**ClassWord 1**

Gestión de redes y comunicaciones

Facultad: Ingeniería de Sistemas

Presentado por:

Orlando Camilo Acosta Vargas

Profesor: Alexander García Pérez, Ingeniería de Sistemas

Universidad EAN

2019

**Índice**

1. **Introducción**
2. **Señal sinusoidal**

**2.1. El período (T) en una sinusoide**.

**2.2. Modelo practico en Matlab**

**2.2.1 ¿Qué es MATLAB?**

**2.2.2 Pasos de desarrollo en Matlab**

**2.2.3 Código de Señal sinusoidal en Matlab**

**2.2.4 Pasos de desarrollo interfaz gráfica de Matlab**

1. **Serie Fourier: generador de señal**
2. **Transformada de Fourier: analizador de espectro.**
3. **Modulación de simulación: analógica + digital**
4. **Bibliografía** 
   1. **Introducción**
   2. **Señal sinusoidal**

 La **Señal de onda sinusoidal representan el valor de la tención**

**de corriente alterna a través de un tiempo** continuamente variable. Está dada por la función.

Donde:

**2.1. El período (T) en una sinusoide**.

Podemos definirlo como conjunto de valores de t que corresponden a un ciclo completo de valores de la función; en este sentido toda función de una variable que repite sus valores en un ciclo completo es una función periódica.

La frecuencia se mide en Hertz (Hz). Muestra el número de ciclos de señal que tuvieron lugar en un segundo.

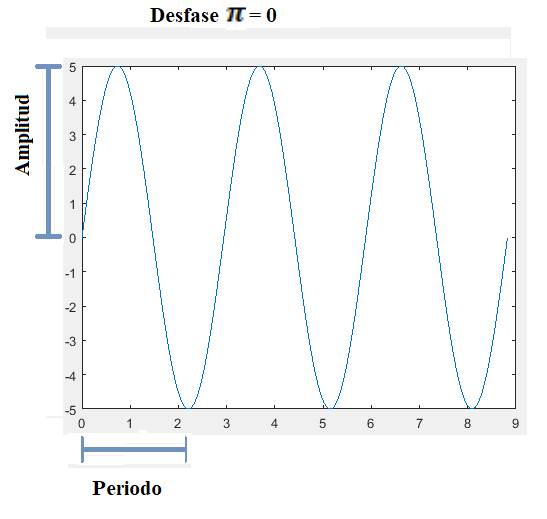
****

Figura (1) Ejemplo de señal sinusoidal

**2.2. Modelo practico en Matlab**

**2.2.1 ¿Qué es MATLAB?**

MATLAB es un lenguaje de alto desempeño diseñado para realizar cálculos técnicos.

Es un ambiente de programación fácil de utilizar donde los problemas y las soluciones se expresan en una notación matemática. MATLAB es un sistema interactivo cuyo elemento básico de datos es el arreglo que no requiere de dimensionamiento previo. Esto permite resolver muchos problemas computacionales, específicamente aquellos que involucren vectores y matrices, en un tiempo mucho menor al requerido para escribir un programa en un lenguaje escalar no interactivo tal como C o Fortran.

MATLAB se utiliza ampliamente en:

• Cálculos numéricos

• Desarrollo de algoritmos

• Modelado, simulación y prueba de prototipos

• Análisis de datos, exploración y visualización

• Graficación de datos con fines científicos o de ingeniería

• Desarrollo de aplicaciones que requieran de una interfaz gráfica de usuario (GUI, Graphical User Interface).

**(**M. C. José Jaime Esqueda Elizondo, Matlab e Interfaces Gráficas, noviembre, 2002**)**

**2.2.2 Pasos de desarrollo en Matlab**

* Inicializamos el entorno de Matlab
* Creamos un nueva Carpeta para el proyecto
* Creamos un nuevo proyecto o un nuevo Script
* Le damos un nombre al archivo y lo guardamos como un archivo .m
* Compilamos el archivo

**2.2.3 Código de Señal sinusoidal en Matlab**

%limpiamos la consola

clc

clear all

close all

%---- DATA IN --

A = 5; %% Amplitud

f = 3400; %% frecuencia

phi = 0; %% desfase

cantper = 3; %%periodo

%--- PROCESS ---

t = linspace(0,cantper\*1/f,1000);%Aquí se maneja como "T" el Periodo de nuestra señal

xt= A\*sin(2\*pi\*f\*t + phi);

%--- OUTPUT ---

%-- Graficacion --

figure(1)

plot(t, xt)

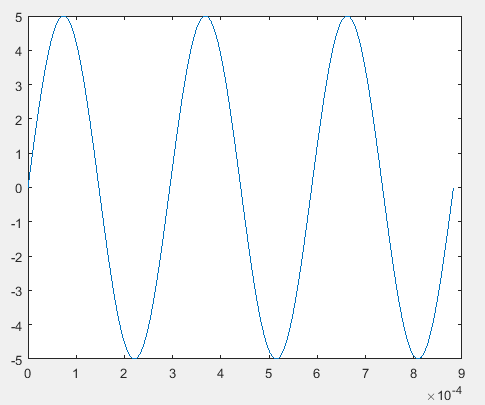


Figura (2) compilación de señal sinusoidal

**2.2.4 Pasos de desarrollo interfaz grafica de Matlab**

* Inicializamos el entorno de Matlab
* Creamos un nueva Carpeta para el proyecto
* En la línea de comandos de Matlab escribimos guide y se nos abrirá la consola de GUI como muestra en la figura 3.

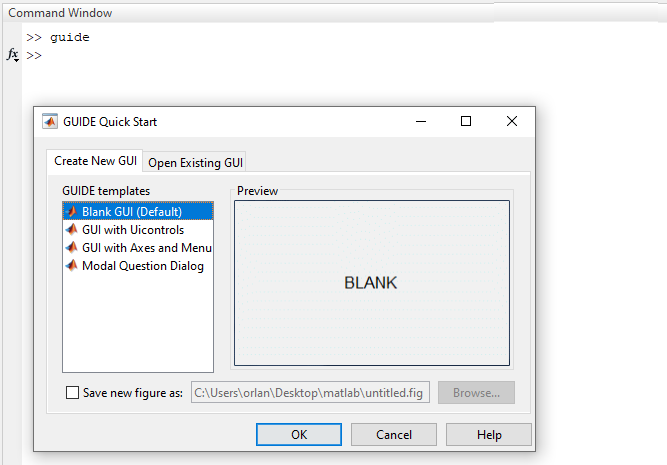


Figura (3) Inicio de interfaz GUI

* Le damos click en ok, y se nos abrirá el área de trabajo GUI como muestra en la figura (4)

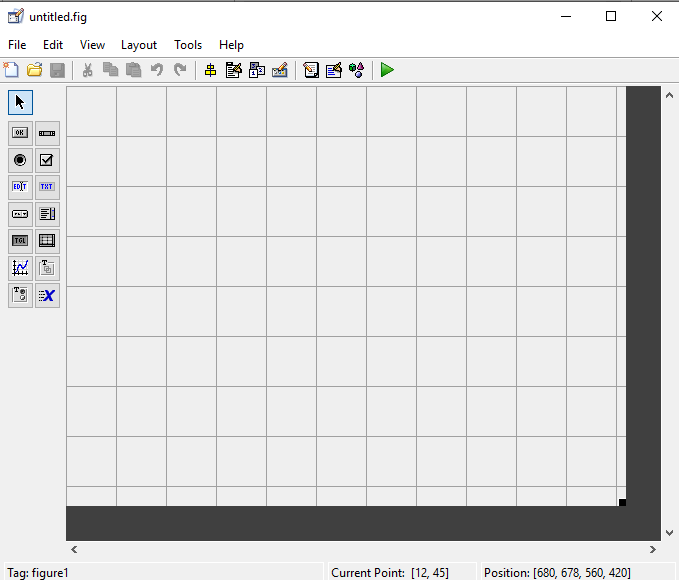


Figura (3) Entorno de trabajo GUI

* Luego le damos click derecho y seleccionamos la propiedad de inspector como muestra en la figura 4

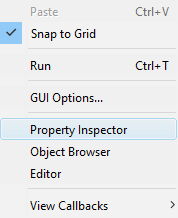


Figura (4) Inspector GUI

* De ahí se nos abrirá la barra de tareas y podremos realizar los cambios al entorno GUI.

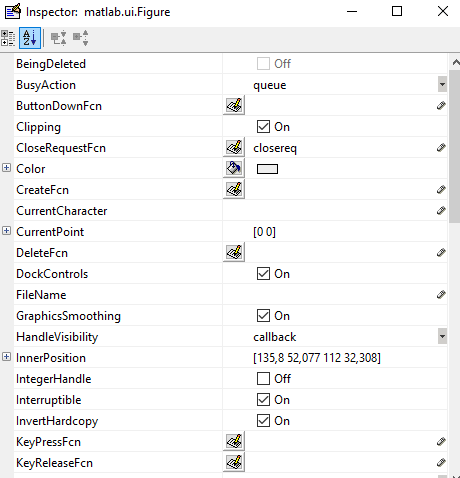


Figura (5) barra de tareas Inspector

* Si queremos le cambiamos el color de nuestra consola

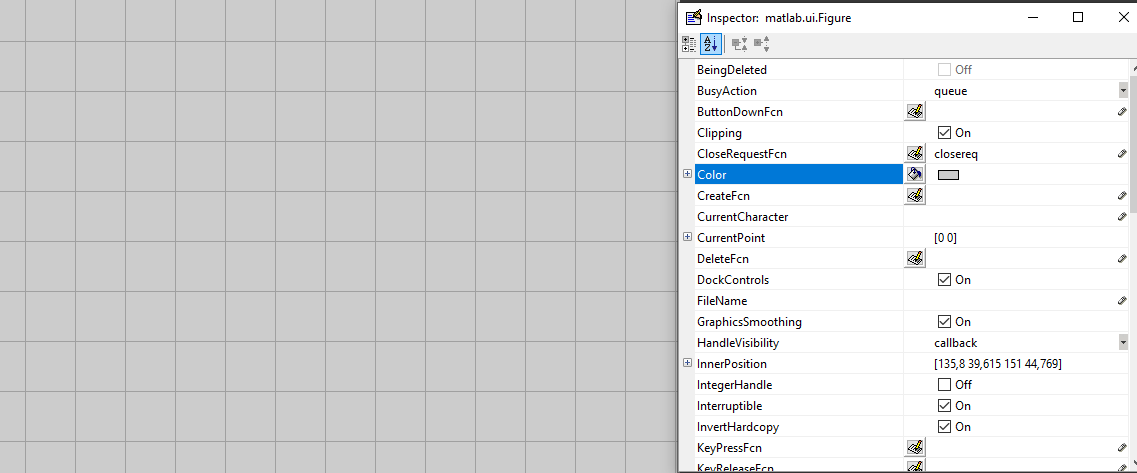


Figura (6) Cambio de color a consola

* Ahora le damos click en la barra derecha en el boton Axes y colocamos 4 de ellos en la consola

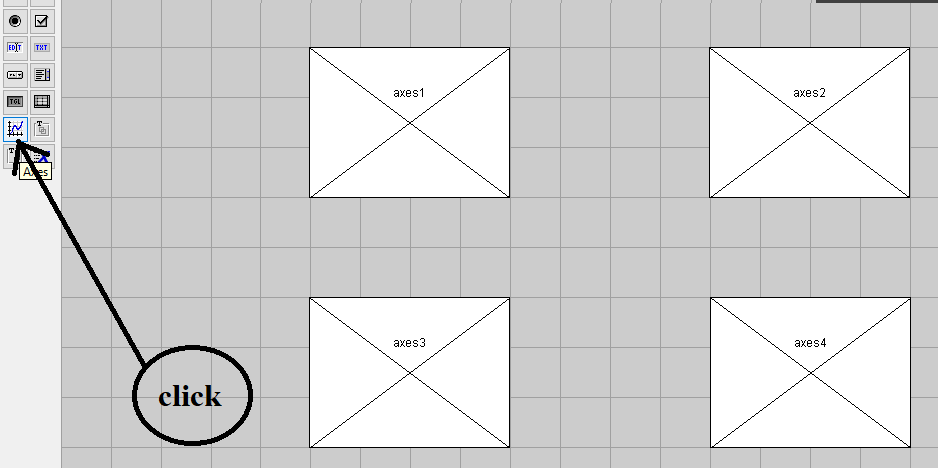
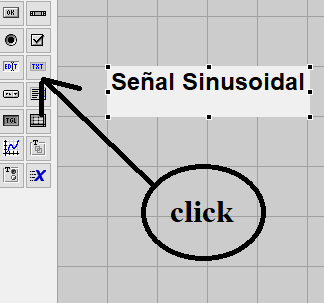


Figura (7) Axes graficador

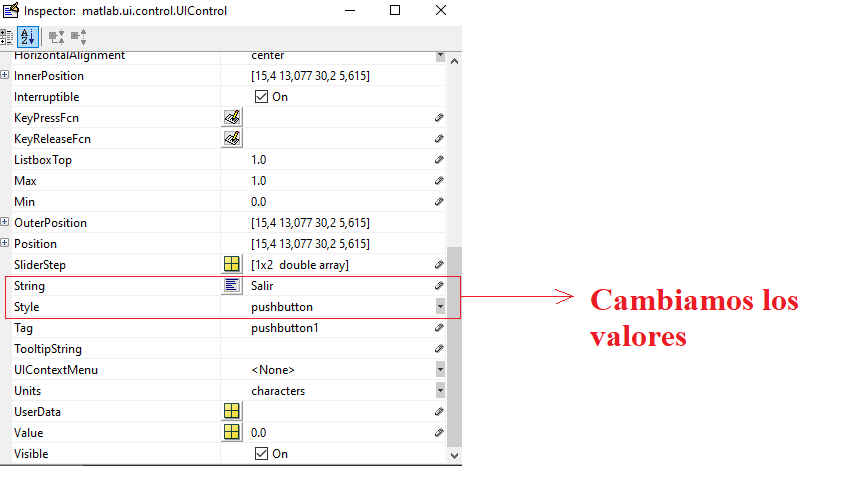
* Ahora en la misma barra hacemos click en txt y agregamos un texto, y le cambiamos el nombre



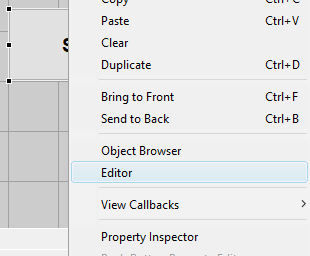
* Ahora en la misma barra le agregamos un push button a la consola



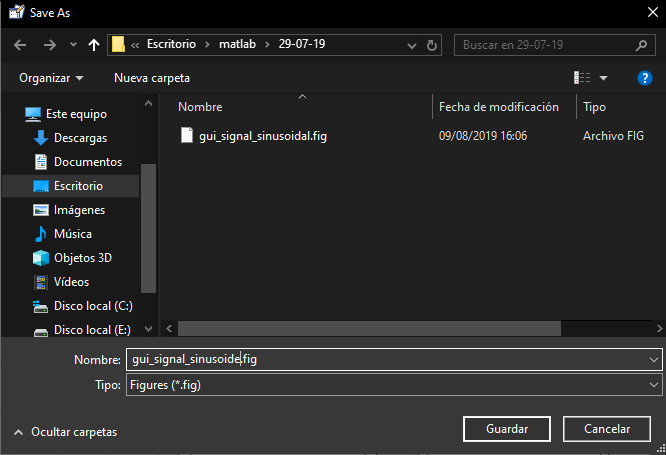
* Ahora le damos click derecho al botón, le damos click en la propiedad inspector y le cambiamos los valores String = salir , style = pushButon



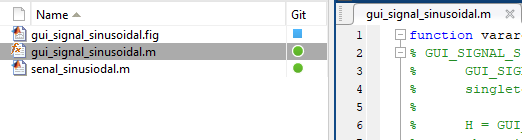
* Realizados los cambios de damos click derecho a el botón y seleccionamos editor



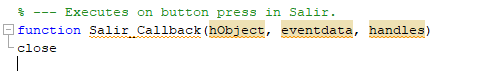
* El cual nos mandara a guarguar el archivo como .fig



* Una ves generado el achivo .fig abriremos el archivo .m



* Buscamos el nombre de la función que le dimos a nuestro botón y le colocamos el código que necesita



* Ahora en nuestra consola GUI creamos otro botón llamado generar y realizamos el mismo procedimiento



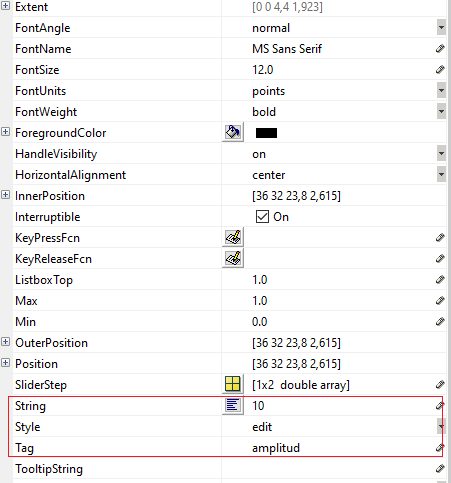
* Una vez ya hecho colocamos 3 otros formatos de textos en nuestro archivo GUI y los llamados Amplitud, frecuencia y periodo



* Luego procedemos a seleccionar el botón edit text y creamos tres iconos para cada uno de ellos.



* Luego procedemos a cambiarle los valores a este botón texto, dándole click derecho, seleccionamos propiedad inspector y cambiamos los valores de la siguiente manera como muestra en la figura



* Esto lo realizamos con cada uno de los botones texto
* Ahora configuramos nuestro archivo en .m en nuestra función

function generar\_Callback(hObject, eventdata, handles)

clc

%------- DATA IN -------%

A= str2num(get(handles.amplitud,'String'))

f = str2num(get(handles.frecuencia,'String'))

cantper = str2num(get(handles.periodo,'String'))

phi = pi;

phi1 = 0;

phi2 = pi/2;

phi3 = pi;

phi4 = 3\*pi/2;

%------- PROCESS -------%

t = linspace(0,cantper\*1/f,500);

xt1 = A\*sin(2\*pi\*f\*t + phi1);

xt2 = A\*sin(2\*pi\*f\*t + phi2);

xt3 = A\*sin(2\*pi\*f\*t + phi3);

xt4 = A\*sin(2\*pi\*f\*t + phi4);

%------- OUTPUT ------%

%--figure(1)--%

axes(handles.axes1)

plot(t,xt1),title('phi=0'),grid on

axes(handles.axes2)

plot(t,xt2),title('phi=pi/2'),grid on

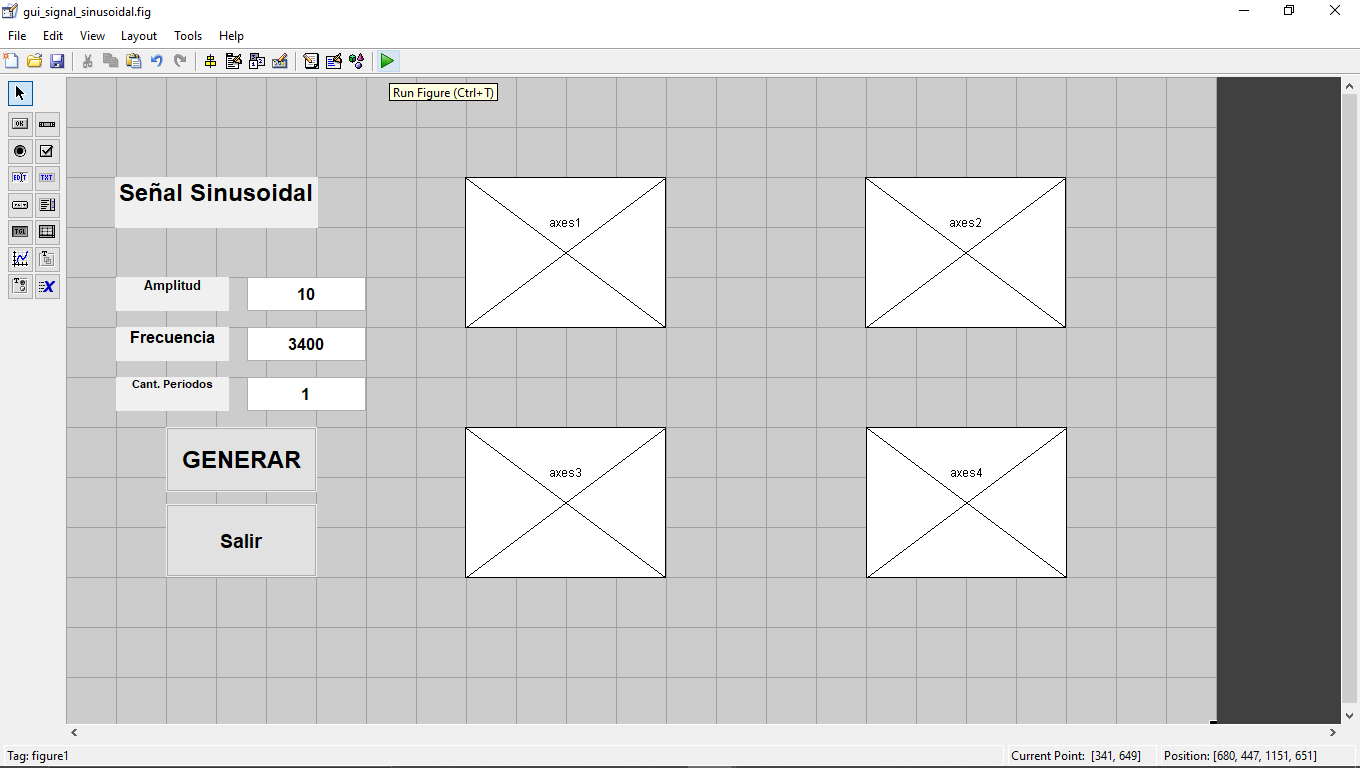
axes(handles.axes3)

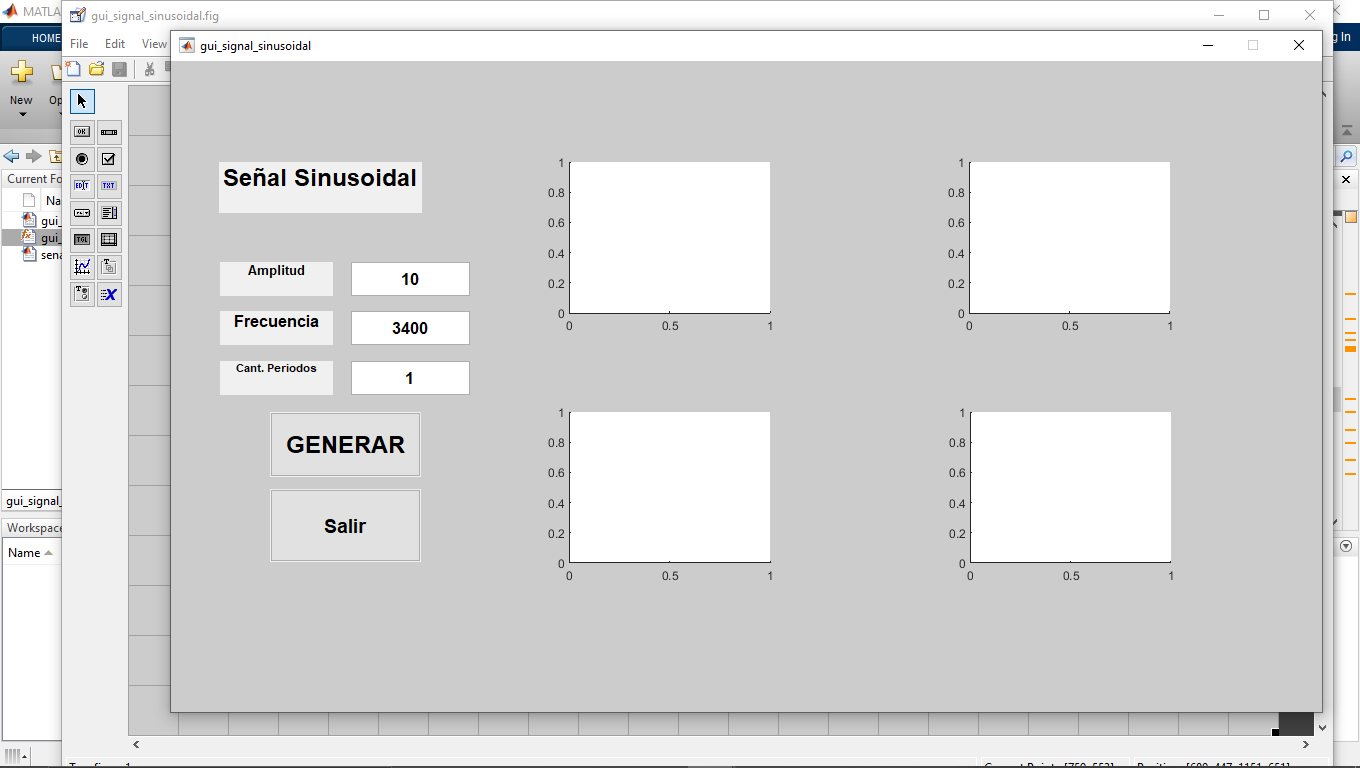
plot(t,xt3),title('phi=pi'),grid on

axes(handles.axes4)

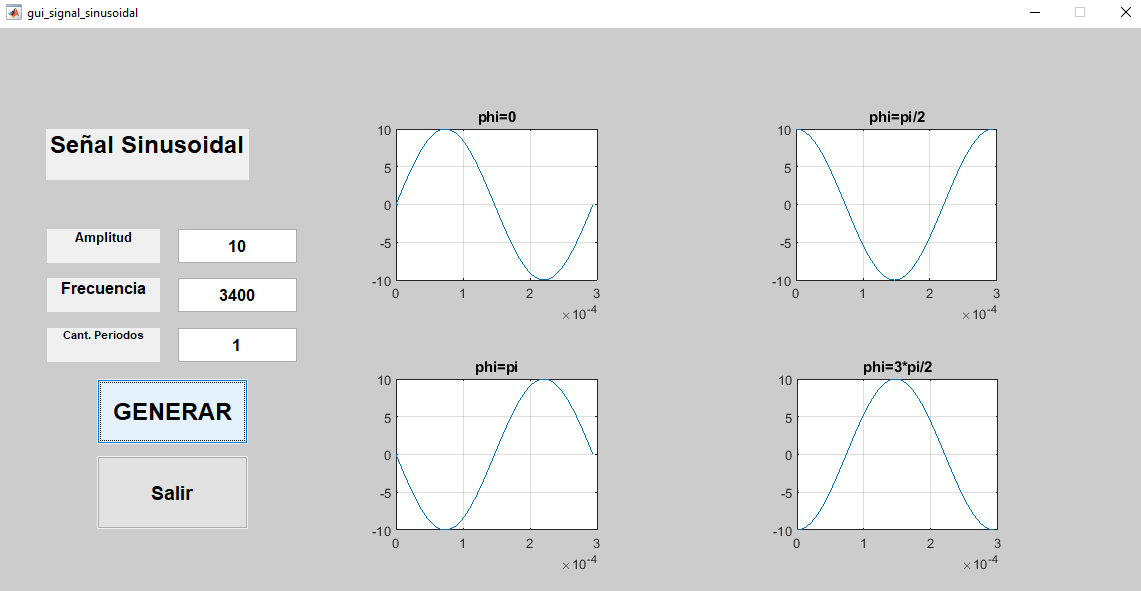
plot(t,xt4),title('phi=3\*pi/2'),grid on

* Por último, compilamos nuestro archivo en GUI como se muestra en la figuras





* Una vez realizado esto le damos click en generar y obtendremos nuestra señal sinusoidal con cada una de las graficas de desfase



* 1. **Serie Fourier: generador de señal**
  2. **Transformada de Fourier: analizador de espectro.**
  3. **Modulación de simulación: analógica + digital**
  4. **Bibliografía**

Stallings,William. Comunicaciones y redes de computadores. Prentice Hall, 2004.Séptima Edición.

Jesús Monzo Espín, Function Sinusoidal, Retrieved August 6, 2019, from <https://www.geogebra.org/m/eAkQEpyP>

Riverglennapts.com. Fundamentos sinusoidal, Retrieved August 6, 2019, from <https://riverglennapts.com/es/fundamentals/432-sinusoidal-wave-signal.html>

M. C. José Jaime Esqueda Elizondo, Matlab e Interfaces Gráficas, noviembre, 2002