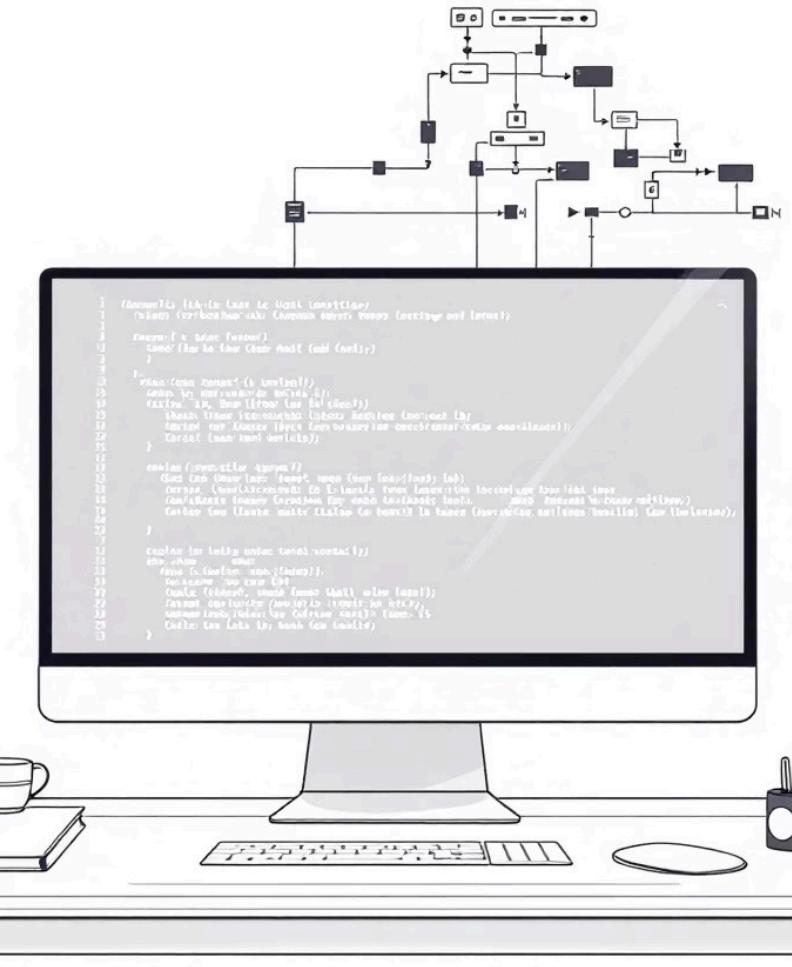


Introducción a la Programación (parte 1)

Este curso proporciona los fundamentos esenciales para comprender la programación desde sus conceptos más básicos. Aprenderás a estructurar el pensamiento lógico a través de algoritmos, diagramas de flujo y pseudocódigo, sentando las bases para cualquier lenguaje de programación que quieras aprender en el futuro.



¿Qué es un algoritmo?

Definición formal

Un **algoritmo** es una secuencia finita, ordenada y precisa de pasos que resuelven un problema.

Finitud

Tiene un número limitado de pasos

Precisión

No hay ambigüedad

Entrada

Datos necesarios para iniciar

Salida

Resultado del proceso

Eficiencia

Uso óptimo de recursos

Un algoritmo es como una **receta de cocina**: tiene ingredientes (entradas), pasos (procesos), y un plato final (salida).

Ejemplos de algoritmos cotidianos

- Hacer una tortilla
- Tomar la presión arterial
- Pasear al perro

Actividades

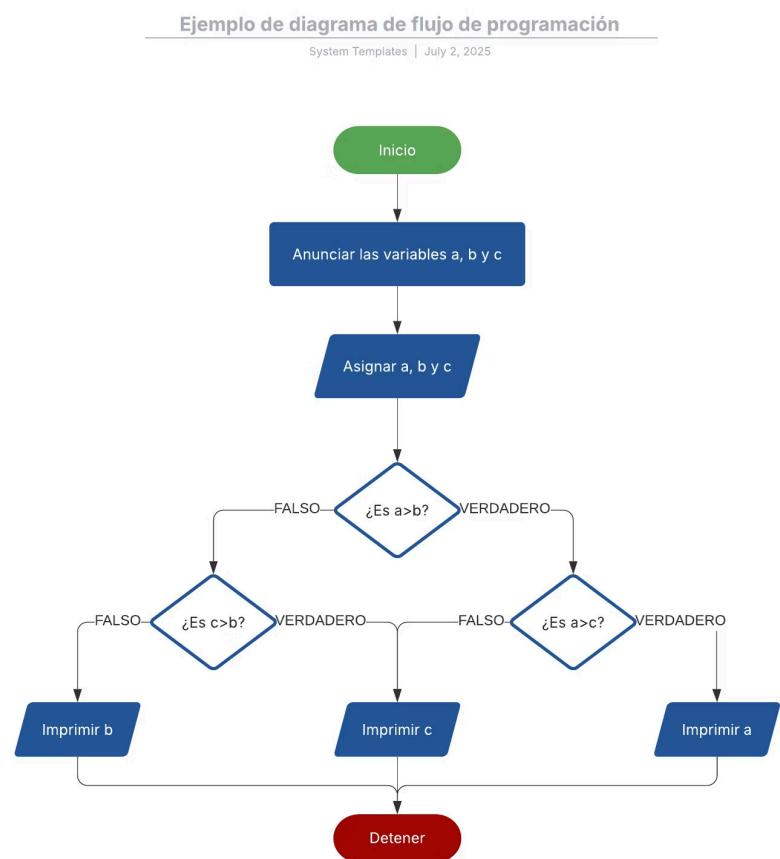
1. Escribe un algoritmo para preparar un bocadillo.
2. Escribe otro para lavarse los dientes y compáralo con el de otros compañeros.
3. ¿Cuál es más claro? ¿Qué pasos se omiten comúnmente?



Diagramas de flujo: símbolos, construcción y ejemplos

¿Qué es un diagrama de flujo?

Un **diagrama de flujo** es una forma gráfica de representar un algoritmo usando **símbolos estándar**.



Herramientas para crear diagramas

- <https://draw.io>
- <https://lucidchart.com>

Ejemplo: Hacer café

Inicio

↓

- ¿Hay café?
- → Sí → Preparar café
- → No → Comprar café

Actividades

1. Dibuja el diagrama de flujo para "hacer huevo frito".
2. Añade condicionales y bucles si es necesario.
3. Convierte el diagrama anterior en pseudocódigo.

Pseudocódigo: sintaxis, lógica y traducción desde diagramas

¿Qué es el pseudocódigo?

Es una forma textual de escribir algoritmos con estructura clara, sin seguir la sintaxis de un lenguaje real.

Ejemplo

Algoritmo CalcularIMC

 Leer peso

 Leer altura

 IMC ← peso / (altura * altura)

 Escribir IMC

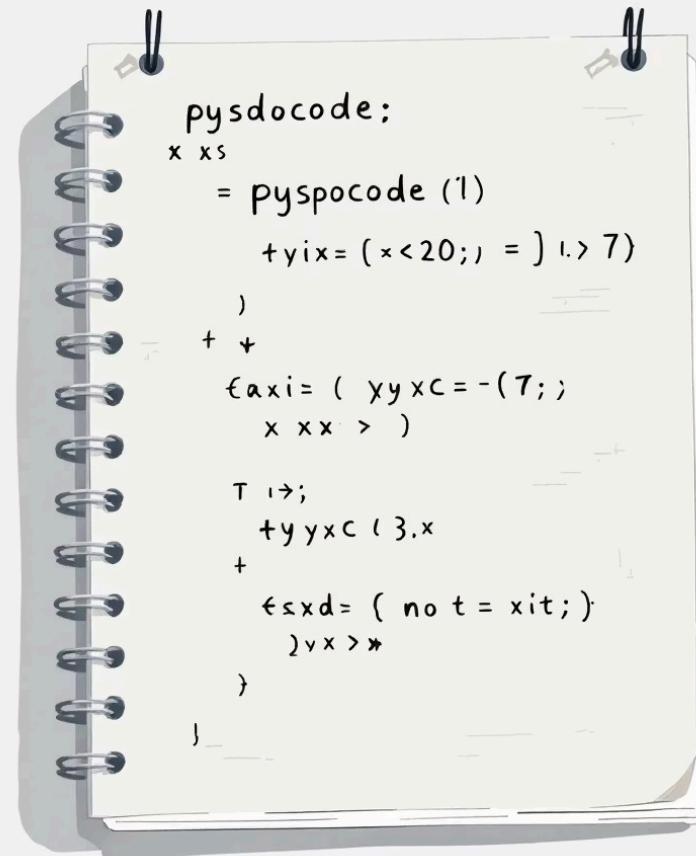
FinAlgoritmo

Estructura básica

- Inicio y fin claramente marcados
- Instrucciones en orden secuencial
- Uso de indentación para estructuras anidadas
- Símbolos de asignación (\leftarrow) para variables

Actividades

1. Convierte tu diagrama de flujo de "hacer té" en pseudocódigo.
2. Escribe pseudocódigo para saludar al usuario si es mayor de edad.



Ventajas del pseudocódigo

- Fácil de entender sin conocimientos técnicos
- Independiente del lenguaje de programación
- Permite centrarse en la lógica sin preocuparse por la sintaxis
- Facilita la transición a cualquier lenguaje real

Instrucciones básicas: asignación, condicionales, bucles

Asignación

Guarda un valor en una variable

```
x ← 5
```

Condicional

Ejecuta código según una condición

```
Si x > 0 entonces  
    Escribir "Positivo"  
FinSi
```

Bucle

Repite código mientras se cumpla una condición

```
Mientras x < 10 hacer  
    x ← x + 1  
FinMientras
```

Ejemplo (Python)

```
x = 5  
if x > 0:  
    print("Positivo")  
while x < 10:  
    x = x + 1  
    print(x)
```

Actividades

1. Implementa un bucle que imprima los números del 1 al 5.
2. Escribe un condicional que clasifique la edad de una persona.

Expresiones por categoría: aritméticas, lógicas y de texto

Aritméticas

Operaciones matemáticas con números

- $(3 + 5) * 2$
- x / y
- $a \% b$ (módulo o resto)

Lógicas

Evalúan a verdadero o falso

- $x > 3 \text{ Y } z < 7$
- $a = b \text{ O } c = d$
- NO ($x > 10$)

Texto

Manipulación de cadenas de caracteres

- "Hola" + " Mundo"
- nombre + " " + apellido
- texto.longitud()



Actividad

Actividades

1. Escribe tres expresiones de cada tipo.
2. Evalúa el resultado y explica el por qué.

Ejemplos de evaluación

- $5 + 3 * 2 = 11$ (no 16, porque la multiplicación tiene prioridad)
- $(x > 5) \text{ Y } (y < 10) = \text{verdadero}$ solo si ambas condiciones se cumplen
- "Hola" + " " + "Mundo" = "Hola Mundo" (concatenación)

Condicionales: if, else, switch/case

Tipos

- if y else
- elif (en Python)
- switch/case (en otros lenguajes como C o Java)

Ejemplo

```
nota = 6
if nota >= 5:
    print("Aprobado")
else:
    print("Suspendido")
```

Estructura switch/case (Java)

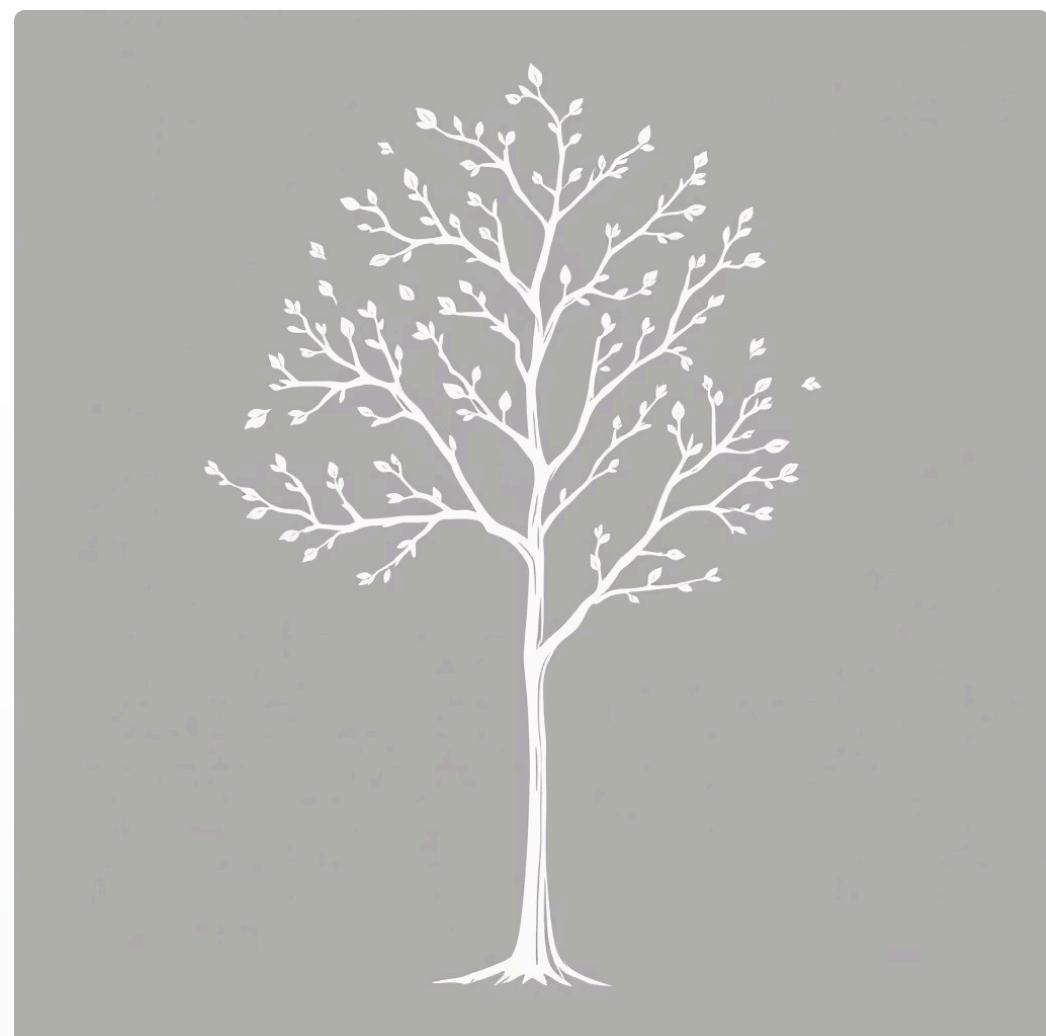
```
switch(dia) {
    case 1:
        System.out.println("Lunes");
        break;
    case 2:
        System.out.println("Martes");
        break;
    // ...
    default:
        System.out.println("Día inválido");
}
```

Condicionales anidados

```
if edad < 18:
    print("Menor de edad")
else:
    if edad < 65:
        print("Adulto")
    else:
        print("Jubilado")
```

Actividades

1. Crea un switch que devuelva el día de la semana según un número.
2. Escribe un algoritmo con múltiples condiciones encadenadas.



Tipos de bucles: while, do-while, for, do-until

Comparación

Bucle	Se evalúa antes	Usa contador	Se ejecuta siempre 1 vez
while	Sí	No	No
do-while	No	No	Sí
for	Sí	Sí	No
do-until	No	A veces	Sí

Ejemplos de bucles

While

```
i = 1  
while i <= 5:  
    print(i)  
    i = i + 1
```

For

```
for i in range(1, 6):  
    print(i)
```

Do-While (en pseudocódigo)

```
i ← 1  
Repetir  
    Escribir i  
    i ← i + 1  
Hasta Que i > 5
```

Actividades

1. Crea un bucle que imprima los números pares del 1 al 20.
2. Modifícalo para que se detenga si encuentra el 12.

Variables y tipos de datos

Tipos básicos

1

Entero

Números sin decimales: 1, -5, 42

Operaciones: suma, resta, multiplicación, división entera

2

Real

Números con decimales: 3.14, -0.5, 2.0

Operaciones: todas las matemáticas, incluyendo decimales

3

Carácter

Un solo símbolo: 'a', '5', '\$'

Operaciones: comparación, conversión

4

Cadena

Secuencia de caracteres: "Hola", "123abc"

Operaciones: concatenación, extracción, búsqueda

5

Booleano

Verdadero o falso

Operaciones: AND, OR, NOT

Ejemplos visuales



Actividad

Actividades

1. Declara una variable de cada tipo.
2. Realiza una operación válida con cada una.

Funciones, procedimientos y parámetros

¿Qué son?

- **Función:** bloque de código que **devuelve un valor**.
- **Procedimiento:** ejecuta acciones, **sin retornar valor**.
- **Rutina:** término general para ambos.

Ejemplo (Pseudocódigo)

```
Funcion Sumar(a, b)
    Retornar a + b
FinFuncion
```

Parámetros de entrada, salida y entrada/salida

Tipo	¿Qué hace?
Entrada	Se usa como dato de entrada, no se modifica
Salida	Devuelve un valor desde la función
Entrada/Salida	Se modifica dentro y fuera de la función

Actividad

Actividades

1. Escribe una función que reciba dos números y retorne el mayor.
2. Crea un procedimiento que muestre tu nombre tres veces.

Paso de parámetros: por valor vs por referencia

Tipo	¿Se modifica la variable original?	Uso común
Por valor		No Números simples
Por referencia		Sí Arreglos, objetos

Ejemplo

```
Funcion Sumar(PorValor a, PorValor b, PorReferencia resultado)
    resultado ← a + b
FinFuncion
```