Práctica 2. Sistema de Notificaciones para DHL

DHL es una plataforma de entregas a domicilio que necesita modernizar su sistema de notificaciones. Actualmente, el sistema está acoplado y difícil de mantener. Debes aplicar patrones de diseño con interfaces.

Situación Actual

La empresa envía notificaciones a sus usuarios (clientes y repartidores) por diferentes canales: SMS, Email, y notificaciones Push. Además, necesitan:

- Agregar funcionalidades adicionales a las notificaciones (como cifrado, registro de auditoría, límite de tasa)
- Cambiar fácilmente entre diferentes proveedores de servicios
- Poder combinar múltiples canales para notificaciones importantes

Requerimientos del Sistema

Inyección de Dependencias (40%)

Debes implementar un sistema base de notificaciones donde:

1. Crear la interfaz base ServicioNotificacion con métodos:
    - enviar(String destinatario, String mensaje)
    - obtenerEstado() - retorna el nombre del servicio

2. Implementar al menos 3 servicios concretos:
    - NotificacionSMS - simula envío por SMS
    - NotificacionEmail - simula envío por correo
    - NotificacionPush - simula notificaciones push móviles

3. Crear la clase GestorPedidos que:
    - Debe recibir un ServicioNotificacion mediante inyección de dependencias (constructor)
    - Tiene métodos como:
        - confirmarPedido(String cliente, String detalles)
        - asignarRepartidor(String repartidor, String direccion)

        - notificarEntrega(String cliente)
    - Cada método debe usar el servicio de notificación inyectado
4. Demostrar la flexibilidad del sistema creando diferentes instancias de GestorPedidos con distintos servicios de notificación.

Patrón Decorador (60%)

Debes extender el sistema para agregar funcionalidades adicionales a las notificaciones de forma dinámica:

1. Crear decoradores concretos que implementen ServicioNotificacion:

    - NotificacionCifrada: Cifra el mensaje antes de enviarlo (puede usar un cifrado simple como invertir el texto u otro (elección libre))

    - NotificacionConLog: Registra en consola la fecha, hora, destinatario y un resumen antes de enviar

    - NotificacionConReintentos: Si el envío "falla" (simular aleatoriamente), reintenta hasta 3 veces

    - NotificacionPrioritaria: Agrega un prefijo "[URGENTE]" al mensaje y registra como prioritaria

2. Permitir combinaciones:

    - Los decoradores deben poder anidarse (Ejemplo: SMS con log + cifrado + reintentos)

3. Implementar un NotificadorMulticanal que:

    - También implemente ServicioNotificacion
    - Internamente tenga una lista de servicios
    - Envíe la notificación por todos los canales configurados

Casos de Prueba Esperados

Deben demostrar en su Main:

1. Envío simple por SMS
2. Envío simple por Email
3. SMS con cifrado
4. Email con log y cifrado
5. Push con reintentos y log
6. Notificación prioritaria por email con todos los decoradores
7. Multicanal (SMS + Email + Push) con log
8. Cambio dinámico de servicio en GestorPedidos

Entregables

1. Código fuente completo con todas las clases e interfaces
2. Diagrama UML mostrando las relaciones entre clases
3. Documento breve (1-2 páginas) explicando:
    - Cómo implementaron la inyección de dependencias
    - Cómo los decoradores agregan funcionalidad
    - Ventajas del diseño implementado
    - Posibles extensiones futuras