

# UNIVERSIDAD DE GUADALAJARA



## CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

### Análisis de Algoritmos

#### Reporte de práctica

Nombre del alumno:	Samuel David Pérez Brambila
Código del alumno:	222966286
Profesor:	Erasmus Gabriel Martínez Soltero
Título de la práctica:	"GUI con Tkinter"
Fecha:	10 de Octubre de 2024

# Metodología

## Código:

```
Spyder (Python 3.12)
Archivo Editar Buscar Código fuente Ejecutar Depurar Terminales Proyectos Herramientas Ver Ayuda

C:\Users\samuel\OneDrive\Escritorio\Archivos UDG\S\Análisis de Algoritmos\Tarea6.py

Tarea6.py X
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 2 14:29:03 2024
4
5 @author: SamuelBrambila
6 """
7
8 from tkinter import *
9 import re
10
11 class aplicacion():
12     def __init__(self):
13         self.raiz = Tk()
14         self.raiz.geometry('800x500')
15         self.raiz.resizable(width=False,height=False)
16         self.raiz.title('Expresiones regulares')
17         label = Label(self.raiz,text="Validación de expresiones regulares")
18         label.pack(side=TOP)
19
20         self.textos = Frame(self.raiz)
21         self.textos.pack(side=TOP)
22         self.frameDeAbajo = Frame(self.raiz)
23         self.frameDeAbajo.pack(side=BOTTOM)
24
25         Label(self.textos, text="Dirección IPv4:").grid(row=0, column=0, padx=10, pady=5, sticky=W)
26         Label(self.textos, text="Correo electrónico:").grid(row=1, column=0, padx=10, pady=5, sticky=W)
27         Label(self.textos, text="Número de teléfono:").grid(row=2, column=0, padx=10, pady=5, sticky=W)
28         Label(self.textos, text="Código postal:").grid(row=3, column=0, padx=10, pady=5, sticky=W)
29
30         self.t1=Entry(self.textos,width=40)
31         self.t1.grid(row=0,column=1,padx=10,pady=10)
32         self.t2=Entry(self.textos,width=40)
33         self.t2.grid(row=1,column=1)
34         self.t3=Entry(self.textos,width=40)
35         self.t3.grid(row=2,column=1)
36         self.t4=Entry(self.textos,width=40)
37         self.t4.grid(row=3,column=1)
38
39         self.b1=Button(self.textos,text='Validar',command=lambda:self.validar(1))
40         self.b1.grid(row=0,column=2,padx=10,pady=10)
41         self.b2=Button(self.textos,text='Validar',command=lambda:self.validar(2))
42         self.b2.grid(row=1,column=2,pady=10)
43         self.b3=Button(self.textos,text='Validar',command=lambda:self.validar(3))
44         self.b3.grid(row=2,column=2,pady=10)
45         self.b4=Button(self.textos,text='Validar',command=lambda:self.validar(4))
46         self.b4.grid(row=3,column=2,pady=10)
```

```
Spyder (Python 3.12)
Archivo Editar Buscar Código fuente Ejecutar Depurar Terminales Proyectos Herramientas Ver Ayuda

C:\Users\samuel\OneDrive\Escritorio\Archivos UDG\S\Análisis de Algoritmos\Tarea6.py

Tarea6.py X
48 self.l1=Label(self.textos,text='...')
49 self.l1.grid(row=0,column=3)
50 self.l2=Label(self.textos,text='...')
51 self.l2.grid(row=1,column=3)
52 self.l3=Label(self.textos,text='...')
53 self.l3.grid(row=2,column=3)
54 self.l4=Label(self.textos,text='...')
55 self.l4.grid(row=3,column=3)
56
57 # Text Area para contar palabras (función adicional)
58 Label(self.textos, text="Ingrese texto para contar las palabras:").grid(row=4, column=0, padx=10, pady=5, sticky=W)
59 self.text_area = Text(self.textos, height=5, width=40)
60 self.text_area.grid(row=5, column=1, padx=10, pady=5)
61 self.word_count_btn = Button(self.textos, text="Contar Palabras", command=self.contar_palabras)
62 self.word_count_btn.grid(row=6, column=1, pady=10)
63 self.word_count_label = Label(self.textos, text="Total palabras: 0")
64 self.word_count_label.grid(row=6, column=2)
65
66 # Checkboxes para negritas y cursivas (función adicional)
67 self.bold_var = IntVar()
68 self.italic_var = IntVar()
69 self.bold_check = Checkbutton(self.textos, text="Negritas", variable=self.bold_var, command=self.aplicar_estilos)
70 self.bold_check.grid(row=7, column=1, sticky=W)
71 self.italic_check = Checkbutton(self.textos, text="Cursiva", variable=self.italic_var, command=self.aplicar_estilos)
72 self.italic_check.grid(row=7, column=1, sticky=E)
73
74 # Botones para cerrar ventana y limpiar lo ingresado en los campos de texto
75 self.bsair=Button(self.frameDeAbajo,text="Salir",command=self.raiz.destroy)
76 self.bsair.pack(side=LEFT, padx=10, pady=10)
77 self.blimpiar=Button(self.frameDeAbajo,text="Limpiar",command=self.limpiar)
78 self.blimpiar.pack(side=LEFT, padx=10, pady=10)
79
80 self.raiz.mainloop()
81
82 def limpiar(self):
83     self.t1.delete(first=0, last='end')
84     self.t2.delete(first=0, last='end')
85     self.t3.delete(first=0, last='end')
86     self.t4.delete(first=0, last='end')
87     self.l1.config(fg='black', text='...')
88     self.l2.config(fg='black', text='...')
89     self.l3.config(fg='black', text='...')
90     self.l4.config(fg='black', text='...')
91     self.text_area.delete(1.0, 'end')
92     self.word_count_label.config(text="Total palabras: 0")
93
```

Spyder (Python 3.12)

Archivo Editar Buscar Código fuente Ejecutar Depurar Terminales Proyectos Herramientas Ver Ayuda

C:\Users\samue\OneDrive\Escritorio\Archivos UDGS\Análisis de Algoritmos\Tarea6.py

Tarea6.py

```
94 def validar(self, numero):
95     if (numero==1):
96         txtAValidar=self.t1.get()
97         x=re.search("(25[0-5]|2[0-4]\d|[01]?d\d?)\.\.){3}(25[0-5]|2[0-4]\d|[01]?d\d?)$)",txtAValidar)
98         if (x):
99             self.l1.config(fg="green",text="IPv4 válida")
100         else:
101             self.l1.config(fg="red",text="IPv4 inválida")
102     elif (numero==2):
103         txtAValidar=self.t2.get()
104         x=re.search("^([a-zA-Z0-9_-]+)+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-]+$",txtAValidar)
105         if (x):
106             self.l2.config(fg="green",text="Correo válido")
107         else:
108             self.l2.config(fg="red",text="Correo inválido")
109     elif (numero==3):
110         txtAValidar=self.t3.get()
111         x=re.search("^([d|0|5|9])$",txtAValidar)
112         if (x):
113             self.l3.config(fg="green",text="Teléfono válido")
114         else:
115             self.l3.config(fg="red",text="Teléfono inválido")
116     else:
117         txtAValidar=self.t4.get()
118         x=re.search("^([d|0|5])$",txtAValidar)
119         if (x):
120             self.l4.config(fg="green",text="Código postal válido")
121         else:
122             self.l4.config(fg="red",text="Código postal inválido")
123
124 def contar_palabras(self):
125     # Contar palabras en el Text Area
126     texto = self.text_area.get(1.0, END)
127     palabras = len(texto.split())
128     self.word_count_label.config(text=f"Total palabras: {palabras}")
129
130 def aplicar_estilos(self):
131     # Aplicar estilos a Entry
132     bold = self.bold_var.get() == 1
133     italic = self.italic_var.get() == 1
134
135     # Estilo para el Entry
136     if bold and italic:
137         font_style = 'bold italic'
138     elif bold:
139         font_style = 'bold'
140     elif italic:
141         font_style = 'italic'
142     else:
143         font_style = 'normal'
144
145     # Aplicar estilo a los Entry
146     self.t1.config(font=(None, 10, font_style))
147     self.t2.config(font=(None, 10, font_style))
148     self.t3.config(font=(None, 10, font_style))
149     self.t4.config(font=(None, 10, font_style))
150
151     # Aplicar estilos al Text Area
152     self.text_area.configure('bold', font=('Helvetica', 10, 'bold'))
153     self.text_area.tag_configure('italic', font=('Helvetica', 10, 'italic'))
154     self.text_area.tag_configure('bold italic', font=('Helvetica', 10, 'bold italic'))
155     self.text_area.tag_configure('normal', font=('Helvetica', 10, 'normal'))
156
157     # Determinar qué estilo aplicar en el Text Area
158     self.text_area.tag_remove('bold', '1.0', 'end')
159     self.text_area.tag_remove('italic', '1.0', 'end')
160     self.text_area.tag_remove('bold italic', '1.0', 'end')
161     self.text_area.tag_remove('normal', '1.0', 'end')
162
163     if bold and italic:
164         self.text_area.tag_add('bold italic', '1.0', 'end')
165     elif bold:
166         self.text_area.tag_add('bold', '1.0', 'end')
167     elif italic:
168         self.text_area.tag_add('italic', '1.0', 'end')
169     else:
170         self.text_area.tag_add('normal', '1.0', 'end')
171
172 app = aplicacion()
```

Nombre Última modificación

- anaconda 13/08/2024 05:07 p. m.
- astrophy 08/05/2024 10:13 a. m.
- conda 13/08/2024 05:09 p. m.
- config 18/04/2024 10:26 p. m.
- continuum 13/08/2024 05:06 p. m.
- idlerc 23/04/2024 08:55 p. m.
- python 09/08/2024 07:49 p. m.
- jupyter 27/08/2024 07:49 p. m.
- local 18/04/2024 10:53 p. m.
- matplotlib 23/04/2024 09:45 p. m.
- quokka 15/02/2024 05:18 p. m.
- shiv 18/04/2024 10:53 p. m.
- spyder-py3 02/10/2024 10:59 p. m.
- vscode 15/02/2024 05:09 p. m.
- wallaby 15/02/2024 05:18 p. m.
- Xilinx 20/02/2024 12:39 a. m.

Terminal I/A X

Python 3.12.3 | packaged by conda-forge | (main, Apr 15 2024, 18:20:11) [MSC v.1938 64 bit (AMD64)]  
Type "copyright", "credits" or "license" for more information.

IPython 8.25.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/samue/OneDrive/Escritorio/Archivos UDGS/Análisis de Algoritmos/Tarea6.py', wdir='C:/Users/samue/OneDrive/Escritorio/Archivos UDGS/Análisis de Algoritmos')  
<unknown>-97: SyntaxWarning: invalid escape sequence '\d'

conda: base (Python 3.12.3) Completions: conda LSP: Python Line 57, Col 61 UTF-8 CRLF RW Mem 77%

Spyder (Python 3.12)

Archivo Editar Buscar Código fuente Ejecutar Depurar Terminales Proyectos Herramientas Ver Ayuda

C:\Users\samue\OneDrive\Escritorio\Archivos UDGS\Análisis de Algoritmos\Tarea6.py

Tarea6.py

```
120 self.word_count_label.config(text=f"Total palabras: {palabras}")
121
122 def aplicar_estilos(self):
123     # Aplicar estilos a Entry
124     bold = self.bold_var.get() == 1
125     italic = self.italic_var.get() == 1
126
127     # Estilo para el Entry
128     if bold and italic:
129         font_style = 'bold italic'
130     elif bold:
131         font_style = 'bold'
132     elif italic:
133         font_style = 'italic'
134     else:
135         font_style = 'normal'
136
137     # Aplicar estilo a los Entry
138     self.t1.config(font=(None, 10, font_style))
139     self.t2.config(font=(None, 10, font_style))
140     self.t3.config(font=(None, 10, font_style))
141     self.t4.config(font=(None, 10, font_style))
142
143     # Aplicar estilos al Text Area
144     self.text_area.configure('bold', font=('Helvetica', 10, 'bold'))
145     self.text_area.tag_configure('italic', font=('Helvetica', 10, 'italic'))
146     self.text_area.tag_configure('bold italic', font=('Helvetica', 10, 'bold italic'))
147     self.text_area.tag_configure('normal', font=('Helvetica', 10, 'normal'))
148
149     # Determinar qué estilo aplicar en el Text Area
150     self.text_area.tag_remove('bold', '1.0', 'end')
151     self.text_area.tag_remove('italic', '1.0', 'end')
152     self.text_area.tag_remove('bold italic', '1.0', 'end')
153     self.text_area.tag_remove('normal', '1.0', 'end')
154
155     if bold and italic:
156         self.text_area.tag_add('bold italic', '1.0', 'end')
157     elif bold:
158         self.text_area.tag_add('bold', '1.0', 'end')
159     elif italic:
160         self.text_area.tag_add('italic', '1.0', 'end')
161     else:
162         self.text_area.tag_add('normal', '1.0', 'end')
163
164 app = aplicacion()
```

Nombre Última modificación

- anaconda 13/08/2024 05:07 p. m.
- astrophy 08/05/2024 10:13 a. m.
- conda 13/08/2024 05:09 p. m.
- config 18/04/2024 10:26 p. m.
- continuum 13/08/2024 05:06 p. m.
- idlerc 23/04/2024 08:55 p. m.
- python 09/08/2024 07:49 p. m.
- jupyter 27/08/2024 07:49 p. m.
- local 18/04/2024 10:53 p. m.
- matplotlib 23/04/2024 09:45 p. m.
- quokka 15/02/2024 05:18 p. m.
- shiv 18/04/2024 10:53 p. m.
- spyder-py3 02/10/2024 10:59 p. m.
- vscode 15/02/2024 05:09 p. m.
- wallaby 15/02/2024 05:18 p. m.
- Xilinx 20/02/2024 12:39 a. m.

Terminal I/A X

Python 3.12.3 | packaged by conda-forge | (main, Apr 15 2024, 18:20:11) [MSC v.1938 64 bit (AMD64)]  
Type "copyright", "credits" or "license" for more information.

IPython 8.25.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/samue/OneDrive/Escritorio/Archivos UDGS/Análisis de Algoritmos/Tarea6.py', wdir='C:/Users/samue/OneDrive/Escritorio/Archivos UDGS/Análisis de Algoritmos')  
<unknown>-97: SyntaxWarning: invalid escape sequence '\d'

conda: base (Python 3.12.3) Completions: conda LSP: Python Line 57, Col 61 UTF-8 CRLF RW Mem 77%

# Resultados

## Interfaz inicial

The screenshot shows the initial state of the 'Expresiones regulares' application. The window title is 'Expresiones regulares'. The main heading is 'Validación de expresiones regulares'. There are four input fields for validation: 'Dirección IPv4:', 'Correo electrónico:', 'Número de teléfono:', and 'Código postal:'. Each field has a 'Validar' button and a status indicator '...'. Below these is a larger text area labeled 'Ingrese texto para contar las palabras:'. At the bottom, there is a 'Contar Palabras' button, a 'Total palabras: 0' label, and two checkboxes for 'Negritas' and 'Cursiva'. At the very bottom are 'Salir' and 'Limpiar' buttons.

## Interfaz con validaciones y conteo de palabras

The screenshot shows the application with sample data entered. The 'Dirección IPv4:' field contains '192.168.100.1' and shows 'IPv4 válida' in green. The 'Correo electrónico:' field contains 'samuelbrambila34@gmail.com' and shows 'Correo válido' in green. The 'Número de teléfono:' field contains '3314678' and shows 'Teléfono inválido' in red. The 'Código postal:' field contains '4513' and shows 'Código postal inválido' in red. The text area contains 'Hola esto es un texto de prueba'. The 'Contar Palabras' button is active, and the 'Total palabras:' label now shows '7'. The 'Negritas' and 'Cursiva' checkboxes remain unchecked. The 'Salir' and 'Limpiar' buttons are still at the bottom.

Interfaz con cambio de texto a negritas y cursiva (puede seleccionarse 1, los 2 o ninguno)

Expresiones regulares

Validación de expresiones regulares

Dirección IPv4:	<input type="text" value="192.168.100.1"/>	<input type="button" value="Validar"/>	IPv4 válida
Correo electrónico:	<input type="text" value="samuelbrambila34@gmail.com"/>	<input type="button" value="Validar"/>	Correo válido
Número de teléfono:	<input type="text" value="3314678"/>	<input type="button" value="Validar"/>	Teléfono inválido
Código postal:	<input type="text" value="4513"/>	<input type="button" value="Validar"/>	Código postal inválido

Ingrese texto para contar las palabras:

Total palabras: 7

☒ Negritas ☒ Cursiva