

PROGRAMACION Lab 0

CARLOS MANUEL DIAZ, ALVARO ANDRES GUZMAN

Resumen— En esta práctica de laboratorio se revisaron y aplicaron conocimientos de Lógica de Programación utilizando C y Python. Primero, se replicó un programa en C que calcula el Índice de Masa Corporal (BMI) y se implementó su equivalente en Python, identificando similitudes y diferencias. Posteriormente, se expandió el programa para manejar múltiples personas, calculando e imprimiendo el BMI de cada una, así como el porcentaje de personas en diferentes categorías de peso. Finalmente, se introdujo la creación de funciones, modificando los programas para calcular y mostrar el peso ideal necesario para lograr un BMI de peso ideal, utilizando funciones tanto en C como en Python para modularizar el código.

Palabras clave—BMI, C y PYTHON
MODULARIDAD, ARREGLO,
PARAMETROS, FUNCION, LOGICA

I. INTRODUCCIÓN

Este laboratorio tiene como objetivo aplicar y reforzar conceptos básicos de programación aprendidos en el curso de Lógica de Programación. A través de la implementación y comparación de un programa para calcular el índice de masa corporal (BMI) en C y Python, se revisarán estructuras fundamentales como variables, sentencias condicionales, ciclos y arreglos. Además, se evaluarán las diferencias entre ambos lenguajes y se desarrollarán funciones para mejorar la modularidad del código, facilitando una comprensión más profunda y práctica de la programación.

• OBJETIVOS GENERAL

Reforzar y aplicar los conocimientos de programación aprendidos en Lógica de Programación mediante la implementación y comparación de programas en C y Python. Esto incluye el manejo de variables, sentencias condicionales, ciclos, y arreglos, así como la creación de funciones. El objetivo es mejorar la comprensión de la lógica de programación y desarrollar habilidades para traducir y adaptar código entre diferentes lenguajes.

• ESPECÍFICOS

- I. **Revisar y aplicar conceptos básicos** de programación, como declaración de variables, sentencias condicionales, ciclos y arreglos, en lenguajes de programación C y Python.
- II. **Comparar y analizar las diferencias y similitudes** entre la implementación de código en C y su traducción a Python, identificando las particularidades de cada lenguaje.
- III. **Desarrollar y utilizar funciones** para mejorar la modularidad del código, incluyendo la capacidad de calcular el peso ideal en función del índice de masa corporal (BMI), tanto en C como en Python.

II. INSTRUMENTOS

- II. Computador
- III. Entornos de trabajo como visual estudio.
- IV. Python
- V. Lenguaje C/C++
- VI. Draw.io

III. MÉTODOS.

Sentencias condicionales

1. **Ejemplo 1:** Compile y ejecute el programa, y realice pruebas con diferentes valores de peso y estatura.



Diagrama de flujo:

[Diagrama ejemplo 1 \(1\).drawio.png](#)

Pseudocódigo

[bmi.c](#)

Análisis:

¿Qué resultados se esperan?

A partir de ingresar la altura en cm de una persona se calculará el índice de masa corporal de la misma.

¿Qué procesos se deben realizar?

- Se declara la función calcBmi, que toma dos parámetros: altura en centímetros (h) y peso en kilogramos (w). La función tiene como objetivo calcular el Índice de Masa Corporal (IMC) utilizando estos dos valores y retornar el resultado.
- Se hace uso de la función main. La función main es el punto de entrada de cualquier programa en lenguaje C.
- Posteriormente, se declaran las variables height, weight y bmi como flotantes. Las variables height y weight almacenarán los valores ingresados por el usuario, y bmi almacenará el resultado del cálculo del IMC.
- La función main solicita al usuario que ingrese su altura en centímetros y peso en kilogramos. Estos valores se leen y almacenan en las variables height y weight, respectivamente.
- La función calcBmi se utiliza para calcular el IMC usando los valores de height y weight.
- Finalmente, se imprime en pantalla el valor del IMC calculado, que se muestra al usuario con un formato adecuado.

¿Qué variables se deben utilizar?

Se utilizan variables

- height;
- weight;
- bmi

Todas estas de tipo flotante

Prueba de escritorio:

```

PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTO

PS C:\Users\Richard Baez\.vscode> & 'c:\Users\Richard Baez\.vscode\bin\Debug\launcher.exe' '--stdin=Microsoft-MIEngine-In-dojvirs;Error-encnwhjd.mo3' '--pid=Microsoft-MIEngine-Pid-mgkddad1
Enter your height in cms: 190
Enter your weight in kgs: 78
Your body mass index is 21.606647 kg/m^2
PS C:\Users\Richard Baez\.vscode>

```

Figura 1: Prueba de escritorio de bmi.c

2. Cree un programa a partir de [bmi.c](#) que le informe al usuario si se encuentra en alguna de las siguientes condiciones: bajo peso, peso normal, sobrepeso u obesidad.

Diagrama de flujo: [Diagrama ejemplo 1 mod \(2\).drawio.png](#)

Pseudocódigo:

[codigo1.c](#)

Análisis:

¿Qué resultados se esperan?

Se ingresa la altura y peso de la persona y el programa le informara en cual de las siguientes condiciones se encuentra: bajo peso, peso normal, sobrepeso u obesidad

¿Qué procesos se deben realizar?

- **Declaración de Funciones:** Se define la función calcBmi para calcular el Índice de Masa Corporal (IMC) a partir de la altura en centímetros y el peso en kilogramos. También se declara determineCategory para determinar la categoría de peso basada en el IMC.
- **Inicialización en main:** Se declaran las variables altura, peso e imc. Se solicita al usuario que ingrese su altura y peso, y se valida que las entradas sean positivas y numéricas.
- **Cálculo del IMC:** La función calcBmi se llama con los valores de altura y peso proporcionados por el usuario para calcular el IMC, y el resultado se almacena en la variable imc.
- Se imprime el valor del IMC calculado con formato de dos decimales.
- **Determinación de Categoría:** Se llama a determineCategory con el IMC calculado para determinar y mostrar la categoría de peso correspondiente (bajo peso, peso normal, sobrepeso u obesidad).

¿Qué variables se deben utilizar?

- height
- weight
- bmi

Prueba de escritorio.

```

PROBLEMAS  SALIDA  CONSOLA DE DEPURACION  TERMINAL

PS C:\Users\Richard Baez\.vscode> & 'c:\Users\
DebugLauncher.exe' '--stdin=Microsoft-MIEngine-
Error-s0lzyqt0.cep' '--pid=Microsoft-MIEngine-
Enter your height in cms: 190
Enter your weight in kgs: 56
Your body mass index is 15.51 kg/m^2
You are underweight
PS C:\Users\Richard Baez\.vscode> ^C
PS C:\Users\Richard Baez\.vscode>
PS C:\Users\Richard Baez\.vscode> & 'c:\Users\
DebugLauncher.exe' '--stdin=Microsoft-MIEngine-
Error-0b0cb3nz.dwx' '--pid=Microsoft-MIEngine-
Enter your height in cms: 160
Enter your weight in kgs: 100
Your body mass index is 39.06 kg/m^2
You are in grade 2 obesity
PS C:\Users\Richard Baez\.vscode>

```

Figura 2: Prueba de escritorio calculo de grado de peso de una persona

3. Ejemplo de código 1 (bmi.c) implementado en Python mediante Chat GTP

Diagrama de flujo:

[Diagrama ejemplo 1 mod \(2\).drawio.png](#)

Pseudocódigo:

[codigo2.py](#)

Análisis:

¿Cuáles son los resultados que se esperan?

A partir de los datos ingresados los cuales son peso y altura se calcula la masa corporal

¿Qué procesos se deben realizar?

- Se declara la función `calc_bmi`, que toma las variables `height` (altura) y `weight` (peso). La función tiene como objetivo calcular el Índice de Masa Corporal (IMC) usando estos dos valores y retornar el resultado.
- Se hace uso de la función `main`. La función `main` es el punto de entrada del programa en Python.
- Se declaran las variables `height` y `weight` como flotantes, que almacenarán los valores ingresados por el usuario para la altura y el peso, respectivamente.
- La función `calc_bmi` se utiliza para calcular el IMC con los valores de

altura y peso proporcionados por el usuario.

- El valor devuelto por `calc_bmi`, que es el IMC calculado, se almacena en la variable `bmi`. Finalmente, el IMC se imprime en pantalla con un formato de dos decimales.

¿Qué variables se deben utilizar?

- height
- weight
- bmi

Prueba de escritorio:

```

PS C:\Users\Richard Baez\.vscode> & "C:/Users/Richard Baez/AppData
.py"
Enter your height in cms: 190
Enter your weight in kgs: 78
Your body mass index is 21.61 kg/m^2
PS C:\Users\Richard Baez\.vscode> & "C:/Users/Richard Baez/AppData
.py"
Enter your height in cms: 167
Enter your weight in kgs: 53
Your body mass index is 19.00 kg/m^2
PS C:\Users\Richard Baez\.vscode>

```

Figura 3: Prueba de escritorio de bmi en Python

4. Cree un programa a partir de [bmi.c](#) que le informe al usuario si se encuentra en alguna de las siguientes condiciones: bajo peso, peso normal, sobrepeso u obesidad mediante Python

Diagrama de flujo: [Diagrama ejemplo 1 mod \(2\).drawio.png](#)

Pseudocódigo:

[codigo3.py](#)

Análisis:

¿Cuáles son los resultados que se esperan?

Se ingresa la altura y peso de la persona y el programa le informara en cuál de las siguientes condiciones se encuentra: bajo peso, peso normal, sobrepeso u obesidad

¿Qué procesos se deben realizar?

- Se declara la función `calc_bmi`, la cual toma las variables `height` y `weight`. La función tiene como objetivo calcular el Índice de Masa Corporal (IMC) usando la fórmula: $IMC = \text{peso} / (\text{altura en metros})^2$. La altura en centímetros se convierte a metros dividiendo por 100.



- Se hace uso de la función main. La función main es el punto de entrada de cualquier programa en Python y se encarga de coordinar la ejecución del programa.
- Se solicitan al usuario los valores de altura en centímetros y peso en kilogramos mediante entradas de teclado. Estos valores se almacenan en las variables height y weight
- Se llama a la función calc_bmi con los valores ingresados por el usuario. La función devuelve el IMC calculado, el cual se almacena en la variable bmi.
- Se imprime el IMC en la pantalla con un formato de dos decimales para facilitar su lectura.
- Se determina e imprime la categoría de peso según el valor del IMC calculado. Las categorías son: bajo peso, peso normal, sobrepeso, obesidad grado 1, obesidad grado 2, y obesidad grado 3.
- La ejecución de la función main solo ocurre si el script es ejecutado directamente, gracias a la comprobación `if __name__ == "__main__":`. Esto asegura que main solo se ejecute cuando el archivo es el programa principal, no cuando es importado como módulo.

¿Qué variables se deben utilizar?

Se utilizan variables de tipo float

- height;
- weight;
- bmi

Prueba de escritorio:

```
PS C:\Users\Richard Baez\.vscode>
/Richard Baez/AppData/Local/Microsoft/WindowsApps/python3.11.0
Enter your height in cms: 34
Enter your weight in kgs: 56
Your body mass index is 484.43 kg/m^2
You are in grade 3 obesity
PS C:\Users\Richard Baez\.vscode> & "C:/Users/Richard Baez/AppData/Local/Microsoft/WindowsApps/python3.11.0"
Enter your height in cms: 167
Enter your weight in kgs: 100
Your body mass index is 35.86 kg/m^2
You are in grade 2 obesity
PS C:\Users\Richard Baez\.vscode> & "C:/Users/Richard Baez/AppData/Local/Microsoft/WindowsApps/python3.11.0"
Enter your height in cms: 1
```

Figura 4: Prueba de escritorio Python código1.c

para cada una de ellas, calcular, almacenar e imprimir el BMI de cada persona

Diagrama de flujo:

[Diagrama BMI4.drawio](#)

Pseudocódigo:

[codigo4.c](#)

Análisis:

¿Cuáles son los resultados que se esperan?

¿Qué procesos se deben realizar?

- Definir la función calcBmi
- h: Altura en centímetros.
- w: Peso en kilogramos.
- Cálculo del IMC
- Convertir la altura a metros dividiendo h por 100.0.
- Calcular el cuadrado de la altura en metros.
- Dividir el peso w entre el cuadrado de la altura en metros para obtener el IMC.
- Valor calculado del IMC.

¿Qué variables se deben utilizar?

- num_personas
- altura
- peso
- bmi
- h
- w
- Bmi

Ciclos y Arreglos

1. Crear un programa a partir de bmi.c que solicite al usuario un número de personas y que posteriormente solicite la estatura y el peso



Prueba de escritorio:

```
PS C:\Users\Richard Baez\.vscode> & 'C:\Users\Richard Baez\cher.exe' '--stdin=Microsoft-MIEngine-In-10qe11a2.j21' '--st
' '--pid=Microsoft-MIEngine-Pid-dtxy5tdx.jc1' '--dbgExe=C:\m
Ingrese el numero de personas: 3

Persona 1
Ingrese su altura en cm: 167
Ingrese su peso en kg: 67

Persona 2
Ingrese su altura en cm: 129
Ingrese su peso en kg: 21

Persona 3
Ingrese su altura en cm: 190
Ingrese su peso en kg: 78

Resultados del BMI:
Persona 1: Su índice de masa corporal es 24.02 kg/m^2
Persona 2: Su índice de masa corporal es 12.62 kg/m^2
Persona 3: Su índice de masa corporal es 21.61 kg/m^2
PS C:\Users\Richard Baez\.vscode> █
```

Figura 5: Prueba de escritorio calculo de BMI de diferentes personas

2. Modifique el programa del punto 1 para que se calcule e imprima el porcentaje de personas que se encuentra en alguna de las siguientes condiciones: bajo peso, peso normal, sobrepeso u obesidad.

Diagrama de flujo:

[Diagrama 7.png](#)

Pseudocodigo:

[codigo5.c](#)

Análisis:

¿Cuáles son los resultados que se esperan?

Implementar un algoritmo que utiliza funciones para sumar la suma de los elementos de una fila dada por el usuario. Asimismo, sumar los elementos de una columna. También determinada por el usuario.

¿Qué procesos se deben realizar?

- Definir la función calcBmi
- Calcular el Índice de Masa Corporal (IMC).
- Convertir la altura a metros, calcular el IMC como peso dividido entre el cuadrado de la altura en metros.
- Retorno: IMC calculado.
- Solicitar al usuario el número de personas e ingresar el valor en num_personas.

- Crear arreglos altura[num_personas], peso[num_personas], y bmi[num_personas] para almacenar las alturas, pesos, y IMC.
- Solicitar y leer altura y peso para cada persona.
- Calcular el IMC usando calcBmi y almacenar el resultado en bmi[i].
- Clasificar el IMC y actualizar los contadores correspondientes (bajo peso, peso normal, sobrepeso, obesidad).
- Calcular y mostrar el porcentaje de personas en cada categoría (bajo peso, peso normal, sobrepeso, obesidad).
- Retorna 0

¿Qué variables se deben utilizar?

- altura []
- peso []
- bmi []
- bajo peso
- peso normal
- sobrepeso
- obesidad
- h
- w
- bmi (en la función calcBmi)

Prueba de escritorio:

```
PROGRAMA DE INGENIERIA ELECTRONICA - CONEXION TERMINAL - TERMINAL
' '--pid=Microsoft-MIEngine-Pid-gcbrgml.02u' '--dbgExe=C:\m
Ingrese el número de personas: 3

Persona 1
Ingrese su altura en cm: 189
Ingrese su peso en kg: 201

Persona 2
Ingrese su altura en cm: 190
Ingrese su peso en kg: 78

Persona 3
Ingrese su altura en cm: 210
Ingrese su peso en kg: 89

Resultados del BMI:
Persona 1: Su índice de masa corporal es 56.27 kg/m^2
Persona 2: Su índice de masa corporal es 21.61 kg/m^2
Persona 3: Su índice de masa corporal es 20.18 kg/m^2

Porcentajes de personas en cada categoría:
Bajo peso: 0.00%
Peso normal: 66.67%
Sobrepeso: 0.00%
Obesidad: 33.33%
```

Figura 6: Prueba de escritorio de cálculo de BMI y en que condición en base al peso se encuentra.

3.Re-implemente los programas de los puntos 1 y 2 en Python



Diagrama de flujo:

[Diagrama BMI4.drawio.png](#)

Pseudocódigo:

[codigo6.py](#)

Análisis:

¿Cuáles son los resultados que se esperan?

Construir un algoritmo para calcular y mostrar la imagen negativa de una imagen en escala de grises con ayuda de funciones. Para este código se utilizarán dos en este caso para calcular la imagen negativa y otra para imprimir una matriz en la consola.

¿Qué procesos se deben realizar?

- Definir la función `calc_bmi`
- Calcular el IMC
- Convertir altura de cm a m.
- Calcular IMC como peso dividido entre el cuadrado de la altura en metros.
- Retornar el IMC calculado.
- Definir la función `main`
- Solicitar número de personas y almacenar en `num_personas`.
- Inicializar listas alturas, pesos, y bmi.
- Leer altura y peso.
- Agregar altura y peso a las listas correspondientes.
- Calcular el IMC y agregar a la lista bmi.
- Imprimir el IMC de cada persona.
- Ejecutar el programa

¿Qué variables se deben utilizar?

- `height_cm`
- `weight_kg`
- `height_m`
- `bmi`
- `num_personas`
- `alturas []`
- `pesos []`
- `bmis []`
- `altura` (dentro del bucle)
- `peso` (dentro del bucle)
- `i` (índice en el bucle)

Prueba de escritorio:

```
Ingrese el numero de personas: 2

Persona 1
Ingrese su altura en cm: 178
Ingrese su peso en kg: 56

Persona 2
Ingrese su altura en cm: 134
Ingrese su peso en kg: 67

Resultados del BMI:
Persona 1: Su índice de masa corporal es 17.67 kg/m^2
Persona 2: Su índice de masa corporal es 37.31 kg/m^2
PS C:\Users\Richard Baez\.vscode>
```

Figura7: Prueba de escritorio cálculo de BMI de diferentes personas en Python.

4. Re-implemente los programas de punto 2 en Python

Diagrama de flujo:

[Diagrama 7.png](#)

Pseudocódigo:

[codigo7.py](#)

Análisis:

¿Cuáles son los resultados que se esperan?

Al ingresar el peso y la estatura de varias personas el programa calculará el índice corporal de cada persona y el porcentaje de personas que estén en bajo peso, peso normal, sobrepeso u obesidad.

¿Qué procesos se deben realizar?

- Definir la función `calc_bmi`
- Calcular el IMC: peso dividido entre el cuadrado de la altura en metros.
- Retornar el IMC calculado.
- Definir la función `main`
- Solicitar el número de personas y almacenar en `num_personas`
- Inicializar contadores: bajo peso, peso normal, sobrepeso, obesidad a 0.
- Leer altura y peso.
- Agregar altura y peso a las listas alturas y pesos.
- Calcular el IMC usando `calc_bmi` y agregar a la lista `bmis`.



- Clasificar el IMC y actualizar los contadores correspondientes.
- Calcular y mostrar el porcentaje de personas en cada categoría (bajo peso, peso normal, sobrepeso, obesidad).
- Llamar a main () si el script se ejecuta como el programa principal.

¿Qué variables se deben utilizar?

- height_cm
- weight_kg
- height_m
- bmi
- num_personas
- bajo peso
- peso normal
- sobrepeso
- obesidad
- alturas []
- pesos []
- bmis []
- altura (dentro del bucle)
- peso (dentro del bucle)
- i (índice en el bucle)

Prueba de escritorio:

```

Ingrese el número de personas: 3

Persona 1
Ingrese su altura en cm: 178
Ingrese su peso en kg: 56

Persona 2
Ingrese su altura en cm: 190
Ingrese su peso en kg: 200

Persona 3
Ingrese su altura en cm: 170
Ingrese su peso en kg: 130

Resultados del BMI:
Persona 1: Su índice de masa corporal es 17.67 kg/m^2
Persona 2: Su índice de masa corporal es 55.40 kg/m^2
Persona 3: Su índice de masa corporal es 44.98 kg/m^2

Porcentajes de personas en cada categoría:
Bajo peso: 33.33%
Peso normal: 0.00%
Sobrepeso: 0.00%
Obesidad: 66.67%
  
```

Figura 8: Prueba de escritorio calculo de BMI y condición de peso en Python

Para terminar, en esta sección pondrá en práctica la creación de funciones en los lenguajes de programación C y Python. Recuerde que las funciones son construcciones que permiten crear código modular, escalable y de mayor legibilidad. Para lo anterior, considere los programas que construyó en los puntos 2 y 4 de la sección [Sentencias Condicionales](#).

1. Realice una modificación a ambos programas para que se le informe al usuario el peso ideal para lograr la condición de peso normal, esto solo en el caso de

que el usuario se encuentre en algunas de las siguientes condiciones: bajo peso, sobrepeso u obesidad. Para lo anterior cree una función que realice el cálculo del peso ideal.

Modificación en c:

Diagrama de flujo:

[Diagrama ejemplo funciones mod \(3\).drawio](#)

Pseudocodigo:

[codigo8.c](#)

Análisis

¿Qué resultados espera?

Este programa calcula el índice de masa corporal de una persona y le informa sobre su categoría de peso: bajo peso, peso normal, sobrepeso, obesidad grado 1, 2 o 3. Y le sugiere un rango de peso ideal para alcanzar un peso normal dentro del IMC (18.5-24.9)

¿Qué procesos se deben realizar?

- Declarar la función calc_bmi
- Parámetros: height_cm (altura en centímetros), weight_kg (peso en kilogramos)
- Objetivo: Calcular el Índice de Masa Corporal (IMC) usando estos dos valores y retornar el resultado
- Fórmula: $IMC = weight_kg / ((height_cm / 100) ** 2)$
- Definir la función main
- Punto de entrada del programa
- Declarar variables en main
- num_personas: Para almacenar el número de personas
- bajo_peso: Contador para personas con IMC bajo peso
- peso_normal: Contador para personas con IMC normal
- sobrepeso: Contador para personas con IMC sobrepeso
- obesidad: Contador para personas con IMC obesidad
- alturas: Lista para almacenar las alturas de las personas
- pesos: Lista para almacenar los pesos de las personas
- bmis: Lista para almacenar los IMC calculados
- Solicitar datos al usuario en main
- Leer y almacenar el número de personas en num_personas
- Para cada persona:
 - Leer y almacenar la altura en centímetros en altura
 - Leer y almacenar el peso en kilogramos en peso
 - Llamar a calc_bmi con altura y peso como argumentos
 - Almacenar el resultado en bmi
 - Incrementar el contador correspondiente (bajo_peso, peso_normal, sobrepeso, obesidad) basado en el valor de bmi
- Calcular e imprimir el IMC en main
- Mostrar el IMC calculado para cada persona



- Calcular e imprimir los porcentajes de personas en cada categoría en main
- Imprimir el porcentaje de personas en las categorías de bajo peso, peso normal, sobrepeso y obesidad

¿Qué variables se deben utilizar?

- Altura
- Peso
- Imc
- Pesoideal
- CalcularIMC

Prueba de escritorio:

```

Ingrese el número de personas: 3

Persona 1
Ingrese su altura en cm: 178
Ingrese su peso en kg: 56

Persona 2
Ingrese su altura en cm: 190
Ingrese su peso en kg: 200

Persona 3
Ingrese su altura en cm: 170
Ingrese su peso en kg: 130

Resultados del BMI:
Persona 1: Su índice de masa corporal es 17.67 kg/m^2
Persona 2: Su índice de masa corporal es 55.40 kg/m^2
Persona 3: Su índice de masa corporal es 44.98 kg/m^2

Porcentajes de personas en cada categoría:
Bajo peso: 33.33%
Peso normal: 0.00%
Sobrepeso: 0.00%
Obesidad: 66.67%

```

Figura 9: Prueba de escritorio de índice de masa corporal de varias personas y porcentajes de peso en c.

Modificación en Python:

Diagrama de flujo:

[Diagrama ejemplo funciones mod \(3\).drawio](#)

Pseudocódigo:

[codigo9.py](#)

Análisis:

¿Cuáles son los resultados que se esperan?

para este programa se espera que informe al usuario su índice de masa corporal (IMC) y que lo clasifique en alguna de las siguientes categorías: bajo peso, peso normal, sobrepeso u obesidad, en caso de que el usuario no se encuentre en peso normal el programa realizara un cálculo y le mostrara los rangos del peso ideal que el usuario debería tener para estar en peso normal

¿Qué procesos se deben realizar?

- Declarar la función calcular_imc
- Parámetros: altura (altura en centímetros), peso (peso en kilogramos)
- Objetivo: Calcular el Índice de Masa Corporal (IMC) usando los valores de altura y peso
- Fórmula: $IMC = peso / ((altura / 100.0) ** 2)$
- Retornar el IMC calculado
- Declarar la función peso_ideal
- Parámetros: altura (altura en centímetros), imc_objetivo (IMC objetivo)
- Objetivo: Calcular el peso ideal basado en la altura y el IMC objetivo
- Fórmula: $peso_ideal = imc_objetivo * ((altura / 100.0) ** 2)$
- Retornar el peso ideal calculado
- Definir la función main
- Solicitar datos al usuario
- Leer y almacenar la altura en centímetros en la variable altura
- Leer y almacenar el peso en kilogramos en la variable peso
- Llamar a calcular_imc con altura y peso
- Almacenar el resultado en la variable
- Si $imc < 18.5$:
- Imprimir que el usuario está bajo de peso
- Imprimir rango de peso ideal entre IMC 18.5 y 24.9
- Si $18.5 \leq imc \leq 24.9$:
- Imprimir que el usuario tiene un peso normal
- Si $25.0 \leq imc \leq 29.9$:
- Imprimir que el usuario tiene sobrepeso
- Imprimir rango de peso ideal entre IMC 18.5 y 24.9
- Si $30.0 \leq imc \leq 34.9$:
- Imprimir que el usuario tiene obesidad grado 1
- Imprimir rango de peso ideal entre IMC 18.5 y 24.9
- Si $35.0 \leq imc \leq 39.9$:
- Imprimir que el usuario tiene obesidad grado 2
- Imprimir rango de peso ideal entre IMC 18.5 y 24.9
- Si $imc \geq 40.0$:
- Imprimir que el usuario tiene obesidad grado 3
- Imprimir rango de peso ideal entre IMC 18.5 y 24.9

¿Qué variables se deben utilizar?

- Altura
- Peso
- Imc
- Peso_ideal
- Calcular_imc



Prueba de escritorio:

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

PS C:\Users\Richard Baez\.vscode>
/Richard Baez/AppData/Local/Microsoft/WindowsApps/python3.11.exe"
Ingrese su altura en centímetros: 190
Ingrese su peso en kilogramos: 45
Su índice de masa corporal es 12.47 kg/m²
Usted está bajo de peso
Su peso ideal debería estar entre 66.78 kg y 89.89 kg
PS C:\Users\Richard Baez\.vscode>
```

Figura10: Prueba de escritorio de índice de masa corporal de varias personas y porcentajes de peso en Python.

VI. CONCLUSIONES

- **Evaluación del Estado de Salud:** El BMI proporciona una medida rápida para clasificar a las personas en categorías de peso como bajo peso, peso normal, sobrepeso y obesidad. Esto ayuda a identificar posibles problemas de salud relacionados con el peso corporal.
- **Utilidad en Programación:** Implementar el cálculo del BMI y su clasificación en lenguajes de programación como C y Python demuestra la aplicabilidad de estructuras de control, ciclos, y funciones en la programación, facilitando la creación de programas funcionales y eficientes.
- **Adaptación a Diferentes Lenguajes:** Convertir el código de un lenguaje a otro (de C a Python) muestra la equivalencia de conceptos en diferentes lenguajes y ayuda a comprender cómo aplicar lógica de programación en distintos entornos.
- **Importancia de la Modularidad:** La creación de funciones para calcular el BMI y el peso ideal enfatiza la importancia de escribir código modular y reutilizable, mejorando la legibilidad y mantenibilidad del código.