



UNIVERSIDAD POLITÉCNICA DE TEXCOCO

INGENIERIA EN SISTEMAS COMPUTACIONALES

Aplicación para el Cálculo del Índice de Masa Corporal (IMC).

Manual de Técnico

GRUPO: 8VSC2

ALUMNO:

Cruz Meráz Martin 13211405318

DISEÑO DE INTERFACES

PROFESOR:

Enrique Tonatiuh Jiménez Borgonio



1.1.1 Índice General

1.1.1	Índice General.....	2
2	Índice de Figuras	4
3	Objetivo	5
4	Introducción.....	5
5	Requisitos de Hardware y Software	5
6	Instalación de Complementos	6
6.1	Instalación de Bibliotecas.....	6
6.2	Instalación de Python	6
6.3	Instalación Terminada	7
7	¿Qué es el Índice de Masa Corporal?.....	7
7.1	Formula del IMC.....	7
7.2	Cálculo de la Formula de IMC	7
7.3	Valores del IMC.	8
7.4	Funcionalidad de Campos de Entrada.....	9
7.4.1	Campo de Nombre	9
7.4.2	Campo de Peso	9
7.4.3	Campo de Altura	10
7.4.4	Campo de Edad.....	10
7.4.5	Campo Tipo de Sexo	10
7.4.6	Botón Calcular	11
7.4.7	Botón Guardar.....	12
7.4.8	Botón Leer CSV	12
8	Funcionamiento de la Aplicación	12
8.1	Cálculo de IMC.....	13
9	Guardar y Visualizar	14
10	Notación Camel.....	15
11	Tipos de Archivos CSV	15
12	Estructura del Código Base	15
12.1	Módulos Importados	15
12.2	Función para Calcular IMC	16
12.3	Función para Guardar Datos en CSV	17
12.4	Función para Leer Datos desde un Archivo CSV.....	17

12.5	Creación de la Ventana Principal y Widgets	18
13	Funcionalidad	20
14	Mantenimiento.....	20
15	Elementos Importantes	21
16	Diagrama de Flujo	22
17	Descripción Del Diagrama	23
17.1	Relaciones entre Nodos	24

2 Índice de Figuras

Figura 6.1 Comando de Instalación Pillow.	6
Figura 6.2 Página oficial de Python.	6
Figura 6.3 Instalación finalizada de Python.	7
Figura 7.1 Imagen de valores respecto al IMC.	9
Figura 7.2 Campo para ingresar Nombre.....	9
Figura 7.3 Campo para Ingresar Peso en Kg.	10
Figura 7.4 Campo para Ingresar la Altura en m.	10
Figura 7.5 Campo para Ingresar la edad en Número.	10
Figura 7.6 Campo de selección de Sexo (Hombre).	11
Figura 7.7 Campo de selección de Sexo (Mujer).	11
Figura 7.8 Botón para calcular IMC.	11
Figura 7.9 Botón utilizado para Guardar Datos.	12
Figura 7.10 Botón para Leer CSV.	12
Figura 8.1 Ingreso de nombre en el campo.....	12
Figura 8.2 Ingreso de peso en Kg.....	13
Figura 8.3 Ingreso de Altura en m.....	13
Figura 8.4 Ingreso de Edad en Numero.....	13
Figura 8.5 Selección de Sexo.	13
Figura 8.6 Ejemplo de IMC Sexo Femenino.....	14
Figura 8.7 Ejemplo de IMC Sexo Masculino.	14
Figura 9.1 Visualización de Datos de Usuario.....	14
Figura 9.2 Visualización de Datos de Usuario.	15
Figura 12.1 Módulos Importados	15
Figura 12.2 Función de Cálculo de IMC.	16
Figura 12.3 Función de Almacenamiento de Datos.....	17
Figura 12.4 Función para Leer Datos.....	17
Figura 12.5 Configuración de Encabezados.	18
Figura 12.6 Creación de Ventanas y Complementos.	18
Figura 12.7 Elementos dentro del Programa.....	19
Figura 12.8 Posicionamiento de Elementos.	19
Figura 16.1 Imagen del Diagrama de Flujo.	22

3 Objetivo

Proporcionar una herramienta accesible y fácil de usar para calcular el Índice de Masa Corporal (IMC) de los usuarios mediante una interfaz gráfica intuitiva. La aplicación fue desarrollada para ayudar a las personas a evaluar su estado nutricional de manera rápida y precisa, facilitando la interpretación de los resultados del IMC. El programa no solo calcula el IMC basado en datos como peso, altura, edad y sexo, sino que también ofrece funcionalidades para guardar los datos en archivos CSV y visualizarlos posteriormente en una interfaz dedicada.

4 Introducción

La Calculadora de IMC con Interfaz Gráfica es una herramienta diseñada para ofrecer una interfaz intuitiva que facilita el cálculo del IMC, una medida clave para evaluar el estado nutricional de una persona. La aplicación proporciona funcionalidades para ingresar datos personales, calcular el IMC, guardar los resultados en archivos CSV y leer estos archivos para visualizarlos de manera organizada.

La aplicación está desarrollada utilizando Python y Tkinter, con soporte adicional para la manipulación de imágenes y archivos CSV a través de la biblioteca Pillow.

5 Requisitos de Hardware y Software

Hardware		
Procesador	Memoria RAM	Espacio en Disco
Procesador moderno compatible con Python (recomendado Intel i3 o equivalente).	Mínimo 2 GB (se recomienda 4 GB para un rendimiento óptimo).	Al menos 50 MB de espacio libre para la instalación de la aplicación y almacenamiento de archivos temporales.

Software	
Sistema Operativo	Python
Windows 10 o superior, macOS 10.14 o superior, o una distribución moderna de Linux.	Versión 3.1 (Python 3.6 o superior es compatible).

6 Instalación de Complementos

Para que la Calculadora de IMC funcione correctamente, es necesario instalar ciertos complementos y bibliotecas adicionales en el entorno de Python. A continuación, se detallan los pasos para realizar las instalaciones pertinentes.

6.1 Instalación de Bibliotecas

Pillow es una biblioteca de Python que proporciona funcionalidades avanzadas para la manipulación de imágenes. Es esencial para cargar y mostrar imágenes en la interfaz gráfica.

Comando de Instalación:

```
pip install pillow
```

Figura 6.1 Comando de Instalación Pillow.

Pillow se utiliza para abrir, manipular y guardar imágenes en diferentes formatos. En esta aplicación, se usa para mostrar imágenes de fondo y botones en la interfaz gráfica.

6.2 Instalación de Python

Para realizar la instalación de Python se necesitará visitar el sitio oficial para obtener el debido instalador de recursos, puede realizarlo desde el siguiente enlace: www.python.org/downloads/



Figura 6.2 Página oficial de Python.

6.3 Instalación Terminada

Una vez concluida la instalación de Python en nuestro sistema se vera de la siguiente forma, es importante marcar la casilla que menciona algo referente al “Add Pach” para que la instalación funcione de forma correcta.

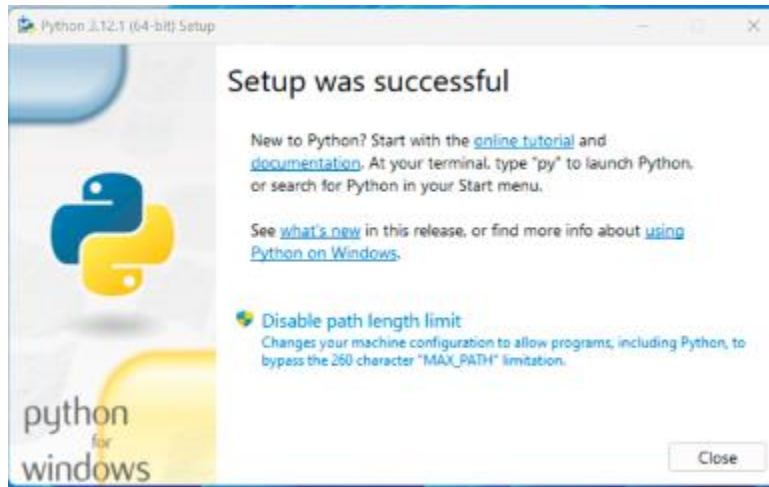


Figura 6.3 Instalación finalizada de Python.

7 ¿Qué es el Índice de Masa Corporal?

El Índice de Masa Corporal (IMC) es una medida que relaciona el peso con la altura de una persona para evaluar si su peso está dentro de un rango saludable. Se utiliza comúnmente en la práctica clínica para determinar si una persona está en riesgo de problemas de salud relacionados con el peso, como la obesidad, el sobrepeso o el bajo peso. Sin embargo, es importante recordar que el IMC no mide directamente la grasa corporal y no tiene en cuenta factores como la masa muscular, la distribución de la grasa o el estado de salud general.

7.1 Formula del IMC

$IMC = \frac{Peso (kg)}{Altura (m)^2}$ Donde:

- ❖ **Peso** es la masa corporal en kilogramos (kg).
- ❖ **Altura** es la estatura en metros (m), y debe ser elevada al cuadrado.

7.2 Cálculo de la Formula de IMC

Para calcular el IMC, sigue estos pasos:

1. **Mide tu peso** en kilogramos (kg). Si tienes tu peso en libras, puedes convertirlo a kilogramos dividiéndolo entre 2.205.
2. **Mide tu altura** en metros (m). Si tienes tu altura en centímetros, divídela entre 100 para obtenerla en metros.

3. **Eleva al cuadrado** tu altura en metros.
4. **Divide** tu peso en kilogramos entre el valor obtenido al elevar la altura al cuadrado.

Ejemplo 1:

Una persona pesa 70 kg y mide 1.75 m.

- Primero, eleva la altura al cuadrado:

$$1.75^2 = 3.0625$$

- Luego, divide el peso por este valor:

$$\text{IMC} = 70/3.0625 = 22.9$$

En este caso, el IMC es aproximadamente 22.9. De acuerdo con las categorías estándar, este valor se encuentra en el rango de **peso normal**.

Ejemplo 2:

Una segunda persona pesa 85 kg y mide 1.80 m.

- Primero, eleva la altura al cuadrado:

$$1.80^2 = 3.24$$

- Luego, divide el peso por este valor:

$$\text{IMC} = 85/3.24 = 26.2$$

En este caso, el IMC es aproximadamente 26.2. Según las categorías estándar, este valor indica **sobrepeso**.

7.3 Valores del IMC.

En terminos generales el Indice de Masa Corporal es utilizado para dar a conocer si una persona posee un peso saludable en reaccion a su propia estatura.



Figura 7.1 Imagen de valores respecto al IMC.

7.4 Funcionalidad de Campos de Entrada

Se presenta a continuación partes destacables del propio sistema Vitales para su óptimo funcionamiento.

7.4.1 Campo de Nombre

❖ Etiqueta (labelNombre):

- **Texto:** "Nombre:"
- **Descripción:** Muestra un texto que indica al usuario que debe ingresar su nombre.
- **Propósito:** Permite al usuario ingresar su nombre.
- **Tipo de Dato:** Texto.
- **Ubicación:** (x=200, y=20) con un ancho de 100 píxeles y una altura de 20 píxeles.



Figura 7.2 Campo para ingresar Nombre.

7.4.2 Campo de Peso

❖ Etiqueta (labelPeso):

- **Texto:** "Peso (kg):"
- **Descripción:** Indica al usuario que debe ingresar su peso en kilogramos.
- **Propósito:** Permite al usuario ingresar su peso en kilogramos.
- **Tipo de Dato:** Numérico (debe ser convertido a float en el cálculo).

- **Ubicación:** (x=200, y=60) con un ancho de 100 píxeles y una altura de 20 píxeles.


 A dark green rectangular box containing the text "Peso (kg):" in purple on the left and a white rectangular input field on the right.

Figura 7.3 Campo para Ingresar Peso en Kg.

7.4.3 Campo de Altura

❖ **Etiqueta (labelAltura):**

- **Texto:** "Altura (m):"
- **Descripción:** Indica al usuario que debe ingresar su altura en metros.
- **Propósito:** Permite al usuario ingresar su altura en metros.
- **Tipo de Dato:** Numérico (debe ser convertido a float en el cálculo).
- **Ubicación:** (x=200, y=100) con un ancho de 100 píxeles y una altura de 20 píxeles.


 A dark green rectangular box containing the text "Altura (m):" in purple on the left and a white rectangular input field on the right.

Figura 7.4 Campo para Ingresar la Altura en m.

7.4.4 Campo de Edad

❖ **Etiqueta (labelEdad):**

- **Texto:** "Edad:"
- **Descripción:** Indica al usuario que debe ingresar su edad.
- **Propósito:** Permite al usuario ingresar su edad.
- **Tipo de Dato:** Numérico (debe ser convertido a int en el cálculo).
- **Ubicación:** (x=200, y=140) con un ancho de 100 píxeles y una altura de 20 píxeles.


 A dark green rectangular box containing the text "Edad:" in purple on the left and a white rectangular input field on the right.

Figura 7.5 Campo para Ingresar la edad en Número.

7.4.5 Campo Tipo de Sexo

❖ **Etiqueta (labelSexo):**

- **Texto:** "Sexo:"
- **Descripción:** Indica al usuario que debe seleccionar su sexo.

❖ **Botón de Radio (radioHombre):**

- **Texto:** "Hombre"
- **Valor:** "Hombre"
- **Propósito:** Permite al usuario seleccionar la opción "Hombre".
- **Ubicación:** (x=200, y=180)

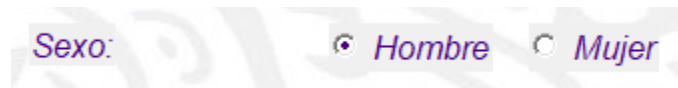


Figura 7.6 Campo de selección de Sexo (Hombre).

❖ **Botón de Radio (radioMujer):**

- **Texto:** "Mujer"
- **Valor:** "Mujer"
- **Propósito:** Permite al usuario seleccionar la opción "Mujer".
- **Ubicación:** (x=300, y=180)

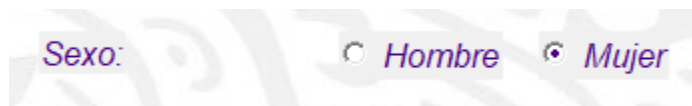


Figura 7.7 Campo de selección de Sexo (Mujer).

7.4.6 Botón Calcular

- **Nombre de Variable:** botonCalcular
- **Imagen Asociada:** imgCalcular (Cargada desde "b1.png")
- **Propósito:** Ejecutar la función calcularImc(), que calcula el Índice de Masa Corporal (IMC) basado en los valores ingresados por el usuario.
- **Ubicación:** (x=50, y=220) con un ancho de 300 píxeles y una altura de 50 píxeles.

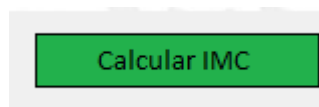


Figura 7.8 Botón para calcular IMC.

7.4.7 Botón Guardar

- **Nombre de Variable:** botonGuardar
- **Imagen Asociada:** imgGuardar (Cargada desde "b2.png")
- **Propósito:** Ejecutar la función guardarDatos(), que guarda los datos ingresados y el IMC calculado en un archivo CSV con el nombre del usuario.
- **Ubicación:** (x=50, y=320) con un ancho de 300 píxeles y una altura de 50 píxeles.

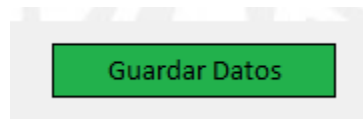


Figura 7.9 Botón utilizado para Guardar Datos.

7.4.8 Botón Leer CSV

- **Nombre de Variable:** botonLeerCsv
- **Imagen Asociada:** imgLeer (Cargada desde "b3.png")
- **Propósito:** Ejecutar la función leerCsv(), que abre un archivo CSV seleccionado por el usuario y muestra los datos en una nueva ventana.
- **Ubicación:** (x=50, y=380) con un ancho de 300 píxeles y una altura de 50 píxeles.

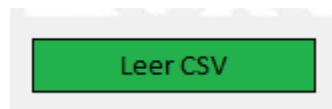


Figura 7.10 Botón para Leer CSV.

8 Funcionamiento de la Aplicación

Nombre: En este campo necesitaremos ingresar nuestro nombre como tal.



Figura 8.1 Ingreso de nombre en el campo.

Peso (Kg): Para el Apartado del peso es necesario ingresarlo en kg de lo contrario no se podrá realizar tal calculo.

A dark green rectangular input field with a white border. On the left, the text "Peso (kg):" is written in a purple, cursive-style font. To the right of this text, the number "70" is entered in the same purple font.

Figura 8.2 Ingreso de peso en Kg.

Altura: En el campo de la altura es necesario ingresarlo en Metros (m), con centímetros.

A dark green rectangular input field with a white border. On the left, the text "Altura (m):" is written in a purple, cursive-style font. To the right of this text, the number "1.78" is entered in the same purple font.

Figura 8.3 Ingreso de Altura en m.

Edad: Hablando de la edad, es necesario ingresarla de forma numérica y no con palabra.

A light brown rectangular input field with a dark green border. On the left, the text "Edad:" is written in a purple, cursive-style font. To the right of this text, the number "20" is entered in the same purple font.

Figura 8.4 Ingreso de Edad en Numero.

Sexo: Aquí es donde se selecciona si el usuario es perteneciente al Sexo Femenino o Masculino.

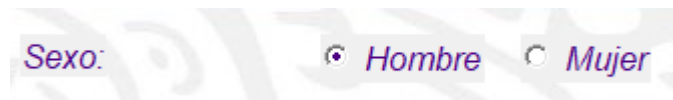
A horizontal form element. On the left, the text "Sexo:" is written in a purple, cursive-style font. To its right are two radio button options. The first option is labeled "Hombre" in a purple, cursive-style font and has a selected radio button (a small circle with a dot in the center). The second option is labeled "Mujer" in a purple, cursive-style font and has an unselected radio button (an empty circle).

Figura 8.5 Selección de Sexo.

8.1 Cálculo de IMC

La fórmula básica del IMC es:

$$\text{IMC} = \text{peso (Kg)} / \text{altura(m)}^2$$

En el código, el IMC se ajusta mediante factores específicos para hombres y mujeres:

1. Factor de Ajuste por Sexo (ks):

- **Hombres:** El factor es 1.0.
- **Mujeres:** El factor es 1.1.

2. Factor de Ajuste por Edad (ka):

- Este factor se calcula basándose en la edad del individuo y se aplica a ambos sexos.

La fórmula completa del IMC ajustado es:

$$\text{IMC} = \text{peso(Kg)} / \text{altura(m)}^2 \times k_s \times k_a$$

Donde:

- **ks** es el factor de ajuste por sexo.
- **ka** es el factor de ajuste por edad.

Calculadora de IMC

Nombre: Natalia

Peso (kg): 65

Altura (m): 1.63

Edad: 25

Sexo: ☐ Hombre ☒ Mujer

Calcular IMC

IMC: 26.91 - Sobrepeso

Guardar Datos

Leer CSV

Figura 8.6 Ejemplo de IMC Sexo Femenino.

Calculadora de IMC

Nombre: Rafael

Peso (kg): 70

Altura (m): 1.78

Edad: 20

Sexo: ☒ Hombre ☐ Mujer

Calcular IMC

IMC: 20.99 - Peso normal

Guardar Datos

Leer CSV

Figura 8.7 Ejemplo de IMC Sexo Masculino.

Así es como se visualizan los cálculos para cada uno de los casos, un hombre y una mujer.

9 Guardar y Visualizar

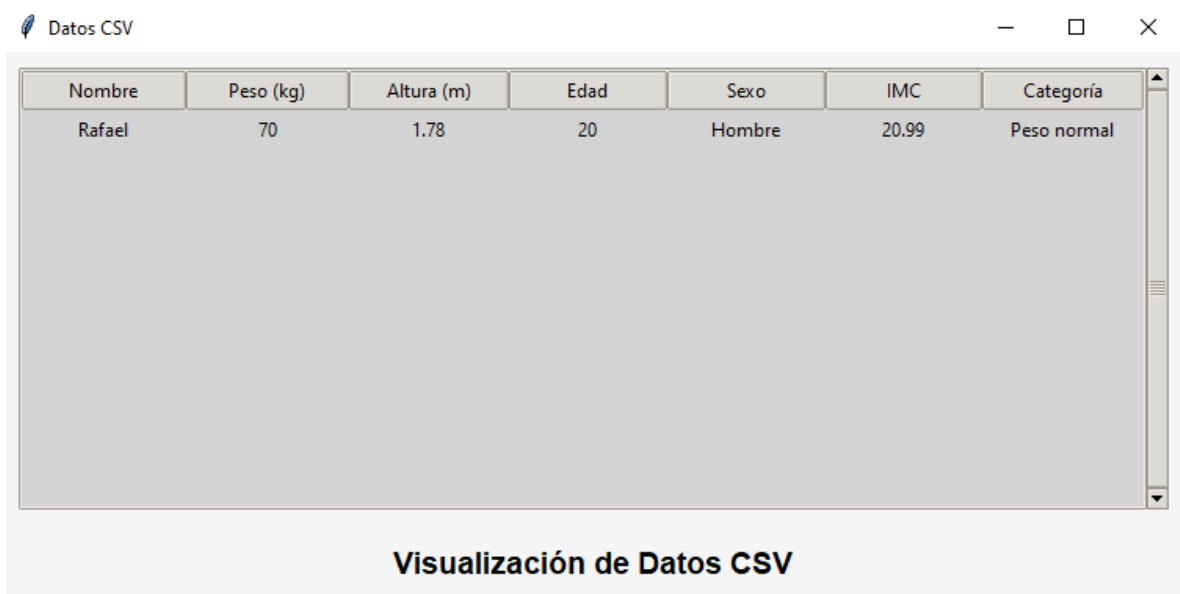
Para el apartado de guardar datos y visualizar los mismos se mostrará de la siguiente forma, al pulsar el botón de leer csv y seleccionar el archivo con el nombre del usuario en cuestión se abrirá una ventana mostrando los datos ingresados.

Datos CSV

Nombre	Peso (kg)	Altura (m)	Edad	Sexo	IMC	Categoría
Natalia	65	1.63	25	Mujer	26.91	Sobrepeso

Visualización de Datos CSV

Figura 9.1 Visualización de Datos de Usuario.



Nombre	Peso (kg)	Altura (m)	Edad	Sexo	IMC	Categoría
Rafael	70	1.78	20	Hombre	20.99	Peso normal

Visualización de Datos CSV

Figura 9.2 Visualización de Datos de Usuario.

10 Notación Camel

La notación **camelCase** es una convención para nombrar variables, funciones, y otros identificadores en la programación, que se utiliza para mejorar la legibilidad del código. En camelCase, el nombre de la variable o función se escribe sin espacios ni guiones, y cada palabra adicional, excepto la primera, comienza con una letra mayúscula.

11 Tipos de Archivos CSV

12 Estructura del Código Base

Es de suma importancia conocer el funcionamiento de cada parte del código, con la finalidad de que pensando a futuro este pueda actualizarse o implementar nuevas funcionalidades, por ello es necesario que tenga un estándar para que sea entendible por cualquier técnico en cuestión.

12.1 Módulos Importados

```
import tkinter as tk # Biblioteca para la creación de interfaces gráficas
from tkinter import filedialog, messagebox, ttk # Importa diálogos de archivos, mensajes y widgets adicionales
from PIL import Image, ImageTk # Biblioteca para la manipulación de imágenes
import csv # Biblioteca para la lectura y escritura de archivos CSV
```

Figura 12.1 Módulos Importados

tkinter: Utilizado para crear la interfaz gráfica del usuario.

filedialog: Para mostrar cuadros de diálogo para la selección de archivos.

messagebox: Para mostrar mensajes emergentes.

ttk: Para widgets adicionales como el Treeview y el Scrollbar.

PIL: Para manipulación de imágenes, carga y visualización en la interfaz.

csv: Para leer y escribir archivos CSV.

12.2 Función para Calcular IMC

```
def calcularImc():
    # Obtener los valores de peso, altura, edad y sexo de las entradas
    peso = float(entryPeso.get()) # Convertir el valor de peso a float
    altura = float(entryAltura.get()) # Convertir el valor de altura a float
    edad = int(entryEdad.get()) # Convertir el valor de edad a int
    sexo = varSexo.get() # Obtener el valor del sexo seleccionado

    # Validar que la altura no sea 0 para evitar división por cero
    if altura == 0:
        messagebox.showerror("Error", "La altura no puede ser 0") # Mostrar mensaje de error
        return

    # Definir el factor ks según el sexo
    ks = 1.0 if sexo == "Hombre" else 1.1 # Establecer el factor ks basado en el sexo
    # Calcular el factor ka basado en la edad
    ka = 1 + 0.01 * (edad - 25) # Calcular el factor ka basado en la edad

    # Calcular el IMC usando los factores ks y ka
    imc = peso / (altura**2) * ks * ka # Fórmula para el cálculo del IMC
    # Determinar la categoría del IMC
    if imc < 18.5:
        categoria = "Bajo peso" # Categoría para IMC menor a 18.5
    elif 18.5 <= imc < 24.9:
        categoria = "Peso normal" # Categoría para IMC entre 18.5 y 24.9
    elif 25 <= imc < 29.9:
        categoria = "Sobrepeso" # Categoría para IMC entre 25 y 29.9
    else:
        categoria = "Obesidad" # Categoría para IMC mayor o igual a 30

    # Actualizar la etiqueta de resultado con el IMC y la categoría
    labelResultado.config(text=f"IMC: {imc:.2f} - {categoria}") # Mostrar el IMC y la categoría en la etiqueta
    return imc, categoria # Retornar el IMC y la categoría
```

Figura 12.2 Función de Cálculo de IMC.

calcularImc: Esta función calcula el IMC basado en el peso, altura, edad y sexo del usuario. Valida la entrada, aplica factores ajustados por sexo y edad, y determina la categoría del IMC.

12.3 Función para Guardar Datos en CSV

```
def guardarDatos():
    # Obtener los valores de las entradas
    nombre = entryNombre.get() # Obtener el nombre del usuario
    peso = entryPeso.get() # Obtener el peso
    altura = entryAltura.get() # Obtener la altura
    edad = entryEdad.get() # Obtener la edad
    sexo = varSexo.get() # Obtener el sexo

    # Calcular el IMC y la categoría
    imc, categoria = calcularImc() # Llamar a la función de cálculo del IMC

    # Preparar los datos para guardar
    datos = [nombre, peso, altura, edad, sexo, f"{imc:.2f}", categoria] # Crear lista con los datos a guardar

    # Guardar los datos en un archivo CSV con el nombre del usuario
    with open(f"{nombre}.csv", mode="w", newline="") as file: # Abrir un archivo CSV para escritura
        writer = csv.writer(file) # Crear un escritor de CSV
        writer.writerow(["Nombre", "Peso (kg)", "Altura (m)", "Edad", "Sexo", "IMC", "Categoría"]) # Escribir encabezados
        writer.writerow(datos) # Escribir los datos
```

Figura 12.3 Función de Almacenamiento de Datos.

guardarDatos: Esta función guarda los datos del usuario, incluidos los resultados del IMC, en un archivo CSV. El nombre del archivo es el nombre del usuario.

12.4 Función para Leer Datos desde un Archivo CSV

```
def leerCsv():
    # Abrir un cuadro de diálogo para seleccionar el archivo CSV
    filePath = filedialog.askopenfilename(filetypes=[("CSV files", "*.csv")]) # Mostrar diálogo para seleccionar archivo
    if filePath:
        # Leer los datos del archivo CSV
        with open(filePath, mode="r", newline="") as file: # Abrir el archivo CSV para lectura
            reader = csv.reader(file) # Crear un lector de CSV
            headers = next(reader) # Leer los encabezados de las columnas
            datos = list(reader) # Leer el resto de los datos en una lista

            # Crear una nueva ventana para mostrar los datos
            ventanaCsv = tk.Toplevel(ventanaPrincipal) # Crear una ventana secundaria
            ventanaCsv.title("Datos CSV") # Establecer el título de la ventana
            ventanaCsv.geometry("600x400") # Establecer el tamaño de la ventana
            ventanaCsv.configure(bg='#F5F5F5') # Establecer el color de fondo de la ventana

            # Crear un Frame para el Treeview y el scrollbar
            frameTreeview = tk.Frame(ventanaCsv, bg='#F5F5F5') # Crear un marco para contener el Treeview
            frameTreeview.pack(expand=True, fill="both", padx=10, pady=10) # Añadir el marco al contenedor principal

            # Crear el Treeview con estilo
            estilo = ttk.Style() # Crear un objeto de estilo
            estilo.theme_use("clam") # Usar el tema "clam" para el estilo
            estilo.configure("Treeview",
                             background="#D3D3D3",
                             foreground="black",
                             rowheight=25,
                             fieldbackground="#D3D3D3") # Configurar el estilo del Treeview
            estilo.map('Treeview', background=[('selected', '#347883')]) # Configurar el color de selección del Treeview
```

Figura 12.4 Función para Leer Datos.

```

# Configurar encabezados
for header in headers:
    tree.heading(header, text=header) # Establecer el texto del encabezado
    tree.column(header, width=100, anchor='center') # Configurar el ancho y la alineación de las columnas

# Insertar datos en el Treeview
for row in datos:
    tree.insert("", "end", values=row) # Añadir cada fila de datos al Treeview

# Crear y agregar scrollbar
scrollbar = ttk.Scrollbar(frameTreeview, orient="vertical", command=tree.yview) # Crear una barra de desplazamiento vertical
scrollbar.pack(side="right", fill="y") # Añadir la barra de desplazamiento al marco
tree.configure(yscroll=scrollbar.set) # Configurar el Treeview para usar la barra de desplazamiento

# Agregar etiqueta de título
labelTitulo = tk.Label(ventanaCsv, text="Visualización de Datos CSV", font=("Helvetica", 14, "bold"), bg='#F5F5F5') # Crear la etiqueta
labelTitulo.pack(pady=10) # Añadir la etiqueta a la ventana

```

Figura 12.5 Configuración de Encabezados.

leerCsv: Esta función abre un cuadro de diálogo para seleccionar un archivo CSV, lee los datos y muestra los datos en una nueva ventana utilizando un Treeview.

12.5 Creación de la Ventana Principal y Widgets

```

# Crear la ventana principal de la aplicación
ventanaPrincipal = tk.Tk() # Crear la ventana principal
ventanaPrincipal.title("Calculadora de IMC") # Establecer el título de la ventana
ventanaPrincipal.geometry("650x500") # Establecer el tamaño de la ventana

# Crear un canvas para poner la imagen de fondo
canvas = tk.Canvas(ventanaPrincipal, width=800, height=600) # Crear un lienzo para la imagen de fondo
canvas.grid(row=0, column=0, rowspan=10, columnspan=4) # Colocar el lienzo en la ventana

# Cargar la imagen de fondo
fondoImg = ImageTk.PhotoImage(Image.open("mar.jpg")) # Cargar la imagen de fondo
canvas.create_image(0, 0, anchor="nw", image=fondoImg) # Mostrar la imagen en el lienzo

# Definir fuente y color
fuente = ("Helvetica", 12, "italic") # Fuente para los textos
colorFuente = "#4B0082" # Color de la fuente

# Crear los widgets de entrada y etiquetas
labelNombre = tk.Label(ventanaPrincipal, text="Nombre:", font=fuente, fg=colorFuente) # Etiqueta para el nombre
entryNombre = tk.Entry(ventanaPrincipal, font=fuente, fg=colorFuente) # Entrada para el nombre

labelPeso = tk.Label(ventanaPrincipal, text="Peso (kg):", font=fuente, fg=colorFuente) # Etiqueta para el peso
entryPeso = tk.Entry(ventanaPrincipal, font=fuente, fg=colorFuente) # Entrada para el peso

labelAltura = tk.Label(ventanaPrincipal, text="Altura (m):", font=fuente, fg=colorFuente) # Etiqueta para la altura
entryAltura = tk.Entry(ventanaPrincipal, font=fuente, fg=colorFuente) # Entrada para la altura

labelEdad = tk.Label(ventanaPrincipal, text="Edad:", font=fuente, fg=colorFuente) # Etiqueta para la edad
entryEdad = tk.Entry(ventanaPrincipal, font=fuente, fg=colorFuente) # Entrada para la edad

```

Figura 12.6 Creación de Ventanas y Complementos.

```

labelSexo = tk.Label(ventanaPrincipal, text="Sexo:", font=fuente, fg=colorFuente) # Etiqueta para el sexo
varSexo = tk.StringVar(value="Hombre") # Variable para almacenar el sexo seleccionado
radioHombre = tk.Radiobutton(ventanaPrincipal, text="Hombre", variable=varSexo, value="Hombre", font=fuente, fg=colorFuente)
radioMujer = tk.Radiobutton(ventanaPrincipal, text="Mujer", variable=varSexo, value="Mujer", font=fuente, fg=colorFuente)

# Cargar las imágenes para los botones
imgCalcular = ImageTk.PhotoImage(Image.open("b1.png")) # Imagen para el botón de calcular
imgGuardar = ImageTk.PhotoImage(Image.open("b2.png")) # Imagen para el botón de guardar
imgLeer = ImageTk.PhotoImage(Image.open("b3.png")) # Imagen para el botón de leer CSV

# Crear botones con imágenes
botonCalcular = tk.Button(ventanaPrincipal, image=imgCalcular, command=calcularImc, borderwidth=0, cursor="hand2") # Botón de calcular
botonGuardar = tk.Button(ventanaPrincipal, image=imgGuardar, command=guardarDatos, borderwidth=0, cursor="hand2") # Botón de guardar
botonLeerCsv = tk.Button(ventanaPrincipal, image=imgLeer, command=leerCsv, borderwidth=0, cursor="hand2") # Botón de leer CSV

labelResultado = tk.Label(ventanaPrincipal, text="IMC:", font=fuente, fg=colorFuente) # Etiqueta para mostrar el resultado

# Ubicar los widgets en la ventana
labelNombre.place(x=50, y=20) # Posicionar la etiqueta del nombre
entryNombre.place(x=200, y=20, width=100, height=20) # Posicionar la entrada del nombre

labelPeso.place(x=50, y=60) # Posicionar la etiqueta del peso
entryPeso.place(x=200, y=60, width=100, height=20) # Posicionar la entrada del peso

labelAltura.place(x=50, y=100) # Posicionar la etiqueta de la altura
entryAltura.place(x=200, y=100, width=100, height=20) # Posicionar la entrada de la altura

labelEdad.place(x=50, y=140) # Posicionar la etiqueta de la edad
entryEdad.place(x=200, y=140, width=100, height=20) # Posicionar la entrada de la edad

```

Figura 12.7 Elementos dentro del Programa.

```

labelEdad.place(x=50, y=140) # Posicionar la etiqueta de la edad
entryEdad.place(x=200, y=140, width=100, height=20) # Posicionar la entrada de la edad

labelSexo.place(x=50, y=180) # Posicionar la etiqueta del sexo
radioHombre.place(x=200, y=180) # Posicionar el RadioButton para Hombre
radioMujer.place(x=300, y=180) # Posicionar el RadioButton para Mujer

botonCalcular.place(x=50, y=220, width=300, height=50) # Posicionar el botón de calcular
labelResultado.place(x=50, y=280, width=300) # Posicionar la etiqueta del resultado

botonGuardar.place(x=50, y=320, width=300, height=50) # Posicionar el botón de guardar
botonLeerCsv.place(x=50, y=380, width=300, height=50) # Posicionar el botón de leer CSV

# Iniciar el bucle principal de la aplicación
ventanaPrincipal.mainloop() # Ejecutar el bucle principal de la interfaz gráfica

```

Figura 12.8 Posicionamiento de Elementos.

ventanaPrincipal: La ventana principal de la aplicación.

canvas: Un lienzo para mostrar una imagen de fondo.

labelNombre, entryNombre, labelPeso, entryPeso, labelAltura, entryAltura, labelEdad, entryEdad, labelSexo, radioHombre, radioMujer: Widgets para ingresar datos del usuario.

imgCalcular, imgGuardar, imgLeer: Imágenes para los botones.

botonCalcular, botonGuardar, botonLeerCsv: Botones para calcular el IMC, guardar datos y leer archivos CSV.

labelResultado: Etiqueta para mostrar el resultado del cálculo del IMC.

13 Funcionalidad

❖ **Iniciar la Aplicación:** Ejecuta el archivo Python que contiene el código del programa.

Ingresar Datos:

- **Nombre:** Escribe tu nombre en el campo correspondiente.
- **Peso (kg):** Ingresa tu peso en kilogramos.
- **Altura (m):** Ingresa tu altura en metros.
- **Edad:** Ingresa tu edad en años.
- **Sexo:** Selecciona tu sexo (Hombre o Mujer).

Calcular IMC:

- Haz clic en el botón "Calcular" para calcular tu IMC.
- El resultado del IMC y su categoría (Bajo peso, Peso normal, Sobrepeso, Obesidad) se mostrarán en la etiqueta de resultados.

Guardar Datos:

- Haz clic en el botón "Guardar" para guardar los datos ingresados y el resultado del IMC en un archivo CSV.

Leer Datos desde CSV:

- Haz clic en el botón "Leer CSV" para abrir un cuadro de diálogo que te permitirá seleccionar un archivo CSV y visualizar sus contenidos en una nueva ventana.

14 Mantenimiento

Actualización de la Biblioteca:

- Asegúrate de mantener actualizadas las bibliotecas utilizadas (tkinter, PIL, csv) para aprovechar las últimas características y correcciones de errores.

Revisión del Código:

- Revisa periódicamente el código para mejorar la eficiencia y corregir posibles errores.

- Añade nuevos comentarios si realizas cambios importantes en el código.

Gestión de Archivos CSV:

- Verifica que los archivos CSV generados se almacenan de manera organizada y se realizan copias de seguridad periódicas.

Pruebas:

- Realiza pruebas regulares para asegurarte de que el programa funcione correctamente después de cualquier actualización o cambio en el código.

15 Elementos Importantes

Función calcularImc:

- Calcula el IMC y determina la categoría correspondiente.

Función guardarDatos:

- Guarda los datos del usuario y los resultados del IMC en un archivo CSV.

Función leerCsv:

- Lee los datos de un archivo CSV seleccionado por el usuario y los muestra en una nueva ventana.

Interfaz Gráfica:

- Proporciona una interfaz fácil de usar para ingresar datos, calcular el IMC y guardar/leer datos de archivos CSV.

Gestión de Archivos:

- Utiliza archivos CSV para almacenar y gestionar los datos del usuario de manera sencilla y accesible.

16 Diagrama de Flujo

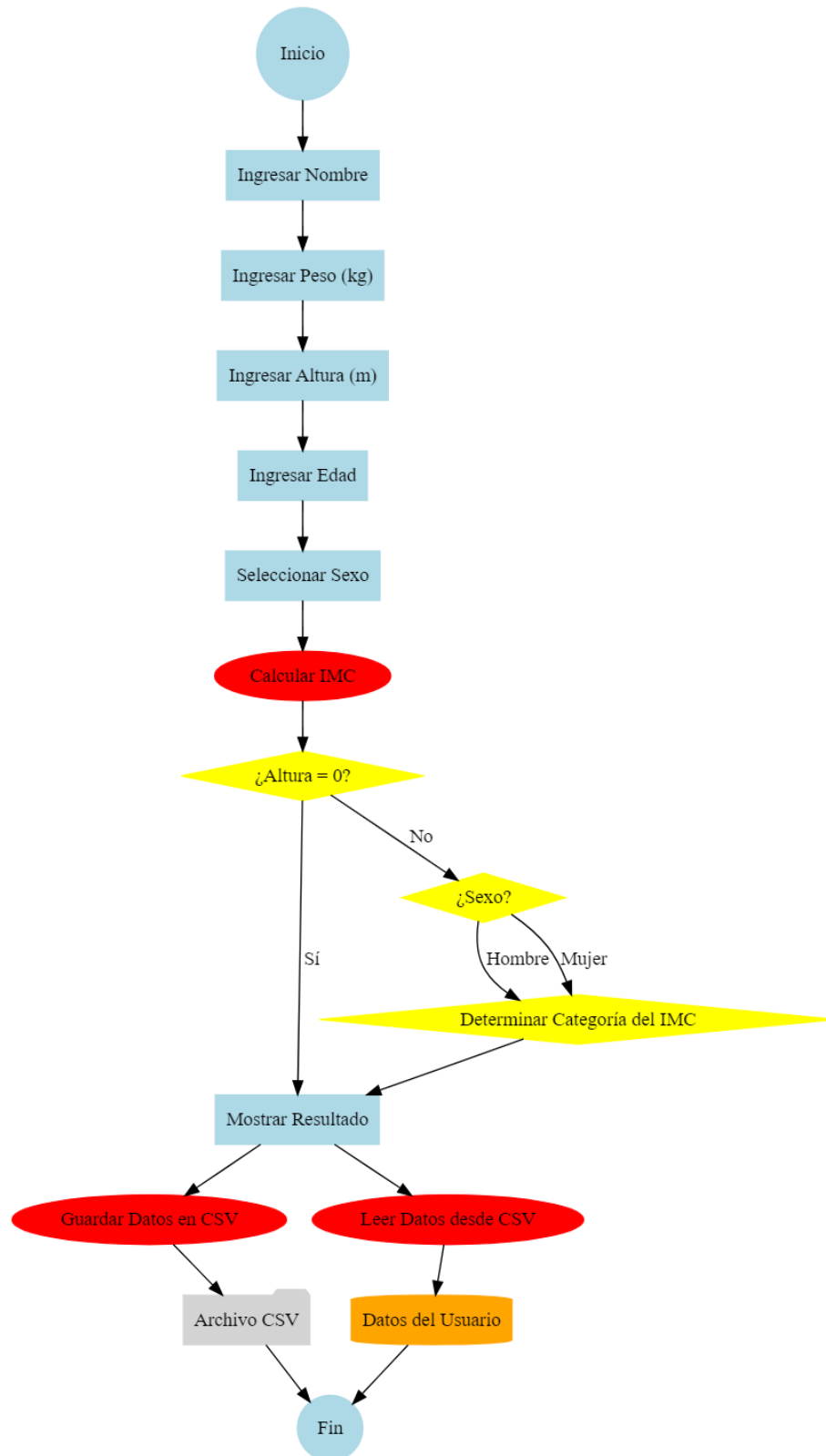


Figura 16.1 Imagen del Diagrama de Flujo.

17 Descripción Del Diagrama

❖ Inicio y Fin:

- **Inicio:** Nodo inicial del proceso, marcado como "Inicio".
- **Fin:** Nodo final del proceso, marcado como "Fin".

❖ Entradas del Usuario:

- **Ingresar Nombre:** El usuario ingresa su nombre.
- **Ingresar Peso (kg):** El usuario ingresa su peso en kilogramos.
- **Ingresar Altura (m):** El usuario ingresa su altura en metros.
- **Ingresar Edad:** El usuario ingresa su edad.
- **Seleccionar Sexo:** El usuario selecciona su sexo (Hombre o Mujer).

❖ Cálculos y Validaciones:

- **Calcular IMC:** El sistema calcula el IMC basado en los datos ingresados.
- **¿Altura = 0?:** Verificación de si la altura ingresada es 0, lo que generaría un error en el cálculo.
- **¿Sexo?:** Determinación del sexo del usuario para ajustar el cálculo del IMC.
- **Determinar Categoría del IMC:** Clasificación del IMC calculado en una categoría (e.g., Bajo peso, Peso normal, Sobrepeso, Obesidad).

❖ Resultados y Almacenamiento:

- **Mostrar Resultado:** Presentación del IMC y su categoría al usuario.
- **Guardar Datos en CSV:** Guardar los datos del usuario y el resultado del IMC en un archivo CSV.
- **Leer Datos desde CSV:** Leer y mostrar los datos desde un archivo CSV existente.

❖ Archivos y Datos:

- **Archivo CSV:** Representación de un archivo CSV donde se almacenan los datos del usuario.
- **Datos del Usuario:** Nodo que representa los datos del usuario leídos desde un archivo CSV.

17.1 Relaciones entre Nodos

❖ Secuencia de Entrada:

- El proceso inicia en el nodo "Inicio".
- Secuencialmente, el usuario ingresa su nombre, peso, altura, edad y selecciona su sexo.

❖ Cálculo del IMC:

- Tras ingresar el sexo, el flujo se dirige al nodo "Calcular IMC".
- Luego, se verifica si la altura es 0. Si es 0, el flujo va directamente a "Mostrar Resultado" con un mensaje de error. Si no es 0, se dirige a la verificación del sexo.

❖ Determinación del Sexo y Categoría:

- Dependiendo del sexo del usuario (Hombre o Mujer), el flujo se dirige a la determinación de la categoría del IMC.
- Después de determinar la categoría del IMC, el resultado se muestra al usuario.

❖ Almacenamiento y Lectura de Datos:

- Desde "Mostrar Resultado", el usuario puede optar por guardar los datos en un archivo CSV o leer datos desde un archivo CSV existente.
- Si se guardan los datos, el flujo va al "Archivo CSV".
- Si se leen los datos, el flujo va a "Datos del Usuario".

❖ Finalización:

- Tanto el "Archivo CSV" como "Datos del Usuario" conducen al nodo "Fin", terminando el proceso.