

# Prueba de Programación Full-Stack

---

## 1. Back-End

### Lenguajes/Frameworks:

- .NET (ASP.NET Core)

### Objetivos:

- Crear dos servicios web API que permitan gestionar un CRUD de usuarios y productos.
- Implementar autenticación básica utilizando JWT en ambas APIs.
- Asegurar que las APIs puedan integrarse y comunicarse entre sí.
- Integración con la base de datos SQL Server.

### Tareas:

- .NET (ASP.NET Core):
  - Crear una API que gestione productos (create, read, update, delete).
  - Crear una API que gestione usuarios (create, read, update, delete).
  - Incluir un endpoint para calcular el stock de productos.
  - Asegurar que ambas APIs puedan ser consumidas por la aplicación Angular.
- Implementar autenticación con JWT (seleccione el lenguaje en el que quiere implementarlo).

## 2. Front-End

### Framework:

- Angular

### Objetivos:

- Crear una aplicación Angular que consuma las APIs desarrolladas en el back-end.
- Implementar un frontend que permita la gestión de usuarios y productos.
- Manejo de autenticación con JWT y rutas protegidas.
- Crear una interfaz dinámica y responsiva.

### Tareas:

- Crear un módulo de autenticación que funcione con las APIs.
- Desarrollar componentes para crear, editar, eliminar y listar usuarios y productos.
- Implementar un sistema de rutas seguras basado en la autenticación.

### 3. Bases de Datos

#### Base de Datos:

- SQL Server

#### Objetivos:

- Diseñar la base de datos para usuarios y productos.
- Crear stored procedures para operaciones complejas.
- Configurar la base de datos para que sea consumida por ambas APIs.

#### Tareas:

- Crear las tablas Users y Products.
- Escribir procedimientos almacenados que optimicen las consultas necesarias.
- Configurar las conexiones para .NET.

### Entrega y Evaluación

#### Entrega:

- Subir el código a un repositorio en GitHub.
- Generar collection de postman en formato json.
- Incluir instrucciones detalladas sobre cómo desplegar las APIs y la aplicación Angular.
- Documentar las decisiones arquitectónicas y las configuraciones de la base de datos.

#### Criterios de Evaluación:

- Funcionalidad: ¿Las APIs y la aplicación Angular funcionan correctamente?
- Código Limpio: ¿El código sigue buenas prácticas y es fácil de entender?
- Seguridad: ¿El sistema de autenticación es robusto?
- Eficiencia: ¿Las consultas y operaciones en la base de datos están optimizadas?