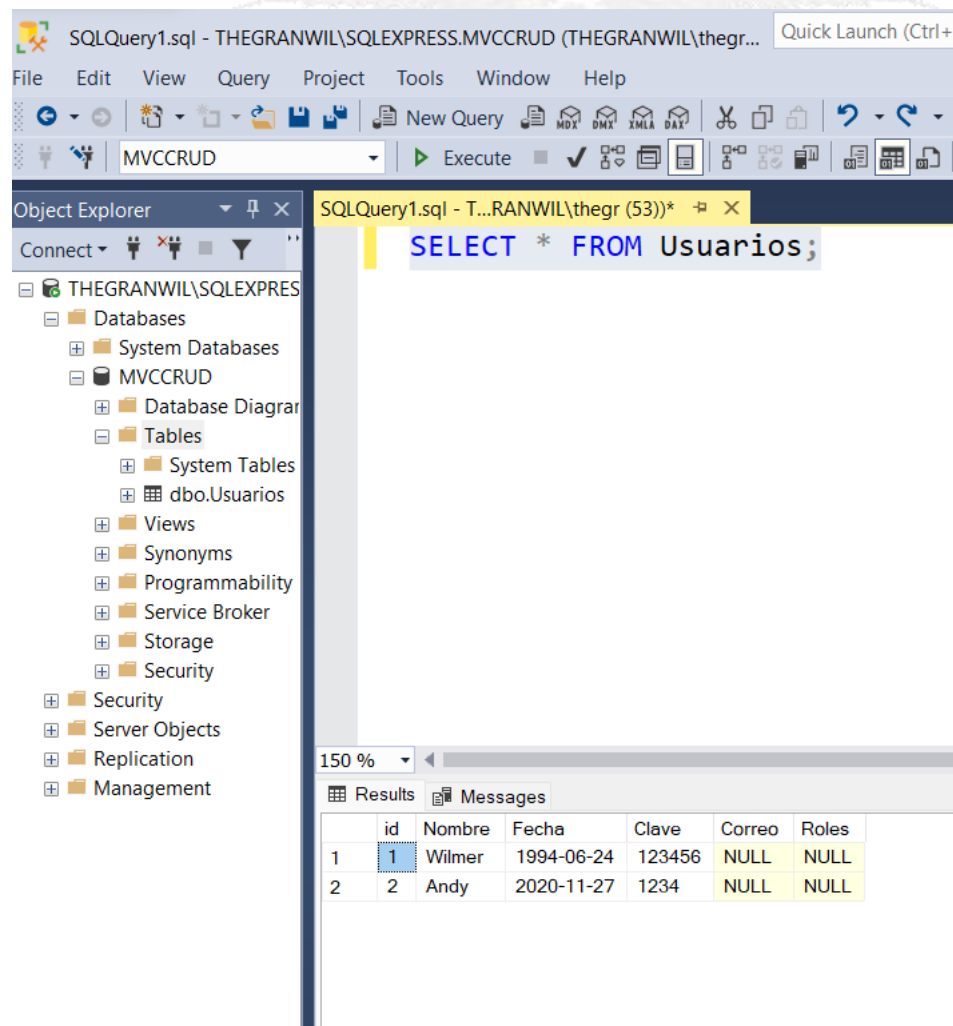


Actividad Práctica: Desarrollo de una API Segura con ASP.NET Core

Nombre: Wilmer Buestan

Objetivo:

Desarrollar una API RESTful segura utilizando ASP.NET Core, implementando prácticas avanzadas de autenticación y middleware.



The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'MVCCRU' database selected. The right pane shows a query window with the following SQL query:

```
SELECT * FROM Usuarios;
```

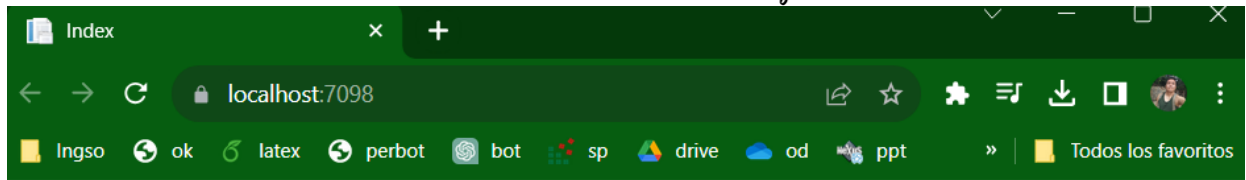
The query results are displayed in a table with the following columns: id, Nombre, Fecha, Clave, Correo, and Roles. The results are as follows:

	id	Nombre	Fecha	Clave	Correo	Roles
1	1	Wilmer	1994-06-24	123456	NULL	NULL
2	2	Andy	2020-11-27	1234	NULL	NULL

Instrucciones:

1. Configuración del Proyecto:

- Crea un nuevo proyecto ASP.NET Core con el patrón de diseño MVC.



Email

Contraseña

Ingresar

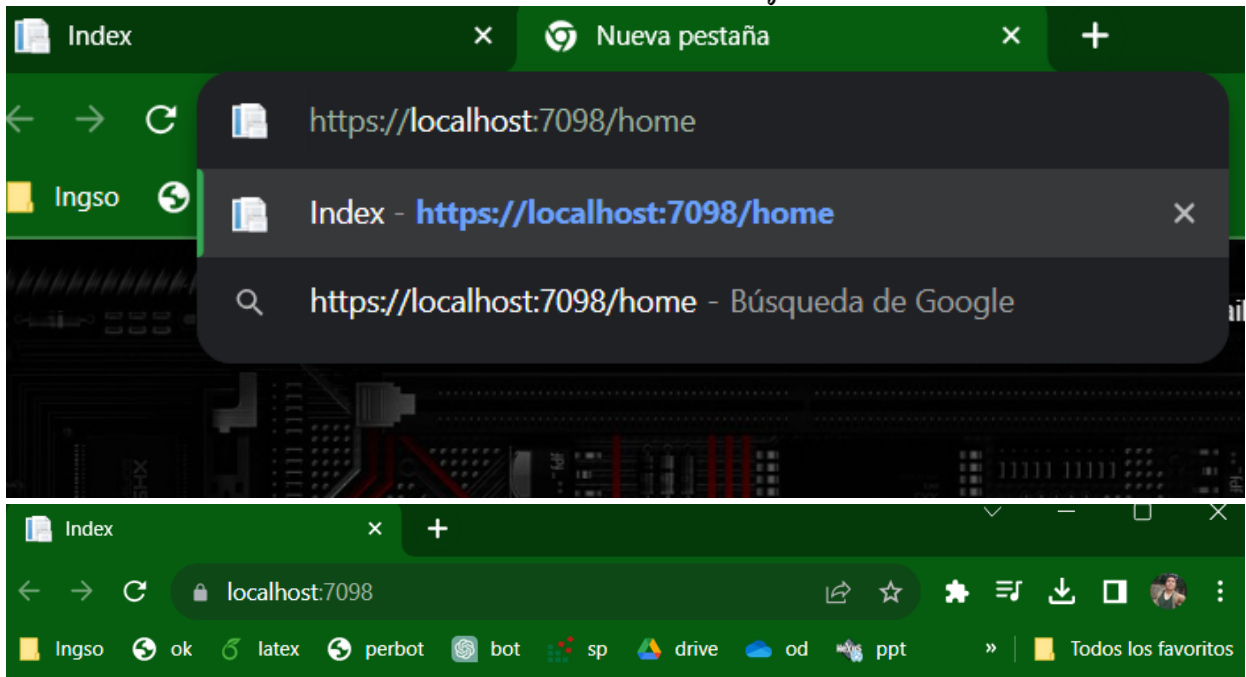
-

- Configure el middleware de autenticación para utilizar tokens JWT.

2. Implementación de la API RESTful:

- Diseña y desarrolla al menos dos controladores para gestionar operaciones CRUD en entidades específicas.

- Utiliza atributos como **[Authorize]** para asegurar que las operaciones requieran autenticación.



Email

Contraseña

Ingresar

3. Seguridad y Middleware:

- Implementa middleware personalizado para registrar las solicitudes a la API antes y después del procesamiento.
- Explore la configuración de políticas de autorización para controlar el acceso a ciertos recursos.

4. Documentación:

- Proporciona documentación clara para la API, incluyendo los endpoints disponibles y los requisitos de autenticación.

Detalles de Acceso

Correo de Acceso: wilo@gmail.com

Contraseña: 123

Implementación

1. Autenticación

Se implementó un sistema de autenticación basado en tokens JWT (JSON Web Tokens) para garantizar la seguridad de la API. Al iniciar sesión con las credenciales proporcionadas (correo y contraseña), se genera un token JWT que debe incluirse en las cabeceras de las solicitudes para acceder a los recursos protegidos.

2. Endpoints Disponibles

2.1 Obtener Datos del Usuario

Endpoint: /api/usuario

Método: GET

Requiere Autenticación: Sí

Descripción: Obtiene la información del usuario autenticado.

2.2 Crear Nuevo Recurso

Endpoint: /api/nuevo-recurso

Método: POST

Requiere Autenticación: Sí

Descripción: Crea un nuevo recurso en la API.

2.3 Actualizar Recurso Existente



Endpoint: /api/actualizar-recurso/{id}

Método: PUT

Requiere Autenticación: Sí

Descripción: Actualiza el recurso identificado por el ID proporcionado.

2.4 Eliminar Recurso

Endpoint: /api/eliminar-recurso/{id}

Método: DELETE

Requiere Autenticación: Sí

Descripción: Elimina el recurso identificado por el ID proporcionado.

3. Middleware de Seguridad

Se implementaron middleware de seguridad para proteger la API contra posibles amenazas, como ataques de CSRF (Cross-Site Request Forgery) y XSS (Cross-Site Scripting). Estos middleware aseguran la integridad de las solicitudes y respuestas.