

Región Metropolitana  
Facultad: Ingeniería en Sistemas de Información  
Curso: Contabilidad II  
Sección: C  
Ciclo: 2



Jonathan Joel Istupe Martínez  
Carné: 7690-23-15804  
Julio Alexander Alvarado Morales  
Carné: 7690-23-13706  
Bryan Josue Alvarez Lopez  
Carné: 7690-23-16816

Fecha: 07/11/2023

# MANUAL DE LA APLICACIÓN

## EDITOR DE TEXTO

### (PROYECT II)

## DOCUMENTACION INTERNA

```
ProyectoIII (2).py X
C: > Users > SINISTER > Downloads > ProyectoIII (2).py > TextEditorApp
1  import tkinter as tk
2  import webbrowser
3  from tkinter import filedialog, messagebox, ttk, simpledialog, colorchooser, scrolledtext
4
5  class TextEditorApp:
6      def __init__(self, root):
7          self.root = root
8          self.root.title("Text Editor")
9
10         self.text_area = scrolledtext.ScrolledText(self.root, wrap="word", undo=True, bg="black", fg="white")
11         self.text_area.pack(expand="yes", fill="both")
12
13         self.root.configure(bg="black")
14
15         self.menu_bar = tk.Menu(self.root, bg="black", fg="white", activebackground="white", activeforeground="black")
16         self.root.config(menu=self.menu_bar)
```

**tkinter:** Se importa la biblioteca principal para crear interfaces gráficas de usuario.

**webbrowser:** Se importa para abrir enlaces web.

**filedialog, messagebox, ttk, simpledialog, colorchooser, scrolledtext:** Se importan módulos específicos de tkinter para diversas funcionalidades.

Se define la clase **TextEditorApp**, que es la aplicación principal del editor de texto.

El método **\_\_init\_\_** de la clase **TextEditorApp** se encarga de la inicialización de la aplicación:

Se crea una ventana principal (**root**) con el título "Text Editor".

Se crea un área de texto desplazable (**text\_area**) donde el usuario puede escribir y editar texto.

```

18     self.file_menu = tk.Menu(self.menu_bar, tearoff=0)
19     self.menu_bar.add_cascade(label="Archivo", menu=self.file_menu)
20     self.file_menu.add_command(label="Abrir", command=self.open_file)
21     self.file_menu.add_command(label="Guardar", command=self.save_file)
22     self.file_menu.add_command(label="Guardar como...", command=self.save_file_as)
23     self.file_menu.add_separator()
24     self.file_menu.add_command(label="Salir", command=self.root.destroy)
25
26     self.edit_menu = tk.Menu(self.menu_bar, tearoff=0)
27     self.menu_bar.add_cascade(label="Editar", menu=self.edit_menu)
28     self.edit_menu.add_command(label="Deshacer", command=self.undo)
29     self.edit_menu.add_command(label="Rehacer", command=self.redo)
30
31     self.format_menu = tk.Menu(self.menu_bar, tearoff=0)
32     self.menu_bar.add_cascade(label="Formato", menu=self.format_menu)
33     self.format_menu.add_command(label="Cambiar tamaño de letra", command=self.change_font_size)
34     self.format_menu.add_command(label="Cambiar color de texto", command=self.change_text_color)

```

Realiza el menú no.1 - menú no.2- menú no.3- menú no.4 esta aparte del código les asigna un nombre y una función

La función sería la que está circulada en color verde, donde las llama.

```

def open_file(self):
    file_path = filedialog.askopenfilename(filetypes=[("Archivos de texto", "*.txt"), ("Todos los archivos", "*.*")])
    if file_path:
        with open(file_path, "r") as file:
            content = file.read()
            self.text_area.delete("1.0", tk.END)
            self.text_area.insert(tk.END, content)
            self.file_path = file_path

def save_file(self):
    if self.file_path:
        content = self.text_area.get("1.0", tk.END)
        with open(self.file_path, "w") as file:
            file.write(content)
    else:
        self.save_file_as()

def save_file_as(self):
    file_path = filedialog.asksaveasfilename(defaulttextextension=".txt", filetypes=[("Archivos de texto", "*.txt"), ("Todos los archivos", "*.*")])
    if file_path:
        self.file_path = file_path
        self.save_file()

```

Se inicializa la variable `file_path` para llevar un registro de la ubicación del archivo actualmente abierto o guardado.

```

style = ttk.Style()
style.configure("TButton", padding=(5, 5, 5, 5), font=('Helvetica', 10))

self.button_font_size = ttk.Button(self.root, text="Tamaño de letra", command=self.change_font_size)
self.button_font_size.pack(side=tk.LEFT, padx=5)

self.button_text_color = ttk.Button(self.root, text="Color de texto", command=self.change_text_color)
self.button_text_color.pack(side=tk.LEFT, padx=5)

self.file_path = None

```

```

76     def undo(self):
77         self.text_area.edit_undo()
78
79     def redo(self):
80         self.text_area.edit_redo()
81
82     def change_font_size(self):
83         size_input = simpledialog.askinteger("Cambiar tamaño de letra", "Ingrese el nuevo tamaño de letra:", parent=self.root)
84
85         if size_input:
86             current_font = self.text_area.cget("font")
87             new_font = (current_font[0], size_input)
88             self.text_area.configure(font=new_font)
89
90     def change_text_color(self):
91         color = colorchooser.askcolor()[1]
92         if color:
93             self.text_area.tag_configure("colored", foreground=color)
94             current_index = self.text_area.index(tk.SEL_FIRST)
95             if current_index:
96                 self.text_area.tag_add("colored", current_index, tk.SEL_LAST)
97             else:
98                 self.text_area.tag_add("colored", "1.0", tk.END)

```

**undo:** Permite deshacer la última acción.

**redo:** Permite rehacer la última acción.

**change\_font\_size:** Permite al usuario cambiar el tamaño de la fuente del texto.

**change\_text\_color:** Permite al usuario cambiar el color del texto seleccionado.

```

100     def show_info(self):
101         app_info = "Información\n\nEsta aplicación es un editor de texto con una interfaz gráfica utilizando el lenguaje PYTHON y la libr
102         messagebox.showinfo("Información", app_info)
103
104     def open_manual(self):
105         repo_url = "https://github.com/Jonathanjips/Proyecto-III#readme"
106         webbrowser.open_new(repo_url)
107
108     def show_integrantes(self):
109         app_info = "Estudiantes\n\n Julio Alexander Alvarado Morales\n          7690-23-13706\n\nJonathan Joel Istupe Martinez\n          7
110         messagebox.showinfo("Estudiantes", app_info)
111         pass
112
113 if __name__ == "__main__":
114     root = tk.Tk()
115     app = TextEditorApp(root)

```

**show\_info:** Muestra información sobre la aplicación en una ventana de diálogo.

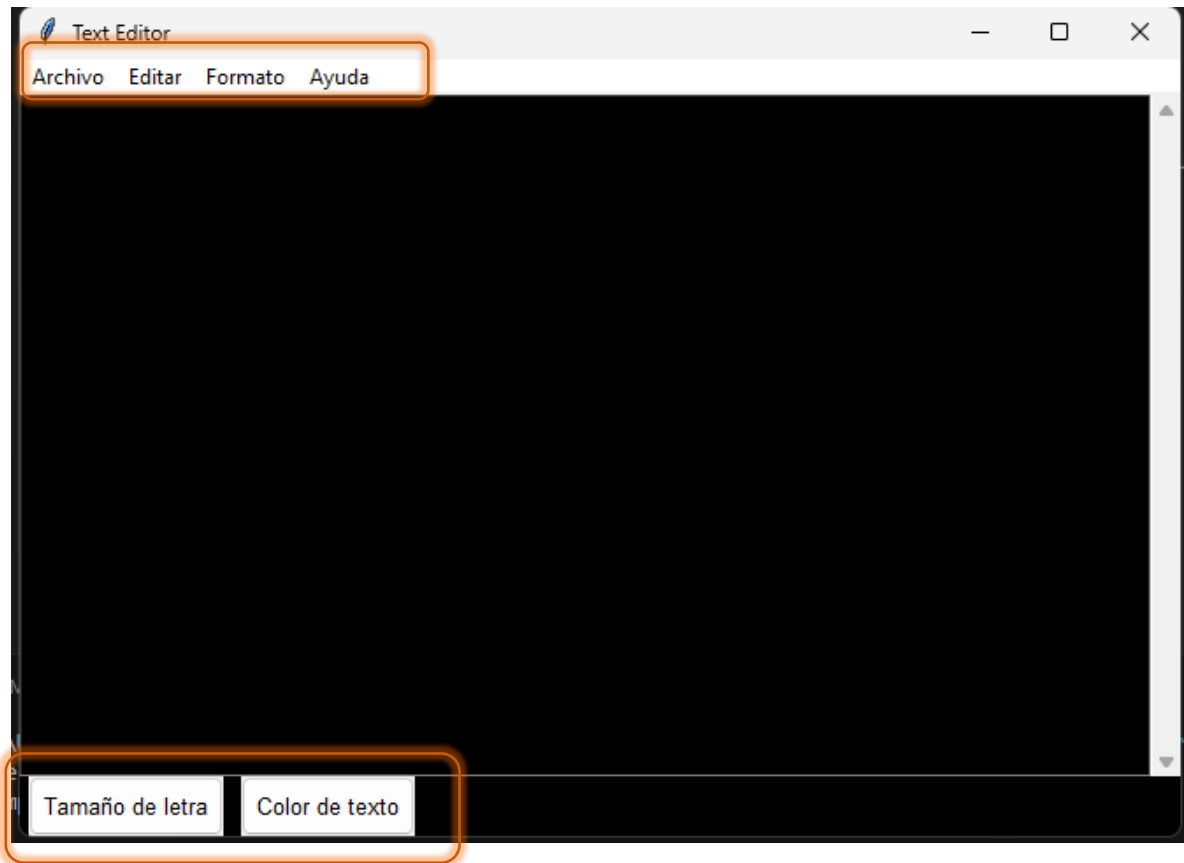
**open\_manual:** Abre un enlace web al manual de usuario de la aplicación.

**show\_integrantes:** Muestra información sobre los estudiantes que crearon la aplicación en una ventana de diálogo.

## DOCUMENTACION EXTERNA

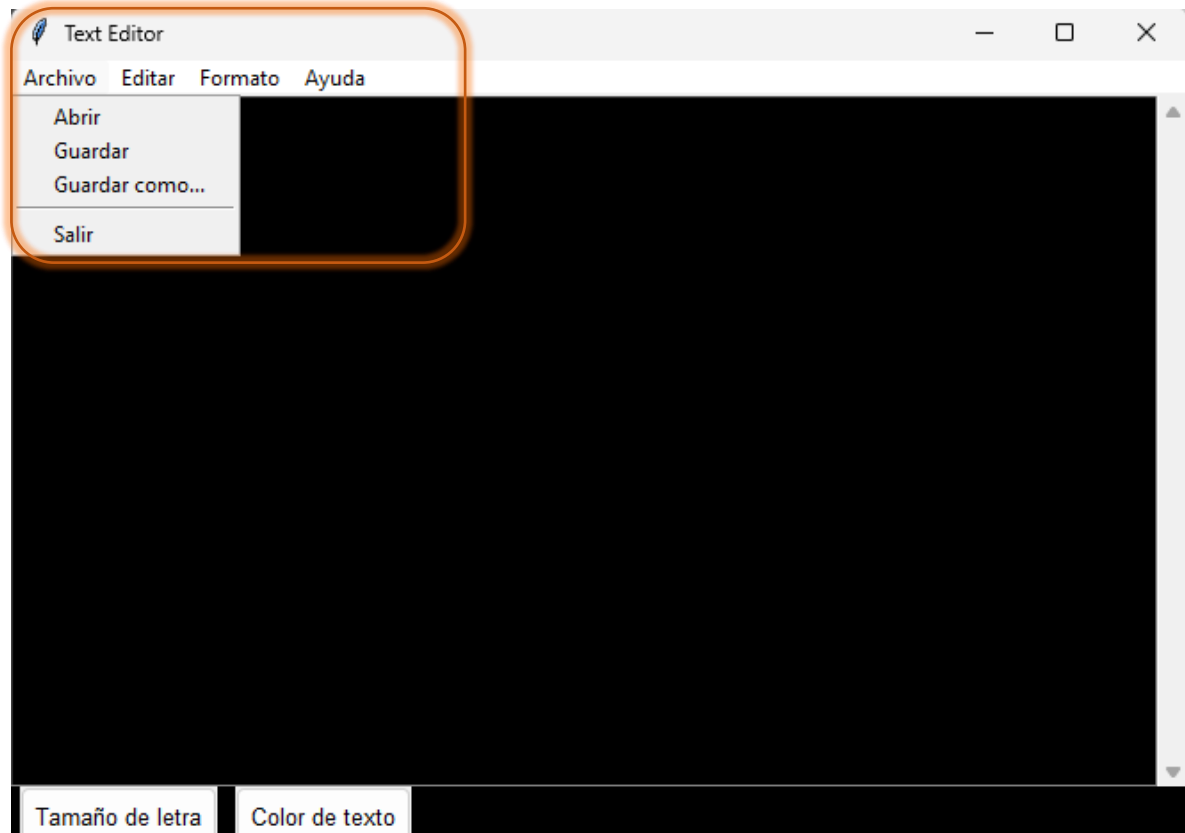
El editor de texto cuenta con varias funciones entre ellas esta:

Archivo, editar, formato y ayuda. Tambien se podrá modificar el tamaño de la letra y el color de texto.



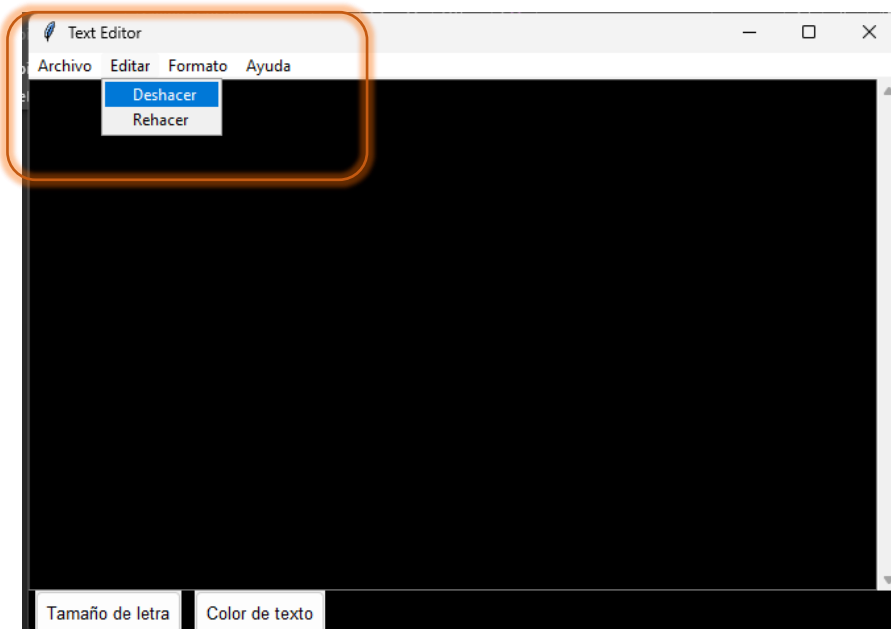
### Archivo:

En archivo se podrá abrir un documento en txt, .cpp, .cs, .py, entre otros, de igual manera podrá guardarlo, o guardar como en caso que sea nuestra primera vez guardando ese archivo. Y también salir.

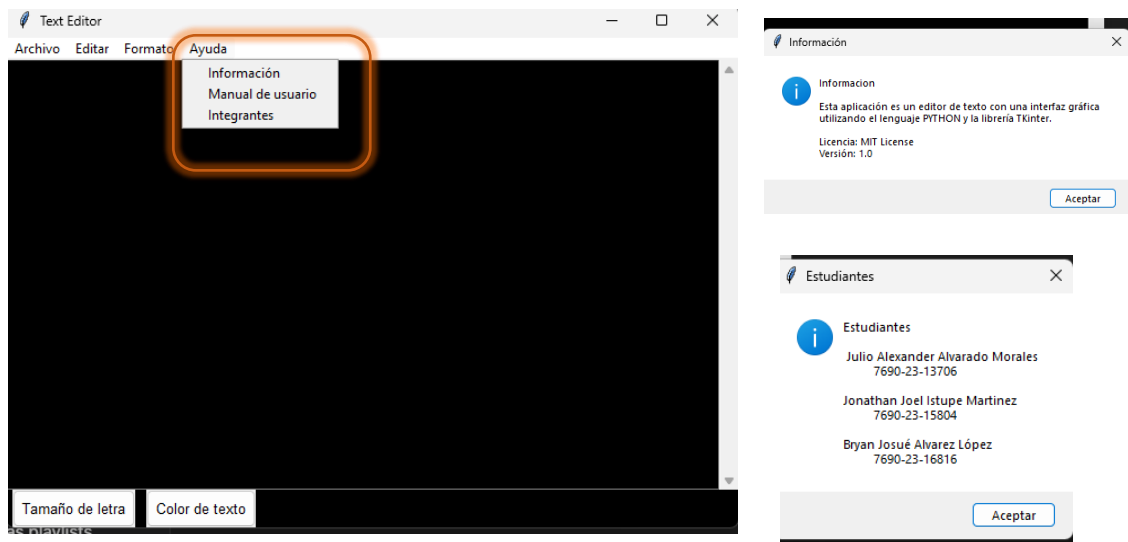


### Editar:

En la parte de editar, se podrá deshacer y rehacer el texto que se haya escrito.



## Ayuda:

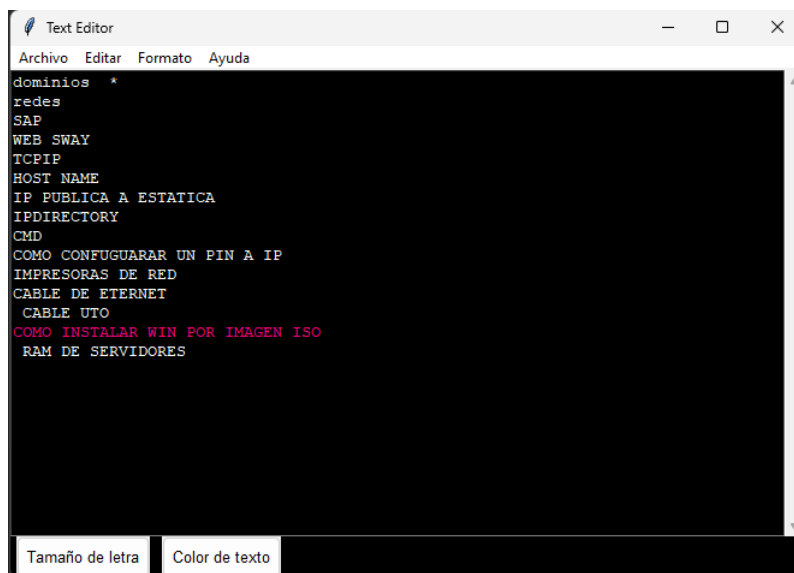


En el apartado de ayuda se podrá encontrar la información

El manual de usuario, donde el usuario será redirigido a Git Hub donde se encontrara la documentación interna y externa de programa Y en el apartado de los integrantes se informara los nombres y carnet de los que realizaron el programa.

## Cambiar de color el texto

Cuando se habrá un archivo y se quiera cambiar el color de la letra se tendrá que seleccionar el párrafo que se desea cambiar de color



Seguidamente seleccionar el botón de color de texto, y seleccionar el color deseado. De igual manera con el botón de cambiar el tamaño de letra.

