

# Laboratorio 0: Lógica de Programación (RA 1)

Mariana Sierra López y Angel Albeiro Ricardo Ruiz

**Resumen—** En este documento se evidenciará los conocimientos adquiridos en lógica de programación en el lenguaje c y la implementación de un nuevo lenguaje python

## I. INTRODUCCIÓN

En la siguiente práctica se busca comprender el funcionamiento del lenguaje de programación python contemplando el repaso de ciclos y arreglos, sentencias condicionales y funciones

### OBJETIVO GENERAL

Realizar los retos propuestos por el docente en Python y comprender su lenguaje.

### OBJETIVOS ESPECÍFICOS

- Comprender el lenguaje de programación Python.
- Desarrollar el código adecuado para los retos propuesto por el docente en Python.
- Comprender el funcionamiento del código en Python.

## II. MÉTODOS E INSTRUMENTOS

En esta práctica se utilizó la aplicación Visual Studio Code.

La práctica da inicio con la realización de modificaciones por medio del lenguaje c y Python del código propuesto por el docente “bmi” que realiza el cálculo del índice de masa corporal (IMC) para una persona a partir de su peso y de la estatura.

```
1 // Include header file for prototypes of scanf and printf
2 #include <stdio.h>
3
4 // Prototype of calcBmi function
5 float calcBmi(float h,float w);
6
7 // Main function - entry point
8 int main(){
9     // local variable declaration
10    float height;
11    float weight;
12    float bmi;
13
14    // Ask user for height in cms
15    printf("Enter your height in cms: ");
16    scanf("%f",&height);
17
18    // Ask user for weight in kgs
19    printf("Enter your weight in kgs: ");
20    scanf("%f",&weight);
21
22    // call calcBmi function
23    bmi=calcBmi(height,weight);
24
25    // Print result
26    printf("Your body mass index is %f kg/m^2",bmi);
27 }
28
29 // Implementation of calcBmi function
30 float calcBmi(float h,float w){
31     float bmi;
32
33     // Compute body mass index
34     bmi=w/((h/100.0)*(h/100.0));
35     return bmi;
36 }
```

Fig 1. Código bmi

En base al código “bmi” se pide realizar las siguientes modificaciones.

1. Compile y ejecute el programa, y realice pruebas con diferentes valores de peso y estatura.
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.
3. Mediante la herramienta de Inteligencia Artificial (IA) Generativa ChatGPT, genera el equivalente en Python (bmi.py) del programa bmi.c. Identifique las similitudes y diferencias entre ambas implementaciones.

Vuelva a implementar el programa del punto 2 en Python, esta vez sin usar ChatGPT .

Para solucionar el punto número 2 se propuso el siguiente código:

```

1  #include <stdio.h>
2  // Function prototype for calcBmi
3  float calcBmi(float h, float w);
4  // Main function - entry point
5  int main() {
6      // Local variable declaration
7      float height;
8      float weight;
9      float bmi;
10     // Ask the user for height in centimeters
11     printf("Enter your height in centimeters: ");
12     scanf("%f", &height);
13     // Ask the user for weight in kilograms
14     printf("Enter your weight in kilograms: ");
15     scanf("%f", &weight);
16     // Call the calcBmi function
17     bmi = calcBmi(height, weight);
18     // Print the result
19     printf("Your body mass index is %.2f kg/m^2\n", bmi);
20     // Classify based on BMI
21     if (bmi <= 18.5) {
22         printf("You are underweight.\n");
23     } else if (bmi <= 24) {
24         printf("You are at a normal weight.\n");
25     } else if (bmi <= 30) {
26         printf("You are overweight.\n");
27     } else {
28         printf("You are obese.\n");
29     }
30
31     return 0;
32 }
33 // Implementation of the calcBmi function
34 float calcBmi(float h, float w) {
35     return w / ((h / 100) * (h / 100));
36 }

```

Fig 2. Código para el punto 2.

Para realizar el programa que se pide en el punto 2, se agregó un "if" para lograr evaluar el tipo de peso en el que se encuentra, es decir, bajo peso, peso normal, sobre peso y obesidad. Para lograr esto se investigó acerca de los rangos que indica cuando una persona se encuentra en peso normal, peso bajo, sobre peso y obeso. Finalmente se encuentra la función para realizar el cálculo del BMI.

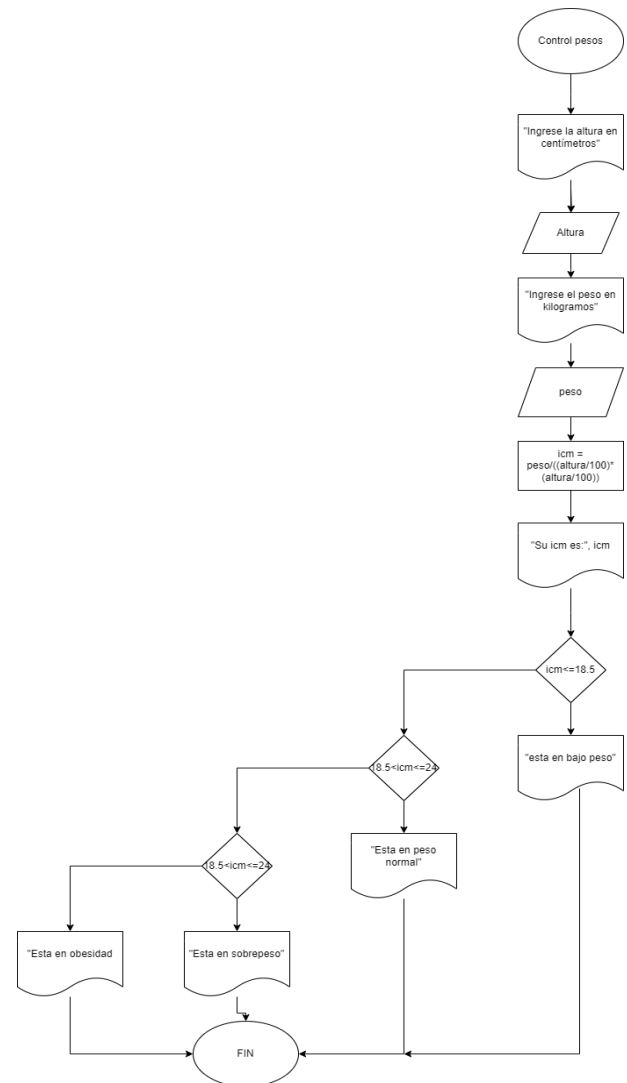


fig. 3 Diagrama de flujo del punto 2

El código de python que se solicitó realizar mediante inteligencia artificial fue elaborado de la siguiente manera:

```
# Funcion para calcular el IMC (indice de Masa Corporal)
def calc_bmi(height, weight):
    bmi = weight / ((height / 100) ** 2)

    # Determinar la categoria del IMC
    if bmi <= 18.5:
        print("You are underweight")
    elif 18.5 < bmi <= 24:
        print("You are at normal weight")
    elif 24 < bmi <= 30:
        print("You are overweight")
    elif bmi > 30:
        print("You are obese")

    return bmi

# Funcion principal
def main():
    # Solicitar la altura en centimetros
    height = float(input("Enter your height in cms: "))

    # Solicitar el peso en kilogramos
    weight = float(input("Enter your weight in kgs: "))

    # Calcular el IMC
    bmi = calc_bmi(height, weight)

    # Mostrar el resultado
    print(f"Your body mass index is {bmi:.2f} kg/m^2")

# Punto de entrada
if __name__ == "__main__":
    main()
```

Fig 4. Código del punto 3. IA.

Por medio de este código se notan grandes diferencias entre el lenguaje C y el lenguaje python principalmente por la declaración de variables. En el lenguaje C era necesario definir una variable como entero (int), carácter(char), tipo flotante(float), entre otras, antes del nombre de la variable. Para mostrar un mensaje en pantalla en lugar de utilizar un “printf” en python es utilizado como “print” y para guardar el dato de una variable se utiliza el “input” en python, para que aparezca en pantalla el resultado de la operación se utiliza “{bmi:.2f}” a diferencia del lenguaje C que se utiliza “%.2f”.

Una diferencia bastante notoria entre el lenguaje C y Python es el uso del “;” y del “{”.

No tener que utilizar el punto y coma consigue ser bastante cómodo a la hora de programar, sin embargo, esto causaría un olvido a la hora de programar en otro lenguaje como en C que es necesario el uso de éste, respecto al corchete es un poco incómodo no utilizarlo, ya que en python en las sentencias condicionales se indica que la instrucción está dentro de ella por medio de espacios.

Finalmente se tiene “if \_\_name\_\_ ==

“\_\_main\_\_” que es utilizado para verificar si el archivo actual se está ejecutando en el programa principal, si es así el código que se encuentra dentro de ese bloque se ejecutará, si el archivo no es del programa principal, no se ejecutará.

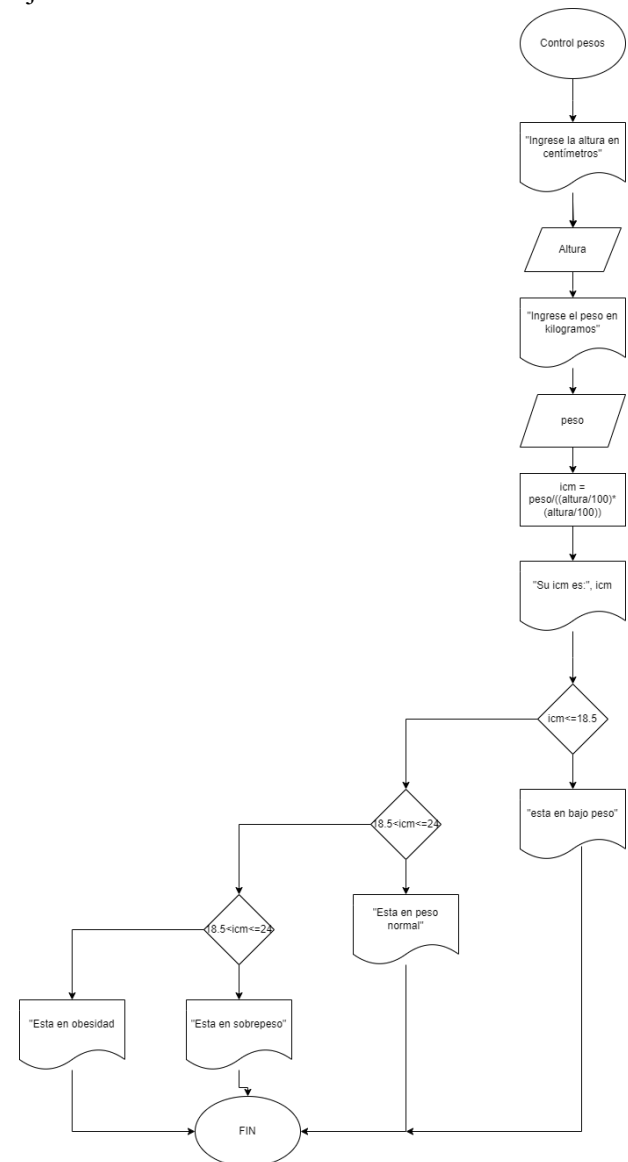


Fig 5. Diagrama de flujo del punto 3.

Para la solución del punto número cuatro que consiste en realizar 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, pero esta vez en python se obtuvo el siguiente código:

```

1 weight = float(input("Enter your weight (in kg): "))
2 height = float(input("Enter your height (in cm): "))
3 bmi = weight / ((height / 100) * (height / 100))
4 print(f"Your BMI is: {bmi:.2f}")
5
6 if bmi < 18.5:
7     print("You are underweight")
8 if 18.5 <= bmi < 24:
9     print("You are at a normal weight")
10 if 24 <= bmi < 30:
11     print("You are overweight")
12 if bmi >= 30:
13     print("You are obese")
14

```

Fig 6. Código del punto 4.

Como se observa en la fig 6 el código es mucho más sencillo que el que realizó chat GPT que se muestra en la fig 4. Una diferencia notoria es que no se utilizó funciones, y esto hace que el código se vea mucho más pequeño.

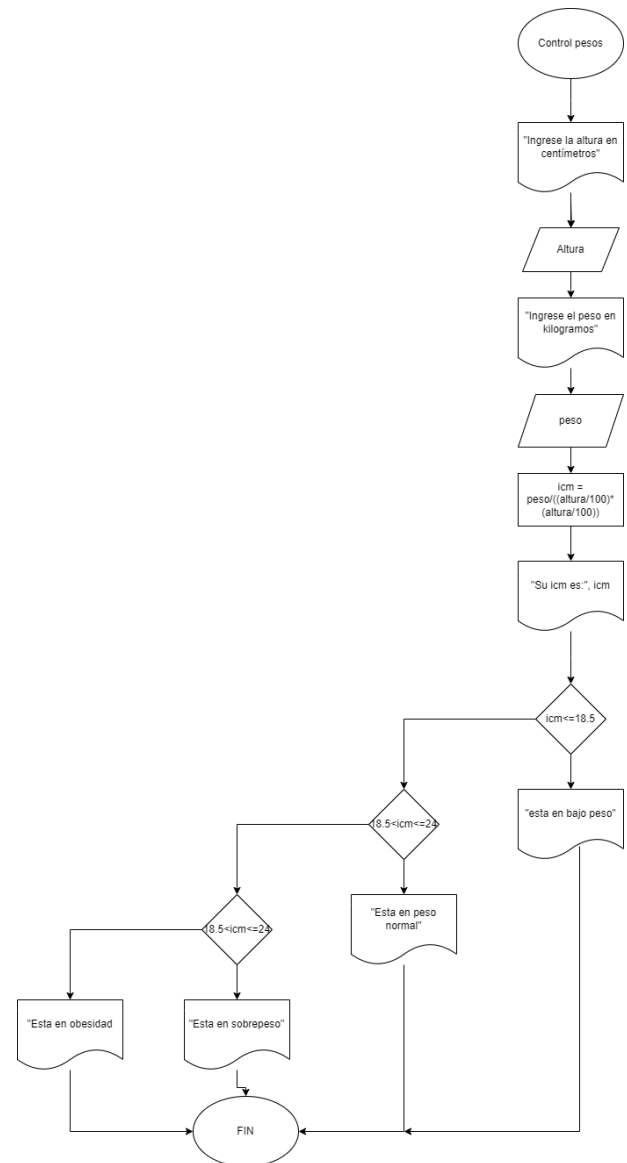


fig 7. diagrama de flujo del punto 4.

Al terminar de desarrollar los códigos que se solicitaron en sentencias y condicionales, continuamos con las actividades que se deben realizar en ciclos y arreglos, los cuales son:

1. Cree un programa a partir de bmi.c que solicita al usuario un número de personas y que posteriormente solicita la estatura y peso para cada una de ellas, calcular, almacenar e imprimir el BMI para cada una de las 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.

3. Vuelva a implementar los programas de los puntos 1 y 2 en Python sin usar Chat GPT .

Según lo que se solicita realizar en el primer punto se obtuvo el siguiente código:

```
1 #include <stdio.h>
2
3 int main() {
4     int persons;
5     printf("How many people: ");
6     scanf("%d", &persons);
7
8     float height[persons], weight[persons];
9     float bmi[persons];
10
11     for (int i = 0; i < persons; i++) {
12         printf("Enter the height for person %d (in cm): ", i + 1);
13         scanf("%f", &height[i]);
14
15         printf("Enter the weight for person %d (in kg): ", i + 1);
16         scanf("%f", &weight[i]);
17
18         bmi[i] = weight[i] / ((height[i] / 100) * (height[i] / 100));
19         printf("The BMI for person %d is %.2f\n", i + 1, bmi[i]);
20     }
21
22     return 0;
23 }
24
```

Fig 8. código que calcula el bmi de cada persona en lenguaje C.

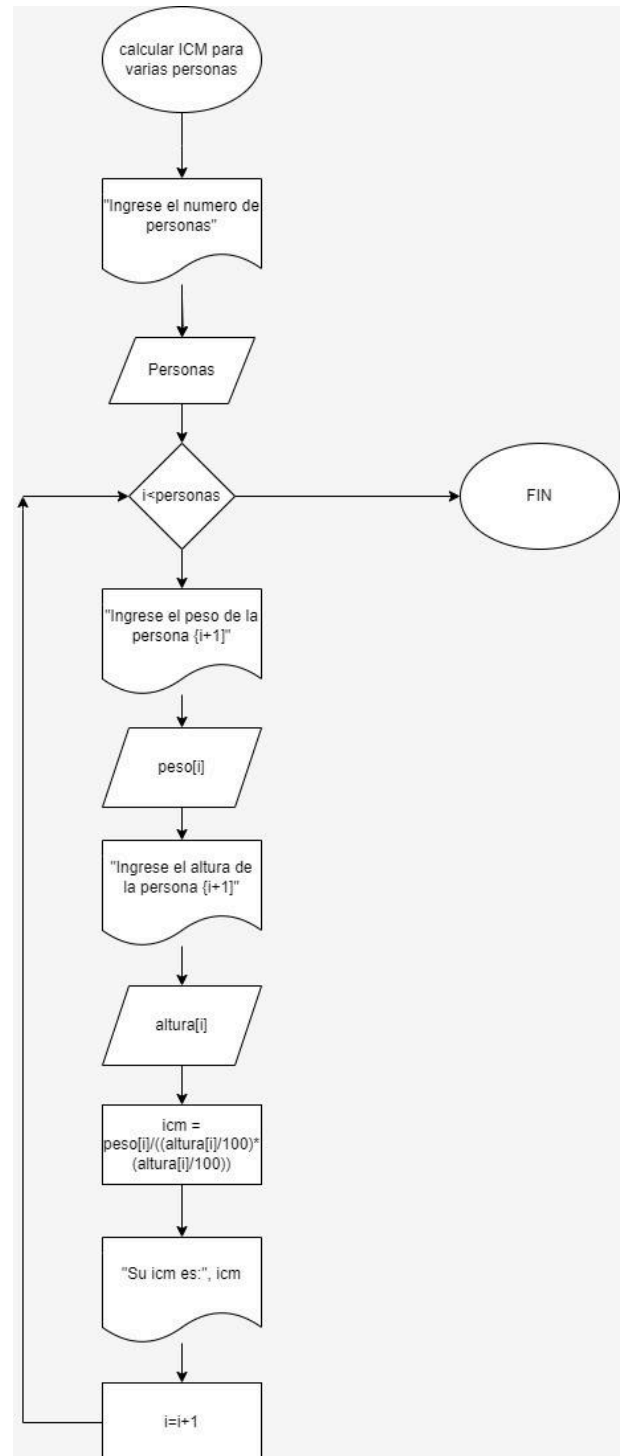


fig 9. diagrama de flujo

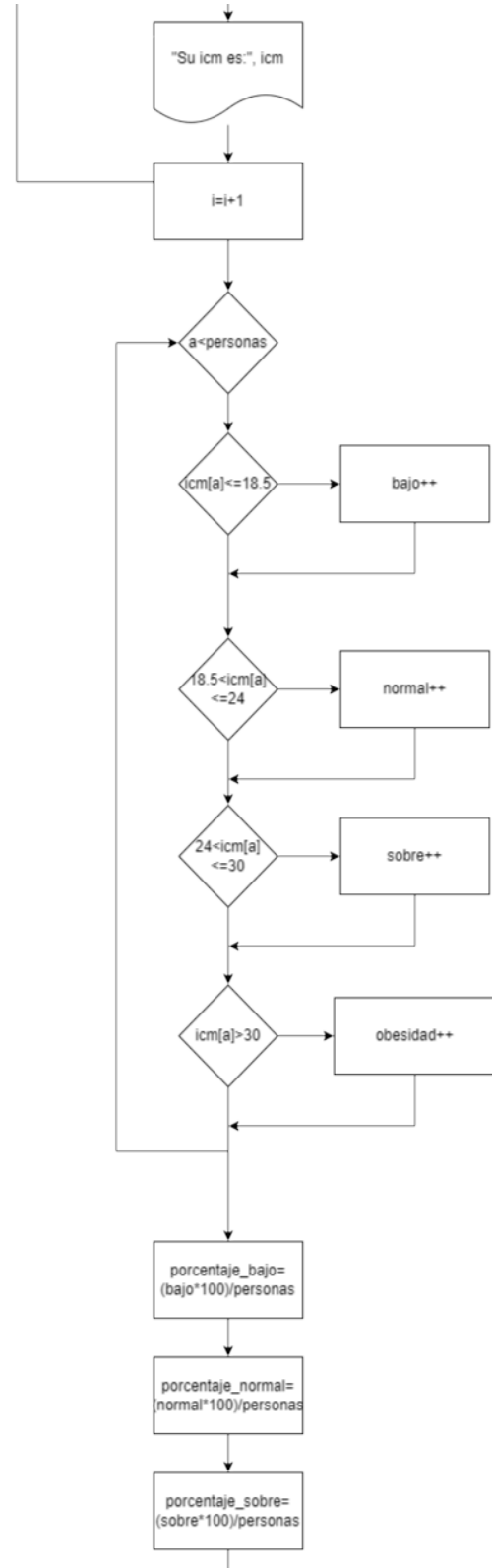
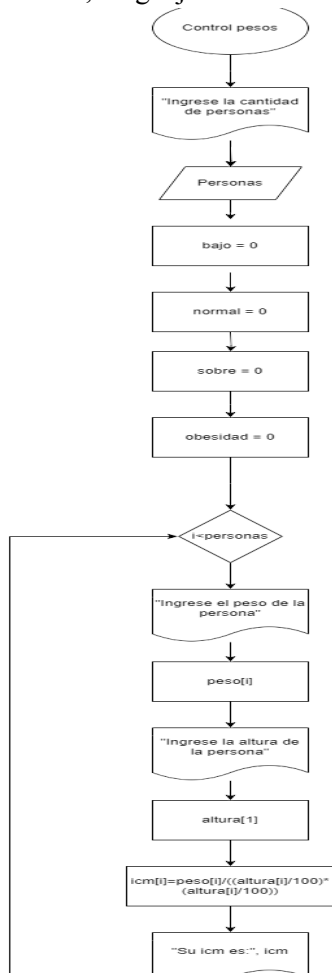
Lo que se solicitó realizar en el punto número 2 se realizó de la siguiente manera:

```

1 #include <stdio.h>
2 int main() {
3     float percentage_underweight, percentage_normal, percentage_overweight, p;
4     int people, underweight = 0, normal = 0, overweight = 0, obesity = 0;
5     printf("How many people: ");
6     scanf("%d", &people);
7     float bmi[people];
8     float height[people], weight[people];
9     for (int i = 0; i < people; i++) {
10         printf("Enter the height for person %d (in cm): ", i + 1);
11         scanf("%f", &height[i]);
12         printf("Enter the weight for person %d (in kg): ", i + 1);
13         scanf("%f", &weight[i]);
14         bmi[i] = weight[i] / ((height[i] / 100) * (height[i] / 100));
15         printf("The BMI for person %d is %.2f\n", i + 1, bmi[i]);
16     }
17     for (int a = 0; a < people; a++) {
18         if (bmi[a] <= 18.5) {
19             underweight++;
20         } else if (bmi[a] > 18.5 && bmi[a] <= 24) {
21             normal++;
22         } else if (bmi[a] > 24 && bmi[a] <= 30) {
23             overweight++;
24         } else if (bmi[a] > 30) {
25             obesity++;
26         }
27     }
28     percentage_underweight = (underweight * 100) / people;
29     percentage_normal = (normal * 100) / people;
30     percentage_overweight = (overweight * 100) / people;
31     percentage_obesity = (obesity * 100) / people;
32     printf("Percentage of people underweight: %.2f\n", percentage_underweight);
33     printf("Percentage of people with normal weight: %.2f\n", percentage_normal);
34     printf("Percentage of people overweight: %.2f\n", percentage_overweight);
35     printf("Percentage of people with obesity: %.2f\n", percentage_obesity);
36     return 0;
37 }

```

Fig 10. Código que calcula el porcentaje del bmi, lenguaje C



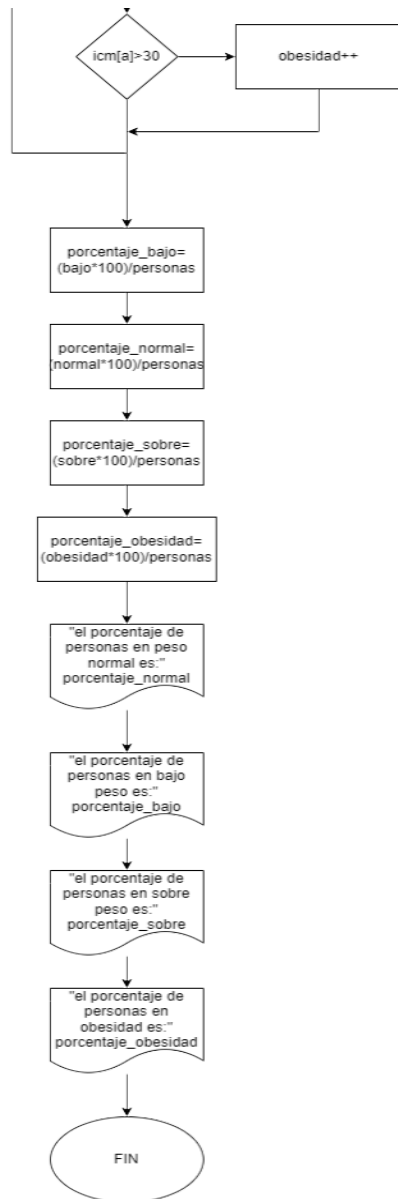


Fig 11. diagrama de flujo

Finalmente para el último punto se pide realizar las mismas actividades pero en lenguaje python, por lo tanto para el punto 1 que consiste en que se cree un programa a partir de bmi.c que solicita al usuario un número de personas y que posteriormente solicita la estatura y peso para cada una de ellas, calcular, almacenar e imprimir el BMI para cada una de las personas. Se obtuvo el siguiente código en python:

```

1 people = int(input("Enter the number of people: "))
2 list = []
3 weights = []
4 heights = []
5
6 for i in range(people):
7     weight = float(input(f"Enter the weight for person {i+1} (in kg): "))
8     height = float(input(f"Enter the height for person {i+1} (in cm): "))
9     weights.append(weight)
10    heights.append(height)
11
12    bmi = weight / ((height / 100) * (height / 100))
13    list.append(bmi)
14    print(f"The BMI for person {i+1} is: {bmi:.2f}")
15

```

Fig 12. código que calcula el bmi de cada persona en python

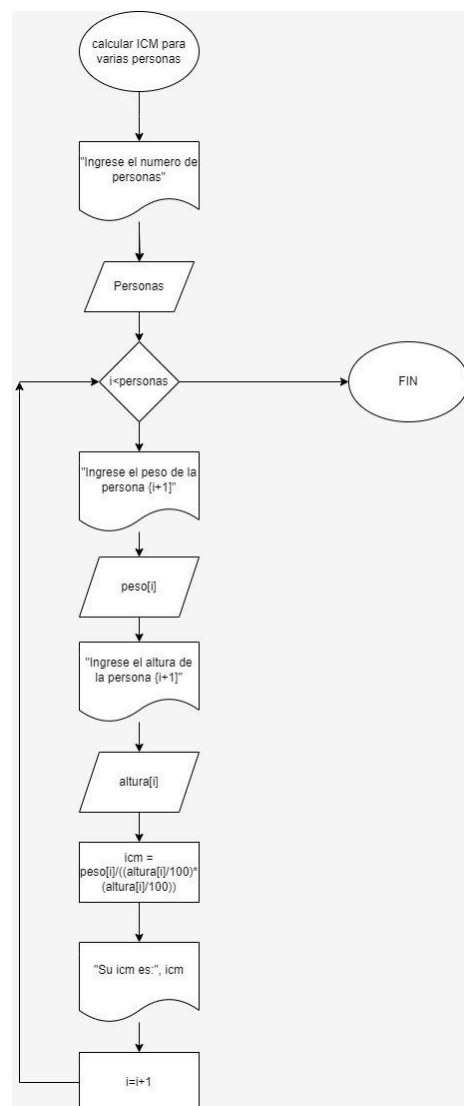


Fig.13 diagrama de flujo

A diferencia del código en lenguaje C que se encuentra en la fig 8, los arreglos son mucho más sencillos en python, el arreglo se deja vacío para indicar que no tiene un

tamaño, para agregar datos al arreglo, se pone el nombre de la lista y “append” y entre paréntesis el nombre de la variable de los datos que se quieren guardar en el arreglo.

Para la otra actividad del último punto se pide que 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. Se obtuvo lo siguiente:

```
1 people = int(input("Enter the number of people: "))
2 list = []
3 weights = []
4 heights = []
5 for i in range(people):
6     weight = float(input(f"Enter the weight for person {i+1}: "))
7     height = float(input(f"Enter the height for person {i+1}: "))
8     weights.append(weight)
9     heights.append(height)
10    bmi = weight / ((height / 100) * (height / 100))
11    list.append(bmi)
12    print(f"Your BMI for person {i+1} is: {bmi:.2f}")
13    underweight = normal = overweight = obesity = 0
14    for bmi in list:
15        if bmi < 18.5:
16            underweight += 1
17        elif 18.5 <= bmi < 24:
18            normal += 1
19        elif 24 <= bmi < 30:
20            overweight += 1
21        else:
22            obesity += 1
23    percentage_underweight = (underweight * 100) / people
24    percentage_normal = (normal * 100) / people
25    percentage_overweight = (overweight * 100) / people
26    percentage_obesity = (obesity * 100) / people
27    print(f"Percentage of people underweight: {percentage_underweight}")
28    print(f"Percentage of people with normal weight: {percentage_normal}")
29    print(f"Percentage of people overweight: {percentage_overweight}")
30    print(f"Percentage of people with obesity: {percentage_obesity}")
```

Fig 14. Código que calcula el porcentaje del bmi. En Python.

En este código al igual que el anterior se evidencia el cambio de los arreglos y el aumento, en lugar de ser “++” como en lenguaje C, en python es “+=1”.

Y por último se pide realizar por funciones un programa que 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 crea una función que realice el cálculo del peso ideal.

Respecto a lo anterior se obtuvo el siguiente resultado en lenguaje C:

```
1 #include <stdio.h>
2
3 // Function prototype for calcIdeal
4 float calcIdeal(float h);
5
6 // Function prototype for calcBmi
7 float calcBmi(float h, float w);
8
9 int main() {
10     // Local variable declaration
11     float height;
12     float weight;
13     float bmi;
14     float ideal;
15
16     // Ask user for height in centimeters
17     printf("Enter your height in centimeters: ");
18     scanf("%f", &height);
19
20     // Ask user for weight in kilograms
21     printf("Enter your weight in kilograms: ");
22     scanf("%f", &weight);
23
24     // Call calcBmi function
25     bmi = calcBmi(height, weight);
26
27     // Print result
28     printf("Your body mass index is %.2f kg/m^2\n", bmi);
29
30     return 0;
31 }
32
33 // Implementation of calcIdeal function
34 float calcIdeal(float h) {
35     float ideal;
36     ideal = 0.75 * h - 62.75;
37     return ideal;
38 }
39
40 // Implementation of calcBmi function
41 float calcBmi(float h, float w) {
42     float bmi, a;
43
44     // compute body mass index
45     bmi = w / ((h / 100.0) * (h / 100.0));
46
47     if (bmi <= 18.5) {
48         printf("is underweight\n");
49         a = calcIdeal(h);
50         printf("Your ideal weight is: %.2f\n", a);
51     } else if (24 < bmi && bmi <= 30) {
52         printf("is overweight\n");
53         a = calcIdeal(h);
54         printf("Your ideal weight is: %.2f\n", a);
55     } else if (18.5 < bmi && bmi <= 24) {
56         printf("is at normal weight\n");
57     } else if (bmi > 30) {
58         printf("is obese\n");
59         a = calcIdeal(h);
60         printf("Your ideal weight is: %.2f\n", a);
61     }
62
63     return bmi;
64 }
```

Fig 15. Código del peso ideal en lenguaje C.



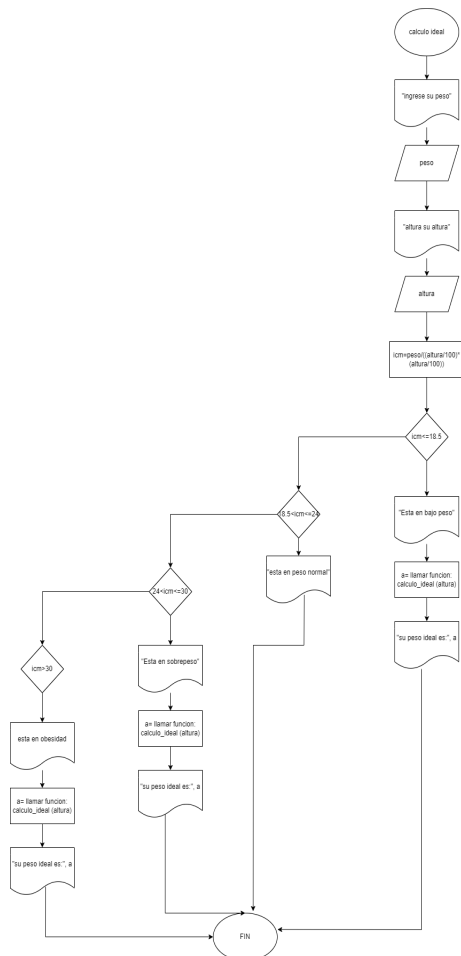


Fig. 16 diagrama de flujo

Lo que se realizó en este código fue agregar una función que calcule el peso ideal dada por la siguiente fórmula  $\text{Peso (Kg)} / \text{Estatura al cuadrado}$ . Luego de haber realizado la función se llama la función en el estado de peso en el que se encuentra la persona, en este caso, sobrepeso, bajo peso y obesidad.

```

1 def calc_ideal(altura):
2     ideal = 0.75 * altura - 62.75
3     return ideal
4
5 weight = float(input("Enter your weight (in kg): "))
6 height = float(input("Enter your height (in cm): "))
7 bmi = weight / ((height / 100) * (height / 100))
8 print(f"Your BMI is: {bmi:.2f}")
9
10 if bmi < 18.5:
11     print("You are underweight")
12     ideal_weight = calc_ideal(height)
13     print(f"Your ideal weight is: {ideal_weight:.2f}")
14 elif 18.5 <= bmi < 24:
15     print("You are at a normal weight")
16 elif 24 <= bmi < 30:
17     print("You are overweight")
18     ideal_weight = calc_ideal(height)
19     print(f"Your ideal weight is: {ideal_weight:.2f}")
20 else:
21     print("You are obese")
22     ideal_weight = calc_ideal(height)
23     print(f"Your ideal weight is: {ideal_weight:.2f}")
24
  
```

Fig.14 Código del cálculo ideal en lenguaje en python.

Este se realiza de la misma forma que en el lenguaje C, claramente con el lenguaje de código que utiliza python.

## CONCLUSIONES

- Se comprende el lenguaje en python y las diferencias con el lenguaje C.
- El lenguaje python es mucho más sencillo para programar
- El lenguaje python al no ser un lenguaje compilado es más rápido a la hora de ejecutar.