



EJERCICIO 2

Mercado Manzo Luis Alfonso



15 DE OCTUBRE DE 2023

```
# -*- coding: utf-8 -*-
```

```
"""
```

```
Created on Tue Sep 12 13:36:02 2023
```

```
@author: usuario
```

```
"""
```

```
import tkinter as tk
```

```
import re
```

```
def validar_telefono():
```

```
    telefono = entrada_telefono.get()
```

```
    # Expresión regular para validar un número de teléfono de 10 dígitos
```

```
    patron = r'^\d{10}$'
```

```
    if re.match(patron, telefono):
```

```
        resultado_telefono.config(text="Teléfono válido")
```

```
    else:
```

```
        resultado_telefono.config(text="Teléfono inválido")
```

```
def validar_correo():
```

```
    correo = entrada_correo.get()
```

```
    # Expresión regular para validar una dirección de correo electrónico
```

```
    patron = r'^[\w\.-]+@[\w\.-]+$'
```

```
    if re.match(patron, correo):
```

```
        resultado_correo.config(text="Correo válido")
```

```
    else:
```

```
        resultado_correo.config(text="Correo inválido")
```

```
def validar_Curp():
```

```

curp = entrada_curp.get()

# Expresión regular para validar una dirección de correo electrónico
patron = r'^\w{4}\d{6}\w{6}\d{2}$'

if re.match(patron, curp):
    resultado_curp.config(text="Curp válido")
else:
    resultado_curp.config(text="Curp inválido")

def validar_RFC():
    RFC = entrada_RFC.get()

    # Expresión regular para validar una dirección de correo electrónico
    patron = r'^\w{4}\d{5}$'

    if re.match(patron, RFC):
        resultado_RFC.config(text="RFC válido")
    else:
        resultado_RFC.config(text="RFC inválido")

def validar_IP():
    IP = entrada_IP.get()

    # Expresión regular para validar una dirección de correo electrónico
    patron = r'^\d{3}\.\d{2}\.\d{2}\.\d{1}$'

    if re.match(patron, IP):
        resultado_IP.config(text="IP válido")
    else:
        resultado_IP.config(text="IP inválido")

def validar_FN():
    FN = entrada_FN.get()

    # Expresión regular para validar una dirección de correo electrónico

```

```

patron = r'^\d{2}\d{2}\d{4}$'

if re.match(patron, FN):
    resultado_FN.config(text="FN válido")
else:
    resultado_FN.config(text="FN inválido")

# Agrega funciones para validar CURP, RFC, IP y fecha de cumpleaños aquí

# Configuración de la ventana principal
ventana = tk.Tk()
ventana.title("Validador de Cadenas")

# Crear etiquetas y entradas de texto para cada tipo de cadena
etiqueta_telefono = tk.Label(ventana, text="Teléfono:")
entrada_telefono = tk.Entry(ventana)
boton_validar_telefono = tk.Button(ventana, text="Validar Teléfono", command=validar_telefono)
resultado_telefono = tk.Label(ventana, text="")

etiqueta_correo = tk.Label(ventana, text="Correo Electrónico:")
entrada_correo = tk.Entry(ventana)
boton_validar_correo = tk.Button(ventana, text="Validar Correo", command=validar_correo)
resultado_correo = tk.Label(ventana, text="")

etiqueta_curp = tk.Label(ventana, text="Curp:")
entrada_curp = tk.Entry(ventana)
boton_validar_curp = tk.Button(ventana, text="Validar curp", command=validar_Curp)
resultado_curp = tk.Label(ventana, text="")

etiqueta_RFC = tk.Label(ventana, text="RFC:")

```

```
entrada_RFC = tk.Entry(ventana)
```

```
boton_validar_RFC = tk.Button(ventana, text="Validar RFC", command=validar_RFC)
```

```
resultado_RFC = tk.Label(ventana, text="")
```

```
etiqueta_IP = tk.Label(ventana, text="IP:")
```

```
entrada_IP = tk.Entry(ventana)
```

```
boton_validar_IP = tk.Button(ventana, text="Validar ip", command=validar_IP)
```

```
resultado_IP = tk.Label(ventana, text="")
```

```
etiqueta_FN = tk.Label(ventana, text="fecha de nacimiento:")
```

```
entrada_FN = tk.Entry(ventana)
```

```
boton_validar_FN = tk.Button(ventana, text="Validar fecha de nacimiento", command=validar_FN)
```

```
resultado_FN = tk.Label(ventana, text="")
```

```
# Coloca elementos en la ventana
```

```
etiqueta_telefono.pack()
```

```
entrada_telefono.pack()
```

```
boton_validar_telefono.pack()
```

```
resultado_telefono.pack()
```

```
etiqueta_correo.pack()
```

```
entrada_correo.pack()
```

```
boton_validar_correo.pack()
```

```
resultado_correo.pack()
```

```
etiqueta_curp.pack()
```

```
entrada_curp.pack()
```

```
boton_validar_curp.pack()
```

```
resultado_curp.pack()
```

```
etiqueta_RFC.pack()
```

```
entrada_RFC.pack()
```

```
boton_validar_RFC.pack()
```

```
resultado_RFC.pack()
```

```
etiqueta_IP.pack()
```

```
entrada_IP.pack()
```

```
boton_validar_IP.pack()
```

```
resultado_IP.pack()
```

```
etiqueta_FN.pack()
```

```
entrada_FN.pack()
```

```
boton_validar_FN.pack()
```

```
resultado_FN.pack()
```

```
# Iniciar la aplicación
```

```
ventana.mainloop()
```

descripcion del codigo:

se crea una ventana con la biblioteca tkinter:

en ella se crearon 6 opciones

- Telefono de 10 digitos (se validan 10 digitos)
- Correo electrónico (valida letras y números junto a @ en medio de la expresión)
- CURP (valida 4 letras, 6 digitos, 6 letras y 2 digitos respectivamente)
- RFC (valida 4 letras, 5 digitos)
- Direccion IP v4 (valida 3 digitos (.) 2 digitos (.) 2 digitos (.) y un digito)
- fecha de cumpleaños DD/MM/AA (valida 2 digitos 2 digitos y 4 digitos)

Teléfono:

Validar Teléfono

Correo Electrónico:

Validar Correo

Curp:

Validar curp

RFC:

Validar RFC

IP:

Validar ip

fecha de nacimiento:

Validar fecha de nacimiento

Teléfono:

3313450987

Validar Teléfono

Teléfono válido

Correo Electrónico:

pancho@gmail.com

Validar Correo

Correo válido

Curp:

lamm970602hjrcnt20

Validar curp

Curp válido

RFC:

lamm25678

Validar RFC

RFC válido

IP:

178.12.27.8

Validar ip

IP válido

fecha de nacimiento:

02061997

Validar fecha de nacimiento

FN válido