Proyecto de Programación Avanzada: Microservicios con WebSockets, Node.js y MSSQL

# Objetivo del Proyecto

Desarrollar un sistema compuesto por microservicios que permita la gestión y monitoreo en tiempo real de pedidos realizados por clientes, utilizando tecnologías modernas como Node.js, Express, WebSocket y Microsoft SQL Server (MSSQL).

# Descripción General

La aplicación simulará un entorno de comercio electrónico o restaurante donde los clientes pueden realizar pedidos y los operadores pueden ver en tiempo real el estado de los pedidos mediante notificaciones WebSocket.  
  
El sistema estará dividido en dos microservicios principales:  
1. Microservicio de Pedidos (creación y actualización de pedidos).  
2. Microservicio de Notificaciones WebSocket (notificación al cliente u operador cuando el estado de un pedido cambia).

# Requisitos Técnicos

- Backend: Node.js con Express.  
- Base de datos: Microsoft SQL Server.  
- WebSocket: Implementación con 'ws' o 'socket.io'.  
- Frontend: HTML básico con formularios y conexión WebSocket.  
- Arquitectura: Cada microservicio con su propio Dockerfile; opcional uso de docker-compose.

# Funcionalidades Mínimas

Cliente:  
- Realiza un nuevo pedido desde un formulario web.  
- Recibe una notificación WebSocket cuando su pedido cambia de estado.  
  
Operador:  
- Visualiza pedidos pendientes.  
- Cambia el estado del pedido.  
- Las actualizaciones se notifican en tiempo real al cliente.

# Estructura de Base de Datos Sugerida

|  |  |
| --- | --- |
| Campo | Tipo |
| idPedido | INT (PK) |
| nombreCliente | VARCHAR(100) |
| detallePedido | TEXT |
| estado | VARCHAR(50) |
| fechaRegistro | DATETIME |

# Criterios de Evaluación

- Implementación completa de CRUD: 20%  
- Integración de WebSocket funcional: 25%  
- Arquitectura desacoplada por microservicios: 20%  
- Diseño y estructura del proyecto: 15%  
- Dockerización: 10%  
- Presentación y documentación: 10%

# Entregables

1. Carpeta del proyecto con código fuente, Dockerfiles, docker-compose.yml, script SQL y archivos HTML.  
2. Documento .docx con descripción del sistema, diagramas, instrucciones de instalación y capturas de pantalla.

# Recomendaciones para los Estudiantes

- Usar async/await para operaciones con MSSQL.  
- Separar la lógica de negocio del controlador.  
- Validar que la conexión WebSocket se mantenga activa.  
- Documentar claramente cómo ejecutar los microservicios con Docker.