

# Lab. No 0. Lógica de programación (RA 1)

Palacios González Yojan  
[yojanspalaciosg@ugvirtual.edu.co](mailto:yojanspalaciosg@ugvirtual.edu.co)

**Resumen—** En esta práctica de laboratorio, se retomaron los conocimientos del lenguaje C vistos en cursos anteriores y se exploraron sus equivalentes en Python, comparando las similitudes y diferencias entre ambos lenguajes. Las actividades incluyeron la implementación de programas para calcular el índice de masa corporal (IMC), lo que permitió evaluar las condiciones de peso de una persona. Se compararon la lógica y la sintaxis entre ambos lenguajes, reforzando el uso de estructuras condicionales, ciclos y funciones.

**Palabras Clave:** Lenguaje C, Python, índice de masa corporal (IMC), sintaxis, similitudes, diferencias.

**Abstract—** In this laboratory practice, knowledge of the C language from previous courses was revisited and its equivalents in Python were explored, comparing the similarities and differences between both languages. Activities included implementing programs to calculate Body Mass Index (BMI), allowing the assessment of a person's weight conditions. The practice involved comparing the logic and syntax of both languages, reinforcing the use of conditional structures, loops, and functions.

**Keywords:** C language, Python, body mass index (BMI), syntax, similarities, differences.

```
// Include header file for prototypes of scanf and printf
#include <stdio.h>

// Prototype of calcBmi function
float calcBmi(float h,float w);

// Main function - entry point
int main(){
    // Local variable declaration
    float height;
    float weight;
    float bmi;

    // Ask user for height in cms
    printf("Enter your height in cms: ");
    scanf("%f",&height);

    // Ask user for weight in kgs
    printf("Enter your weight in kgs: ");
    scanf("%f",&weight);

    // Call calcBmi function
    bmi=calcBmi(height,weight);

    // Print result
    printf("Your body mass index is %f kg/m^2",bmi);
}

// Implementation of calcBmi function
float calcBmi(float h,float w){
    float bmi;

    // Compute body mass index
    bmi=w/((h/100.0)*(h/100.0));
    return bmi;
}
```

Figura 1: Código proporcionado para la primera actividad

## I. INTRODUCCIÓN

En esta primera práctica de laboratorio, se retomarán los conocimientos adquiridos en cursos previos, enfocándose en conceptos clave como funciones, ciclos y arreglos, y sentencias condicionales. El objetivo es entender la lógica del funcionamiento del lenguaje C y, posteriormente, implementar los mismos códigos en Python para comparar y comprender las diferencias y similitudes entre ambos lenguajes. Para esta práctica, se usará el IDE *Visual Studio Code*. Los códigos realizados se subirán a la plataforma *Github* junto a este informe.

## II. METODOLOGÍA

En este laboratorio se proponen tres actividades, cada una con varios ejercicios. La primera actividad se centra en las sentencias condicionales, tomando como base el código proporcionado en la figura 1, que calcula el índice de masa corporal, se desarrollarán diferentes ejercicios.

En el primer ejercicio, se pide compilar, ejecutar y realizar pruebas con diferentes valores de peso (en kg) y estatura (en cm), para los cuales se usarán los valores de la tabla 1.

Tabla 1

Valores de prueba usados para la práctica de laboratorio

	Valores de prueba		
	Prueba 1	Prueba 2	Prueba 3
Estatura	180	163	153
Peso	90	67	72

Con estos valores, se realizan las pruebas para el código proporcionado en las figuras 2, 3 y 4.

```
Enter your height in cms: 180
Enter your weight in kgs: 90
Your body mass index is 27.777779 kg/m^2
```

Figura 2: Ejecución del código con los valores de la prueba 1

```
Enter your height in cms: 163
Enter your weight in kgs: 67
Your body mass index is 25.217358 kg/m^2
```

Figura 3: Ejecución del código con los valores de la prueba 2

```
Enter your height in cms: 153
Enter your weight in kgs: 72
Your body mass index is 30.757401 kg/m^2
```

Figura 4: Ejecución del código con los valores de la prueba 3

En el siguiente ejercicio, se pide modificar el código proporcionado de la figura 1 para que le diga al usuario en qué categoría de peso se encuentra, según su IMC. En la modificación, solamente se le añadieron condiciones “if” y “else if” luego de imprimir el IMC del usuario, como se observa en la figura 5.

```
printf("Your body mass index is %f kg/m^2\n",bmi);

if(bmi<18.5){
    printf("You are underweight");
}
else if((bmi>=18.5)&&(bmi<=24.9)){
    printf("You are healthy");
}
else if((bmi>=25)&&(bmi<=29.9)){
    printf("You are overweight");
}
else if(bmi>=30){
    printf("You are obesity");
}
}
```

Figura 5: Clasificación de categoría de peso según el IMC

En la figura 6 se comprueba el funcionamiento del código modificado, usando los valores de la prueba 1 propuestos en la tabla 1.

```
Enter your height in cms: 180
Enter your weight in kgs: 90
Your body mass index is 27.777779 kg/m^2
You are overweight
```

Figura 6: Ejecución del código modificado para el ejercicio 2, con los valores de la prueba 1

Para el ejercicio número tres, se debe utilizar la IA ChatGPT para generar el equivalente en Python del código proporcionado en la figura 1. Luego, identificar las similitudes y diferencias entre ambas implementaciones. Véase figura 7.

```
# Define calculate_bmi function
def calculate_bmi(height_cm, weight_kg):
    return weight_kg / ((height_cm / 100) ** 2)

# Main function
def main():
    height = float(input("Enter your height in cm: "))
    weight = float(input("Enter your weight in kg: "))

    bmi = calculate_bmi(height, weight)
    print(f"Your body mass index is {bmi:.2f} kg/m^2")

# Execute main function
if __name__ == "__main__":
    main()
```

Figura 7: Implementación del código proporcionado de la figura 1 en Python hecho con IA

Las principales diferencias entre ambas implementaciones son:

- En Python se usa “#” para realizar comentarios. En C se usa “//”
- En Python las funciones deben ser declaradas antes del main principal, debido a que Python es un lenguaje interpretado, es decir, línea por línea. En C se pueden declarar antes o después del main principal

- En Python se organizan las instrucciones mediante la indentación. A diferencia de C que se usan llaves
- En Python no se usa “;”, a diferencia de C que sí es necesario
- En Python, para imprimir el valor de una variable, se debe poner entre llaves el nombre de la variable, seguido de “.” y el tipo de dato que se va a imprimir, por lo que no tienen que tener un orden de impresión. A diferencia de C que según el orden que se va a imprimir un dato, la variable debe ir en el mismo orden respectivamente.
- En Python no hay que llamar librerías para usar las funciones de leer y imprimir, se usa input para leer y print para imprimir. A diferencia de C que se debe llamar la librería para poder usar ambas funciones.

Las principales similitudes entre ambas implementaciones son:

- En ambas implementaciones, la lógica de asignación y uso de variables es equivalente
- La declaración y el llamado de funciones es muy similar, tienen la misma estructura y organización lógica

El cuarto ejercicio solicita re-implementar el código hecho en el ejercicio dos en Python sin usar la IA. En la figura 8 se visualiza el código propuesto.

```
"""This program is handed for people with age is over or equal at 20"""

# Function to calculate the Body Mass Index (BMI)
def calc_bmi(height, weight):
    bmi = weight / ((height / 100.0) ** 2)
    return bmi

# Main function - entry point
def main():
    # Ask user for height in cms
    height = float(input("Enter your height in cms: "))

    # Ask user for weight in kgs
    weight = float(input("Enter your weight in kgs: "))

    # Call calc_bmi function
    bmi = calc_bmi(height, weight)

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

    # Evaluate the BMI status
    if bmi < 18.5:
        print("You are underweight")
    elif 18.5 <= bmi <= 24.9:
        print("You are healthy")
    elif 25 <= bmi <= 29.9:
        print("You are overweight")
    elif bmi >= 30:
        print("You are obesity")

# Execute the main function
if __name__ == "__main__":
    main()
```

Figura 8: Re-implementación del código del punto 2 en python.

Como se observa en la figura 8, no son muchos los cambios en la estructura y organización del código, y la lógica es la misma que en C, logrando el mismo objetivo que el código de la figura 5 para el ejercicio dos. En la figura 9 se comprueba el funcionamiento, usando los valores de la prueba 2 propuestos en la tabla 1.

```

Enter your height in cms: 163
Enter your weight in kgs: 67
Your body mass index is 25.22 kg/m^2
You are overweight

```

**Figura 9:** Comprobación del funcionamiento del código de la figura 8 con los valores de prueba 2

Como se logra observar en la figura 9, el código implementado realiza correctamente las instrucciones y el cálculo de IMC.

A continuación, se realiza la actividad número dos, la cual se centra en los ciclos y arreglos.

El primer ejercicio pide crear un programa a partir del código de la figura 1, que solicite un número de personas, luego el peso y estatura de cada persona, y finalmente calcule e imprima el IMC de cada persona. El código propuesto se muestra a continuación.

```

/*This program is handed for people with age is over or equal at 20*/
// Include header file for prototypes of scanf and printf
#include <stdio.h>

// Prototype of calcBmi function
float calcBmi(float h,float w);

// Main function - entry point
int main(){
    // Local variable declaration
    int numPeople;
    float bmi;

    //Ask user for number of people
    printf("Enter the number of people: ");
    scanf("%d",&numPeople);

    //Declare array after asking the user for the number of people
    float height[numPeople];
    float weight[numPeople];

    //Loop for heights and weight of all people
    for(int i=0; i<numPeople; i++){
        // Ask user for height in cms
        printf("Enter the height in cms of person %d: ", i+1);
        scanf("%f",&height[i]);

        // Ask user for weight in kgs
        printf("Enter your weight in kgs: ");
        scanf("%f",&weight[i]);
    }

    //Loop to calculate and print bmi of each person
    for(int i=0; i<numPeople; i++){
        // Call calcBmi function
        bmi=calcBmi(height[i],weight[i]);
        // Print result
        printf("the body mass index of person %i is %f kg/m^2\n",i+1, bmi);
    }

    // Determine and print the BMI category for each person based on their BMI value
    if(bmi<18.5){
        printf("the person %i are underweight\n", i+1);
    }
    else if((bmi>=18.5)&&(bmi<=24.9)){
        printf("the person %i are healthy\n", i+1);
    }
    else if((bmi>=25)&&(bmi<=29.9)){
        printf("the person %i are overweight\n", i+1);
    }
    else if(bmi>=30){
        printf("the person %i are obesity\n", i+1);
    }
}

// Implementation of calcBmi function
float calcBmi(float h,float w){
    float bmi;

    // Compute body mass index
    bmi=w/((h/100.0)*(h/100.0));
    return bmi;
}

```

**Figura 10:** Código propuesto para el ejercicio 1 de la actividad dos

En la figura 11, se comprueba el funcionamiento del código de la figura 10. Se asigna un valor de tres personas, cada una de las personas tendrá los valores de prueba 1, 2 y 3 de la tabla 1 respectivamente.

```

Enter the number of people: 3
Enter the height in cms of person 1: 180
Enter your weight in kgs: 90
Enter the height in cms of person 2: 163
Enter your weight in kgs: 67
Enter the height in cms of person 3: 153
Enter your weight in kgs: 72
the body mass index of person 1 is 27.777779 kg/m^2
the person 1 are overweight
the body mass index of person 2 is 25.217358 kg/m^2
the person 2 are overweight
the body mass index of person 3 is 30.757401 kg/m^2
the person 3 are obesity

```

**Figura 11:** Comprobación del funcionamiento del código de la figura 10, con los valores de prueba de la tabla 1.

Para el ejercicio número dos, se solicita modificar el código de la figura 10 para que calcule e imprima el porcentaje de personas que se encuentra en cada categoría del IMC. Al código de la figura 10 se le añadieron las variables que se utilizarán en el porcentaje de cada categoría, como se puede observar en la figura 12. En la figura 13, se agregó un contador cada vez que se cumpliera una condición "if" o "else if"; finalmente, se muestra el porcentaje de personas que pertenecen a cada categoría del IMC.

```

int main(){
    // Local variable declaration
    int underWeight=0, healthy=0, overWeight=0, obesity=0;
}

```

**Figura 12:** Variables añadidas al código de la figura 10 para el ejercicio dos

```

if(bmi<18.5){
    printf("the person %i are underweight\n", i+1);
    underWeight+=1;
}
else if((bmi>=18.5)&&(bmi<=24.9)){
    printf("the person %i are healthy\n", i+1);
    healthy+=1;
}
else if((bmi>=25)&&(bmi<=29.9)){
    printf("the person %i are overweight\n", i+1);
    overWeight+=1;
}
else if(bmi>=30){
    printf("the person %i are obesity\n", i+1);
    obesity+=1;
}

//Print percent of every category from bmi
printf("The percentage of people with underweight is: %.2f\n", percentage(underWeight, numPeople));
printf("The percentage of people with healthy weight is %.2f\n", percentage(healthy, numPeople));
printf("The percentage of people with overweight is %.2f\n", percentage(overWeight, numPeople));
printf("The percentage of people with obesity is %.2f\n", percentage(obesity, numPeople));
}

```

**Figura 13:** Contadores agregados en las condiciones e impresiones de los porcentajes de cada categoría del IMC

En la figura 14 se observa la función creada para calcular el porcentaje.

```

float percentage(float i, float j)
{ float percentage;

    percentage=(i/j)*100;
    return percentage;
}

```

**Figura 14:** Función para calcular el porcentaje de personas en cada categoría del IMC

Para comprobar el funcionamiento del código propuesto para el ejercicio número dos, se utilizan los mismos valores del ejercicio uno de la figura 11. Véase el funcionamiento a continuación.

```

Enter the number of people: 3
Enter the height in cms of person 1: 180
Enter your weight in kgs: 90
Enter the height in cms of person 2: 163
Enter your weight in kgs: 67
Enter the height in cms of person 3: 153
Enter your weight in kgs: 72
the body mass index of person 1 is 27.777779 kg/m^2
the person 1 are overweight
the body mass index of person 2 is 25.217358 kg/m^2
the person 2 are overweight
the body mass index of person 3 is 30.757401 kg/m^2
the person 3 are obesity
The percentage of people with underweight is: 0.00
The percentage of people with healthy weight is 0.00
The percentage of people with overweight is 66.67
The percentage of people with obesity is 33.33

```

**Figura 15:** Comprobación del código del ejercicio dos, con los valores de prueba de la tabla 1.

El último ejercicio de la actividad dos, propone re-implementar los programas de los ejercicios 1 y 2 en python. En la figura 16 se observa el código del ejercicio 1 re-implementado en Python.

```

"""This program is handed for people with age is over or equal at 20"""

# Function to calculate the Body Mass Index (BMI)
def calc_bmi(height, weight):
    bmi = weight / ((height / 100.0) ** 2)
    return bmi

# Main function - entry point
def main():

    # Ask user for number of people
    numPeople = int(input("Enter the number of people: "))

    height=[None]*numPeople
    weight=[None]*numPeople

    for i in range (0,numPeople):

        # Ask user for height in cms
        height[i] = float(input(f"Enter the height in cms of person {i+1}: "))

        # Ask user for weight in kgs
        weight[i] = float(input(f"Enter the weight in kgs of person {i+1}: "))

    for i in range (0,numPeople):
        # Call calc_bmi function
        bmi = calc_bmi(height[i], weight[i])
        print(f"The body mass index of person {i+1} is {bmi:.2f} kg/m^2")

    # Evaluate the BMI status
    if bmi < 18.5:
        print(f"The person {i+1} are underweight")
    elif 18.5 <= bmi <= 24.9:
        print(f"The person {i+1} are healthy")
    elif 25 <= bmi <= 29.9:
        print(f"The person {i+1} are overweight")
    elif bmi >= 30:
        print(f"The person {i+1} are obesity")

# Execute the main function
main()

```

**Figura 16:** Re-implementación del código del ejercicio uno en Python

En la figura 17, se realiza la re-implementación del código del ejercicio 2 en Python. Para la comprobación del funcionamiento del código, se usaron los valores de estatura de la tabla 1, y para los valores de peso, se intercambian con la persona siguiente (por ejemplo, la persona uno con el peso de la persona dos). Véase figura 18.

```

"""This program is handed for people with age is over or equal at 20"""

# Function to calculate the Body Mass Index (BMI)
def calc_bmi(height, weight):
    bmi = weight / ((height / 100.0) ** 2)
    return bmi

#Function to calculate the percentage of every category from bmi
def calc_percentage(i, j):
    percentage = (i/j)*100
    return percentage

# Main function - entry point
def main():
    # Ask user for number of people
    numPeople = int(input("Enter the number of people: "))

    height=[None]*numPeople
    weight=[None]*numPeople

    # Initialize counters for each BMI category
    underWeight = 0
    healthy = 0
    overWeight = 0
    obesity = 0

    for i in range (0,numPeople):

        # Ask user for height in cms
        height[i] = float(input(f"Enter the height in cms of person {i+1}: "))

        # Ask user for weight in kgs
        weight[i] = float(input(f"Enter the weight in kgs of person {i+1}: "))

    for i in range (0,numPeople):
        # Call calc_bmi function
        bmi = calc_bmi(height[i], weight[i])
        print(f"The body mass index of person {i+1} is {bmi:.2f} kg/m^2")

        # Evaluate the BMI status
        if bmi < 18.5:
            print(f"The person {i+1} are underweight")
            underWeight = underWeight+1
        elif 18.5 <= bmi <= 24.9:
            print(f"The person {i+1} are healthy")
            healthy = healthy+1
        elif 25 <= bmi <= 29.9:
            print(f"The person {i+1} are overweight")
            overWeight = overWeight+1
        elif bmi >= 30:
            print(f"The person {i+1} are obesity")
            obesity = obesity+1

    #Call function percentage for every category
    underWeightPercentage = calc_percentage(underWeight, numPeople)
    healthyPercentage = calc_percentage(healthy, numPeople)
    overWeightPercentage = calc_percentage(overWeight, numPeople)
    obesityPercentage = calc_percentage(obesity, numPeople)

    #Print percent of every category from bmi
    print(f"The percentage of people with underweight is: {underWeightPercentage:.2f}%\n")
    print(f"The percentage of people with healthy weight is {healthyPercentage:.2f}%\n")
    print(f"The percentage of people with overweight is {overWeightPercentage:.2f}%\n")
    print(f"The percentage of people with obesity is {obesityPercentage:.2f}%\n")

# Execute the main function
main()

```

**Figura 17:** Re-implementación del código del ejercicio dos en Python

```

Enter the number of people: 3
Enter the height in cms of person 1: 180
Enter the weight in kgs of person 1: 67
Enter the height in cms of person 2: 163
Enter the weight in kgs of person 2: 72
Enter the height in cms of person 3: 153
Enter the weight in kgs of person 3: 90
The body mass index of person 1 is 20.68 kg/m^2
The person 1 are healthy
The body mass index of person 2 is 27.10 kg/m^2
The person 2 are overweight
The body mass index of person 3 is 38.45 kg/m^2
The person 3 are obesity
The percentage of people with underweight is: 0.00%
The percentage of people with healthy weight is 33.33%
The percentage of people with overweight is 33.33%
The percentage of people with obesity is 33.33%

```

**Figura 18:** Comprobación del código re-implementado del ejercicio dos con los valores de prueba de la tabla 1, intercambiando los pesos



Para la tercera actividad, la cual está enfocada en funciones, se deben de tener en cuenta los ejercicios dos y cuatro de la actividad 1, sobre sentencias condicionales. El ejercicio que se pide en esta actividad es realizar una modificación a ambos programas para que se le informe al usuario cuál es el peso ideal para entrar en la categoría de IMC saludable. Por lo tanto, hay que crear funciones para calcular el peso saludable.

En la figura 19, se muestra la función creada para el programa en C del ejercicio dos de la actividad uno, que calcula el peso en el que debe estar el usuario para estar saludable.

```
//Implementation of evaluateBmi
void evaluateBmi(float bmi, float height) {
    if (bmi < 18.5) {
        printf("You are underweight\n");
        printf("Your ideal weight should be between %.2f and %.2f kgs\n", calcMinWeight(height), calcMaxWeight(height));
    }
    else if ((bmi >= 18.5) && (bmi <= 24.9)) {
        printf("You are healthy\n");
    }
    else if ((bmi >= 25) && (bmi <= 29.9)) {
        printf("You are overweight\n");
        printf("Your ideal weight should be between %.2f and %.2f kgs\n", calcMinWeight(height), calcMaxWeight(height));
    }
    else if (bmi >= 30) {
        printf("You are obese\n");
        printf("Your ideal weight should be between %.2f kgs and %.2f kgs\n", calcMinWeight(height), calcMaxWeight(height));
    }
}
```

**Figura 19:** Función para determinar la categoría del IMC y el peso saludable de cada persona, en lenguaje C

En la figura 20, se muestra el código principal, para observar los cambios que se realizaron al reemplazar parte del código por funciones.

```
// Main function - entry point
int main(){
    // Local variable declaration
    float height;
    float weight;
    float bmi;

    // Ask user for height in cms
    printf("Enter your height in cms: ");
    scanf("%f",&height);

    // Ask user for weight in kgs
    printf("Enter your weight in kgs: ");
    scanf("%f",&weight);

    // Call calcBmi function
    bmi=calcBmi(height,weight);

    // Print result
    printf("Your body mass index is %f kg/m^2\n",bmi);

    //Call evaluateBmi function
    evaluateBmi(bmi, height);
}
```

**Figura 20:** Código principal modificado con las funciones creadas en lenguaje C

Para comprobar el funcionamiento del código con las funciones creadas, se usaron los valores de prueba de la prueba 3 de la tabla 1, como se observa en la figura 21

```
Enter your height in cms: 153
Enter your weight in kgs: 72
Your body mass index is 30.757401 kg/m^2
You are obesity
Your ideal weight should be between 43.31 kgs and 58.29 kgs
```

**Figura 21:** Comprobación del código modificado para el ejercicio uno de la actividad cuatro

En el siguiente ejercicio de la actividad cuatro, se modifica el código del ejercicio cuatro de la actividad dos. En la figura 22 se observa la función creada para evaluar el IMC del usuario, y el peso ideal para estar saludable.

```
#Function to evaluate the BMI status and print results
def evaluateBmi(bmi, height):
    minWeight = calcMinWeight(height)
    maxWeight = calcMaxWeight(height)

    if bmi < 18.5:
        print("You are underweight")
        print(f"Your ideal weight should be between (minWeight:.2f) kgs and (maxWeight:.2f) kgs")
    elif 18.5 <= bmi <= 24.9:
        print("You are healthy")
    elif 25 <= bmi <= 29.9:
        print("You are overweight")
        print(f"Your ideal weight should be between (minWeight:.2f) kgs and (maxWeight:.2f) kgs")
    elif bmi >= 30:
        print("You are obese")
        print(f"Your ideal weight should be between (minWeight:.2f) kgs and (maxWeight:.2f) kgs")
```

**Figura 22:** Función para determinar la categoría del IMC y el peso saludable de cada persona, en Python

Al crear la función para determinar la categoría del IMC y el peso saludable de cada persona, el código principal cambia un poco. En la figura 23 se observan los cambios realizados en el código principal.

```
# Main function - entry point
def main():
    # Ask user for height in cms
    height = float(input("Enter your height in cms: "))

    # Ask user for weight in kgs
    weight = float(input("Enter your weight in kgs: "))

    # Call calc_bmi function
    bmi = calc_bmi(height, weight)

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

    # Call evaluateBmi function
    evaluateBmi(bmi, height)
```

**Figura 23:** Código principal modificado con las funciones creadas en Python

En la figura 24 se comprueba el funcionamiento del código modificado con las funciones creadas. Se usaron los valores de prueba de la prueba 1 de la tabla 1.

```
Enter your height in cms: 180
Enter your weight in kgs: 90
Your body mass index is 27.78 kg/m^2
You are overweight
Your ideal weight should be between 59.94 kgs and 80.68 kgs
```

**Figura 24:** Comprobación del código modificado para el ejercicio dos de la actividad cuatro

### III. CONCLUSIONES

La lógica entre los dos lenguajes de programación usados en esta práctica de laboratorio es bastante similar, con algunos cambios en la sintaxis y la simplicidad del lenguaje, siendo Python el más destacado entre ambos por su sintaxis más clara y la simplicidad del lenguaje.

La práctica refuerza el uso de funciones y subraya la importancia de la modularización en los códigos. Tanto en C como en Python, la creación de funciones permitió organizar el código de una manera más sencilla, entendible y escalable, lo que a su vez mejora la legibilidad y facilita la reutilización y/o modificación del código.

El uso de IA para generar código facilita el entendimiento de los diferentes lenguajes de programación, ayudando a observar las diferencias y similitudes más notorias, lo que potencia la capacidad de resolución para las actividades propuestas a lo largo de la práctica.

Aunque la IA sea útil para generar código fácilmente, es crucial tener un buen dominio de los conceptos usados para comprender la lógica y el funcionamiento del código generado, y corregir errores o añadir instrucciones si es necesario.