



UNIVERSIDAD POLITÉCNICA DE TEXCOCO



# Manual de Usuario

Programa: CALCULADORA DE BMI

Autores:

López Serrano Esmeralda

Texcoco de Mora, Estado de México, 2024

# Índice General

1. Objetivos.....	3
1. Introducción .....	4
3. Descripción de los archivos del programa.....	7

# Índice de Figuras

Ilustración 1Comando de Tkinter.....	6
Ilustración 2Comando de Pillow.....	7
Ilustración 3Importación de bibliotecas.....	7
Ilustración 4Función calcular.....	8
Ilustración 5Función guardar .....	9
Ilustración 6.Función guardar .....	10
Ilustración 7Pyinstaller .....	11
Ilustración 8Pyinstaller .....	11
Ilustración 9Crear .exe .....	11

## 1. Objetivos

El programa tiene como objetivo general:

Desarrollar una aplicación de escritorio intuitiva y fácil de usar que permita a los usuarios calcular su Índice de Masa Corporal (BMI) de manera precisa y rápida, proporcionando también la funcionalidad de almacenar los resultados en archivos CSV para su posterior análisis y seguimiento.

Para lograr los objetivos generales el programa cumple con siguientes objetivos específicos:

☐ **Desarrollar una Interfaz Gráfica Intuitiva:**

- Implementar una interfaz gráfica de usuario (GUI) utilizando Tkinter que permita a los usuarios ingresar fácilmente sus datos personales.

☐ **Calcular el Índice de Masa Corporal (BMI):**

- Crear una función que calcule el BMI utilizando la fórmula estándar y que pueda manejar entradas tanto en metros como en centímetros.

☐ **Manejo de Errores y Validación de Datos:**

- Implementar mecanismos para la validación de datos ingresados y manejo de errores, asegurando que los usuarios ingresen valores válidos y reciban retroalimentación inmediata en caso de errores.

☐ **Guardar Resultados en Archivos CSV:**

- Proveer una funcionalidad para guardar los resultados del cálculo del BMI junto con los datos personales en un archivo CSV, facilitando el seguimiento y análisis a largo plazo.

☐ **Incorporar Elementos Visuales Atractivos:**

- Utilizar imágenes y gráficos en la interfaz para mejorar la experiencia del usuario y hacer la aplicación más atractiva.

□ **Documentar el Proceso de Instalación y Uso:**

- Crear una guía de instalación y uso detallada que permita a cualquier usuario instalar y utilizar la aplicación sin dificultades.

□ **Asegurar la Portabilidad de la Aplicación:**

- Asegurar que la aplicación funcione correctamente en cualquier sistema con Python 3.x y las bibliotecas necesarias instaladas.

## **1. Introducción**

El Índice de Masa Corporal (BMI) es una medida que se utiliza para determinar si una persona tiene un peso saludable en relación con su altura. El BMI se calcula dividiendo el peso de una persona en kilogramos por el cuadrado de su altura en metros. Este proyecto presenta una aplicación de escritorio desarrollada en Python que permite a los usuarios calcular su BMI de manera fácil y rápida.

La calculadora de BMI utiliza la biblioteca Tkinter para crear una interfaz gráfica de usuario (GUI) intuitiva y amigable, donde los usuarios pueden ingresar sus datos personales como nombre, edad, sexo, peso y altura. Además, la aplicación permite guardar los resultados en un archivo CSV, lo que facilita el seguimiento del BMI a lo largo del tiempo.

El objetivo de esta herramienta es proporcionar una manera simple y accesible para que los usuarios puedan monitorizar su salud y hacer ajustes en su estilo de vida si es necesario. Con una instalación y uso sencillo, esta calculadora de BMI es ideal para cualquier persona interesada en mantener un registro de su estado físico.

## 1.1 Funcionamiento del programa

- Cálculo del BMI

El BMI se calcula utilizando la siguiente fórmula:

$$BMI = \frac{PESO}{(ALTURA)^2}$$

- Guardado de datos

Los datos del usuario se guardan en un archivo CSV con el nombre del usuario. El archivo incluye las columnas: Nombre, Edad, Sexo, Peso, Altura, BMI.

## 2. Instalación

Instalar una calculadora de Índice de Masa Corporal (IMC) creada en Python implica varios pasos. Aquí te dejo una guía detallada:

### 2.1 Requerimientos mínimos del sistema

Para garantizar el correcto funcionamiento del programa, el sistema debe cumplir con los siguientes requisitos mínimos:

- **Sistema operativo:** Windows, macOS, Linux
- **Procesador:** Cualquier procesador Intel o AMD x86-64
- **Python:** Versión 3.6 o superior
- **Bibliotecas:** tkinter (incluida con Python), PIL (Pillow), csv, os

## 2.2 Instalación de Python y bibliotecas necesarias

### □ Python:

- Descargue e instale la versión más reciente de Python desde el sitio oficial: [python.org](https://python.org).
- Durante la instalación, asegúrese de seleccionar la opción "Add Python to PATH" para facilitar el uso de Python desde la línea de comandos.

### □ Tkinter:

- Tkinter viene preinstalado con Python en la mayoría de las distribuciones. Sin embargo, si es necesario instalarlo, use los siguientes comandos según su sistema operativo:
  - **Windows:** Tkinter debería estar incluido con la instalación de Python.
  - **macOS:** Tkinter debería estar incluido con la instalación de Python.
  - **Linux:** Use el siguiente comando para instalar Tkinter:

```
bash
```

```
sudo apt-get install python3-tk
```

*Ilustración 1 Comando de Tkinter*

### □ Pillow:

- Pillow es una biblioteca de Python que permite trabajar con imágenes. Para instalar Pillow, abra una terminal o el símbolo del sistema y ejecute el siguiente comando:



```
bash  
  
pip install Pillow
```

*Ilustración 2 Comando de Pillow*

### 3. Descripción de los archivos del programa

#### 3.1 Archivo examen.py

Este archivo contiene el código principal del programa y se compone de las siguientes secciones:

- **Importación de bibliotecas:** Se importan las bibliotecas necesarias como tkinter, messagebox, PhotoImage, PIL, csv y os.

```
import tkinter as tk  
from tkinter import messagebox  
from PIL import Image, ImageTk  
import csv  
import os
```

*Ilustración 3 Importación de bibliotecas*

#### Funciones principales:

- **calcular\_bmi():** Calcula el índice de masa corporal (BMI) basado en el peso y altura ingresados por el usuario y muestra el resultado en la interfaz gráfica.

esta función toma entradas de usuario para nombre, edad, sexo, peso y altura, realiza las conversiones necesarias y calcula el BMI, mostrando el resultado en una etiqueta de la interfaz gráfica. Si se ingresan valores inválidos, se muestra un mensaje de error.

```
def calcular_bmi():  
    try:  
        nombre = entry_nombre.get()  
        edad = int(entry_edad.get())  
        sexo = entry_sexo.get()  
        peso = float(entry_peso.get())  
        altura = float(entry_altura.get())  
  
        if altura > 3:  
            altura /= 100  
  
        bmi = peso / (altura ** 2)  
  
        label_resultado.config(text=f"BMI: {bmi:.2f}")  
  
    except ValueError:  
        messagebox.showerror("Error", "Por favor, ingrese valores válidos.")
```

*Ilustración 4 Función calcular*

- **guardar\_datos():** Guarda los datos del usuario en un archivo CSV.

Esta función toma entradas de usuario para nombre, edad, sexo, peso y altura, realiza las conversiones necesarias y calcula el BMI, luego guarda estos datos en un archivo CSV específico para el nombre del usuario, y si se ingresan valores inválidos, muestra un mensaje de error.



```
def guardar_datos():
    try:
        nombre = entry_nombre.get()
        edad = int(entry_edad.get())
        sexo = entry_sexo.get()
        peso = float(entry_peso.get())
        altura = float(entry_altura.get())

        if altura > 3:
            altura /= 100

        bmi = peso / (altura ** 2)

        nombre_archivo = f"{nombre}.csv"

        archivo_nuevo = not os.path.isfile(nombre_archivo)

        with open(nombre_archivo, mode='a', newline='') as file:
            writer = csv.writer(file)

            if archivo_nuevo:
                writer.writerow(["Nombre", "Edad", "Sexo", "Peso", "Altura", "BMI"])

            writer.writerow([nombre, edad, sexo, peso, altura, bmi])

        messagebox.showinfo("Guardado", f"Datos guardados en {nombre_archivo}")

    except ValueError:
        messagebox.showerror("Error", "Por favor, ingrese valores válidos.")
```

*Ilustración 5 Función guardar*

- **Configuración de la interfaz gráfica:** Utiliza Tkinter para crear una interfaz gráfica de usuario (GUI) con campos de entrada para el nombre, edad, sexo, peso y altura del usuario.

```
root = tk.Tk()
root.title("Calculadora de BMI")

image = Image.open("fondo.png")
fondo = ImageTk.PhotoImage(image)

canvas = tk.Canvas(root, width=fondo.width(), height=fondo.height())
canvas.pack(fill="both", expand=True)
canvas.create_image(0, 0, image=fondo, anchor="nw")

canvas_width = fondo.width()
canvas_height = fondo.height()

x_center = canvas_width // 2
y_start = 50
y_step = 60

font_style = ("Helvetica", 16, "bold italic")
title_font_style = ("Helvetica", 24, "bold")
```

*Ilustración 6.Función guardar*

- **Widgets de la GUI:** Incluye etiquetas, entradas de texto, botones y un canvas para la imagen de fondo.
- **Botones y etiquetas de resultado**

## 3.2 Archivo .exe

Crear un archivo .exe para un programa en Python generalmente implica empaquetar el script Python, sus dependencias y cualquier recurso necesario (como archivos de datos o gráficos) en un solo archivo ejecutable. Esto permite que el programa se ejecute en un sistema sin necesidad de tener Python instalado. Uno de los métodos más comunes para hacer esto es utilizando la herramienta PyInstaller.

**Instalar PyInstaller:** Asegúrate de tener PyInstaller instalado. Puedes instalarlo usando pip:

```
bash  
  
pip install pyinstaller
```

*Ilustración 7Pyinstaller*

- **Preparar el Script Principal:** Asegúrate de que tu script principal (por ejemplo, main.py) contiene el código necesario y es el punto de entrada de tu aplicación.
- **Crear el Archivo .spec (opcional):** PyInstaller genera un archivo .spec que describe cómo construir el ejecutable. Puedes personalizarlo si necesitas incluir recursos adicionales.

```
bash  
  
pyinstaller --name=my_bmi_calculator --onefile --windowed main.py
```

*Ilustración 8Pyinstaller*

**Ejecutar PyInstaller:** Corre PyInstaller con las opciones adecuadas para crear un archivo .exe:

```
bash  
  
pyinstaller --onefile --windowed main.py
```

*Ilustración 9Crear .exe*

- □ **--onefile:** Crea un solo archivo ejecutable.
  - **--windowed:** Evita que se abra una ventana de consola (útil para aplicaciones gráficas).
- **Distribución:** Una vez que PyInstaller haya terminado, encontrarás el archivo ejecutable en la carpeta dist/. Este archivo .exe puede ser distribuido y ejecutado en cualquier sistema Windows compatible.

## Descripción del Archivo .exe

El archivo .exe generado por el proceso anterior es un ejecutable independiente de Windows que contiene:

1. **El código de tu aplicación:** Esto incluye las funciones `calcular_bmi` y `guardar_datos`, así como cualquier otra funcionalidad que hayas implementado en tu script principal (`main.py`).
2. **Dependencias de Python:** Todas las librerías de Python que tu programa necesita para funcionar, incluidas en el archivo .exe.
3. **Recursos Adicionales:** Cualquier recurso adicional que hayas especificado, como iconos, imágenes, archivos de configuración, etc.
4. **Intérprete de Python:** Una versión del intérprete de Python empaquetada con tu aplicación, lo que elimina la necesidad de que el usuario final tenga Python instalado en su sistema.

## Ejecución del Archivo .exe

El usuario final solo necesita hacer doble clic en el archivo .exe para ejecutar tu aplicación. La aplicación se comportará igual que si se estuviera ejecutando directamente desde Python, pero sin requerir una instalación previa de Python en el sistema del usuario.

## Consideraciones Finales

- **Pruebas:** Asegúrate de probar el archivo .exe en diferentes máquinas para garantizar que funcione correctamente en diversos entornos.
- **Mantenimiento:** Si realizas actualizaciones a tu script, deberás volver a empaquetar el archivo .exe.
- **Compatibilidad:** El archivo .exe solo funcionará en sistemas operativos Windows. Para otros sistemas operativos, necesitarás empaquetar la aplicación utilizando herramientas específicas para esos entornos.

Espero que esta descripción te sea útil para entender el proceso de creación y el contenido de un archivo .exe para tu programa en Python.

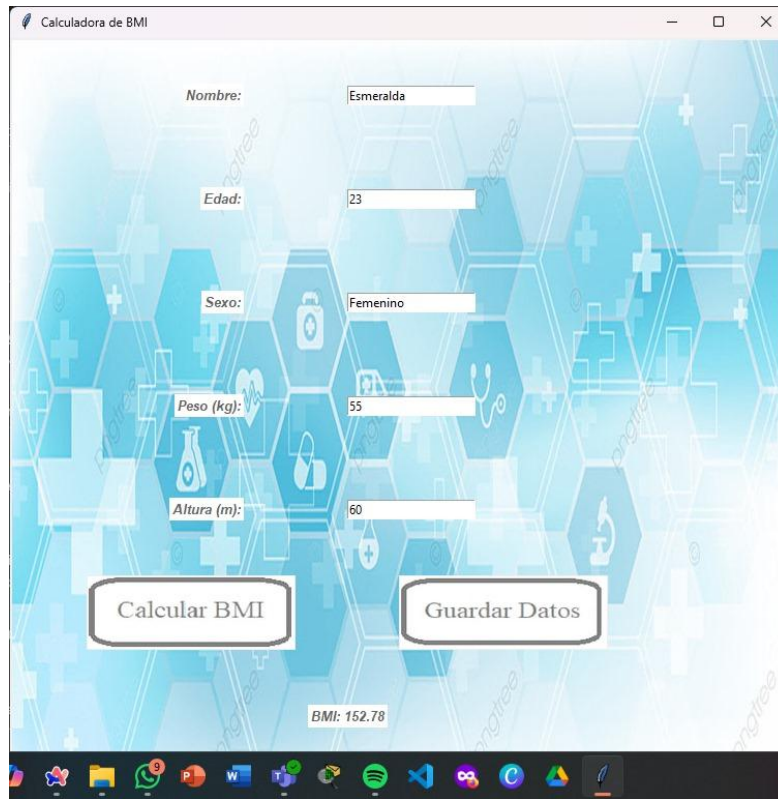


Ilustración 10 IMC.exe