

# Intro a P00

## Contexto

---

- El software moderno es cada vez más complejo.
- Los programas requieren modelar entidades del mundo real.
- El paradigma estructurado resulta limitado al manejar grandes volúmenes de datos y relaciones.
- Necesitamos una forma de organizar el código de manera más cercana a cómo pensamos y entendemos el mundo.

## Paradigma estructurado (procedural)

---

- Basado en funciones y procedimientos que operan sobre datos.
- El flujo de control es difícil de seguir en sistemas grandes.
- Baja reutilización de código.
- Difícil mantenimiento y escalabilidad.
- Poca correspondencia entre el modelo del problema y la solución en código.

## Evolución del software

---

- En el mundo real interactuamos con "cosas" que tienen **estado** y **comportamiento**.
- Queremos representar entidades como: clientes, productos, autos, sensores, etc.
- La P00 propone una solución basada en **objetos**: unidades que encapsulan datos + lógica.
- Permite modelar sistemas complejos de forma más natural.

## Fundamentos

---

# ¿Qué es la Programación Orientada a Objetos?

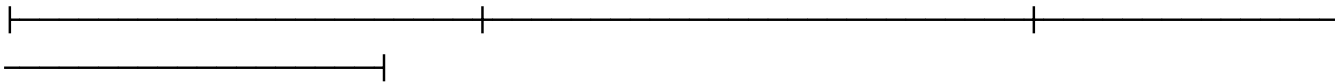
---

- Paradigma de programación basado en el concepto de **objetos**.
- Un objeto tiene:
  - **Atributos**: representan su estado.
  - **Métodos**: representan su comportamiento.
- Los objetos se crean a partir de **estructuras llamadas clases**.
- La idea es encapsular datos y lógica en una unidad coherente.

## Comparación con otros paradigmas

---

| | Estructurada | P00 |



| Organización | Funciones y datos separados | Entidades combinadas (objetos) |

| Reutilización | Limitada | Alta (herencia, composición) |

| Escalabilidad | Baja | Alta |

| Mantenimiento | Difícil | Más sencillo y modular |

## Ventajas

---

- Mejor organización del código
- Mayor modularidad y reutilización
- Facilita el mantenimiento y escalabilidad
- Más cercano al modelo mental humano

## Desventajas

---

- Mayor curva de aprendizaje
- Puede haber sobreingeniería

- No siempre es necesario para problemas simples

## Aplicaciones

---

- Desarrollo de software a gran escala
- Aplicaciones móviles (Android usa Java/Kotlin)
- Videojuegos y simulaciones
- Sistemas empresariales y backend
- Interfaces gráficas de usuario (GUI)
- Videojuego:
  - Jugador , Enemigo , Arma
- Sistema bancario:
  - Cuenta , Cliente , Transacción
- Aplicación de mensajería:
  - Usuario , Mensaje , Chat