



Hola Bienvenido!


Registra tu asistencia

<https://forms.gle/6hfdT5U5v5fF34RQ8>



Deploy / Despliegue en la nube

Conocer de manera teórica los conceptos que implica subir un proyecto a la nube y analizar sus ventajas y desventajas frente a diferentes tecnologías, para posteriormente implementar de forma práctica el despliegue (deploy) en servicios gratuitos.



Deploy (despliegue) = publicar una aplicación en internet para que otros puedan usarla.

Implica subir el proyecto a un **servidor remoto en la nube**.

Permite que el **backend (Express)** y el **frontend (Vue)** estén disponibles 24/7 mediante una URL.



En el proyecto visto la semana pasada tenemos la siguiente estructura:

Backend (API con Node + Express) → maneja peticiones, lógica y acceso a datos.

Frontend (Vue.js) → interfaz visual, usa **Axios** para consumir la API.

Para desplegar::

- **Unir ambos** en un solo servidor (fullstack).
- O **separarlos** (una URL para el API y otra para el frontend).

Formas de deploy (gratis)

Servicio	Ideal para	Características
Render	Backend (Express)	Fácil de usar, gratuita con “sleep mode” (se apaga por inactividad).
Vercel	Frontend (Vue, React)	Muy rápida, gratuita, conecta con GitHub.
Netlify	Frontend (Vue)	Ideal para sitios SPA, integra automáticamente desde GitHub.
Railway	Backend + DB	Gratuita con límite de horas mensuales.
GitHub Pages	Frontend estático	Solo para proyectos sin backend dinámico.

Formas de deploy (vía pago)

Servicio	Ventajas	Ejemplo de uso
AWS (Amazon Web Services)	Escalable, soporte para Express, MongoDB, etc.	Producción empresarial.
Google Cloud Platform (GCP)	Integra IA, bases de datos, y despliegue automático.	APIs y apps complejas.
Microsoft Azure	Soporte empresarial, monitoreo, y CI/CD.	Grandes organizaciones.
DigitalOcean / Linode	VPS económico, control total.	Desarrolladores avanzados.



Pasos prácticos de deploy (ejemplo con Render + Vercel)

Backend (Express) en Render

1. Crear una cuenta en [Render.com](https://render.com).
2. Subir código a **GitHub**.
3. En Render → “**New Web Service**”.
4. Conectar repositorio.



Definir:

- Build Command: `npm install`
- Start Command: `node server.js`

Esperar a que Render genere una URL como:


– <https://mi-api.onrender.com/api/...>

Frontend (Vue) en Vercel

1. Crear una cuenta en [Vercel.com](https://vercel.com).
2. Importar proyecto Vue desde GitHub.
3. En configuración:
 - Build Command: `npm run build`
 - Output Directory: `dist`
4. Listo → Aparecera una URL como:
`https://mi-app.vercel.app`

Importante: en tu código Vue, cambia las rutas de Axios para apuntar a la URL real de tu backend Render.

Flujo final en producción

1.  Usuario entra a **tu dominio (Vercel)**.
2. Vue envía peticiones **Axios** a tu API en **Render**.
3. El servidor Express responde con datos reales.
4. Todo funciona online, sin depender de tu computadora.



Conclusión

- Hacer deploy es el paso final del desarrollo profesional.
- Las **opciones gratuitas** son perfectas para prácticas, portafolios y proyectos académicos.
- Las **opciones de pago** ofrecen estabilidad, seguridad y escalabilidad para uso empresarial.