

Resumen de la configuración y ejecución de la prueba

Tecnologías utilizadas:

- Backend: API RESTful desarrollada con .NET 9
- Frontend: Angular 18
- Base de datos: SQL Server

Configuración del backend (API):

- Desarrollada en ASP.NET Core con Entity Framework Core
- Utiliza una arquitectura de repositorios y servicios
- Configurada para trabajar con SQL Server
- CORS habilitado para permitir solicitudes desde <http://localhost:4200>

Configuración del frontend:

- Desarrollado en Angular 18
- Diseñado para comunicarse con la API a través de HTTP

Proceso de ejecución del proyecto:

a. Compilación de la API:

- Navegar al directorio del proyecto de la API
- Ejecutar el comando de compilación (`dotnet build`)
- Iniciar la API (`dotnet run`)

b. Ejecución del frontend Angular:

- Navegar al directorio del proyecto Angular
- Ejecutar `ng serve` para iniciar el servidor de desarrollo de Angular

5. Acceso a la aplicación:

- La API estará disponible en la URL configurada (<http://localhost:5082/api>)
- La aplicación Angular estará accesible en <http://localhost:4200>

6. Características adicionales:

- Swagger habilitado para documentación de la API en entorno de desarrollo
- Implementación de seguridad básica con redirección HTTPS y autorización

Este proceso de ejecución en dos pasos (compilar la API y luego ejecutar el frontend) asegura que el backend esté listo y en funcionamiento antes de que el frontend intente comunicarse con él. Es una práctica común en el desarrollo de aplicaciones web full-

stack, especialmente cuando se trabaja con tecnologías separadas para el backend y el frontend, configurar la conexión con la base de datos.

Estructura de la API:

a. Repositorios (StoreSampleAPI/Repositories):

- Implementan la lógica de acceso a datos (ej: CustomerRepository, OrderRepository)

b. Interfaces (StoreSampleAPI/Interfaces):

- Definen contratos para los repositorios (ej: ICustomerRepository, IOrderRepository)

c. Modelos (StoreSampleAPI/Models):

- Representan entidades de la base de datos (ej: Customer, Order, Product)

d. Controladores (StoreSampleAPI/Controllers):

- Manejan solicitudes HTTP y definen endpoints (ej: CustomersController, OrdersController)

e. DTOs (StoreSampleAPI/DTOs):

- Objetos para transferencia de datos entre capas (ej: CustomerDTO, OrderDTO)

f. Servicios (StoreSampleAPI/Services):

- Contienen la lógica de negocio (ej: CustomerService, OrderService)

Esta estructura sigue las mejores prácticas de desarrollo en .NET, incluyendo el patrón repositorio, inyección de dependencias y separación de preocupaciones. La combinación de .NET 9 para el backend y Angular 18 para el frontend permite crear una aplicación web moderna y escalable. El proceso de ejecución en dos pasos asegura que el backend esté listo antes de que el frontend intente comunicarse con él.