



UNIVERSIDAD POLITÉCNICA DE TEXCOCO

INGENIERÍA EN ROBÓTICA

APLICACIÓN PARA CÁLCULO DE ÍNDICE DE MASA CORPORAL

GRUPO: 8VSC2

INTEGRANTES:

1.- Diego Antonio de la Cruz Cruz





■

Indice

Objetivos	4
Introducción	5
Fórmula para calcular el índice de masa corporal	5
Cálculo de índice de masa Corporal	6
Instalación	6
Requerimientos	6
Módulos Utilizados	6
Herramientas para ejecutable	7
Descripción de uso	7
Formulario de la aplicación	8
Descripción de Código	9
Implementación de módulos	9
Código de interfaz	9
Código de Formulario	10
Diagrama de flujo	14

■



Indice de figuras

Imag formula general	5
Imag calculo	6
Imag interfaz	7
Imag Formulario de la aplicación	8
Imag Calculo	8
Imag Ventana de exito	8
Imag Archivo CSV	9
Imag modulos	9
Imag interfaz	10
Imag configuracion de interfaz	10
Imag configuracion de texto	11
Imag configuración cursor, sexo y guardar	11
Imag, configuración de cálculos	12
Imag guardar la informacion	13
Imag diagrama de funcionamiento	14

Índice de tabla

Tabla de requerimiento	6
------------------------	---

Objetivos

En este documento tiene como objetivo hacer un manual el cual los usuarios podrán hacer uso de una aplicación para calcular el índice de la masa corporal para obtener el resultado del estado de su salud

Objetivo General:

Desarrollar una aplicación móvil para calcular el índice de masa corporal (IMC) de los usuarios, proporcionando resultados precisos y recomendaciones personalizadas sobre la salud y el bienestar basadas en el IMC calculado.

Objetivos Específicos:

- Diseñar una interfaz de usuario intuitiva y accesible que permita a los usuarios ingresar sus datos personales, como peso, altura y edad, de manera sencilla y rápida, asegurando una experiencia de usuario eficiente y amigable.
- Implementar un algoritmo preciso para el cálculo del IMC que utilice los datos proporcionados por los usuarios y genere resultados exactos, además de clasificar el IMC en categorías estandarizadas como bajo peso, peso normal, sobrepeso y obesidad.
- Desarrollar una funcionalidad de recomendaciones personalizadas basada en el IMC calculado, que ofrezca sugerencias prácticas para mejorar la salud, como cambios en la dieta y recomendaciones de ejercicio, adaptadas a las necesidades individuales de cada usuario.

Introducción

En la era moderna, donde el bienestar y la salud son prioridades cada vez mayores, contar con herramientas accesibles para el monitoreo y la mejora de nuestra salud personal se ha vuelto indispensable. El Índice de Masa Corporal (IMC) es una métrica ampliamente utilizada para evaluar la relación entre el peso y la altura de una persona, proporcionando una estimación general del estado nutricional y el riesgo de salud asociado a problemas como el sobrepeso y la obesidad.

En este contexto, el desarrollo de una aplicación móvil para calcular el IMC surge como una solución innovadora y conveniente. Esta aplicación está diseñada para permitir a los usuarios obtener una evaluación rápida y precisa de su IMC, con la capacidad de ingresar datos como peso, altura y edad de manera sencilla e intuitiva. Más allá del cálculo básico, la aplicación ofrece una interfaz amigable y accesible que facilita la interacción del usuario y asegura una experiencia sin complicaciones.

Fórmula para calcular el índice de masa corporal

Índice de Masa Corporal

$$\text{IMC} = \frac{\text{Peso (Kg)}}{\text{Altura (m)}^2}$$

Imag formula general

Nota: a la formula general se la agregado dos datos adicionales KS y KA

KS = 1.0 si es hombre y 1.1 si es mujer

KA = para ajustar la edad $1+0.01*(\text{edad}-25)$

Cálculo de índice de masa Corporal

$$IMC = \frac{\text{peso KG}}{(\text{Altura})^2} * K_S * K_A$$

$$IMC = \frac{54 \text{ KG}}{(1.54)^2} * 1.0 * 1$$

$$IMC = 22.769$$

calculo para
hombre

$$IMC = \frac{54 \text{ KG}}{(1.54)^2} * 1.1 * 1$$

$$IMC = 25.046$$

calculo para
mujer

Imag calculo Instalación

El lenguaje que se utilizo para crear la aplicación fue Python el cual podemos descargarlo en su pagina oficial <https://www.python.org/> y se utilizo el editor de código Visual Studio Code el cual lo puede descargar en el siguiente link <https://code.visualstudio.com/> el cual se pueden instalar en sistemas operativos como Linux, Windows, etc.

Requerimientos

La aplicación en si no necesita mucha especificación para ser utilizado, pero en esta ocasión la aplicación fue creado en el un sistema de operativo

Sistema Operativo	Procesador	RAM	Espacio en disco	Graficos
Windows	Cualquier procesador en tu sistema es aceptable	8 GB	1 TB de almacenamiento	No se requiera de ninguna tarjeta grafica

Tabla de requerimiento

Módulos Utilizados

Modulo Tkinter: es la interfaz por defecto de Python para el kit de herramientas de GUI Tk. Tanto Tk como tkinter están disponibles en la mayoría de las plataformas Unix, así como en sistemas Windows (Tk en sí no es parte de Python, es mantenido por ActiveState).

Modulo CSV: Los archivos con formato csv (Comma Separated Values) son archivos de texto, que en cada línea contienen valores separados por algún delimitador (por defecto,

una coma). Es uno de los formatos más utilizados para importar y exportar datos tales como planillas y bases de datos.

Herramientas para ejecutable

Pip install pyinstaller: se utiliza para empaquetar código Python en aplicaciones ejecutables independientes para varios sistemas operativos. Toma un script de Python y genera un solo archivo ejecutable que contiene todas las dependencias necesarias y puede ejecutarse en computadoras que no tienen Python instalado.

Pip install auto-py-to-exe: Es una aplicación de escritorio con una interfaz gráfica (GUI) que facilita el proceso de conversión de scripts de Python en archivos ejecutables (.exe para Windows, o el formato correspondiente para otros sistemas operativos). Proporciona una forma más amigable de usar PyInstaller, permitiéndote configurar las opciones y ver una vista previa de la conversión sin necesidad de escribir comandos complejos en la línea de comandos.

Descripción de uso

Una vez que tenemos el ejecutable hecho lo que hacemos es hacerlo funcionar

Interfaz de inicio: esta parte es la inicial el cual deberá seleccionar el botón



Imag interfaz

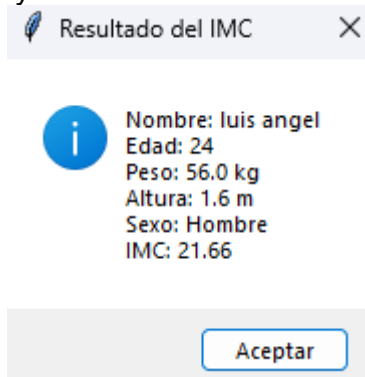
Formulario de la aplicación



Imag Formulario de la aplicación

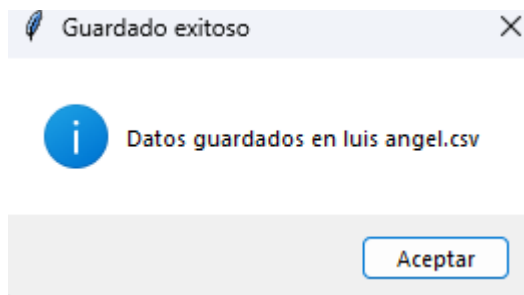
Botones

Botón para Calcular: este botón tiene la funcionalidad de calcular el índice de masa corporal con los datos obtenidos y los muestra en una ventana donde tiene la información

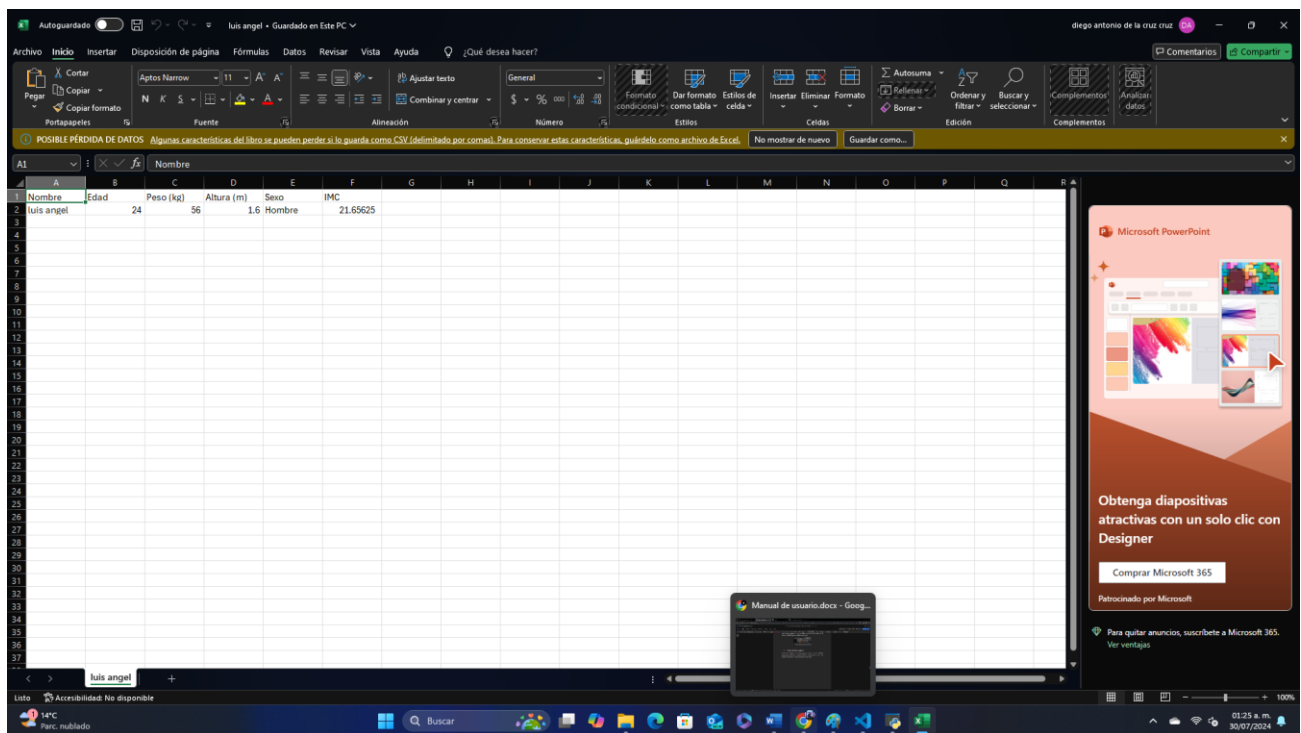


Imag Calculo

Botón de guardar: El botón cumple con la funcionalidad para guardar la información en un archivo CSV



Imag Ventana de exito



Imag Archivo CSV

Descripción de Código Implementación de módulos

```

1  import tkinter as tk #modulo para crear ventanas
2  from tkinter import PhotoImage, messagebox, Toplevel # se usa para mostrar una ventana
3  import tkinter.font as tkfont #creación y gestión de fuentes personalizadas en widgets
4  import csv #Permite manejar archivos CSV fácilmente
5

```

Imag modulos
Código de interfaz

```

145 def button_clickud():
146     # Crea la ventana principal y configura la interfaz gráfica
147     root = tk.Tk()
148     root.title("Botón con IMC")
149
150     # Configura el tamaño de la ventana principal
151     root.geometry("1100x640") # Tamaño arbitrario
152
153     # Carga y configura la imagen de fondo
154     backimage = tk.PhotoImage(file="ejemplo.png")
155     backlabel = tk.Label(root, image=backimage)
156     backlabel.place(relwidth=1, relheight=1)
157
158     # Carga la imagen para el botón
159     butima = PhotoImage(file="boton.png")
160
161     # Crea el botón con la imagen y la función de clic para abrir la calculadora IMC
162     imabutton = tk.Button(root, image=butima, command=abrir_calculadora_imc, borderwidth=4, cursor="hand2")
163     imabutton.pack(pady=100, padx=100) # Centra el botón en la ventana
164
165     # Ejecuta el bucle principal de la ventana
166     root.mainloop()
167
168 def abrir_calculadora_imc():
169     # Crea una ventana emergente para la calculadora IMC
170     root = tk.Toplevel() # Usa Toplevel para abrir una nueva ventana sin cerrar la principal
171     app = IMCCalculator(root)
172     root.mainloop()
173
174 if __name__ == "__main__":
175     button_clickud()
176

```

Imag interfaz Código de Formulario

```

6 class IMCCalculator:
7     def __init__(self, root):
8         # Inicializa la calculadora de IMC y configura la interfaz gráfica
9         self.root = root
10        self.root.title("Calculadora de IMC")
11
12        # Configura la imagen de fondo para la ventana principal
13        self.configurar_fondo(["ejemplo.png"])
14
15        # Configuración de la fuente y propiedades del texto
16        font = ('Arial', 20) # Fuente Arial, tamaño 20
17
18        # Configura etiquetas para los campos de entrada con fuente Arial
19        self.labnom = tk.Label(root, text="Nombre:", font=font, fg='green')
20        self.labnom.grid(row=0, column=0, padx=10, pady=10, sticky=tk.E)
21
22        self.labedad = tk.Label(root, text="Edad (años):", font=font, fg='green')
23        self.labedad.grid(row=1, column=0, padx=10, pady=10, sticky=tk.E)
24
25        self.labpeso = tk.Label(root, text="Peso (kg):", font=font, fg='green')
26        self.labpeso.grid(row=2, column=0, padx=10, pady=10, sticky=tk.E)
27
28        self.labalt = tk.Label(root, text="Altura (m):", font=font, fg='green')
29        self.labalt.grid(row=3, column=0, padx=10, pady=10, sticky=tk.E)
30
31        self.labsexo = tk.Label(root, text="Sexo:", font=font, fg='green')
32        self.labsexo.grid(row=4, column=0, padx=10, pady=10, sticky=tk.E)

```

Imag configuracion de interfaz

```

38     # Configura las entradas de texto con la fuente Arial
39     self.entnomb = tk.Entry(root, font=font)
40     self.entnomb.grid(row=0, column=1, padx=10, pady=10)
41
42     self.entedad = tk.Entry(root, font=font)
43     self.entedad.grid(row=1, column=1, padx=10, pady=10)
44
45     self.entpeso = tk.Entry(root, font=font)
46     self.entpeso.grid(row=2, column=1, padx=10, pady=10)
47
48     self.entalt = tk.Entry(root, font=font)
49     self.entalt.grid(row=3, column=1, padx=10, pady=10)
50

```

Imag configuracion de texto

```

51     # Configura los botones de radio para seleccionar el sexo
52     self.radvar = tk.StringVar()
53     self.radvar.set("Hombre")
54     self.radhombre = tk.Radiobutton(root, text="Hombre", variable=self.radvar, value="Hombre", font=font, fg='green')
55     self.radhombre.grid(row=4, column=1, padx=10, pady=5, sticky=tk.W)
56     self.radmujer = tk.Radiobutton(root, text="Mujer", variable=self.radvar, value="Mujer", font=font, fg='green')
57     self.radmujer.grid(row=4, column=1, padx=10, pady=5, sticky=tk.E)
58
59     # Configura los botones para calcular el IMC y guardar los datos en CSV
60     self.btncal = tk.Button(root, text="Calcular IMC", command=self.calcular_imc, cursor="hand2", font=font)
61     self.btncal.grid(row=5, column=0, columnspan=2, padx=10, pady=10, sticky=tk.N+tk.S+tk.E+tk.W)
62
63     self.btnguar = tk.Button(root, text="Guardar en CSV", command=self.guardar_en_csv, cursor="hand2", font=font)
64     self.btnguar.grid(row=6, column=0, columnspan=2, padx=10, pady=10, sticky=tk.N+tk.S+tk.E+tk.W)
65
66     def configurar_fondo(self, imagen_path):
67         # Configura la imagen de fondo para la ventana principal
68         fondo = PhotoImage(file=imagen_path)
69         fonlabel = tk.Label(self.root, image=fondo)
70         fonlabel.place(x=0, y=0, relwidth=1, relheight=1)
71         fonlabel.image = fondo # Mantiene la referencia para evitar que Python elimine la imagen
72

```

Imag configuración cursor, sexo y guardar

```

73     def calcular_imc(self):
74         # Calcula el IMC (Índice de Masa Corporal) y muestra el resultado
75         try:
76             # Obtiene los valores introducidos por el usuario
77             peso = float(self.entpeso.get())
78             altura = float(self.entalt.get())
79             edad = int(self.entedad.get())
80             nombre = self.entnomb.get()
81             sexo = self.radvar.get()
82
83             # Verifica que la altura sea mayor que cero
84             if altura <= 0:
85                 messagebox.showerror("Error", "La altura debe ser mayor que cero.")
86                 return
87
88             # Calcula el IMC con los factores de corrección basados en el sexo y la edad
89             ks = 1.0 if sexo == "Hombre" else 1.1
90             ka = 1 + 0.01 * (edad - 25)
91             imc = (peso / (altura ** 2)) * ks * ka
92
93             # Muestra el IMC calculado en un mensaje emergente
94             messagebox.showinfo("IMC Calculado", f"El IMC calculado es: {imc:.2f}")
95
96             # Muestra los resultados en una ventana emergente
97             restexto = (
98                 f"Nombre: {nombre}\n"
99                 f"Edad: {edad}\n"
100                 f"Peso: {peso} kg\n"
101                 f"Altura: {altura} m\n"
102                 f"Sexo: {sexo}\n"
103                 f"IMC: {imc:.2f}"
104             )
105             messagebox.showinfo("Resultado del IMC", restexto)
106
107         except ValueError:
108             # Muestra un mensaje de error si los valores introducidos no son válidos
109             messagebox.showerror("Error", "Por favor, introduce valores numéricos válidos.")
110

```

Imag, configuración de cálculos

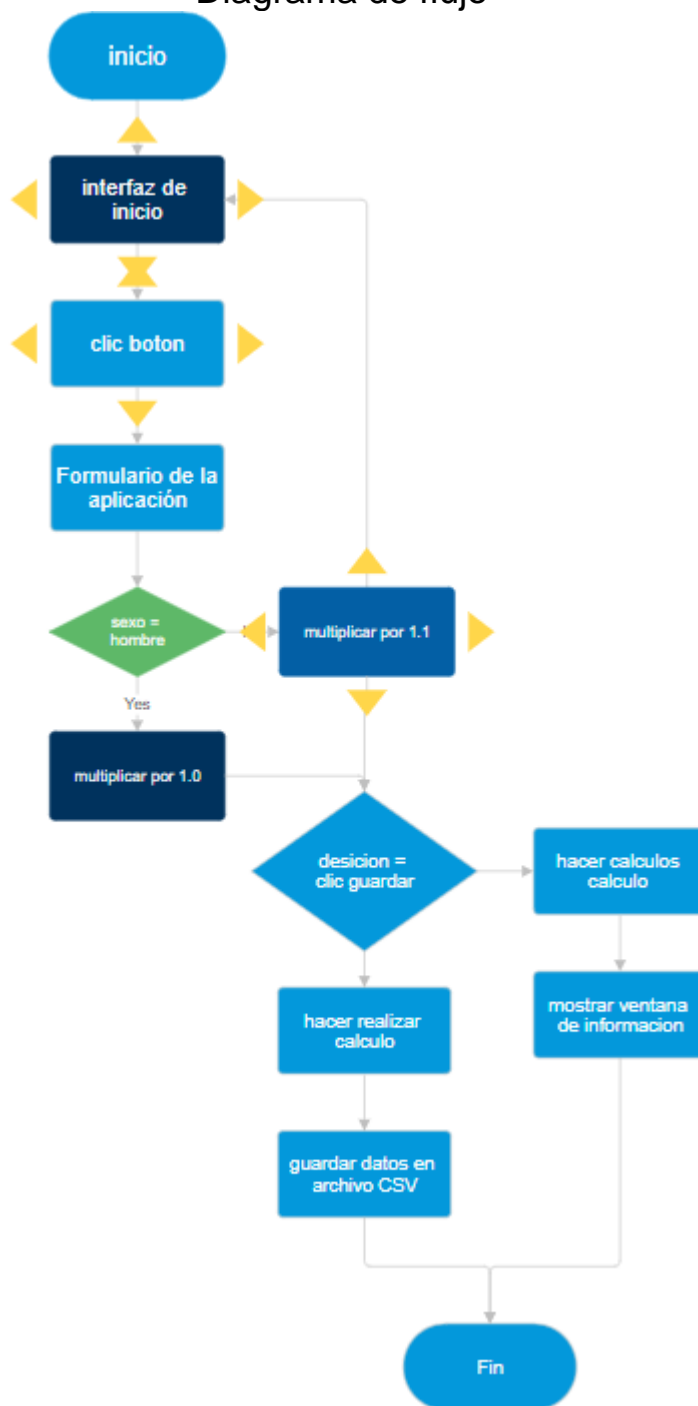
```

111 def guardar_en_csv(self):
112     # Guarda los datos introducidos en un archivo CSV
113     try:
114         # Obtiene los valores introducidos por el usuario
115         peso = float(self.entpeso.get())
116         altura = float(self.entalt.get())
117         edad = int(self.entedad.get())
118         nombre = self.entnomb.get()
119         sexo = self.radvar.get()
120
121         # Verifica que la altura sea mayor que cero
122         if altura <= 0:
123             messagebox.showerror("Error", "La altura debe ser mayor que cero.")
124             return
125
126         # Calcula el IMC con los factores de corrección basados en el sexo y la edad
127         ks = 1.0 if sexo == "Hombre" else 1.1
128         ka = 1 + 0.01 * (edad - 25)
129         imc = (peso / (altura ** 2)) * ks * ka
130
131         # Guarda los datos en un archivo CSV con el nombre del usuario
132         nombre_archivo = f"{nombre}.csv"
133         with open(nombre_archivo, mode='w', newline='') as file:
134             writer = csv.writer(file, delimiter=',', quotechar='"', quoting=csv.QUOTE_MINIMAL)
135             writer.writerow(['Nombre', 'Edad', 'Peso (kg)', 'Altura (m)', 'Sexo', 'IMC'])
136             writer.writerow([nombre, edad, peso, altura, sexo, imc])
137
138         # Muestra un mensaje de éxito al guardar el archivo
139         messagebox.showinfo("Guardado exitoso", f"Datos guardados en {nombre_archivo}")
140
141     except ValueError:
142         # Muestra un mensaje de error si los valores introducidos no son válidos
143         messagebox.showerror("Error", "Por favor, introduce valores numéricos válidos.")
144

```

Imag guardar la informacion

Diagrama de flujo



Imag diagrama de funcionamiento