

Prueba Técnica – SemiSenior Avanzado / Pre-Senior

Esta prueba técnica está diseñada para evaluar un perfil de **SemiSenior avanzado** con responsabilidades de **Senior**. El objetivo es medir no solo la capacidad de escribir código funcional, sino también de modelar reglas de negocio, tomar decisiones de arquitectura y mantener buenas prácticas de testing y consistencia.

Contexto

La empresa logística **FastTrack Logistics** necesita un sistema para gestionar **Pedidos de Compra (Purchase Orders)** y su **Inventario**. El sistema debe permitir registrar pedidos, recibirlos, y reflejar automáticamente los cambios en inventario.

Requerimientos

- Modelar las entidades principales: Kiosk, Product, PurchaseOrder, OrderDetail, InventoryMovement.
- Aplicar principios de DDD (constructores protegidos, métodos estáticos Create, validaciones básicas).
- Implementar repositorio genérico (IRepository) o utilizar ORM (EF Core / Sequelize).
- Garantizar que al eliminar un Kiosk, sus órdenes se eliminen en cascada.
- Implementar el servicio de dominio ReceivePurchaseOrder: marcar la orden como recibida, generar movimientos de inventario, manejar transacciones (todo o nada).
- Reglas de negocio: una orden no puede recibirse dos veces, no se permiten cantidades negativas, inventario siempre consistente.
- Tests unitarios: creación de orden válida, rechazo de orden duplicada, actualización de inventario correcta.

Extras (para destacar Seniority)

- Explicar decisiones de arquitectura (ej: por qué usar repositorios, VO para ProductName o Quantity).
- Proponer una mejora escalable (ej: eventos de dominio PurchaseOrderReceived para integrarse con facturación).
- Justificar trade-offs (ej: ON DELETE CASCADE vs. NO ACTION).
- Diseñar un endpoint REST o GraphQL para exponer la operación ReceivePurchaseOrder.
- Sugerir un pipeline de CI/CD básico para este proyecto.

Criterios de Evaluación

- Correcto modelado de dominio y relaciones entre entidades.
- Aplicación de buenas prácticas de diseño y patrones de arquitectura.
- Calidad de los tests (cobertura de casos normales y edge cases).
- Capacidad para justificar decisiones técnicas.
- Claridad del código y documentación entregada.

Entregables

- Código fuente del proyecto (repositorio o zip).
- Instrucciones claras para ejecutar el proyecto (README).
- Scripts o migraciones de base de datos (si aplica).
- Tests implementados y ejecutables.
- Documento breve (o comentarios) explicando decisiones de diseño y mejoras propuestas.

Tiempo Estimado: 4–6 horas.

Nota: Se valorará la calidad y claridad por sobre la cantidad de funcionalidades.