



# Taller de Programación Orientada a Objetos (POO)

**Lenguajes:** Python, Java y C++

**Público:** Aprendices de nivel Técnico en Desarrollo de Software

---

## Preguntas del Taller

### Parte 1 — Investigación Teórica

1. ¿Qué es la Programación Orientada a Objetos (POO) y cuáles son sus ventajas frente a la programación estructurada?
  2. ¿Qué es una **clase**? ¿Qué es un **objeto**? Explica con tus palabras y da un ejemplo.
  3. Define los conceptos de **atributo**, **método** y **constructor** en POO.
  4. Explica qué significa **encapsulación**, **herencia**, **polimorfismo** y **abstracción**.
  5. ¿Qué es UML y para qué sirven los diagramas de clases, de objetos, de secuencia y de actividad?
  6. ¿Cuáles son las diferencias principales al implementar POO en **Python**, **Java** y **C++**?
- 

### Parte 2 — Diagramas UML

7. Elige un mini-proyecto (ejemplo: biblioteca, gestión de cursos o tienda online) y responde:
    - ¿Qué clases identificas en el problema?
    - ¿Qué atributos y métodos tendría cada clase?
    - ¿Qué relaciones existen entre ellas?
  8. Realiza un **diagrama de clases** para el problema elegido.
  9. Crea un **diagrama de objetos** con instancias concretas de tus clases.
  10. Dibuja un **diagrama de secuencia** que muestre cómo interactúan los objetos en un caso de uso (ej.: inscribir estudiante, prestar libro, hacer compra).
  11. Dibuja un **diagrama de actividad** con el flujo de pasos del caso de uso elegido.
-



### Parte 3 — Implementación en Python

12. Implementa en Python las clases de tu diagrama.
  13. Agrega atributos, métodos y una relación de herencia.
  14. Muestra un ejemplo de polimorfismo en tu código.
  15. Ejecuta tu programa y muestra la salida.
- 

### Parte 4 — Implementación en Java o C++

16. Implementa el mismo ejercicio en **Java** o **C++**.
  17. Explica cómo se definen las clases y métodos en el lenguaje que elegiste.
  18. Muestra cómo aplicaste herencia y polimorfismo en tu programa.
  19. Ejecuta tu código y muestra la salida.
- 

### Parte 5 — Reflexión Final

20. ¿Qué similitudes y diferencias encontraste entre Python, Java y C++ al trabajar con POO?
  21. ¿Qué dificultades tuviste al diseñar los diagramas UML?
  22. ¿Qué parte del taller consideras más importante para tu formación como técnico en desarrollo de software?
  23. Propón una mejora o extensión para tu ejercicio (ej.: agregar persistencia, interfaz gráfica o más clases).
- 

### Entregables

- Respuestas escritas a las preguntas.
- Diagramas UML en imágenes o PDF.
- Código fuente en Python.
- Código fuente en Java o C++.
- Breve explicación del ejercicio en clase y en video (3–5 min).