

# Introducción a la programación con MatLAB

## Módulo 08 - Archivos en matlab

Agustín - Andrés - Gabriel - Fernando<sup>1</sup>

<sup>1</sup> Universidad Tecnológica Nacional  
Facultad Regional Buenos Aires

2018

IEEE  
Sección Argentina



# Importación de datos

Los datos se almacenan en muchos formatos diferentes. Algunos ejemplos podrían ser :

- Sonido : se almacena en un archivo .wav
- Imagen : archivos .jpg
- Tablas de excel : .xls

Para conocer los formatos admitidos por MATLAB escribir **doc fileformats** en la ventana de comandos. Aunque para nuevas opciones se recomienda recurrir a la web de **Mathworks**.

# Tipos de archivo soportados por Matlab

Tipo de archivo	Extensión	Observación
Texto	.mat .dat .txt	Area de trabajo matlab Datos ASCII Datos ASCII
Formatos comunes de datos científicos	.cdf .fits .hdf	Datos comunes Transporte de imágenes Datos jerárquicos
Datos de hoja de cálculo	.xls .wk1 .tiff	Hoja de cálculo Excel Lotus 123 Archivo de imagen etiquetado
Datos de imagen	.bmp .jpeg o jpg .gif	Mapa de bits Grupo experto fotográfico unido Formato de intercambio gráfico
Datos de audio	.au .wav .flac	Audio Archivo wave Microsoft Free Lossless Audio Codec
Película	.avi	Archivo intercalado audio/video

# Archivos disponibles

## Utilidad

Puedo conocer los archivos del directorio desde un script.

```
dir;  
script=dir('*.m')  
texto=dir('*.txt')
```

# Archivos disponibles

pero... ¿cómo accedo a esa información ?



# Importación de datos

Conociendo el tipo de formato a importar puede utilizar una función de importación.  
Por ejemplo :

```
[data , fs] = audioread( 'ArrozConLeche.wav' );
```

Lee la canción *ArrozConLeche*

para esta función

**audioread** soporta archivos :WAVE (.wav), OGG (.ogg), FLAC (.flac), AU (.au), AIFF (.aiff, .aif), AIFC (.aifc) y en versiones recientes incluye MP3.

## Cargar datos desde un archivo ASCII

- Un archivo ASCII contiene datos como texto
- Todas las filas contienen el mismo número de datos.

Un ejemplo de archivo **texto.txt** puede ser :

texto.txt			
2.5	7	-3.2	4
5	2.1	3.7	12
-2	-0.3	37	-19
4	3.2	-1	0

### Comando

Ver comando : **load()**

Para cargar el archivo *texto.txt* se escribe :

```
variable = load( 'texto.txt' ) ;
```

# Función textread

- Lee string y datos numéricos desde un archivo utilizando especificadores de conversión.
- Los especificadores de conversión son por ejemplo formato de datos.
- La función es útil cuando el archivo tiene un formato uniforme.

Un ejemplo de archivo **text.txt** puede ser :

texto.txt			
2.5	7	-3.2	4
5	2.1	3.7	12
-2	-0.3	37	-19
4	3.2	-1	0

## Comando

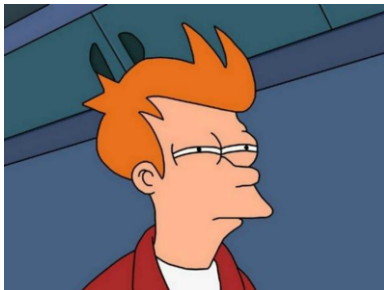
```
variable = textread('texto.txt');
```



## archivo de texto

### Comando

escribir el siguiente archivo y leerlo :  
Hola, soy un txt.



# Función textread

Para leer un archivo **.dat**. Por ejemplo :

```

                personas.dat
Manuel    Hombre    20    Mayor
Camila    Mujer     19    Mayor
Juan      Hombre    33    Mayor
Florencia Mujer     14    Menor

```

## Formato

```
[A,B,C, ...] = textread('archivo','formato',N)
```

N es el número de filas que se deseen leer. El valor -1 permite leer todo el archivo.

Para cargar el archivo *personas.dat* se escribe :

```
[nom, tipo , edad , est]=textread ( 'personas.dat' , '%s %s %f %s' , -1)
```

# Función dlmread

## Comando

Ver comando : **dlmread()**

La función **dlmread** permite leer una lista de valores desde un archivo separado por delimitadores.

Ej. Leer la siguiente tabla de datos separados por ;

signal.dat			
4;	3;	2.4;	7
-3;	0.33;	20;	12
1;	1.7;	9;	12.4
0.33;	9.3;	-2;	3.3

```
datos=dlmread('signal.dat', ';');
```

# Función xlsread

## Comando

Ver comando : **xlsread()**

- xlsread lee una hoja de cálculo de formato excel (xls)
- Las celdas vacías o de texto serán retornadas como NaN en el dato

Ej. Leer la siguiente tabla de datos

datos.xls

4	3	2.4	7
-3	0.33	20	12
1	1.7	9	12.4
0.33	9.3	-2	3.3

```
datos=xlsread('datos.xls');
```

# Función xlsread

Ej. Leer la siguiente tabla de datos

datos.xls

Canal 1	Canal 2	Canal 3	Canal 4
4	3	2.4	7
-3	0.33	20	12
1	1.7	9	12.4
0.33	9.3	-2	3.3

```
[datos , canales , todo]=xlsread( 'datos.xls' ) ;  
% devuelve numeros,texto y los datos sin procesar
```

## sobre los archivos xls

Microsoft no es copado

El uso de las funciones dedicadas depende de la aplicación **Microsoft Excel**



IEEE  
Sección Argentina



# Exportación de datos

Se verán tres formas de exportar datos :

- save - Salva el espacio de trabajo (workspace)
- dlmwrite - Guarda un arreglo utilizando delimitadores
- xlswrite - Guarda un arreglo en una hoja de excel
- audiowrite - Crea un archivo de audio

# Función save

## Comando

Ver comando : **save** y **load**

La función **save** guarda el espacio de trabajo (workspace) en forma binaria creando un archivo .mat

La función **load** carga el archivo .mat recuperando el workspace salvado.

## Tener en cuenta

Para guardar el workspace con un determinado nombre se escribe :

```
save( 'mi_workspace.mat' );  
% genera el archivo y guarda
```



# Función save

Ej. Ejecutar las siguientes líneas. Obtener conclusiones.

```
save( 'mi_workspace.mat' );  
clear  
load( 'mi_workspace.mat' );
```

# Función dlmwrite

## Comando

Ver comando : **dlmwrite**

La función **dlmwrite** escribe el arreglo en un archivo delimitado por ASCII

## Comando

Ver comando : **type**

La función **type** visualiza el archivo .txt

# Función dlmwrite

Ej. Ejecutar las siguientes líneas. Obtener conclusiones.

```
Matriz = magic(3) ;  
dlmwrite( 'Archivo.txt', Matriz, '&' ) ;  
type( 'archivo.txt' ) ;
```

% posteriormente

```
Matriz = magic(3) ;  
dlmwrite( 'Archivo.dat', Matriz, '&' ) ;  
load( 'archivo.txt' ) ;
```

# Función xlswrite

## Comando

### Ver comando : **xlswrite**

La función **xlswrite** Guarda arreglo numérico o matriz en una hoja de Excel.  
Ej. Ejecutar las siguientes líneas. Obtener conclusiones.

```
Archivo = 'PrimeraTabla';  
Matriz = [1 2 3 4 ; 5 6 7 8 ; 9 10 11 12];  
xlswrite ( Archivo , Matriz );
```

# Función audioread

## Comando

Ver comando : **audiowrite**

Ej. Ejecutar las siguientes líneas. Obtener conclusiones.

```
fs=8000;           % f de muestreo
tiempo=3;          % tiempo en segundos
f=440;             % frecuencia del tono

T = 1/fs;          % periodo de muestreo
t = (0:fs*tiempo)*T; % Tiempo en ms
w=2*pi*f;
signal=sin(w.*t);

sound(signal,fs)
audiowrite('tono.wav',signal,fs);
```

# Consultas

**¿Preguntas  
ó Comentarios?**



IEEE  
Sección Argentina

