Examen 2ª Evaluación

El examen consiste en desarrollar tres proyectos relacionados entre sí:

1. **API REST con .NET 9**
2. **Aplicación de escritorio en WPF (MVVM)**
3. **Aplicación web en Angular**

API REST con .NET 9

La API gestionará tres entidades con las siguientes relaciones:

- **Usuarios**
- **Productos**
- **Pedidos**
- Relación **N a N** entre Usuarios y Productos.

Base de Datos:

- **Password**: `jS7R4qwUYx9e6uM5k8Zhfv`
- **Nombre de la base de datos**: `TiendaDB`
- **Puerto**: `5432`
- **Usuario de la base de datos**: `api_user`

Endpoints requeridos

Autenticación

POST /auth/login

Request:

```
{
  "email": "usuario@example.com",
  "password": "123456"
}
```

Response:

```
{
  "token": "eyJhbGciOiJIUzI1NiIsInR..."
}
```

POST /auth/register

Request:

```
{
  "nombre": "Juan Pérez",
```

```
  "email": "juan@example.com",
  "password": "123456"
  "rol": "WPF_User"
}
```

Roles en la API:

- **WPF_User**: Usuario con permisos para realizar operaciones CRUD sobre las entidades **Productos** y **Pedidos**.
- **Angular_User**: Usuario con permisos solo para realizar operaciones CRUD sobre la entidad **Usuarios**.
- **Cread** un usuario cuyo rol es **WPF_User** y su nombre será WPF **DAM** y otro para el rol **Angular_User** cuyo nombres es **Angular_DAM**, ambas usuarios tendrán la misma contraseña: G,CKvs4j7WnR+-T"m!Q~6{

Gestión de Entidades

Get/Post/Delete /usuarios

Response:

```
[
  {
      "id": 1,
      "nombre": "Juan Pérez",
      "email": "juan@example.com",
  }
]
```

Get/Post/Delete /productos

Response:

```
[
  {
    "id": 2,
    "nombre": "Laptop",
    "precio": 1200.99,
  }
]
```

Get/Post/Delete /pedidos

Response:

```
[
  {
    "id": 1,
    "fecha": "2025-03-05",
    "usuario": 1
    "productos": [2, 5]
  }
]
```

Aplicación WPF (MVVM)

Menú de navegación

El menú lateral contendrá las siguientes opciones:

- **Productos**: Muestra la lista de productos en una DataGrid.
- **Pedidos**: Muestra la lista de pedidos en una DataGrid.
- **Importar y Exportar Datos**: Carga o descarga información en JSON.

Views detalladas

View Productos

- DataGrid con las columnas:
  - ID
  - Nombre
  - Precio

View Pedidos

- StackPanel con las columnas:
  - ID
  - Fecha
  - Usuario que realizó el pedido
- Al hacer clic en un pedido, se mostrará una vista lateral con los productos que contiene, se podrán añadir y borrar más productos.

Importar y Exportar Datos

- **Importar**: Se abrirá un OpenFileDialog para cargar un JSON con datos de usuarios, productos y pedidos.
- **Exportar**: Se podrá guardar un archivo JSON con los datos de usuarios, productos y pedidos actuales mediante SaveFileDialog.

Aplicación Angular - Gestión de Usuarios

Para la gestión de usuarios crearás una aplicación Angular. Los usuarios gestionados en esta aplicación no son los mismos que los usuarios de autenticación JWT.

1. **Página de Gestión de Usuarios:**
    - Deberás crear una página donde se muestren los usuarios registrados en la tienda. Esta página debe incluir una lista de usuarios con las siguientes columnas:
        - ID
        - Nombre
        - Email
2. **Operaciones CRUD: L**os usuarios deben poder ser añadidos, modificados y eliminados desde la interfaz.
    - Añadir Usuario:
        - Se deberá permitir añadir nuevos usuarios con un formulario que solicite al menos:
            - Nombre
            - Email
    - Modificar Usuario:
        - Al hacer clic en un usuario de la lista, se deberá permitir editar los campos de su nombre y email.
    - Eliminar Usuario:
        - Se debe permitir eliminar un usuario de la lista mediante un botón de "Eliminar".