

Informe de Despliegue de Aplicación Front-End en Vercel

1. Introducción

Este informe documenta el proceso de selección, configuración y despliegue de una aplicación front-end en la plataforma Vercel. Se detallan las razones para elegir esta plataforma, sus características principales, la configuración de la capa gratuita, la implementación de GitHub Actions para el despliegue continuo y la URL de la aplicación desplegada.

2. Plataforma Escogida: Vercel

2.1. Características Principales de Vercel

Vercel es una plataforma especializada en el despliegue de aplicaciones front-end, con un enfoque en proyectos desarrollados con frameworks como React, Vue y Next.js. Algunas de sus principales características son:

- **Despliegue Automático:** Integración directa con GitHub, GitLab y Bitbucket para desplegar la aplicación al hacer un "push".
- **CDN Global:** Ofrece una red de distribución de contenido (CDN) para mejorar la velocidad de carga.
- **SSL Automático:** Proporciona certificados SSL gratuitos.
- **Vista Previa de Ramas:** Permite generar versiones previas de la aplicación en ramas antes de publicarlas en producción.
- **Funciones Serverless:** Soporte para ejecutar funciones en el servidor sin necesidad de configuraciones complejas.

2.2. Comparación entre la Capa Gratuita y la de Pago

Característica	Capa Gratuita	Capa de Pago (Pro/Enterprise)
Dominios personalizados	Sí (con vercel.app)	Sí
Ancho de banda	Ilimitado	Ilimitado
Funciones Serverless	Límite de 10s de ejecución	Mayor tiempo de ejecución
Almacenamiento de logs	1 mes	Ilimitado
Soporte técnico	Básico	Prioritario
Implementaciones simultáneas	Limitadas	Sin límites

Cuándo Pasar a la Capa de Pago:

- Cuando se requieran tiempos de ejecución mayores para funciones serverless.
- Si se necesita soporte técnico prioritario.
- Para mayor almacenamiento de logs y análisis de rendimiento avanzado.

3. Acceso a la Capa Gratuita

Para acceder a la capa gratuita de Vercel, se siguieron estos pasos:

1. Se creó una cuenta en [Vercel](#) utilizando GitHub.
2. Se vinculó un repositorio de GitHub a un nuevo proyecto en Vercel.
3. Se configuró el proyecto y se realizó el primer despliegue.
4. Se verificó que la URL generada estuviera operativa.

(Capturas de pantalla al final del informe)

4. Despliegue de la Aplicación

La aplicación seleccionada para el despliegue fue el front-end del examen del primer trimestre. Para garantizar un despliegue continuo y automatizado, se configuró un flujo de trabajo en **GitHub Actions**.

4.1. Configuración de GitHub Actions

Se creó un archivo de workflow en `.github/workflows/recu.yml` con los siguientes pasos:

Se creó un archivo de GitHub Actions

El siguiente código representa el flujo de trabajo configurado:

```
name: Exam-recu-DAW
on: push
jobs:
  nombres:
    runs-on: ubuntu-latest
    steps:
      - name: 🍷 Nombre de usuario
        run: echo ${{ github.actor }}
      - name: 📁 Nombre repositorio
        run: echo ${{ github.repository }}
      - name: 🌿 Nombre rama
        run: echo ${{ github.ref_name }}

  deploy:
    runs-on: ubuntu-latest
    steps:
      - name: 📦 Checkout del repositorio
        uses: actions/checkout@v4
```

- name: 🛠 Construir la imagen Docker
run: docker build -t calculadora .
- name: 🚀 Instalar Vercel CLI
run: npm install -g vercel
- name: 📦 Desplegar en Vercel
run: vercel --prod --yes --token \${{ secrets.VERCEL_TOKEN }}

4.2. Configuración de Secretos en GitHub

Para que la acción pueda autenticar en Vercel, se agregó el secreto en GitHub:

- **VERCEL_TOKEN:** Token generado en Vercel.

5. Resultado Final y URL de la Aplicación

Una vez finalizado el proceso, la aplicación se desplegó correctamente en Vercel. La URL de la aplicación es:

<https://examen-front-alpha.vercel.app/>

el repositorio es:

<https://github.com/zono32/examen-front.git>

el ID del último commit:

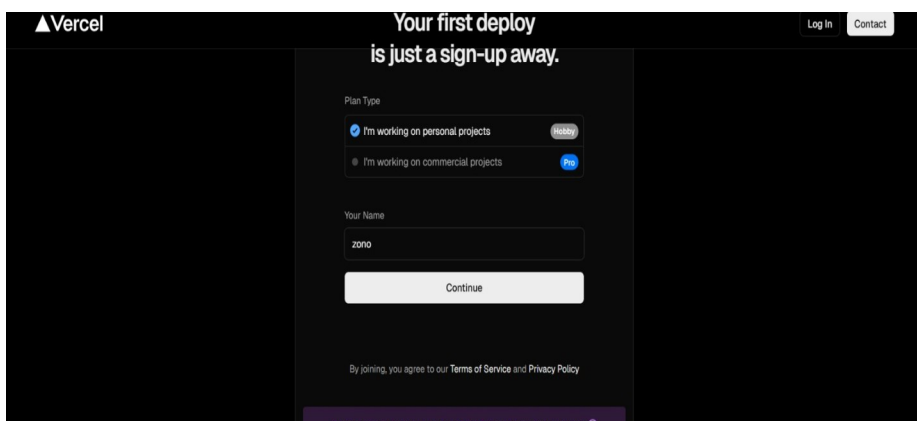
db35a4f0aad20e6309ec7a6c9a18b2951545d240

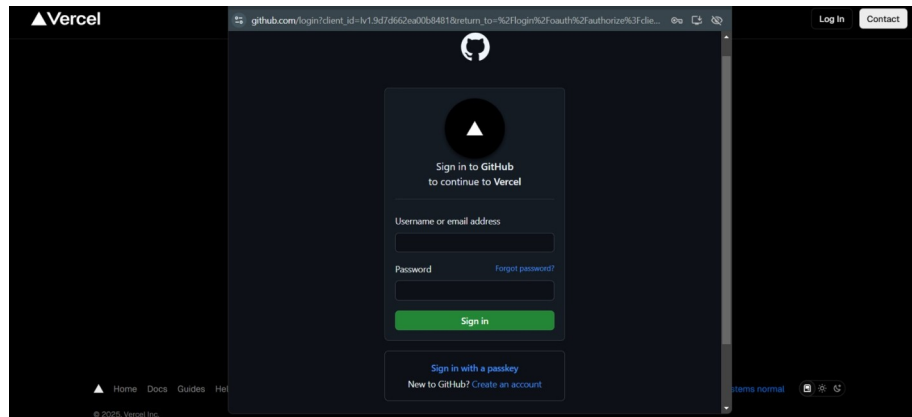
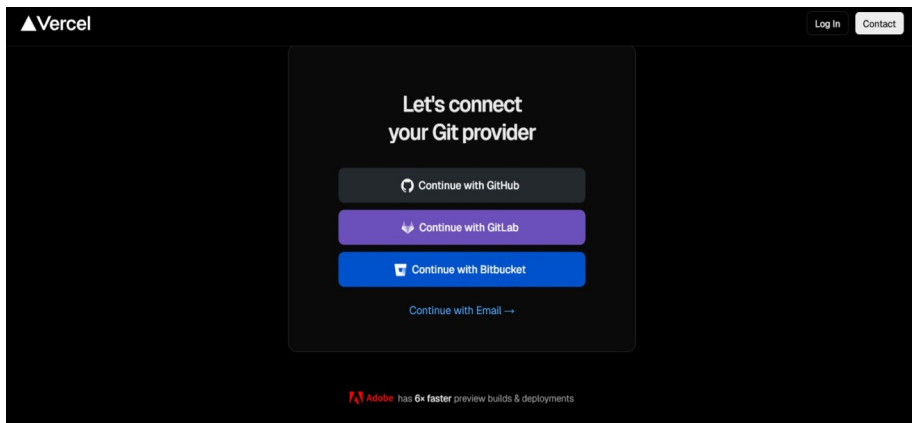
6. Conclusiones

- Vercel es una plataforma robusta y eficiente para el despliegue de aplicaciones front-end, con una excelente integración con GitHub y un entorno serverless optimizado.
- La capa gratuita es suficiente para proyectos personales o de pequeña escala, pero en entornos profesionales podría ser necesario un plan de pago.
- GitHub Actions permite automatizar el proceso de despliegue, asegurando que cada cambio en el código se refleje en producción sin intervención manual.

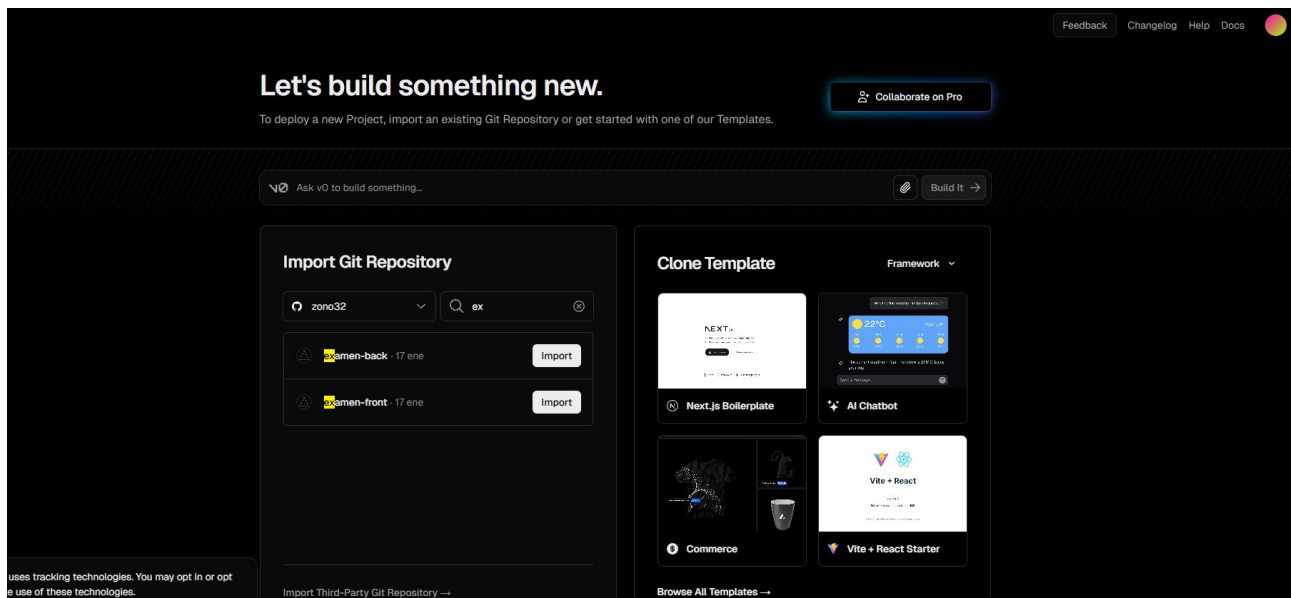
7. Capturas de pantalla

Crear cuenta en vercel: (creé la cuenta de Vercel con la que ya tengo en github)





Una vez creada la cuenta en Vercel importas un proyecto de github y haces el despliegue.



New Project

Importing from GitHub
zono32/examen-front ↗ main

Choose where you want to create the project and give it a name.

Vercel Team

Project Name

kinso's projects Hobby ↕

examen-front

Framework Preset

Other ▾

Root Directory

./

Edit

> Build and Output Settings

> Environment Variables

Deploy

kinso's projects Hobby ↕ / examen-front ↕

Feedback Changelog Help Docs 🔔

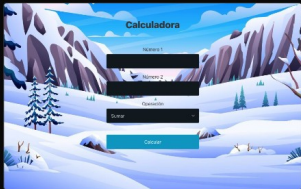
Project Deployments Analytics Speed Insights Logs Observability ⚙️ Firewall Storage Settings

examen-front

Repository Usage Domains Visit ▾

Production Deployment

Build Logs Runtime Logs Instant Rollback



Deployment

examen-front-rmz55vod-kinso's-projects-0bcb802d.vercel.app

Domains

examen-front-beryl.vercel.app ↗ +2

Status Created ↕

● Ready 2m ago by zono32 🧑

Source

↗ main

↩ 86ca1f4 comit recu

> Deployment Configuration

⊗ Fluid Compute ● Deployment Protection ⊗ Skew Protection

To update your Production Deployment, push to the "main" branch.

Learn More

Firewall 24h

Analytics