



Instituto Tecnológico Superior de San Andrés  
Tuxtla

Ingeniería Mecatrónica

**Investigación:**

Interfaces Graficas de Usuario. GUI

**Docente:**

Mtro. Roberto Esteban Guerrero Hernández

**Presenta:**

BRIANA PAOLA MIGUELES LOPEZ

Grupo: 311 B

San Andrés Tuxtla, 2024

Los primeros conceptos de interfaz gráfica de usuario se desarrollaron en los años 70 en la empresa Xerox. Su propósito principal era permitir manejar ordenadores con el ratón y el teclado en lugar de solo con instrucciones en formato de texto. **Xerox Alto** fue el primer PC con una interfaz gráfica. Apple le siguió en los años 80 con el Macintosh. Con la aparición de los *smartphones* y las tabletas, el principio de la interfaz gráfica ha pasado por grandes cambios. Hoy en día, hay pantallas que se pueden utilizar con simples gestos y movimientos de dedos.

### **Definición: ¿qué es una GUI?**

La GUI es una interfaz entre la persona y la máquina. El objetivo de esta interfaz gráfica es representar el código del *backend* de un sistema **de la forma más clara posible para el usuario** para simplificarle las tareas diarias. Para esto, son muy importantes los iconos y las imágenes, ya que solo estos permiten una aplicación universal e independiente del texto. Por ejemplo, casi todo el mundo sabe cómo es un icono de wifi, mientras que la palabra varía mucho en los distintos idiomas.

### **Definición**

Una *graphical user interface* o GUI es una interfaz que se puede utilizar para controlar PC, tabletas y otros dispositivos. Las GUI utilizan elementos gráficos como iconos, menús e imágenes para facilitar el manejo del usuario humano. Tanto los sistemas operativos como las aplicaciones utilizan una interfaz gráfica de usuario. De hecho, casi todos los programas para usuarios finales de hoy en día vienen con esta interfaz.

¿Cómo funciona una GUI?

La GUI es una **interfaz de usuario** que permite a los usuarios comunicarse con el ordenador. Suele estar basada en la interacción a través del ratón y el teclado (aunque el control a través de gestos es cada vez más común): al mover el ratón, el puntero se desplaza en la pantalla. La señal del dispositivo se transmite al ordenador, que luego la traduce en un movimiento equivalente en la pantalla. Por ejemplo, si un usuario hace clic en un determinado icono de programa en el menú, se ejecuta la instrucción correspondiente y se abre el programa.

La GUI es, por lo tanto, una especie de **traductor** en la comunicación entre el humano y la máquina. Sin la GUI, tendrías que utilizar la línea de comandos para llamar a programas y aplicaciones. Esto podría representarse así (el ejemplo muestra cómo abrir el explorador):  
¿Cuáles son los componentes de una GUI?

Una GUI combina **el diseño visual y las funciones de programación**. Por esto, ofrece botones, menús desplegables, campos de navegación, campos de búsqueda, iconos y *widgets*. Los desarrolladores deben tener siempre en cuenta la facilidad de uso.

Los **componentes más comunes de una GUI** son los siguientes:

- Campos de entrada
- Ventana
- Lienzo (*canvas*)
- Marcos
- Botones
- Bloques de encabezado
- Campos de texto

La papelera de reciclaje es un buen ejemplo de un elemento común a la mayoría de los sistemas operativos. Además, su representación gráfica tanto en Windows como en Mac es la imagen de una papelera real. De este modo, el usuario sabe de inmediato que sirve para borrar documentos y archivos.

Al escribir las GUI, los desarrolladores se basan en la **programación basada en eventos**, porque las acciones del usuario son impredecibles. Por esto las GUI no se pueden programar linealmente, sino que deben escribirse de tal manera que un comando solo se ejecute cuando el usuario dé la señal para este.

¿Qué requisitos debe cumplir una GUI?

Una buena GUI debe ser, sobre todo, **fácil de utilizar para el usuario**. Durante la fase de diseño, es importante que la GUI se controle de manera intuitiva. No obstante, para lograrlo es necesario comprender bien las necesidades y preferencias del usuario. Un diseño orientado a la experiencia de usuario tiene el propósito de proporcionar al usuario lo que este espera de la aplicación. Si vas a desarrollar una GUI, presta atención a los siguientes aspectos:

- **Mantén la simplicidad:** se recomienda prescindir de elementos de diseño innecesarios y elegir designaciones simples y fáciles de comprender.
- **Diseña con orientación a un objetivo:** cada página debe estar bien estructurada; cada elemento debe tener una función clara.
- **Consistencia:** si se utilizan múltiples elementos y gráficos, es importante que todos los componentes individuales sean consistentes entre sí.
- **Diseño y tipografía:** las unidades de diseño, los colores y los textos deben resaltar u ocultar el elemento, dependiendo de la finalidad del componente. También es importante utilizar fuentes y tamaños de fuente apropiados para la función y fáciles de comprender.
- **Actualizaciones de usuario:** una interfaz gráfica de usuario también debe informar sobre errores y cambios en el estado.

### ¿Cuáles son las ventajas y los inconvenientes de una GUI?

Las interfaces gráficas de usuario han ganado gran popularidad por las numerosas ventajas que ofrecen a los usuarios. Sin embargo, tienen también algunos inconvenientes. En la siguiente tabla, te explicamos las ventajas y desventajas de diversas GUI.

Ventajas	Inconvenientes
Simple y fácil de usar	Menor flexibilidad: solo se pueden ejecutar instrucciones preprogramadas
Diseño visualmente atractivo	La funcionalidad del sistema no se puede cambiar ni adaptar
Incluso los usuarios con menos conocimientos técnicos pueden ejecutar aplicaciones simples con la GUI	Las GUI requieren una cantidad de memoria relativamente grande en el sistema
La representación visual facilita enormemente la búsqueda de documentos y archivos	Las GUI son más lentas que las interfaces basadas únicamente en la línea de comandos
El sistema responde a instrucciones que el usuario puede indicar de forma muy intuitiva gracias al formato visual	Para el desarrollador, es comparativamente más difícil diseñar una GUI fácil de utilizar

<b>Ventajas</b>	<b>Inconvenientes</b>
El usuario puede navegar entre varias aplicaciones de forma rápida y sencilla	Algunas aplicaciones requieren más tiempo para ejecutarse

### **En resumen**

Sin las interfaces gráficas de usuario, la revolución digital no habría sido concebible. Las GUI permiten incluso a los usuarios inexpertos familiarizarse rápidamente con los programas. Los símbolos fáciles de comprender y el texto explicativo aseguran una gran facilidad de uso.

## LISTA DE COTEJO PARA REPORTE DE PRÁCTICA

<b>INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA</b>			<b>ASIGNATURA: PROGRAMACION AVANZADA</b>	
<b>NOMBRE DEL DOCENTE: MTI. ROBERTO E. GUERRERO HERNANDEZ</b>			<b>FIRMA DEL DOCENTE</b>	
<b>DATOS GENERALES DEL PROCESO DE EVALUACIÓN</b>				
<b>NOMBRE(S) DEL ALUMNO(S):</b> MIGUELES LOPEZ BRIANA PAOLA		<b>MATRICULA:</b> 231U0386		<b>FIRMA DEL ALUMNO(S):</b>
<b>PRODUCTO: REPORTE DE PRACTICA</b>	<b>NOMBRE DE LA PRACTICA:</b> INTERFACES DE USUARIO	<b>FECHA:</b> DICIEMBRE- 2024		<b>PERIODO ESCOLAR:</b> AGOSTO – DICIEMBRE 2024
<b>INSTRUCCIONES</b>				
<p>Revisar las actividades que se solicitan y marque con una "X" en los apartados "SI" cuando la evidencia se cumple; en caso contrario marque "NO". En la columna "OBSERVACIONES" indicaciones que puedan ayudar al alumno a saber cuáles son las condiciones no cumplidas, si fuese necesario.</p>				
VALOR DEL REACTIVO	CARACTERÍSTICA PARA CUMPLIR (REACTIVO)	CUMPLE		OBSERVACIONES
		SI	NO	
3%	Presentación El trabajo cumple con los requisitos de: a. Buena presentación	X		
3%	b. No tiene faltas de ortografía	X		
4%	c. Mismo Formato (letra arial 12, títulos con negritas)		X	
4%	d. Maneja el lenguaje técnico apropiado	X		
10%	<b>Desarrollo:</b> Sigue una metodología y sustenta todos los pasos que se realizaron en el análisis y desarrollo en la aplicación de los temporizadores dentro de la programación del PLC y aplicando los conocimientos obtenidos, es analítico y bien ordenado.	X		
6%	<b>Responsabilidad:</b> Entregó la práctica en la fecha y hora señalada.	X		
30%	<b>CALIFICACIÓN</b>	27 %		

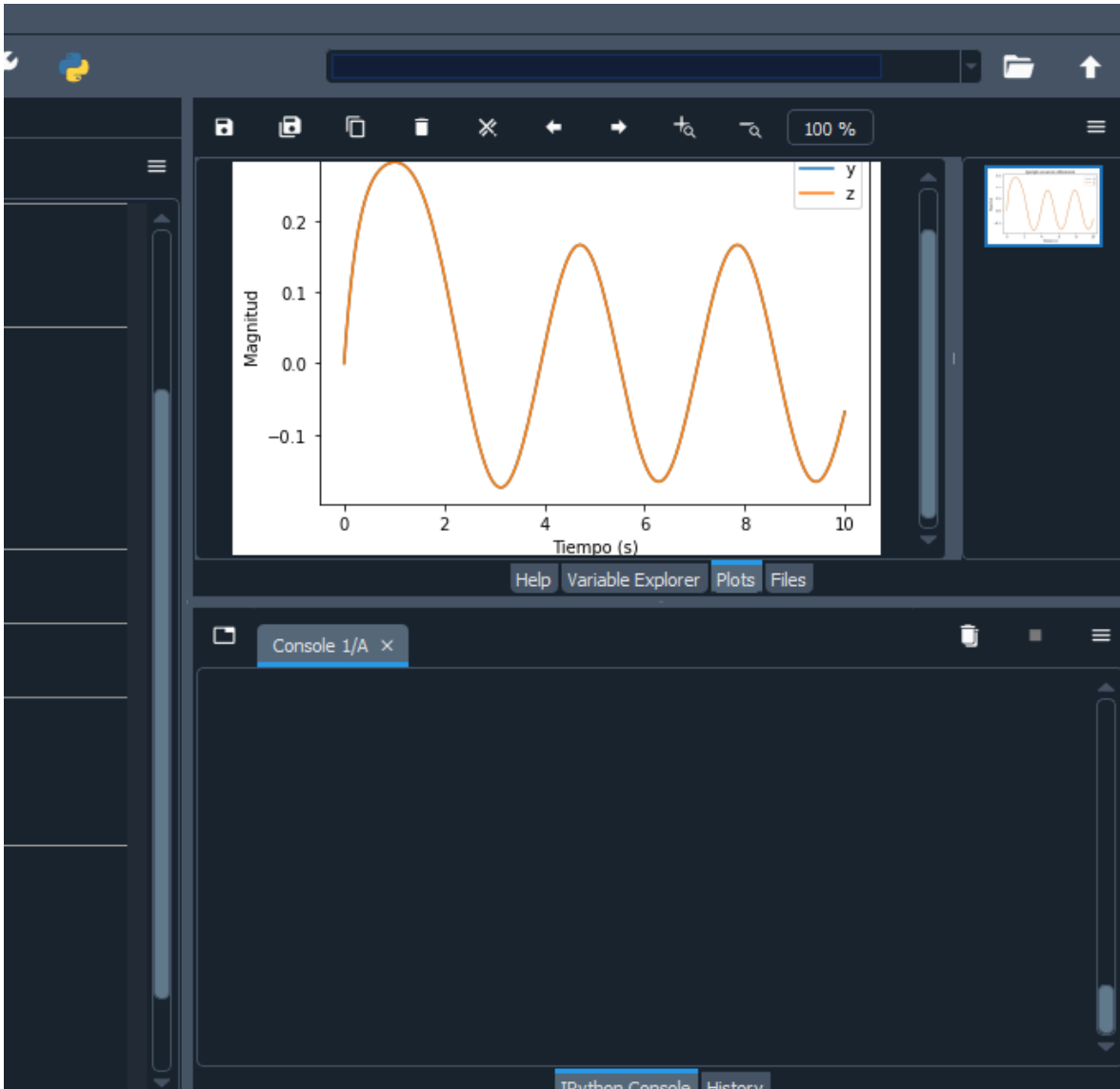


GRUPO: 311 - B ALUMNO: Briana Paola Migueles López

Realizar una práctica en donde se muestre el código de una ecuación diferencial, puedes imprimir la pantalla del IDE para ver el código y la grafica GUI.

Úl & cããÁ  
G ãã Á

```
Spyder (Python 3.11)
File Edit Search Source Run Debug Consoles Projects Tools View Help
untitled3.py* x
7
8   %% Librerias
9   import numpy as np
10  import matplotlib.pyplot as plt
11  from scipy.integrate import odeint
12
13  %% Declaracion de modelo
14
15  def Modelo(X,t):
16      y, z = X[0], X[1]
17      dXdt=[0,0]
18      dXdt[0]=z
19      dXdt[1]=-4*y-3*z+np.sin(2*t)
20      return dXdt
21
22  %% Condiciones iniciales
23  X0 = [0,1]
24
25  %% valores de tiempo
26  t=np.linspace(0,10,200)
27
28  %% Resolviendo la ODE
29
30  X = odeint(Modelo,X0,t)
31  y = X[:,0]
32  z = X[:,1]
33
34  %% Plotear resultados
35
36  ax = plt.figure().add_subplot(111)
37  ax.plot(t,y,linewidth = 1.5,label = "y")
38  ax.plot(t,z,linewidth = 1.5,label = "z")
39  ax.set_xlabel('Tiempo (s)')
40  ax.set_ylabel('Magnitud')
41  ax.set_title('Ejemplo ecuacion diferencial')
42  ax.legend(loc='best')
```







INSITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA  
ING. MECATRONICA  
PROGRAMACION AVANZADA  
EXAMEN



GRUPO: 311B ALUMNO: BRIANA PAOLA MIGUELES LOPEZ

1.- Realiza un programa en lenguaje Python, en donde se muestre el código y la gráfica para resolver una ecuación diferencial. Puedes copiar el código que usaste en el IDE para programarlo y pegas la pantalla resultante del código compilado.

```
### Librerias
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

### Declaracion de modelo

def Modelo(X,t):
    y, z = X[0], X[1]
    g, L = 9.81, 1
    dXdt=[0,0]
    dXdt[0]=z
    dXdt[1]=-g*np.sin(y)/L
    return dXdt

### Condiciones iniciales
X0 = [0.9*np.pi,0]

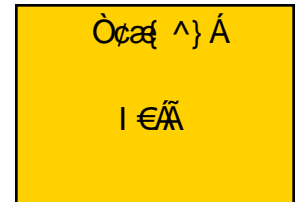
### valores de tiempo
t=np.linspace(0,10,400)

### Resolviendo la ODE

X = odeint(Modelo,X0,t)
y = X[:,0]
z = X[:,1]

### Plotear resultados

ax = plt.figure().add_subplot(111)
ax.plot(t,y,linewidth = 1.5,label = "y")
ax.plot(t,z,linewidth = 1.5,label = "z")
ax.set_xlabel('Tiempo (s)')
ax.set_ylabel('Magnitud')
ax.set_title('Ejemplo ecuacion diferencial')
ax.legend(loc='best')
```



Ejemplo ecuacion diferencial

