

Desarrollar una Aplicación de Gestión de Cuentas con Arquitectura de Servicios y Autenticación por Token

SG Financial Technology necesita desarrollar una aplicación de gestión de cuentas para sus clientes con un nivel de seguridad adicional. La aplicación debe estar orientada a una arquitectura de servicios y permitir a los usuarios realizar las siguientes acciones:

1. Crear una cuenta.
2. Realizar depósitos en una cuenta.
3. Realizar retiros de una cuenta.
4. Consultar el saldo de una cuenta.

Además, para garantizar la seguridad, todos los usuarios deben autenticarse mediante tokens.

Requisitos Técnicos:

1. Utiliza C# y .NET Core para desarrollar los servicios.
2. Almacena los datos de las cuentas en una base de datos SQL Server.
3. Implementa un sistema de control de versiones utilizando GitHub.
4. Utiliza autenticación basada en JWT Token para proteger las llamadas a los servicios/controllers (dependiendo de que se utilice).
5. Cada servicio debe ser independiente y manejar una funcionalidad específica.

Instrucciones:

1. Cree un repositorio en Git para su proyecto y compártelo.
2. Implemente los servicios en C# y .NET Core, incluyendo la autenticación basada en tokens.
3. Utilice SQL Server para almacenar los datos de las cuentas.
4. Proporcione documentación clara sobre cómo configurar y ejecutar la aplicación, listar los endpoints creados (si corresponde), los requests permitidos e incluyendo cómo obtener y utilizar tokens de autenticación.
5. Proporcione los scripts de creación e inserción de datos para la base de datos. No es necesario crear una cuenta en Azure o similar, se probará en un ambiente local.

Documentación Esperada:

1. **Configuración y Ejecución:**
 - Instrucciones pasó a paso sobre cómo configurar y ejecutar la app.
2. **Autenticación:**
 - Detalles sobre cómo obtener un JWT Token.
 - Ejemplo de uso del JWT Token para autenticar las llamadas a los servicios.
3. **Endpoints Creados (si corresponde):**
 - Listado de todos los endpoints disponibles en cada servicio.
 - Ejemplos de requests permitidos y respuestas esperadas.

Aclaraciones:

- Los servicios deben ser independientes y comunicarse entre sí de manera eficiente.
- Se valorará la claridad y la calidad del código, así como la estructura del proyecto y la documentación proporcionada.