

Buenas prácticas en desarrollo web y Deploy

Autor: L. Falcon

Buenas prácticas en desarrollo web

Rutas en HTML:

Siempre usar '/' para las direcciones y rutas relativas, para evitar problemas de carga de recursos.

Ejemplo:

```
<script src='/js/app.js'></script>
```

Logs de depuración:

No dejar console.log en el código final, pueden mostrar información sensible y ensuciar la consola.

Para borrarlos rápido: usar buscar/replace en el editor (search -> console.log).

Control de versiones (Git/GitHub):

No trabajar directamente en main, salvo para releases o merges finales.

Crear ramas (branch) para cada funcionalidad o fix, y luego hacer pull request hacia main.

Deploy

Pasar tu aplicación desde el entorno de desarrollo (tu PC) hasta el entorno de producción (un servidor en internet) para que los usuarios la puedan usar.

****Flujo típico 'Proceso de CI/CD'****

Etapa	Acciones
Build	pnpm install pnpm build
Test	pnpm test (Opcional) Análisis estático con eslint o prettier.
Deploy	Hosting automático (Vercel, Netlify). Hosting personalizado (AWS, DigitalOcean, VPS, Docker + Nginx). Backend: reinicio de procesos (pm2 restart app).

****Opciones de Deploy****

AWS:

Súper flexible (EC2, S3, RDS, Lambda, etc.). Ideal para proyectos grandes o cuando necesitas control total. Más configuración manual (seguridad, permisos, balanceadores, etc.).

Vercel:

Ideal para proyectos front-end y funciones serverless pequeñas. Automático: `pnpm install`
`pnpm build` Deploy directo al conectar GitHub/GitLab.

Netlify:

Muy similar a Vercel, fuerte en static sites y JAMStack. Tiene 'functions' como serverless.

Render / Railway:

Intermedio entre Vercel y AWS. Fácil de usar, pero soporta front + backend en una misma plataforma.