

Elección de entorno

El entorno elegido es web/mobile, priorizando la experiencia en celulares pero con la posibilidad de usar la misma aplicación en computadoras de escritorio si fuera necesario.

Justificación:

El cliente expresó que necesita acceder principalmente desde el celular, ya que el relevamiento de bienes se hace en movimiento.

Una aplicación web responsive evita instalar software adicional y siempre se mantiene actualizada en todos los dispositivos.

La opción de escritorio fue descartada, ya que no se ajusta a la necesidad de movilidad y practicidad del cliente.

Pros del entorno elegido (web/mobile):

- Acceso inmediato desde cualquier dispositivo con internet.
- Facilidad de uso y diseño amigable.
- Actualizaciones automáticas y centralizadas.
- Bajo costo de desarrollo en comparación con apps nativas.

Contras:

- Dependencia de conexión a internet.
- Funcionalidades limitadas en comparación con una app nativa (ejemplo: notificaciones push avanzadas).

Elección de lenguajes y tecnologías

Las tecnologías seleccionadas buscan garantizar un sistema ágil, confiable y escalable:

React + JavaScript (Frontend)

Por qué sí:

Permite crear interfaces modernas, dinámicas y fáciles de usar.

Compatible con diseño responsive para celulares y computadoras.

Amplia comunidad y soporte de librerías para optimizar el desarrollo.

Por qué no:

Requiere una curva de aprendizaje mayor que un desarrollo web simple en HTML/CSS/JS.

Pros: Componentes reutilizables, gran rendimiento, escalabilidad.

Contras: Configuración inicial más compleja.

Node.js + Express (Backend)

Por qué sí:

Ideal para aplicaciones en tiempo real, como la actualización automática de inventarios.

Express ofrece un framework rápido y sencillo para manejar rutas, permisos y validaciones.

El uso de JavaScript en frontend y backend unifica el stack, facilitando la comunicación.

Por qué no:

No es la mejor opción para procesos muy pesados en CPU.

Pros: Ligero, escalable, rápido para prototipos y sistemas productivos.

Contras: Menor rendimiento que lenguajes como Java en cálculos complejos.

MySQL (Base de datos)

Por qué sí:

Sistema robusto y ampliamente probado para manejar relaciones entre usuarios, roles, categorías y productos.

Permite consultas rápidas para búsquedas por código, nombre o estado.

Gratuito, estable y seguro.

Por qué no:

Menos flexible que bases NoSQL (ej: MongoDB) para datos no estructurados.

Pros: Seguridad, estabilidad, transacciones confiables.

Contras: Modelo de datos menos adaptable frente a cambios drásticos.