UNIVERSIDAD DE LIMA

# INGENIERÍA DE SOFTWARE II

## Laboratorio N°01: POO

En este taller desarrollaremos un sistema básico de compras en línea usando los principios de Programación Orientada a Objetos (POO) en Java. El objetivo es aplicar clases, objetos, herencia, interfaces y manejo de listas, a través de un flujo de desarrollo progresivo que simula un caso real.

### Ejercicio 01

Se le ha encargado crear una clase **Producto** con información relevante y mostrar detalles de varios productos.

Tareas:

1. Crear una clase llamada Producto con los atributos:
    a. nombre (String)
    b. precio (double)
    c. stock (int)
2. Crear un método mostrarInfo() que imprima los datos del producto
3. Desde el **main**, crear al menos 2 objetos de tipo Producto y mostrar su información.
4. Validar que el precio y el stock no puedan ser negativos (puedes lanzar una excepción o usar un condicional).

### Ejercicio 02:

Ahora se le ha solicitado extender la funcionalidad para crear un sistema de carrito de compras que permita manejar diferentes tipos de productos, utilizando herencia y polimorfismo.

Tareas:

1. Convierte la clase **Producto** en una clase abstracta con sus atributos y método abstracto mostrarInfo().
2. Crea al menos dos subclases que hereden de Producto:
    a. **ProductoFisico**: con un atributo adicional peso (double).
    b. **ProductoDigital**: con un atributo adicional url Descarga (String)

3. Implementa el método mostrarInfo() en cada subclase, mostrando los datos correspondientes al tipo de producto.
4. Crea la clase **Carrito**, que contenga una lista de productos (ArrayList<>), e implementa:
    a. agregarProducto(Producto producto): debe verificar que haya stock y descontarlo si es posible.
    b. mostrarResumen(): debe listar todos los productos agregados y mostrar el total a pagar.
5. En el main, crea una lista mixta de productos físicos y digitales, permite al usuario seleccionar cuáles agregar al carrito, y muestra el resumen de la compra.

**Ejercicio 03:**

Para finalizar el flujo, ahora se le solicita simular el proceso de pago mediante el uso de una interfaz Pago, que permita representar distintas formas de pago.

Tareas:

1. Crear una interfaz **Pago** con el método:
    a. realizarPago(double monto): boolean
2. Crear una clase abstracta **MetodoPago** que implemente Pago y contenga un atributo monto.
3. Crear dos clases concretas:
    a. **PagoConTarjeta**: verifica si el número de tarjeta tiene 16 dígitos.
    b. **PagoConTransferencia**: simula una verificación simple (por ejemplo, si se ingresó un código de confirmación).
4. Desde el main, permitir al usuario seleccionar el método de pago y mostrar si fue exitoso o no.

Archivos sugeridos:

- Pago.java (interfaz)
- MetodoPago.java (abstracta)
- PagoConTarjeta.java, PagoConTransferencia.java
- Main.java (flujo de carrito + pago)