ACTIVIDAD 2.- CREACION DE UNA GUI

QUIROZ TRUJILLO, OMAR ALEJANDRO

Traductores de Lenguaje II

Introducción

Una GUI (Graphical User Interface) o Interfaz Gráfica de Usuario es un sistema que permite la interacción entre el usuario y una computadora mediante elementos visuales como ventanas, botones, iconos y menús. A diferencia de las interfaces de línea de comandos (CLI), que requieren ingresar texto para ejecutar comandos, las GUIs hacen que la interacción sea más intuitiva y accesible, permitiendo el uso del ratón, pantallas táctiles y otros dispositivos de entrada.

El objetivo de esta actividad es que los estudiantes realice una GUI donde valide con expresiones regulares algunas cadenas

¿Qué nos beneficia una GUI?

1. Facilidad de Uso
- Permite a los usuarios interactuar con el traductor de forma intuitiva sin necesidad de conocimientos técnicos.
- Elementos visuales como botones, menús y cuadros de texto hacen que la navegación sea sencilla.
2. Mayor Accesibilidad
- Facilita el uso a personas con discapacidades mediante funciones como reconocimiento de voz, lectura en voz alta y ajuste de fuentes.
- Soporte para múltiples formatos de entrada (texto, voz, imágenes) que amplían su utilidad.
3. Velocidad y Eficiencia
- Un diseño optimizado permite realizar traducciones rápidas con solo unos clics.
- Funciones como la autocompletación o la detección automática del idioma agilizan el proceso.
4. Interactividad y Experiencia de Usuario Mejorada
- Las GUI modernas pueden incluir animaciones, sonidos y retroalimentación visual para mejorar la experiencia del usuario.

- Opciones de personalización, como modo oscuro o ajustes de idioma preferidos, mejoran la comodidad de uso.

5. Integración con Otras Tecnologías

- Conexión con asistentes de voz (Alexa, Google Assistant).
- Compatibilidad con aplicaciones móviles, traductores en tiempo real y plataformas en la nube.

Código Fuente

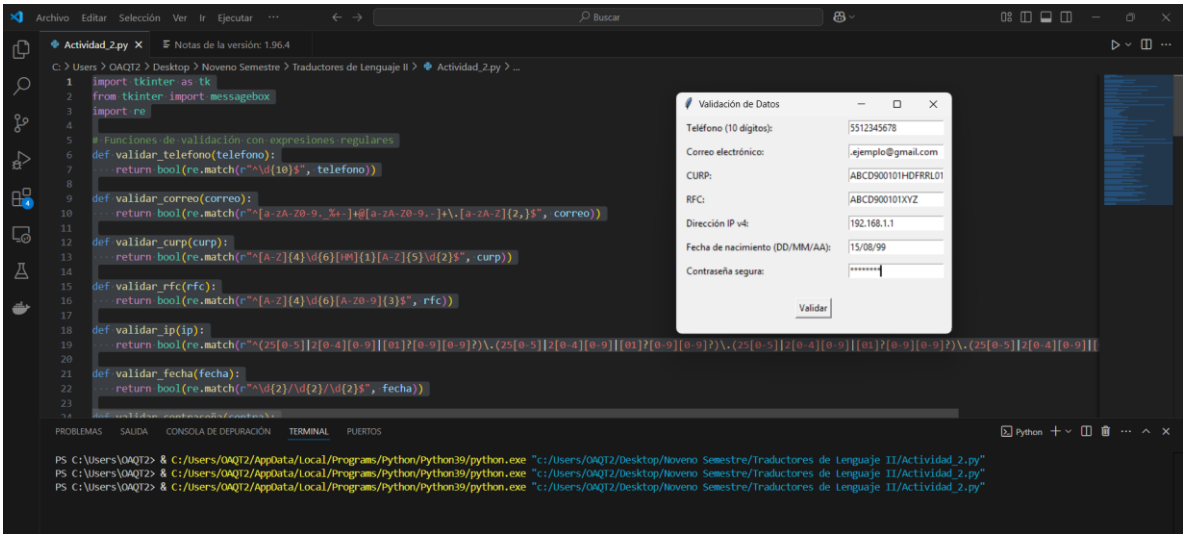
```
import tkinter as tk
from tkinter import messagebox
import re

# Funciones de validación con expresiones regulares
def validar_telefono(telefono):
    return bool(re.match(r"^\d{10}$", telefono))

def validar_correo(correo):
    return bool(re.match(r"^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$", correo))

def validar_curp(curp):
    return bool(re.match(r"^[A-Z]{4}\d{6}[HM]{1}[A-Z]{5}\d{2}$", curp))

def validar_rfc(rfc):
    return bool(re.match(r"^[A-Z]{4}\d{6}[A-Z0-9]{3}$", rfc))

def validar_ip(ip):
    return bool(re.match(r"^(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$", ip))
```

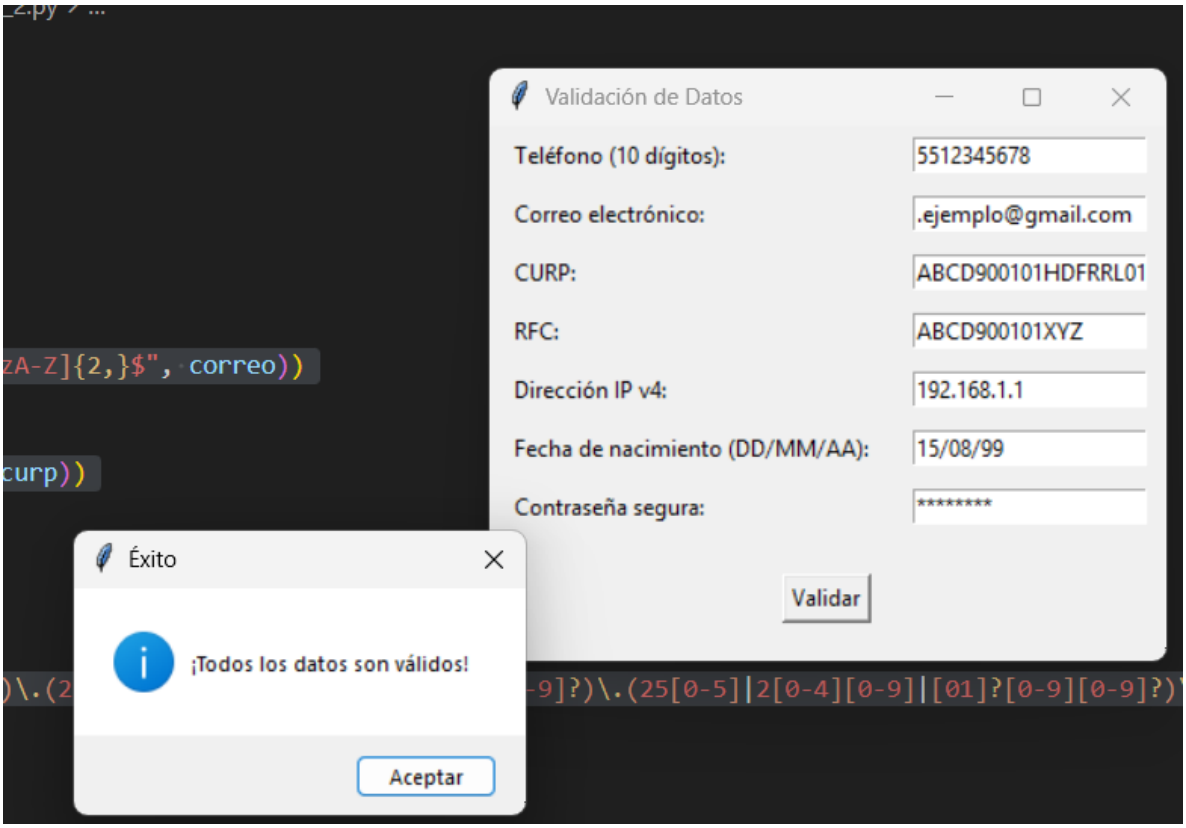
```
def validar_fecha(fecha):
    return bool(re.match(r"^\d{2}/\d{2}/\d{2}$", fecha))


def validar_contraseña(contra):
    return bool(re.match(r"^(?=.*[A-Za-z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-
z\d@$!%*?&]{8,}$", contra))


# Función que valida los campos de texto
def validar():
    telefono = entry_telefono.get()
    correo = entry_correo.get()
    curp = entry_curp.get()
    rfc = entry_rfc.get()
    ip = entry_ip.get()
    fecha = entry_fecha.get()
    contrasena = entry_contraseña.get()


    if not validar_telefono(telefono):
        messagebox.showerror("Error", "Teléfono inválido. Debe tener 10 dígitos.")
        return
    if not validar_correo(correo):
        messagebox.showerror("Error", "Correo electrónico inválido.")
        return
    if not validar_curp(curp):
        messagebox.showerror("Error", "CURP inválida.")
        return
    if not validar_rfc(rfc):
        messagebox.showerror("Error", "RFC inválido.")
        return
    if not validar_ip(ip):
```

```
        messagebox.showerror("Error", "Dirección IP inválida.")
        return
    if not validar_fecha(fecha):
        messagebox.showerror("Error", "Fecha de nacimiento inválida. Debe ser en formato DD/MM/AA.")
        return
    if not validar_contraseña(contrasena):
        messagebox.showerror("Error", "Contraseña inválida. Debe tener al menos 8 caracteres, una letra, un número y un carácter especial.")
        return

    messagebox.showinfo("Éxito", "¡Todos los datos son válidos!")

# Crear la ventana principal
ventana = tk.Tk()
ventana.title("Validación de Datos")

# Crear los campos de entrada
label_telefono = tk.Label(ventana, text="Teléfono (10 dígitos):")
label_telefono.grid(row=0, column=0, padx=10, pady=5, sticky="w")
entry_telefono = tk.Entry(ventana)
entry_telefono.grid(row=0, column=1, padx=10, pady=5)

label_correo = tk.Label(ventana, text="Correo electrónico:")
label_correo.grid(row=1, column=0, padx=10, pady=5, sticky="w")
entry_correo = tk.Entry(ventana)
entry_correo.grid(row=1, column=1, padx=10, pady=5)

label_curp = tk.Label(ventana, text="CURP:")
label_curp.grid(row=2, column=0, padx=10, pady=5, sticky="w")
```

```
entry_curp = tk.Entry(ventana)
entry_curp.grid(row=2, column=1, padx=10, pady=5)

label_rfc = tk.Label(ventana, text="RFC:")
label_rfc.grid(row=3, column=0, padx=10, pady=5, sticky="w")
entry_rfc = tk.Entry(ventana)
entry_rfc.grid(row=3, column=1, padx=10, pady=5)

label_ip = tk.Label(ventana, text="Dirección IP v4:")
label_ip.grid(row=4, column=0, padx=10, pady=5, sticky="w")
entry_ip = tk.Entry(ventana)
entry_ip.grid(row=4, column=1, padx=10, pady=5)

label_fecha = tk.Label(ventana, text="Fecha de nacimiento (DD/MM/AA):")
label_fecha.grid(row=5, column=0, padx=10, pady=5, sticky="w")
entry_fecha = tk.Entry(ventana)
entry_fecha.grid(row=5, column=1, padx=10, pady=5)

label_contrasena = tk.Label(ventana, text="Contraseña segura:")
label_contrasena.grid(row=6, column=0, padx=10, pady=5, sticky="w")
entry_contraseña = tk.Entry(ventana, show="*")
entry_contraseña.grid(row=6, column=1, padx=10, pady=5)

# Botón para validar
boton_validar = tk.Button(ventana, text="Validar", command=validar)
boton_validar.grid(row=7, column=0, columnspan=2, pady=20)

# Iniciar la ventana
ventana.mainloop()
```

Resultados


Archivo Editar Selección Ver Ir Ejecutar
Actividad_2.py
Notas de la versión: 1.96.4
C: > Users > OAQT2 > Desktop > Noveno Semestre > Traductores de Lenguaje II > Actividad_2.py > ...
import tkinter as tk
from tkinter import messagebox
import re
# Funciones de validación con expresiones regulares
def validar_telefono(telefono):
return bool(re.match(r"^\d{10}$", telefono))
def validar_correo(correo):
return bool(re.match(r"^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$", correo))
def validar_curp(curp):
return bool(re.match(r"^[A-Z]{4}\d{6}[HM]{1}[A-Z]{5}\d{2}$", curp))
def validar_rfc(rfc):
return bool(re.match(r"^[A-Z]{4}\d{6}[A-Z0-9]{3}$", rfc))
def validar_ip(ip):
return bool(re.match(r"^(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[
def validar_fecha(fecha):
return bool(re.match(r"^\d{2}/\d{2}/\d{2}$", fecha))
Validación de Datos
Teléfono (10 dígitos):
5512345678
Correo electrónico:
.ejemplo@gmail.com
CURP:
ABCD900101HDFRRL01
RFC:
ABCD900101XYZ
Dirección IP v4:
192.168.1.1
Fecha de nacimiento (DD/MM/AA):
15/08/99
Contraseña segura:
Validar
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
Python
PS C:\Users\OAQT2> & C:/Users/OAQT2/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/OAQT2/Desktop/Noveno Semestre/Traductores de Lenguaje II/Actividad_2.py"
PS C:\Users\OAQT2> & C:/Users/OAQT2/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/OAQT2/Desktop/Noveno Semestre/Traductores de Lenguaje II/Actividad_2.py"
PS C:\Users\OAQT2> & C:/Users/OAQT2/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/OAQT2/Desktop/Noveno Semestre/Traductores de Lenguaje II/Actividad_2.py"


Validación de Datos
Teléfono (10 dígitos):
5512345678
Correo electrónico:
.ejemplo@gmail.com
CURP:
ABCD900101HDFRRL01
RFC:
ABCD900101XYZ
Dirección IP v4:
192.168.1.1
Fecha de nacimiento (DD/MM/AA):
15/08/99
Contraseña segura:
****
Validar
Éxito
¡Todos los datos son válidos!
Aceptar

Al tener algún cambio o no tener correctamente alguna sección se avisara