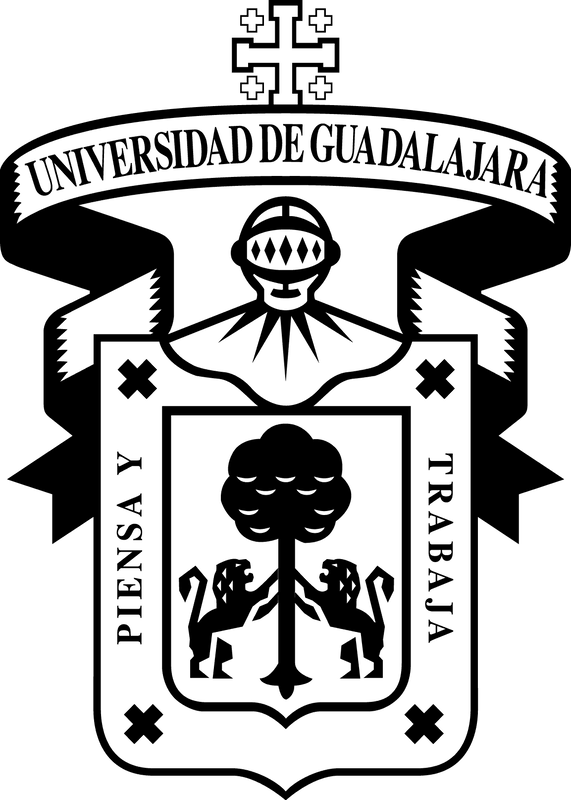
Seminario de Solución de problemas de Traductores de Lenguajes II

# Centro Universitario de Ciencias Exactas en ingenierías

# Universidad de Guadalajara



Maestro: [LUIS FELIPE MUNOZ MENDOZA](mailto:luis.munoz.m@academicos.udg.mx)

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 analisis sintactico y semantico 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:

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

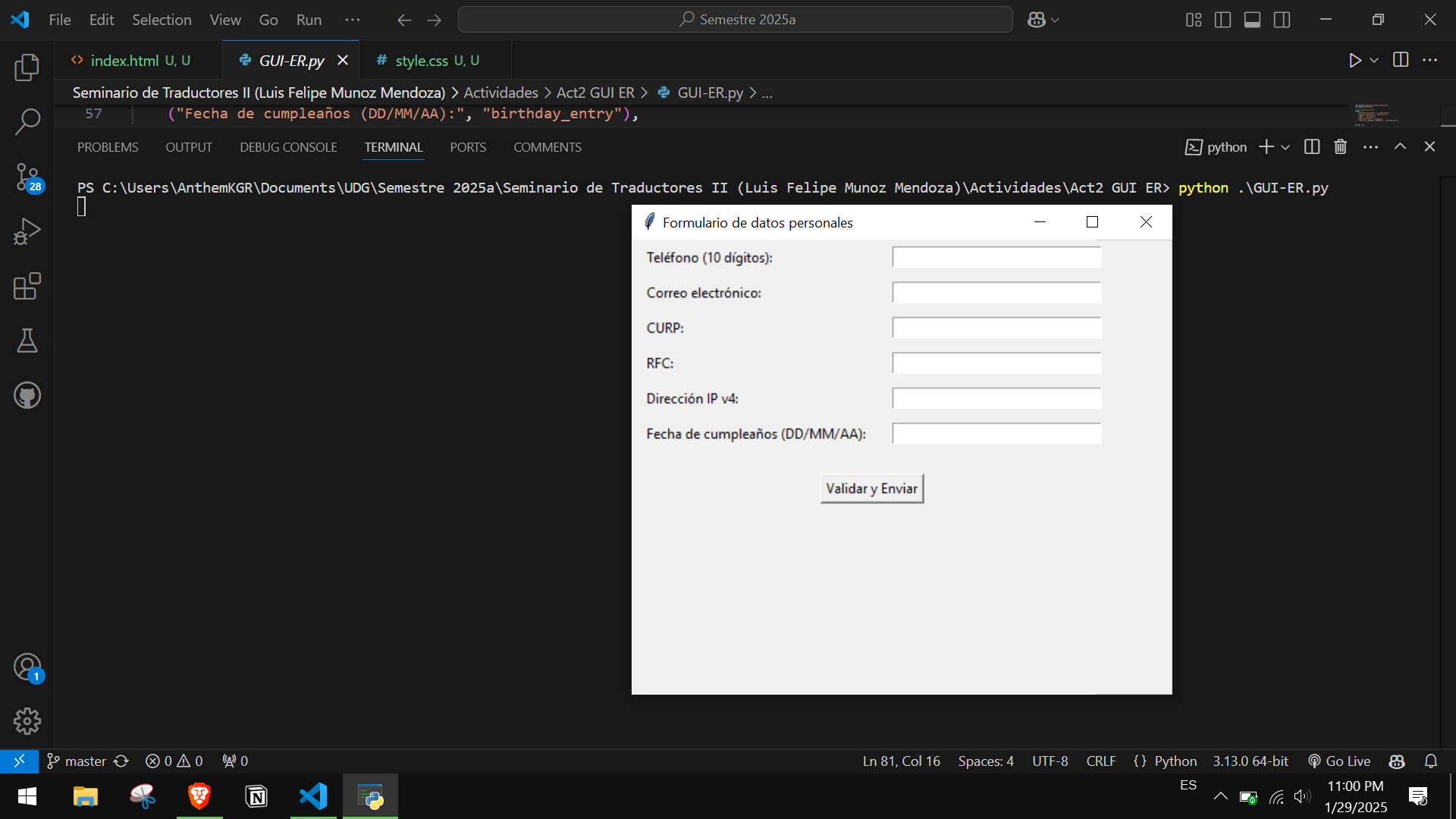
submit\_button = tk.Button(root, text="Validar y Enviar", command=validate\_and\_submit)

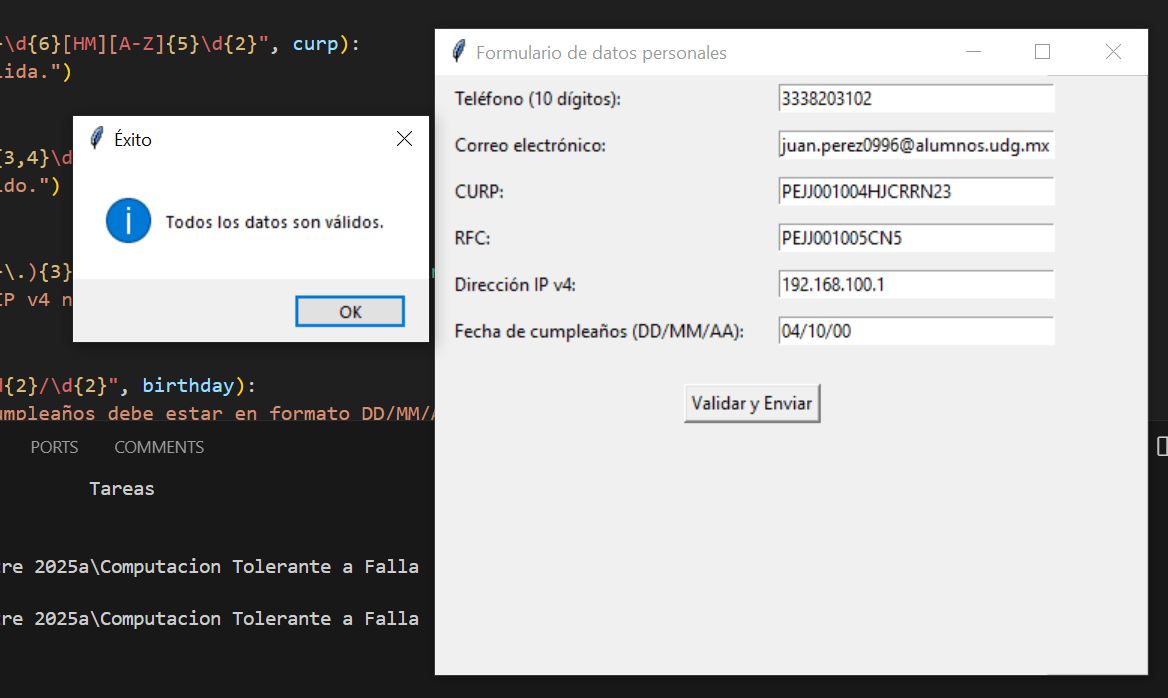
submit\_button.grid(row=len(fields), column=0, columnspan=2, pady=20)

# Run the application

root.mainloop()

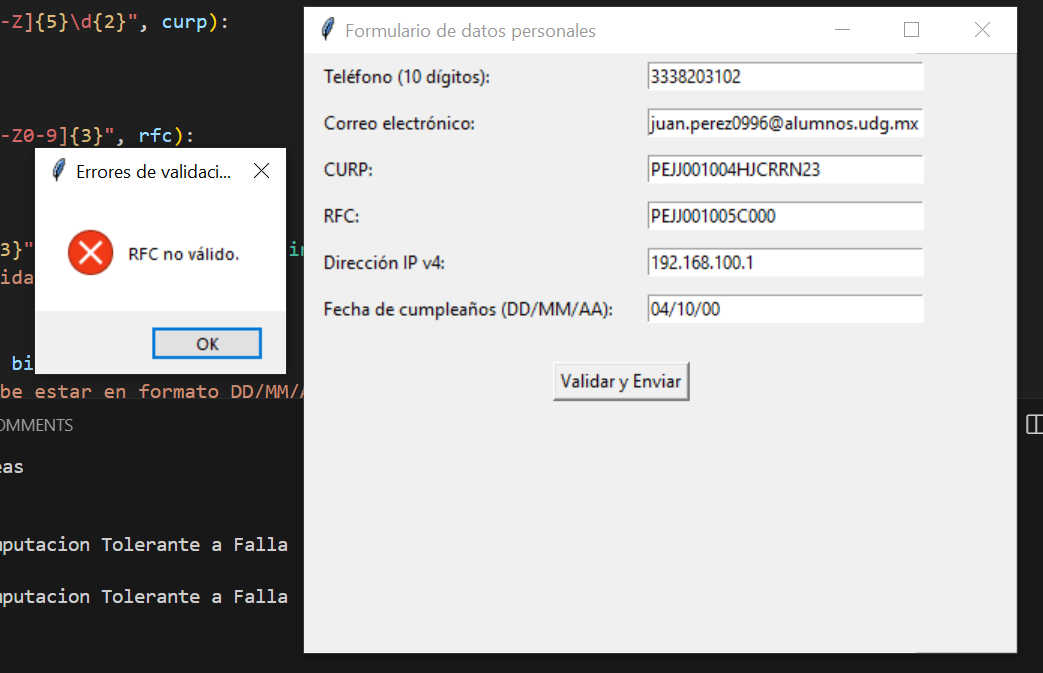
Pruebas:





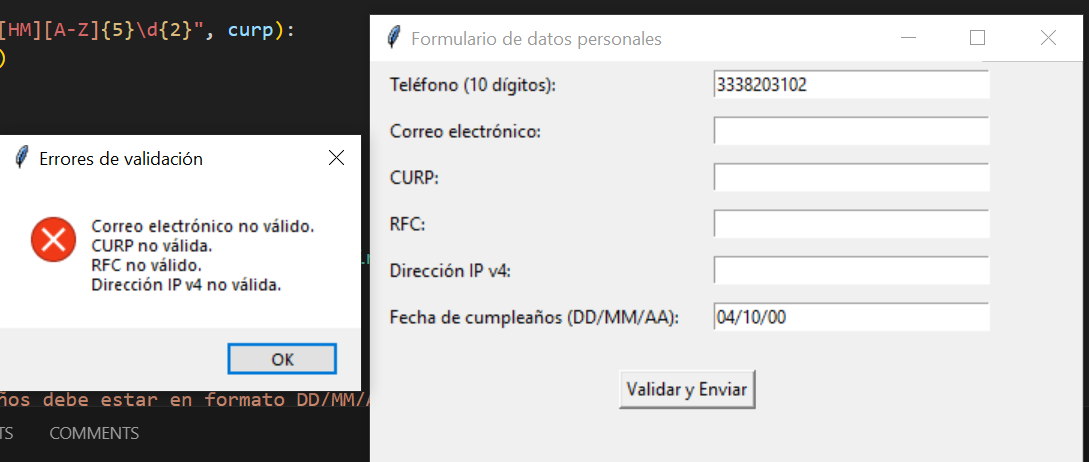
Estas capturas muestran el resultado de la validación cuando todos los datos son correctos.

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



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, porqué 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.