

Centro Universitario de Ciencias Exactas e Ingeniería

Sem. de Solución de Problemas de Traductores 2

Actividad 2 Gui de ER

LUIS FELIPE MUNOZ MENDOZA

Juan Pablo Hernández Orozco

219294285

Objetivo:

El objetivo principal es implementar una interfaz gráfica (GUI) que facilite la validación de entradas de datos mediante patrones definidos con expresiones regulares. Esto permite detectar y notificar al usuario cualquier error en la sintaxis de los datos introducidos, asegurando que estos cumplan con el formato requerido

Introducción

El presente proyecto tiene como finalidad desarrollar una aplicación gráfica en Python que permita validar diferentes tipos de datos utilizando expresiones regulares. Se busca garantizar que la información ingresada por el usuario se ajuste a los formatos establecidos para cada campo. La aplicación valida los siguientes tipos de cadena:

- Teléfono de 10 dígitos
- Correo electrónico
- CURP
- RFC
- Dirección IP v4
- Fecha de cumpleaños en formato DD/MM/AA

Desarrollo:

El código se estructura en dos partes principales:

- Interfaz Gráfica (GUI):

Se utiliza la librería Tkinter para crear la ventana principal y organizar los distintos widgets (etiquetas, campos de texto y botón). Cada campo de entrada corresponde a uno de los datos que se desean validar (teléfono, email, CURP, etc.).

- Validación de Entradas:

La función `validate_entries()` se encarga de:

Recuperar y limpiar (con `.strip()`) los datos introducidos en cada campo.

Definir y compilar, mediante la librería `re`, las expresiones regulares correspondientes para cada tipo de dato.

Validar cada entrada comparándola con su respectiva expresión regular. En caso de que alguna validación falle, se acumulan mensajes de error en una lista.

Mostrar, a través de ventanas emergentes (`messagebox`), un mensaje de éxito si todos los campos son válidos o un mensaje de error detallado en caso contrario.

- Detalle de las Expresiones Regulares Utilizadas

Teléfono de 10 dígitos:

```
regex_phone = re.compile(r'^\d{10}$')
```

Valida que la cadena contenga exactamente 10 números.

Correo Electrónico:

```
regex_email = re.compile(r'^[w\.-]+@[w\.-]+\w+$')
```

Verifica un formato básico de dirección de correo, permitiendo letras, números, puntos y guiones.

CURP (Clave Única de Registro de Población):

```
regex_curp = re.compile(r'^[A-Z]{4}\d{6}[HM][A-Z]{5}[0-9A-Z]\d$')
```

Este patrón es una versión simplificada que valida la estructura general de una CURP.

RFC (Registro Federal de Contribuyentes):

```
regex_rfc = re.compile(r'^([A-ZÑ&]{3,4})(\d{6})([A-Z\d]{3})$')
```

Valida el formato esperado para personas físicas, que incluye letras, dígitos y caracteres especiales.

Dirección IP v4:

Se utiliza un patrón que valida cada uno de los cuatro octetos, asegurando que cada uno se encuentre en el rango de 0 a 255:

```
regex_ip = re.compile(  
    r'^(?:25[0-5]|2[0-4]\d|[01]?\d?\d)\.){3}'  
    r'(?:25[0-5]|2[0-4]\d|[01]?\d?\d)$'  
)
```

Fecha de Cumpleaños (DD/MM/AA):

```
regex_birthday = re.compile(r'^(0[1-9]|12)\d3[01])/(0[1-9]|1[0-2])\d{2}$')
```

Asegura que el día se encuentre entre 01 y 31, el mes entre 01 y 12, y el año sea de dos dígitos.

- Funcionamiento e Interacción con el Usuario

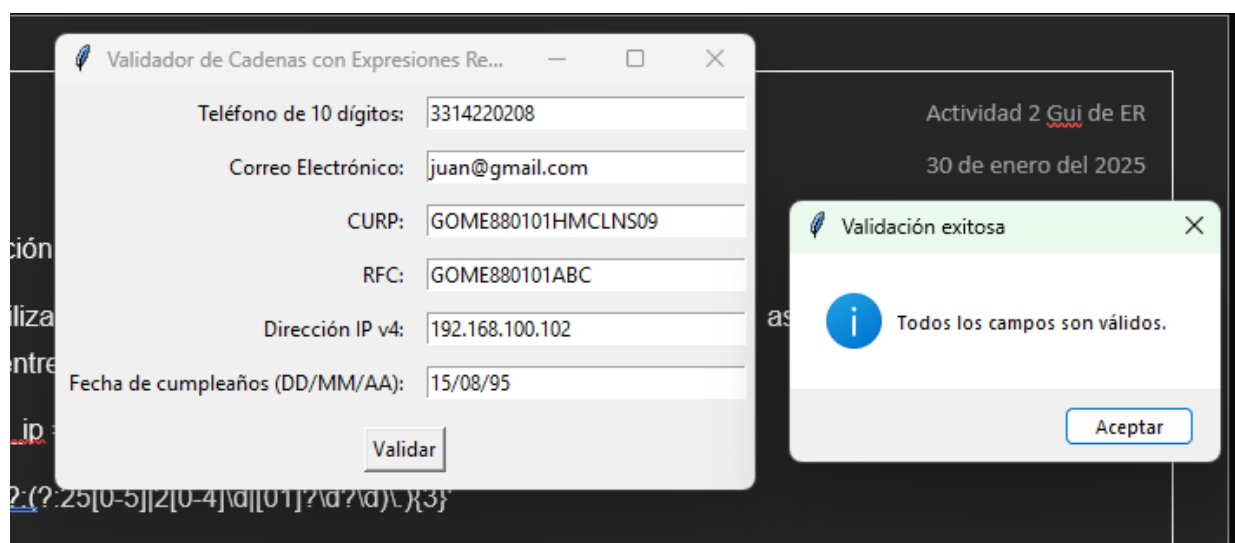
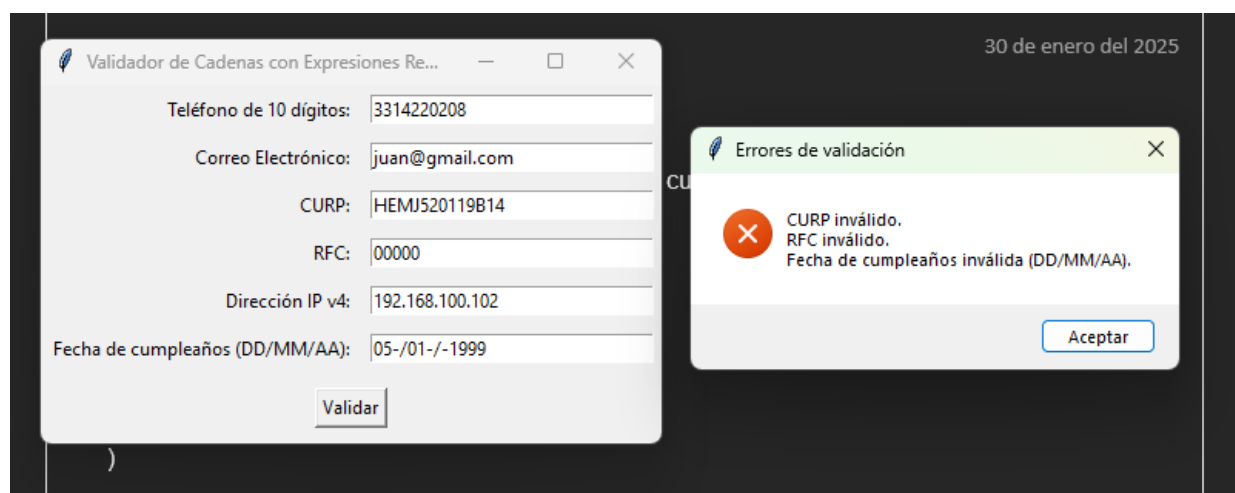
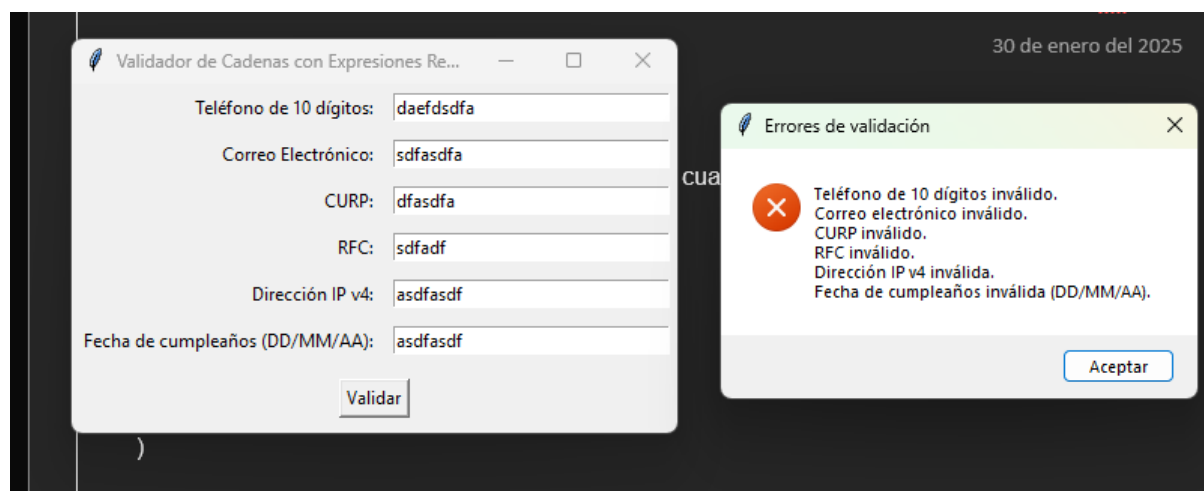
Al iniciar la aplicación se despliega una ventana con campos para cada tipo de dato.

El usuario ingresa la información en cada uno de los campos y presiona el botón “Validar”.

La función `validate_entries()` procesa las entradas:

Si alguna de las validaciones falla, se muestra una ventana de error indicando específicamente cuáles campos tienen un formato inválido.

Si todas las entradas cumplen con los patrones definidos, se muestra una ventana informando que “Todos los campos son válidos.”



Conclusión

El desarrollo de esta aplicación demuestra el uso eficaz de expresiones regulares para la validación de datos en aplicaciones gráficas. La solución implementada permite:

- Facilitar la interacción del usuario mediante una GUI intuitiva.
- Garantizar que los datos ingresados se ajusten a los formatos requeridos, reduciendo errores en la entrada de información.
- Servir como base para extender la validación a otros tipos de datos o mejorar los patrones existentes según sea necesario.