

Presentado por : Alexander Paz  
Ingeniero de sistemas  
Fecha: 30 de sept. de 25  
Cali - Valle del cauca

### Prueba Técnica

La prueba se realizó con .NET CORE y el lenguaje C#. se utilizaron herramientas para base de datos Sql server, mapping , migraciones, fluentValidations y contenedores.

Se propuso una arquitectura Monolítica orientada a separar cada componente, pensando en una futura migración a microservicios si se llega a requerir.

Se aplicó buenas prácticas de programación y de desarrollo de Api orientado a REST.

Se aplico clean arquitectura para separar las capas de sus responsabilidades y se dejó el dominio limpio de aplicación e infraestructura; con esto evitamos dependencia de tecnología y garantizamos que las reglas de negocio solo vivan en el Dominio de la aplicación.

Se utilizaron contenedores de docker para empaquetar la aplicación en una imagen y poderla utilizar de forma rápida y fácil para la prueba. adicionalmente, esta orientación de despliegue nos permite apuntar a arquitecturas que nos permita escalar de forma rápida.

para ejecutar la prueba se debe:

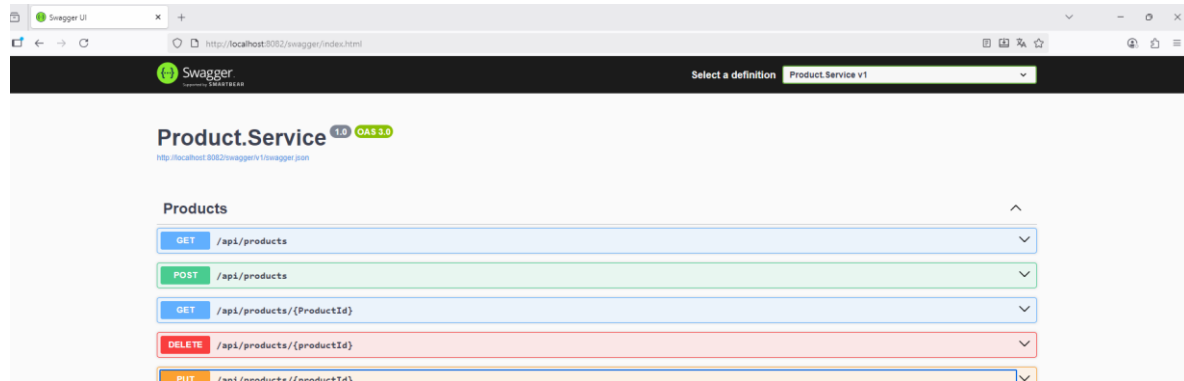
dentro de la raíz del proyecto (donde se encuentra la solución) ejecutar el comando:

```
docker-compose up -d
```

en el archivo docker-compose.yml está la configuración del contenedor de la aplicación y de la base de datos.

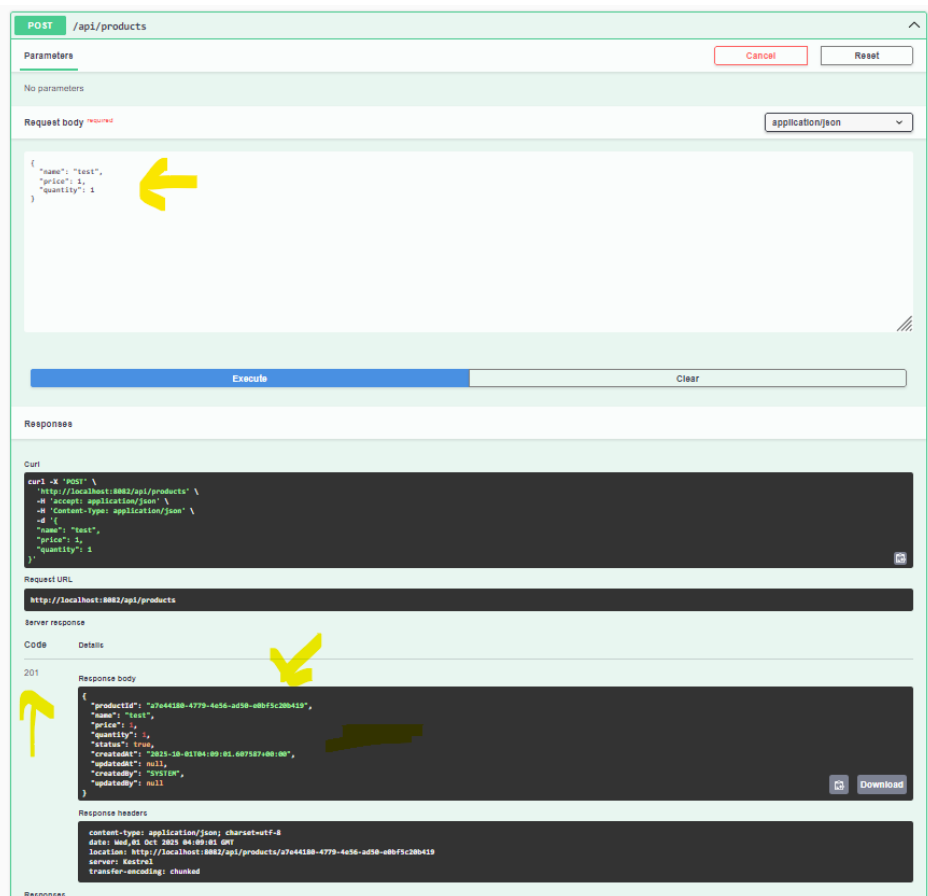
Luego, acceder a la url por donde se expuso la aplicación desde el contenedor Docker puerto 8082. Aquí podemos ver la api utilizando Swagger para ver la documentación.

<http://localhost:8082/swagger/index.html>

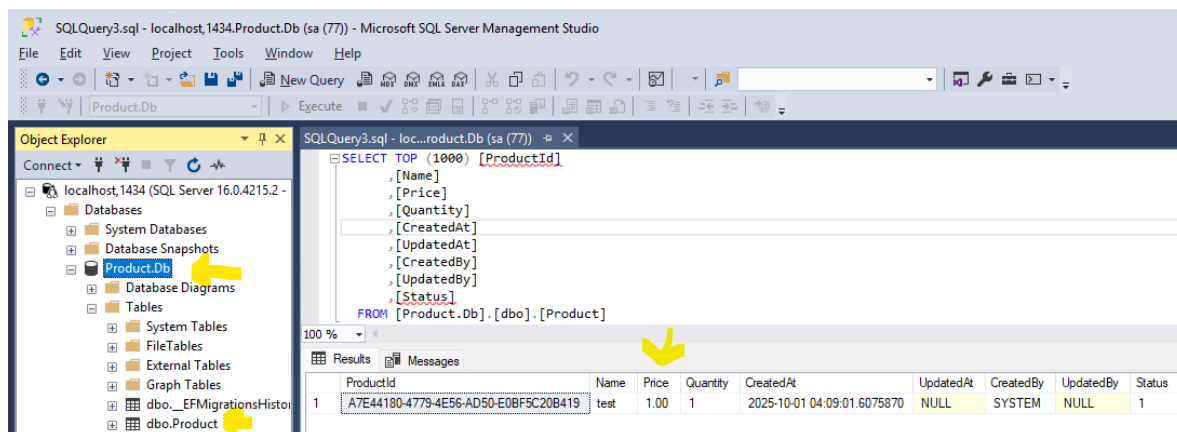


Finalmente, podemos realizar la utilización de la API y por medio de ella manipular los datos alojados en sqlserver. la aplicación está configurada para que realice la migración de la base de datos automáticamente para efectos de esta prueba.

Creando un registro desde la api.



Verificando el registro creado en la bd.



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'Product.Db', with 'dbo.Product' highlighted. The SQL Query window on the right contains the following query:

```
SELECT TOP (1000) [ProductId],
                [Name],
                [Price],
                [Quantity],
                [CreatedAt],
                [UpdatedAt],
                [CreatedBy],
                [UpdatedBy],
                [Status]
FROM [Product.Db].[dbo].[Product]
```

The query results are displayed in a table with the following columns: ProductId, Name, Price, Quantity, CreatedAt, UpdatedAt, CreatedBy, UpdatedBy, and Status. A single record is shown with the following values:

ProductId	Name	Price	Quantity	CreatedAt	UpdatedAt	CreatedBy	UpdatedBy	Status
A7E44180-4779-4E56-AD50-E0BF5C20B419	test	1.00	1	2025-10-01 04:09:01.6075870	NULL	SYSTEM	NULL	1

Todos los endpoints se probaron y funcionan correctamente.

FIN