

Protocolos de APIs

REST (Representational State Transfer)

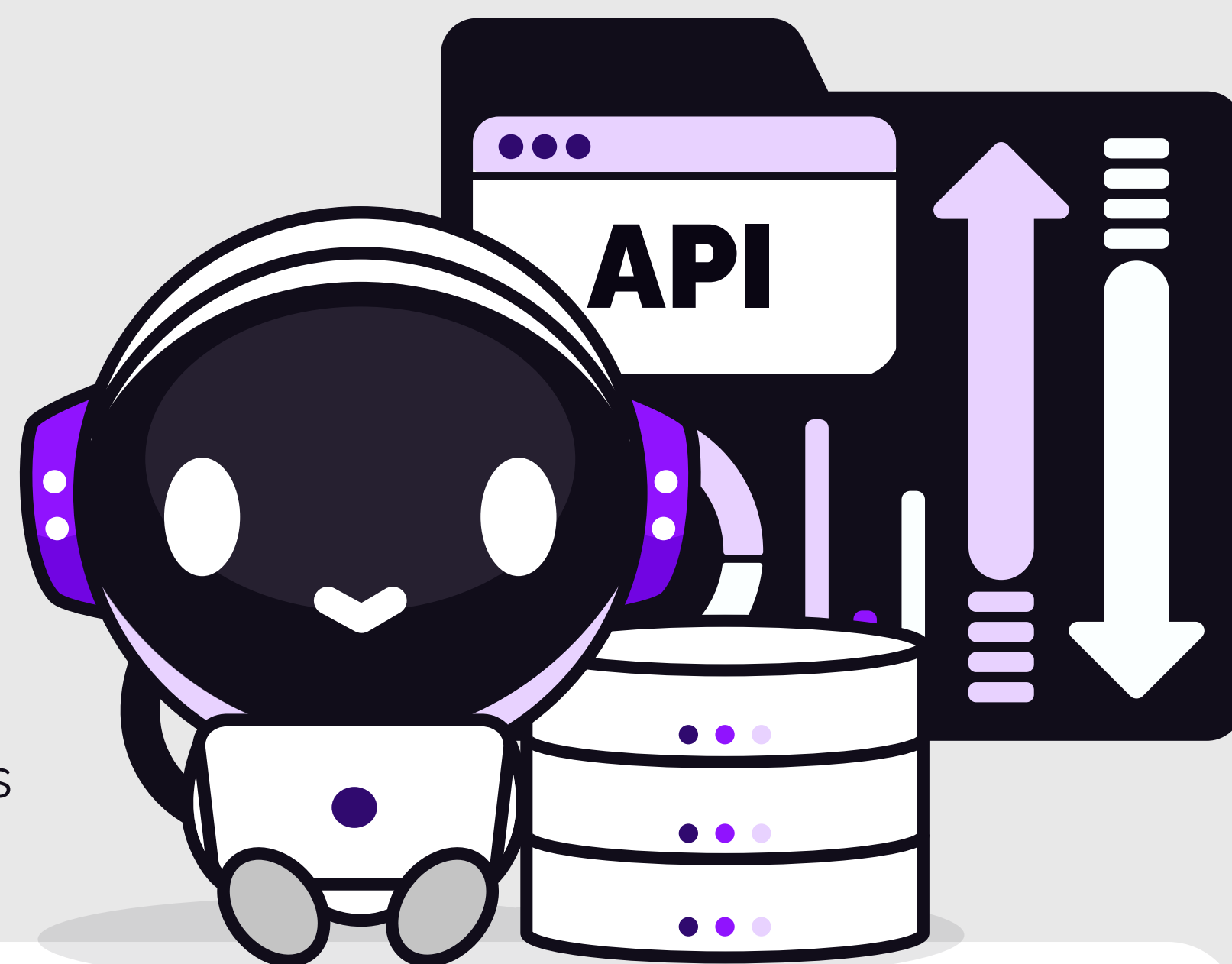
Es una interfaz que permite la comunicación entre computadoras en internet.

Características:

- Usa los **métodos del protocolo HTTP** para realizar operaciones de CRUD con los recursos.
- Suele devolver los datos en formato JSON o XML.
- Arquitectura ligera y flexible, por lo que es ideal para aplicaciones web y móviles.
- Stateless, no es necesario mantener un estado de la sesión en el servidor.

Beneficios

- ⌚ Mayor rapidez y eficiencia en las comunicaciones debido a su simplicidad.
- 🔗 Mejora la interoperabilidad entre sistemas al ser independiente del lenguaje de programación.
- 📦 Es altamente escalable y se puede cachear fácilmente.



GraphQL

Es un lenguaje de consultas que define las especificaciones de cómo una aplicación cliente debe solicitar datos a un servidor remoto.

Características:

- Permite a los clientes recuperar solo los datos que necesitan en una única solicitud.
- Se pueden especificar la forma exacta de los datos que se desean recibir.
- Ideal para aplicaciones donde se necesita una gran flexibilidad en las consultas de datos, como aplicaciones móviles y en tiempo real.

Beneficios

- ⌚ Mejora la eficiencia de las solicitudes al servidor al poder obtener solo los datos requeridos.
- ⚙️ Facilita la evolución de la API sin romper las versiones anteriores.

SOAP (Simple Object Access Protocol)

Es un protocolo que define reglas rígidas de comunicación. Tiene varios estándares asociados que controlan todos los aspectos del intercambio de información.

Características:

- Utiliza un protocolo más estricto de comunicación basado en XML.
- Se apoya en estándares de seguridad y transacciones para garantizar la integridad de los datos.
- Es más común en entornos empresariales y aplicaciones críticas.

Beneficios

- ✓ Mayor seguridad al contar con estándares específicos para ello.
- 📄 Define claramente la estructura de los mensajes a intercambiar entre los sistemas.
- ✓ Es ideal para operaciones que requieran una alta confiabilidad y transaccionalidad.

[🔗](#) Links de referencia

1

2

3

[▶](#) Cursos relacionados



Más recursos

Síguenos: [🐙](#) /Fernando_Her85 [🐙](#) /DevTalles [in](#) /DevTalles

www.devtalles.com

