

Desarrollador Backend .NET

Desafío Técnico

Sistema de Notificaciones de Inventario

Contexto

Desarrollar un sistema simple para la gestión de actualizaciones de inventario entre dos microservicios, usando RabbitMQ como middleware de mensajería.

Requerimientos

- API de Inventario (Productor):**
 - Crear un API REST en .NET Core para gestionar productos de inventario
 - Endpoints requeridos:
 - GET `/api/products` - Listar todos los productos
 - GET `/api/products/{id}` - Obtener un producto por ID
 - POST `/api/products` - Crear un nuevo producto
 - PUT `/api/products/{id}` - Actualizar un producto existente
 - DELETE `/api/products/{id}` - Eliminar un producto
 - Modelo de Producto debe incluir: ID, Nombre, Descripción, Precio, Stock, Categoría
- Servicio de Notificaciones (Consumidor):**
 - Crear un segundo servicio que reciba mensajes de actualización de inventario
 - Implementar un consumidor de RabbitMQ que procese los mensajes
 - Guardar un registro de las actualizaciones en una base de datos local
- Integración con RabbitMQ:**
 - Configurar un exchange tipo 'direct' llamado 'inventory_exchange'
 - Configurar colas para diferentes tipos de eventos (creación, actualización, eliminación)
 - Implementar un publisher en el API de Inventario
 - Implementar un consumer en el Servicio de Notificaciones
 - Implementar manejo básico de errores y reintentos
- Resiliencia:**
 - Implementar un patrón Circuit Breaker para la comunicación con RabbitMQ
 - Manejar escenarios de error y garantizar que no se pierdan mensajes

Entregables

1. Código fuente de ambos servicios en repositorios Git
2. Archivo README con instrucciones para ejecutar la solución

3. Diagrama simple de la arquitectura implementada
4. Documentación de la API (puede ser Swagger)
5. Script Docker Compose para levantar todo el ambiente (incluyendo RabbitMQ)

Tecnologías requeridas

- .NET Core 6.0 o superior
- RabbitMQ
- Entity Framework Core
- SQLite (para simplificar, o base de datos de preferencia del desarrollador)
- Docker