Justificación de Tecnologías

Backend:

Tecnología nativa

Se utiliza JavaScript vanilla porque proporciona total control sobre el código, máximo rendimiento sin dependencias externas, y compatibilidad universal con todos los entornos Node.js y navegadores.

Framework de Desarrollo: NodeJS - Express.js

Se selecciona Express.js como framework backend para crear una API RESTful escalable que sirva como base del sistema de gestión de proyectos. Esta tecnología resuelve la complejidad de desarrollar un servidor desde cero, proporcionando una estructura organizada para manejar rutas, autenticación y lógica de negocio. Es importante por ser la solución más madura y performante para Node.js, beneficiando tanto a desarrolladores con mayor productividad como a usuarios finales con una API confiable y eficiente.

Tecnología de Persistencia: PostgreSQL

PostgreSQL se implementa como base de datos relacional para garantizar la integridad y consistencia de los datos empresariales del sistema. Resuelve el desafío del manejo confiable de información transaccional mediante su robusto sistema ACID, constraints y relaciones bien definidas. Su importancia radica en ofrecer características empresariales como triggers y validaciones a nivel de base de datos, beneficiando a la organización con datos confiables y a los desarrolladores con potentes capacidades de consulta.

Sistema de Autenticación: JWT + BCrypt

La combinación JWT y BCrypt se utiliza para implementar un sistema de autenticación seguro y sin estado. Esta solución aborda el problema de la gestión segura de sesiones y protección de credenciales, siendo crucial para prevenir accesos no autorizados y fugas de información. Beneficia directamente a los usuarios mediante la protección de sus datos personales y a la empresa al salvaguardar su información sensible.

Suite de Testing: Jest + Supertest

Se implementa Jest con Supertest para asegurar la calidad y estabilidad del software mediante pruebas automatizadas. Esta suite resuelve la necesidad de detectar errores tempranamente y prevenir regresiones en un sistema empresarial crítico. Su importancia se manifiesta en la reducción de costos de mantenimiento y el aumento de confianza en los despliegues, beneficiando tanto al equipo de desarrollo como a los clientes finales.

Frontend:

Framework de Desarrollo: React con Vite

React con Vite se selecciona como framework principal para construir una interfaz de usuario moderna y escalable, sirviendo como base para desarrollar una aplicación web empresarial single-page. Resuelve el problema de la complejidad en el manejo de interfaces dinámicas mediante componentes reutilizables y estado predecible, siendo importante porque permite un desarrollo ágil con hot-reload para iteraciones rápidas y optimizaciones de rendimiento automáticas. Beneficia directamente a los desarrolladores con una experiencia de desarrollo eficiente y a los usuarios finales con una aplicación responsive y de alto rendimiento.

Tecnología de Persistencia: TanStack Query con Axios

TanStack Query con Axios se implementa para la persistencia y gestión de datos del servidor, sirviendo como capa intermedia entre el frontend y las APIs REST del backend. Resuelve problemas de cache manual, estados de carga complejos y sincronización de datos en tiempo real, siendo crucial porque automatiza el caching, background refetch y manejo de errores, mejorando la experiencia de usuario con datos siempre actualizados. Beneficia a los desarrolladores simplificando la lógica de datos y a los usuarios con interfaces más rápidas y confiables que responden inmediatamente mientras se sincronizan en segundo plano.