

## 1. Análisis Comparativo (Benchmarking) del Stack

La elección de un stack tecnológico es la decisión fundamental que impactará el rendimiento, la escalabilidad y la mantenibilidad del proyecto. El stack propuesto (React, Express, PostgreSQL) se ha seleccionado tras un análisis comparativo riguroso.

### 1.1. Tabla de Benchmarking sobre herramientas Frontend

Criterio de Decisión	React	Vue.js	Angular
Curva de Aprendizaje	Moderada. Requiere entender JSX y el flujo de estado, pero es más simple que Angular.	Baja. Considerado el más fácil de aprender, con una sintaxis muy amigable.	Alta. Es un framework completo (no una librería) con conceptos complejos (TypeScript, Inyección de Dependencias, Módulos).
Rendimiento	Muy Alto. Usa un Virtual DOM que optimiza las actualizaciones de la UI, ideal para PWA.	Muy Alto. Similar a React en rendimiento, a veces marginalmente más rápido en renderizado inicial.	Alto. Usa un DOM real incremental, pero su bundle size (tamaño del paquete inicial) suele ser más grande.
Ecosistema y Comunidad	Masivo. Es el estándar de la industria. Acceso a una cantidad inmensa de librerías, tutoriales y talento (desarrolladores).	Grande y en crecimiento. Ecosistema muy sólido, pero más pequeño que el de React.	Grande. Respaldado por Google, muy usado en entornos corporativos y empresariales.
Soporte PWA	Excelente. <code>create-react-app</code> ofrece plantillas PWA listas para usar. Integración nativa con <code>Workbox</code> para Service Workers.	Excelente. Vue CLI tiene un soporte PWA fantástico y fácil de configurar.	Excelente. Angular CLI tiene un módulo PWA ( <code>@angular/pwa</code> ) que automatiza gran parte del proceso.

### 1.2. Tabla de Benchmarking sobre herramientas Backend

Criterio de Decisión	Express.js	Django - Python	Spring Boot - Java
Curva de Aprendizaje	Baja. Es un framework minimalista y no "opinado". La complejidad la añade el desarrollador, no el framework.	Moderada. Es "opinado" y "baterías incluidas". Te fuerza a su estructura (MVT), pero resuelve mucho (Admin, ORM).	Alta. Requiere un profundo conocimiento de Java, el ecosistema Spring, e inyección de dependencias.
Rendimiento	Alto. Al estar sobre Node.js (V8), es extremadamente rápido para operaciones I/O (consultas a BD, APIs), que es el 90% del trabajo.	Bueno. Python es más lento que Node.js en ejecución pura, pero muy eficiente para tareas de datos y scripting.	Muy Alto. La JVM es una máquina de rendimiento robusta y probada para cómputo pesado y alta concurrencia.

Flexibilidad vs. Estructura	Flexibilidad Total. Es un "lienzo en blanco". Ideal para construir APIs RESTful personalizadas y microservicios.	Estructurado. Te da todo hecho (Admin, ORM, Seguridad), lo que acelera el desarrollo inicial, pero puede ser restrictivo.	Estructurado. Muy robusto y seguro, pero verboso. Diseñado para arquitecturas empresariales complejas.
Integración (Stack)	Perfecta. Usa JavaScript, el mismo lenguaje que React. Reduce la carga cognitiva (un solo lenguaje "Full-Stack").	Buena. Python es el lenguaje del modelo predictivo, lo que podría ser una ventaja, pero requiere 2 lenguajes en el stack web.	Compleja. Integrar React con un backend Java es común, pero añade complejidad de build tools y tooling.

### 1.3. Tabla de Benchmarking sobre DBMS para Base de datos

Criterio de Decisión	PostgreSQL	MySQL	MongoDB - NoSQL
Modelo de Datos	Relacional (Objeto-Relacional). El más avanzado y compatible con SQL estándar.	Relacional. El estándar de la industria durante años, muy robusto y fiable.	Documental (NoSQL). Almacena datos en documentos JSON/BSON flexibles.
Integridad de Datos (ACID)	Excepcional. Considerado el "gold standard" para la integridad transaccional (crucial para ventas e inventario).	Muy Alta. Cumple totalmente con ACID, es la base de aplicaciones transaccionales por décadas.	Baja (Eventual). No está diseñado para transacciones complejas entre múltiples "tablas" (colecciones). No es su fortaleza.
Tipos de Datos y Extensibilidad	Superior. Soporta tipos de datos avanzados (JSONB, GIS, Arrays). Su sistema de extensiones (ej. PostGIS) es inigualable.	Buena. Soporta JSON, pero sus capacidades de consulta sobre JSON son más limitadas que JSONB de Postgres.	Excelente (Flexibilidad). El esquema es dinámico, lo que permite agilidad. Pero esta flexibilidad es un riesgo para datos financieros.
Integración (Datos/Modelo)	Excelente. Es la BD preferido por la comunidad de Data Science y Python (modelos predictivos) por su robustez analítica.	Muy Buena. También tiene excelentes conectores, pero Postgres es generalmente preferido para analítica avanzada.	Pobre. El modelo predictivo necesitará datos estructurados y relacionales (historial de ventas). Extraer esto de MongoDB es ineficiente.

### 1.4. Justificación de la Elección del Stack (PERN)

La elección del stack PERN (PostgreSQL, Express.js, React.js, Node.js) no es arbitraria; es una decisión estratégica diseñada para las necesidades específicas del proyecto KALLARI.

- Sinergia JavaScript: El uso de React (frontend) y Express (backend) nos permite operar en un ecosistema JavaScript unificado. Esto reduce drásticamente la fricción de desarrollo, simplifica el tooling y permite al equipo de desarrollo ser "full-stack" de manera más eficiente.
- Reactividad para PWA: React es la elección óptima para el frontend debido a su arquitectura basada en componentes y su Virtual DOM. Esto nos permite crear una interfaz de usuario extremadamente rápida y fluida, que es un requisito no negociable para una PWA que debe sentirse como una aplicación nativa, tanto en el catálogo de productos como en los dashboards de administración.
- Seguridad y Flexibilidad en la API: Express.js proporciona el equilibrio perfecto. Es minimalista, lo que nos da control total sobre la lógica de negocio, pero lo suficientemente robusto para implementar, mediante su sistema de middleware, la arquitectura de seguridad compleja que este proyecto requiere (JWT, roles, rutas protegidas). No nos impone una estructura rígida como Django, permitiéndonos diseñar la API exactamente como la necesitamos.
- Integridad de Datos y Futuro Analítico: PostgreSQL es la piedra angular. Este proyecto gestionará transacciones (ventas) e inventario, donde la integridad de datos (ACID) no es opcional. Elegir una base de datos NoSQL (como MongoDB) sería un error técnico grave. Más importante aún, PostgreSQL es la base de datos preferida para la ciencia de datos. Su capacidad para manejar consultas analíticas complejas y sus tipos de datos avanzados (como JSONB) la convierten en la fuente de datos perfecta y robusta para alimentar el modelo predictivo de demanda.