



UNIVERSIDAD NACIONAL  
DE LA PLATA

Facultad de Ingeniería  
Departamento de Electrotecnia

# ADQUISICIÓN DE DATOS - TEKTRONIX TDS360

---

DISEÑO DE APLICACIÓN EN MATLAB R2023B

CHRISTIAN YOEL HERRERA

11 DE MARZO DE 2025



# Índice general

<b>1. Introducción</b>	<b>5</b>
1.1. Instalación del Driver Prolific . . . . .	5
1.2. Requisitos de la Aplicación . . . . .	6
1.3. Comunicación Serial . . . . .	6
<b>2. Programación en MatLab</b>	<b>9</b>
2.1. Funciones Implementadas . . . . .	9
2.2. Configuración en el Osciloscopio . . . . .	10
<b>3. Manual de la Aplicacion</b>	<b>11</b>
3.1. Conexión USB . . . . .	11
3.1.1. Ajustes de Tiempo . . . . .	12
3.1.2. Ajustes de los Canales . . . . .	12
3.2. Exportar en Archivo .MAT . . . . .	13
3.3. Exportar como .CSV . . . . .	13
3.4. Importar archivo .MAT . . . . .	13
3.5. Exportar Imágenes . . . . .	14
3.5.1. Exportar como Archivo de Imagen . . . . .	14
3.5.2. Exportar al Portapales . . . . .	14



# Capítulo 1

## Introducción

El osciloscopio de la marca Tektronix, específicamente el modelo TDS 360 permite una comunicación serial utilizando el puerto RS232. Actualmente existen adaptadores de RS232 a USB con lo cual se diseñó una aplicación haciendo uso del Software App Designer (de la suite de MatLab) que logre iniciar la comunicación, configurar el osciloscopio y recibir el vector de valores correspondientes a las muestras almacenadas en el registro del Osciloscopio.

### 1.1. Instalación del Driver Prolific

Dado que Windows instala una versión del driver que no permite hacer uso del adaptador (ya que reconoce que el chip no es original), se deberá instalar el Driver adecuado. Para eso, en la sección de **Administrador de Dispositivos**, se debe desinstalar completamente el que instala Windows:

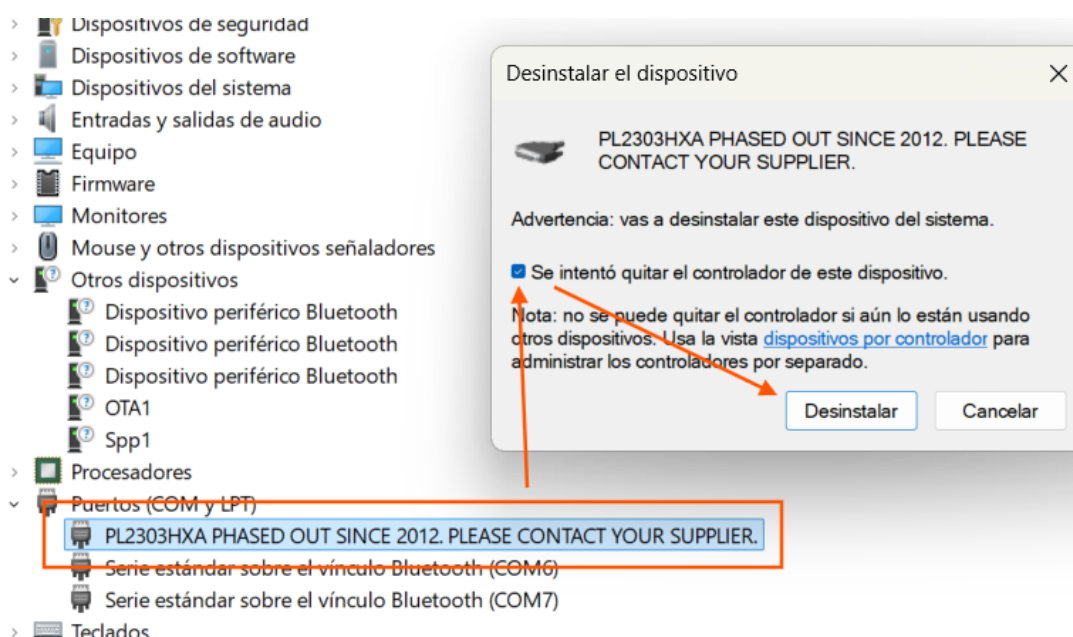


Imagen 1.1: Adm. de Dispositivos - Eliminar Driver

Luego de este paso, sin desconectar el adaptador, se procede a instalar el nuevo driver usando el instalador adjunto al instructivo.

En este punto, si la instalación fue correcta, entonces se utiliza la opción del menú: **Buscar cambios de Hardware**. En la lista deberá aparecer el nombre:

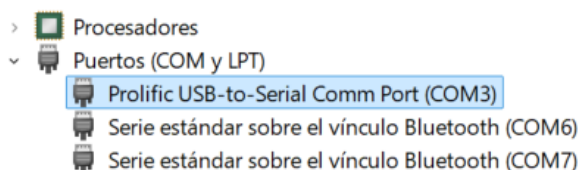


Imagen 1.2: Adm. de Dispositivos - Driver Prolific

Si luego de refrescar la lista, sigue apareciendo el driver anterior, se podrá forzar la actualización de la siguiente forma:

1. Seleccionar el elemento de la lista y en las opciones desplegadas elegir *Actualizar controlador*.
2. Seleccionar *Examinar mi PC en busca de controladores*.
3. Seleccionar *Elegir en una lista de controladores disponibles en el equipo*.
4. Y de la lista, seleccionar **Prolific USB-to-Serial Comm Port**

**Nota:** La instalación automática del driver por parte de Windows se logra porque Windows Update está habilitado. En caso de requerir desactivar esta funcionalidad momentaneamente se puede usar el ejecutable Windows Update Blocker v1.8)

## 1.2. Requisitos de la Aplicación

La primera versión que se logro realizar, tiene el siguiente aspecto.

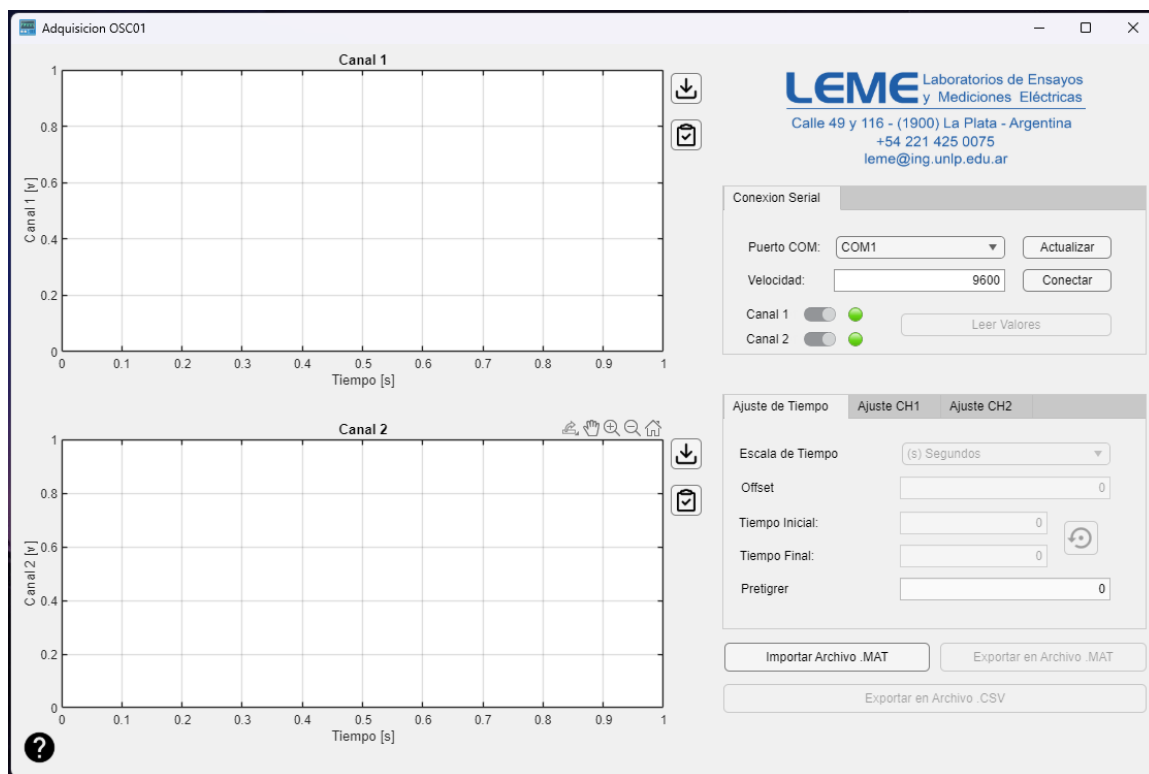


Imagen 1.3: Versión preliminar - Diseño

El diseño es sencillo, se intenta que disponga de las siguientes funcionalidades:

- Que permita configurar los valores esenciales de la comunicación serial, por ejemplo, el puerto **COM** y los **Bits por segundo**.
- Que verifique la correcta comunicación con el Osciloscopio.
- Que luego de solicitar los valores, se disponga de casilleros donde se pueda ajustar los limites de tiempo.
- Que los limites del eje vertical de cada canal se pueda modificar.
- Que se permita modificar los títulos y las leyendas de cada gráfica.
- Que cada canal, disponga de un casillero donde se pueda ingresar un factor, con el fin de aplicar tal valor a cada valor del eje vertical.

## 1.3. Comunicación Serial

En primer lugar, al usar un conversor de RS232 a USB, se procedió a enviar los primeros comandos usando el **Monitor Serial** del Arduino IDE (Disponible en el [Link](#)). Los comandos que se usan son los que se enumeran a continuacion. En aquellos donde se especifique **CHn**, podra valer **CH1** o **CH2**.

- **DATA:ENCDG ASCII** : Configura la codificación de los valores que recibo cuando solicito las muestras almacenadas. Esta codificación podrá ser: **ASCII** , **Rlbynary**
- **HEADER OFF** : Permite que las respuestas no contengan el identificador del comando, de esta forma la respuesta sera unicamente los valores que solicitemos.
- **DATA:WIDTH 2** : Al configurar en **2** , cuando se soliciten las muestras tomadas, el osciloscopio nos devolverá los valores con usando 2 bytes.
- **DATA:SOURCE CHn** : Permite seleccionar un canal especifico, para luego descargar los valores de este canal.
- **HORIZONTAL:Main?** : El osciloscopio devolverá el valor de los  $s/div$
- **CHn:SCALE?** : Permite obtener el valor de los  $v/div$ . La elección del canal se hace con **CH1** o **CH2** .
- **WFMPRE:CHn:XINCR?** : Permite obtener el valor del tiempo entre dos muestras consecutivas.
- **WFMPRE:CHn:YMULT?** : Permite obtener el valor por el cual se debe multiplicar cada muestra.
- **WFMPRE:CHn:YZERO?** : Devuelve el valor del offset vertical.
- **WFMPRE:CHn:YOFF?** : Devuelve el valor del offset de voltage.
- **CURVE?** : Al enviar este comando, el osciloscopio devolvera los puntos guardados en su registro, en el formato especificado por **DATA:ENCDG** .
- **HORIZONTAL:TRIGGER:POSITION?** : Es el valor, en porcentaje, de tiempo con respecto a la ventana total de tiempo que se tiene como pre-triger.

El procedimiento para obtener el vector del canal 1 es el siguiente:

```

1 write("DATA:ENCDG RIBINARY");
2 write("HEADER OFF");
3 write("DATA:WIDTH 2");
4
5 write("DATA:SOURCE CH1");
6 write("CURVE?");
7
8 xincr = writeread("WFMPRE:CH1:XINCR?");
9 ymult = writeread("WFMPRE:CH1:YMULT?");
10 yzero = writeread("WFMPRE:CH1:YZERO?");
11 yoff = writeread("WFMPRE:CH1:YOFF?");
12
13 adc_raw = readbinblock();

```

Finalmente, una vez que se logre recibir el vector de puntos, y los demás valores se procede con la conversión de unidades, con el objetivo de armar los vectores de tiempo y tensión.

$$volts = (adc\_raw - y_{off}) \cdot y_{mult} + y_{zero} \quad (1.1)$$

$$t = 0 : x_{incr} : (length(volts) - 1) \cdot x_{incr} \quad (1.2)$$

La ecuación 1.2 hace uso de la sintaxis de Matlab, logrando formar un vector desde cero hasta la ultima muestra leida, con un tiempo entre muestra de  $x_{incr}$ .

Estos comandos fueron los necesarios para descargar todos los valores y datos del canal 1, para hacer uso del canal 2, se repite desde la linea 4 (`write("DATA:SOURCE CH1");`) haciendo uso del CH2.





## Capítulo 2

# Programación en MatLab

### 2.1. Funciones Implementadas

La primer versión usando App Designer, de la suite de MatLab, incorpora funciones que hacen posible el intercambio de información. A continuación se puede ver la versión resumida que permite obtener las muestras. En la aplicación, esto resulta mas complejo ya que se decide si alguno de los canales se deshabilita para su lectura, el control de la ventana de progreso y el manejo de los errores.

```

1 function LeerValores(app)
2 %Leo valores
3 flush(app.serial.connection,"input");
4 flush(app.serial.connection,"output");
5
6 %Solicito los valores de las configuraciones
7 app.raw_data.dt = str2double(writeread(app.serial.connection,"WFMPRE:CH1:XINCR?"));
8 pp.raw_data.trig_pos = str2double(writeread(app.serial.connection, "HORIZONTAL:TRIGGER:POSITION?"));
9
10
11 %Solicito las muestras del canal 1
12 writeline(app.serial.connection,"DATA:SOURCE CH1"); pause(0.5);
13 writeline(app.serial.connection,"CURVE?"); pause(0.5);
14 ch_t = readline(app.serial.connection);
15 app.raw_data.ch1 = str2double(split(ch_t, ","));
16
17 %Leo sus configuraciones
18 app.raw_data.ch1_mult = str2double(writeread(app.serial.connection, "WFMPRE:CH1:YMULT?"));
19 app.raw_data.ch1_zero = str2double(writeread(app.serial.connection, "WFMPRE:CH1:YZERO?"));
20 app.raw_data.ch1_off = str2double(writeread(app.serial.connection, "WFMPRE:CH1:YOFF?"));
21
22 %Solicito las muestras del canal 2
23 writeline(app.serial.connection,"DATA:SOURCE CH2"); pause(0.5);
24 writeline(app.serial.connection,"CURVE?"); pause(0.5);
25 ch_t = readline(app.serial.connection);
26 app.raw_data.ch2 = str2double(split(ch_t, ","));
27
28 %Leo sus configuraciones
29 app.raw_data.ch2_mult = str2double(writeread(app.serial.connection, "WFMPRE:CH2:YMULT?"));
30 app.raw_data.ch2_zero = str2double(writeread(app.serial.connection, "WFMPRE:CH2:YZERO?"));
31 app.raw_data.ch2_off = str2double(writeread(app.serial.connection, "WFMPRE:CH2:YOFF?"));
32 end

```

Por otra parte, se presenta a continuación la función callback que se ejecuta al presionar el botón conectar.

```

33 function btn_serialConnectButtonPushed(app, event)
34 app.serial.connection = serialport(app.dd_serialPort.Value, app.ef_serialBPS.Value, "Timeout", 10);
35
36 %Configuro el Encoding y los Bytes
37 writeline(app.serial.connection,"DATA:ENCDG ASCII");
38 writeline(app.serial.connection,"HEADER OFF");
39 writeline(app.serial.connection,"DATA:WIDTH 2");
40
41 %Verifico que sea el modelo correcto
42 resp = writeread(app.serial.connection, "ID?");
43 if ~strcmp(strip(resp, 'right'), "ID TEK/TDS 360,CF:91.1CT,FV:v1.09")
44     e.message = 'Erro, el ID no corresponde al Osciloscopio TDS360';
45     e.identifier = 'Serial:IDError';
46     error(e);
47 end
48 end

```

El resto de la programación no es relevante en cuanto a la comunicación con el osciloscopio, por tal motivo se adjunta al documento, un archivo .zip con la versión editable de la misma.

## **2.2. Configuración en el Osciloscopio**

El osciloscopio tiene el menú para configurar los parámetros de la comunicación serial, la aplicación esta desarrollada para comunicarse con los mismos parámetros que tiene de fabrica el osciloscopio, por tal motivo si hubiera problemas en la comunicación, no siendo una motivo la velocidad de transmisión, entonces se podrá ajustar la configuración de fabrica y asi resolver el inconveniente.

## Capítulo 3

# Manual de la Aplicacion

### 3.1. Conexión USB

Como se menciona en las secciones anteriores, se requiere el adaptador RS232-USB, por tal motivo se adjunta el driver correspondiente a el modelo utilizado en el Laboratorio.

Al abrir la aplicación se tiene la sección de **Conexión**, Después de seleccionar el puerto COM y la Velocidad de Transmisión correspondiente al Osciloscopio, si se presiona **Conectar** se visualizara el siguiente mensaje.

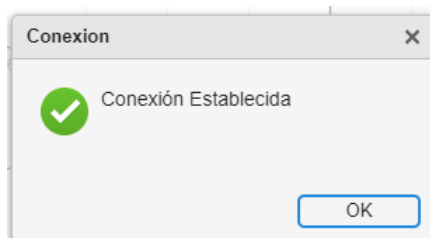


Imagen 3.1: Cuadro de dialogo - Conexión Exitosa

En este momento, el bus de datos no se encuentra ocupado, simplemente configuro el Encoding y luego verifico el modelo del osciloscopio. Ahora se permite leer los valores del osciloscopio.

Al presionar **Leer Valores** realizara la rutina desarrollada en la sección anterior. Lee los parámetros correspondiente al tiempo, luego el canal 1 junto con sus parámetros y por ultimo el canal 2 con sus parámetros. Finalmente, la aplicación armara el vector de tensión, y con la cantidad de puntos leídos estará en condiciones de armar el vector de tiempo.

En caso de existir algún error en la comunicación o en los datos, se visualizara en pantalla con un mensaje indicando el mismo.

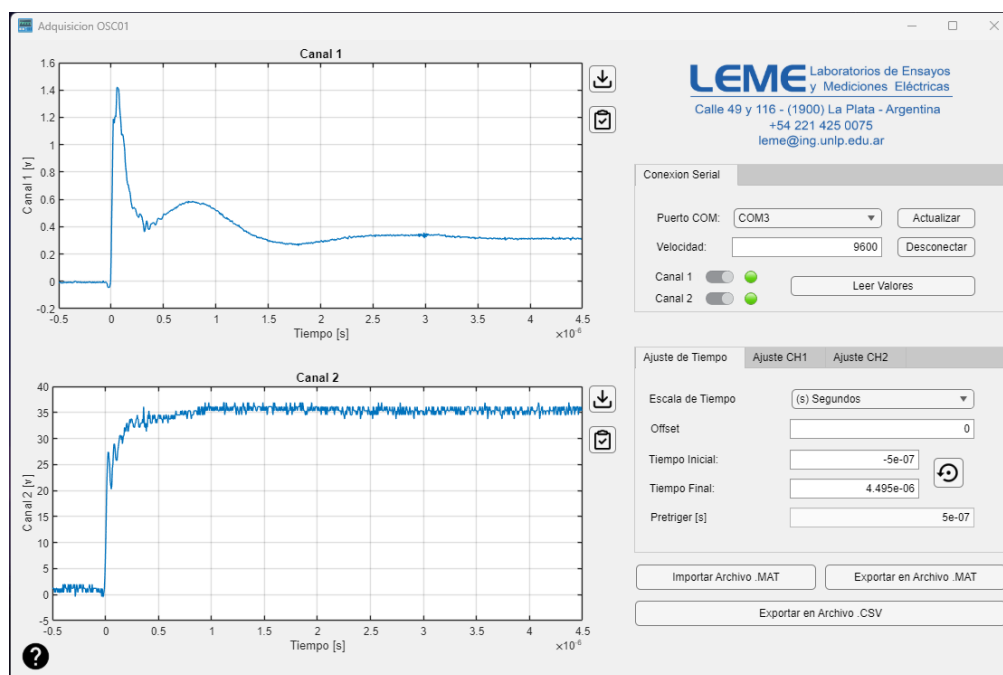


Imagen 3.2: Descarga correcta de valores

Una vez leído los valores y presentados en los ejes correspondientes, se habilitan los controles correspondiente al tiempo, y a los volts en cada canal.

### 3.1.1. Ajustes de Tiempo

La sección de Ajustes de Tiempo permite cambiar las unidades de la escala de tiempo, de esta forma se elimina el multiplicador  $\times 10^{-6}$  como se ve en la imagen 3.2.

Imagen 3.3: Control de ajustes de tiempo

Ademas, se permite configurar un Offset de tiempo, en las unidades elegidas en el selector. Esto permite desplazar el valor de tiempo cero con el fin de anular el tiempo del pre-triger, o bien simplemente desplazar el eje de tiempo.

Los dos campos de **Tiempo Inicial** y **Tiempo Final** permiten acotar la ventana de tiempo, con el fin de poder dejar visible una ventana de tiempo específica. En caso de requerir restaurar la ventana al tiempo que muestreo el osciloscopio, se utiliza el boton para restablecerlo, ubicado entre los dos campos nombrados.

Finalmente, el ultimo campo no es editable y simplemente deja visualizar el valor de tiempo que dura el pre-triger, es decir, sera el tiempo transcurrido entre la primer muestra y el disparo del trigger.

### 3.1.2. Ajustes de los Canales

Al igual que los ajustes de tiempo, se permite ajustar algunos parámetros de las gráficas de cada canal. Los primeros dos, **Título** y **Etiqueta** son los textos que aparecerán en las gráficas, siendo esto de ayuda para el momento en que se deba exportar la imagen para algún informe.

Imagen 3.4: Control de ajustes de tiempo

El **Multiplicador** permitirá aplicar un factor por el cual se multiplica cada elemento del vector de tensión correspondiente al canal de la ficha seleccionada.

Los valores de **Valor Mínimo** y **Valor Máximo** simplemente ajustan los limites del eje vertical con el fin de dar libertad al momento de exportar la imagen. También se dispone del botón que permite restablecer y ajustar los limites a la amplitud de la curva.

En el Canal 2, se dispone de los mismos parámetros, siendo aplicados a la grafica inferior.

## 3.2. Exportar en Archivo .MAT

El archivo `.mat` es propio de MatLab y permite que desde el propio software se puedan leer las variables para lograr manipular o trabajar con estas muestras leídas.

Para generar este archivo, simplemente se presiona el botón **Exportar en Archivo .MAT**. Se abrirá una ventana donde se ubicará seleccionará la ruta y el nombre del mismo, logrando así generar el archivo en cuestión.

Para no modificar la información que se leyó, se devuelve la siguiente estructura de datos:

- `dt` : Es el valor de tiempo entre muestras.
- `offset_t` : Es el desplazamiento que se realiza por el usuario en el vector tiempo.
- `ch1` : Contiene los datos del canal 1, los cuales serán:
  - `ch1.raw` : Será el vector que contiene la totalidad de las muestras leídas, correspondientes al canal 1.
  - `ch1.ymult` : Es el factor por el cual se deberá multiplicar cada muestra.
  - `ch1.yoff` : Es el valor que se le deberá restar al valor RAW leído (antes de usar el multiplicador).
  - `ch1.yzero` : Es el valor de tensión que se le deberá sumar al valor ya convertido en tensión.
  - `ch1.titulo` : Corresponde al texto usado en el título de la gráfica.
  - `ch1.leyenda` : Corresponde al texto usado en la leyenda del eje vertical.
- `ch2` : Se repite la misma estructura que en `ch1`.

Para lograr usar estos valores en matlab, se puede implementar el siguiente script:

```
1 load('NombreArchivo.MAT');
2
3 t = (0:dt:(length(ch1.raw) - 1)*dt) - trig_pos;
4 ch1_v = (ch1.raw - ch1.yoff) .* ch1.ymult + ch1.yzero;
5
6 plot(t, ch1 .* ch1.factor)
```

De esta forma, en MatLab tendremos ahora el vector `t` y el vector `ch1_v`, los cuales representan el tiempo y los volts (del canal 1). El valor del factor que se utilizó como multiplicador se almacena en `ch1.factor`.

## 3.3. Exportar como .CSV

Para permitir exportar archivos y manipularlos en aplicaciones como Microsoft Excel, se implementa un botón que permite exportarlos con la extensión .CSV. Básicamente hace uso de la función `writetable()` de MatLab, la cual está configurada para que guarde únicamente 3 columnas:

- `Tiempo` : Es el valor del tiempo para cada muestra descargada.
- `CH1` : Serán las muestras, en Volts del canal 1.
- `CH2` : Al igual que el anterior, serán las muestras, en Volts del canal 2.

Por último, el delimitador se configura en **tabs**.

## 3.4. Importar archivo .MAT

La importación de este archivo permite que vuelva a visualizar la curva que se encontraba almacenada en el archivo .MAT, de esta forma se podría modificar los valores de los campos disponibles y volver a guardar en un archivo nuevo, o bien como alternativa para editar y luego sobrescribir el archivo.

Para esto, la aplicación dispone del botón **Importar Archivo .MAT**, la cual primero verificará que las variables mencionadas en la sección 3.2 estén todas contenidas en el archivo. Si se cumple esta condición, entonces se cargarán los vectores y los valores, completando así en cada campo el valor correspondiente.

## 3.5. Exportar Imágenes

Para esta funcionalidad, se dispone de dos métodos:



Imagen 3.5: Botones para exportar imagen

El primero corresponde permite **Descargar** como un archivo de imagen, mientras que el segundo permite **Copiar** al portapapeles. Como se vera a continuación, cada uno de ellos permitirá elegir la extensión de la imagen.

### 3.5.1. Exportar como Archivo de Imagen

Al presionar el primer botón, aparecerá un cuadro de dialogo que permitirá seleccionar si el archivo se exportara como imagen vectorial (PDF), o simplemente como una imagen (PNG).

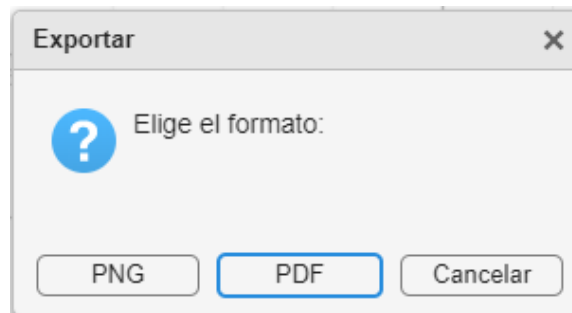


Imagen 3.6: Selección del Formato

Luego de la selección, se solicitara la ruta y el nombre del archivo que se exportara.

### 3.5.2. Exportar al Portapapeles

Al presionar el segundo botón, aparecerá un cuadro de dialogo que permitira seleccionar si el archivo se copiara como imagen vectorial (SVG), o como simplemente imagen (PNG).

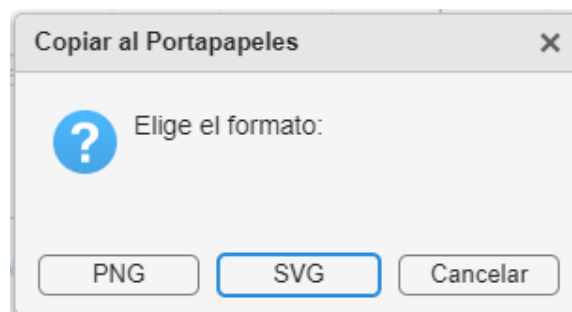


Imagen 3.7: Selección del Formato

Luego de la selección, simplemente usar la imagen como si de un texto copiado se tratase.