

Documento de requerimientos de software

Zohar

Fecha:18/11/2024

Contenido

Historial de Versiones 3

Información del Proyecto 3

Tabla Comparativa de Tecnologías4

Justificación de la Selección de Tecnologías para tu Proyecto6

Historial de Versiones

Fecha	Versión	Autor	Organización	Descripción
18/11/2024	V1.0	Kevin Guapi David Piray	Industrias ARBEN	Tabla Comparativa de las diferentes herramientas para el proyecto.

Información del Proyecto

Empresa / Organización	Industrias ARBEN
Proyecto	Zohar
Fecha de preparación	27/10/2024
Cliente	Industrias ARBEN
Patrocinador principal	David Piray y Kevin Guapi
Gerente / Líder de Proyecto	David Piray y Kevin Guapi
Gerente / Líder de Análisis de negocio y requerimientos	David Piray y Kevin Guapi

Tabla Comparativa de Tecnologías

Criterio	Flutter (Dart)	React (JavaScript)	Angular (TypeScript)	Swift (iOS) / Kotlin (Android)	Node.js con Express.js	Django (Python)	Firebase	MySQL
Plataforma de Desarrollo	Web y Móvil (Multiplataforma)	Web (React Native para Móvil)	Web (Angular Mobile para Móvil)	Solo Móvil (iOS/Android)	Backend (APIs REST)	Backend (Web Framework)	Backend (BaaS)	Base de Datos (Relacional)
Curva de Aprendizaje	Moderada	Baja	Alta	Alta	Baja	Moderada	Baja	Moderada
Productividad	Alta (Código Único para Web y Móvil)	Alta (Solo Web, requiere React Native para Móvil)	Media (Requiere más configuración)	Baja (Aplicaciones nativas separadas)	Alta (Simplicidad y Rapidez en APIs)	Media (Más configuración para APIs)	Muy Alta (Desarrollo Rápido)	Media (Mantenimiento Requiere SQL)
Rendimiento	Alto (Móvil), Medio (Web)	Alto (Web)	Medio (Web)	Muy Alto (Nativo)	Alto (Escalable con Node.js)	Medio (Web, No tan Escalable)	Alto (Tiempo Real)	Alto
Escalabilidad	Media-Alta	Alta	Alta	Alta	Alta (Microservicios)	Media	Media (Uso Intensivo Costoso)	Alta
Facilidad de Integración	Alta (con Firebase, APIs)	Muy Alta	Alta	Baja	Muy Alta (APIs RESTful)	Media	Muy Alta (Servicios Integrados)	Media

Compatibilidad Multiplataforma	Sí	No (Solo Web, requiere React Native)	No (Solo Web, limitado en Móvil)	No	No	No	No (Frontend Separado Necesario)	No
Capacidades Nativas	Sí (Geolocalización, Notificaciones)	Limitadas	Limitadas	Muy Alta (Nativas)	No (Necesita Plugins)	No	Sí (Push, Auth, DB Tiempo Real)	No
Soporte Comunidad	Amplia y Creciente	Muy Amplia	Muy Amplia	Media	Muy Amplia	Amplia	Amplia	Amplia
Costos de Implementación	Bajo (Gratis con Plugins)	Bajo	Bajo	Alto (Licencias y Desarrollo Separado)	Bajo (Gratis)	Bajo	Variable (Pago por Uso)	Bajo (Gratis, Mantenimiento o Costoso)
Recomendado Para	Apps Multiplataforma (Web y Móvil)	Apps Web Interactivas	Apps Web Complejas	Apps Nativas con Alta Performance	APIs REST y Microservicios	Aplicaciones Web Robusta	Aplicaciones Rápidas con Auth y DB	Bases de Datos Relacionales

Justificación de la Selección de Tecnologías para tu Proyecto

Con base en la tabla comparativa anterior, las tecnologías seleccionadas (**Flutter**, **Node.js con Express.js**, **Firebase**) se destacan por sus beneficios específicos que las hacen adecuadas para tu proyecto, especialmente teniendo en cuenta un equipo pequeño y la necesidad de desarrollar tanto para web como para móvil.

1. Flutter (Dart) para Desarrollo Multiplataforma

- **Ventajas:**
 - **Código Único para Web y Móvil:** Con Flutter, puedes mantener una sola base de código para ambas plataformas (Android, iOS y Web), lo que reduce significativamente el esfuerzo de desarrollo. Alternativas como React requieren aprender React Native para aplicaciones móviles, mientras que Angular no está optimizado para desarrollos móviles.
 - **Capacidades Nativas y Flexibilidad:** Flutter permite acceder a funcionalidades nativas como geolocalización y notificaciones push sin necesidad de desarrollos adicionales, lo que agiliza el proceso.
 - **Web y Móvil:** Considerando que necesitas transportar funcionalidades de la web a móvil y mantenerlas sincronizadas, Flutter es ideal para evitar duplicación de esfuerzo y reducir la complejidad.

2. Node.js con Express.js para Backend (APIs REST)

- **Ventajas:**
 - **Simplicidad y Rapidez:** Node.js con Express.js permite crear APIs de forma rápida y eficiente, ideal para un equipo pequeño con poca experiencia en backend. Comparado con NestJS o Django, que son más complejos y requieren mayor curva de aprendizaje, Express.js proporciona la simplicidad necesaria para tu caso.
 - **Escalabilidad:** Si bien Django es excelente para aplicaciones web robustas, Node.js sobresale en arquitecturas orientadas a eventos y microservicios, lo que lo hace adecuado para un sistema que podría escalar en el futuro.
 - **Eficiencia en la Comunicación con Frontend:** Dado que tu frontend será desarrollado en Flutter, usar Node.js facilita la comunicación a través de APIs REST, asegurando un rendimiento óptimo.

3. Firebase para Base de Datos y Autenticación

- **Ventajas:**
 - **Desarrollo Rápido y Simplicidad:** Firebase ofrece servicios integrados como autenticación, bases de datos en tiempo real y notificaciones push que permiten una implementación más rápida comparado con configuraciones manuales en bases de datos tradicionales como MySQL o PostgreSQL.
 - **Enfocado en Aplicaciones en Tiempo Real:** Firebase es ideal para gestionar datos en tiempo real y autenticación de usuarios, lo que es beneficioso para tu sistema de asignación de clientes y seguimiento de distribuidores.

- **Ahorro en Infraestructura:** Firebase elimina la necesidad de configurar y gestionar servidores dedicados, lo que es ideal si buscas reducir costos y simplificar la administración de la infraestructura.