

# PARADIGMAS IMPERATIVOS Y DECLARATIVOS.

ALFONSO  
CRUZ  
MENESES

# DEFINICIONES

Paradigma Imperativo: Se centra en describir cómo debe trabajar el programa paso a paso. Se basa en una secuencia de instrucciones que cambian el estado del sistema para alcanzar un resultado

Paradigma Declarativo: Se centra en describir qué resultado se desea obtener, sin especificar explícitamente los pasos o el flujo de control para llegar a él.



# CARACTERÍSTICAS

Característica	Paradigma Imperativo	Paradigma Declarativo
<b>Enfoque</b>	El proceso (Algoritmos y control).	El resultado (Lógica y propiedades).
<b>Estado</b>	Cambia mediante asignaciones y variables.	Es mayormente inmutable (sin efectos secundarios).
<b>Control de Flujo</b>	Bucles (for, while), condicionales y saltos.	Recursión, funciones de orden superior y filtros.
<b>Transparencia</b>	Baja (depende del estado global).	Alta (transparencia referencial).

# VENTAJAS Y LIMITACIONES

# Paradigma Imperativo

- Ventajas: \* Mayor control sobre el hardware y la memoria.
    - Más intuitivo para quienes piensan en secuencias lógicas paso a paso.
    - Altamente eficiente en rendimiento bruto.
  - Limitaciones: \* Código más propenso a errores (bugs de estado).
    - Más difícil de paralelizar debido a la mutabilidad.
    - Tiende a volverse complejo y difícil de mantener en proyectos grandes



# VENTAJAS Y LIMITACIONES



## Paradigma Declarativo

- Ventajas: \* Código más conciso, legible y fácil de razonar.
  - Facilita enormemente el procesamiento en paralelo y concurrente.
  - Menor probabilidad de errores laterales (side effects).
- Limitaciones: \* Suele tener un mayor consumo de recursos (memoria/CPU).
  - Curva de aprendizaje más pronunciada (especialmente funcional/lógica).
  - Puede ser difícil de optimizar para tareas de muy bajo nivel.

# APLICACIONES

Aplicaciones del Declarativo  
Dominante en bases de datos, inteligencia artificial, procesamiento de datos masivos y diseño de interfaces.

- Lenguaje 1: SQL (Solo dices qué datos quieres, no cómo buscarlos en el disco).
- Lenguaje 2: Haskell (Programación funcional pura basada en funciones matemáticas).

# APLICACIONES Y LENGUAJES

Aplicaciones del Imperativo  
Se usa principalmente en desarrollo de sistemas operativos, motores de videojuegos y aplicaciones donde el rendimiento es crítico.

- Lenguaje 1: C (El rey del control de bajo nivel).
- Lenguaje 2: Java (En su núcleo tradicional orientado a objetos).

- Lenguaje 1: C (El rey del control de bajo nivel).
- Lenguaje 2: Java (En su núcleo tradicional orientado a objetos).

# APLICACIONES Y LENGUAJES

## Paradigma Imperativo

- Ventajas:
  - \* Mayor control sobre el hardware y la memoria.
  - Más intuitivo para quienes piensan en secuencias lógicas paso a paso.
  - Altamente eficiente en rendimiento bruto.

- Limitaciones:
  - \* Código más propenso a errores (bugs de estado).
  - Más difícil de paralelizar debido a la mutabilidad.
  - Tiende a volverse complejo y difícil de mantener en proyectos grandes.
  -

# REFERENCIAS

GRACIAS POR  
SU ATENCIÓN