

Manejo de Excepciones y Pruebas Unitarias

Contexto (base)

Se reutiliza el modelo del TP2/TP3 (Usuario, CarritoDeCompras, ItemCarrito, Producto, Pedido, DetallePedido, EstadoPedido y jerarquía Empleado/Administrador/Vendedor)

Excepciones personalizadas

Definir al menos 5 de estas clases

1. CantidadInvalidaException
 - Se lanza al agregar ítems con cantidad ≤ 0 en `CarritoDeCompras.agregar(...)` o `Usuario.agregarAlCarrito(...)`.
2. ProductoInactivoException
 - Al intentar agregar al carrito un `Producto` con `activo == false`.
3. ItemNoEncontradoException
 - Al intentar `remove(p)` en el carrito y no existir el ítem correspondiente.
4. CheckoutCarritoVacioException
 - En `Usuario.checkout()` o `GestorPedidos.crearDesdeCarrito(...)` si el carrito no tiene ítems.
5. EstadoInvalidoException
 - En `Pedido.confirmar()` si la transición de estado no es **CREADO** → **CONFIRMADO**.
6. PrecioInvalidoException
 - En `GestorProductos.modificarPrecio(sku, nuevoPrecio)` si `nuevoPrecio  $\leq 0$` .
7. UsuarioNoEncontradoException / PedidoNoEncontradoException / ProductoNoEncontradoException
 - En búsquedas de gestores cuando el elemento no existe y la operación **no** puede continuar sin ese dato.

Pueden agregar otras si su diseño lo requiere

Consignas de implementación

1. Agregar las excepciones listadas y documentar con JavaDoc cada una (cuándo y dónde se lanza).
2. Actualizar métodos del modelo y gestores para que validen reglas y **lancen/propaguen** las excepciones.
3. Mensajes de error: proveer message significativo (ej.: "No se puede confirmar un pedido en estado ENVIADO").
4. Implementar las pruebas unitarias correspondientes