

**MATRÍZ DE LEDS (8X8) CON MATLAB Y GUIDE PARA CREAR DOS
IMÁGENES AL LADO DE UN SEMÁFORO**

**MORGAN G. VÁSQUEZ
INGENIERO MECATRÓNICO**

INTRODUCCIÓN

Las interfaces gráficas de usuario (GUI-Graphical User Interface en inglés), es la forma en que el usuario interactúa con el programa o el sistema operativo de una computadora. Una GUI contiene diferentes elementos gráficos tales como: botones, campos de texto, menús, gráficos, etc. Cada uno de estos elementos se pueden programar para que cumplan una función específica en el momento en que el usuario haga uso de ellas. Para la solución del problema que se ha planteado en ésta actividad, haremos uso de la herramienta guide de Matlab la cual nos permite dar una solución usando elementos llamados radiobuttons, los cuales serán los leds que queremos activar. Estos tienen dos estados, activado o desactivado, que también podríamos llamar 1 (activado) y 0 (desactivado). A partir de esto crearemos la matriz de 8x8, los botones y la gráfica que necesitaremos para visualizar la imagen para caminar y parar junto con el resto de puntos a solucionar.

PROBLEMA

En una empresa de publicidad, se requiere implementar un programa que permita al diseñador bosquejar dos imágenes a visualizar en una matriz de leds a instalar al lado de un semáforo, la idea es mostrar una imagen que le indique al peatón cuando caminar y otra para cuando parar; para ello el aplicativo que diseñe debe permitir:

- Crear una matriz (no inferior a 8x8)
- El programa debe permitir que el usuario establezca que leds encenderán para formar la imagen a partir del ingreso de un 1 lógico en las casillas que determine, para cada imagen
- Mostrar la matriz diseñada con la imagen a mostrar para caminar y parar.
- Simular con un temporizador, la visualización alterna de cada matriz, el usuario determina dicho tiempo.
- Mostrar una matriz que contenga la coincidencia de led encendido para la imagen de caminar y parar.

CÓDIGO FUENTE

Cuando se crea una interfaz gráfica con Matlab (guide), se crean 2 archivos, uno donde se crean los botones, gráficos, etc y otro donde se realiza el código para estos elementos. Para ésta actividad, hemos nombrado a éstos archivos **matriz_leds.fig** y **matriz_leds.m** respectivamente. Además, para iniciar con el programa hemos creado un archivo llamado **matriz_leds_lanzador.m**.

Para editar la interfaz gráfica, debemos abrir Matlab y en la ventana de comandos escribiremos guide. Se abrirá una venta que nos permitirá crear un archivo nuevo o abrir uno existente.

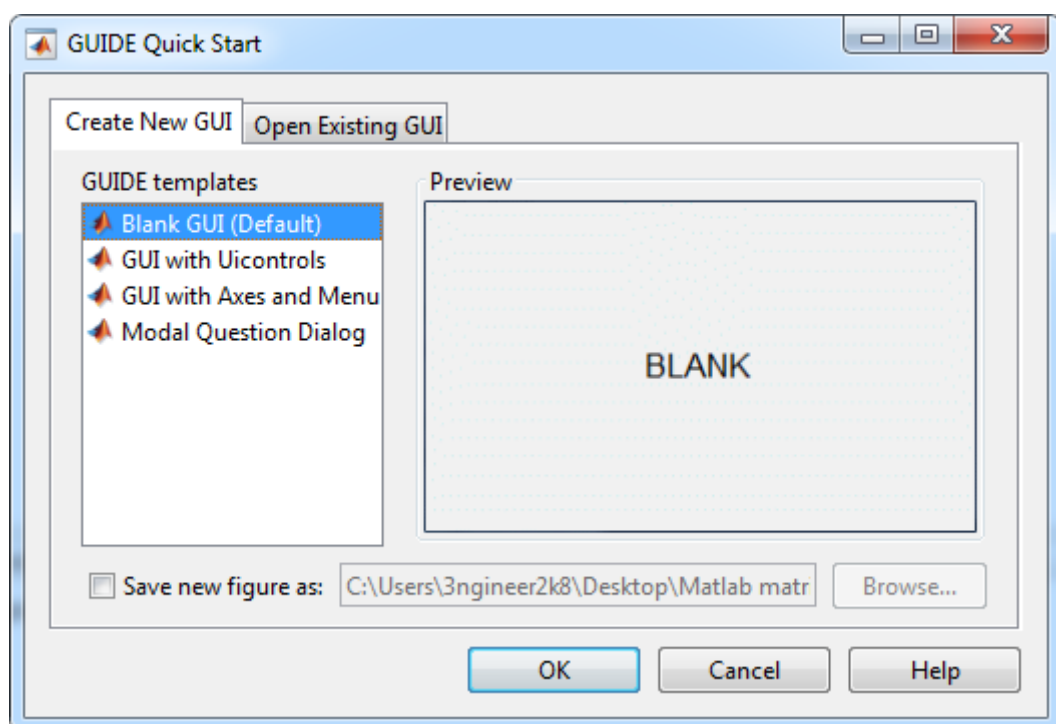


Figura 1. Ventana de inicio para interfaz gráfica de Matlab.

En la figura 2 podemos ver la ventana de edición de la interfaz gráfica de Matlab, desde allí podemos agregar o quitar elementos. Para visualizar o "correr" el programa debemos presionar el botón verde con forma de triángulo señalado por el cuadro rojo.

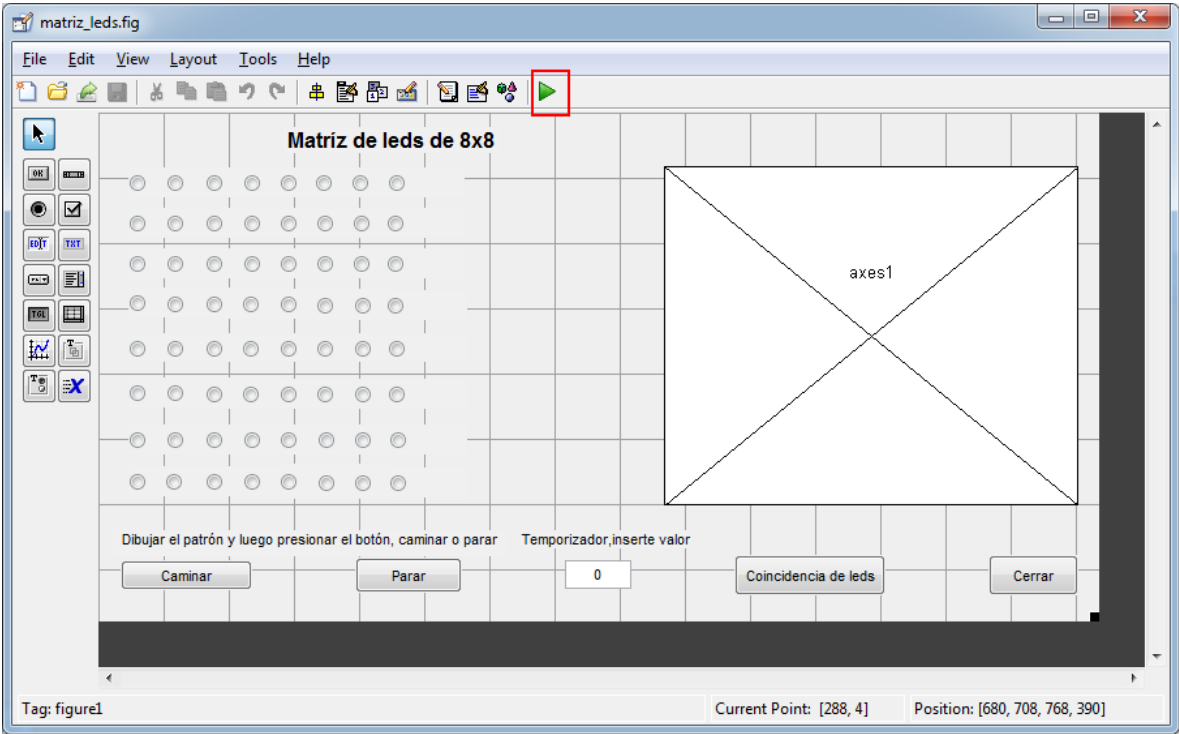


Figura 2. Editor de interfaz gráfica de Matlab.

Para agregar o modificar el código de un elemento, debemos seguir los pasos que vemos en la figura 3. Señalamos el elemento y con el click derecho desplegamos el menú.

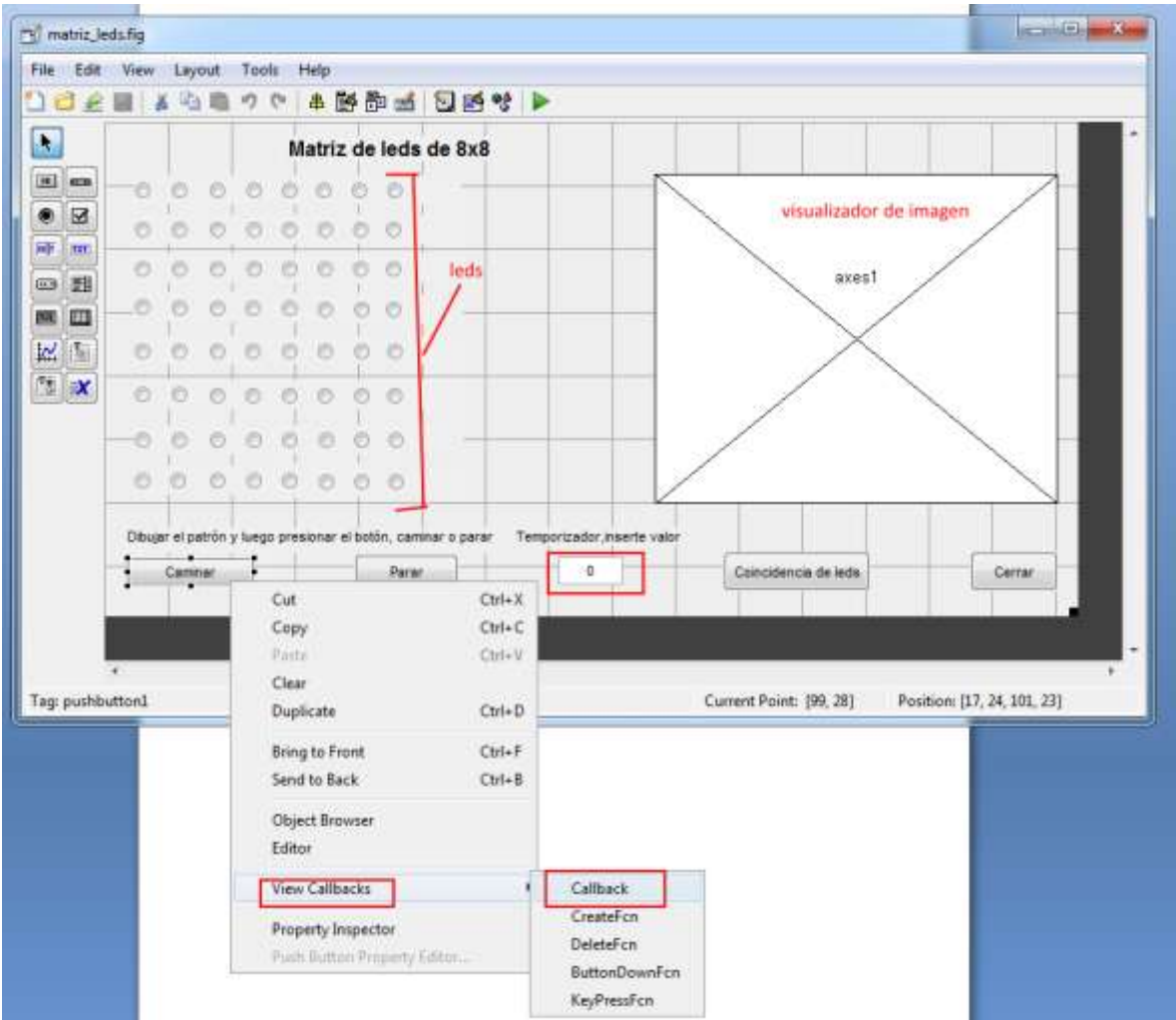


Figura 3. Agregar código a elemento.

A continuación podremos ver el código para el botón que hemos seleccionado, **Caminar**. Éste botón obtiene el estado de los leds asignados para crear la imagen Caminar.

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)

% al presionar el botón 1 (Caminar), se ejecutarán los comandos que
%siguen a continuación
% obtener el estado de cada led (radiobutton), por defecto 0
% al ser una matriz de 8x8 necesitaremos 64 botones
m11=get(handles.radiobutton1, 'Value');
m12=get(handles.radiobutton2, 'Value');
m13=get(handles.radiobutton3, 'Value');
m14=get(handles.radiobutton4, 'Value');
m15=get(handles.radiobutton5, 'Value');
m16=get(handles.radiobutton6, 'Value');
m17=get(handles.radiobutton7, 'Value');
m18=get(handles.radiobutton8, 'Value');

m21=get(handles.radiobutton9, 'Value');
m22=get(handles.radiobutton10, 'Value');
m23=get(handles.radiobutton11, 'Value');
m24=get(handles.radiobutton12, 'Value');
m25=get(handles.radiobutton13, 'Value');
m26=get(handles.radiobutton14, 'Value');
m27=get(handles.radiobutton15, 'Value');
m28=get(handles.radiobutton16, 'Value');

m31=get(handles.radiobutton17, 'Value');
m32=get(handles.radiobutton18, 'Value');
m33=get(handles.radiobutton19, 'Value');
m34=get(handles.radiobutton20, 'Value');
m35=get(handles.radiobutton21, 'Value');
m36=get(handles.radiobutton22, 'Value');
m37=get(handles.radiobutton23, 'Value');
m38=get(handles.radiobutton24, 'Value');

m41=get(handles.radiobutton25, 'Value');
m42=get(handles.radiobutton26, 'Value');
m43=get(handles.radiobutton27, 'Value');
m44=get(handles.radiobutton28, 'Value');
m45=get(handles.radiobutton29, 'Value');
m46=get(handles.radiobutton30, 'Value');
m47=get(handles.radiobutton31, 'Value');
m48=get(handles.radiobutton32, 'Value');

m51=get(handles.radiobutton33, 'Value');
m52=get(handles.radiobutton34, 'Value');
m53=get(handles.radiobutton35, 'Value');
m54=get(handles.radiobutton36, 'Value');
m55=get(handles.radiobutton37, 'Value');
m56=get(handles.radiobutton38, 'Value');
m57=get(handles.radiobutton39, 'Value');
m58=get(handles.radiobutton40, 'Value');

m61=get(handles.radiobutton41, 'Value');
m62=get(handles.radiobutton42, 'Value');
m63=get(handles.radiobutton43, 'Value');
m64=get(handles.radiobutton44, 'Value');
m65=get(handles.radiobutton45, 'Value');
m66=get(handles.radiobutton46, 'Value');
m67=get(handles.radiobutton47, 'Value');
m68=get(handles.radiobutton48, 'Value');

m71=get(handles.radiobutton49, 'Value');
m72=get(handles.radiobutton50, 'Value');
m73=get(handles.radiobutton51, 'Value');
m74=get(handles.radiobutton52, 'Value');
m75=get(handles.radiobutton53, 'Value');
```

```

m76=get(handles.radiobutton54,'Value');
m77=get(handles.radiobutton55,'Value');
m78=get(handles.radiobutton56,'Value');

m81=get(handles.radiobutton57,'Value');
m82=get(handles.radiobutton58,'Value');
m83=get(handles.radiobutton59,'Value');
m84=get(handles.radiobutton60,'Value');
m85=get(handles.radiobutton61,'Value');
m86=get(handles.radiobutton62,'Value');
m87=get(handles.radiobutton63,'Value');
m88=get(handles.radiobutton64,'Value');

% se crea la matriz con los estados de los leds(radiobuttons)y se
% visualiza en la ventana de comandos de Matlab

MLED=[m11 m12 m13 m14 m15 m16 m17 m18;
      m21 m22 m23 m24 m25 m26 m27 m28;
      m31 m32 m33 m34 m35 m36 m37 m38;
      m41 m42 m43 m44 m45 m46 m47 m48;
      m51 m52 m53 m54 m55 m56 m57 m58;
      m61 m62 m63 m64 m65 m66 m67 m68;
      m71 m72 m73 m74 m75 m76 m77 m78;
      m81 m82 m83 m84 m85 m86 m87 m88]

global walk %asignamos la variables global para usarla en
            %otra parte del programa
walk=MLED; %cambiamos el nombre de variable
caminar=im2bw(walk); %convierte imagen a blanco y negro 0 negro, 1
$blanco
imshow(caminar) %muestra la imagen

saveas(gcf,'caminar.png') %guardamos imagen de los leds activados para
caminar para luego usarla con el temporizador

```

Hacemos lo mismo para ver el código del botón **Parar**. Éste botón obtiene el estado de los leds asignados para crear la imagen Parar.

```

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)

% obtener el estado de cada led, por defecto 0

m11=get(handles.radiobutton1,'Value');
m12=get(handles.radiobutton2,'Value');
m13=get(handles.radiobutton3,'Value');
m14=get(handles.radiobutton4,'Value');
m15=get(handles.radiobutton5,'Value');
m16=get(handles.radiobutton6,'Value');
m17=get(handles.radiobutton7,'Value');
m18=get(handles.radiobutton8,'Value');

m21=get(handles.radiobutton9,'Value');
m22=get(handles.radiobutton10,'Value');
m23=get(handles.radiobutton11,'Value');
m24=get(handles.radiobutton12,'Value');
m25=get(handles.radiobutton13,'Value');
m26=get(handles.radiobutton14,'Value');
m27=get(handles.radiobutton15,'Value');
m28=get(handles.radiobutton16,'Value');

m31=get(handles.radiobutton17,'Value');
m32=get(handles.radiobutton18,'Value');
m33=get(handles.radiobutton19,'Value');
m34=get(handles.radiobutton20,'Value');
m35=get(handles.radiobutton21,'Value');
m36=get(handles.radiobutton22,'Value');
m37=get(handles.radiobutton23,'Value');
m38=get(handles.radiobutton24,'Value');

```

```

m41=get(handles.radiobutton25,'Value');
m42=get(handles.radiobutton26,'Value');
m43=get(handles.radiobutton27,'Value');
m44=get(handles.radiobutton28,'Value');
m45=get(handles.radiobutton29,'Value');
m46=get(handles.radiobutton30,'Value');
m47=get(handles.radiobutton31,'Value');
m48=get(handles.radiobutton32,'Value');

m51=get(handles.radiobutton33,'Value');
m52=get(handles.radiobutton34,'Value');
m53=get(handles.radiobutton35,'Value');
m54=get(handles.radiobutton36,'Value');
m55=get(handles.radiobutton37,'Value');
m56=get(handles.radiobutton38,'Value');
m57=get(handles.radiobutton39,'Value');
m58=get(handles.radiobutton40,'Value');

m61=get(handles.radiobutton41,'Value');
m62=get(handles.radiobutton42,'Value');
m63=get(handles.radiobutton43,'Value');
m64=get(handles.radiobutton44,'Value');
m65=get(handles.radiobutton45,'Value');
m66=get(handles.radiobutton46,'Value');
m67=get(handles.radiobutton47,'Value');
m68=get(handles.radiobutton48,'Value');

m71=get(handles.radiobutton49,'Value');
m72=get(handles.radiobutton50,'Value');
m73=get(handles.radiobutton51,'Value');
m74=get(handles.radiobutton52,'Value');
m75=get(handles.radiobutton53,'Value');
m76=get(handles.radiobutton54,'Value');
m77=get(handles.radiobutton55,'Value');
m78=get(handles.radiobutton56,'Value');

m81=get(handles.radiobutton57,'Value');
m82=get(handles.radiobutton58,'Value');
m83=get(handles.radiobutton59,'Value');
m84=get(handles.radiobutton60,'Value');
m85=get(handles.radiobutton61,'Value');
m86=get(handles.radiobutton62,'Value');
m87=get(handles.radiobutton63,'Value');
m88=get(handles.radiobutton64,'Value');

% se crea la matriz con los estados de los leds(radiobuttons) y se
% visualiza en la ventana de comandos de Matlab

MLED=[m11 m12 m13 m14 m15 m16 m17 m18;
      m21 m22 m23 m24 m25 m26 m27 m28;
      m31 m32 m33 m34 m35 m36 m37 m38;
      m41 m42 m43 m44 m45 m46 m47 m48;
      m51 m52 m53 m54 m55 m56 m57 m58;
      m61 m62 m63 m64 m65 m66 m67 m68;
      m71 m72 m73 m74 m75 m76 m77 m78;
      m81 m82 m83 m84 m85 m86 m87 m88]

global stop
stop=MLED;
parar=im2bw(stop);
imshow(parar)

saveas(gcf,'parar.png') %guardamos imagen de los leds activados para
parar

```

En la figura 3 podemos ver una caja para ingresar el valor del **temporizador**, después de ingresar el valor se debe presionar la tecla ENTER. El tiempo se

ingresa en segundos. A continuación vemos el código para la caja de texto o caja para el valor del temporizador.

```
function edit1_Callback(hObject, eventdata, handles)

%código de la caja de texto para introducir el tiempo
val=get(hObject,'String'); %Almacenar valor ingresado en la caja
tiempo=str2num(val);      %Transformar a formato numero
handles.edit1=tiempo;     %Almacenamos el valor ingresado en la caja con
el nombre tiempo
guidata(hObject,handles); %Salvar datos de la aplicación

C=imread('caminar.png','png'); %guardamos en C la imagen caminar.png
P=imread('parar.png','png');   %guardamos en P la imagen parar.png
figure,image(C)                %abrimos una figura con la imagen
%caminar.png
axis off                       %quitamos los ejes

pause(tiempo)                  %permite asignar un tiempo de espera
image(P),axis off              %abrimos imagen para parar.png,quitamos ejes
```

En un apartado de la actividad nos solicitan mostrar una matriz que contenga los leds que coincidan para formar la imagen de Caminar y parar. El código que se usó para solucionarlo está dentro del botón **Coincidencia de leds**, a continuación vemos el código.

```
% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)

%matriz que muestra los valores o leds encendidos que coinciden para
%caminar y parar

global walk stop %utilizamos las variables creadas anteriormente
coin=and(walk,stop) %matriz con valores coincidentes
```

El botón **Cerrar**, sólo cumple la función de cerrar la interfaz gráfica que hemos hecho. A continuación vemos el código:

```
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)

close all % cerrar todo
```

Ahora, para iniciar el programa desde Matlab, debemos abrir y ejecutar el archivo **matriz_leds_lanzador.m**. Lo que contiene éste archivo es, la versión de Matlab que utilizamos y comandos para cerrar ventanas abiertas y limpiar ventana de comandos. Esto con el fin de que funcione correctamente el programa. Al ejecutarse se inicia la interfaz gráfica que hemos hecho. Ver figura 4.

```
% Matlab R2018b
% Matriz de leds
close all, clear all,clc
matriz_leds
fprintf('Matriz de leds de 8x8')
```

Hemos puesto aquí el código más importante para el desarrollo de la actividad. Se adjuntarán los ficheros completos que contienen todo el código. Hay elementos que no contienen ningún código debido a que no se utilizaron.

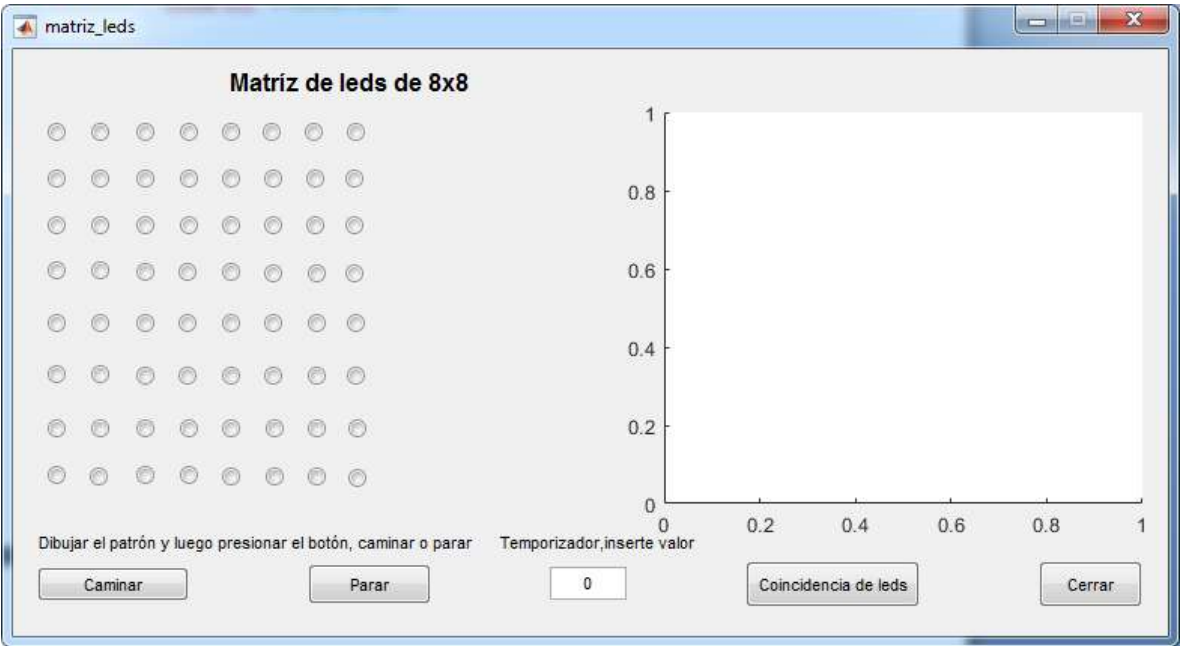


Figura 4. Interfaz gráfica

El funcionamiento de la interfaz gráfica es como se describe a continuación. Queremos mostrar una imagen que indique cuando caminar y otra para cuando parar. Para hacer esto, primero debemos activar los leds que deseemos y por último dar click en el botón Caminar. Para hacerlo sencillo, escogeremos la letra C para caminar y la letra P para parar.



Figura 5. Letra C que indica caminar.

En la Figura 5 podemos ver que se ha creado la imagen a partir de los leds que hemos activado. Los leds activados son 1 lógico y los leds apagados 0 lógico. En la imagen se ven los leds activados de color blanco y los leds apagados de color negro. Esto está definido por el comando im2bw que permite visualizar la matriz de leds partiendo de estados lógicos. Además de visualizar la imagen, internamente se guarda la imagen en un archivo para utilizarla en el temporizador. Las imágenes se llamarán caminar.png y parar.png. Cada vez que se quiera crear una imagen, se deben seleccionar los leds y luego darle click al botón (Caminar o Parar).

Para la siguiente imagen, debemos activar los botones que deseemos para formar la letra P (parar), quedando de la siguiente forma:

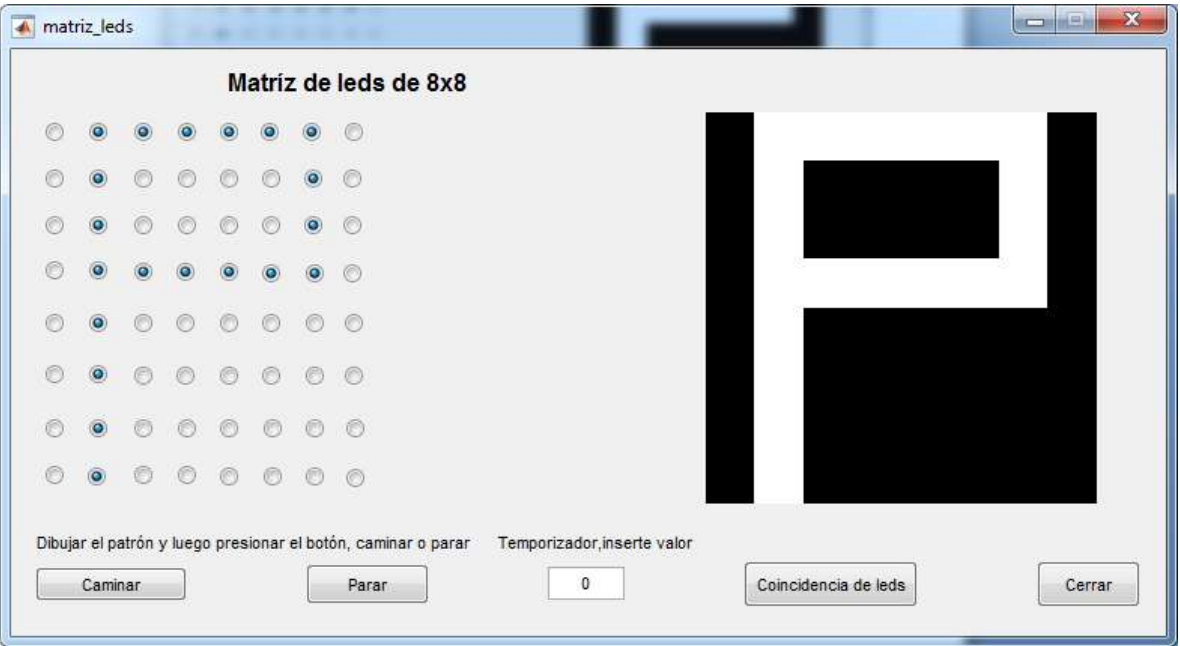


Figura 6. Letra P que indica parar.

Ahora, debemos hacer uso del temporizador para visualizar primero la imagen de Caminar y luego la de parar. El tiempo de duración entre las imágenes lo asignaremos por medio de la caja de texto. Ingresamos el valor y luego presionamos ENTER. Asignamos un tiempo de 4 segundos y se abrirá la imagen de Caminar primero, después aparecerá la imagen para Parar.

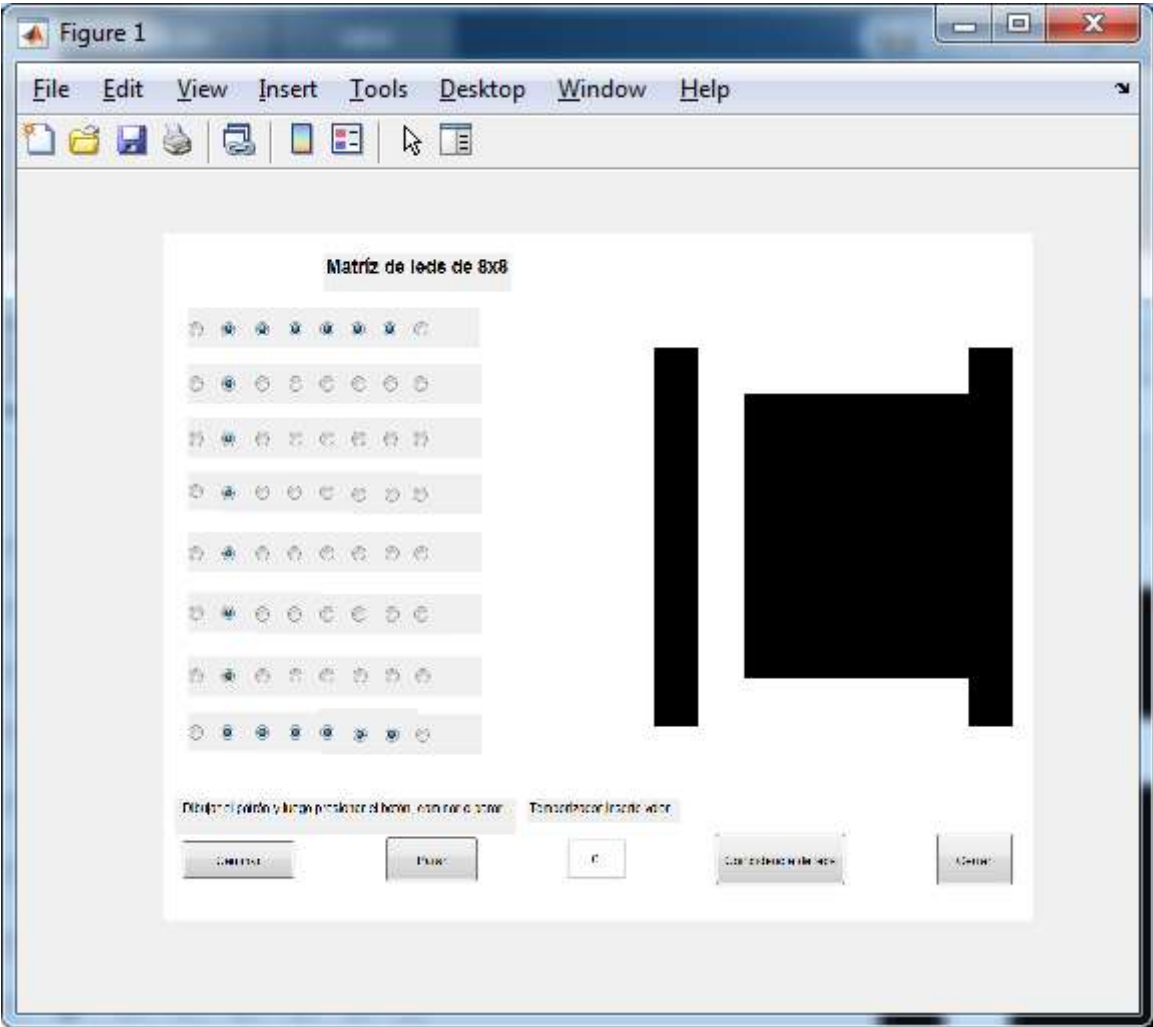


Figura 7. Imagen para Caminar, iniciada con el temporizador.

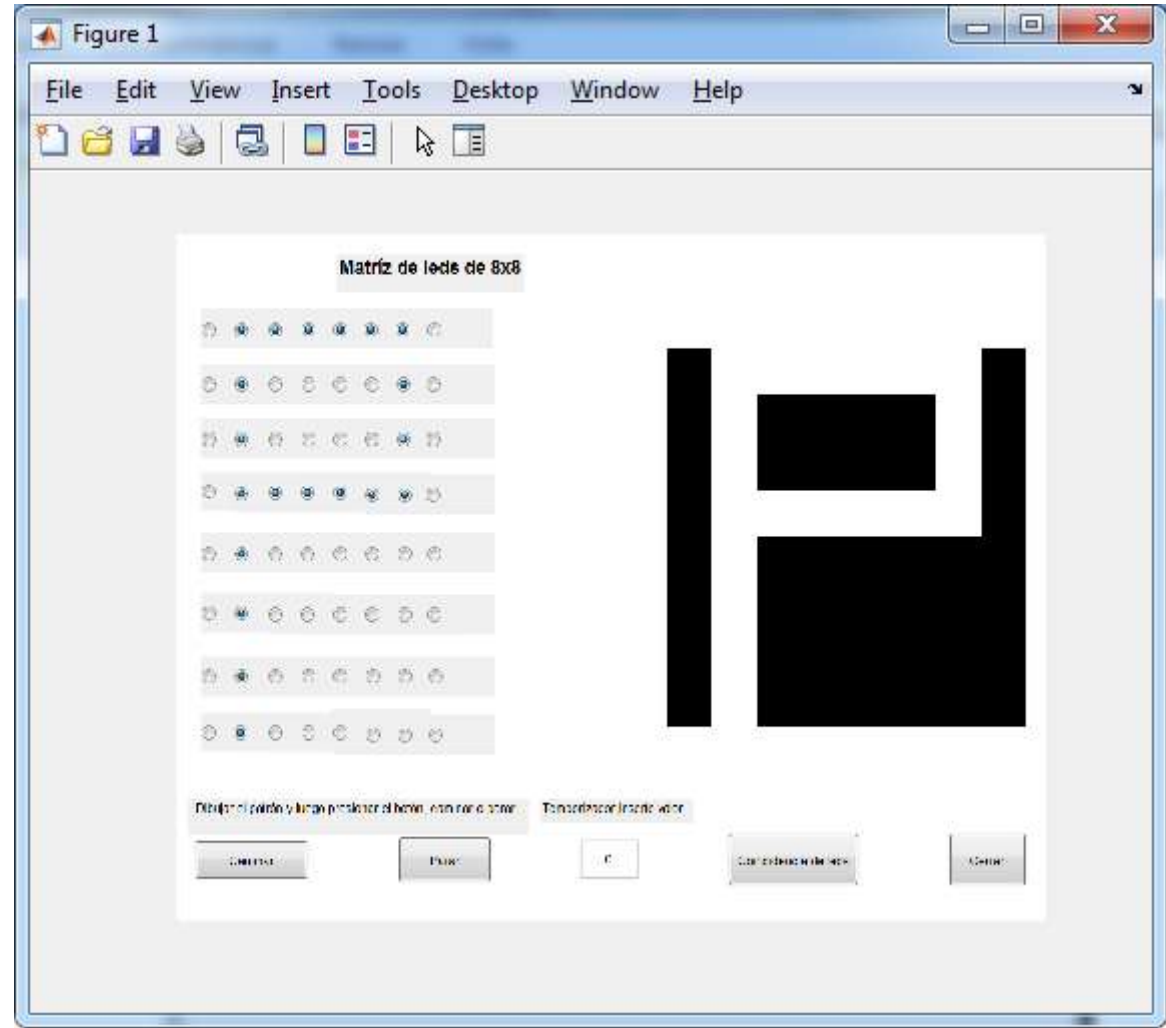


Figura 8. Imagen para Parar, aparece después de la imagen de Caminar.

Cada vez que utilizamos los botones para crear las imágenes de Caminar y Parar, podemos ver en la ventana de comandos de Matlab las respectivas matrices de 1 y 0 para Caminar y Parar. Además, hemos creado una matriz walk (caminar en inglés) y stop (parar en inglés) para poder encontrar los leds que coincidan para crear las imágenes dichas imágenes.

Como debemos crear una matriz que visualice los leds que coincidan, debemos asignar como variable global las matrices walk y stop para poderlas usar en un botón extra llamado **Coincidencia de leds**.

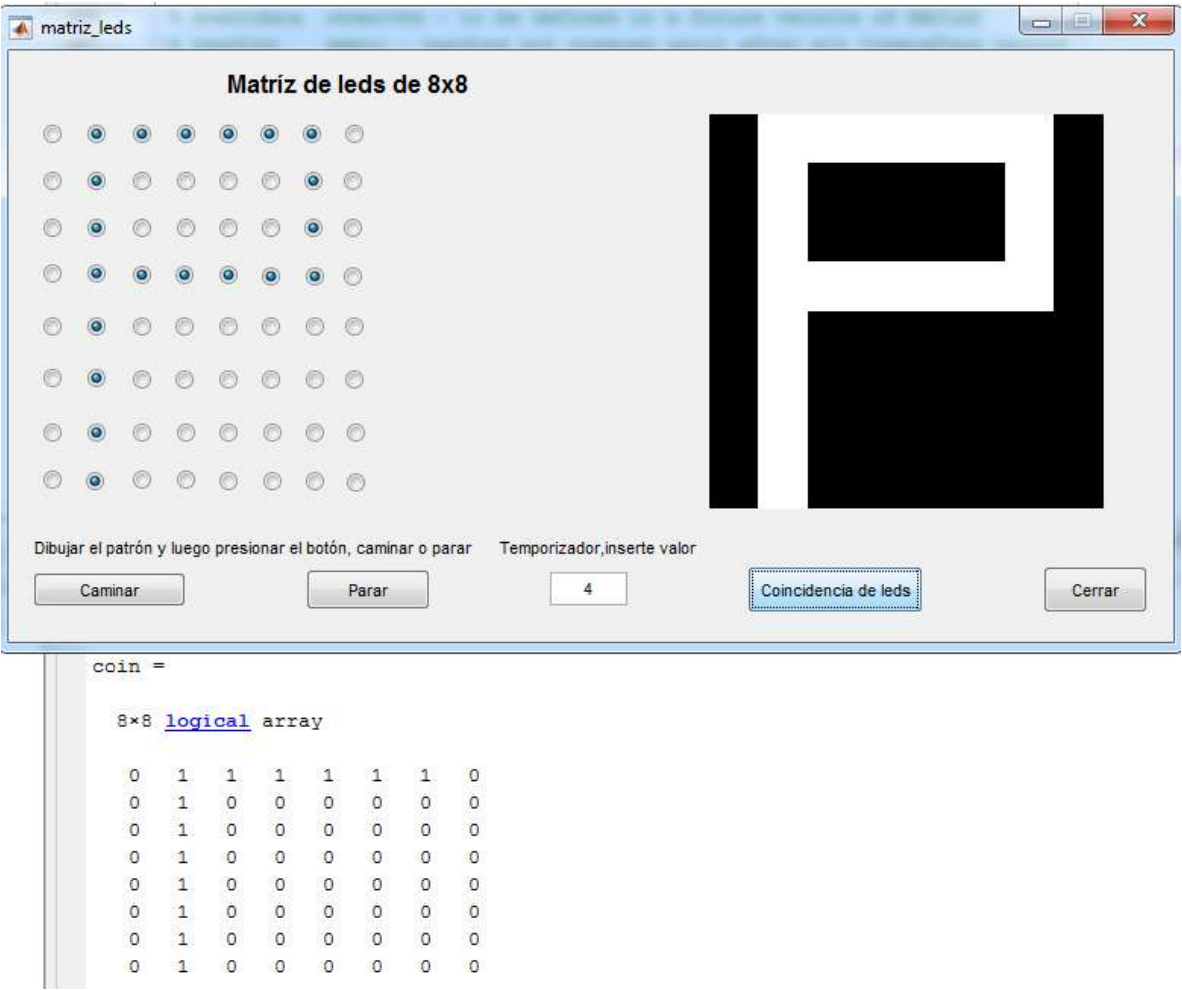


Figura 9. Matriz con leds coincidentes.

CONCLUSIONES

Ésta actividad nos ha permitido conocer más de cerca la interfaz gráfica de Matlab y su utilidad. Basados en la documentación encontrada en internet sobre guide de Matlab, hemos podido encontrar una solución al problema, más no es la única solución. Existen una gran variedad de software para este propósito, pero para ingeniería y con todo el contenido y ayuda que trae Matlab, además de guías en internet, hace a este software una gran herramienta.

Hemos decidido asignar una matriz fija de 8×8 , ya que era el mínimo permitido. A partir de allí buscamos la solución para cada apartado de la actividad.

REFERENCIAS BIBLIOGRÁFICAS

[1] https://la.mathworks.com/help/matlab/creating_guis/about-the-simple-guide-gui-example.html

[2] https://www.dspace.espol.edu.ec/bitstream/123456789/10740/11/MATLAB_GUIDE.pdf

[3] http://catarina.udlap.mx/u_dl_a/tales/documentos/lep/garcia_b_s/capitulo3.pdf