

Análisis de código: ISOmeter GO!

Versión Álvaro Ballarini:

Tanto el Frontend como el Backend se encuentran en proceso de reestructuración, migrando de lenguaje JavaScript a TypeScript. A grandes rasgos son lo mismo, solo que el segundo es fuertemente tipado, lo que conlleva a un mejor manejo de tipos de datos y reducción de errores inesperados.

Frontend:

- Legibilidad y mantenibilidad: Buena legibilidad, se podría mejorar la mantenibilidad.
- Arquitectura y diseño: Arquitectura MVC. Tiene buena organización y división de carpetas por funcionalidad, pero en algunos casos no es claro y no agrupa el contenido asociado a cada una.
- Seguridad: Generación de un token encriptado al iniciar sesión y validación de usuarios. . Ambos sistemas guardan el token en el almacenamiento local del navegador, lo cual no es lo ideal.
- Manejo de errores: Buen manejo de errores.
- Rendimiento: Rendimiento estable, pero mejorable.
- Consistencia y estándares: Buena consistencia, pero tiene posibilidad de mejora. En algunos casos hay una “mega división” de carpetas y archivos que podrían estar unificados.
- Configuración y despliegue: Posee una configuración sencilla, solo falta el instructivo para realizarla. Enfocado a correr en cualquier entorno.

Resumen: el código está bien, solo falta pulir detalles y agrupar funcionalidades por entidad.

Backend:

El Backend está muy bien hecho:

- Correcta distribución de carpetas por funcionalidad
- Escalable
- Alta legibilidad

Puntos a tener en cuenta:

- Refactorización y limpieza de código en el archivo principal.
- Limpiar los controladores: no usar lógica de negocio dentro de ellos, para facilitar lectura.

Versión Matías y Benicio:

Es una versión hecha en su mayoría con inteligencia artificial, código robusto y estructurado, pero en algunos casos difícil de entender o interpretar sobre todo en el apartado Backend.

Frontend:

- Legibilidad y mantenibilidad: Un poco más intuitivo, buena legibilidad, se podría mejorar la mantenibilidad.

- Arquitectura y diseño: Arquitectura MVC. Tiene buena organización y división de carpetas por funcionalidad. Muy intuitivo y ordenado.
- Seguridad: Generación de un token encriptado al iniciar sesión y validación de usuarios. Ambos sistemas guardan el token en el almacenamiento local del navegador, lo cual no es lo ideal.
- Manejo de errores: Manejan errores en algunos casos importantes, pero se podría mejorar. Sobre todo, en inputs de datos.
- Rendimiento: Rendimiento estable.
- Consistencia y estándares: Buenos estándares, pero se podría mejorar. En ambos sistemas se usa el “spanglish” para nombrar archivos.
- Configuración y despliegue: Posee una configuración sencilla, falta el instructivo para realizarla. Despliegue ajustado para correr en el laboratorio, con posibilidad de expansión.

Resumen: el código es legible, ordenado y escalable.

Backend:

- Correcta distribución de carpetas por funcionalidad
- Escalable
- Poco legible

Puntos a tener en cuenta:

- Refactorización y limpieza de código en el archivo principal.
- Limpiar los controladores: no usar lógica de negocio dentro de ellos, para facilitar lectura. Separar por entidad / funcionalidad.