

**CARRERA PROFESIONAL**

# **DESARROLLO DE SISTEMAS DE INFORMACIÓN FUNDAMENTOS DE PROGRAMACIÓN**

**Tema**

**ESTRUCTURAS DE CONTROL  
SECUENCIAL**

## CONCEPTOS BÁSICOS

Las estructuras de control secuencial son fundamentales en programación y nos permiten definir el flujo de ejecución de un programa, ejecutando una tras otra en orden predefinido, y ahora detallaremos ese flujo:

- **Secuencia:** La estructura de control más básica es la secuencia. En la secuencia, las instrucciones se ejecutan una tras otra, en el orden en que están escritas. Por ejemplo, en un programa, podrías tener una instrucción que pida al usuario que ingrese su nombre, seguida de otra que imprima un saludo usando ese nombre<sup>1</sup>.
  - **Secuencia Lineal:** La secuencia es la estructura de control más simple y básica en la programación. En esta estructura, las instrucciones se ejecutan una tras otra, en el orden en que están escritas en el código. Esto significa que la ejecución de una instrucción depende de que la instrucción anterior haya sido ejecutada.
  - **Orden de Ejecución:** Las instrucciones dentro de una secuencia se ejecutan en orden secuencial, comenzando desde la primera instrucción y avanzando hasta la última. Cada instrucción se ejecuta una vez que la anterior ha finalizado su ejecución.
  - **Flujo de Control Lineal:** El flujo de control en una secuencia es lineal, lo que significa que el programa sigue una trayectoria única y predecible a través de las instrucciones. Esta trayectoria lineal se determina por el orden en que están escritas las instrucciones en el código.
  - **Ejemplo Simple:** Aquí tienes un ejemplo simple en pseudocódigo que ilustra una secuencia de instrucciones para calcular el área de un rectángulo:

---

<sup>1</sup> <https://disenowebakus.net/estructuras-control.php>

1. Definir longitud como 10
2. Definir anchura como 5
3. Calcular área multiplicando longitud por anchura
4. Imprimir el resultado del cálculo del área

En este ejemplo, las instrucciones se ejecutan una tras otra en orden secuencial. Primero se define la longitud y la anchura del rectángulo, luego se calcula el área multiplicando la longitud por la anchura, y finalmente se imprime el resultado del cálculo.

- **Aplicaciones:** La estructura de control secuencial se utiliza ampliamente en la programación para llevar a cabo una variedad de tareas, desde operaciones matemáticas simples hasta tareas más complejas que requieren múltiples pasos. Esta estructura es fundamental en la construcción de programas y algoritmos, ya que proporciona una forma de especificar el orden en que deben ejecutarse las instrucciones<sup>2</sup>.
- **Bloques de código<sup>3</sup>:** En muchos lenguajes de programación, puedes agrupar un conjunto de instrucciones en un bloque de código. Esto permite que ese bloque se trate como una sola unidad dentro de la secuencia. Por ejemplo, podrías tener un bloque de código que realiza un cálculo específico o maneja una tarea particular.
  - **Agrupación de Instrucciones:** Los bloques de código permiten agrupar un conjunto de instrucciones relacionadas en una sola unidad. Esto hace que el

<sup>2</sup> <https://elpensante.com/estructuras-secuenciales/>

<sup>3</sup> <https://keepcoding.io/blog/que-es-estructura-secuencial-programacion/>

código sea más organizado y fácil de entender, ya que las instrucciones que realizan una tarea específica están agrupadas juntas.

- **Alcance (Scope):** Los bloques de código también determinan el alcance de las variables y otros identificadores en un programa. Las variables declaradas dentro de un bloque de código generalmente solo son accesibles dentro de ese bloque y no están disponibles fuera de él. Esto ayuda a evitar conflictos de nombres y a mantener la modularidad y la encapsulación en el código.
- **Estructura de Control:** Los bloques de código son utilizados en conjunción con estructuras de control como las instrucciones condicionales (if, else) y los bucles (for, while). Estas estructuras a menudo contienen bloques de código que se ejecutan solo si se cumple una condición específica o que se repiten varias veces.
- **Legibilidad y Mantenibilidad:** La utilización de bloques de código mejora la legibilidad y mantenibilidad del código. Al agrupar instrucciones relacionadas, facilita a los desarrolladores entender el propósito de cada bloque y realizar cambios de manera más segura, ya que pueden hacer modificaciones en una unidad lógica de código sin afectar otras partes del programa.
- **Ejemplo de Bloque de Código:** Aquí tienes un ejemplo simple que utiliza un bloque de código dentro de una estructura condicional:

if condicion:

```
# Inicio del bloque de código
```

```
    instrucion_1
```

```
    instrucion_2
```

# Fin del bloque de código

- En este ejemplo, las instrucciones instrucion\_1 y instrucion\_2 están dentro del bloque de código que se ejecuta si se cumple la condición especificada.

El bloque de código comienza después de los dos puntos que siguen a la condición y termina cuando la indentación regresa al nivel anterior

- **Control de flujo:** Aunque las estructuras de control secuencial siguen una secuencia lineal, es posible alterar ese flujo mediante estructuras de control de flujo, como las instrucciones condicionales y los bucles. Estas estructuras permiten tomar decisiones basadas en ciertas condiciones y repetir ciertas partes del código mientras se cumplen ciertas condiciones.
- **Instrucciones condicionales:** Las instrucciones condicionales permiten ejecutar ciertas partes del código solo si se cumple una condición específica. Por ejemplo, podrías tener una instrucción condicional que imprima un mensaje solo si el usuario ha ingresado un número mayor que cero.
- **Bucles:** Los bucles permiten repetir un bloque de código múltiples veces, mientras se cumpla una condición específica. Esto es útil cuando deseas realizar una tarea repetitiva sin tener que escribir el mismo código una y otra vez. Por ejemplo, podrías usar un bucle para imprimir los números del 1 al 10.

## Ejemplos

- Calcular el área de un triángulo

Inicio

```
// Definir las variables
```

```
Real base, altura, area
```

```
// Solicitar al usuario que ingrese la base del triángulo
```

Escribir "Ingrese la base del triángulo: "

Leer base

// Solicitar al usuario que ingrese la altura del triángulo

Escribir "Ingrese la altura del triángulo: "

Leer altura

// Calcular el área del triángulo

area = (base \* altura) / 2

// Mostrar el resultado

Escribir "El área del triángulo es: ", area

Fin

- Convertir grados Celsius a Fahrenheit

Inicio

// Definir las variables

Real celsius, fahrenheit

// Solicitar al usuario que ingrese la temperatura en grados Celsius

Escribir "Ingrese la temperatura en grados Celsius: "

Leer celsius

// Convertir Celsius a Fahrenheit

fahrenheit = (celsius \* 9/5) + 32

// Mostrar el resultado

Escribir "La temperatura en grados Fahrenheit es: ", fahrenheit

Fin

- Calcular el perimetro de un rectangulo

Inicio

// Definir las variables

Real longitud, anchura, perimetro

// Solicitar al usuario que ingrese la longitud del rectángulo

Escribir "Ingrese la longitud del rectángulo: "

Leer longitud

// Solicitar al usuario que ingrese la anchura del rectángulo

Escribir "Ingrese la anchura del rectángulo: "

Leer anchura

// Calcular el perímetro del rectángulo

perimetro = 2 \* (longitud + anchura)

// Mostrar el resultado

Escribir "El perímetro del rectángulo es: ", perimetro

Fin

### Ejercicios Propuestos

#### 1. Calcular el promedio de tres numeros

- a. Escribe un algoritmo que calcule el promedio de tres numeros dados .
- b. Solicitar al usuario que ingrese tres numeros
- c. Calcular el promedio de los tres numeros.
- d. Mostrar el resultado del calculo del promedio

#### 2. Calcular el área de un rectangulo

- a. Escribe un algoritmo que calcule el área de un rectángulo dados sus longitud y s anchura
- b. Solicitar al usuario que ingrese la ongitud del rectangulo
- c. Solicitar al usuario que ingrese la anchura del rectángulo
- d. Calcular el area del rectangulo
- e. Mostrar el resultado

