# Tarea 1: Continuous deployment (CD)

## Elección de la Plataforma

### 1.1 Comparativa de las opciones más destacadas y sencillas:

|  |  |  |  |
| --- | --- | --- | --- |
| **Plataforma** | **Recursos Gratuitos** | **Mejor Uso** | **Capa de Pago Necesaria** |
| **Render** | 512 MB RAM, 0.5 vCPU, 750 MB disk | Backend y servicios Dockerizados | Mayor RAM/CPU o uptime |
| **Vercel** | 100 GB ancho de banda, serverless | Frontend estático o Next.js | Más builds o tráfico |
| **Netlify** | 125k invocaciones, 100 GB ancho | Frontend estático y serverless ligero. No soporta docker | Más tráfico/invocaciones |
| **Heroku** | Sin plan gratuito (mínimo 5$) | Prototipos backend o front simple, dockerizados | Actividad constante |
| **Railway** | $5 créditos (~500 horas) | Servicios Dockerizados y bases de datos | Más créditos |
| **Fly.io** | Sin plan grauito pay per use | Backend avanzado y distribuido | Más RAM/CPU/storage |

### 1.2. Elección Final:

La plataforma elegida es **RENDER** por ser la única que tiene un plan gratuito a coste 0 y que además tiene soporte para desplegar apps dockerizadas desde dockerhub. El plan más generoso es el de VERCEL, pero sin embargo este hace el despliegue mediante la clonación del repositorio de github, similar al github action checkout.

He decidido dockerizar y desplegar una pequeña app que hice en verano para aprender a usar el framework next.js. Utiliza la librería fakers.js.

**Repositorio de github:** <https://github.com/bmascat/data-generator>

**Último commit:** feat(workflows): add contidion to deploy for events different of pull request ( c0e2bfee50c528a8f9a62dbbb3251830776ce7ca )

## 2. Configuración en Render

1. **Registro en Render**:
   * Crea una cuenta en [Render](https://render.com) si aún no tienes una.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. **Nuevo Servicio Web**:
   * En tu dashboard de Render, haz clic en **+ Add new > Web Service**.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* + Interfaz de usuario gráfica, Aplicación

    Descripción generada automáticamenteSelecciona la opción **Existing Image** y añadimos el link a nuestra imagen en el registro de Docker Hub (por ejemplo, docker.io/<dockerhub-username>/<repository-name>:latest).

1. **Conectar con Docker Hub**:
   * Configura los siguientes parámetros:
     + **Instance Type**: Selecciona la instancia gratuita.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. **Despliegue**:
   * Render hará el pull de la imagen y desplegará automáticamente el servicio.
   * Una vez finalizado, Render generará una URL pública para tu aplicación.
   * En mi caso es: <https://data-generator-4ghg.onrender.com/>

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* + Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

    Descripción generada automáticamenteEn render el primer despliegue es manual, para automatizar futuros despligues lo hacemos a través de un webhook configurado en el .yaml del github action.
  + Configuramos un secret en el repo de github con la url proporcionada por RENDER y creamos el job en el .yaml

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

