

JOSE DAVID PORRAS GONZALEZ

**UNIVERSIDAD ECCI
FACULTAD DE INGENIERÍA
COORDINACIÓN DE INGENIERIA BIOMÉDICA
TECNOLOGIA EN ELECTROMEDICINA
BOGOTÁ, D.C.
2018**



ANALISIS DE DATOS CON MATLAB

**JOSE DAVID PORRAS GONZALEZ
CODIGO: 9514**

Monografía de grado para optar al título de Tecnólogo en Electromedicina

**Asesor
ANGEL VALENTIN MOLINA MOJICA
Ing. Electrónico MsC En Ing. Biomédica**

**UNIVERSIDAD ECCI
FACULTAD DE INGENIERÍA
COORDINACIÓN DE INGENIERIA BIOMÉDICA
TECNOLOGIA EN ELECTROMEDICINA
BOGOTÁ, D.C.**

Aceptación

Firma del jurado

Firma del jurado

Firma del jurado

Agradecimientos

Después de un trabajo intenso durante seis meses, después de obtener conocer y difundir información de mucha importancia para mí. Sólo tengo agradecimientos para finalizar mi trabajo de grado. Ha sido de gran aprendizaje y conocimiento todo lo que he aprendido en este trabajo tanto como para mi vida personal como para mi vida profesional.

Primero que todo, me gustaría agradecer a mi tutor de esta monografía el profesor Ángel Valentín molina Mojica por su apoyo en cuanto a la realización del trabajo. Doy gracias por haberme dado la oportunidad de haber llevado a cabo este proyecto de gran importancia para mí aportándome diferente herramientas para finalizar este trabajo.

También me gustaría agradecer a mis padres por su comprensión y dedicación y por haber siempre estado ahí para mí.

Finalmente a mis amigos por haber estado durante mi proceso y haberme apoyado para la finalización de este proyecto.

¡Muchas gracias a todos!

José David porras González

Bogotá, 22 de julio de 2018

Tabla de contenido

ABSTRACT	9
1. INTRODUCCIÓN.....	10
2. PLANTEAMIENTO DEL PROBLEMA.....	11
3. FORMULACIÓN DEL PROBLEMA	12
4. JUSTIFICACIÓN.....	12
5. OBJETIVOS.....	13
5.1 OBJETIVO GENERAL	13
5.2 OBJETIVOS ESPECIFICOS	13
6. MARCO TEORICO	14
6.1 ¿QUE ES EL SOFTWARE MATLAB?.....	14
6.1.1. Entorno de programación en Matlab	14
6.1.2. Workspace.....	15
6.1.3. Current folder	15
6.1.4. Editor	15
6.1.5. Comand window	15
6.2 VECTORES Y MATRICES	15
6.2.1. Acceso a elementos de una matriz	15
6.2.2. Operaciones matriciales.....	16
6.3 ESTRUCTURAS PARA PROGRAMACIÓN.....	18
6.3.1. Sentencia for	18
6.3.2. Sentencia while.....	19
6.3.3. Sentencia if.....	20
6.3.4. Sentencia break.....	21
6.3.5. Sentencia continue.....	21
6.3.6. Graficas.....	22
6.3.7. Almacenamiento y llamado de Datos.....	23
6.4 TRANSFORMADA DE FOURIER.....	24
6.5 IMAGEN DIGITAL.....	24
6.6 Filtros de paso bajo:.....	24
6.7 Filtro de paso alto:.....	24

6.8 Filtros de banda	25
7. ANALISIS DE IMÁGENES.....	26
7.1 Descripción de una imagen	26
7.2 Etapas de análisis de una imagen.....	27
8. Nivel de grises de una imagen.....	28
9. Umbralización de una imagen	29
10. ANALISIS DE SEÑALES.....	31
10.1 ¿Que es una señal?.....	31
10.2 Señal analógica.....	31
10.3 Señal digital.....	31
11. CONCLUSIONES.....	36
12. CRONOGRAMA DE ACTIVIDADES	37
13.REFERENCIAS BIBLIOGRÁFICAS	38

Ilustración 1 Entorno de Matlab. Fuente:autor.	15
Ilustración 2 Ejemplo comando for. Fuente:autor.....	19
Ilustración 3 Ilustración comando while. Fuente :autor.....	20
Ilustración 4 Ejemplo como utilizar el comando if y else if. Fuente:autor	23
Ilustración 5 Como utilizar el comando plot. Fuente: autor.....	24
Ilustración 6 Comando plot en la interfaz de matlab.....	25
Ilustración 7 Ilustración filas y columnas en una imagen. Fuente:autor.	28
Ilustración 8 Etapas en el análisis de imágenes. Fuente:autor.....	29
Ilustración 9 nivel de grises de una imagen. Fuente:autor	30
Ilustración 10 umbral de una imagen. Fuente:autor	31
Ilustración 11 imagen original ejemplo. Fuente:autor	31
Ilustración 12 Análisis de Imágenes. Fuente:autor.....	32
Ilustración 13 Amplitud y frecuencia de una señal analógica y digital	32
Ilustración 14 paso a paso analisis de señales	33
Ilustración 15 Umbralizar una señal	33
Ilustración 16 Señal electrocardiográfica.....	34
Ilustración 17 Datos representativos luego de una un proceso de umbralización	34
Ilustración 18 Umbralización con pérdida de datos	35

El análisis de datos (imágenes y señales) a través del software matlab, tiene como objetivo principal la manipulación de estos ya sea en el dominio del tiempo a fin de buscar eventos característicos relacionados con la temporalidad y el dominio de la frecuencia evidenciando el comportamiento frecuencial que posee una data en particular según un fenómeno específico. El presente trabajo muestra un enfoque descriptivo de la importancia de la utilización de software Matlab como instrumento robusto para analizar señales electrofisiológicas e imágenes médicas.

Matlab es una herramienta muy versátil y robusta que permite analizar datos asociados a la representación de fenómenos en distintas áreas del conocimiento, particularmente en ingeniería biomédica es indispensable el análisis de señales que puedan representar el comportamiento de un órgano en particular o análisis de imágenes médicas que de cierta forma hacen una representación bidimensional de estructuras asociadas al cuerpo humano.

Un enfoque a considerar desarrollado en este documento es la particularidad de como evidenciar la importancia asociada a las rutinas específicas a utilizar según los datos que se estén analizando, en otras palabras, como se puede abordar una señal que hace representación de un vector y una imagen que en su defecto estará representada por una matriz.

Se buscará adentrar al lector del documento a fin de conocer el software en general para poder realizar procesamiento de señales digitales fisiológicas e imágenes médicas, enfocándose en la comprensión de la temporalidad y en análisis espectral de los datos analizados. Como resultado final se utilizarán señales e imágenes reales para mostrar análisis con metodologías sencillas aplicables por estudiantes de tecnología en electromedicina e ingeniería biomédica.

Palabras clave: Análisis de datos, análisis en frecuencia y tiempo, imagen, señal, muestreo, software.

The analysis of data (images and signals) through Matlab software has as main objective the manipulation of these either in the time domain in order to look for characteristic events related to the temporality and the domain of the frequency evidencing the frequency behavior that has a particular data according to a specific phenomenon. The present work shows a descriptive approach of the importance of using Matlab software as a robust instrument to analyze electrophysiological signals and medical images.

Matlab is a very versatile and robust tool that allows analyzing data associated with the representation of phenomena in different areas of knowledge, particularly in biomedical engineering. It is essential to analyze signals that may represent the behavior of a particular organ or medical image analysis. In a certain way they make a two-dimensional representation of structures associated with the human body.

One approach to consider developed in this document is the particularity of how to demonstrate the importance associated with the specific routines to be used according to the data being analyzed, in other words, how can a signal that represents a vector and an image be addressed that in its defect will be represented by a matrix.

We will seek to introduce the reader of the document in order to know the software in general to be able to process physiological digital signals and medical images, focusing on the understanding of temporality and spectral analysis of the analyzed data. As a result, real signals and images will be used to show analysis with simple methodologies applicable by students of technology in electromedicine and biomedical engineering.

Keywords: Analysis of data, analysis in frequency and time, image, signal, sampling, software.

Al adentrar al estudiante en el procesamiento de señales a través del software Matlab, es necesario describir los procesos que deben ser llevados a cabo para introducir a cada uno de los estudiantes. Se debe tener pleno conocimiento de lo que es una señal y su procesamiento digital, además de esto se dará a conocer el manejo de la plataforma Matlab con breves explicaciones de cómo utilizar este software, un breve adentramiento de lo que son las señales, e identificar, conocer y comprender el comportamiento de señales fisiológicas analizándolas desde una perspectiva más objetiva.

En el siguiente trabajo se dará una breve introducción de los que es una señal analógica que es aquella que presenta una variación continua en el tiempo, en esta se podrán observar parámetros como lo son la altura de pulso, la duración y la frecuencia, de este modo cualquier señal puede ser descompuesta en su frecuencia y para esto se debe llevar a cabo un proceso matemático que permita esta descomposición el cual se denomina análisis de Fourier en el cual se dará una breve explicación de cómo funciona y como se aplica en las señales para su posterior análisis.

Teniendo un conocimiento básico del software se dará una explicación o guía de cuáles son los comandos más comunes que se utilizan en este software, se darán ejemplos de cómo usarlos para posteriormente realizar el análisis de señales o de imágenes.

Se llevara a cabo un análisis tanto de imágenes como de señales en el cual el estudiante o la persona que lea el documento pueda ver de una manera más práctica y fácil como se procesa una imagen en este caso adquirida mediante una tomografía cerebral y posteriormente se pueda analizar comparando si tiene posibles patologías (tumores), identificándolos y analizándolos, luego de realizar ejemplos prácticos asociado al análisis de matrices se podrá comprender más los procedimientos utilizados en el análisis de imágenes médicas.

En el procesamiento de señales se analizarán señales fisiológicas aplicando conocimientos básicos de algoritmia, de esta forma se pueden estudiar los diferentes tipos de comportamientos tanto en el dominio de la frecuencia y tiempo de las señales.

En general el trabajo consistirá en elaborar un documento que permita de una forma más sencilla conocer sobre el análisis y procesamiento de datos vectoriales y matriciales para estudiantes salientes de la tecnología en electromedicina.

2. PLANTEAMIENTO DEL PROBLEMA

Los estudiantes que están saliendo del ciclo tecnológico, aunque han visto en distintas asignaturas conceptos asociados al análisis de datos, carecen de una profundización adecuada en el tema. Las temáticas abordadas no son suficientes, en cuanto a la utilización de un software específico. Durante la tecnología se aborda de forma categórica la lógica de la programación como fundamento único que le permite a un estudiante analizar e implementar ciertos pasos o instrucciones que den solución a un problema particular bajo aplicaciones de software.

La poca usabilidad de software de alto nivel como Matlab en los primeros semestres de la carrera, hace que los estudiantes tengan algunos tropiezos asociados al entendimiento de la sintaxis propia o estructura de ejecución en una plataforma como esta. El estudiante pierde gran parte de su tiempo estudiando cómo abordar una estructura adecuada de programación, y no enfatiza en profundizar realmente como solucionar un problema particular de la cotidianidad. Debido a esto los estudiantes no asimilan adecuadamente percepciones del análisis de datos en los dominios del tiempo y la frecuencia, siendo estos conceptos de vital importancia para evaluar el comportamiento de una señal (vector) o una imagen (matriz) y entender cómo se comportan los datos, para en base a esto tratar de brindar algún tipo de análisis de tipo cualitativo y porque no cuantitativo.

3. FORMULACIÓN DEL PROBLEMA

¿Sería de ayuda práctica en el estudiante tecnológico la utilización de una cartilla o documento que lo oriente en el análisis vectorial y matricial de datos (señales e imágenes)?

4. JUSTIFICACIÓN

Si el estudiante que está saliendo del ciclo tecnológico mejora considerablemente el análisis de datos vectoriales y matriciales, este, abordará sin problemas temáticas propias de la ingeniería y será capaz de analizar una señal electrofisiológica o una imagen médica a fin de cuantificar o detectar características propias de los datos analizados. Cuando se logre reducir considerablemente el tiempo asociado a la comprensión de la sintaxis del software Matlab, los estudiantes estarán capacitados para abordar rápidamente problemas complejos en el análisis de datos de gran tamaño.

5. OBJETIVOS

5.1 OBJETIVO GENERAL

Estructurar un documento que permita al estudiante saliente de tecnología en electromedicina, la fácil comprensión en el análisis de datos vectoriales y matriciales.

5.2 OBJETIVOS ESPECIFICOS

- Estudio del arte orientado al análisis de señales e imágenes médicas.
- Identificar los procesos que deben llevar a cabo para realizar un procesamiento de una señal fisiológica.
- Analizar señales electrofisiológicas reales.
- Analizar imágenes médicas.
- Crear un documento que permita enfocar al estudiante en el procesamiento de señales con Matlab.

6.1 ¿QUE ES EL SOFTWARE MATLAB?

MATLAB es el nombre abreviado de “MATriz LABoratory”. Es un programa para realizar cálculos numéricos con vectores y matrices, y por tanto se puede trabajar también con números escalares (tanto reales como complejos), con cadenas de caracteres y con otras estructuras de información más complejas. Matlab es un lenguaje de alto rendimiento para cálculos técnicos, es al mismo tiempo un entorno y un lenguaje de programación. Uno de sus puntos fuertes es que permite construir nuestras propias herramientas reutilizables. Podemos crear fácilmente nuestras propias funciones y programas especiales (conocidos como M-archivos) en código Matlab, los podemos agrupar en Toolbox (también llamadas librerías): colección especializada de M-archivos para trabajar en clases particulares de problemas. Matlab, a parte del cálculo matricial y álgebra lineal, también puede manejar polinomios, funciones, ecuaciones diferenciales ordinarias, gráficos.[1]

6.1.1. Entorno de programación en Matlab

La interfaz de utilización del software Matlab ofrece varias ventanas que permiten de cierta forma la manipulación adecuada de procesos importantes en el software.

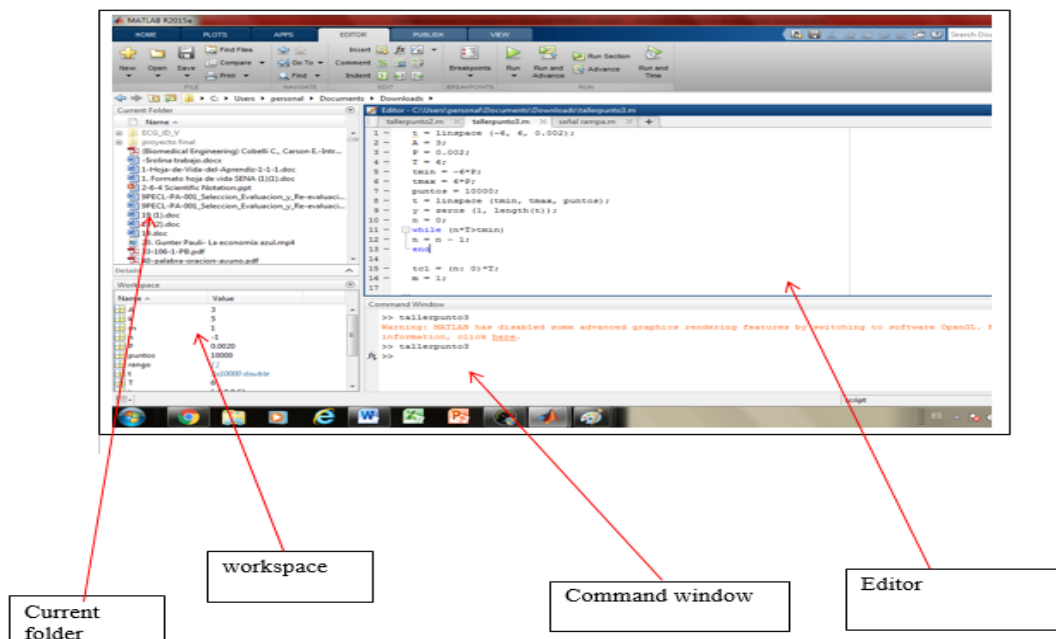


Ilustración 1, Entorno de Matlab

6.1.2. Workspace

El workspace es una de las ventanas que se encuentra en la pantalla inicial de software Matlab, básicamente es una ventana en la cual se almacenan las variables que hemos creado, aparte de borrar y crear variables, resulta fácil guardar todas las instrucciones que hemos tecleado en el comand window.

6.1.3. Current folder

Es una ventana en donde se puede visualizar los archivos existentes en la carpeta donde se guardan los datos necesarios junto a la programación.

6.1.4. Editor

Ventana principal de Matlab donde se realiza el código de programación.

6.1.5. Comand window

Se utiliza para introducir órdenes directamente por el usuario (seguidas de enter). Los resultados de las órdenes introducidas se muestran en esta misma pantalla. Cuando las órdenes se envían desde un programa previamente escrito, que en Matlab recibe el nombre de M-file, los resultados también aparecen en esta ventana. [2]

6.2 VECTORES Y MATRICES

De la misma forma que se definen variables escalares pueden definirse vectores y matrices. Por ejemplo, para definir la matriz.

```
M= [1 2 3 4 5 6]
```

Basta con teclear,

```
M = [1 ,2; 3,4; 5,6];
```

Donde los términos de una misma fila aparecen separados por comas, y el punto y coma hace de separador entre filas. Para trasponer M, basta teclear M'.

Existen una serie de funciones que definen matrices estándar como:

- Matriz de unos de dimensión N1xN2: ones (N1, N2);
- Matriz de ceros de dimensión N1xN2: zeros (N1, N2);
- Matriz identidad de dimensión N1xN1: eye (N1);

6.2.1. Acceso a elementos de una matriz

Imaginemos que hemos definido la matriz,

$$M = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

Dicha matriz aparecerá en el workspace una vez se ejecuta la instrucción y puede ser utilizada hasta que se borre.

Para acceder al elemento (i, j) de la matriz M tecleamos $M(i, j)$
Para acceder a la fila i de la matriz M tecleamos $M(i, :)$

Por ejemplo:

$$M(1, :)$$

Nos devolverá el vector fila

$$M[1 \ 2]$$

Para acceder a la columna j de la matriz M tecleamos $M(:, j)$

También podemos acceder a un rango determinado de la matriz. La orden

$$M(i1:i2, j1:j2)$$

Nos permite acceder a los elementos de la matriz M situados desde la fila i1 a la fila i2 y desde la columna j1 hasta la columna j2.

Por ejemplo

$$M(2:3, 1:1)$$

Nos devolvería el vector columna:

3

5

6.2.2. Operaciones matriciales

Matlab permite operar con matrices utilizando funciones como:

- Multiplicar, sumar y restar matrices A y B (siempre que tengan las dimensiones Adecuadas):

$A * B$; $A + B$; $A - B$;

- Inversa de la matriz A $Inv(A)$
- Determinante de la matriz A $Det(A)$
- Valores y vectores propios de los elementos de la matriz A: $Eig(A)$;
- Descomposición de cholesky de la matriz A: $Chol(A)$;
- Elemento de la diagonal de la matriz $Diag(A)$;
- Suma y producto, por columnas, de los elementos de la matriz A:

$Sum(A)$; $Prod(A)$;

- Matlab también permite operar “elemento a elemento” entre matrices.

$$A = \begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix} \quad B = \begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}$$

Matlab nos permite multiplicar ,dividir, dichas matrices elemnto a elemento. Para ellos debemos poner un punto delate del operador que deseemos utilizar:

Tabla 1: Resultados de operaciones entre las matrices A y B.

Operación	Orden	Resultado				
Multiplicación	$A .* B$	<table border="1"> <tr> <td>2</td> <td>12</td> </tr> <tr> <td>30</td> <td>56</td> </tr> </table>	2	12	30	56
2	12					
30	56					
División	$A ./ B$	<table border="1"> <tr> <td>0.5000</td> <td>0.7500</td> </tr> <tr> <td>0.8333</td> <td>0.8750</td> </tr> </table>	0.5000	0.7500	0.8333	0.8750
0.5000	0.7500					
0.8333	0.8750					
Potencia	$A.^B$	<table border="1"> <tr> <td>1</td> <td>81</td> </tr> <tr> <td>15625</td> <td>5764801</td> </tr> </table>	1	81	15625	5764801
1	81					
15625	5764801					

Para obtener la suma y producto por filas utilizaremos las mismas funciones, pero sobre la matriz transpuesta. [3]

$Sum(A')$; $Prod(A')$;

6.3 ESTRUCTURAS PARA PROGRAMACIÓN

La programación se lleva a cabo mediante un lenguaje parecido al lenguaje de alto nivel como BASIC o C, esto permite que se puedan agrupar sentencias que utiliza frecuentemente:

6.3.1. Sentencia for

Un bloque *for* en cada iteración asigna a la variable la columna *i*-ésima de la expresión y ejecuta las órdenes. En la práctica las expresiones suelen ser del tipo escalar:

```
For variable = expresión  
    <Orden>  
    <Orden>  
    ...  
    <Orden>
```

End

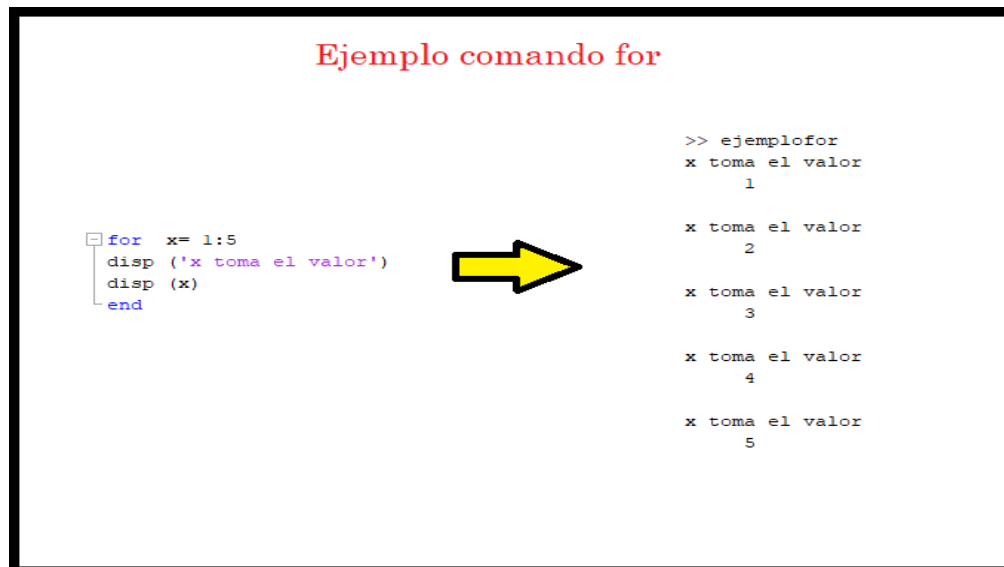


Ilustración 2, Ejemplo comando for.

- La sentencia **for** sirve para recorrer dato a dato ya sea un vector o una matriz.

6.3.2. Sentencia while

Un bloque *while* ejecuta las órdenes mientras todos los elementos de la expresión sean verdaderos.

While <expresión>

<Orden>

end

Ejemplo comando while

```
while a < 5
    disp ('a es menor que 5 ya que vale ')
    disp (a)
    a = a +1;
end
```

➔

```
>> ejemplowhile
a es menor que 5 ya que vale
3
a es menor que 5 ya que vale
4
```

Ilustración 3, Ilustración comando while

- El comando **while** muestra la sentencia, que mientras una condición como la del ejemplo 'a sea menor que 5 'se cumpla muestre los valores que sean menores al número indicado.

6.3.3. Sentencia if

Un bloque *if*, lo que hace es evaluar una expresión lógica y si es cierta ejecuta las órdenes que encuentre antes del *end*. Puede escribirse de varias maneras:

```
if <expresión>  
    <Órdenes evaluadas si la expresión es verdadera>  
end
```

Puede que nos interese que en caso de no ejecutar dicha orden ejecute otra distinta. Esto se lo indicaremos usando *e/se* dentro del bloque.

```
if <expresión>  
    <Órdenes evaluadas si la expresión es verdadera>  
else  
    <Órdenes evaluadas si la expresión es falsa>  
end
```

Si queremos dar una estructura mucho más completa, usaremos la más general donde sólo se evalúan las órdenes asociadas con la primera expresión verdadera de todas.

En cuanto la evalúe deja de leer el resto y se dirige directamente al *end*.

```
if <expresión1>  
    <Órdenes evaluadas si la expresión1 es verdadera>  
elseif <expresión2>  
    <Órdenes evaluadas si la expresión2 es verdadera>  
elseif <expresión3>  
    <Órdenes evaluadas si la expresión3 es verdadera>  
elseif  
    ...  
    ...  
else  
    <Órdenes evaluadas si ninguna otra expresión es verdadera>  
end
```

Ejemplo comando if y else if

```

>> valoresdeglucemia
valores de glicemia 160

a =
    60
valor bajo
valores de glicemia 2 83
b =
    83
valor normal
valor de glicemia 3 105
c =
    105
valor alto

```

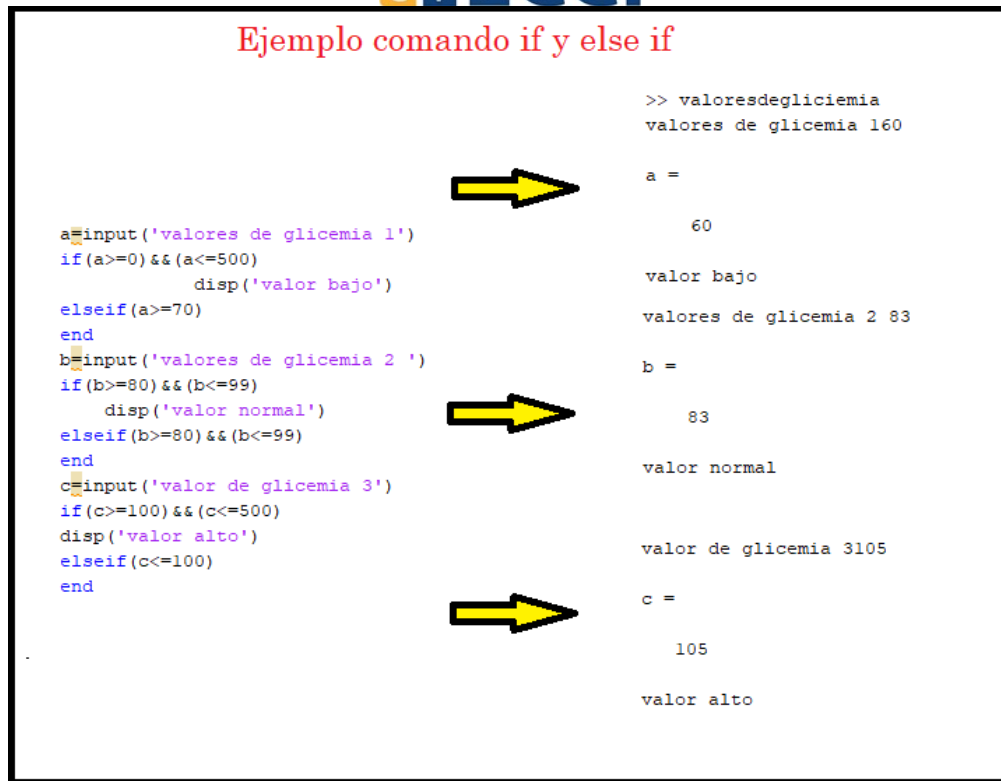


Ilustración 4, Ejemplo como utilizar el comando if y else if.

En el anterior ejemplo se puede ver como se utiliza el comando **if** de acuerdo a una expresión, si la expresión es verdadera el comando se ejecuta, también se utiliza el comando **elseif**, siempre y cuando se cumplan la condición ya especificada en el ejemplo se puede observar como un paciente puede observar si sus valores de glicemia son bajos normales o altos dependiendo de la condición de cada una. Los datos ingresados manualmente son comparados en cada condicional a fin de buscar una respuesta verdadera.

6.3.4. Sentencia break

Si queremos que en un momento dado termine la ejecución de un bucle *for* o un bucle *while* usaremos *break*.

6.3.5. Sentencia continue

La sentencia *continue* hace que se pase inmediatamente a la siguiente iteración del bucle *for* o del bucle *while* saltando todas las órdenes que hay entre el *continue* y el fin del bucle en la iteración actual.

6.3.6. Graficas

La orden *plot* genera una gráfica. Los argumentos deben ser vectores de la misma longitud.

Ejemplo:

La función *plot* nos permite otras opciones como superponer gráficas sobre los mismos ejes:

Ejemplo:

Como utilizar el comando *plot* para graficar

```

den=[1 2 24 48 25 50]
r1=roots(den)

den=[20 16 2 8 5 18]
r2=roots(den)
%plot(1,2)
figure(1)
zplane(0,r2)
                
```

```

>> ejemplo

den =
     1     2    24    48    25    50

r1 =
-0.0000 + 4.7863i
-0.0000 - 4.7863i
-2.0000
 0.0000 + 1.0416i
 0.0000 - 1.0446i

den =
    20    16     2     8     5    18

r2 =
-1.2352
 0.6373 + 0.6633i
 0.6373 - 0.6633i
-0.4197 + 0.8277i
-0.4197 - 0.8277i
                
```




Ilustración 5, Como utilizar el comando *plot*.

Se ejecuta el comando anterior en el software matlab se utiliza el comando **plot** para poder graficar y luego de esto permitirá ver el resultado del código ejecutado.

Resultado grafica utilizando el comando plot

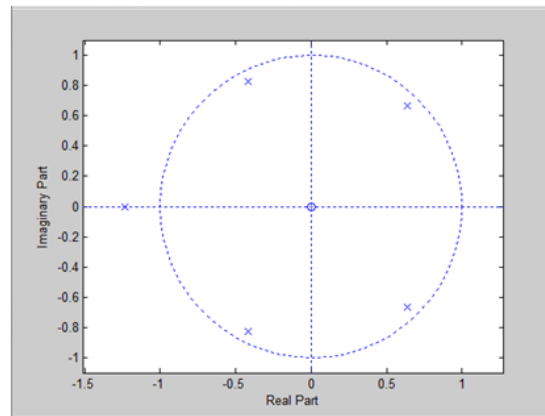


Ilustración 6, Comando plot en la interfaz de matlab. Fuente: matlab.

Se puede observar el resultado de la imagen utilizando el comando **plot**, para que la imagen posea una mejor presentación se pueden agregar títulos cambiar de color dependiendo de cómo queramos la gráfica.

6.3.7. Almacenamiento y llamado de Datos

Matlab permite guardar y cargar datos de los archivos del computador. En el menú File, la opción Save Workspace as... guarda todas las variables actuales y Import Data... carga variable de un espacio de trabajo guardado previamente.

Otra forma sería guardar el estado de una sesión de trabajo con el comando save antes de salir:

```
>> save
```

Al teclear esto, automáticamente se crea un fichero llamado matlab.mat. Puede recuperarse la siguiente vez que se arranque el programa con el comando load:

```
>> load [2]
```

6.4 TRANSFORMADA DE FOURIER

La transformada de Fourier es una herramienta importante dentro del procesamiento digital de imágenes. Esta es utilizada para descomponer una imagen en sus componentes seno y coseno. La salida de la transformación representa a la imagen en el dominio de Fourier o dominio de la frecuencia, mientras que la imagen de entrada está en el dominio espacial. Cada punto de la imagen en el dominio de Fourier representa una frecuencia particular contenida en la imagen en el dominio del espacio. La transformada de Fourier de una imagen nos permite aplicar distintos tipos de filtros a la imagen para contribuir a una correcta visualización de esta.

6.5 IMAGEN DIGITAL

Una imagen en un sentido muy amplio “es una representación, sin parecido o imitación de un objeto o cosa, es decir, una descripción gráfica, algo introducido para representar otra cosa más”. Una imagen contiene información descriptiva acerca del objeto que representa. Otra definición de imagen, orientada a una imagen gráfica, en inglés “picture” la define como “Una representación hecha por medio de dibujos, pinturas o fotografía”. La palabra digital se relaciona al cálculo por métodos numéricos o por unidades discretas. De esta manera podemos definir a una imagen digital como la representación numérica de un objeto. Una imagen monocroma puede ser definida como una función en dos dimensiones $f(x, y)$, donde x e y son coordenadas en el plano espacial, y la amplitud de f en cualquier par de coordenadas (x, y) es llamada la intensidad, nivel de gris o brillo de la imagen en ese punto. Cuando x , y y los valores de intensidad de f son finitos, cantidades discretas, esta imagen puede ser representada de manera digital (en una computadora).

6.6 Filtros de paso bajo:

Los filtros de paso bajo atenúan las componentes de medias-altas frecuencias y dejan intactas las bajas en función de la frecuencia de corte que se elija. Gráficamente eliminan todo lo que no sean variaciones suaves de nivel de gris. El más sencillo es el filtro de paso bajo ideal:

Donde $D(u, v)$ es la distancia euclídea de (u, v) al origen del plano de frecuencias. Este filtro suprime las altas frecuencias mayores que un cierto valor D_0 , que se denomina frecuencia de corte, y deja las demás tal como están.

6.7 Filtro de paso alto:

Deja inalterables las altas frecuencias y atenúa o elimina las bajas frecuencias. El resultado en el dominio del espacio consiste en un realzado de los cambios bruscos

de niveles de grises. De ahí que se use para detectar bordes. Las áreas de nivel de gris constantes o con poca variación se corresponden con bajas frecuencias, que se suprimen.

Donde $D(u, v)$ es la distancia euclídea de (u, v) al origen del plano de frecuencias. Este filtro suprime las frecuencias menores o iguales que un cierto valor D_0 que se denomina frecuencia de corte.

6.8 Filtros de banda

Un filtro de banda atenúa las altas y bajas frecuencias, pero mantiene intactas las frecuencias que se encuentran en una banda determinada. En el caso del filtro ideal, solo deja pasar las frecuencias que están entre dos frecuencias de corte. Se puede obtener un filtro de banda multiplicando uno de paso bajo por uno de paso alto, en el que la frecuencia de corte del de paso bajo sea superior a la de paso alto. El opuesto al filtro de paso de banda sería de “rechazo de banda” en el que se atenúan las frecuencias de la banda pero se mantienen las frecuencias fuera de ella.[4]

Una imagen es una función bidimensional $f(x, y)$, donde x e y representan las coordenadas espaciales y el valor de f en cualquier par de coordenadas (x, y) representa la intensidad de la imagen en dicho punto.

- Una imagen digital $f[x, y]$ descrita en un espacio 2D discreto se deriva de una imagen análoga $f(x, y)$ en un espacio 2D continuo a través de un proceso llamado digitalización.
- Matlab almacena las imágenes como vectores bidimensionales (matrices), en el que cada elemento de la matriz corresponde a un sólo pixel.[5]

7.1 Descripción de una imagen

- Al digitalizarla, la imagen continua en 2D es dividida en M filas y N columnas.
- La intersección de una fila y una columna se llama pixel.
- El valor asignado a las coordenadas $[m, n]$ con $m = \{1, 2, \dots, M\}$ y $n = \{1, 2, \dots, N\}$ es $f(m, n)$
- Es común que una imagen contenga sub-imágenes llamadas regiones de interés. [5]

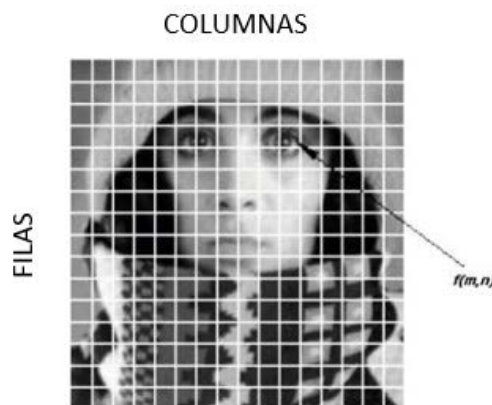


Ilustración 7, Ilustración filas y columnas en una imagen.

7.2 Etapas de análisis de una imagen

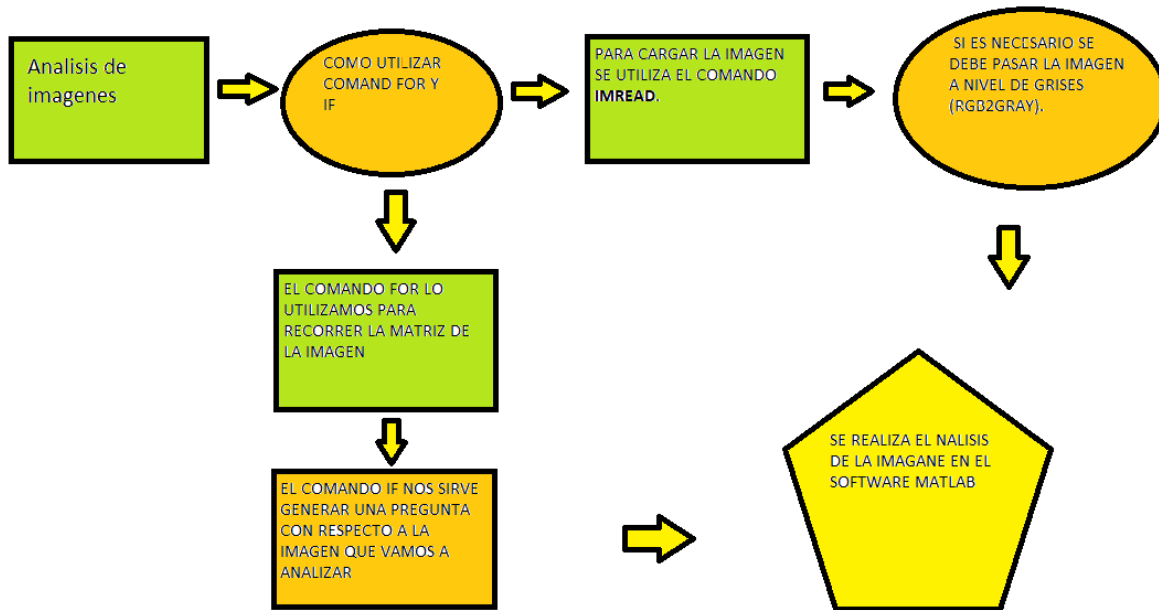


Ilustración 8, Etapas en el análisis de imágenes.

- En el anterior grafico se observa cómo se puede procesar y analizar una imagen con los comandos for e if, muestra paso a paso como debemos utilizar cada uno de estos comandos y de una manera fácil, también se puede observar cómo podemos cargar la imagen y también si es necesario pasar la imagen a escala de grises para seguir un posterior análisis de esta. [6]

8. Nivel de grises de una imagen

Ejemplo de como pasar una imagen en color a nivel de grises

```
imag=imread('diagrama-cerebro-670x410.jpg');  
title('imagen original')  
grises=rgb2gray(imag);  
imshow(grises)
```

Imagen original


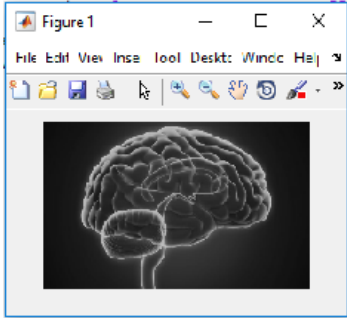


Imagen en nivel de grises



The figure illustrates the process of converting a color image to grayscale. On the left, a block of MATLAB code is shown. On the right, the original color image of a brain diagram is displayed above a downward-pointing arrow, which leads to a screenshot of a MATLAB window titled 'Figure1'. This window shows the same brain diagram but in grayscale, demonstrating the result of the `rgb2gray` function.

Ilustración 9, nivel de grises de una imagen.

- Para poder pasar la imagen que está a color en niveles de grises, utilizamos el comando **rgb2gray**, como lo muestra el ejemplo una vez cargada la imagen original se aplica el comando en la imagen y con el comando **imshow** podremos observar la imagen y su resultado en nivel de grises.

9. Umbralización de una imagen

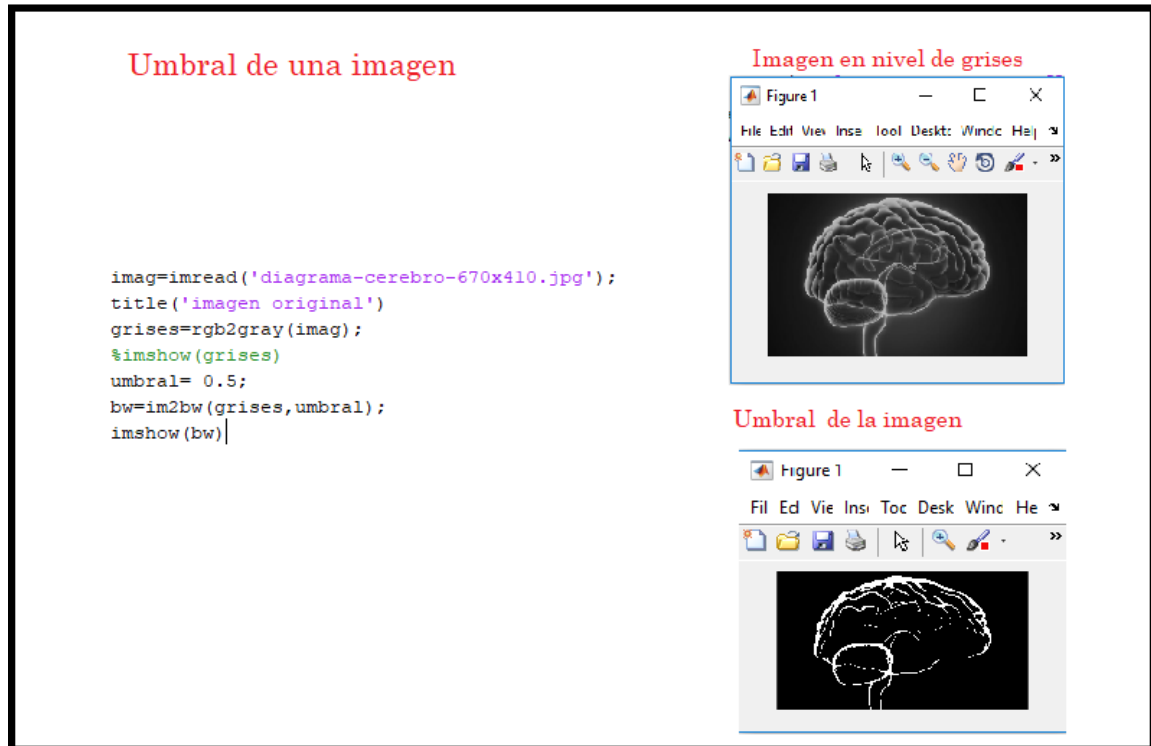


Ilustración 10, umbral de una imagen.

- si se quiere controlar el umbral de una imagen podemos utilizar la línea de código que se ve representada en la imagen, si queremos graduar el porcentaje de umbral lo podemos cambiar en la variable declarada umbral.

Ejemplo análisis de imágenes:

- buscar una imagen medica tomografía cerebral con tumor.
- graficar de color rojo o colocar un * de color rojo representando las partes de niveles de gris más altos de la imagen, que en este ejemplo podría representar una patología particular asociado a una malformación en el cerebro.

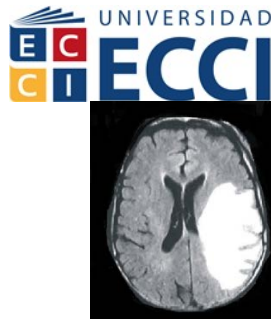


Ilustración 11, ejemplo imagen original.

Ejemplo analisis de imagenes

```
clear all
close all
clc

%en una imagen graficar de color
%rojo o colocar un *
%de color rojo valores
%que representen el tumor.

imag=imread('tumorcerebral2.jpg');
title('imagen original')
grises=rgb2gray(imag);
imshow(grises)
[f,c]=size(grises);
hold on
for j=1:c
    for i=1:f
        % disp (imag(i,j));
        if grises(i,j)>=250;
            hold on
            plot(j,i,'*r');
        end
    end
end
```

A screenshot of a MATLAB figure window titled 'Figure'. The window displays the same grayscale MRI brain scan as in Illustration 11. A yellow arrow points from the MATLAB code on the left to the figure window. The figure window shows the original image with several red asterisks (*) overlaid on the bright tumor area, indicating the detected pixels.

Ilustración 12, Análisis de Imágenes.

- se puede observar la imagen de una tomografía de un cerebro con tumor, el ejercicio nos propone que a través de un asterisco de color rojo marquemos la zona donde se encuentra el tumor o en su defecto los niveles de grises más altos.
- Se carga la imagen original, si está a color se pasa a nivel de grises, se utiliza el comando **for** para recorrer la matriz que genera la imagen [f,c] de este modo logramos que recorra la matriz tanto en sus filas como en sus columnas, luego de esto utilizamos el comando **if** para realizar una pregunta con respecto a la imagen donde, si supera ciertos pixeles se debe marcar la imagen de color rojo y nos debe identificar más o menos de cuanto es el tamaño del tumor. [7]

10. ANALISIS DE SEÑALES

10.1 ¿Que es una señal?

Una señal es un suceso que sirve para iniciar una acción; es decir puede incitar a la acción. Con las restricciones de energía y potencia, el interés se centra en el concepto de señal y también en la respuesta de un sistema a una señal dada. El diagrama de la figura 1.1 muestra las funciones de la señal, el sistema y la respuesta.[8]

10.2 Señal analógica

Una señal analógica es una señal que varía de forma continua a lo largo del tiempo. La mayoría de las señales que representan una magnitud física (temperatura, luminosidad, humedad, etc.) son señales analógicas. Las señales analógicas pueden tomar todos los valores posibles de un intervalo; y las digitales solo pueden tomar dos valores posibles.

Las señales análogas se pueden percibir en todos los lugares, por ejemplo, la naturaleza posee un conjunto de estas señas como es la luz, la energía, el sonido, etc., estas son señales que varían constantemente. Un ejemplo muy práctico es cuando el arco iris se descompone lentamente y en forma continua. Cuando los valores del voltaje o la tensión tienden a variar en forma de corriente alterna se produce una señal eléctrica analógica. En este caso se incrementa durante medio ciclo el valor de la señal con signo eléctrico positivo; y durante el siguiente medio ciclo, va disminuyendo con signo eléctrico negativo. Es desde este momento que se produce un trazado en forma de onda senoidal, ya que este da a lugar a partir del cambio constante de polaridad de positivo a negativo.

10.3 Señal digital

Una señal digital es aquella que presenta una variación discontinua con el tiempo y que sólo puede tomar ciertos valores discretos. Su forma característica es ampliamente conocida: la señal básica es una onda cuadrada (pulsos) y las representaciones se realizan en el dominio del tiempo. Sus parámetros son:

- Altura de pulso (nivel eléctrico)
- Duración (ancho de pulso)
- Frecuencia de repetición (velocidad pulsos por segundo)

Las señales digitales no se producen en el mundo físico como tales, sino que son creadas por el hombre y tiene una técnica particular de tratamiento, y como dijimos anteriormente, la señal básica es una onda cuadrada, cuya representación se realiza necesariamente en el dominio del tiempo. La utilización de señales digitales para transmitir información se puede realizar de varios modos: el primero, en función del número de estados distintos que pueda tener. Si son dos los estados posibles, se dice que son binarias, si son tres, ternarias, si son cuatro, cuaternarias y así sucesivamente. Los modos se representan por grupos de unos y de ceros, siendo, por tanto, lo que se denomina el contenido lógico de información de la señal. La segunda posibilidad es en cuanto a su naturaleza eléctrica. Una señal binaria se puede representar como la variación de una amplitud (nivel eléctrico) respecto al tiempo (ancho del pulso). Las señales digitales sólo pueden adquirir un número finito de estados diferentes, se clasifican según el número de estados (binarias, ternarias, etc.) Y según su naturaleza eléctrica (unipolares y bipolares). Una señal digital varía de forma discreta o discontinua a lo largo del tiempo. Parece como si la señal digital fuera variando «a saltos» entre un valor máximo y un valor mínimo. [9]

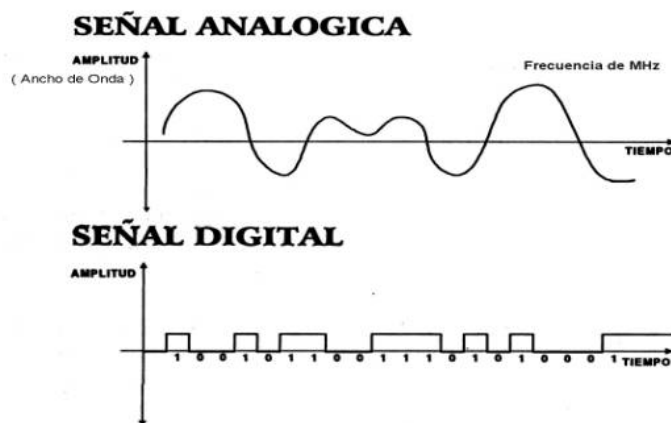


Ilustración 13, amplitud y frecuencia de una señal analógica y digital.

La ilustración 14 muestra un sencillo diagrama de bloques de las posibilidades ofertadas a la hora de poder analizar una señal digital, en este caso una señal electrofisiológica particular.

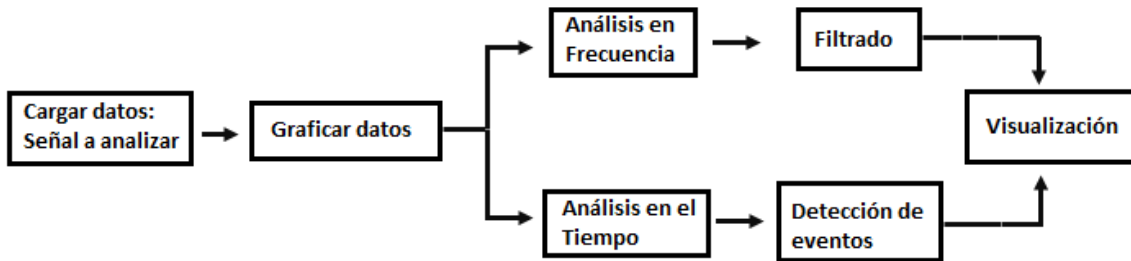


Ilustración 14, pasó a paso análisis de señales.

Lo ideal es poder recorrer dato a dato (comando for) un vector particular que haría referencia a una señal fisiológica, una vez se esté internamente en el vector existirá la posibilidad de hacer preguntas (comando if) relacionadas con amplitudes de la señal en instantes de tiempo particulares. A continuación, se mostrará un ejemplo donde se quiere detectar puntos característicos mediante umbralización sobre la señal de electrocardiografía ECG. En la ilustración 15 se muestra el comportamiento de la señal cargada con la función load. La idea fundamental del algoritmo se centra en la detección de los picos máximos asociados a la onda R de la señal.

Este procedimiento se logra mediante la utilización de los comandos for e if que permiten recorrer la señal y hacer preguntas relacionadas con el umbral respectivamente. [10]

```

    umbral=0.8*max(vector);
    for i=1:length(vector)
        if vector(i)>=umbral
            plot(i,vector(i),'r*')
        end
    end
  
```

Ilustración 15, Umbralizar una señal.

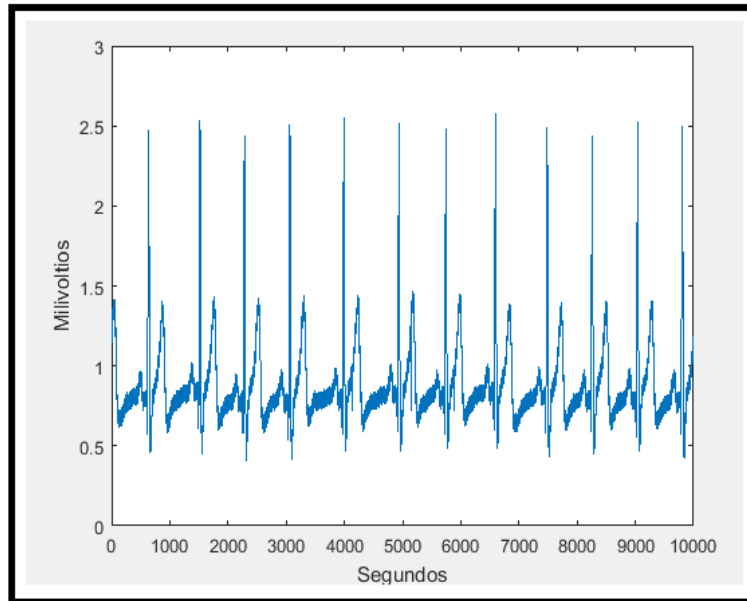


Ilustración 16, Señal electrocardiográfica.

El proceso de umbralización es una tarea de cuidado, esto debido a que mientras se varia el umbral existe la posibilidad de encontrar más o menos datos según el requerimiento.

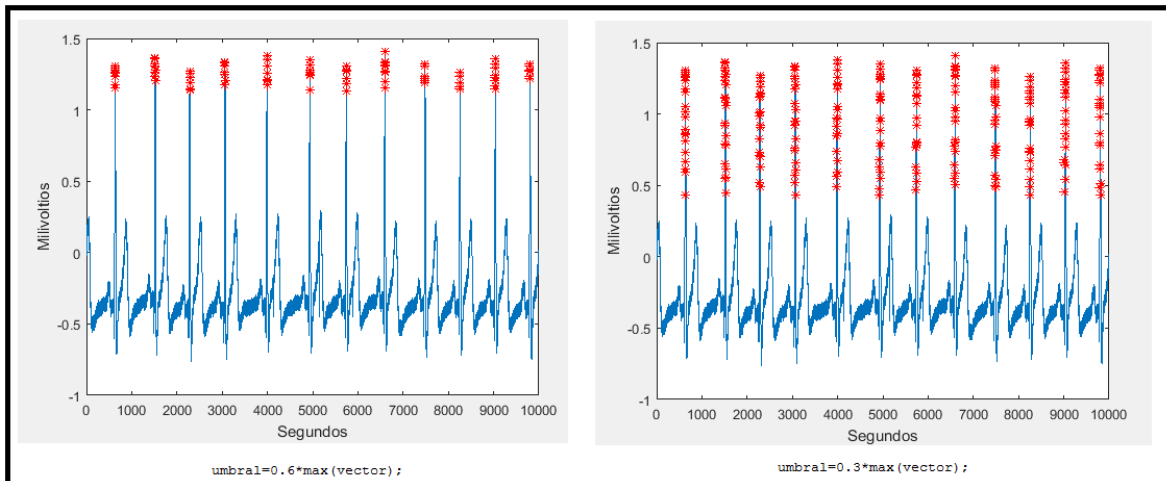


Ilustración 17, Datos representativos luego de una un proceso de umbralización.

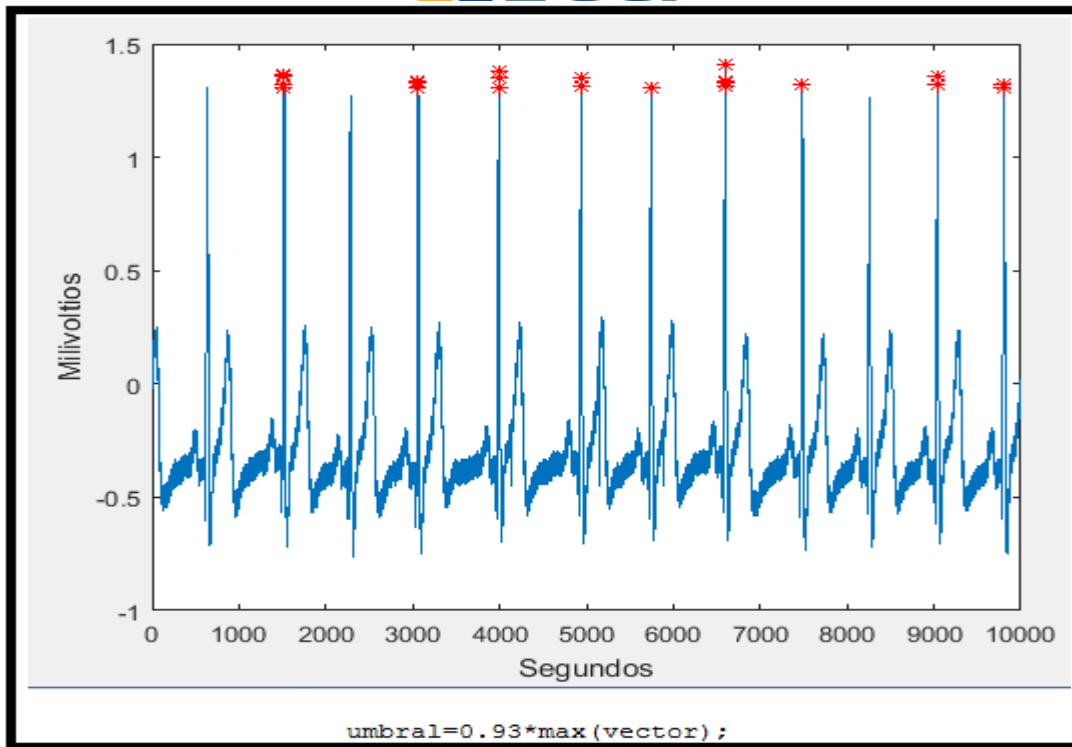


Ilustración 18, Umbralización con pérdida de datos.

Las ilustraciones 17 y 18 muestran los resultados luego de hacer varias umbralizaciones, es importante resaltar que a medida que se le asignan valores al umbral, los resultados pueden representar más localización o menos localizaciones asociadas al máximo dato en cada onda R de la señal ecg.

- Se logró crear un documento, el cual permite al estudiante que viene de tecnología entender de una manera más didáctica y fácil el manejo del software matlab orientado al procesamiento de señales e imágenes.
- Se consiguió analizar señales utilizando comandos muy comunes como los son if, for etc.
- A través de los análisis propuestos en el trabajo pude comprender más sobre como es el procesamiento de señales y como utilizar los comandos de la manera correcta para obtener un buen resultado.
- En el transcurso del trabajo, obtuve mayor conocimiento de lo que es una señal y como puedo analizarla.
- En función de esquemas básicos se consiguió mostrar procesos de cómo se analizan las imágenes y también de una manera más pedagógica para que el estudiante lo explore y lo entienda.
- explorando el software matlab descubrí comandos nuevos que me aportan más a mi conocimiento.
- Tras abordar el procesamiento de señales y de imágenes, se observó y se analizó el diferente comportamiento que presentan las diferentes señales fisiológicas.
- Después de realizar la cartilla identifique y analice la importancia de este documento para los estudiantes ya que es parte fundamental para que logren entender de una manera más práctica la forma de analizar y comprender las señales y las imágenes utilizando medios como lo son el software matlab.

12. CRONOGRAMA DE ACTIVIDADES

Duración Actividades	Febrero	Marzo	Abril	Mayo	Junio	Julio
Planteamiento del problema	■					
Justificación		■				
Objetivos		■				
Hipótesis			■			
Marco de referencia		■	■			
Metodología			■	■		
Resultados y discusión				■	■	
Elaboración del documento final				■	■	■
Revisión bibliográfica	■	■	■	■	■	■

- En la anterior tabla se puede observar como fue el proceso que se llevó a cabo para realizar el trabajo, primero se planteó el problema y que era lo que se quería realizar con este trabajo luego de esto se siguió con lo que son los objetivos, hipótesis. se comenzó a investigar y se creó un marco de referencia o marco teórico en el cual se investigó y se refirió cada uno de los textos citados para sustentar y dar apoyo a la cartilla.
- Durante estos meses se dio seguimiento a la revisión bibliográfica y se realizaron los diferentes tipos de análisis de señales que se dieron en el software matlab.
- Para finalizar la cartilla se revisaron los resultados que se dieron con los análisis de señales y diferentes ejercicios propuestos, se llevó a cabo la elaboración del documento final reuniendo toda la información.

13.REFERENCIAS BIBLIOGRÁFICAS

- [1] Desmond j. Digham, Dicolas j. Higham. (2000). *Matlab guide.uk: siam*
- [2] Fernández, c. *Manual básico matlab*. Recuperado el 10 de mayo del 2018 de <http://webs.ucm.es/centros/cont/descargas/documento11541.pdf>
- [3]González, i. (2010-2011).*El lenguaje de programación-vectores y matrices* .recuperado el 20 de julio de 2018 de <http://www.hep.uniovi.es/iglez/IFC/C/7-Vectores%20y%20matrices.pdf>.
- [4] Friedrich, v. (2013). *Transformada de Fourier de una Imagen y filtrado de una imagen en el dominio de la frecuencia. Funciones de Variable Compleja*. Recuperado el 15 de febrero de 2018 de <http://lcr.uns.edu.ar/fvc/NotasDeAplicacion/FVC-Virginia%20Friedrich.pdf>.
- [5] Ambardar, a. (2003). *Procesamiento de señales analógicas y digitales (2ª ed.)*.México: s.a. ediciones.
- [6] Universidad politécnica de Madrid departamento de electrónica. (2002).*introducción al procesamiento digital de señales* .España: elai-upm-doc□001-01.
- [7]Gabriela, g. (2006).*metodología de investigación básica: proceso de adquisición de señales fisiológicas*. España: UNED.
- [8] Wainschenker, r. Massa, j. Tristan, p. *Procesamiento digital de imágenes* .recuperado el 08 de junio 2018 de <http://www.exa.unicen.edu.ar/catedras/pdi/FILES/TE/CP1.pdf>.
- [9] Barchiesi, J. (2008).*introducción al procesamiento digital de señales*. Chile: universitarias de Valparaíso.
- [10] M. en C. Arturo Curiel Anaya, M. en C. Mariano Pozas Cárdenas. (2013).*Comunicación en redes* .España: universidad autónoma del estado de hidalgo, instituto de ciencias básicas e ingeniería.

[11] Rodríguez, j. (2005). *Adquisición de señales biológicas*. Recuperado el 20 de julio de 2018 de <http://www.usc.es/catedras/telemedicina/2005/materialAsignatura/AdquisicionSenalesBiologicas.pdf>