UNIVERSIDAD CENTRAL DEL ECUADOR

FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

CARRERA DE SISTEMAS DE INFORMACIÓN

SEGURIDAD Y GESTIÓN DE RIESGO EN LAS TI

ESTUDIANTE

Heredia Nicolalde
Vanessa Nayeli

TEMA

Programa para validar una cédula ecuatoriana.

DOCENTE

CHRISTIAN PATRICIO
ESPINOSA MARIN

FECHA

26/11/2025

VALIDAR UNA CÉDULA ECUATORIANA

1. Descripción del sistema

El *Validador de Cédula Ecuatoriana* es una aplicación de escritorio desarrollada en Python con Tkinter que permite verificar la autenticidad de una cédula ecuatoriana mediante el algoritmo oficial del **dígito verificador (módulo 10)**.
El programa valida longitud, provincia, tercer dígito y cálculo del dígito final.

2. Requisitos del sistema

Software

- Windows 10 o superior
- Python 3.11+ (solo para desarrollo)
- Ejecutable .exe para usuarios finales (no requiere Python)

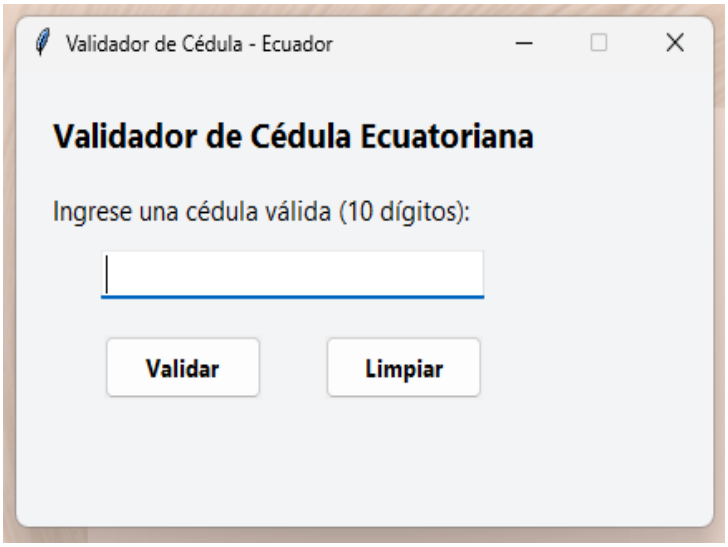
Librerías integradas

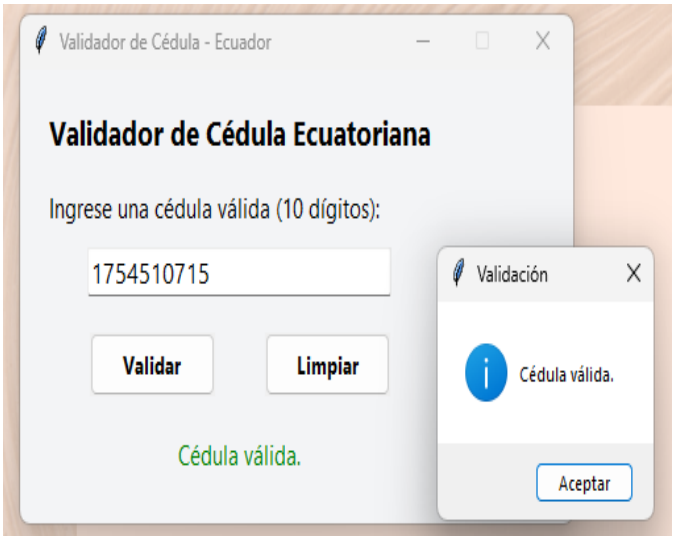
- tkinter
- ttk
- messagebox (incluido en Tkinter, no requiere instalación)

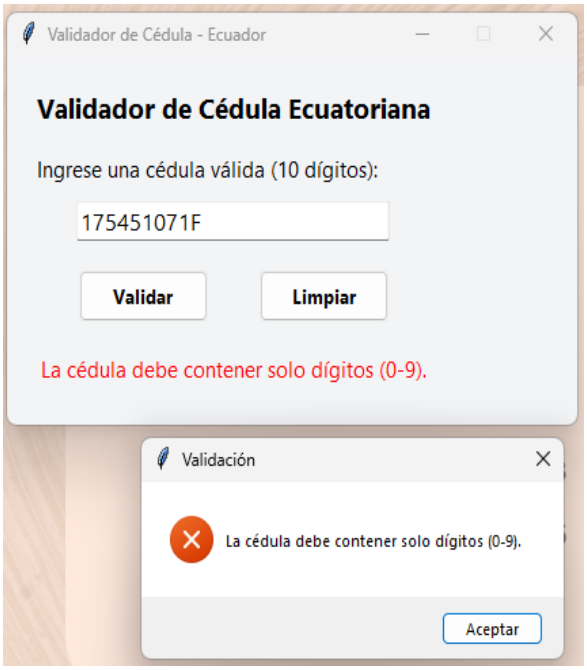
3. Pantalla principal

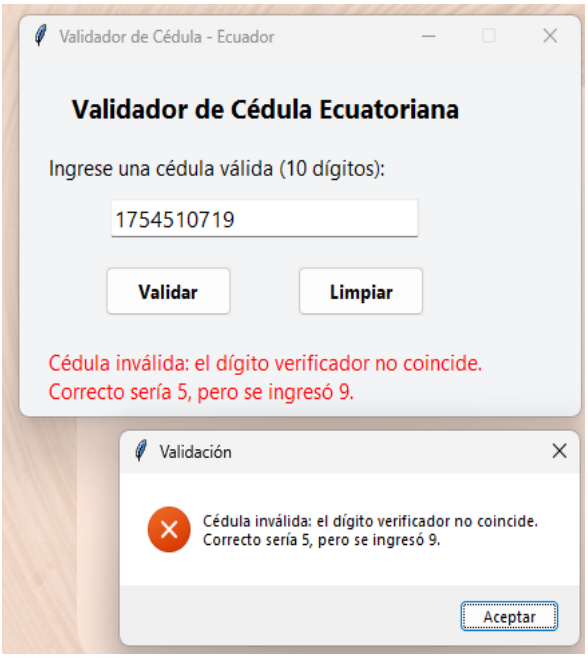
La interfaz contiene:

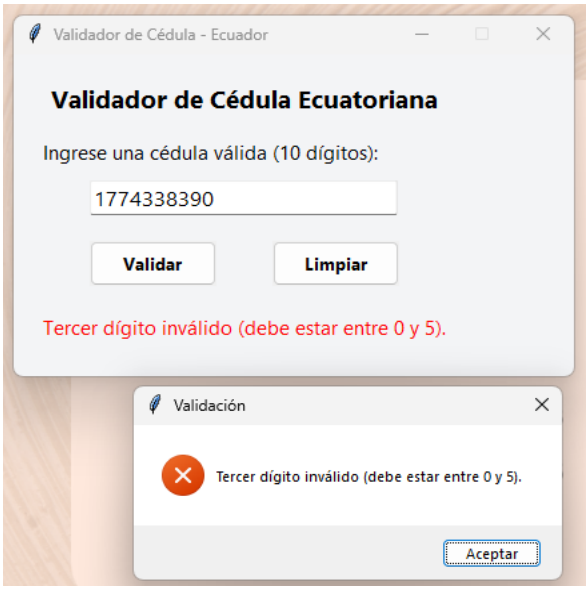
- Título principal
- Campo para ingresar la cédula
- Botón Validar
- Botón Limpiar
- Mensaje de resultado
- Alertas emergentes de éxito o error

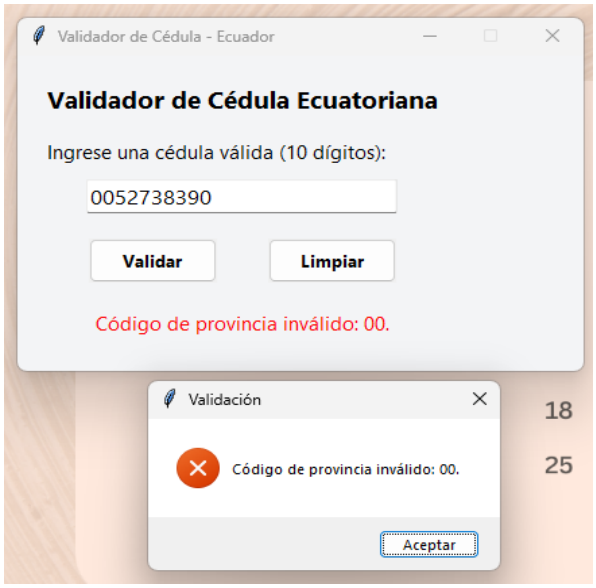












4. Funcionamiento del sistema

4.1 Validar una cédula

1. Escriba los **10 dígitos** de la cédula.
2. Presione el botón **Validar**, o simplemente presione **ENTER**.
3. El sistema revisa automáticamente:
    - Que sean solo números
    - Que tenga 10 dígitos
    - Código de provincia
    - Tercer dígito
    - Dígito verificador
4. Se mostrará un mensaje:
    - **"Cédula válida"** (en verde)

- O un mensaje de error específico (en rojo)

Además, aparece una **ventana emergente** con la información.

4.2 Limpiar la pantalla

Haga clic en **Limpiar** para: Vaciar el campo, borrar el resultado o volver a enfocar el cuadro de texto.

5. Cálculo del dígito verificador (algoritmo interno)

El programa aplica el algoritmo oficial:

1. Multiplica los primeros 9 dígitos por coeficientes fijos:
   2,1,2,1,2,1,2,1,2
2. Si un producto es mayor o igual a 10, se resta 9.
3. Se suman todos los valores.
4. Se obtiene el módulo 10.
5. Si es 0 → el dígito verificador es 0
   Si no → se resta a 10

6. Código del programa en Python

```
import tkinter as tk                    # Importa la librería base de
Tkinter

from tkinter import ttk, messagebox     # ttk para estilo moderno,
messagebox para alertas



def calcular_digito_verificador(primeros_nueve: str) -> int:

    """Calcula el dígito verificador usando el algoritmo oficial (módulo
10)."""

    coeficientes = [2, 1, 2, 1, 2, 1, 2, 1, 2]   # Coeficientes fijos
usados en la norma

    suma = 0                                      # Acumulador de suma

    for i, ch in enumerate(primeros_nueve):      # Recorrer cada dígito
con su índice

        digit = int(ch)                           # Convertir carácter a
entero

        prod = digit * coeficientes[i]            # Multiplicar por
coeficiente correspondiente
```

```
        if prod >= 10:                              # Si el resultado tiene
dos dígitos...

            prod -= 9                               # ...se resta 9
(equivalente a sumar dígitos)

        suma += prod                                # Sumar al acumulador

    resto = suma % 10                               # Sacar módulo 10

    return 0 if resto == 0 else (10 - resto)      # Retornar dígito
verificador calculado



def validar_cedula(cedula: str) -> (bool, str):

    """Valida una cédula ecuatoriana y devuelve (estado, mensaje)."""

    cedula = cedula.strip()                          # Eliminar espacios
innecesarios

    if not cedula.isdigit():                         # Verificar que sean
solo números

        return False, "La cédula debe contener solo dígitos (0-9)."

    if len(cedula) != 10:                            # Validar longitud

        return False, "La cédula debe tener exactamente 10 dígitos."

    prov = int(cedula[:2])                           # Extraer código de
provincia (primeros 2 dígitos)

    if not (1 <= prov <= 24 or prov == 30):        # Validar rango
permitido

        return False, f"Código de provincia inválido: {prov:02d}."

    tercer = int(cedula[2])                          # Extraer tercer
dígito

    if tercer > 5:                                   # Tercer dígito válido
solo entre 0-5

        return False, "Tercer dígito inválido (debe estar entre 0 y 5)."

    primeros_nueve = cedula[:9]                    # Extraer primeros 9
dígitos

    dig_verif_esperado = calcular_digito_verificador(primeros_nueve)  #
Calcular DV esperado
```

```
    dig_verif_real = int(cedula[9])                # DV real ingresado
por usuario

    if dig_verif_esperado != dig_verif_real:       # Comparar DV

        return False, (

            f"Cédula inválida: el dígito verificador no coincide.\n"

            f"Correcto sería {dig_verif_esperado}, pero se ingresó
{dig_verif_real}."

        )



    return True, "Cédula válida."



def on_validar_click():

    """Acción al presionar 'Validar'."""

    ced = entry_cedula.get()                        # Obtener texto del
campo

    valido, mensaje = validar_cedula(ced)           # Validar cédula

    if valido:                                      # Si es válida

        label_resultado.config(text=mensaje, foreground="green")

        messagebox.showinfo("Validación", mensaje)                #
Mostrar ventana informativa

    else:                                           # Si es inválida

        label_resultado.config(text=mensaje, foreground="red")

        messagebox.showerror("Validación", mensaje)               #
Ventana de error

def on_limpiar_click():

    """Limpia el campo y mensaje."""

    entry_cedula.delete(0, tk.END)                  # Borrar contenido
del Entry

    label_resultado.config(text="")                 # Limpiar etiqueta de
resultado
```

```
    entry_cedula.focus_set()                          # Enfocar nuevamente
el campo

def on_enter_key(event):

    """Permite validar con la tecla ENTER."""

    on_validar_click()                                # Relanza la
validación

root = tk.Tk()                                        # Crear ventana
principal

root.title("Validador de Cédula - Ecuador")           # Título de la
ventana

root.geometry("420x250")                              # Tamaño fijo

root.resizable(False, False)                          # Evitar
redimensionado

style = ttk.Style()                                   # Crear objeto estilo

style.configure("TFrame", background="#F3F4F6")       # Fondo suave

style.configure("TLabel", background="#F3F4F6",

                font=("Segoe UI", 11))                # Fuente moderna

style.configure("TButton",

                font=("Segoe UI", 10, "bold"),

                padding=5)                            # Botones estilizados

style.map("TButton",

          foreground=[("pressed", "white"), ("active", "black")],

          background=[("pressed", "#2563EB"), ("active", "#93C5FD")])  #
Colores dinámicos

# ---------- MARCO PRINCIPAL ----------

frame = ttk.Frame(root, padding=20)                   # Contenedor con
padding

frame.pack(expand=True, fill="both")                  # Expandir en ventana

# ---------- TÍTULO ----------

label_titulo = ttk.Label(
```

```
    frame,
    text="Validador de Cédula Ecuatoriana",
    font=("Segoe UI", 14, "bold")
)
label_titulo.grid(row=0, column=0, columnspan=3, pady=(0, 15))
# ---------- INSTRUCCIÓN ----------
label_instruccion = ttk.Label(
    frame,
    text="Ingrese una cédula válida (10 dígitos):"
)
label_instruccion.grid(row=1, column=0, columnspan=3, sticky="w")
# ---------- CAMPO DE ENTRADA ----------
entry_cedula = ttk.Entry(frame, width=25, font=("Segoe UI", 12))
entry_cedula.grid(row=2, column=0, columnspan=3, pady=10)
entry_cedula.focus_set()                            # Cursor en el campo
entry_cedula.bind("<Return>", on_enter_key)         # Permitir validar
con Enter
# ---------- BOTONES ----------
btn_validar = ttk.Button(frame, text="Validar", command=on_validar_click)
btn_validar.grid(row=3, column=0, padx=5, pady=10, sticky="e")
btn_limpiar = ttk.Button(frame, text="Limpiar", command=on_limpiar_click)
btn_limpiar.grid(row=3, column=2, padx=5, pady=10, sticky="w")
# ---------- RESULTADO ----------
label_resultado = ttk.Label(frame, text="", font=("Segoe UI", 11))
label_resultado.grid(row=4, column=0, columnspan=3, pady=10)
# ---------- INICIAR PROGRAMA ----------
root.mainloop()
```