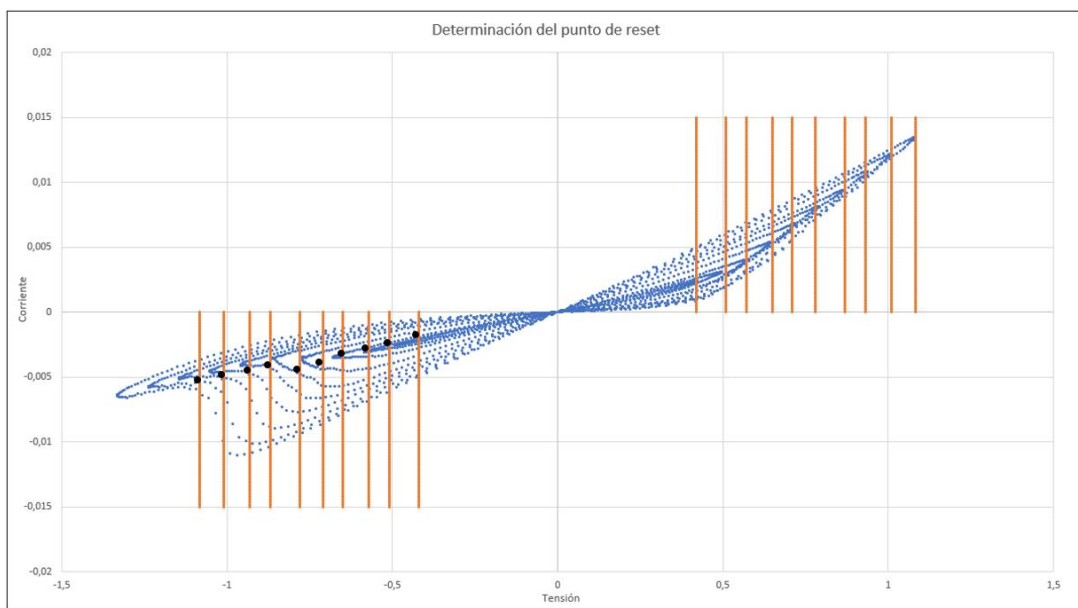


Ilustración 13 - Obtención de tensiones para condición de reset completo

Como segundo punto, se ha procedido a desarrollar un método que permite la obtención de los datos interpolados a partir de 2 curvas próximas. Para simplificar los cálculos, se ha realizado este proceso mediante interpolación lineal.

En primer lugar, se han analizado las curvas $R(V)$ y $G(V)$. En la Ilustración 14 se muestra la resistencia en función de la tensión; en la **¡Error! No se encuentra el origen de la referencia.** se muestra la conductancia en función de la tensión. Los puntos naranja y rojo muestran el camino que siguen las curvas cuando la tensión es decreciente. Los puntos azul y verde muestran el camino que siguen las curvas cuando la tensión es creciente. En vista de estos resultados, se ha determinado que el método a proseguir es el siguiente

- a. Si $V'(t) \geq 0$ y se alcanzó previamente el reset, entonces se realiza la interpolación de datos sobre las curvas $R(V)$.
- b. Si $V'(t) \geq 0$ pero no se alcanzó previamente el reset, entonces se realiza la interpolación de datos sobre las curvas $G(V)$.
- c. Si $V'(t) < 0$, la interpolación se realiza sobre las curvas $G(V)$.

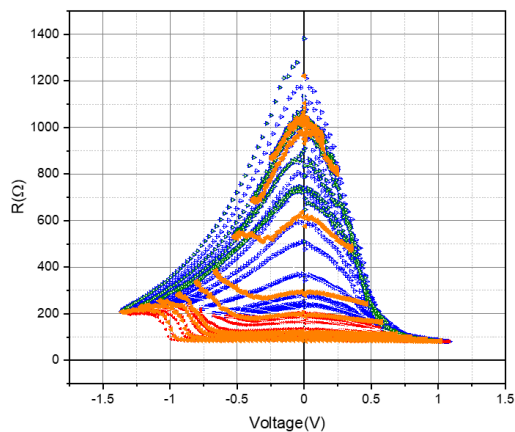


Ilustración 14 - R(V)

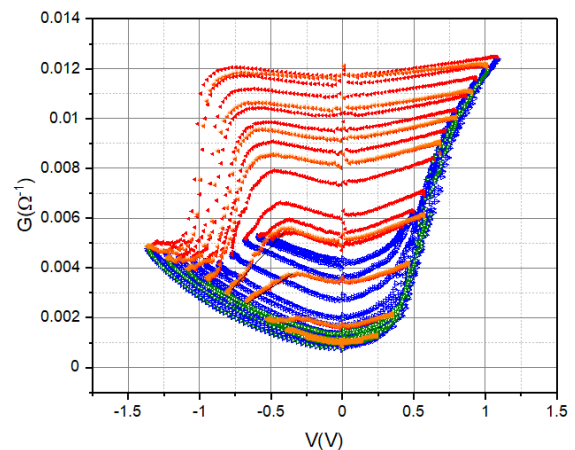


Ilustración 15 - G(V)

Para obtener las curvas interpoladas, se ha desarrollado el siguiente algoritmo:

- I. Se determina el máximo M_a de la curva $V(t)$ objeto de la simulación. Este máximo, es comparado con todos los máximos de la curva experimental, para obtener entre que 2 ciclos tendremos que realizar la interpolación. Llamaremos a los máximos de los ciclos de tensiones de set superior e inferior M_S y M_I , respectivamente.

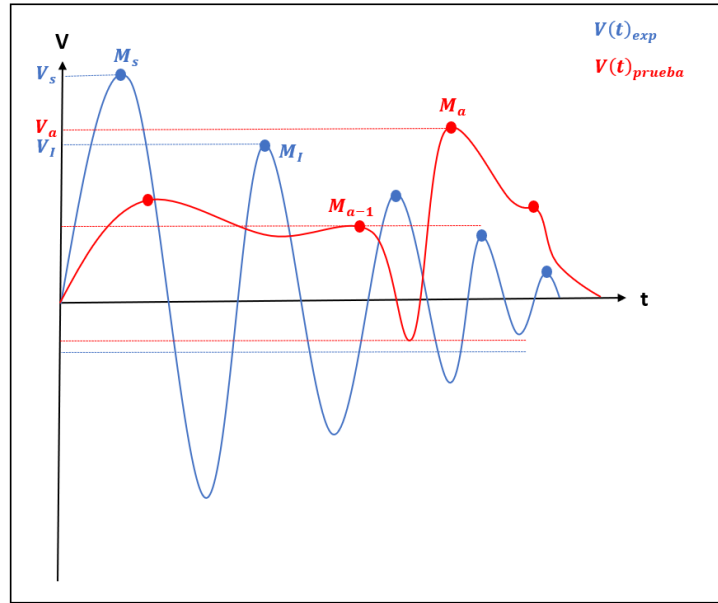


Ilustración 16 - Determinación de ciclos superior e inferior

- II. Encontrados estos valores, se puede calcular un coeficiente α_A que da cuenta de a que distancia se encuentra el valor V_A de las tensiones V_I, V_S

$$\alpha_A = \frac{V_A - V_I}{V_S - V_I}$$

Ecuación 1

- III. Una vez que sabemos la distancia de ese punto a los otros dos, y que conocemos los ciclos entre los que interpolar, es necesario extraer la conductancia (o resistencia, según su caso), de dichos ciclos. En el ejemplo de la Ilustración 16, observamos que desde la tensión máxima (positiva) del punto anterior, no se da un reset completo (ya que la tensión debería caer como mínimo hasta la línea inferior azul (la tensión máxima del siguiente nivel cambiada de signo). Entonces son necesarias para la interpolación las curvas $G(V)$. Cada punto de $G_A(V)$ puede expresarse como:

$$G_A(V) = G_I(V) + \alpha_A (G_S(V) - G_I(V))$$

Ecuación 2

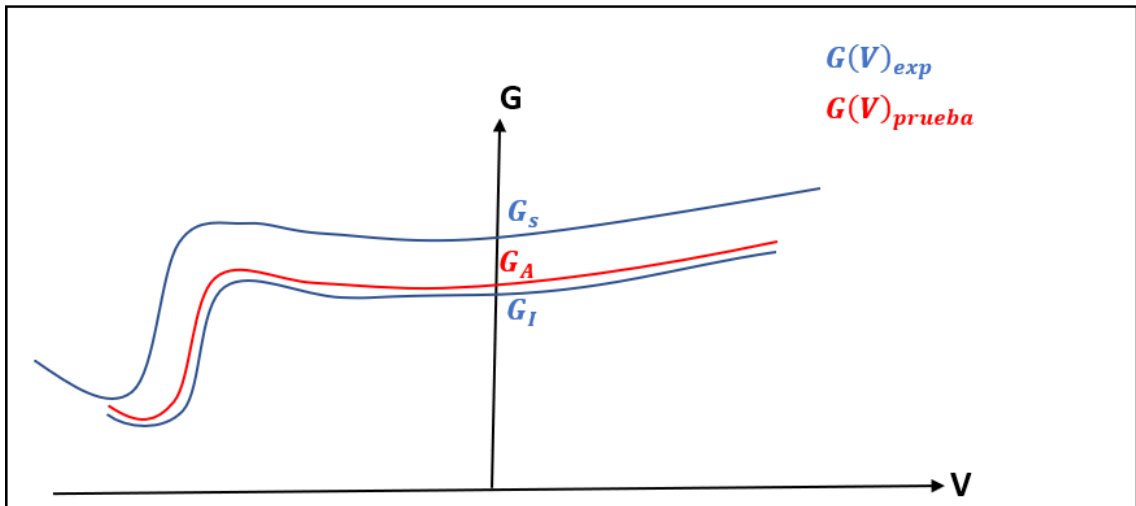


Ilustración 17 - Interpolación de las curvas de conductancia

No obstante, volviendo al ejemplo de la Ilustración 16, es de notar que el $M_{A-1} < M_A$, por lo que únicamente se seguirá la curva G_A hasta que se alcance como máximo V_{A-1} . A partir de este punto, se deberá seguir la curva de set ya que el material estará promocionando a otro estado LRS, por lo que será necesario recurrir a las curvas R_A . El proceso es idéntico al indicado para las conductancias.

- IV. Para interpolar los datos, se crea un conjunto de puntos virtuales $V_v(t)$ entre la tensión mínima de $G_I(V)$ y la tensión máxima de $G_I(V)$. Para cada uno de estos puntos virtuales, se buscan los entornos de las curvas inferior y superior en los que está contenido dicho punto. Se interpola entonces el valor G para cada una de ellas, y se promedian para sacar el punto virtual de la curva $G_A(V)$.

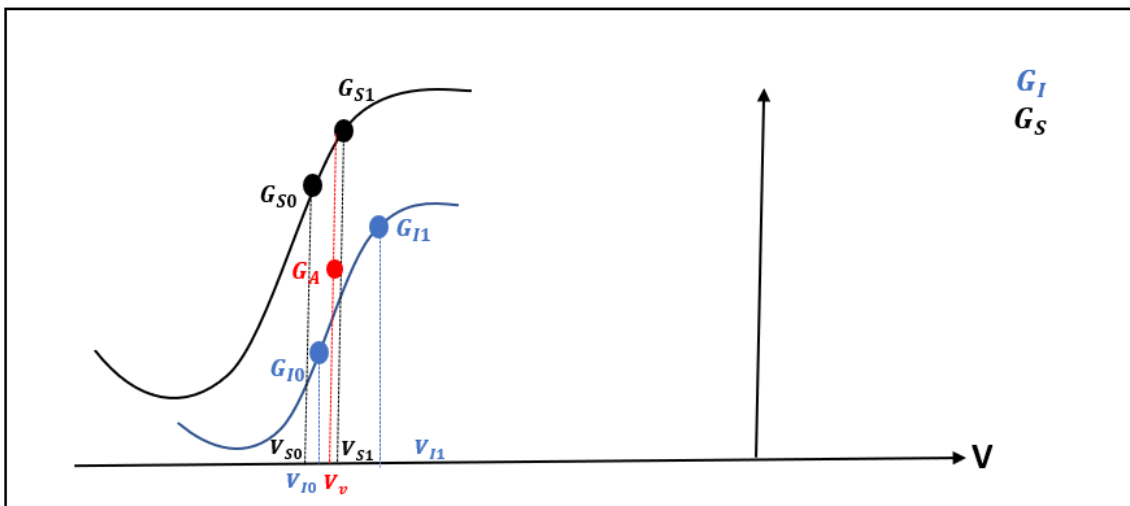


Ilustración 18 - Obtención de puntos virtuales de curvas de conductancia

De las expresiones de interpolación lineal, se obtiene:

$$G_S = G_{S0} + \frac{V_V - V_{S0}}{V_{S1} - V_{S0}} (G_{S1} - G_{S0})$$

Ecuación 3

$$G_I = G_{I0} + \frac{V_V - V_{I0}}{V_{I1} - V_{I0}} (G_{I1} - G_{I0})$$

Ecuación 4

Entonces, por la Ecuación 2, se obtiene el punto G_A buscado. Se repite así este proceso para cada punto virtual que se haya definido. Es necesario tener en cuenta que se está realizando una aproximación lineal al problema, por lo que el paso de tensiones con las que se hayan tomado los datos experimentales debe ser pequeño para obtener los mejores resultados, sobre todo en la zona de reset de la curva.

Si se tratara con las curvas de resistencia, el proceso sería el mismo.

Para finalizar, hay que volver a interpolar, ya que no se conoce exactamente el valor que toma G en los puntos de la función $V(t)$, sino que la información disponible es la de los puntos virtuales establecidos. Entonces, todo punto de $G(V)$ que se encuentre en el entorno de tensiones de $G_I(V)$, puede obtenerse mediante:

$$G_i(V_i) = G_{A,i-1}(V_i) + \frac{V_i - V_{A,i-1}}{V_{A,i+1} - V_{A,i-1}} (G_{A,i+1} - G_{A,i-1})$$

Ecuación 5

Idénticamente cuando se esté interpolando la curva de resistencias:

$$R_i(V_i) = R_{A,i-1}(V_i) + \frac{V_i - V_{A,i-1}}{V_{A,i+1} - V_{A,i-1}} (R_{A,i+1} - R_{A,i-1})$$

Conocidos G o R , se halla la corriente I para generar la característica $I(V)$.

$$R = \frac{V}{I}; G = \frac{I}{V}$$

2.4.4 Planteamiento de la función triangular.

Para asegurar el requisito funcional IV, se ha modelado la construcción de una función $V(t)$ triangular de n periodos, en que los picos de tensión se vayan incrementando en cada período T una cantidad ΔV_p y el primer pico tenga tensión V_0 .

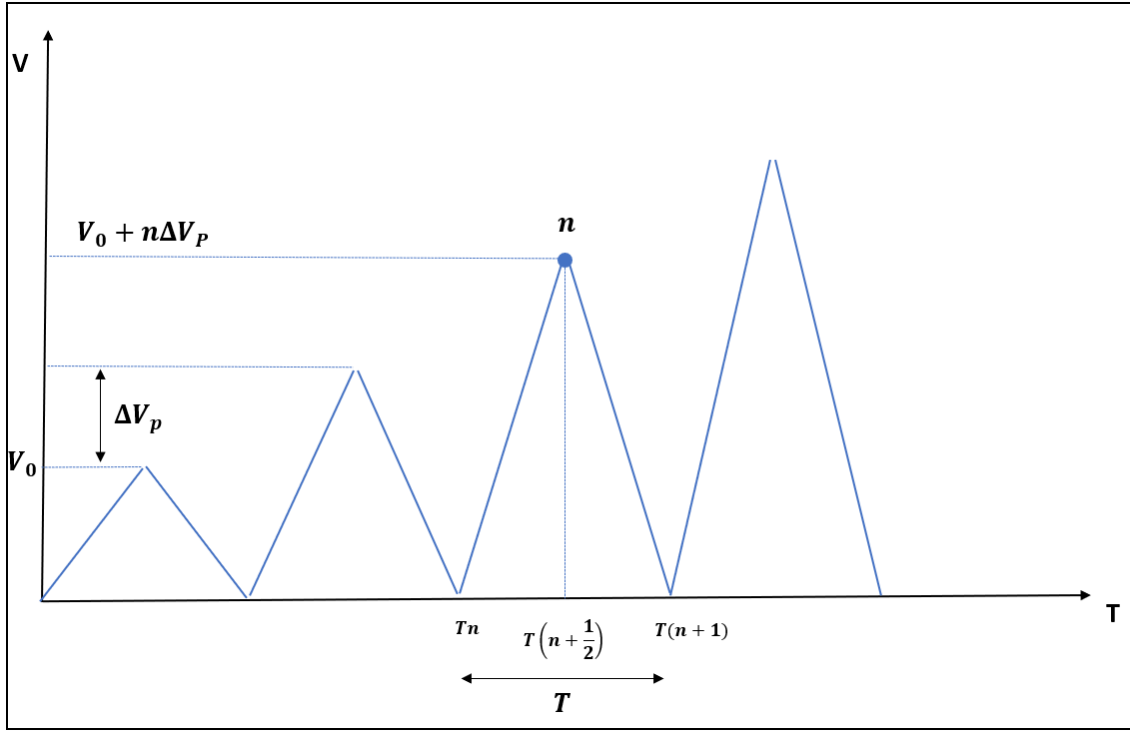


Ilustración 19 - Obtención de expresión para una función triangular creciente

De la Ilustración 19, se obtiene explícitamente la forma de la expresión $V(t)$ en el caso de una función triangular. En primer lugar, se observa que debe ser una composición de 2 funciones lineales, en las que la pendiente de cada recta será función de cada uno de los triángulos:

$$V(t; n) = V_{up}(t; n) + V_{down}(t; n)$$

Donde las funciones $V_{up}(t; n)$ y $V_{down}(t; n)$ vienen dadas por las expresiones:

$$V_{up}(t; n) = \alpha(n) + \beta(n)t$$

$$V_{down}(t; n) = \delta(n) + \gamma(n)t$$

Ahora se imponen las condiciones de contorno.

a. Las pendientes son iguales pero de diferente signatura:

$$\gamma = -\beta$$

b. Las funciones V_{up} y V_{down} se anulan cuando t es n veces T ($n = 0, 1, 2, \dots$):

$$V_{up}(Tn) = 0 \rightarrow \alpha = -\beta Tn$$

$$V_{down}(T(n+1)) = 0 \rightarrow \delta = \beta T(n+1)$$

c. En los puntos $t = T\left(n + \frac{1}{2}\right)$ ambas funciones adoptan el valor $V_0 + n\Delta V_p$:

$$\beta \left(T\left(n + \frac{1}{2}\right) - Tn \right) = V_0 + n\Delta V_p \rightarrow \beta = \frac{2(V_0 + n\Delta V_p)}{T}$$

Sustituyendo los parámetros por sus respectivos valores y haciendo algunas simplificaciones, las expresiones resultantes son:

$$V_{up}(t; n) = \frac{2(V_0 + n\Delta V_p)}{T}(t - Tn) \quad \forall t \in \left(Tn, T\left(n + \frac{1}{2}\right)\right)$$

$$V_{down}(t; n) = 2(V_0 + n\Delta V_p)\left(n + 1 - \frac{t}{T}\right) \quad \forall t \in \left(T\left(n + \frac{1}{2}\right), T(n + 1)\right)$$

2.4.5 Desarrollo de la capa lógica de DRM

Comprendido el desarrollo de la interfaz de usuario, y como se ha planteado el núcleo de DRM, se ha elaborado la capa encargada del tratamiento de los datos generados mediante las modificaciones que realice el usuario desde la citada interfaz (capa lógica). Por ende, la capa lógica es, en última instancia, la responsable de realizar los cálculos pertinentes para obtener, en base a unos datos de entrada, una simulación ante variaciones arbitrarias de la tensión de entrada $V(t)$.

Para su desarrollo, teniendo en cuenta la arquitectura modelada para DRM: los elementos encargados de implementar estas tareas deben encontrarse en el paquete métodos. Estos elementos son:

- ✓ ExtracionesDatosExperimentales.java.
Encargada de analizar las curvas $I(V)$ crecientes y decrecientes importadas en los ficheros csv de datos de entrada. Entre otras tareas, se encarga de:
 - a. Formatear a datos numéricos los datos importados y reconstruir así la curva $I(V)$.
 - b. Realizar los cálculos oportunos para obtener de esta las curvas $R(V)$, $G(V)$ y $G(P)$.
 - c. Extraer las tensiones en las que se produce un cambio de ciclo.
 - d. A partir de esas tensiones, separar los datos de la curva original en los diferentes niveles que la conforman (comportamiento multinivel del material que compone la RRam).

Tras completar este proceso, resulta un conjunto de datos disgregados según los niveles o ciclos de barrido a los que han sido sometidos los datos experimentales, y que se utilizarán como datos de entrada para la simulación. Así mismo, el lazo $I(V)$ y las curvas $R(V)$, $G(V)$ y $G(P)$ se notifican a la PantallaDatosEntrada.java, para poder visualizar los datos base que se han importado, y servir así de apoyo a la caracterización.

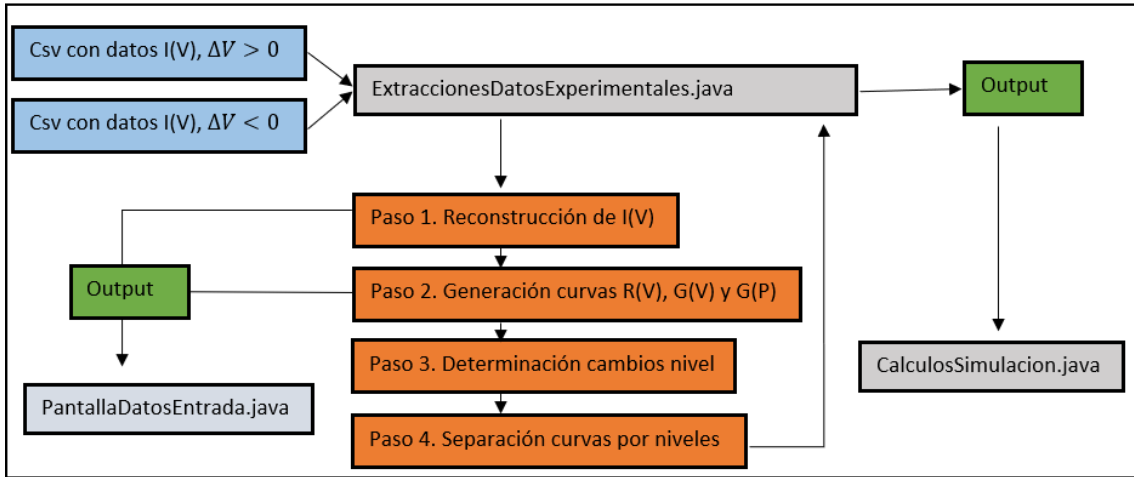


Ilustración 20 - Ciclo de vida de *ExtraccionesDatosExperimentales.java*

✓ *GeneradoraTensiones.java*.

Su misión consiste en generar o importar las curvas $V(t)$ de las que se pretende conocer o simular la característica $I(V)$. En el caso de los barridos predefinidos, ha sido necesario conocer explícitamente la expresión de estas funciones $V(t)$. Para el barrido manual, únicamente se importan los datos $V(t)$ desde un fichero csv.

Estos datos sirven como entrada para la simulación, y además se envían a cada una de las pantallas para disponer en ellas de la información de la forma que adopta la tensión de entrada.

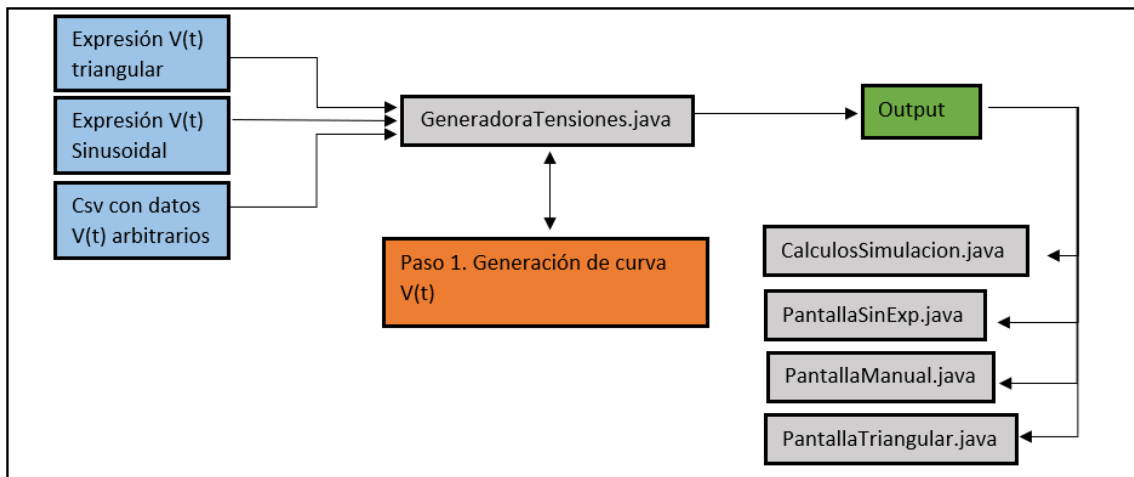


Ilustración 21 - Ciclo de vida *GeneradoraTensiones.java*

✓ *CálculosSimulación.java*

Tiene como objetivo implementar el núcleo de DRM, es decir, generar las características $I(V)$ de una función de entrada $V(t)$ arbitraria, en base a los datos disponibles introducidos en un primer barrido (creciente o decreciente) en tensiones. Estos datos se proporcionan a las pantallas que permiten su visualización. Para el desarrollo, se han seguido los siguientes pasos, extraídos en el planteamiento:

- a. Obtener los máximo de la tensión de entrada.

- b. A partir de esos máximos, determinar las curvas de los niveles inferior y superior comparando esos máximos de tensión con los máximos de los datos experimentales, que ya están disgregados por ciclos.
- c. Decidir si a tiempo t , ΔV es mayor o menor que cero, para determinar si la interpolación se debe hacer a través de las curvas $R(V)$ o $G(V)$.
- d. Si $\Delta V \geq 0$, entonces debe comprobar si en el ciclo previo se ha producido un reset completo del material. Si la respuesta ha sido afirmativa, tiene que volver siguiendo $R(V)$, mientras que si es negativa, regresará siguiendo $G(V)$.
- e. Si $\Delta V < 0$, siempre se continuará por la $G(V)$.
- f. Debe calcular cuan cerca o lejos se presenta el máximo de $V(t)$ de los máximos de los niveles inferior y superior.
- g. Conocida esta proximidad, el paso es interpolar todos los datos intermedios entre las dos curvas cercanas, y realizar una nueva interpolación para conocer el valor de I exacto a tiempo t .

Esta secuencia de pasos podría resumirse en el siguiente diagrama de flujo:

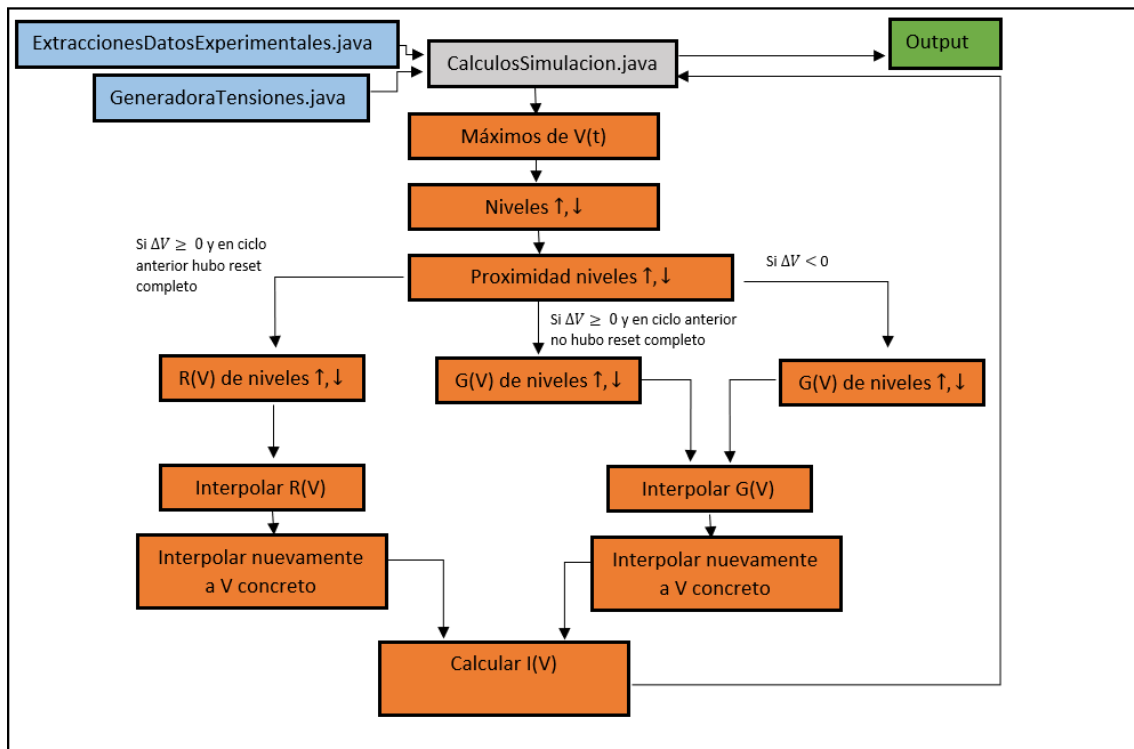


Ilustración 22 - Diagrama de flujo del Core de DRM

Conocidos por separado cada uno de los flujos individuales, puede sintetizarse el comportamiento de la capa lógica a alto nivel.

- a. Importar los datos experimentales.
- b. Generar las tensiones $V(t)$.
- c. Calcular las curvas simuladas $I(V)$.

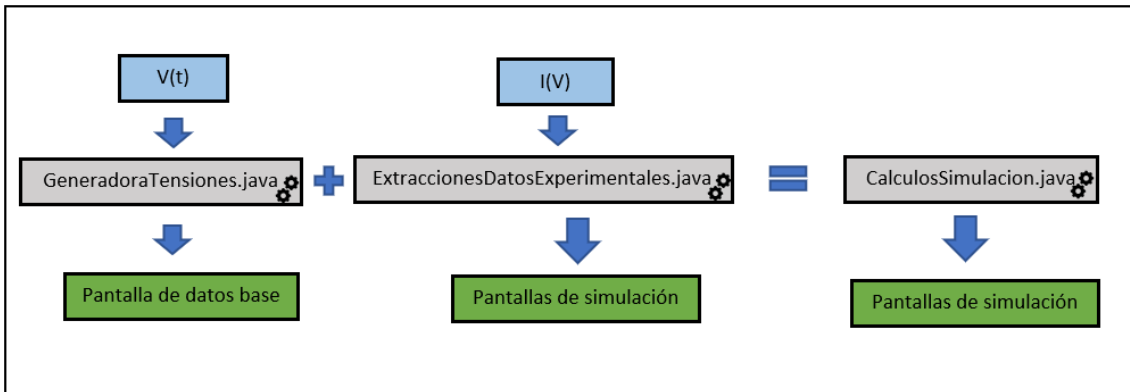


Ilustración 23 - Comportamiento de capa lógica a alto nivel

2.5 Fase IV. Pruebas de DRM

La necesidad de que un buen software no presente errores de ejecución o comportamientos inesperados que impidan su óptimo funcionamiento, ha hecho necesario plantear una pequeña fase de pruebas en la que se verifiquen varios aspectos:

- ✓ En primer lugar, no deben producirse errores en tiempo de ejecución que bloqueen el software.
- ✓ En segundo lugar, se debe verificar que todos los requisitos funcionales están correctamente implementados y desarrollados.
- ✓ En tercer lugar, se debe comprobar que el aspecto visual de la aplicación se corresponde con el definido.
- ✓ Por último, debe validarse el método implementado en el núcleo, es decir, que se generen correctamente las curvas simuladas.

Los 3 primeros puntos, se han etiquetado como pruebas generales funcionales, mientras que el último, como pruebas generales de calibración.

2.5.1 Pruebas generales funcionales

Primero se ha comprobado que no se produzcan errores de ejecución, para lo cual se ha verificado la correcta captura de excepciones ante varias circunstancias que pudieran darse:

- a. La ruta de los ficheros base no existe.
- b. Se configura un valor mínimo del parámetro superior al máximo.
- c. Se intenta introducir texto en campos numéricos.
- d. El fichero no tiene la extensión esperada.

Posteriormente, se ha navegado por cada pantalla y se han realizado las interacciones necesarias para verificar que el sistema cumple con los requisitos especificados en la Fase I.

Por último, se ha validado que el aspecto visual de DRM sea el definido en la fase II:

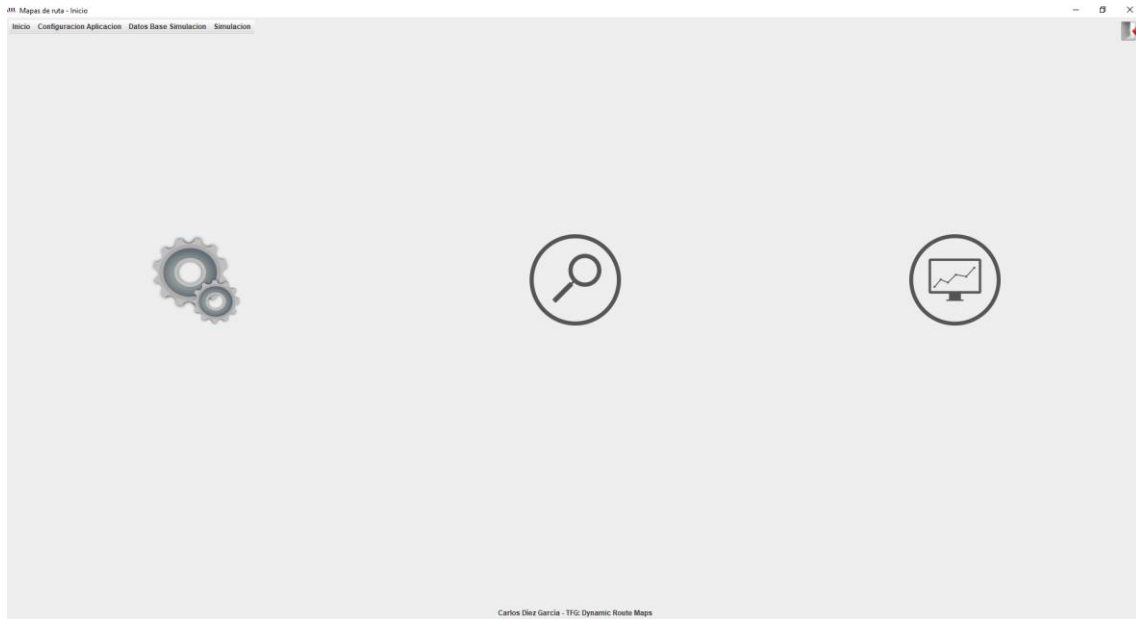


Ilustración 24 - Pantalla de Inicio

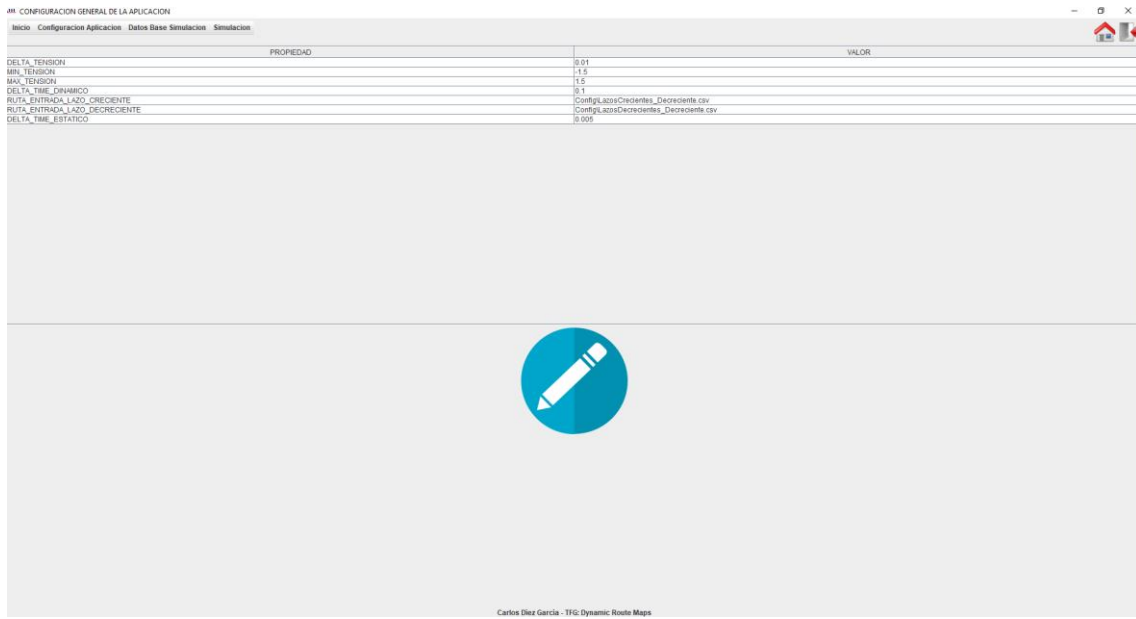


Ilustración 25 - Pantalla de configuración

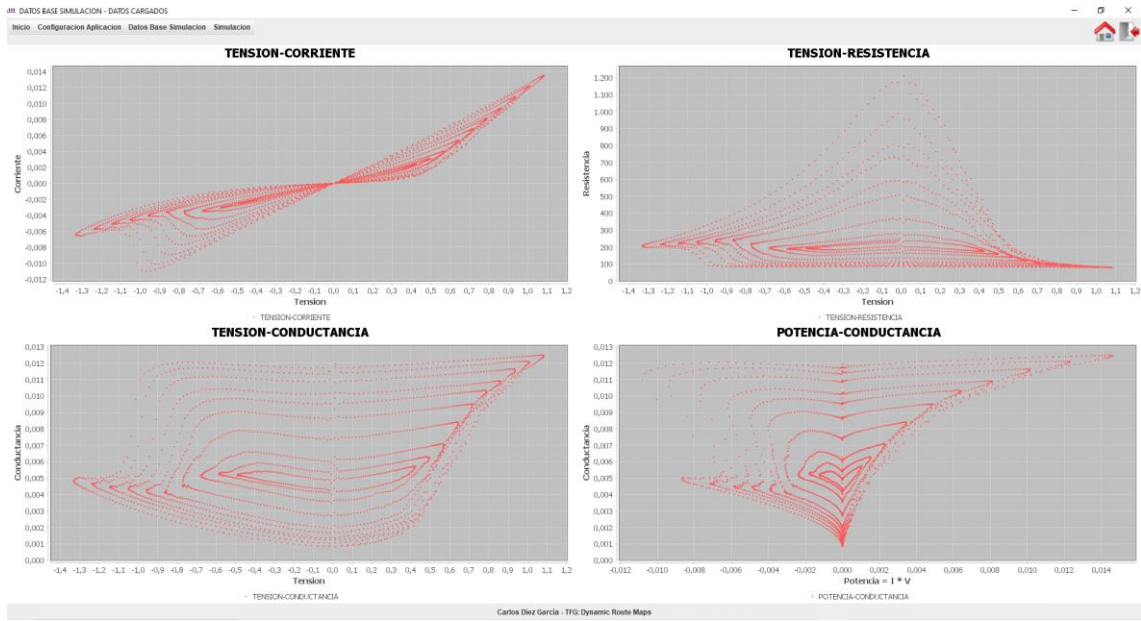


Ilustración 26 - Pantalla de datos base

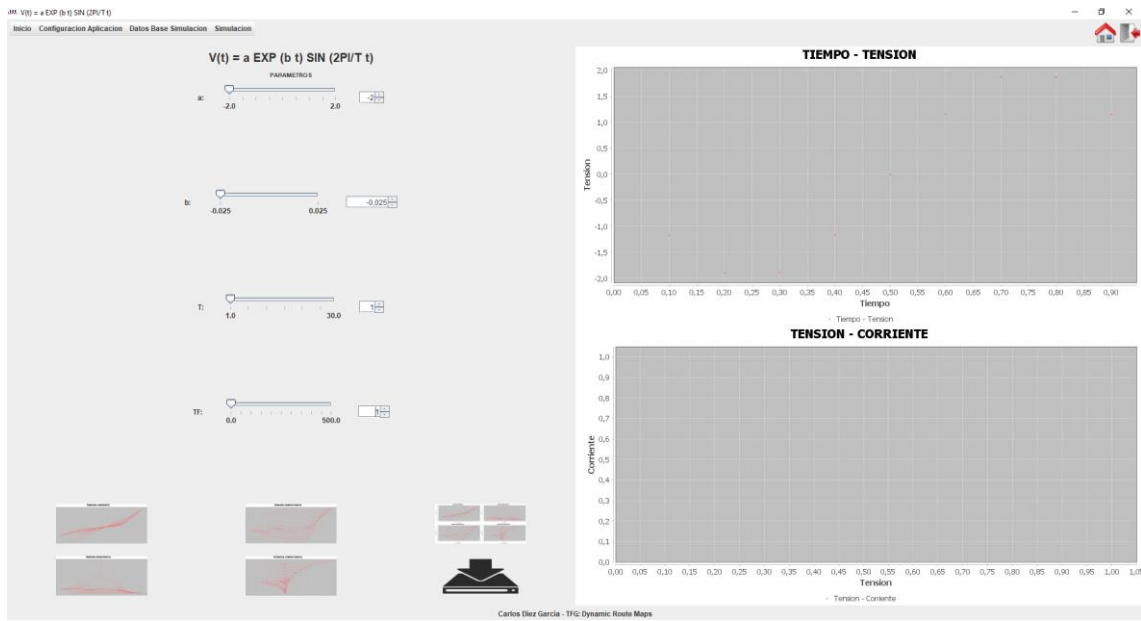


Ilustración 27 - Pantalla de simulación de barrido predefinido

2.5.2 Pruebas generales de calibración

Comprobados los aspectos funcionales, se ha verificado que el algoritmo planteado en la fase III, genere (con sus limitaciones) la misma característica $I(V)$ que se ha introducido, cuando el barrido manual elegido se corresponde con los datos de entrada.

Para ello, primero se han visualizado los datos de entrada decrecientes:

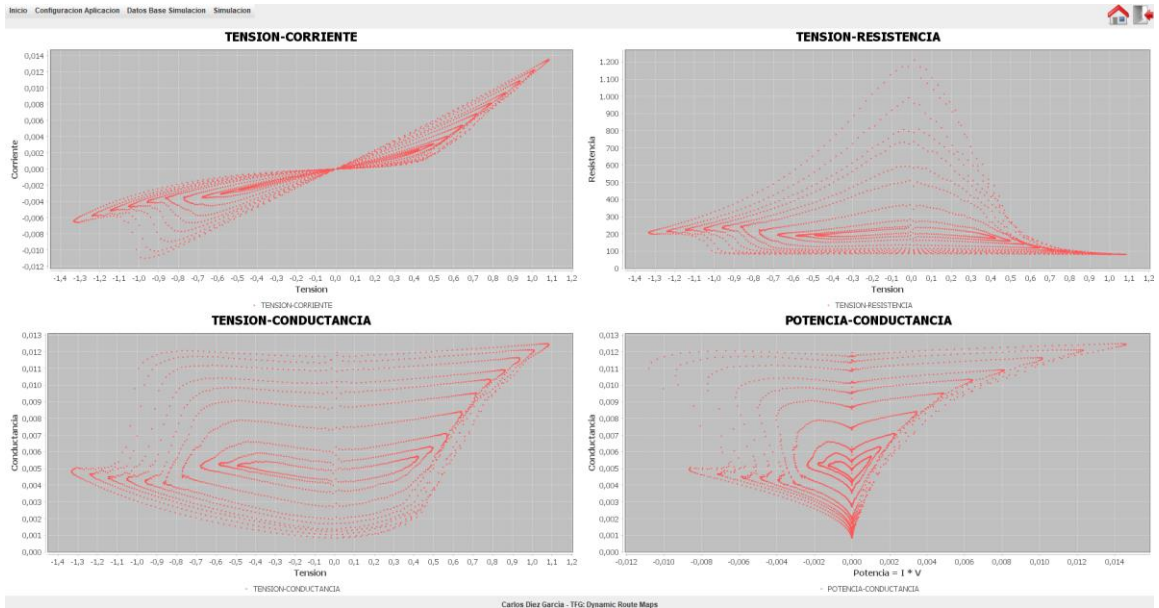


Ilustración 28 - Visualización de datos de entrada

Después se ha llamado al barrido manual introduciendo la misma $V(t)$ de los datos base:

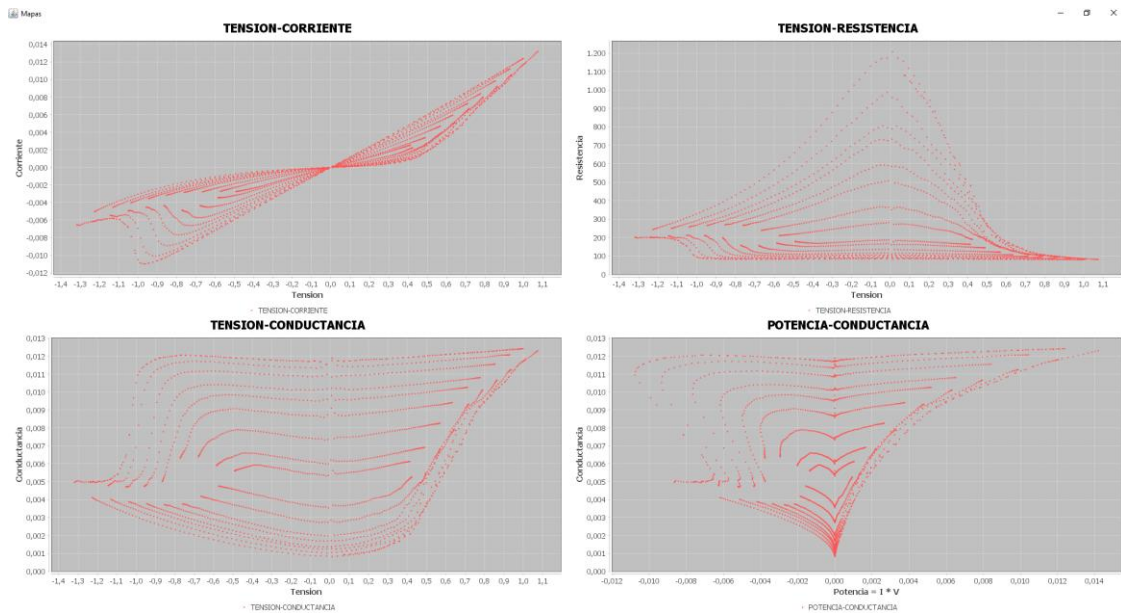


Ilustración 29 - Validación del algoritmo

De la comparativa de Ilustración 28 con Ilustración 29, se comprueba que el algoritmo empleado responde correctamente cuando los datos coinciden con los de entrada. También es cierto que las curvas de los diferentes niveles no llegan a cerrarse por las consideraciones expuestas en el planteamiento (Ilustración 17).

Por último, se han exportado estas curvas a un fichero e insertado la curva real y la simulada en la misma gráfica:

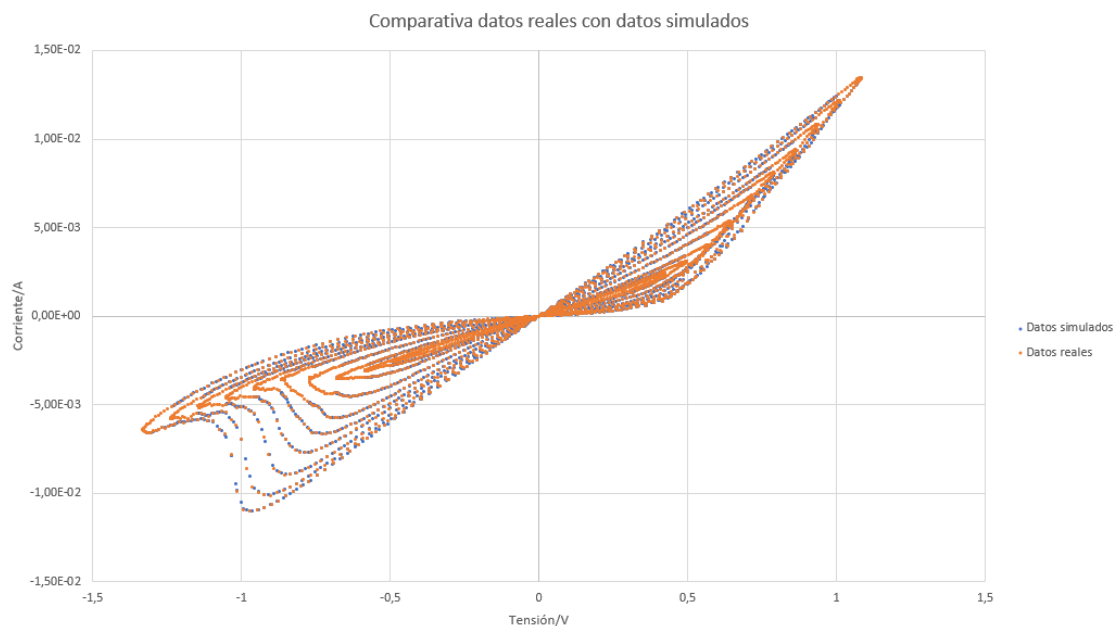


Ilustración 30- Comparativa datos reales con datos generados por DRM

Con las limitaciones expuestas, se concluye que el método de esta comparativa que el método está correctamente calibrado, y puede emplearse para aproximar el comportamiento de un material ante variaciones arbitrarias de la tensión fuente. En el capítulo 3 se verán varias aplicaciones de DRM.

2.6 Fase V. Exportación del ejecutable DRM

Tras la finalización de la fase de desarrollo, se ha exportado el proyecto como ejecutable .jar que implementa todas las librerías y paqueterías necesarias para su ejecución, de modo que no se requiera de ninguna instalación adicional y software pueda emplearse como fichero portable.

En el Anexo I del documento se encuentra el manual de usuario de DRM.

2.7 Limitaciones

Tras una profunda revisión del código fuente y la batería de pruebas realizadas durante la fase IV, se concluyen una serie de limitaciones del modelo empleado:

- ✓ Para determinadas funciones $V(t)$ de entrada, parece que el algoritmo implementado en la capa lógica no responde correctamente, ya que se dibujan curvas, o a parecen algunas que no deberían aparecer. Sería necesario realizar una segunda batería de pruebas para localizar el foco de estos errores y solucionarlos. En concreto, parece no funcionar correctamente cuando se mezclan tramos consecutivos de “Reset” y “No reset”.

- ✓ La configuración de algunos parámetros parece fallar bajo ciertas combinaciones de los mismos.

2.8 Mejoras o funcionalidades futuras

Como posibles mejoras a implementar de cara a conseguir mejores resultados se contemplan:

- ✓ El planteamiento de formas $V(t)$ no definidas en la versión actual del software, que sean configurables desde nuevas pantallas.
- ✓ La integración en un mismo gráfico de los datos experimentales con los datos simulados.
- ✓ Implementar un método que ajuste para cada ciclo, los puntos experimentales a funciones polinómicas de grado n , y que permitan obtener explícitamente nuevos puntos no definidos, consiguiendo simular de este modo los puntos que actualmente no tienen representación.
- ✓ Sustituir la interpolación lineal en la zona de reset por una interpolación más realista, como por ejemplo, la oblicua.

Capítulo 3.

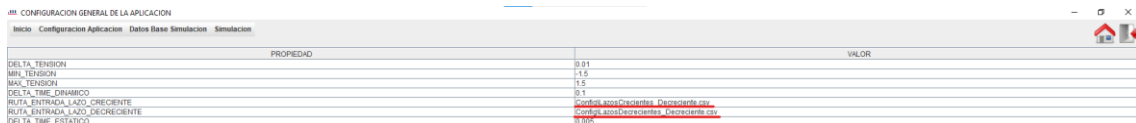
Simulación de curvas I(V) con DRM

3.1 Introducción

Para explicitar el objetivo de DRM, se utiliza el software desarrollado para estudiar y analizar las características tensión-corriente del material HfO_2 del que se ha tomado el barrido experimental.

3.2 Configuración inicial

Primero se indican las rutas en las que se encuentran los datos disponibles del material, medidos en un barrido y separados por tensiones crecientes y decrecientes. En el caso que se presenta, los datos se encuentran en el mismo directorio “Config”, por lo que se puede indicar la ruta relativa tomando como base la ruta del fichero ejecutable (el directorio raíz de Config y el debe ser el mismo):



PROPIEDAD	VALOR
DELTA_TENSION	0.01
MIN_TENSION	-1.5
MAX_TENSION	1.5
DELTA_TIME_DINAMICO	0.1
RUTA_ENTRADA_LAZO_CRECIENTE	config\datos\Crecientes_Decremente.csv
RUTA_ENTRADA_LAZO_DECRECIENTE	config\datos\Crecientes_Decremente.csv
DELTA_TIME_ESTADICO	0.001

Ilustración 31 - Selección de ficheros de entrada

Estas rutas solo se modifican en caso de cambiar de muestra, por lo que en lo sucesivo, no se volverán a editar.

3.3 Barrido Sinusoidal Exponencial

Se estudia el caso de una función de entrada aplicada durante 500s de la forma:

$$V(t) = 0,75V \operatorname{Sin}\left(\frac{2\pi}{30s}t\right) e^{0,001s^{-1}t}$$

Habiendo seleccionado la pestaña de barrido sinusoidal exponencial, se van configurando los parámetros mediante los sliders, spinners o introduciendo el valor deseado siempre que se encuentre en el rango configurado. A medida que se realizan modificaciones, a la derecha de la pantalla se irán representando los gráficos $V(t), I(V)$. Como la función escogida oscila rápidamente, estos gráficos

dinámicos no serán muy buenos. No obstante, si se pulsa en los botones de generación, se representará un mayor número de puntos.

Pasemos a analizar ahora lo que sucede en este barrido:

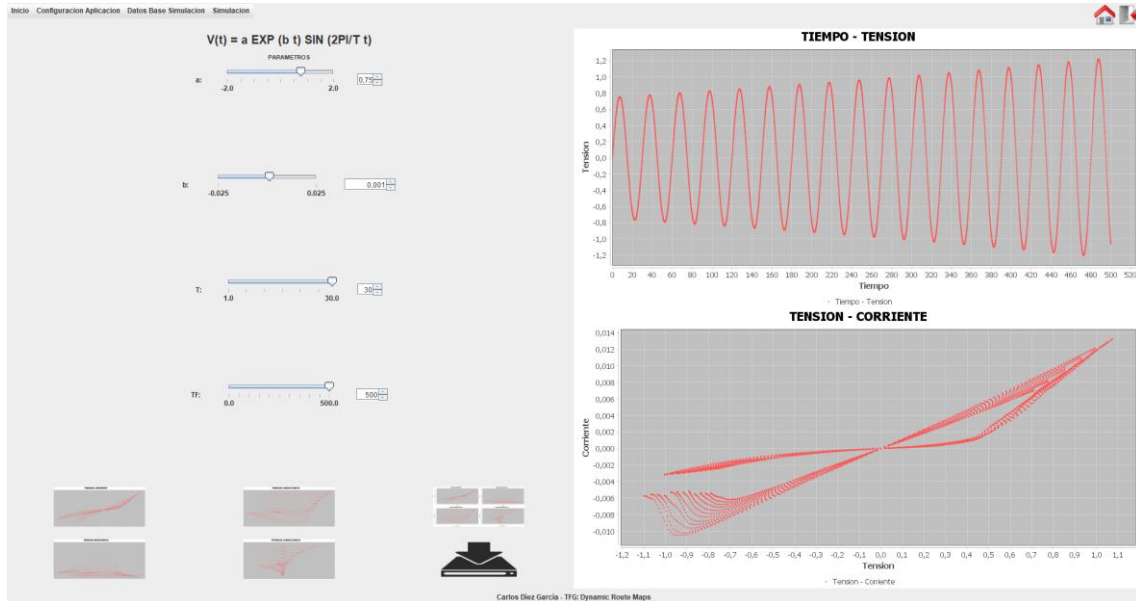


Ilustración 32 - Análisis $V(t)$ e $I(V)$ de barrido sinusoidal exponencial

Se produce un primer set a LRS cuando se eleva inicialmente la tensión a 0,75V. A partir de este momento, cuando baja la tensión, la corriente cae hasta intentar llegar a reset completo para esa tensión máxima de partida. Cuando esto no sucede, no hay cambio a HRS, y al incrementar la tensión la corriente regresa por la misma curva y promociona hasta el siguiente estado de baja resistencia.

Para comprender mejor las transiciones de estado, se analizan el resto de curvas:

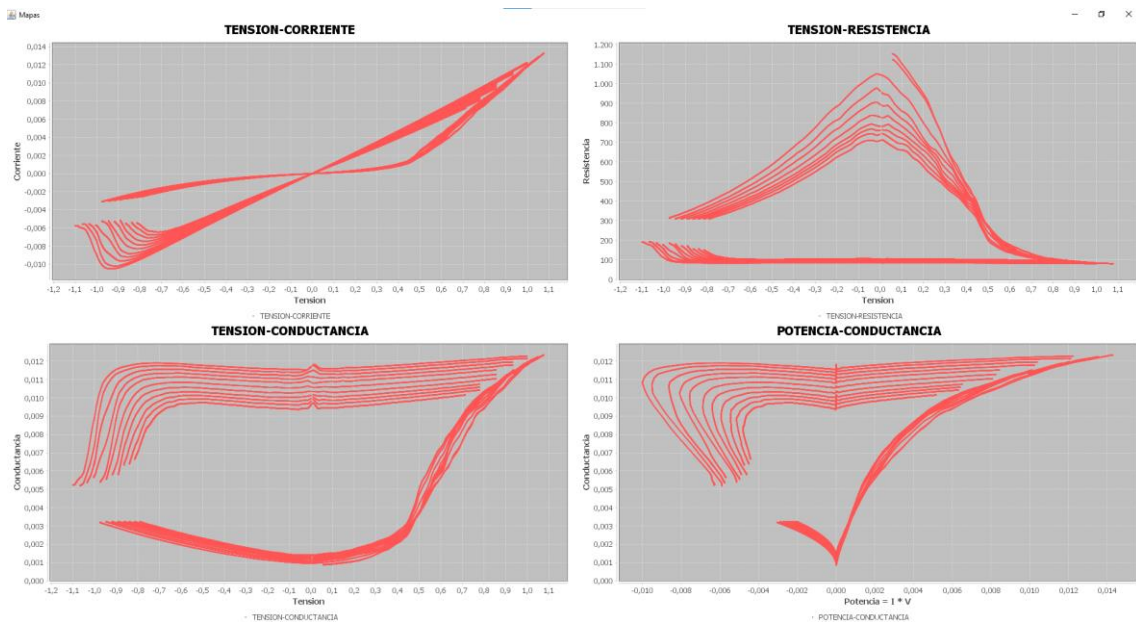


Ilustración 33 - Análisis conjunto para $V(t)$ sinusoidal exponencial

Si se modifica ahora la función de entrada por una que recorra más rango de tensiones y oscile más rápidamente, se obtiene el mapa completo de ciclos y transiciones $HRS \rightarrow LRS$ y viceversa:

$$V(t) = 0,45V \sin\left(\frac{2\pi}{2s}t\right) e^{0,002s^{-1}t}$$

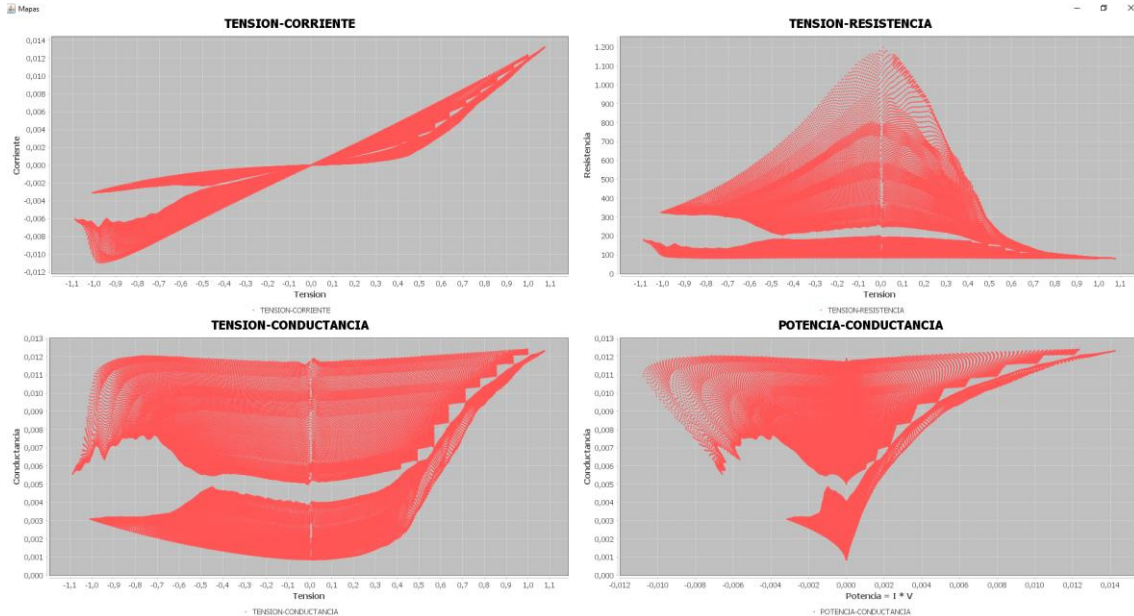


Ilustración 34 - Mapa completo multinivel del óxido de hafnio

En vista de la Ilustración 34, se infiere que a medida que se incrementa progresivamente la tensión de set, las curvas tienden a un estado de set máximo, a partir del cual se producirá la ruptura del material. Así mismo, centrando la atención en la curva de conductancia-potencia, puede extraerse la tendencia que siguen los puntos en los que comienza el reset, y pudiera ser que esta tendencia sugiera la máxima tensión a la que ha sido sometido el dispositivo cuando la pendiente de dicha curva se anule (curva compliance⁸).

Así mismo, también refleja la asimetría entre los estados de set y reset, y como la formación del filamento conductor es más lenta que su destrucción.

⁸ Se entiende como curva compliance a la característica I-V obtenida en el momento de máxima aplicación de voltaje al dispositivo en directa, dependiendo todos los estados intermedias de esta curva.

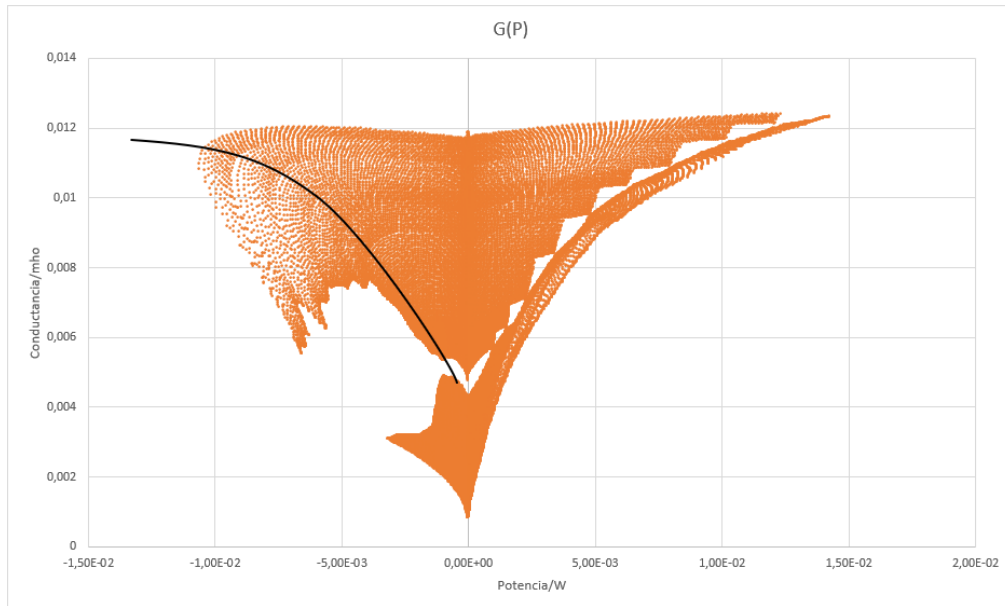


Ilustración 35 - Tendencia de los puntos de reset

3.4 Barrido triangular

Con la finalidad de comprensión de aquellos casos en los que no pueda producirse un reset completo, se ha analizado el caso de una tensión $V(t)$ triangular en directa, en la que en ningún caso puede producirse un reset. Se han establecido unos parámetros tales que permitan cómodamente la visualización, y por consiguiente, el análisis:

$$V_{up}(t; n) = \frac{2(V_0 + n\Delta V_p)}{T}(t - Tn) \quad \forall t \in \left(Tn, T\left(n + \frac{1}{2}\right)\right)$$

$$V_{down}(t; n) = 2(V_0 + n\Delta V_p)\left(n + 1 - \frac{t}{T}\right) \quad \forall t \in \left(T\left(n + \frac{1}{2}\right), T(n + 1)\right)$$

Donde, el número máximo de triángulos $n = 10$, el periodo de cada triángulo es de 30s, la tensión del primer pico es de 0,45V y la tensión crece en cada pico 0,07V:

CAPÍTULO 2 – SIMULACIÓN DE CURVAS I(V) CON DRM

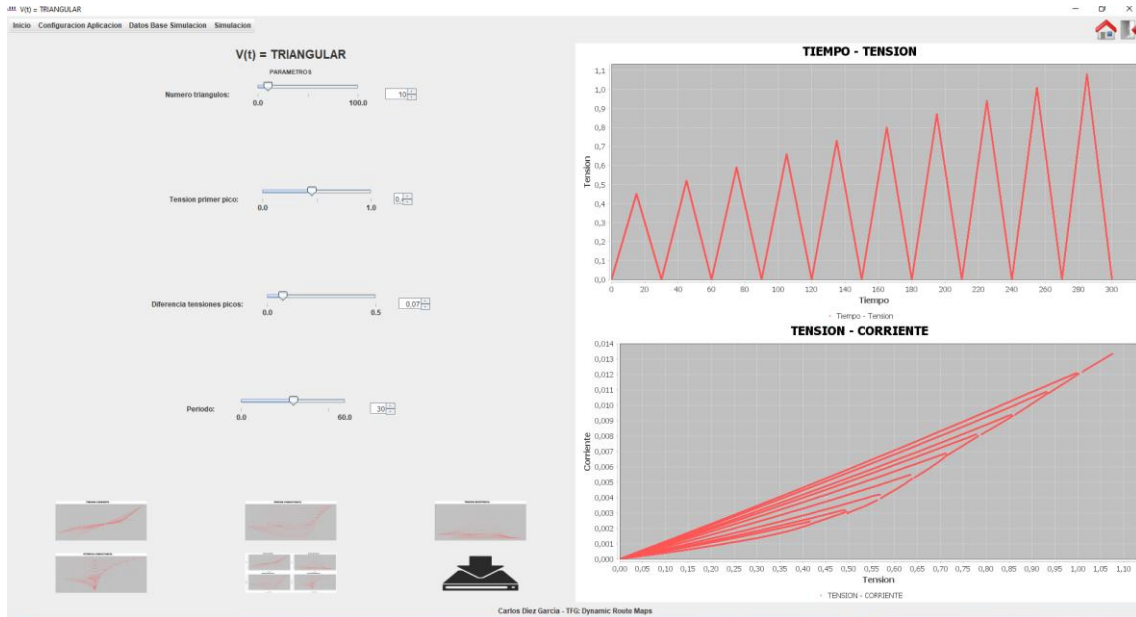


Ilustración 36 - Barrido triangular donde no se produce ningún reset

En el primer pico, se produce un primer set hasta 0,45V (estado LRS). Como después la tensión cae a 0, no se produce la transición $LRS \rightarrow HRS$. Por lo tanto, al volver a incrementarse la tensión, la corriente regresa por el mismo camino de partida hasta alcanzar la tensión máxima anterior. Así pues, superada dicha tensión, promociona al siguiente nivel de set a 0,52V. El sistema sigue este proceso de promociones hasta alcanzar la máxima tensión.

Esto también se ilustra con el resto de gráficas, donde también se ve la tendencia del sistema hacia una curva común.

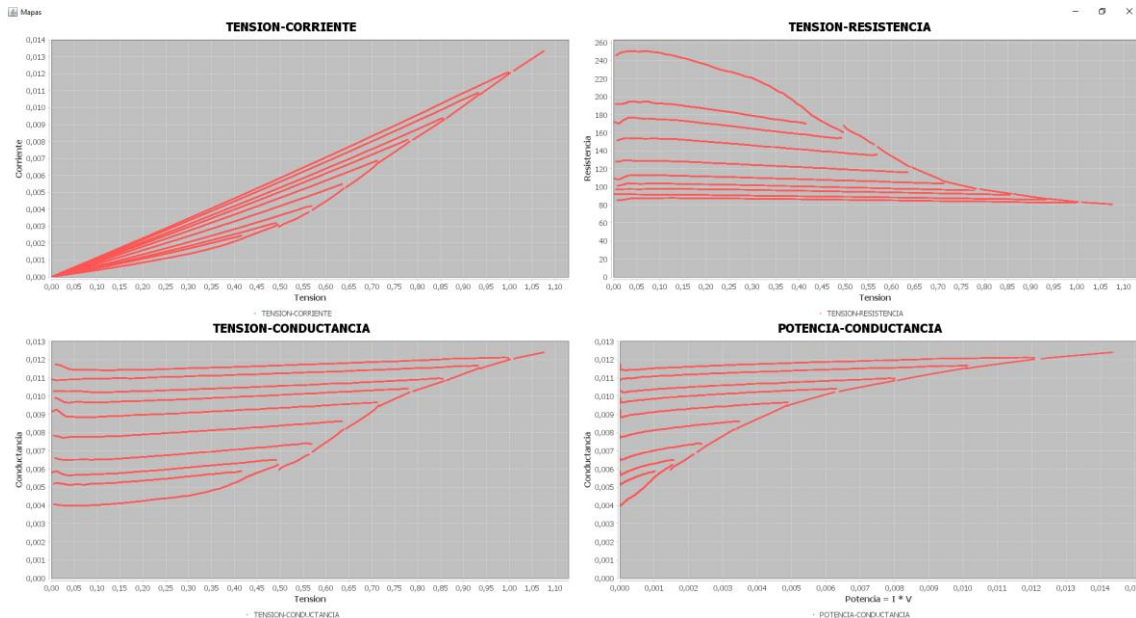


Ilustración 37 - Gráfico conjunto para simulación triangular.

3.5 Función $V(t)$ amortiguada

Se analiza ahora la entrada de tensión

$$V(t) = 0,65V \sin\left(\frac{2\pi}{30} t\right) e^{-0,001t}$$

En este caso, no se alcanzará nunca la condición de reset completo, ya que el barrido se realiza en tensión decreciente no alcanzando nunca el mínimo de tensión necesario para que este se produzca. Aparece un primer set de acuerdo a la primera subida de tensión, y después el sistema permanece oscilando en el mismo estado en el que se encuentre (no aparece ninguna transición $LRS \rightarrow HRS$).

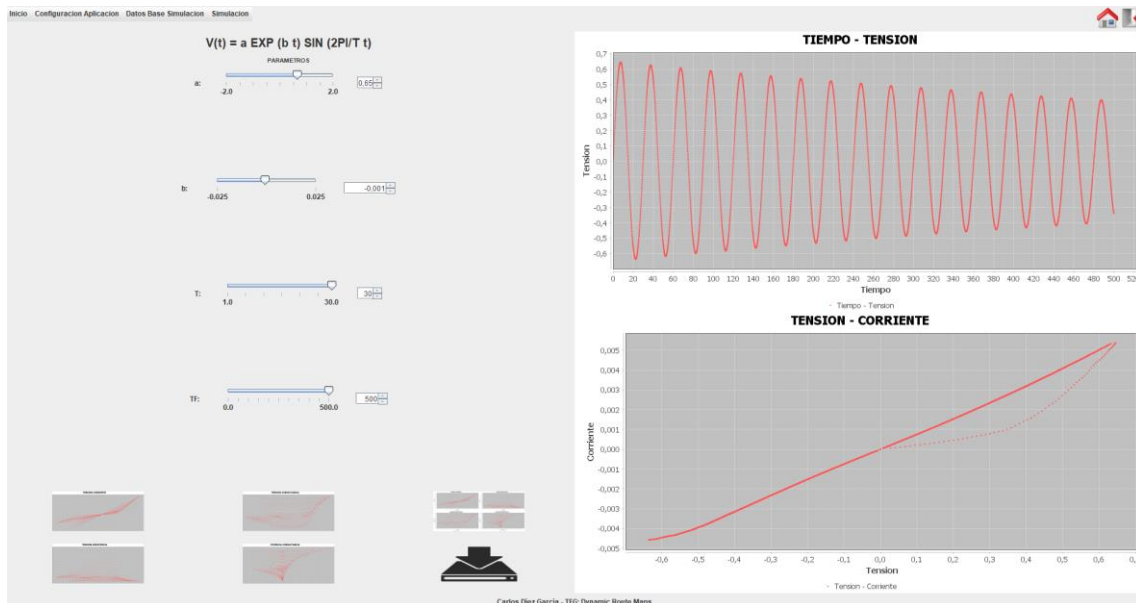


Ilustración 38 - $V(t)$ Amortiguada

3.6 Función $V(t)$ arbitraria

Para terminar con el estudio de aplicación de DRM, se simula una $V(t)$ arbitraria.



Ilustración 39 - Función $V(t)$ arbitraria

- En el tramo 1, la tensión se incrementa hasta los 0,8V produciéndose el set inicial.
- Al disminuir la tensión en el tramo 2 hasta -0,85V, se produce un set que permite la transición $LRS \rightarrow HRS$, por lo que en el tramo 3 vuelve por la curva correspondiente a dicho estado.
- Si se incrementa de nuevo la tensión, debería visualizarse la curva que llega hasta 1V, pero no se está mostrando. No obstante, sí que se visualiza parte de la curva de bajada de tramo 4 hasta los -0,8V, y la subida hasta 0,4V por el tramo 5 que se realiza por la misma curva, ya que no hay transición de estado.

El que no se muestre la curva de subida puede deberse bien a un problema en la configuración, o bien a un fallo en la lógica implementada al modelo. Para su corrección, sería necesaria una revisión del código en una versión 2.0 de DRM, ya que el software se comporta correctamente ante variaciones regulares de la tensión.

Conclusiones

En este trabajo se han revisado las propiedades de las memorias de acceso aleatorio resistivo, dando una breve introducción teórica al mecanismo de conmutación resistiva en estructuras Metal-Aislante-Metal, en las que el aislante es un material óxido.

En particular, se ha expuesto la presencia de múltiples estados intermedios de alta y baja resistencia en las RRAM, y de como podrían ser la solución a las exigencias de precisión en la determinación de los pesos asociados a cada neurona en los sistemas neuromórficos implementados por algoritmos de aprendizaje automático e inteligencias artificiales.

Debido a esta necesidad de comprender y obtener estos estados intermedios, se ha detallado el proceso de elaboración del software DRM, capaz de recibir como entrada mediciones experimentales de diferentes ciclos correspondientes a estados intermedios de una RRAM de material aislante HfO_2 . Se ha realizado este proceso asumiendo una interpolación lineal para obtener los datos entre 2 curvas experimentales de diferente nivel consecutivas.

Este software desarrollado permite el análisis y caracterización de materiales mediante la visualización de las curvas experimentales de sus parámetros de admitancia, así como la predicción de estados intermedios nuevos no obtenidos en las mediciones experimentales.

El software ha sido testeado introduciendo unos datos de entrada similares los obtenidos en el barrido experimental, arrojando unos resultados similares.

Se han analizado algunas limitaciones y puntos de mejora de DRM, como son el estudio de funciones de interpolación no lineales en el entorno de los puntos de cada nivel donde se produce el reset, así como la revisión del método para ciertas funciones de entrada.

Para finalizar, se ha utilizado DRM para construir los mapas de ruta de algunas entradas $V(t)$ arbitrarias analizando cada uno de los tramos, y para obtener un mapa lo más completo posible del esquema multinivel de la estructura MIM basada en HfO_2 , obteniéndose una tendencia de los puntos de reset hacia la curva compliance.

Referencias

1. Yu S, Member S, Shim W, Peng X, Member GS. RRAM for Compute-in-Memory : From Inference to Training. 2764 IEEE Trans CIRCUITS Syst Regul Pap. 2021;68(7):2753–65.
2. Ossorio OG, Vinuesa G, Garcia H, Sahelices B, Dueñas S, Castán H, et al. Performance Assessment of Amorphous HfO₂-Based RRAM Devices for Neuromorphic Applications O. G. Ossorio. In Chicago; 2021.
3. Ilias A. Tayeb, Zhao F, Abdullah JM, Cheong KY. Resistive Switching Behaviour in Polymannose Film for Multistate Non-volatile Memory Application. Mater Chem C. 2020;
4. Lee JS, Lee S, Noh TW. Resistive switching phenomena: A review of statistical physics approaches. Appl Phys Rev. 2015;2(3).
5. Wang H, Yan X. Overview of Resistive Random Access Memory (RRAM): Materials , Filament Mechanisms , Performance Optimization , and Prospects. 2019;1900073:1–12.
6. Milo V, Zambelli C. Multilevel HfO₂-based RRAM devices for low-power neuromorphic networks Multilevel HfO₂-based RRAM devices for low-power neuromorphic networks. 2019;081120(April). Available from: <https://doi.org/10.1063/1.5108650>
7. Gómez JMM. Enfoque metodológico para el diseño de interfaces. Gerenc Tecnol Inform. 2013;12:59–73.
8. Reynoso CB. Introducción a la Arquitectura de Software. 2004;
9. Sánchez J. Java 2 Incluye Swing, Threads, programación en red, JavaBeans, JDBC y JSP / Servlets. 2004;

Anexo I. Manual de Usuario de DRM



Facultad de Ciencias – Universidad de Valladolid

Dynamic-Route-Maps

Manual de Usuario – DRM Versión 1.0.0

Carlos Diez García
30-6-2021



Índice de contenidos

1. INTRODUCCIÓN	4
2. SOFTWARE NECESARIO.	4
3. INSTALACIÓN Y CONFIGURACIÓN DE DYNAMIC-ROUTE-MAPS.....	4
4. DETALLE DE COMPONENTES DE LA APLICACIÓN	5
4.1 ELEMENTOS COMUNES.....	5
4.2 PANTALLA DE INICIO.	6
4.2.1 <i>Icono Configuración.</i>	6
4.2.2 <i>Icono Datos Base.</i>	7
4.2.3 <i>Icono Simulación.</i>	7
4.3 PANTALLAS DE CONFIGURACIÓN.	8
4.3.1 <i>Pantalla de Configuración General</i>	8
4.3.2 <i>Pantalla Configuración Parámetros Barrido Sinusoidal Exponencial</i>	10
4.3.3 <i>Pantalla Configuración Parámetros Barrido Triangular</i>	11
4.4 PANTALLA DE DATOS BASE	13
4.5 PANTALLA DE SIMULACIÓN	15
4.5.1 <i>Pantalla de Barrido Sinusoidal Exponencial</i>	17
4.5.2 <i>Pantalla de Barrido Triangular</i>	18
4.5.3 <i>Pantalla de Barrido Manual</i>	18
5. ERRORES FRECUENTES.....	20
6. CONSIDERACIONES FINALES	22



Índice de ilustraciones

ILUSTRACIÓN 1 - ELEMENTOS COMUNES -----	5
ILUSTRACIÓN 2 - CONFIRMACIÓN PARA SALIR DE LA APLICACIÓN -----	5
ILUSTRACIÓN 3 - ICONO DE INICIO -----	5
ILUSTRACIÓN 4 - PANTALLA DE INICIO DE DRM -----	6
ILUSTRACIÓN 5 - TOOLTIP EXPLICATIVO DEL BOTÓN. -----	6
ILUSTRACIÓN 6 - ELECCIÓN DE PANTALLA DE CONFIGURACIÓN -----	6
ILUSTRACIÓN 7 - ELECCIÓN DE PANTALLA DE SIMULACIÓN -----	7
ILUSTRACIÓN 8 - ERROR CARÁCTER NO NUMÉRICO -----	8
ILUSTRACIÓN 9 - PANTALLA DE CONFIGURACIÓN GENERAL -----	8
ILUSTRACIÓN 10 - FORMATO CORRECTO DE DATOS DE ENTRADA EN FICHEROS .CSV -----	13
ILUSTRACIÓN 11 - ERROR AL LEER LOS CSV DE DATOS BASE -----	13
ILUSTRACIÓN 12 - PANTALLA DE DATOS BASE -----	14
ILUSTRACIÓN 13 - GRÁFICO GENERADO DINÁMICAMENTE POR MANIPULACIÓN DE PARÁMETROS -----	15
ILUSTRACIÓN 14 - MISMOS PARÁMETROS PERO GRÁFICO GENERADO ESTÁTICAMENTE -----	16
ILUSTRACIÓN 15 - PANTALLA DE EXPORTACIÓN DE DATOS -----	16
ILUSTRACIÓN 16 - MODIFICACIÓN DE PROPIEDADES DEL GRÁFICO -----	17
ILUSTRACIÓN 17 - PANTALLA DE BARRIDO SINUSOIDAL EXPONENCIAL -----	17
ILUSTRACIÓN 18 - PANTALLA DE BARRIDO TRIANGULAR CON CAMPOS BLOQUEADOS -----	18
ILUSTRACIÓN 19 - PANTALLA DE BARRIDO MANUAL E IMPORTACIÓN DE DATOS -----	19
ILUSTRACIÓN 20 - ADVERTENCIA AL SELECCIONAR UN FICHERO NO .CSV -----	19
ILUSTRACIÓN 21 - ERROR DE PARÁMETROS DE BARRIDO -----	20



1. Introducción

Dynamic-Route-Maps se presenta como una aplicación que sirve de apoyo para comprender y analizar de manera más eficiente y rápida el comportamiento de los materiales que, por sus características, presentan el fenómeno físico de la conmutación resistiva. Este análisis se ejecuta en base a una caracterización previa del material mediante un barrido creciente o decreciente en tensión (Curva I-V). Así mismo, permite simular el comportamiento del material ante una variación arbitraria del campo eléctrico al que está siendo sometido (tensión fuente).

El presente documento contiene la información necesaria para la correcta utilización del software Dynamic-Route-Maps. Se encuentran especificados los requisitos previos necesarios para su funcionamiento e instalación, así como la interfaz de usuario final, con detalle de cada componente y navegación a través de la misma.

Se incluye del mismo modo, una sección de errores frecuentes enfocada a solucionar los problemas que pudieran surgir durante la ejecución del programa.

2. Software necesario.

Para poder ejecutar correctamente Dynamic-Route-Maps en el computador (de aquí en adelante, DRM), es necesario tener instalado JAVA, con una versión igual o superior a la 1.8. Es posible descargarlo gratuitamente desde la página oficial de [Oracle](https://www.oracle.com/).

No se requiere de ningún otro software específico para poder hacer uso de la aplicación.

Pese a que es posible iniciarlo desde cualquier hardware, es cierto que el elevado número de cálculos que se requieren para la simulación, hacen necesario disponer de un equipo que cuente con una memoria RAM igual o superior a 8Gb.

3. Instalación y configuración de Dynamic-Route-Maps

DRM es un fichero ejecutable de formato “.jar”, por lo que no es necesaria ninguna instalación avanzada del programa para que pueda funcionar en el sistema. Esto permite que el software pueda ser ejecutado libremente en cualquiera de los 3 sistemas operativos más conocidos (Linux, Windows y Mac).

No obstante, para poder abrir la aplicación, sí que es un requisito indispensable que la carpeta “Config” y el fichero ejecutable se encuentren situados en el mismo directorio. En caso contrario, el programa no podrá lanzarse, o bien se lanzará pero no podrá funcionar adecuadamente. Para prevenir futuros errores, es recomendable alojar ambos archivos en una carpeta que no se vaya a modificar, y crear un acceso directo en el escritorio al ejecutable.



4. Detalle de componentes de la aplicación.

En esta sección se explica detalladamente cada uno de los componentes del aplicativo y como usar cada una de las funciones que se encuentran disponibles.

4.1 Elementos comunes.

En toda la aplicación (a salvedad de la pantalla en la que se visualizan los 4 gráficos de la simulación) existen 2 elementos comunes: el menú superior, y el botón salir de la aplicación (Ilustración 1).

El menú superior permite acceder a cada uno de los módulos que conforman DRM de forma directa, independientemente de la pantalla en la que uno se encuentre.

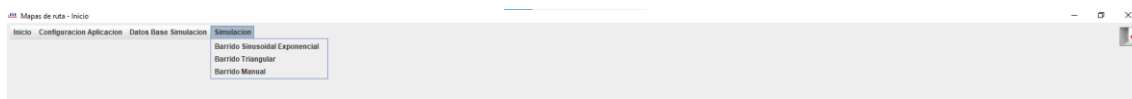


Ilustración 1 - Elementos comunes

Por su parte, el botón salir permite terminar la ejecución en cualquier pantalla, solicitando previamente la confirmación del usuario (Ilustración 2). Si se pulsa en “No” o en el icono del aspa, la aplicación continuará ejecutándose; si se pulsa en “Si”, el programa se cerrará.

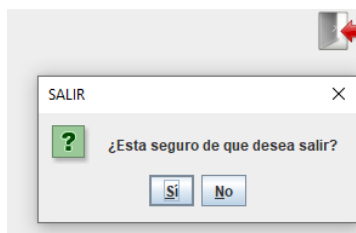


Ilustración 2 - Confirmación para salir de la aplicación

También existe, salvo en la pantalla de inicio y junto al botón salir, un icono de una casa (Ilustración 3) que permite regresar en cualquier momento a la pantalla de inicio.



Ilustración 3 - Icono de inicio

4.2 Pantalla de inicio.

Tras lanzar el ejecutable “Dynamic-Route-Maps.jar” se abrirá una interfaz de usuario mostrando la pantalla de inicio de la aplicación (Ilustración 4). En el contexto central de la pantalla se encuentran 3 botones en forma de iconos que permiten acceder a cada uno de los diferentes módulos de DRM.

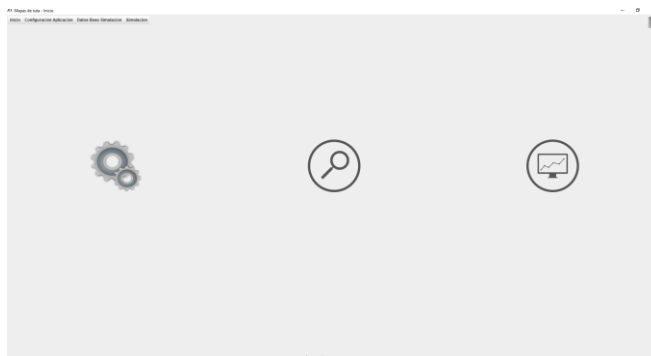


Ilustración 4 - Pantalla de Inicio de DRM

Estos iconos son, de izquierda a derecha: [icono de configuración](#), [icono de visualización de datos base](#), e [icono de simulación](#). Además, si se deja el cursor del ratón colocado encima de cada uno de los iconos, se muestra un tooltip con el nombre del botón en cuestión (Ilustración 5).



Ilustración 5 - Tooltip explicativo del botón.

4.2.1 Icono Configuración.

Tras pulsar en el icono de configuración, se mostrará un mensaje solicitando la pantalla de configuración a la que se desea acceder (Ilustración 6). Las posibles opciones son:

- Configuración General.
- Configuración Barrido Sinusoidal Exponencial.
- Configuración Barrido Triangular.

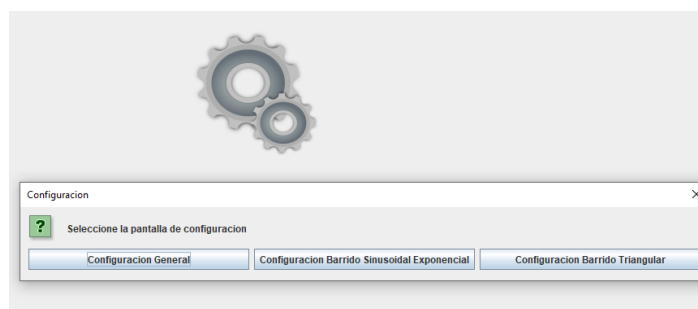


Ilustración 6 - Elección de pantalla de configuración



ANEXO I



Tras elegir una de las 3 opciones, el sistema redirigirá a la pantalla correspondiente para poder configurar los parámetros deseados.

4.2.2 Icono Datos Base.

Al elegir este icono se mostrará la pantalla que permite visualizar gráficamente los datos que se han tomado como base para la simulación.

4.2.3 Icono Simulación.

Tras hacer click en este icono, se mostrará un mensaje similar al que se muestra al elegir el botón "Configuración", pero esta vez solicitando el tipo de simulación que se pretende probar (Ilustración 7). Las opciones posibles son:

- Barrido Sinusoidal Exponencial
- Barrido Triangular
- Barrido Manual



Ilustración 7 - Elección de pantalla de Simulación



4.3 Pantallas de Configuración.

Tras haber accedido a alguna de las pantallas de configuración, bien desde el icono de configuración, o bien desde el menú superior, se presenta una pantalla con una tabla de datos con 2 columnas en la zona superior, y un botón de modificar en la parte inferior.

La columna “PROPIEDAD” hace referencia al nombre de la propiedad que se pretende modificar, y la columna “VALOR” al valor que se le pretende dar a dicha propiedad. No es posible modificar la descripción de la propiedad, pero sí su valor de acuerdo a las necesidades del usuario.

Los campos numéricos admiten valores decimales tanto con formato punto como con formato coma. No obstante, si se intenta asignar un valor no numérico a un campo numérico el sistema avisará con un error (Ilustración 8) y no permitirá esta modificación.

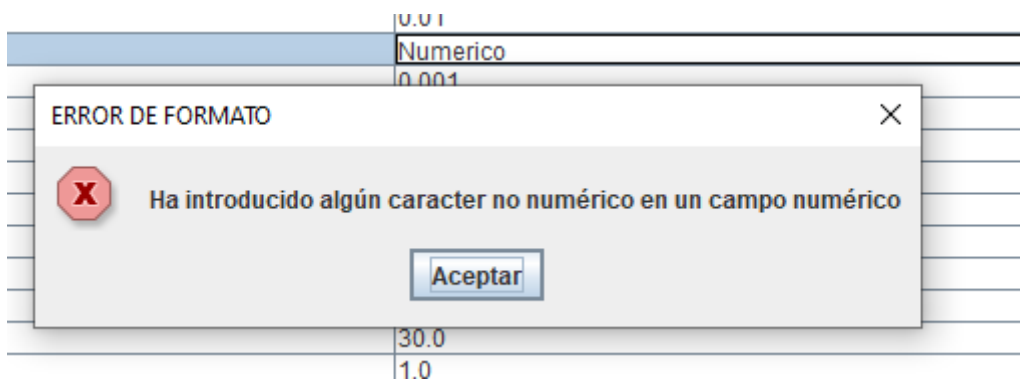


Ilustración 8 - Error carácter no numérico

4.3.1 Pantalla de Configuración General

Permite configurar algunos parámetros globales de la aplicación (Ilustración 9).

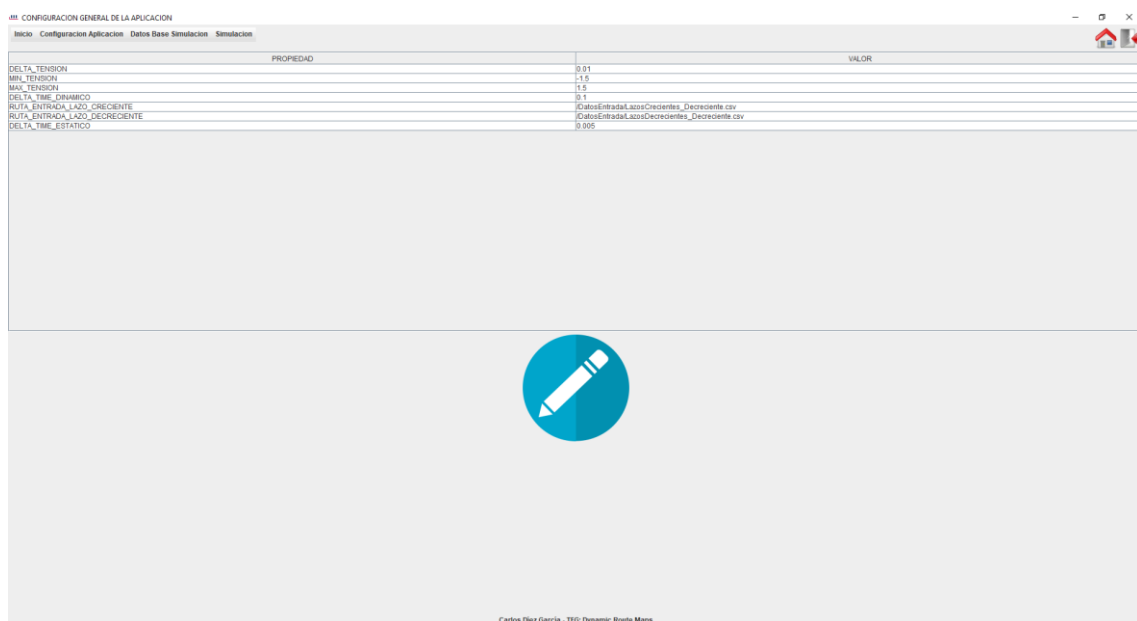


Ilustración 9 - Pantalla de Configuración General



ANEXO I



En detalle, los parámetros que se pueden configurar son los siguientes:

- **DELTA TENSION**: regula el paso de tensiones que se emplea para interpolar cada curva. Valores muy bajos darán mejores resultados de las curvas simuladas, pero se ralentizará la ejecución. Por el contrario, valores más altos agilizarán la ejecución del programa pero los resultados serán pobres. El valor recomendado es de 0.01.
- **MAX TENSION**: regula la tensión máxima que se dibujará en los resultados. No regula la tensión máxima a aplicar, sino la máxima que se pintará. El valor recomendado es el valor máximo que tenga la tensión en los datos que se empleen de base.
- **MIN TENSION**: regula la tensión mínima que se dibujará en los resultados. No regula la tensión mínima a aplicar, sino la más baja que se pintará. El valor recomendado es el valor mínimo que tenga la tensión en los datos que se empleen de base.
- **DELTA TIME DINAMICO**: regula el paso temporal entre punto y punto de los datos que se simulan, cuando la simulación se realiza en tiempo real. Como los datos van cambiando a medida que se modifican los parámetros, no se recomienda que este valor sea inferior a 0.1 a menos que se disponga de un hardware potente (superior a 16gb de RAM), ya que se estresará el software e incluso podría llegar a congelar la pantalla.
- **DELTA TIME ESTATICO**: regula el paso temporal entre punto y punto de los datos que se simulan, cuando la simulación no se realiza en tiempo real. Esta configuración se aplica para los botones de la pantalla de simulación que generan las gráficas. El valor recomendado es de 0,1 o 0,01 veces el valor de la propiedad DELTA_TIME_DINAMICO.
- **RUTA ENTRADA LAZO CRECIENTE**: indica la ruta donde se encuentra el fichero del que se extraerán los datos que servirán de base para todas las simulaciones, cuando las tensiones sean crecientes. Este fichero debe tener las siguientes características para que DRM funcione como se espera:
 - ✓ Debe ser un fichero de extensión “.csv” cuyos datos estén separados por el delimitador “;”.
 - ✓ Debe tener una columna “SOURCE” en la que los datos sean las tensiones del lazo, y una columna “CURRENT” en la que los datos sean las corrientes del lazo.
 - ✓ Todas las tensiones deben ser crecientes (es decir, $\Delta V > 0 \forall t$), por lo que es necesario haber preparado previamente los datos de entrada.

Si la ruta indicada no existe, al intentar visualizar los [datos base](#) que cargan estos ficheros, se muestra un error avisando de una mala lectura del csv, y el sistema redirige a la pantalla de configuración general para que se revisen las rutas de entrada (Ilustración 11).



ANEXO I



Para cada tipo de material del que se dispongan datos, se debe actualizar el valor de estos parámetros con el directorio (ruta absoluta, o relativa si se sitúan dentro de la carpeta “Config”) en el que se encuentren los nuevos ficheros (ya que cada material tendrá unas curvas base diferentes).

- RUTA ENTRADA LAZO DECRECIENTE: la utilidad es la misma que la propiedad anterior, pero cuando las tensiones sean decrecientes. Tiene las mismas restricciones que RUTA_ENTRADA_LAZO_DECRECIENTE, salvo que en este caso todas las tensiones deben ser decrecientes..

4.3.2 Pantalla Configuración Parámetros Barrido Sinusoidal Exponencial

Permite configurar los parámetros que afectan al barrido particular en el que la tensión de entrada adopta la forma:

$$V(t) = a \operatorname{Sin}\left(\frac{2\pi}{T} t\right) e^{bt}$$

Ecuación 1 – Función sinusoidal exponencial

El significado de cada parámetro es:

- VALOR MINIMO PARAMETRO A: especifica el mínimo valor que puede tomar “a” en (Ecuación 1).
- VALOR MAXIMO PARAMETRO A: especifica el máximo valor que puede tomar “a” en (Ecuación 1);
- PASO PARAMETRO A: indica la diferencia entre 2 valores consecutivos del parámetro “a”.
- VALOR MINIMO PARAMETRO B: especifica el mínimo valor que puede tomar “b” en (Ecuación 1).
- VALOR MAXIMO PARAMETRO B: especifica el máximo valor que puede tomar “b” en (Ecuación 1);
- PASO PARAMETRO B: indica la diferencia entre 2 valores consecutivos del parámetro “b”.
- VALOR MINIMO PARAMETRO T: especifica el mínimo valor que puede tomar “T” en (Ecuación 1).



- VALOR MAXIMO PARAMETRO T: especifica el máximo valor que puede tomar “T” en (Ecuación 1);
- PASO PARAMETRO T: indica la diferencia entre 2 valores consecutivos del parámetro “T”.
- VALOR MINIMO PARAMETRO TF: especifica el valor mínimo para el tiempo máximo que estará aplicándose la tensión en la simulación.
- VALOR MAXIMO PARAMETRO TF: especifica el valor máximo para el tiempo máximo que estará aplicándose la tensión en la simulación.
- PASO PARAMETRO TF: indica la diferencia entre 2 valores consecutivos del parámetro “TF”.

4.3.3 Pantalla Configuración Parámetros Barrido Triangular

Permite configurar los parámetros que afectan al barrido cuando la función de entrada adopta la forma de una función triangular, es decir:

$$V(n, t) = 2(V_0 + n\Delta V_p) \left(\frac{1}{T}t - n \right) \quad \forall t \in \left(Tn, Tn + \frac{T}{2} \right)$$
$$V(n, t) = 2(V_0 + n\Delta V_p) \left(n + 1 - \frac{1}{T}t \right) \quad \forall t \in \left(Tn + \frac{T}{2}, T(n + 1) \right)$$

Ecuación 2 - Función Triangular

Las distintas variables configurables de este barrido son:

- NUMERO_MAXIMO_TRIANGULOS: especifica el valor máximo que puede alcanzar n en Ecuación 2.
- NUMERO_MINIMO_TRIANGULOS: idéntico al anterior, pero con el valor mínimo.
- PASO_TRIANGULOS: indica el número de triángulos en que aumentará la variable n cada vez que se intente aumentar este valor.



ANEXO I



- **TENSION_MAXIMA_PRIMER_PICO:** especifica el valor máximo puede tomar la variable V_0 en Ecuación 2, que se corresponde con la tensión máxima del primer triángulo en la $V(t)$.
- **TENSION_MINIMA_PRIMER_PICO:** similar al anterior, pero con el valor mínimo.
- **PASO_TENSION_PRIMER_PICO:** representa lo que aumenta o disminuye la tensión cada vez que se modifica este parámetro.
- **VALOR_MAXIMO_PERIODO:** indica el máximo valor posible para el parámetro T en Ecuación 2, y da cuenta del tiempo que tarda en completarse un triángulo completo.
- **VALOR_MINIMO_PERIODO:** igual que el anterior, pero para el valor mínimo.
- **PASO_PERIODO:** establece lo que se incrementa o disminuye el parámetro T cada vez que se modifica el mismo.
- **DELTA_TENSION_PICOS_MAXIMA:** especifica el máximo valor que puede tomar ΔV_0 en la Ecuación 2, es decir, lo que se incrementa el pico de la tensión cada vez que aparece un nuevo triángulo.
- **DELTA_TENSION_PICOS_MINIMA:** idéntico al anterior, pero con el valor mínimo posible.
- **PASO_TENSION_PICOS_MAXIMA:** representa lo que aumenta o disminuye ΔV_p cada vez que sube o baja este valor.



4.4 Pantalla de Datos Base

Esta pantalla muestra al usuario los datos que se están utilizando de base para la simulación, es decir, los datos que se toman de los ficheros que se hayan configurado desde la configuración general en los parámetros RUTA_ENTRADA_LAZO_CRECIENTE y RUTA_ENTRADA_LAZO_DECRECIENTE.

Estos ficheros de entrada, deben ser ficheros de formato “.csv”, cuyos campos sean “SOURCE” y “CURRENT” (para albergar respectivamente los datos de tensión y de corriente medidos directamente en la muestra que se intente caracterizar), y los valores deberán estar separados por el delimitador ‘;’ y sin comillas (Ilustración 10).

```
CH1 Current;CH1 Time;CH1 Source
0.00001222;0.1005;0.01315405
0.00004119;0.2005;0.04551774
0.00007226;0.3005;0.07772953
0.00010569;0.4005;0.10976746
0.00014298;0.5005;0.14160969
0.00018129;0.6005;0.17323453
0.00022494;0.7005;0.20462046
0.00027775;0.8005;0.23574612
0.00034189;0.9005;0.26659034
```

Ilustración 10 - Formato correcto de datos de entrada en ficheros .csv

Esto es especialmente importante por 2 motivos:

- I. Si las columnas del .csv no se llaman “SOURCE” y “CURRENT”, los datos que se cargaran para la simulación no van a ser, en general, correctos.
- II. Si los valores no se encuentran en el formato indicado, bien por presentar comillas, o bien por no estar separados por ‘;’, se muestra un error (Ilustración 11) advirtiendo que existe un problema en los datos de entrada, y el sistema redirige a la pantalla de configuración general para que se revise la ruta de los ficheros de entrada.

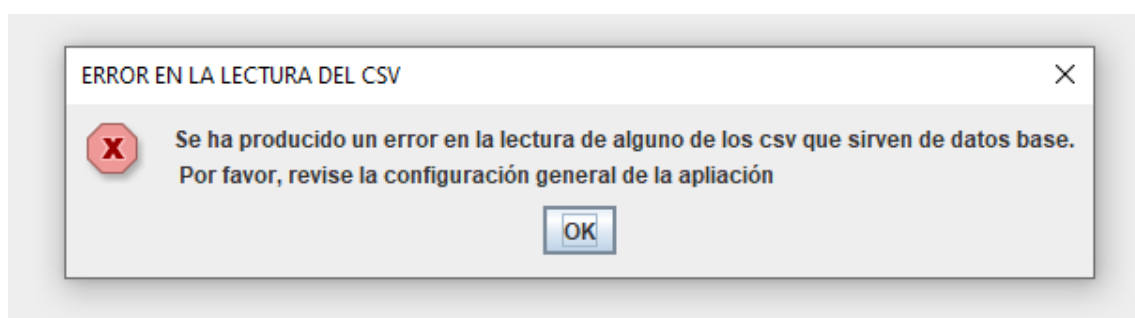


Ilustración 11 - Error al leer los csv de datos base

Si la lectura de los ficheros base ha sido correcta, al acceder a la pantalla de datos base (bien desde el menú superior, o bien desde el icono en la pantalla de inicio), se muestra una pantalla en la que es posible visualizar los 4 gráficos principales objeto de estudio.

Estas cuatro gráficas dependen del material que se pretenda analizar, pero deben haberse obtenido realizando un barrido sinusoidal creciente (o decreciente) en



ANEXO I



tensiones a través del tiempo sobre dicho material, anotando las tensiones de salida en el generador y las corrientes inferidas en el material.

El aspecto de esta pantalla es el siguiente (Ilustración 12):

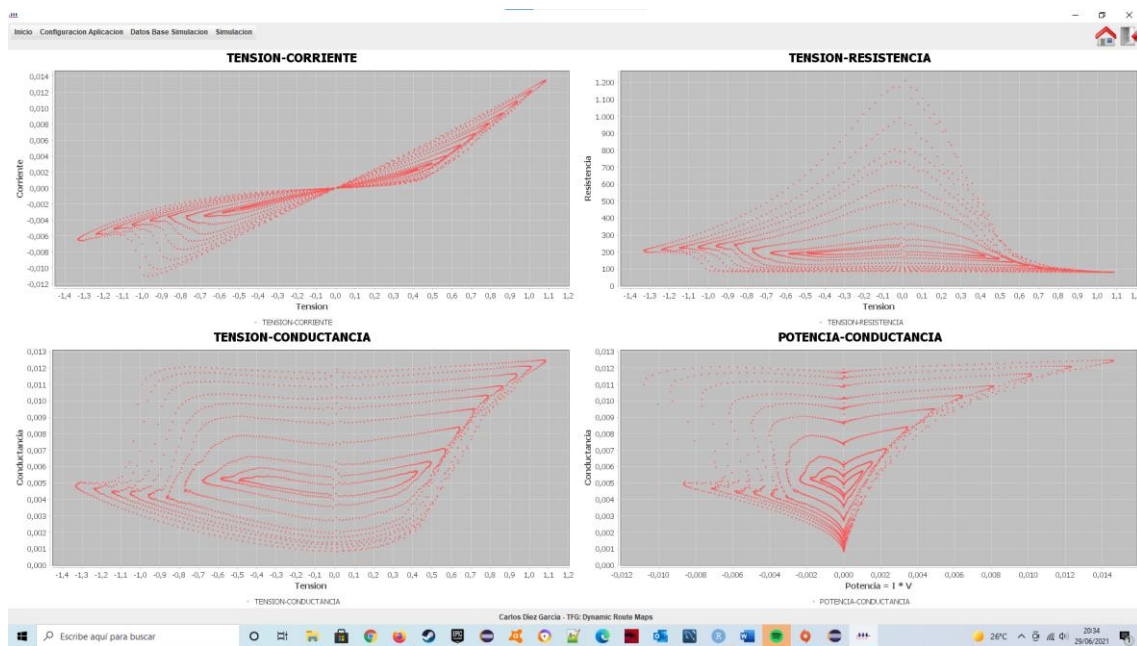


Ilustración 12 - Pantalla de datos Base

Esto es de gran utilidad, ya que permite disponer de un golpe de vista el comportamiento que ha tenido el material durante la exposición al barrido, e incluso a otros efectos, puede emplearse para visualizar, a partir de los datos de tensión corriente, las 4 gráficas más características del comportamiento del material si el pretexto es meramente con carácter de visualización:

- Curva I(V) arriba a la izquierda.
- Curva R(V) arriba a la derecha.
- Curva G(V) abajo a la izquierda.
- Curva G(P) abajo a la derecha.

4.5 Pantalla de Simulación

Estas pantallas representan el núcleo de DRM y permiten obtener, en base a los datos iniciales introducidos, las corrientes generadas en el material a partir de una variación de tensión $V(t)$, en principio, arbitraria. Son accesibles a través del icono de [Simulación de la pantalla de inicio](#), y a [partir del menú superior](#).

DRM permite establecer simulaciones de formas de $V(t)$ regulares predefinidas como las indicadas en Ecuación 1 y Ecuación 2, y también arbitrarias mediante la importación de un fichero .csv de datos externos.

Como puntos comunes, las 3 pantallas tienen la siguiente información:

- I. A la izquierda de la pantalla, se establecen los valores de los parámetros de la función $V(t)$, si procede. Estos valores pueden modificarse a través de sliders de control (para su manipulación rápida) y de spinners (cuando se pretende ser más preciso). También es posible introducir manualmente un valor mediante el teclado en el espacio reservado para el valor, siempre que se encuentre dentro del. De igual manera, los valores máximos y mínimos de estos parámetros pueden editarse desde la configuración de cada pantalla según lo indicado en el apartado [“Pantallas de Configuración”](#).

A medida que se realicen cambios se modificarán dinámicamente los gráficos generados (Ilustración 13) en la parte derecha, representando datos cada X tiempo, siendo X el valor de la propiedad “DELTA_TIME_DINAMICO” de la configuración general. Es por eso que dicho valor debe ser, por lo general, no muy preciso.

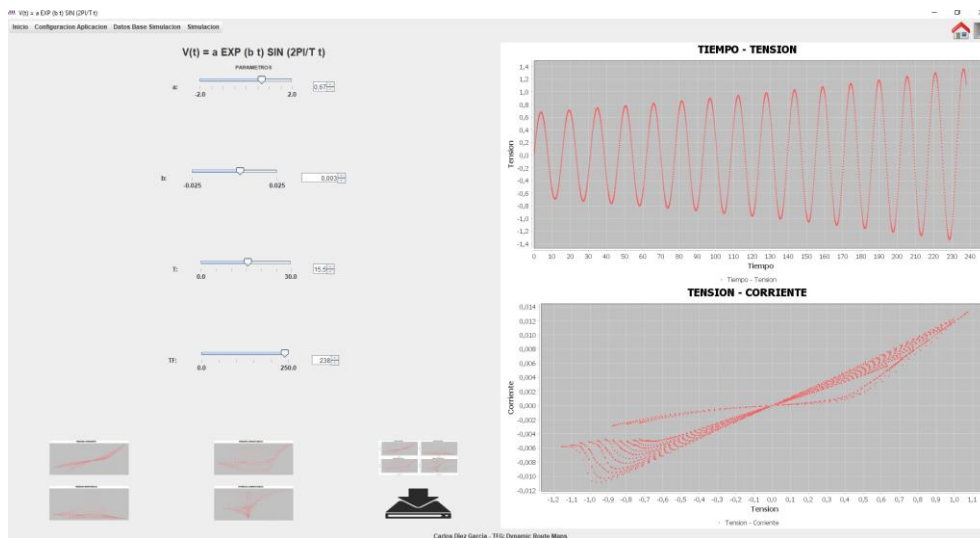


Ilustración 13 - Gráfico generado dinámicamente por manipulación de parámetros

Existen una serie de iconos (botones) cada uno con el dibujo de la gráfica correspondiente a simular. Tras pulsarlos, se refrescarán los gráficos ya pintados según se hayan establecido los parámetros, pero con el paso temporal indicado en “DELTA_TIME_ESTATICO”, por lo que proporcionan información más detallada y precisa del comportamiento (Ilustración 14). El proceso de



ANEXO I



generación de los gráficos puede demorarse más o menos tiempo en función de la potencia computacional del hardware empleado.

Si se deja el cursor del ratón situado sobre cada icono, aparece un tooltip indicando el tipo de gráfica que generará.

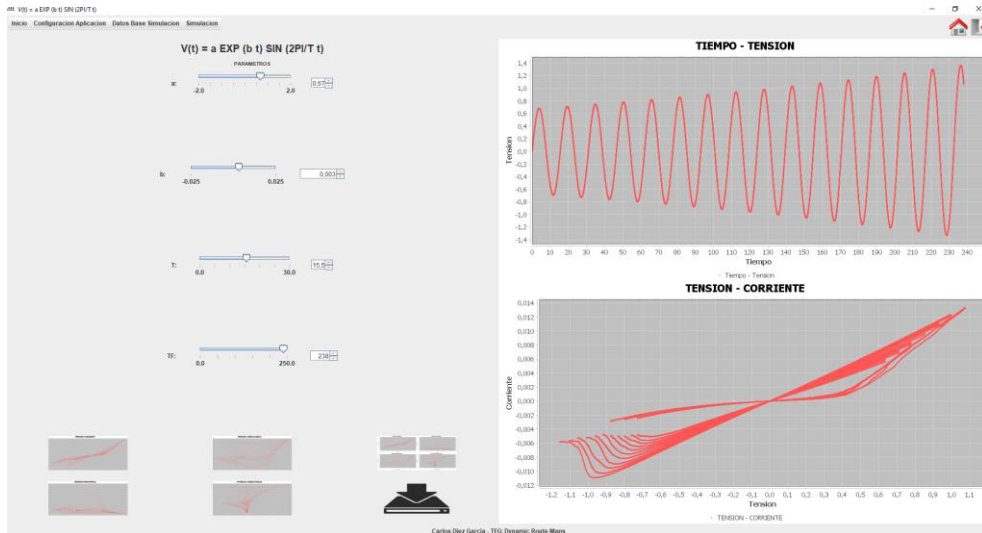


Ilustración 14 - Mismos parámetros pero gráfico generado estáticamente

A mayores, existen otros 2 botones: uno con un icono de 4 gráficos que permite visualizar simultáneamente todas las gráficas en una nueva ventana similar a la visualización de los datos base, y otro que permite exportar (Ilustración 15) todos los datos en formato .csv separado por ';', que está preparado para ser importado, por ejemplo, a hojas de cálculo. Se exportarán los datos de tiempo, tensión, corriente simulada, resistencia simulada, conductancia simulada y potencia simulada.

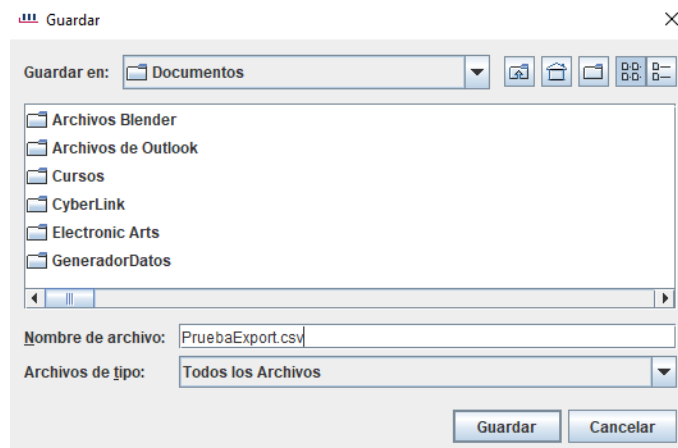


Ilustración 15 - Pantalla de exportación de datos



ANEXO I



- II. A la derecha de la pantalla existe una zona inicialmente vacía, en la que se muestran los gráficos cuando se hayan establecido los valores de los parámetros.

Es posible modificar algunas propiedades del gráfico (Ilustración 16), como la escala, ejes, etc., haciendo click derecho sobre el mismo. De igual manera es posible copiar el gráfico como imagen, guardarlo en formato .png e imprimirlo. Además, se permite hacer zoom sobre una zona concreta haciendo arrastrando el ratón con el botón izquierdo pulsado.

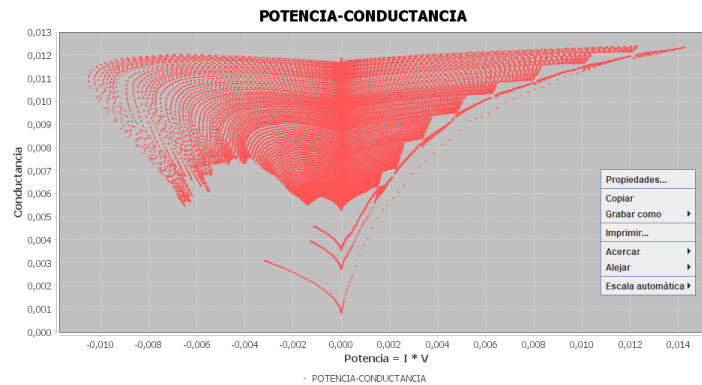


Ilustración 16 - Modificación de propiedades del gráfico

A continuación, se detallan las particularidades de cada una de las pantallas.

4.5.1 Pantalla de Barrido Sinusoidal Exponencial

Esta pantalla (Ilustración 17) no presenta particularidades respecto a lo indicado en el punto [4.5 Pantalla de Simulación](#).

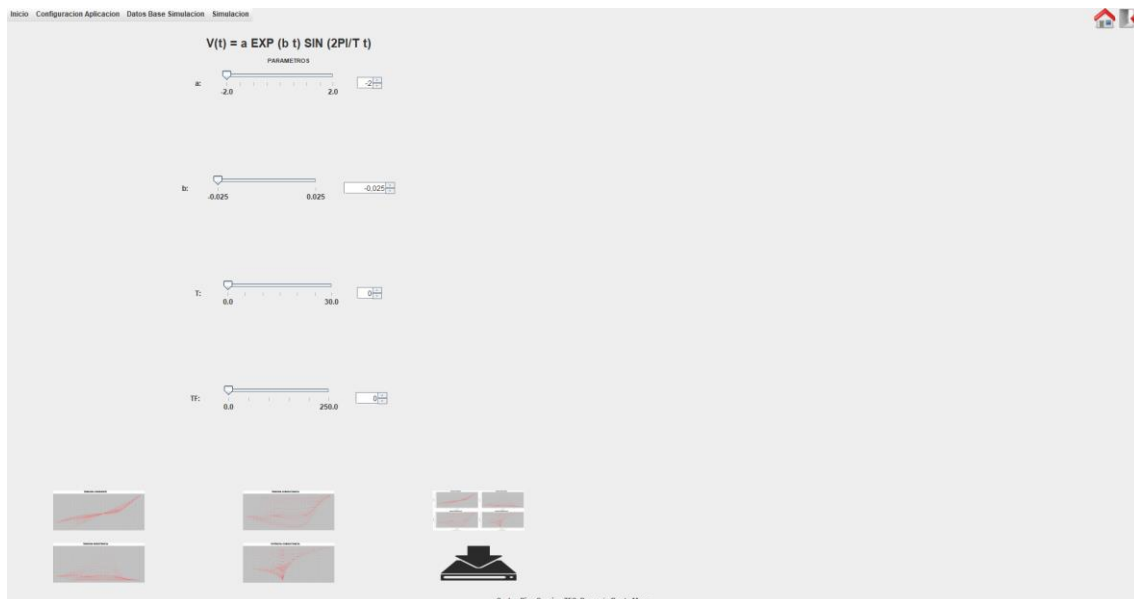


Ilustración 17 - Pantalla de Barrido Sinusoidal Exponencial

4.5.2 Pantalla de Barrido Triangular

Al acceder por primera vez al barrido triangular (Ilustración 18), tendremos todos los parámetros bloqueados salvo el periodo. Tras aumentar este valor a una cantidad superior a la unidad, se produce el desbloqueo del resto de parámetros y botones para poder configurar el barrido. Si, tras estar configurado, se vuelve a modificar el valor del periodo a 0, entonces se volverán a bloquear el resto de parámetros y botones, y los gráficos que hubiese generados en la parte derecha de la pantalla desaparecerán.

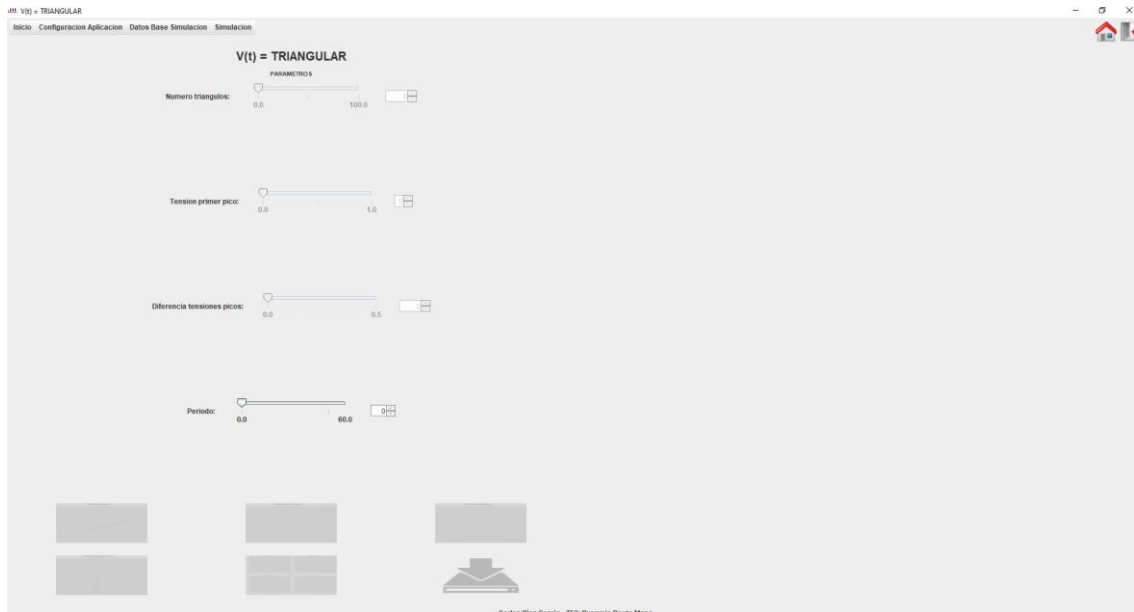


Ilustración 18 - Pantalla de barrido Triangular con Campos Bloqueados

4.5.3 Pantalla de Barrido Manual

La pantalla de barrido manual difiere de las anteriores en que aquí no es posible configurar ningún parámetro, ya que los datos para los que se simulan las curvas son directamente importados de un fichero .csv separado por ';', con 2 columnas: una para el tiempo, y otra para la tensión.

Tras acceder a la pantalla, se debe pulsar en el único icono que estará activo para importar los datos. Una vez elegido el fichero, si no ha habido ningún error se habilitarán el resto de botones que son similares a los de las pantallas anteriores.



ANEXO I

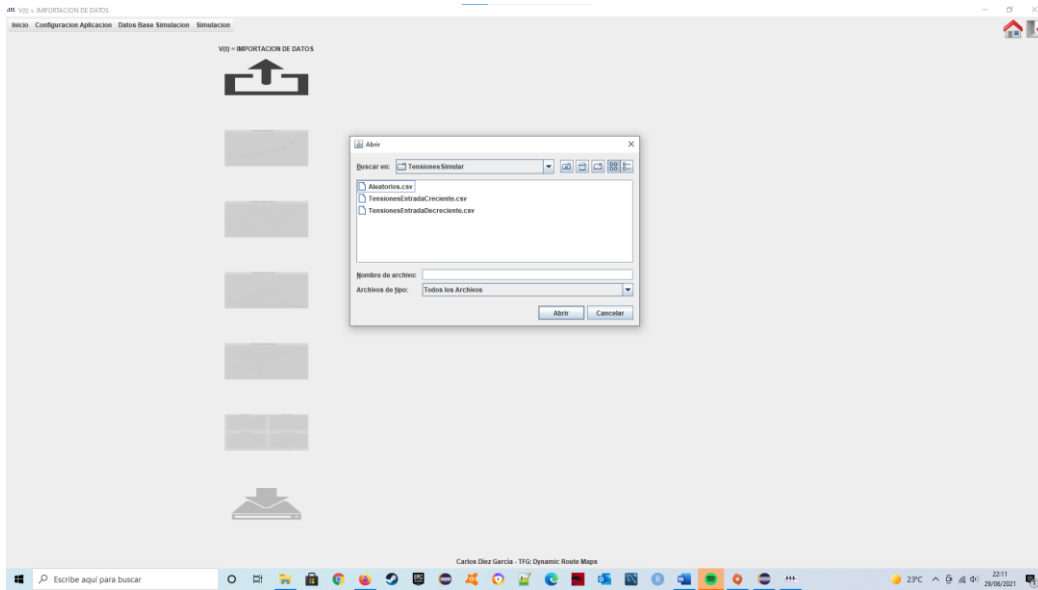


Ilustración 19 - Pantalla de Barrido Manual e importación de datos

Si no se selecciona un fichero .csv, o bien no se selecciona ningún fichero, aparecerá un mensaje advirtiéndolo de la situación y los botones permanecerán deshabilitados. Del mismo modo, si estuvieran desbloqueados, pero se pulsa en importar, será necesario volver a elegir el fichero ya que los iconos no se podrán pulsar (Ilustración 20).

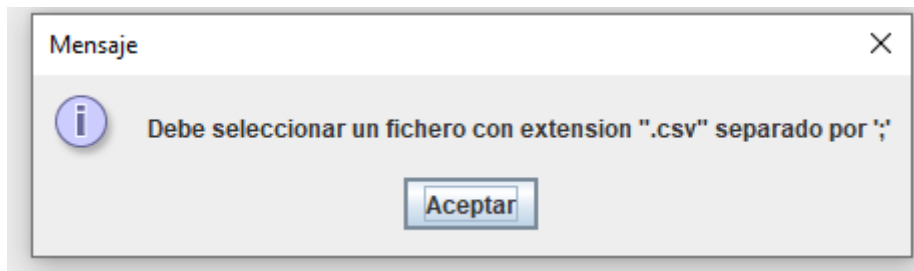


Ilustración 20 - Advertencia al seleccionar un fichero no .csv

5. Errores frecuentes

Se listan a continuación una serie de errores que pueden darse a lo largo de la utilización de DRM.

- **La aplicación no se inicia, o se inicia, pero no es posible realizar ninguna acción.**

Es posible que por accidente se haya movido el directorio “Config” que debe acompañar al fichero ejecutable a otro lugar. Simplemente, termine la ejecución del programa si es que se inició, y vuelva a colocar ambos ficheros en el mismo directorio raíz. Para evitar este problema, se recomienda crear un acceso directo del ejecutable en el escritorio (lo que permite además cambiar el icono del mismo), y dejar situados el fichero Config y el ejecutable real en otra ruta que no se vaya a modificar.

- **Se pulsa en modificar pero no se guardan los cambios del último valor modificado en las tablas de configuración.**

Esto se debe a que es necesario pulsar en algún otro campo cuando se ha editado el último valor, o bien pulsar la tecla “enter” al terminar de editarlo.

- **No se visualizan los datos base que se han introducido o se visualizan incorrectamente.**

Esto puede deberse a dos motivos:

- I. Los datos de tensión y corriente no están en las columnas “Source” y “Current” respectivamente.
- II. El formato del csv presenta los datos entrecomillados, o no están separados por “;”.

- **Aparece el error “Alguno de los parámetros de barrido no está configurado correctamente (Ilustración 21)”.**

Es posible que en alguna modificación se haya introducido el valor mínimo de un parámetro que es superior al máximo que se definió para el mismo parámetro, o que la modificación de algún “PASO” no permita configurar correctamente los sliders y spinners. En este caso, el sistema advierte de este error y redirige a la pantalla de configuración correspondiente para solucionarlo.

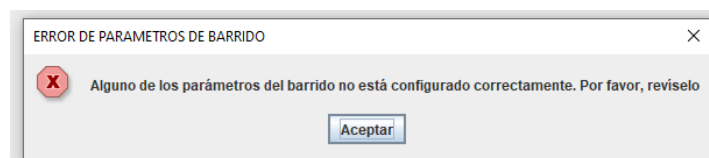


Ilustración 21 - Error de parámetros de barrido

- **Los sliders y spinners no se comportan debidamente**

Es posible que ciertas configuraciones de valores máximos, mínimos y pasos de los parámetros procuren situaciones en las que estos deslizadores no se



ANEXO I



comporten correctamente. Esto sucede comúnmente en los parámetros que adoptan valores enteros, como el período en la función triangular, por tema de redondeos. Para evitar este problema, es mejor no modificar el valor de los parámetros “PASO”. Así mismo, la modificación del paso de algunos parámetros puede hacer manifestarse el error anterior.

- **No se genera ningún gráfico.**

Si los datos que se emplean como base para la simulación son datos de barridos decrecientes que se detienen antes de llegar a tensión 0, es posible que algunas combinaciones de parámetros hagan que no se alcance nunca esa tensión mínima para la que existen datos. Del mismo modo sucede para tensiones altas, ya que el sistema no es capaz de simular tensiones superiores a 1.5V ni inferiores a -1.5V (parámetros configurables desde la configuración general). Adicionalmente, puesto que se ha considerado que el estado base de la muestra para todas las simulaciones es puro (no ha sufrido ningún set ni reset), no se generarán curvas hasta que se haya alcanzado la tensión mínima en directa para la que existen datos base.

- **El sistema no responde tras intentar generar un gráfico mediante alguno de los botones.**

Si tratamos con conjuntos de datos importados muy pesados, o bien si hemos establecido el DELTA_TIME_ESTATICO a un valor muy pequeño, es muy probable que el sistema tarde tiempo en ejecutar todos los cálculos de la simulación. La solución a este problema es o bien dejar trabajar al computador, o bien cerrar el programa con el botón del aspa, ya que el icono habitual para salir de la aplicación estará bloqueado hasta que finalicen las operaciones internas.

- **El sistema no responde ante ninguna acción.**

Es posible, aunque improbable, que el sistema se quede congelado por causa de algún error interno de la aplicación no detectado, ya que el software no ha pasado por la pertinente fase de pruebas que requiere todo programa computacional. En ese caso, la forma de proceder es intentar matar el proceso (“Ctrl + Alt + Sup” en Windows, “CMD+Alt+Esc” en Mac, o accediendo al monitor del sistema en Linux).



6. Consideraciones finales

Este manual está diseñado para comprender el funcionamiento a alto nivel y poder navegar cómodamente a través de la interfaz de DRM, entendiendo como configurar cada uno de los parámetros y aprendiendo a solucionar las diferentes situaciones no deseadas que pudieran surgir de su utilización. No pretende ser un documento que explique cada uno de los algoritmos utilizados en el cálculo de la simulación, sino simplemente una guía a nivel de usuario que permita su adecuado manejo.