



## **Proyecto Desarrollo de Software**

**Instituto Tecnológico Liceo Cristiano**

### **-Integrantes:**

- Kevin lozano
- Kevin Zambrano
- Carlos Mantilla
- Ashley Almeida
- Rubén Cobos

### **-Materia:**

Fundamento de Hardware y software

### **-Maestro:**

Luigi García

### **-Año:**

2024 -2025

- 
- **Definición general del proyecto de software:** Aplicación que permita calcular el FFMI, AFFMI, su IMC, su masa libre de grasa y su grasa total. Este proyecto está dirigido a usuarios promedios sin mucha experiencia en lo que sería programación.
  - **Especificación de requerimientos del proyecto:** El programa realiza cálculos para proporcionar los valores del FFMI, AFFMI, su IMC, etc. Cuyo limite se basa en que no se permite la entrada de letras por parte del usuario ni el uso de comas para representar decimales.
  - **Procedimiento de instalación y prueba:** El programa fue creado por medio del código Python y para poder ser ejecutada se tiene que tener instalado visual estudio code y el archivo con el código
- 

### Diseño del modelo de datos

Datos de entrada		
Cuadro_estatura = Entry(Cuadro)	Cuadro_kilos = Entry(Cuadro)	Cuadro_Grasa = Entry(Cuadro)
Datos internos		
def calcular()	.Button	def Limpiar_campos()
Datos de salida		
Pantalla = tk()	Pantalla.tittle("Calculadora de FFMI")	Pantalla.mainloop()
Cuadro = Frame(Pantalla)	Cuadro_IMC = StringVar()	Cuadro_GT = StringVar()
Cuadro_ML = StringVar()	Cuadro_FFMI = StringVar()	Cuadro_AFFMI = StringVar()

---

## Documentación técnica – Especificación API

```
1  from tkinter import *
2
3  # Genera la pantalla
4  Pantalla = Tk()
5  Pantalla.title("Calculadora de FFMI")
6
7  Pantalla.mainloop()
```

Se inicio utilizando la línea `from tkinter import *` en Python se usa para importar todos los elementos (clases, funciones, constantes, etc.) del módulo tkinter. tkinter es una biblioteca estándar de Python que proporciona herramientas para crear interfaces gráficas de usuario (GUIs).

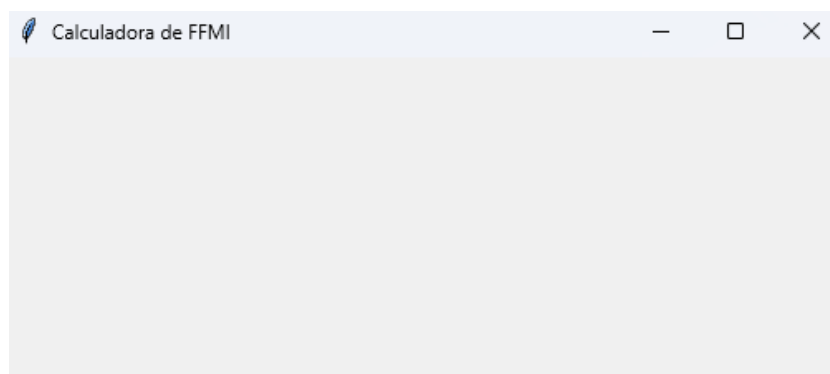
Cuando usas `from tkinter import *`, estás importando todos los componentes de tkinter al espacio de nombres actual. Esto significa que puedes usar directamente las clases y funciones de tkinter sin necesidad de prefijarlas con tkinter.

`Pantalla = tk()`

Luego de esto se le coloca un nombre a la ventana para así poder llamarla al momento de modificarla en este caso se le coloco el nombre de “Pantalla” seguido de un `tk()` la representa la ventana principal de la aplicación gráfica la ventana principal de la aplicación gráfica.

`Pantalla.tittle(“Calculadora de FFMI”)`

Llamamos a la ventana utilizando su nombre (en este caso “Pantalla”) seguido de un `.tittle` y abrimos un paréntesis donde colocaremos el nombre del software es decir la esta línea de código permite cambiar el título del programa.



### Pantalla.mainloop()

La función `mainloop()` en el contexto de `tkinter` es el método que inicia el bucle principal de la aplicación gráfica.

Cuando llamas a `Pantalla.mainloop()`, le estás diciendo a `tkinter` que comience a escuchar y manejar eventos, como clics de botones, movimientos del ratón, teclado, etc. Este bucle es necesario para que la ventana gráfica responda a las interacciones del usuario.

```
4  Pantalla = Tk()
5  Pantalla.title("Calculadora de FFMI")
6  Pantalla.resizable(False, False)
7  Pantalla.iconbitmap("Calculator.ico")
8  Pantalla.geometry("800x500")
9  Pantalla.config(background='#7cdaf9')
```

### Pantalla.resizable(False, False)

El método `.resizable` de un objeto `Tk` en `tkinter` se utiliza para habilitar o deshabilitar la capacidad de cambiar el tamaño de la ventana principal de la aplicación gráfica. Este método controla si el usuario puede ajustar el tamaño de la ventana tanto horizontal como verticalmente (Al decir `false, false` en realidad se está diciendo `width=False, height=False`).

### Pantalla.iconbitmap("Calculator.ico")

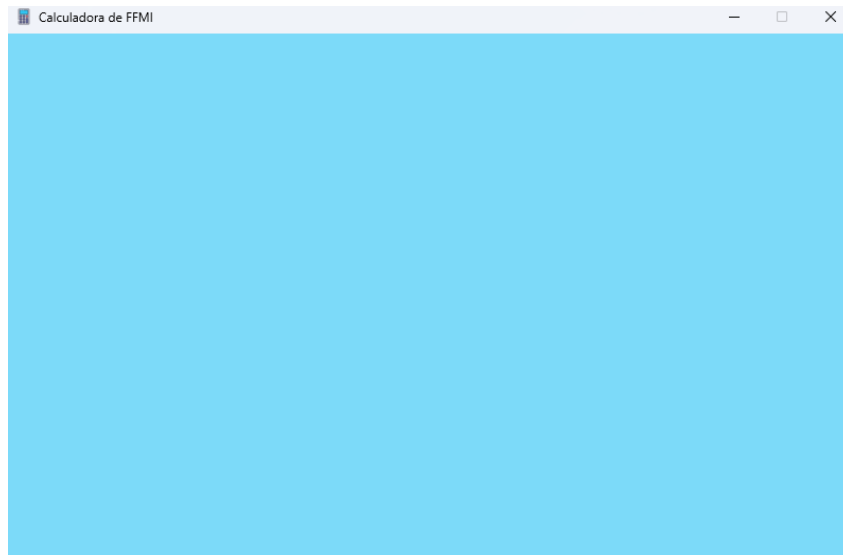
El `.iconbitmap` permite al usuario cambiar el icono de la ventana para poder usarlo se tienen que descargar una imagen y convertirla en `.ico` (El programa no se podrá ejecutar dentro de otro ordenador si no posee al imagen.ico para eliminar este problema se tiene que crear un ejecutable con ayuda de `PyInstaller`).

### Pantalla.geometry("800x500")

El `.geometry` establece el tamaño de la ventana (ancho x alto en píxeles).

### Pantalla.config(background= '#7cdaf9')

El método `config` en un objeto de `tkinter` se usa para configurar o actualizar las opciones y propiedades de un widget o ventana. Puedes usarlo para establecer o cambiar varias opciones de un widget después de su creación en este caso se le está cambiando el fondo de la ventana (`background`) a un color cercano al celeste (`'#7cdaf9'`).



```
Cuadro = Frame(Pantalla)
Cuadro.pack(side="left")
Cuadro.config(background='#f6f7f9', width="700", height="450")
Cuadro.config(highlightbackground="black", highlightthickness=1.5)
```

`Cuadro = Frame(Pantalla)`

Esta línea de código crea una nueva instancia de la clase Frame y la almacena en la variable Cuadro. Un Frame es un widget contenedor en tkinter que se utiliza para agrupar y organizar otros widgets dentro de una ventana o de otro Frame.

`Cuadro.pack(side="left")`

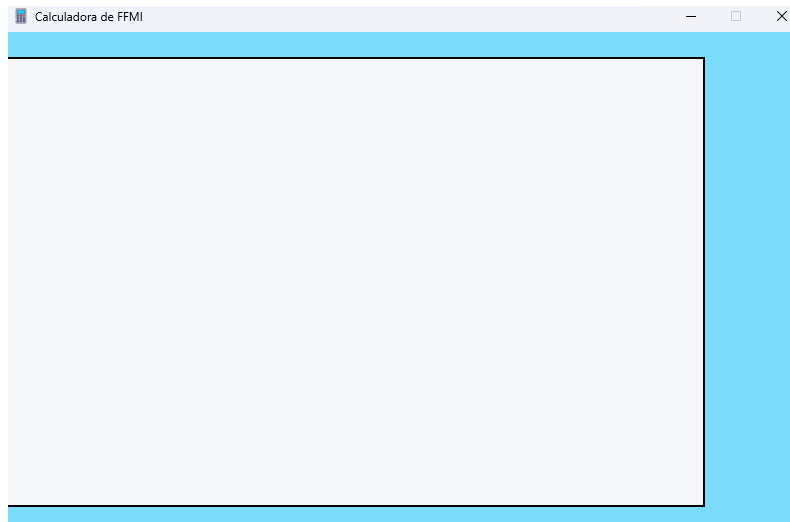
El `.pack()` se usa para posicionar widgets dentro de un contenedor, como un Frame o una ventana principal. El método `pack()` organiza los widgets en bloques y los coloca en la ventana de acuerdo con las opciones proporcionadas, en este caso al colocar (`side="left"`) se le está especificando al frame que se coloque a la izquierda de la ventana "Pantalla".

`Cuadro.config(background='#f6f7f9', width="700", height="450")`

Con esta línea se está configurando las propiedades del frame, tanto el color de su fondo y su largo y ancho.

`Cuadro.config(highlightbackground="black", highlightthickness=1.5)`

Se configura el frame para que obtenga un borde de color negro (`highlightbackground="black"`) y su grosor (`highlightthickness=1.5`) se debe tener en claro cuál es el tipo de diseño que se le quiere colocar a este borde eso especifica la palabra (`highlight`).

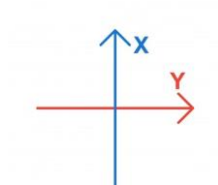


```
Titulo = Label(Cuadro, text="Calculadora", background='#f6f7f9', font=("Arial", 15))
Titulo.place(x=20, y=20)
```

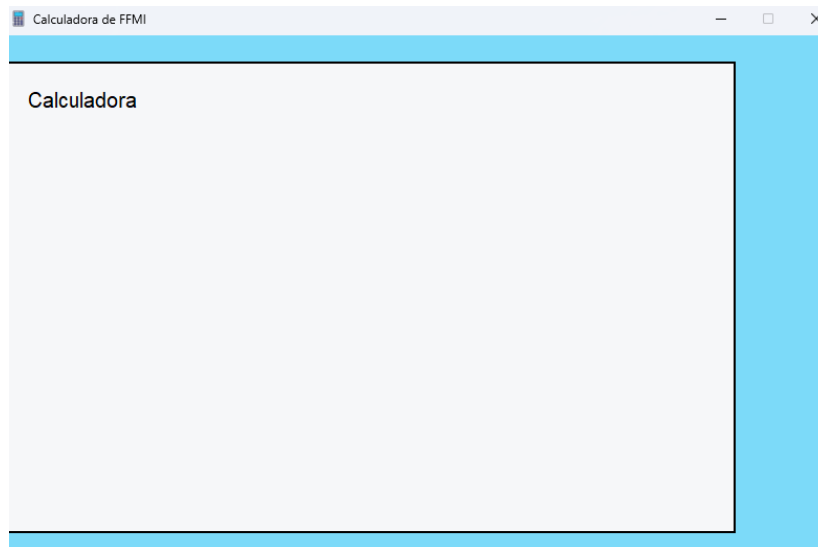
`Titulo = Label(Cuadro, text="Calculadora", background='#f6f7f9', font=("Arial", 15))`

Con esta línea creamos un label el cual es utilizado para mostrar texto o imágenes en una ventana y lo asignamos a nuestro frame (Cuadro) luego de esto escribimos nuestro texto (`text="calculadora"`), le cambiamos el fondo y usando `font=("Arial", 15)` le asignamos una fuente y tamaño de letra a nuestro label.

`Titulo.place(x=20, y=20)`



Con el `.place` podemos mover de lugar nuestro label usando el eje de X y Y



```
Estatura = Label(Cuadro, text="Ingrese su altura en metros:", background='#f6f7f9', font=("Arial", 11))
Estatura.place(x=20, y=70)
Cuadro_estatura = Entry(Cuadro)
Cuadro_estatura.place(x=210, y=72)

Kilos = Label(Cuadro, text="Ingrese su peso en kilos:", background='#f6f7f9', font=("Arial", 11))
Kilos.place(x=20, y=110)
Cuadro_kilos = Entry(Cuadro)
Cuadro_kilos.place(x=190, y=112)

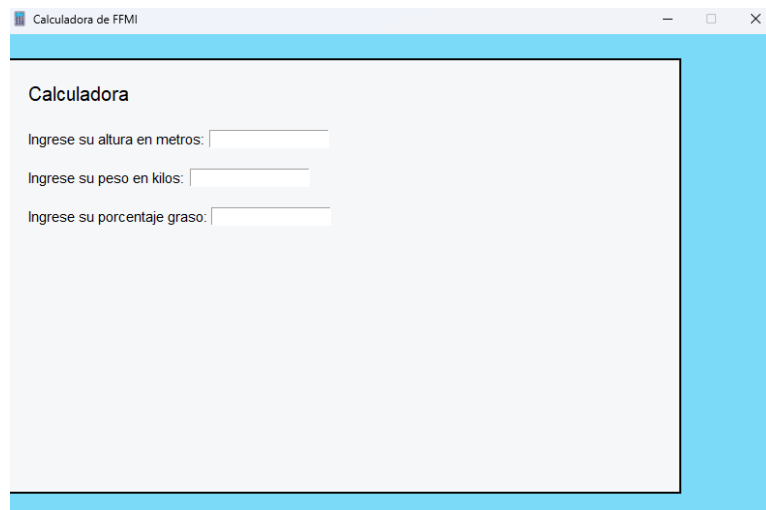
Grasa = Label(Cuadro, text="Ingrese su porcentaje graso:", background='#f6f7f9', font=("Arial", 11))
Grasa.place(x=20, y=150)
Cuadro_Grasa = Entry(Cuadro)
Cuadro_Grasa.place(x=212, y=152)
```

`Cuadro_estatura = Entry(Cuadro)`

`Cuadro_kilos = Entry(Cuadro)`

`Cuadro_Grasa = Entry(Cuadro)`

El `.Enty()` crea una instancia del widget `Entry` y la almacena en las variables `Cuadro_kilos`, `Cuadro_estatura` y `Cuadro_Grasa`. El widget `Entry` se utiliza para permitir que los usuarios ingresen texto en un campo de entrada de una sola línea, en otras palabras con el `.Enty(Cuadro)` creamos los espacios en donde el usuario depositara la información que posterior mente será calculada (es importante recalcar que dentro del paréntesis se colocara el lugar en donde aparecerá el enty en este caso como dentro del paréntesis colocamos “Cuadro” este aparecerá dentro de nuestro frame).



```
Cuadro_IMC = StringVar() # Variable de control para el IMC
IMC = Label(Cuadro, text="Su IMC es de:", background='#f6f7f9', font=("Arial", 11))
IMC.place(x=20, y=250)
Entrada_IMC = Entry(Cuadro, textvariable=Cuadro_IMC, state='readonly')
Entrada_IMC.place(x=120, y=252)

Cuadro_GT = StringVar()
GT = Label(Cuadro, text="Su grasa total en kilos es:", background='#f6f7f9', font=("Arial", 11))
GT.place(x=270, y=250)
Entrada_GT = Entry(Cuadro, textvariable=Cuadro_GT, state='readonly')
Entrada_GT.place(x=445, y=252)

Cuadro_ML = StringVar()
ML = Label(Cuadro, text="Su masa libre de grasa en kilo es:", background='#f6f7f9', font=("Arial", 11))
ML.place(x=20, y=300)
Entrada_ML = Entry(Cuadro, textvariable=Cuadro_ML, state='readonly')
Entrada_ML.place(x=250, y=302)

Cuadro_FFMI = StringVar()
FFMI = Label(Cuadro, text="Su FFMI es de:", background='#f6f7f9', font=("Arial", 11))
FFMI.place(x=20, y=350)
Entrada_FFMI = Entry(Cuadro, textvariable=Cuadro_FFMI, state='readonly')
Entrada_FFMI.place(x=130, y=352)

Cuadro_AFFMI = StringVar()
AFFMI = Label(Cuadro, text="Su AFFMI aproximadamente es de:", background='#f6f7f9', font=("Arial", 11))
AFFMI.place(x=20, y=400)
Entrada_AFFMI = Entry(Cuadro, textvariable=Cuadro_AFFMI, state='readonly')
Entrada_AFFMI.place(x=260, y=402)
```

Cuadro\_IMC = StringVar()

Cuadro\_GT = StringVar()

Cuadro\_ML = StringVar()

Cuadro\_FFMI = StringVar()

Cuadro\_AFFMI = StringVar()

StringVar es una clase que se utiliza para gestionar variables de texto en tkinter y facilita la vinculación entre las variables de texto y los widgets que muestran o permiten modificar esos datos es decir StringVar puede ser útil para manejar los datos de entrada y salida de manera eficiente (Estas líneas son las que permitirán al código mostrar la información calculada dentro de el respectivo enty que se haya asignado al momento de presionar un botón).



```
def calcular():
    try:
        kilos = float(Cuadro_kilos.get())
        estatura = float(Cuadro_estatura.get())
        grasa = float(Cuadro_Grasa.get())
        gt = kilos * (grasa/100)
        imc = kilos / (estatura ** 2)
        ml = kilos * (1 - grasa / 100)
        ffm_i = ml / (estatura ** 2)
        affm_i = ffm_i + (6.3 * (1.8 - estatura))
        Cuadro_GT.set(f"{gt:.2f}")
        Cuadro_IMC.set(f"{imc:.2f}")
        Cuadro_ML.set(f"{ml:.2f}")
        Cuadro_FFMI.set(f"{ffm_i:.2f}")
        Cuadro_AFFMI.set(f"{affm_i:.2f}")
    except ValueError:
        Cuadro_IMC.set("Error")
        Cuadro_ML.set("Error")
        Cuadro_FFMI.set("Error")
        Cuadro_AFFMI.set("Error")
```

Por medio de `def calcular()` definimos una función en este caso las operaciones matemáticas para obtener `gt`, `imc`, `ml`, `ffm_i` y `affm_i`. Utilizando la funciones “try” y “except” haremos que el programa identifique algún error en caso no poder calcular los datos colocados por el usuario por medio de `.set` se mostrara mensajes de error en los resultados a esto lo conocemos como un “ValueError”, si la información proporcionada por el usuario es correcta se procederá a realizar los respectivos cálculos.

```
Calcular = Button(Cuadro, text="Calcular", command=calcular)
Calcular.place(x=440, y=112)
```

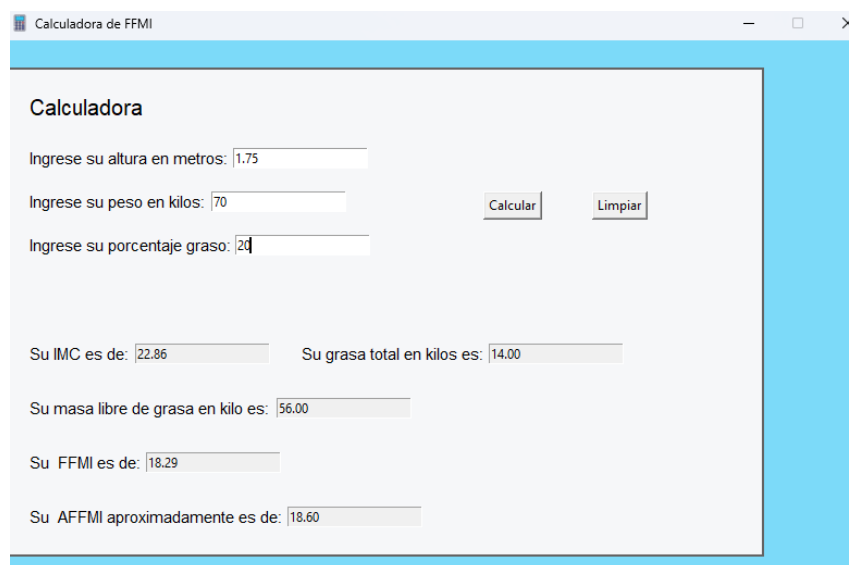
Button es la manera en la que creamos los botones y por medio de “command” llamaremos a la función “calcular” que hemos creado con anterioridad

```
def limpiar_campos():
    Cuadro_estatura.delete(0, END)
    Cuadro_kilos.delete(0, END)
    Cuadro_Grasa.delete(0, END)
    Cuadro_IMC.set("")
    Cuadro_GT.set("")
    Cuadro_ML.set("")
    Cuadro_FFMI.set("")
    Cuadro_AFFMI.set("")

Limpiar = Button(Cuadro, text="Limpiar", command=limpiar_campos)
Limpiar.place(x=540, y=112)
```

Creamos la función “Limpiar\_campos” y dentro de esta colocamos los nombres de los cuadros en donde el usuario ingresa la información seguido de un .delete (.delete() se utiliza para eliminar el contenido de un widget de entrada de texto, como un campo Entry o un área de texto “Text”) dentro de este .delete se coloca 0, END (0 indica que el delete borra el comentario del usuario desde su inicio hasta END que es el final), finalmente llamamos a los StringVar() con un .set y les indicamos que muestren “” lo cual significa que mostraran nada.

Creamos un button colocamos comand=limpiar\_campos para llamar a la acción y cada vez que el usuario presione ese botón el programa se limpiara para volver a poderse usar.



Calculadora de FFMI

**Calculadora**

Ingrese su altura en metros:

Ingrese su peso en kilos:

Ingrese su porcentaje grasa:

Su IMC es de:  Su grasa total en kilos es:

Su masa libre de grasa en kilo es:

Su FFMI es de:

Su AFFMI aproximadamente es de: