

# Selección de plataformas

INTEGRANTES: Tomas Beltran, Ivo Barbato, Martin Colussi, Facundo Herrera, Felipe Lemir

## Frontend

- **GitHub Pages:**
  - **Límites del plan gratuito:** Soporta solo sitios estáticos.
  - **Proceso de despliegue:** Repositorio de GitHub con una rama especial ( `main` o `gh-pages` ).
  - **Integración CI/CD:** Fácil de integrar con GitHub Actions.
  - **Características especiales:** Buena opción para documentación y proyectos estáticos.
  - **Limitaciones técnicas:** No permite backend directo ni bases de datos.
- **Netlify:**
  - **Límites del plan gratuito:** 100 GB de ancho de banda, despliegues continuos.
  - **Proceso de despliegue:** Vía GitHub, GitLab o Bitbucket.
  - **Integración CI/CD:** Integración nativa con sistemas de control de versiones.
  - **Características especiales:** Funciones Lambda (sin servidor).
  - **Limitaciones técnicas:** Funciones sin servidor limitadas en el plan gratuito.
- **Vercel:**
  - **Límites del plan gratuito:** 100 GB de ancho de banda mensual.
  - **Proceso de despliegue:** Soporte para Next.js, GitHub, GitLab.
  - **Integración CI/CD:** Despliegue automático en `push`.
  - **Características especiales:** Despliegues rápidos, soporte directo para Next.js.
  - **Limitaciones técnicas:** Restricciones en ancho de banda y funciones de servidor en el plan gratuito.
- **Firebase Hosting:**
  - **Límites del plan gratuito:** 10 GB de almacenamiento y 360 MB diarios de ancho de banda.
  - **Proceso de despliegue:** CLI de Firebase o integración continua.
  - **Integración CI/CD:** Se puede integrar con GitHub Actions.
  - **Características especiales:** Permite usar Firebase Authentication, Firestore.
  - **Limitaciones técnicas:** Ancho de banda limitado en el plan gratuito.
- **Surge.sh:**
  - **Límites del plan gratuito:** Solo para sitios estáticos, sin funciones de backend.
  - **Proceso de despliegue:** CLI sencilla, ideal para prototipos.
  - **Integración CI/CD:** No tiene soporte nativo de CI/CD.
  - **Características especiales:** Ideal para MVP de sitios estáticos.
  - **Limitaciones técnicas:** Carece de funcionalidad avanzada y CI/CD nativo.

## Backend

- **Railway:**
  - **Recursos gratuitos disponibles:** 500 horas de uso mensual.
  - **Lenguajes soportados:** Node.js, Python, Java, etc.
  - **Proceso de configuración:** GUI sencilla, o con Docker y CLI.
  - **Limitaciones de uso:** Limitado a 500 horas al mes en plan gratuito.

- **Tiempo de inactividad permitido:** Puede pausar automáticamente cuando se agota el límite.
- **Render:**
  - **Recursos gratuitos disponibles:** Servicios de web gratis con un límite de 750 horas.
  - **Lenguajes soportados:** Node.js, Python, Ruby, etc.
  - **Proceso de configuración:** CLI de Render y panel.
  - **Limitaciones de uso:** Ancho de banda y almacenamiento limitado.
  - **Tiempo de inactividad permitido:** Puede entrar en modo de suspensión.
- **Fly.io:**
  - **Recursos gratuitos disponibles:** 3 vCPU y 256 MB de RAM.
  - **Lenguajes soportados:** Multilenguaje con soporte para Docker.
  - **Proceso de configuración:** Panel intuitivo y CLI.
  - **Limitaciones de uso:** Escalado limitado en el plan gratuito.
  - **Tiempo de inactividad permitido:** Sin suspensión automática.
- **Heroku:**
  - **Recursos gratuitos disponibles:** 550-1000 horas según el uso de la cuenta.
  - **Lenguajes soportados:** Node.js, Ruby, Python, PHP.
  - **Proceso de configuración:** CLI Heroku y Git.
  - **Limitaciones de uso:** Suspensión cada 30 minutos sin actividad.
  - **Tiempo de inactividad permitido:** Se suspende cuando está inactivo.
- **Deta.sh:**
  - **Recursos gratuitos disponibles:** Sin límite de horas.
  - **Lenguajes soportados:** Python y Node.js.
  - **Proceso de configuración:** CLI de Deta y panel de administración.
  - **Limitaciones de uso:** Es adecuado para aplicaciones pequeñas.
  - **Tiempo de inactividad permitido:** Sin suspensión automática.

## Base de datos

- **MongoDB Atlas:**
  - **Capacidad de almacenamiento gratuito:** 512 MB en el plan gratuito.
  - **Límites de conexiones:** Limitado en conexiones simultáneas.
  - **Características de backup:** Snapshots básicos.
  - **Herramientas de administración:** Panel de administración completo.
  - **Limitaciones de consultas:** Puede haber límites en el rendimiento.
- **Supabase:**
  - **Capacidad de almacenamiento gratuito:** 500 MB.
  - **Límites de conexiones:** 100 conexiones simultáneas.
  - **Características de backup:** Backups diarios.
  - **Herramientas de administración:** Interfaz amigable.
  - **Limitaciones de consultas:** Adecuado para MVP.
- **PlanetScale:**
  - **Capacidad de almacenamiento gratuito:** 5 GB.
  - **Límites de conexiones:** Limitado en el plan gratuito.

- **Características de backup:** Snapshots manuales.
- **Herramientas de administración:** UI para administración y escalado.
- **Limitaciones de consultas:** Pensado para cargas pequeñas a medianas.
- **Firebase Realtime Database:**
  - **Capacidad de almacenamiento gratuito:** 1 GB.
  - **Límites de conexiones:** Limitado a 100 simultáneas.
  - **Características de backup:** Sin backups automáticos.
  - **Herramientas de administración:** Muy amigable y con buena integración.
  - **Limitaciones de consultas:** Coste elevado en planes superiores.
- **ElephantSQL:**
  - **Capacidad de almacenamiento gratuito:** 20 MB en el plan gratuito.
  - **Límites de conexiones:** Limitado a conexiones básicas.
  - **Características de backup:** Copias de seguridad automáticas.
  - **Herramientas de administración:** UI sencilla y confiable.
  - **Limitaciones de consultas:** Ideal para datos no críticos.

## Matriz de evaluación

1. Facilidad de uso (1-5)
2. Documentación disponible (1-5)
3. Límites del plan gratuito (1-5)
4. Integración con otras herramientas (1-5)
5. Confiabilidad del servicio (1-5)
6. Proceso de despliegue (1-5)
7. Comunidad y soporte (1-5)

Siendo (5) el mejor

### Frontend

Plataforma	Facilidad de Uso	Documentación	Plan Gratuito	Integración CI/CD	Confiabilidad	Despliegue	Comunicación
<b>Netlify</b>	5	5	4	5	5	5	4
<b>Vercel</b>	5	5	4	5	5	5	5
<b>Firebase Hosting</b>	4	5	3	4	4	4	4

### Backend

Plataforma	Facilidad de Uso	Documentación	Plan Gratuito	Integración CI/CD	Confiabilidad	Despliegue	Comunicación
Railway	4	4	4	5	4	4	4
Render	4	4	4	5	4	4	4

### Base de datos

Plataforma	Facilidad de Uso	Documentación	Plan Gratuito	Integración CI/CD	Confiabilidad	Despliegue	Comunicación
<b>Firebase Realtime Database</b>	4	5	4	4	4	4	5

## Análisis de Compatibilidad

### 1. Requerimiento técnicos

- **Lenguajes de programación utilizados:**
  - **Java:** Backend para la lógica de negocio y servicios.
  - **JavaScript (con React o Vue.js):** Frontend para la creación de la interfaz de usuario.
  - **HTML/CSS:** Estructuración y diseño de las vistas.
- **Frameworks:**
  - **Spring Boot:** Para el desarrollo del backend y API REST.
  - **Hibernate:** ORM para la interacción con la base de datos.
  - **React.js o Vue.js:** Para la construcción de la interfaz minimalista.
- **Bases de datos:**
  - **MySQL:** Base de datos relacional para almacenar datos de usuarios, productos, pedidos, entre otros.
- **APIs y servicios externos:**
  - **Google reCAPTCHA:** Para la autenticación y validación de usuarios.
  - **API de servicios de envío:** Para la selección de métodos de envío.
  - **Email Verification API:** Verificación de correos electrónicos durante el registro.
  - **SMS Verification Service:** Para la validación de números telefónicos.
- **Requisitos de almacenamiento:**
  - Capacidad suficiente para almacenar datos de usuarios, productos, carritos de compras y pedidos.
  - Sistemas de almacenamiento seguros para información confidencial y verificada.

### 2. Restricciones del proyecto:

- **Presupuesto:**
  - \$15,000 a \$35,000 USD.
- **Tiempo de desarrollo:**
  - 15 semanas.
- **Conocimientos técnicos del equipo:**
  - El equipo tiene experiencia en **Java**, **bases de datos**, y desarrollo web con **React.js** o **Vue.js**.
- **Requisitos de escalabilidad:**
  - El sistema debe ser capaz de **escalar horizontalmente** en caso de un aumento en el tráfico o de la base de datos a medida que crece el número de usuarios y pedidos.
- **Requisitos de seguridad:**
  - Autenticación de usuarios con verificación de datos personales.
  - **Protección contra ataques de fuerza bruta** y seguridad de datos sensibles mediante cifrado.

## Justificación de las herramientas finales elegidas

- **Frontend:** Vercel y Netlify y Firebase Hosting (Es mucho mejor para usar APIs externas integración sencilla) destacan por su facilidad de despliegue, integración CI/CD, y capacidad de escalado, lo cual es ideal para un proyecto frontend de crecimiento rápido.
- **Backend:** Render y Railway fueron seleccionados por sus recursos gratuitos son plataformas basadas en el cloud y facilitan la configuración. Ambos permiten escalado en planes gratuitos, lo cual es esencial para soportar el MVP.
- **Bases de Datos:** Firebase realtime database es ideal debido que si necesitas actualizaciones en tiempo real es facil integrar con otras herramientas de firebase rapido de implementar, pero la desventaja es que tiene limitaciones en consultas complejas y estructuracion de datos

## Prueba de Concepto

### 1. Vercel (Frontend) + Render (Backend) + Firebase Realtime Database

- **Razones de selección:** Vercel proporciona un despliegue rápido y eficiente para el frontend, lo cual es ideal para mantener tiempos de respuesta bajos. Render, por su parte, soporta backend en Java y permite configuraciones escalables, esenciales para la lógica de negocio. Finalmente, Firebase Realtime Database es una opción conveniente para datos en tiempo real, permitiendo sincronización instantánea y sencilla con el frontend.
- **Plan de pruebas:** Realizaré un despliegue de prueba básico, seguido de pruebas de integración para verificar la comunicación entre Firebase y el frontend. También mediré los tiempos de respuesta de las operaciones CRUD básicas en Firebase. El proceso de actualización se evaluará a través de cambios en Firebase, Render y Vercel, asegurándome de que las actualizaciones de datos se reflejen correctamente en el frontend y el backend.

### 2. Netlify (Frontend) + Railway (Backend) + Supabase (Base de datos relacional)

- **Razones de selección:** Netlify es fácil de configurar y permite el despliegue de funciones sin servidor, lo que resulta útil para las operaciones simples en el frontend. Railway, con su soporte para múltiples lenguajes, ofrece una configuración rápida y recursos escalables en Java para el backend. Finalmente, Supabase proporciona una base de datos SQL que facilita consultas más complejas y gestión eficiente de datos relacionales.
- **Plan de pruebas:** En esta combinación, iniciaré con un despliegue básico y luego realizaré pruebas de integración entre el frontend y el backend. Evaluaré el tiempo de respuesta de las consultas SQL en Supabase y su capacidad para gestionar datos relacionales en operaciones CRUD. Además, probaré el despliegue automático de Netlify y Railway para validar la estabilidad y consistencia en las actualizaciones de ambas plataformas.

## Pasos para la Prueba de Concepto

1. **Despliegue de prueba básico:** Realizaré el despliegue en ambas combinaciones y verificaré que todas las conexiones, como API, bases de datos y CI/CD, funcionen correctamente.
2. **Pruebas de integración:** Ejecutaré funciones clave (como autenticación de usuario y operaciones CRUD) para asegurar que los datos se muestren y actualicen correctamente en el frontend y el backend.
3. **Medición de tiempos de respuesta:** Utilizaré herramientas como Postman y las herramientas de Network en el navegador para medir el tiempo de respuesta de las API y validar la eficiencia de cada combinación.
4. **Evaluación del proceso de actualización:** Simularé actualizaciones en el sistema, modificando endpoints de API o estructuras de datos en la base de datos, y observaré la respuesta del sistema completo a estos cambios.
5. **Documentación de problemas:** Registraré cualquier inconsistencia o problema identificado en cada fase de la prueba de concepto, especialmente en la integración de datos y el despliegue de actualizaciones.