

Universidad Mariano Gálvez de Guatemala  
Faculta De Ingeniería En Sistemas De Información  
Sede: Boca del Monte, Villa Canales  
Catedrático: Mélvín Calí  
Curso: Algoritmos



**PROYECTO III APLICACIÓN CON INTERFAZ  
GRAFICA EN PYTHON**

Nombre	Carnet
Ruth Estefanía Pérez Damián	7690-22-14922
Crosvy Rolando Culajay Zabala	7690-22-18420
Jose Rodolfo Samayoa Matías	7690-22-17518
Kevin Antulio Poitán Cos	7690-22-3891



## INDICE DE CONTENIDO.

	Pág.
1. Introducción	03
2. ¿Cómo acceder aplicación Proyecto III?	04
3. Opciones de archivo campo	05
4. Opciones de campo archivo	06
5. Opciones de Editar	07
6. Opciones de Ayuda	08
7. Crear Nuevo Archivo	08
8. Uso de función copiar y pegar	09
9. Crear archivo C++	10
10. Crear archivo C#	11
11. Crear archivo Python	12
12. Información de interfaz gráfica	13
13. Vista de campo manual	14
14. Direccionamiento a GitHub	15
15. Cerrar aplicación	16



## INTRODUCCION

Este curso de algoritmos y proyecto III está orientado a programación en Python está orientado a usar interfaz graficas. En sus contenidos hemos condensado los temas más importantes de la programación de Python. Y nuestros integrantes aportaremos a los usuarios la base suficiente para que empiecen a utilizar la aplicación de forma correcta y puedan profundizar en todos los aspectos la App. Por lo cual haremos un manual y video para que nuestro usuario pueda tener un mejor uso con nuestra aplicación y plantearemos nuestro código y vistas de pantalla que ilustrarán los aspectos de nuestro proyecto de aplicación de interfaz gráfica. Esperando que este manual de usuario sea una guía para el buen uso de nuestro proyecto.



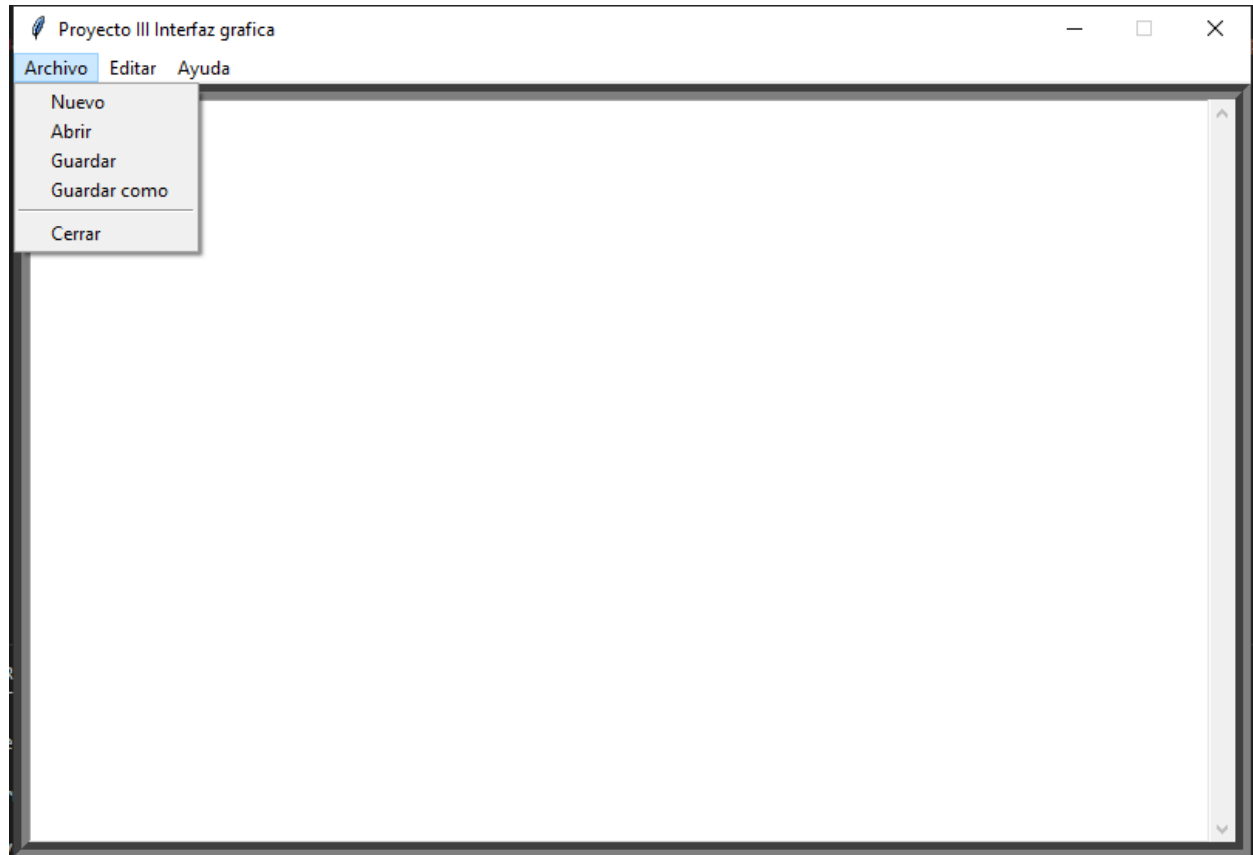
- Para poder acceder tenemos que compilar y ejecutar el código digitado en nuestra aplicación de IDE (Visual Studio Code) para poder genera una ventana que nos muestra nuestra interface gráfica.



- En nuestra aplicación veremos los siguientes campos:
  - Archivo
  - Editar
  - Ayuda



- En el primer campo encontraremos las siguientes opciones:



- **Nuevo**
- **Abrir**
- **Guardar**
- **Guardar Como**
- **Cerrar**



➤ En el segundo campo encontraremos las siguientes opciones:



- **Deshacer**
- **Rehacer**
- **Cortar**
- **Copiar**
- **Pegar**



- En el tercer campo encontraremos las siguientes opciones:



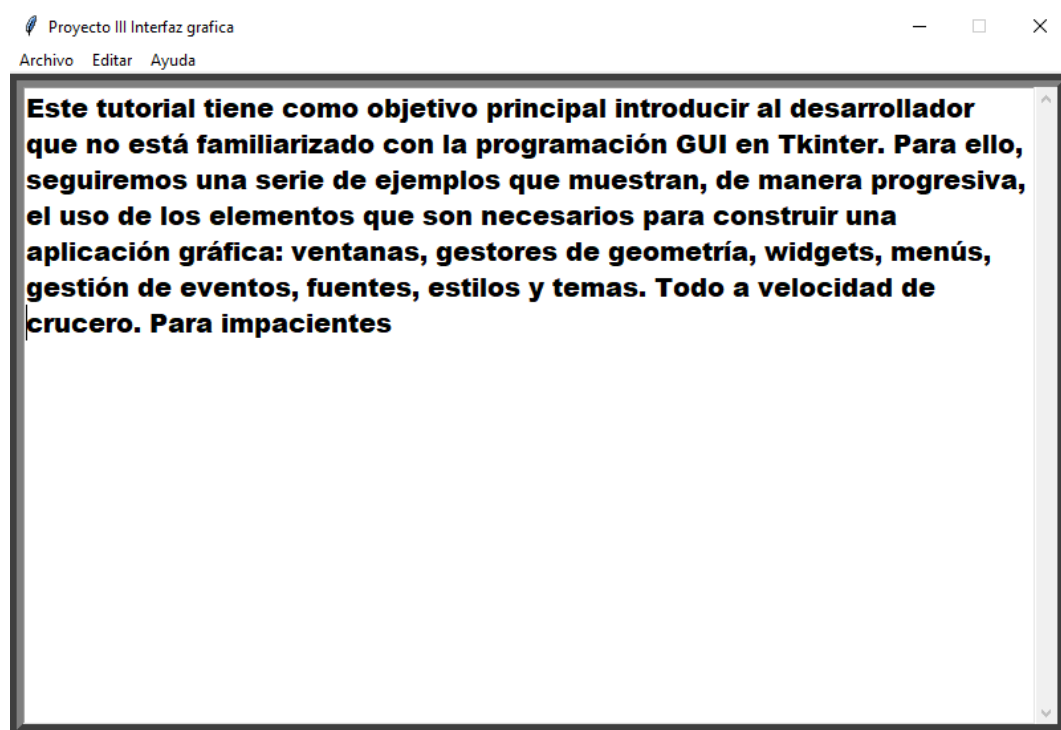
- Acerca de (encontraremos Información que se despegara al final del manual de usuario, donde encontraremos la información de la APP (versión, link de enlace a GitHub, integrantes del Proyecto III)



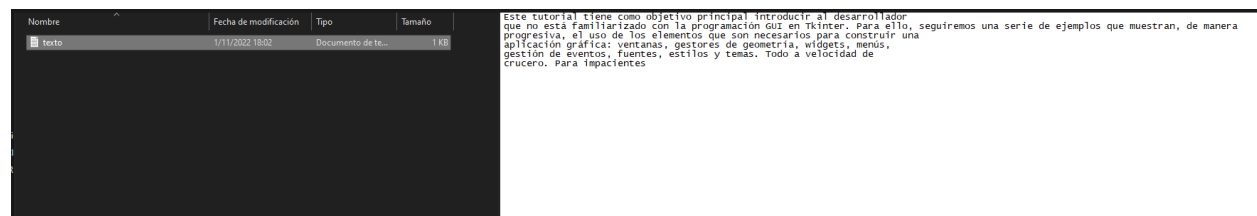
Al tener las ventas de nuestra aplicación empezaremos a mostrarle el uso de nuestra aplicación esto como a utilizar los campos de **Archivo, Editar y ayuda** dentro de los las opciones de (nuevo, guardar, deshacer, cortar etc.) para el buen uso de las mismas

En nuestro primer campo usaremos **Archivo** con la opción de **Nuevo**

Ingreso de archivo texto



Donde usaremos la opción de guardar el archivo texto.

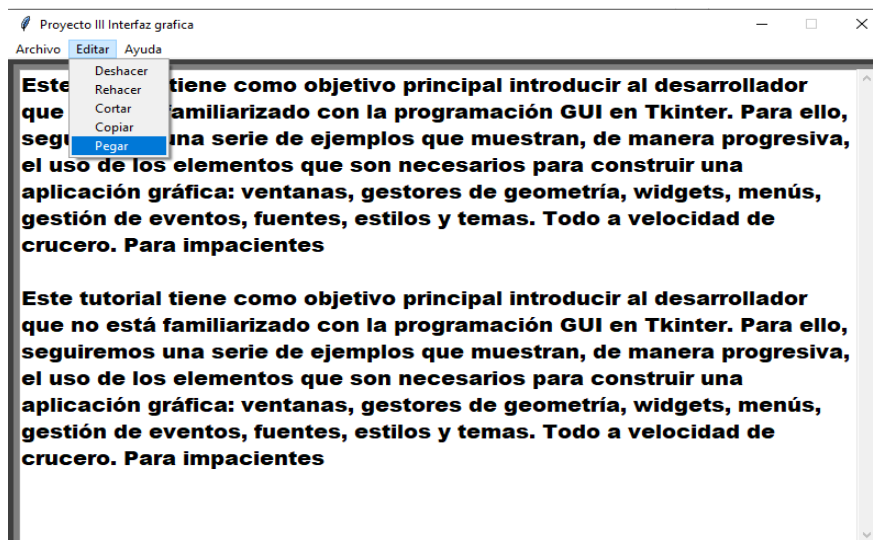
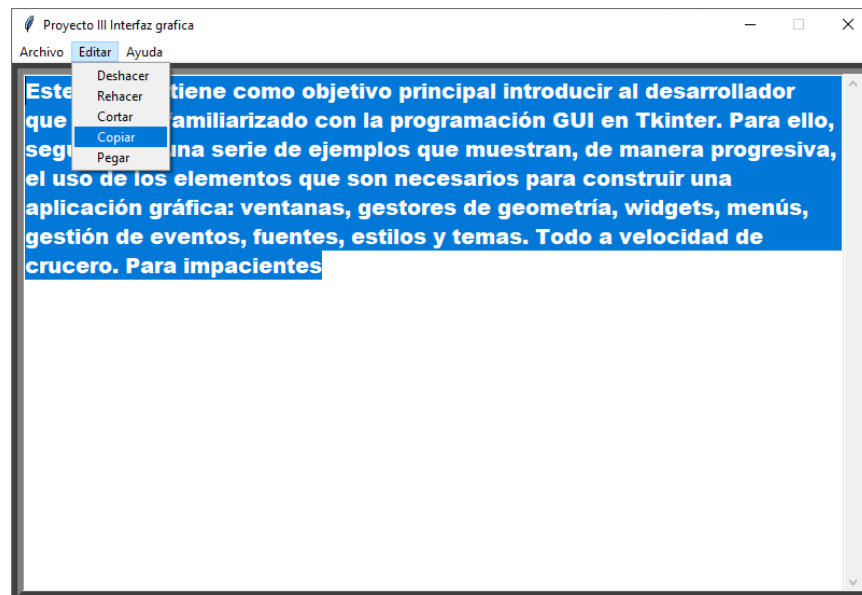


Archivo guardado con éxito.





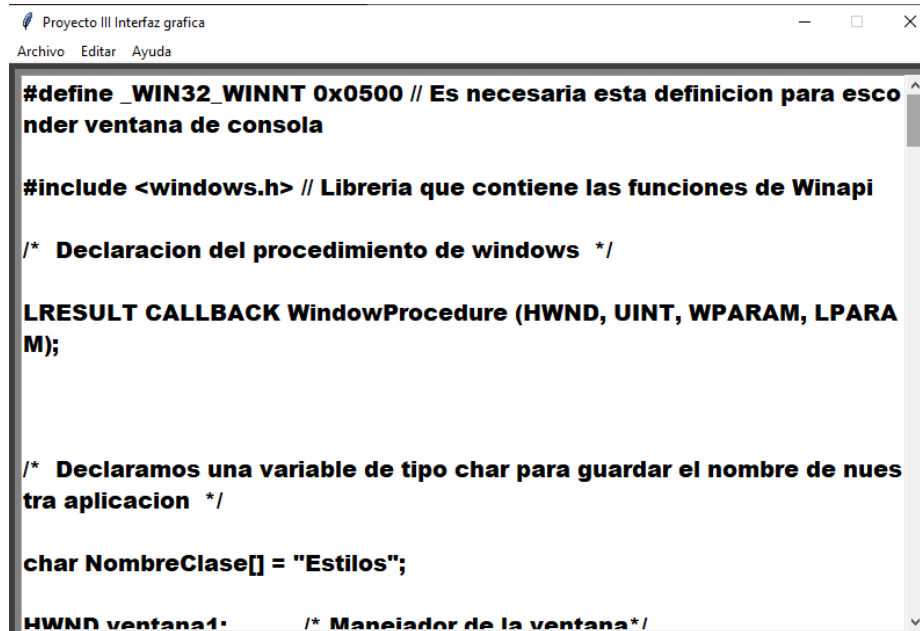
Usaremos la el segundo campo de Editar, con las opciones de copiar y pegar.



➤ Texto copiado y pegado con éxito.



## Se creará un archivo den C++ (cpp.)



```
#define _WIN32_WINNT 0x0500 // Es necesaria esta definicion para esconder ventana de consola

#include <windows.h> // Libreria que contiene las funciones de Winapi

/* Declaracion del procedimiento de windows */

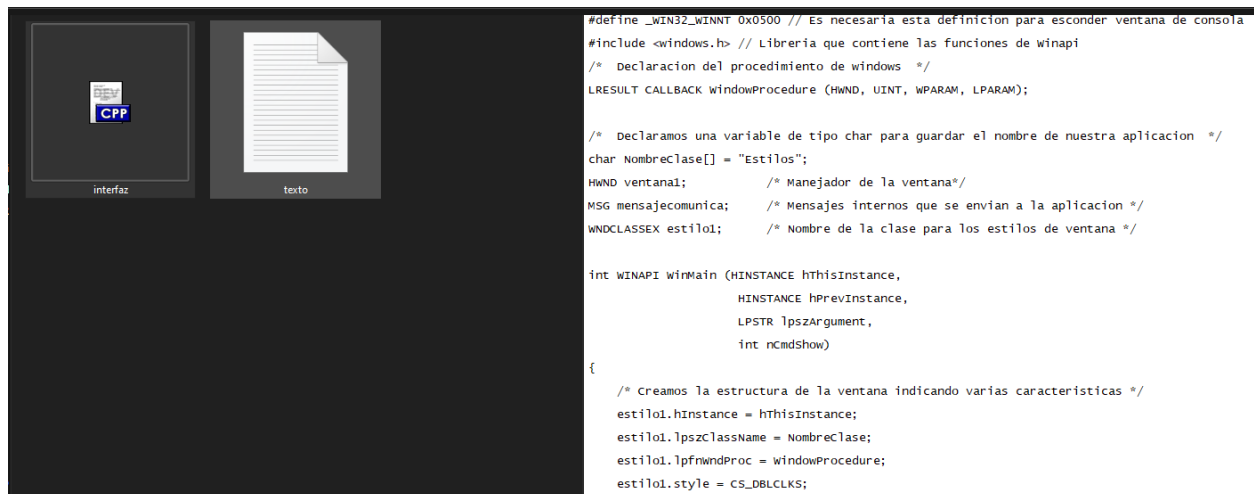
LRESULT CALLBACK WindowProcedure (HWND, UINT, WPARAM, LPARAM);

/* Declaramos una variable de tipo char para guardar el nombre de nuestra aplicacion */

char NombreClase[] = "Estilos";

HWND ventana1; /* Manejador de la ventana*/
```

## Archivo CPP creado con éxito



```
#define _WIN32_WINNT 0x0500 // Es necesaria esta definicion para esconder ventana de consola
#include <windows.h> // Libreria que contiene las funciones de Winapi
/* Declaracion del procedimiento de windows */
LRESULT CALLBACK WindowProcedure (HWND, UINT, WPARAM, LPARAM);

/* Declaramos una variable de tipo char para guardar el nombre de nuestra aplicacion */
char NombreClase[] = "Estilos";
HWND ventana1; /* Manejador de la ventana*/
MSG mensajecomunica; /* Mensajes internos que se envian a la aplicacion */
WNDCLASSEX estilo1; /* Nombre de la clase para los estilos de ventana */

int WINAPI WinMain (HINSTANCE hThisInstance,
                    HINSTANCE hPrevInstance,
                    LPSTR lpszArgument,
                    int nCmdShow)
{
    /* Creamos la estructura de la ventana indicando varias características */
    estilo1.hInstance = hThisInstance;
    estilo1.lpszClassName = NombreClase;
    estilo1.lpfnWndProc = WindowProcedure;
    estilo1.style = CS_DBLCLKS;
```



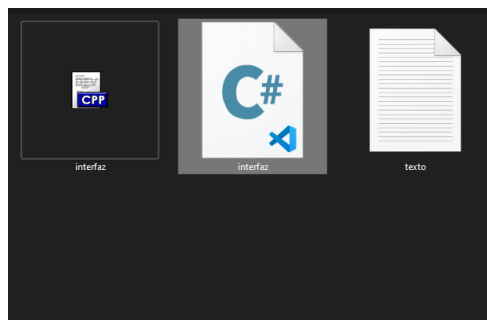
## Se creará archivo CS.

```
Proyecto III Interfaz grafica
Archivo Editar Ayuda

//Constructor
public FormMenuPrincipal()
{
    InitializeComponent();
    //Estas lineas eliminan los parpadeos del formulario o controles e
n la interfaz grafica (Pero no en un 100%)
    this.SetStyle(ControlStyles.ResizeRedraw, true);
    this.DoubleBuffered = true;
}

//RESIZE METODO PARA REDIMENSIONAR/CAMBIAR TAMAÑO A FORMU
LARIO EN TIEMPO DE EJECUCION -----
private int tolerance = 12;
private const int WM_NCHITTEST = 132;
private const int HTBOTTOMRIGHT = 17;
private Rectangle sizeGripRectangle;
protected override void WndProc(ref Message m)
{
    switch (m.Msg)
    {
```

Archivo creado con éxito.



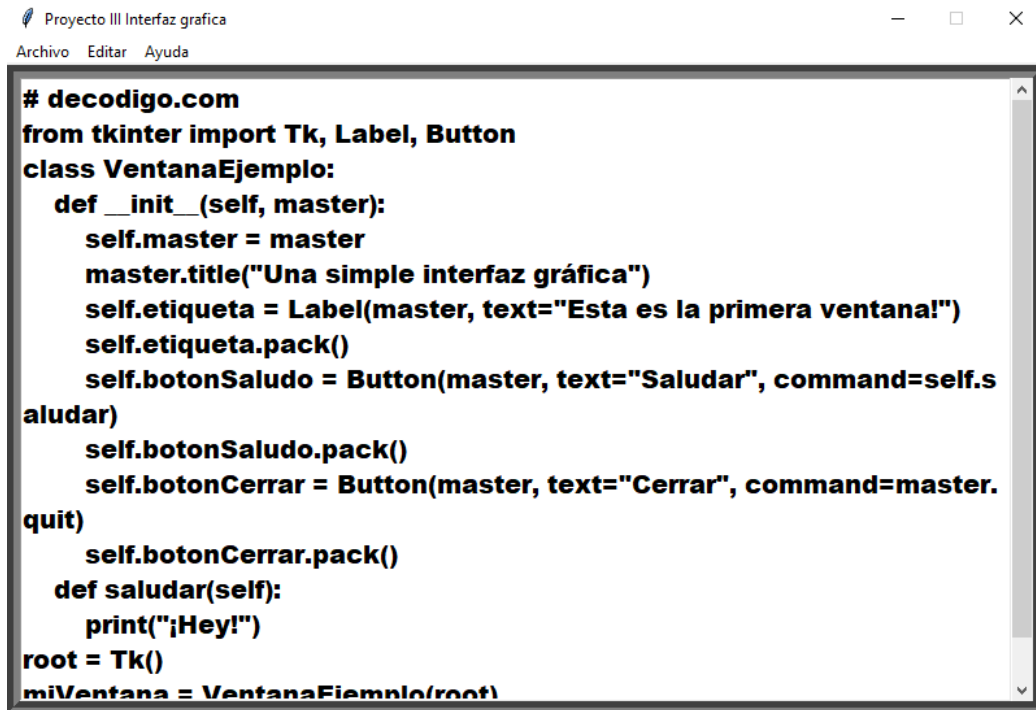
```
//Constructor
public FormMenuPrincipal()
{
    InitializeComponent();
    //Estas lineas eliminan los parpadeos del formulario o controles en la interfaz grafica (Pero no en un 100%)
    this.SetStyle(ControlStyles.ResizeRedraw, true);
    this.DoubleBuffered = true;
}

//RESIZE METODO PARA REDIMENSIONAR/CAMBIAR TAMAÑO A FORMULARIO EN TIEMPO DE EJECUCION
private int tolerance = 12;
private const int WM_NCHITTEST = 132;
private const int HTBOTTOMRIGHT = 17;
private Rectangle sizeGripRectangle;
protected override void WndProc(ref Message m)
{
    switch (m.Msg)
    {
        case WM_NCHITTEST:
            base.WndProc(ref m);
            var hitPoint = this.PointToClient(new Point(m.LParam.ToInt32() & 0xffff, m.LParam.ToInt32() >> 16));
            if (sizeGripRectangle.Contains(hitPoint))
            {
                m.Result = new IntPtr(HTBOTTOMRIGHT);
                break;
            }
        default:
            base.WndProc(ref m);
            break;
    }
}

//-----DIBUJAR RECTANGULO // EXCLUIR ESQUINA PANEL
protected override void OnSizeChanged(EventArgs e)
```

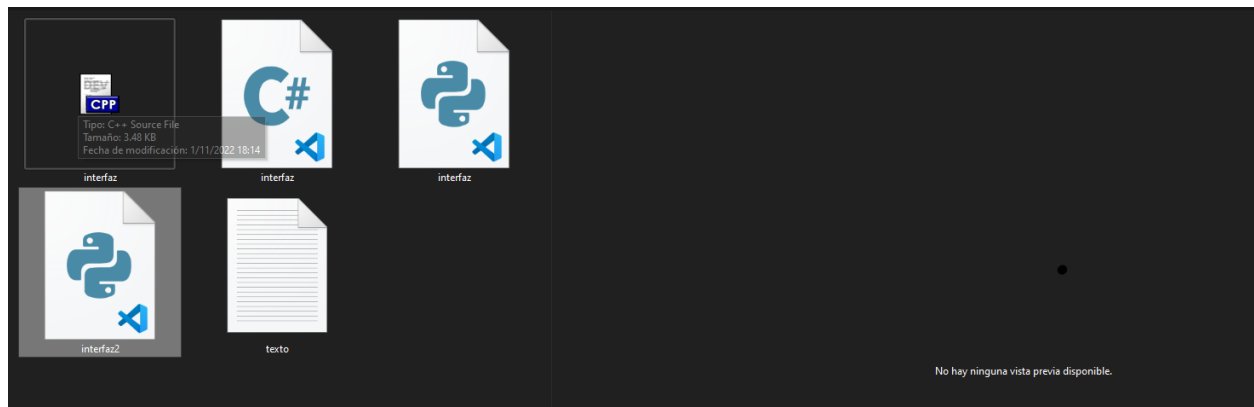


Se creará un archivo en Py.

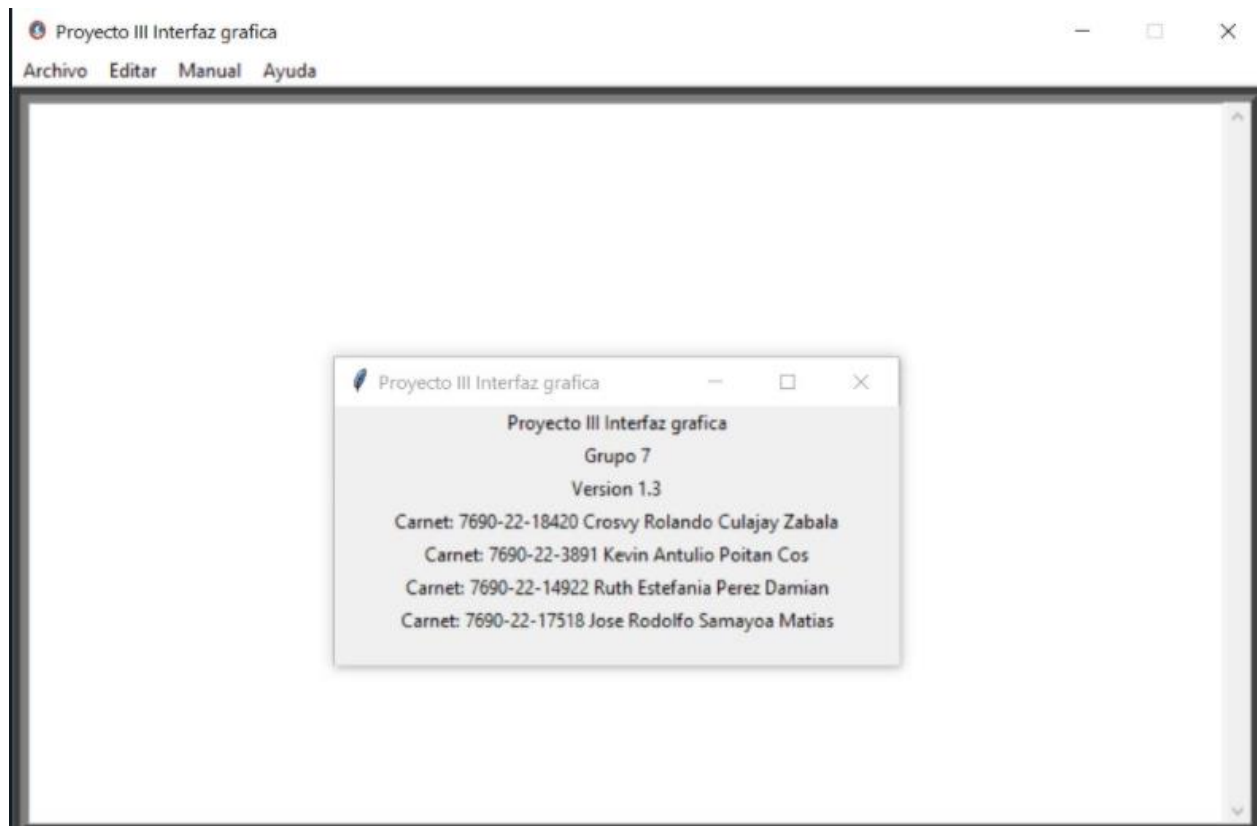


```
# decodigo.com
from tkinter import Tk, Label, Button
class VentanaEjemplo:
    def __init__(self, master):
        self.master = master
        master.title("Una simple interfaz gráfica")
        self.etiqueta = Label(master, text="Esta es la primera ventana!")
        self.etiqueta.pack()
        self.botonSaludo = Button(master, text="Saludar", command=self.saludar)
        self.botonSaludo.pack()
        self.botonCerrar = Button(master, text="Cerrar", command=master.quit)
        self.botonCerrar.pack()
    def saludar(self):
        print("¡Hey!")
root = Tk()
miVentana = VentanaEjemplo(root)
```

Archivo creado con éxito.



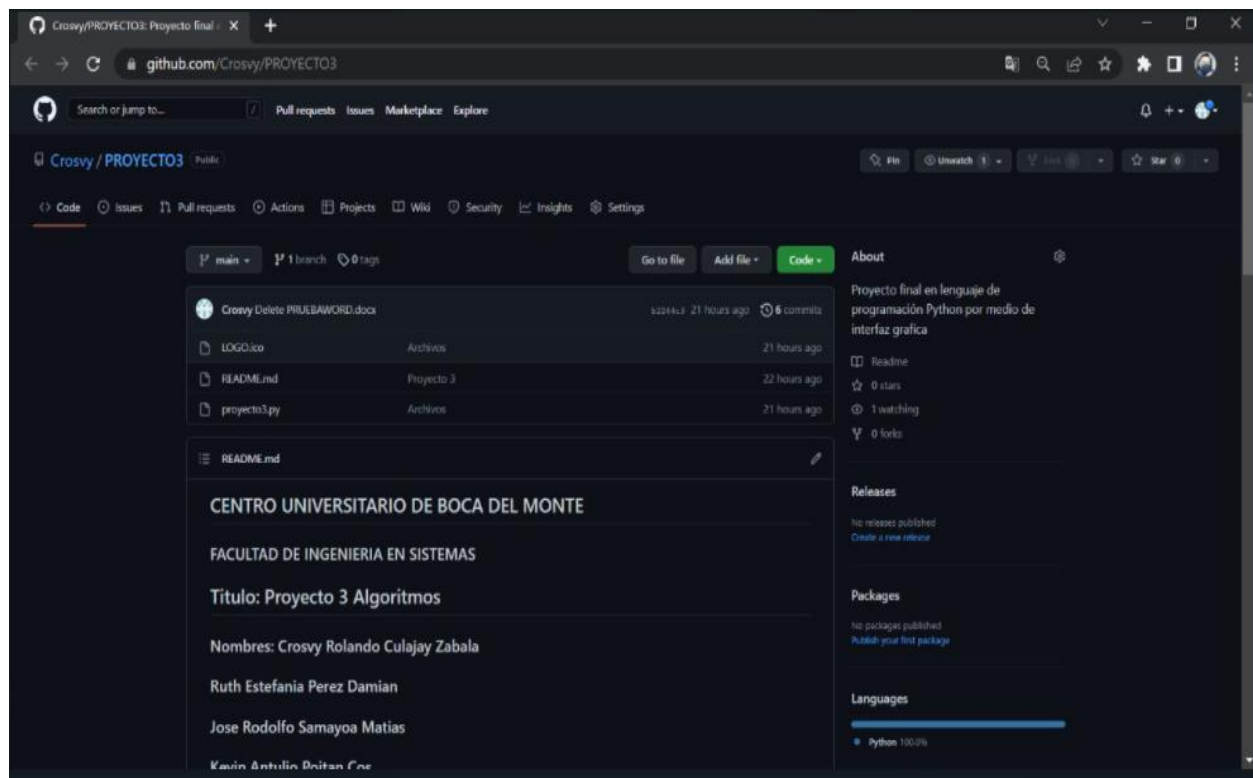
## INFORMACION DE INTERFAZ GRAFICA



## VISTA DE CAMPO MANUAL



## DIRECCIONAMIENTO A GITHUB



## CERRAR APLICACION

