

Ejercicio de Entrevista Técnica

Tu tarea es desarrollar una API con capa de datos usando .NET CORE, C#, y una base de datos relacional, siguiendo los principios de Arquitectura Limpia (Clean Architecture) y utilizando metodologías de Desarrollo Guiado por Pruebas (TDD).

Tu desarrollo debe basarse en una historia de usuario informal que crearás y que se debe incluir en tu presentación. La aplicación debe permitir a los usuarios crear, leer, actualizar y eliminar registros de la base de datos a través de los endpoints de la API. Además, **deberás crear un usuario, iniciar sesión como usuario y asegurar que la información del usuario se almacene en la base de datos (Vía API).**

Para demostrar tu capacidad de trabajar con sistemas de almacenamiento modernos, **no puedes utilizar Entity Framework** en este ejercicio.

Componentes de la aplicación:

- **Base de datos:** Crea una base de datos relacional con al menos una tabla para almacenar datos de la aplicación. La tabla debe tener un identificador único (clave primaria) y al menos dos campos adicionales. Esta tabla debe estar separada de los datos de usuarios/seguridad.
- **API:** Desarrolla una API web ASP.NET con endpoints que permitan a los usuarios realizar operaciones CRUD en la tabla de la base de datos. Cada endpoint debe tener el verbo HTTP adecuado, parámetros, y valores de retorno. Además, crea una segunda API con endpoints para la creación de usuarios, inicio de sesión, y endpoints autorizados y no autorizados.
- **Capa de datos:** Desarrolla una capa de acceso a datos que interactúe con la base de datos y proporcione las operaciones CRUD necesarias para los endpoints de la API.
- **Capa de lógica de negocio:** Desarrolla una capa de lógica de negocio que incluya todas las reglas de negocio y validaciones de la aplicación. Esta capa debe ser independiente de la capa de datos y de la API.
- **Pruebas unitarias:** Escribe pruebas unitarias para todos los componentes de la aplicación, incluidas las capas de acceso a datos, lógica de negocio y endpoints de la API.

La aplicación debe ser totalmente ejecutable de manera local, y se prefiere el uso de Docker. No se requiere desplegar código ni utilizar servicios en la nube.

Presentación y Revisión de Código:

Una vez que completes el ejercicio, deberás presentar tu proyecto. Durante la presentación, debes explicar tu historia de usuario, decisiones de diseño, la arquitectura técnica y demostrar la funcionalidad de la aplicación.

Después de la presentación, se realizará una revisión del código y se te hará preguntas sobre tus decisiones de codificación. Se evaluará tu proyecto basado en los siguientes criterios:

- **Arquitectura Limpia:** Tu arquitectura debe seguir los principios de Clean Architecture, incluyendo la separación de responsabilidades y la independencia de los componentes.
- **Desarrollo Guiado por Pruebas (TDD):** Tu proyecto debe seguir las metodologías TDD e incluir pruebas unitarias para todos los componentes.
- **Calidad de código:** Tu código debe estar bien organizado, ser legible y seguir las mejores prácticas.
- **Funcionalidad:** La aplicación debe realizar las operaciones CRUD requeridas y la autenticación de usuarios sin errores ni bugs.
- **Historia de usuario:** La historia de usuario debe guiar el desarrollo de la aplicación y estar incluida en la presentación.
- **Presentación:** La presentación debe ser clara, concisa y demostrar un buen entendimiento del proyecto.