

COMPARATIVA: ALWAYSDATA VS RENDER

Característica	Alwaysdata	Render
Tipo de hosting	Hosting clásico compartido	Plataforma moderna de despliegue continuo
Panel de control	Gráfico tipo cPanel (interfaz web)	Interfaz moderna basada en despliegues Git
Subida de archivos	Vía FTP o panel web	Vía GitHub (push automático)
Soporte web estática	✓ HTML, CSS, JS	✓ HTML, CSS, JS
Soporte backend	✓ PHP, Python, Node.js, etc. (manual)	✓ Node.js, Python, Ruby, etc. (con build)
Bases de datos	✓ MySQL, PostgreSQL, SQLite	✓ PostgreSQL (gratuita)
Deploy automático desde Git	✗ (no automático, tú subes los archivos)	✓ Sí, cada push actualiza el sitio
Dominio personalizado	✓ Sí	✓ Sí
Certificado SSL	✓ Automático	✓ Automático
Usuario ideal	Usuarios que vienen de hosting tradicional	Desarrolladores modernos (Node, Git, API REST)
Curva de aprendizaje	Muy baja (ideal para principiantes)	Media (requiere uso de Git y entorno moderno)
Plan gratuito	100 MB disco + 1 sitio	Web estática y backend gratis con límites

¿CUÁNDO USAR ALWAYSDATA?

Usa Alwaysdata si:

- Tus alumnos **empiezan con HTML/CSS/JS** y necesitas un entorno web rápido y fácil de entender
- Quieres que los alumnos **suban archivos manualmente**, sin usar Git
- Trabajas con **PHP** o **formularios sencillos**

- Quieres simular un **hosting tradicional** (tipo cPanel)

🎓 Ideal para las **primeras prácticas web** (estáticas o con backend sencillo)

¿CUÁNDO USAR RENDER?

Usa Render si:

- Tus alumnos trabajan con **Node.js, Python, Flask, Express, PostgreSQL**, etc.
- Quieres enseñar buenas prácticas modernas: **repositorio Git → despliegue automático**
- Quieres integrar **frontend + backend + base de datos**
- Quieres simular un entorno de **despliegue profesional** (tipo **Vercel, Heroku**, etc.)

🎓 Ideal para **proyectos más avanzados** y fullstack

¿Y PARA EL AULA?

Nivel de práctica	Plataforma recomendada
HTML/CSS básico	● Alwaysdata
JS con DOM	● Alwaysdata
Formularios y PHP	● Alwaysdata
Node.js API REST	● Render
Web con PostgreSQL	● Render
Despliegue continuo	● Render
GitHub + integración	● Render