🧪 Prueba Técnica Fullstack — Escaleno

> Bienvenido/a a la prueba técnica para desarrolladores fullstack en **Escaleno**.
> El objetivo es simple, pero te da espacio para destacar si decides ir más allá.
> Se evalúa la funcionalidad, claridad, estructura del código y buenas prácticas.

🎯 Objetivo

Desarrollar una aplicación web que permita **cargar un archivo PDF o JPG con la revisión técnica de un vehículo**, y que el sistema sea lo suficientemente **inteligente como para detectar si el documento es real o presenta adulteraciones**.

Puedes utilizar técnicas de visión computacional, validación de metadatos, OCR, o cualquier estrategia que consideres adecuada para determinar la autenticidad del documento.

📋 Requisitos Obligatorios

Estos puntos son imprescindibles para considerar válida la entrega:

- Subida de archivos PDF o imágenes (JPG/JPEG/PNG)
- Procesamiento del documento para extraer información clave
- Verificación automática de autenticidad del documento (real o adulterado)
- Mensaje claro del resultado al usuario (ej: “Documento válido” / “Documento con posibles adulteraciones”)
- Diseño **responsive** y **usable**

⚙️ Convenciones Técnicas

- Puedes usar **JavaScript o TypeScript**
  (preferimos JS + Express, pero TS también está OK)
- Puedes tener **todo en un solo repo**, pero suma puntos separar **backend** y **frontend**
- Sube el/los repositorios a **GitHub**
- Usa **commits semánticos** según Conventional Commits
- Incluye un `README.md` claro con instrucciones para correr el proyecto

---

🤖 Uso de IA

Puedes usar inteligencia artificial como apoyo, sabemos que es parte del flujo de trabajo moderno, pero no abuses, lo sabremos 👀. Todos la usan ¿Qué te hará destacar frente a los demás?

🚀 Requisitos Opcionales (para destacar)

> No son obligatorios, pero marcan una gran diferencia
> Ideal si quieres demostrar tu nivel técnico

🧱 Arquitectura & Código

- Clean Architecture o separación clara de capas (presentación, dominio, datos)
- Uso de patrones: `Factory`, `Repository`, `Adapter`, etc.

🧠 Estado & Datos

- Manejo de errores: fallbacks, retries, loaders
- Implementación de estructuras de datos: colas (`Queue`), mapas (`Map`), búsquedas, etc.

🧩 Funcionalidades Extra

- Historial de documentos verificados
- Comparación visual del archivo contra plantillas válidas
- Sistema de usuarios con autenticación
- Dockerizar tus aplicaciones

🧪 Testing (solo Backend)

- Pruebas unitarias (lógica o rutas, ideal 80% coverage)
- Librerías sugeridas: `Jest`, `Testing Library`, `Vitest`, etc.

Recursos Útiles

Recurso	Enlace
Ejemplos de revisión técnica	https://drive.google.com/drive/folders/1CGM8VeGbaPw-5Ltie-OjrSXYLvCPCmzc

---

Buena suerte.
Equipo Técnico – Escaleno