

**Objetivo:** Desarrollar una API para un servicio de E-Commerce que permita gestionar y verificar las siguientes funcionalidades:

1. **Verificación de stock de productos**
2. **Validación de cupones de descuento**
3. **Beneficios para usuarios premium**
4. **Cálculo de costos de envío**
5. **Estimación de fecha de entrega**

## **Requisitos**

### **1. Endpoints requeridos:**

- **GET /stock/{productId}**
  - **Descripción:** Este endpoint debe recibir un productId y verificar si hay suficiente stock del producto.
  - **Reglas:**
    - Si el stock es mayor a 0, retornar true.
    - Si no hay stock disponible, retornar false.
    - Considerar diferentes almacenes y consolidar el stock.
- **POST /apply-coupon**
  - **Descripción:** Este endpoint debe recibir un couponCode y verificar si el cupón es válido y aplicable a la compra.
  - **Reglas:**
    - Verificar si el cupón está vigente y no ha expirado.
    - Validar si el cupón es aplicable al carrito de compra (productos específicos o categorías).
    - Si el cupón es válido, devolver el porcentaje o monto de descuento.
- **GET /premium-benefits/{userId}**
  - **Descripción:** Verificar si el usuario identificado por userId tiene suscripción premium y cuáles son sus beneficios.
  - **Reglas:**
    - Si el usuario es premium, aplicar beneficios como envíos gratuitos o descuentos adicionales.
    - Devolver los beneficios específicos del usuario.

- **POST /calculate-shipping**

- **Descripción:** Este endpoint debe recibir una dirección de destino y calcular el costo de envío.
- **Reglas:**
  - Tener en cuenta la distancia desde el almacén más cercano.
  - Considerar tarifas según peso, tamaño de los productos y tipo de envío (rápido o regular).
  - Si el usuario es premium, aplicar descuentos en el envío o envío gratuito.

- **GET /estimate-delivery/{orderId}**

- **Descripción:** Estimar la fecha de entrega de un pedido basado en la ubicación del usuario y el método de envío seleccionado.
- **Reglas:**
  - Estimar fecha de entrega según la distancia entre el almacén y la dirección del cliente.
  - Considerar tiempos adicionales para envíos internacionales o en zonas remotas.
  - Mostrar un rango de fechas de entrega (mínimo-máximo).

## Instrucciones

1. Implementa estos endpoints utilizando **C# .NET**. Puedes usar cualquier framework .NET o NET Core
2. Asegúrate de seguir principios SOLID y buenas prácticas de desarrollo.
3. Usa **Dapper** para el acceso a datos.
4. Incluye en el proyecto un archivo README.md que explique cómo ejecutar la API y cómo probar cada uno de los endpoints.
5. Los datos pueden ser simulados (mock) o almacenados en una base de datos en memoria.
6. Se valorará la correcta implementación de la inyección de dependencias, el manejo adecuado de errores y la claridad del código.

## Extras (Puntos adicionales):

- Si tienes tiempo, implementa pruebas unitarias para alguno de los endpoints.
- Considera la posibilidad de implementar validaciones adicionales para garantizar la seguridad y la integridad de los datos (por ejemplo, validación de JWT para autenticación de usuarios).