

Seminario de Solución de problemas de Traductores de Lenguajes II

Centro Universitario de Ciencias Exactas en
ingenierías

Universidad de Guadalajara



Maestro: **LUIS FELIPE MUNOZ MENDOZA**

Juan Antonio Pérez Juárez
Código: 215660996
Carrera: INCO

Actividad 1.- GUI de ER

Introducción:

Esta actividad tiene como objetivo el desarrollo de una interfaz gráfica en un lenguaje de programación para el análisis sintáctico y semántico de varias entradas del usuario, que en este caso debe Validar:

- - Teléfono de 10 dígitos
- - Correo electrónico
- - CURP
- - RFC
- - Dirección IPv4
- - Fecha de cumpleaños DD/MM/AA

Desarrollo:

Primero que nada, nosotros seleccionamos el lenguaje de programación python, por que cuenta con una librería especializada en las interfaces gráficas y una para el uso de expresiones regulares, por lo que es más sencillo que en otros lenguajes de programación, como por ejemplo Java, que no lo tiene.

Además, Python tiene mucho soporte y mucha documentación.

Así que nos decantamos por eso.

Así que aquí tenemos el código:x

Código:

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

def validate_and_submit():
    phone = phone_entry.get()
    email = email_entry.get()
    curp = curp_entry.get()
    rfc = rfc_entry.get()
    ip = ip_entry.get()
    birthday = birthday_entry.get()

    errors = []
```

```

# Validate phone number
if not re.fullmatch(r"\d{10}", phone):
    errors.append("Teléfono debe tener 10 dígitos.")

# Validate email
if not re.fullmatch(r"^[^\s]+@[^\s]+\.[^\s]+$", email):
    errors.append("Correo electrónico no válido.")

# Validate CURP
if not re.fullmatch(r"[A-Z]{4}\d{6}[HM][A-Z]{5}\d{2}", curp):
    errors.append("CURP no válida.")

# Validate RFC
if not re.fullmatch(r"[A-ZÑ&]{3,4}\d{6}[A-Z0-9]{3}", rfc):
    errors.append("RFC no válido.")

# Validate IPv4 address
if not re.fullmatch(r"(\d{1,3}\.){3}\d{1,3}", ip) or not all(0 <=
int(part) <= 255 for part in ip.split('.')):
    errors.append("Dirección IP v4 no válida.")

# Validate birthday
if not re.fullmatch(r"\d{2}/\d{2}/\d{2}", birthday):
    errors.append("Fecha de cumpleaños debe estar en formato DD/MM/AA.")

# Show errors or success message
if errors:
    messagebox.showerror("Errores de validación", "\n".join(errors))
else:
    messagebox.showinfo("Éxito", "Todos los datos son válidos.")

# Create the main window
root = tk.Tk()
root.title("Formulario de datos personales")
root.geometry("400x400")

# Labels and Entry fields
fields = [
    ("Teléfono (10 dígitos):", "phone_entry"),
    ("Correo electrónico:", "email_entry"),
    ("CURP:", "curp_entry"),
    ("RFC:", "rfc_entry"),
    ("Dirección IP v4:", "ip_entry"),
    ("Fecha de cumpleaños (DD/MM/AA):", "birthday_entry"),
]

entries = {}
for i, (label_text, var_name) in enumerate(fields):

```

```

label = tk.Label(root, text=label_text, anchor="w")
label.grid(row=i, column=0, padx=10, pady=5, sticky="w")
entry = tk.Entry(root, width=30)
entry.grid(row=i, column=1, padx=10, pady=5)
entries[var_name] = entry

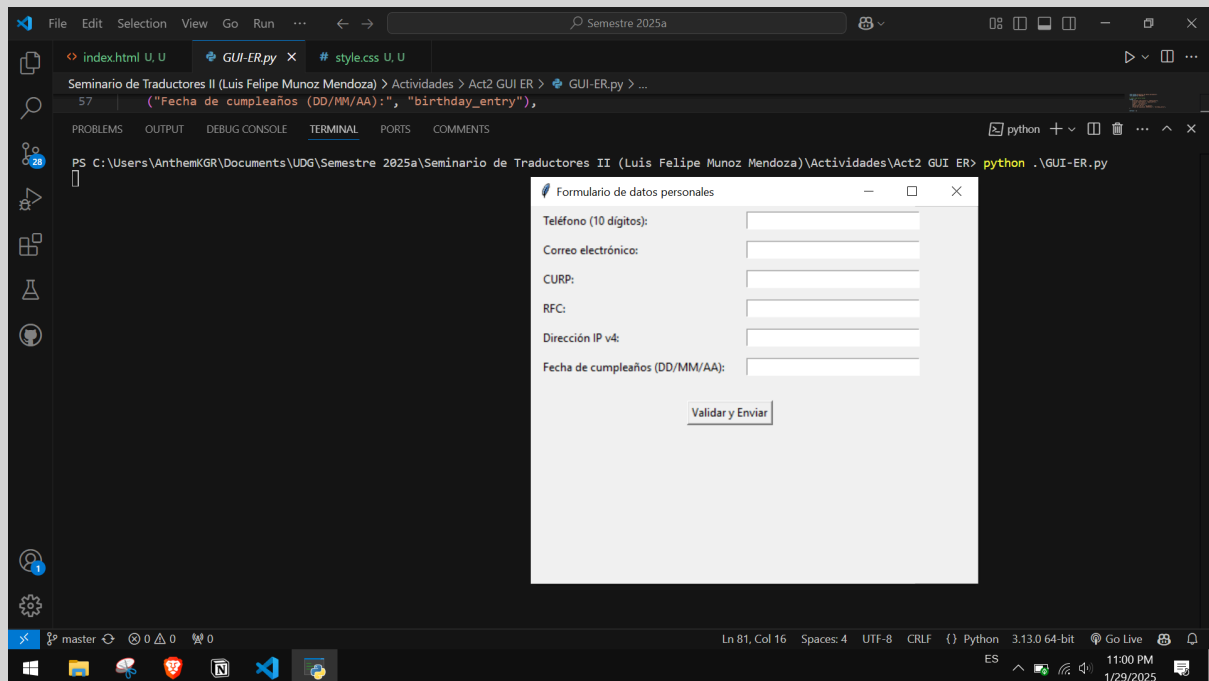
# Map entry variables
phone_entry = entries["phone_entry"]
email_entry = entries["email_entry"]
curp_entry = entries["curp_entry"]
rfc_entry = entries["rfc_entry"]
ip_entry = entries["ip_entry"]
birthday_entry = entries["birthday_entry"]

# Submit button
submit_button = tk.Button(root, text="Validar y Enviar",
command=validate_and_submit)
submit_button.grid(row=len(fields), column=0, columnspan=2, pady=20)

# Run the application
root.mainloop()

```

Pruebas:



The image shows a web application interface with a dark-themed code editor on the left and a light-themed form titled "Formulario de datos personales" on the right. The code editor contains JavaScript validation logic for fields like 'curp' and 'birthday'. The form contains input fields for 'Teléfono (10 dígitos)', 'Correo electrónico', 'CURP', 'RFC', 'Dirección IP v4', and 'Fecha de cumpleaños (DD/MM/AA)'. All fields are filled with valid data. A blue information icon dialog box is overlaid on the code editor, displaying the message "Éxito" and "Todos los datos son válidos." with an "OK" button. A "Validar y Enviar" button is at the bottom of the form.

Field	Value
Teléfono (10 dígitos):	3338203102
Correo electrónico:	juan.perez0996@alumnos.udg.mx
CURP:	PEJJ001004HJCRRN23
RFC:	PEJJ001005CN5
Dirección IP v4:	192.168.100.1
Fecha de cumpleaños (DD/MM/AA):	04/10/00

Estas capturas muestran el resultado de la validación cuando todos los datos son correctos.

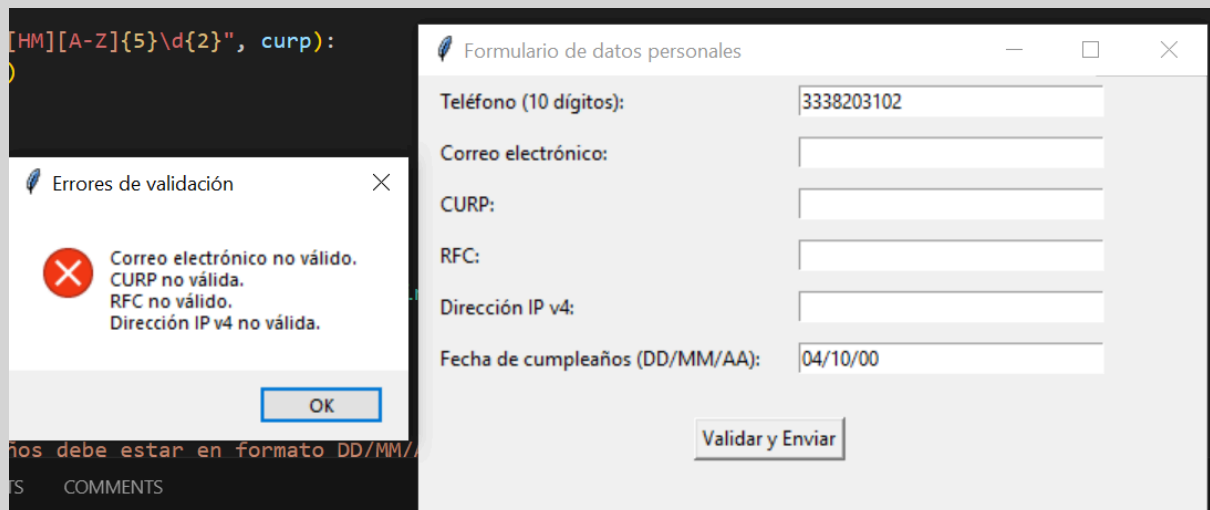
Ahora veamos el caso cuando alguno de los datos no son correctos.

The image shows the same web application interface as before, but with an error. The 'RFC' field in the form now contains 'PEJJ001005C000' (an extra zero). A red error dialog box is overlaid on the code editor, displaying the message "Errores de validaci..." and "RFC no válido." with an "OK" button. The "Validar y Enviar" button is still present at the bottom of the form.

Field	Value
Teléfono (10 dígitos):	3338203102
Correo electrónico:	juan.perez0996@alumnos.udg.mx
CURP:	PEJJ001004HJCRRN23
RFC:	PEJJ001005C000
Dirección IP v4:	192.168.100.1
Fecha de cumpleaños (DD/MM/AA):	04/10/00

Aquí le moví un poco el RFC, pero el error principal es que puse un dígito de más.

Y ahora dejaremos unos campos en blancos para ver cómo se comporta el software.



Conclusión:

Para esta actividad la verdad si me puse un tiro con el código, porque la verdad siempre he sido muy malo para teoría de la computación, terriblemente malo, así que me tuve que avocar a la programación de la interfaz gráfica para que mis compañeros de equipo me ayudasen a programar la parte de la funcionalidad en cuanto a la validación con expresiones regulares.

Así que espero poder recordar algo de la teoría de la computación, que por desgracia no encuentro mis apuntes de ese semestre.