

UNIVERSIDAD DE GUADALAJARA



CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

TRADUCTORES DE LENGUAJES II

Reporte de práctica

Nombre del alumno:	Fernando Benavides Castro
Profesor:	Erasmus Gabriel Martínez Soltero
Título de la práctica:	“Tarea 1. Expresiones Regulares”
Fecha:	04 septiembre 2023

Introducción

En esta tarea, se realiza un programa que se encargue de validar entradas basadas en expresiones regulares, con una interfaz gráfica en lenguaje de mi elección. Su principal propósito es permitir al usuario ingresar diferentes tipos de datos y validar si estos datos son correctos según ciertos patrones predefinidos. En este caso se solicitaron validar cuatro distintas expresiones, las cuales el programa debe mostrar si son expresiones válidas o no.

Metodología

El programa se creo con el lenguaje de programacion Python, en conjunto con sus librerias -tkinter- y -re-, las cuales son de ayuda para crear la

interfaz gráfica y manejar expresiones regulares respectivamente. El programa es una herramienta con interfaz gráfica que permite al usuario validar diferentes tipos de datos (como correos electrónicos, fechas, horas y direcciones IP) para comprobar si cumplen con formatos específicos.

Funciones:

validar-correo(), validar-fecha(), validar-hora(), validar-ip():

Estas funciones recogen el texto ingresado por el usuario en los campos correspondientes y aplican una expresión regular (patrón) para validar si el texto coincide con el formato esperado. Luego, invocan a `mostrar-resultado` para mostrar si el texto es válido o no.

validar(texto, patrón):

Es una función genérica que recibe un texto y un patrón. Retorna verdadero si el texto coincide con el patrón (es decir, si es válido) y falso en caso contrario.

mostrar-resultado(resultado, fila):

Esta función actualiza la interfaz gráfica para mostrar el resultado de la validación. Si el dato ingresado es válido, muestra -Entrada válida- en verde; si no lo es, muestra -Entrada inválida- en rojo.

La interfaz gráfica presenta etiquetas para describir cada tipo de dato, campos donde el usuario puede ingresar el dato a validar y botones para iniciar la validación. También hay una columna que muestra el resultado de la validación.

```
main.py X
D: > INCO FER 2023B > Traductores 2 > Tarea 1. Expresiones Regulares > main.py > mostrar_resultado
1 import re
2 import tkinter as tk
3
4 def validar_correo():
5     text = correo_entrada.get()
6     pattern = r"^[a-zA-Z0-9_+]+@[a-zA-Z0-9]+\.[a-zA-Z0-9-]+$"
7     mostrar_resultado(validar(text, pattern), 0)
8
9 def validar_fecha():
10    text = fecha_entrada.get()
11    pattern = r"^(0[1-9]|12)[0-9]/(0[1-9]|1[0-2])/d{4}$"
12    mostrar_resultado(validar(text, pattern), 1)
13
14 def validar_hora():
15    text = hora_entrada.get()
16    pattern = r"^(01\d|2[0-3]):([0-5]\d)$"
17    mostrar_resultado(validar(text, pattern), 2)
18
19 def validar_ip():
20    text = ip_entrada.get()
21    pattern = r"^((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$"
22    mostrar_resultado(validar(text, pattern), 3)
23
24 def validar(text, pattern):
25    return re.match(pattern, text)
26
27 def mostrar_resultado(is_valid, row):
28    if is_valid:
29        resultados[row].config(text="Entrada válida", fg="green")
30    else:
```

```

31 resultados[row].config(text="Entrada inválida", fg="red")
32
33 app = tk.Tk()
34 app.title("Validador de expresiones")
35
36 resultados = []
37
38 # Correo electrónico
39 correo_label = tk.Label(app, text="Correo electrónico:")
40 correo_label.grid(row=0, column=0, padx=20, pady=5, sticky=tk.W)
41 correo_entrada = tk.Entry(app, width=30)
42 correo_entrada.grid(row=0, column=1, padx=20, pady=5)
43 correo_boton = tk.Button(app, text="Validar", command=validar_correo)
44 correo_boton.grid(row=0, column=2, padx=20, pady=5)
45 resultados.append(tk.Label(app, text=""))
46 resultados[-1].grid(row=0, column=3, padx=20, pady=5)
47
48 # Fecha
49 fecha_label = tk.Label(app, text="Fecha (DD/MM/YYYY):")
50 fecha_label.grid(row=1, column=0, padx=20, pady=5, sticky=tk.W)
51 fecha_entrada = tk.Entry(app, width=30)
52 fecha_entrada.grid(row=1, column=1, padx=20, pady=5)
53 fecha_boton = tk.Button(app, text="Validar", command=validar_fecha)
54 fecha_boton.grid(row=1, column=2, padx=20, pady=5)
55 resultados.append(tk.Label(app, text=""))
56 resultados[-1].grid(row=1, column=3, padx=20, pady=5)
57
58 # Hora
59 hora_label = tk.Label(app, text="Hora (HH:MM):")
60 hora_label.grid(row=2, column=0, padx=20, pady=5, sticky=tk.W)
61 hora_entrada = tk.Entry(app, width=30)
62 hora_entrada.grid(row=2, column=1, padx=20, pady=5)
63 hora_boton = tk.Button(app, text="Validar", command=validar_hora)
64 hora_boton.grid(row=2, column=2, padx=20, pady=5)
65 resultados.append(tk.Label(app, text=""))
66 resultados[-1].grid(row=2, column=3, padx=20, pady=5)
67
68 # Dirección IP
69 ip_label = tk.Label(app, text="Dirección IP:")
70 ip_label.grid(row=3, column=0, padx=20, pady=5, sticky=tk.W)
71 ip_entrada = tk.Entry(app, width=30)
72 ip_entrada.grid(row=3, column=1, padx=20, pady=5)
73 ip_boton = tk.Button(app, text="Validar", command=validar_ip)
74 ip_boton.grid(row=3, column=2, padx=20, pady=5)
75 resultados.append(tk.Label(app, text=""))
76 resultados[-1].grid(row=3, column=3, padx=20, pady=5)
77
78 app.mainloop()

```

Resultados

En esta sección se muestran los resultados obtenidos así como los datos de entrada utilizados para la ejecución del programa;


Si los datos de entrada se ingresaron CORRECTAMENTE y son válidos en el lenguaje, el programa muestra -Entrada válida- en letras verdes, como se ve a continuación:

The screenshot shows a window titled "Validador de expresiones" with four rows of input fields and buttons. Each row has a label, an input field, a "Validar" button, and a result label.

Field	Input	Button	Result
Correo electrónico:	fer.benavides2002@gmail.com	Validar	Entrada válida
Fecha (DD/MM/YYYY):	04/09/2023	Validar	Entrada válida
Hora (HH:MM):	05:22	Validar	Entrada válida
Dirección IP:	187.191.8.159	Validar	Entrada válida

Si los datos de entrada se ingresaron INCORRECTAMENTE y son

inválidos en el lenguaje, el programa muestra -Entrada inválida- en letras rojas, como se ve a continuación:



The screenshot shows a Tkinter window titled "Validador de expresiones". It contains four rows of input fields, each with a "Validar" button and a status label. All four inputs are marked as "Entrada inválida" (Invalid input) in red text.

Field	Value	Button	Status
Correo electrónico:	fer.benavides2002gmail.com	Validar	Entrada inválida
Fecha (DD/MM/YYYY):	01/13/2023	Validar	Entrada inválida
Hora (HH:MM):	25:22	Validar	Entrada inválida
Dirección IP:	256.191.8.159	Validar	Entrada inválida

Conclusiones

El programa demostró ser una herramienta eficiente para validar datos, facilitando la tarea mediante una interfaz gráfica fácil de usar. Las expresiones regulares nos dan precisión para analizar y filtrar entradas. Su uso en el programa demuestra la importancia de las mismas al momento de hacer validación y procesamiento de datos, algo que se lleva a cabo día con día en nuestros dispositivos.

Referencias

- Python, R. (2021, 27 agosto). Introducción a TCL/TK (Tkinter) - recursos Python. Recursos Python. <https://recursospython.com/guias-y-manuales/introduccion-a-tkinter/>
- Re — expresiones regulares. (s. f.). <https://rico-schmidt.name/pymotw-3/re/index.html>
- Re — regular expression operations. (s. f.). Python documentation. <https://docs.python.org/3/library/re.html>