

Resumen de opciones de APIs

1. APIs de la central

GraphQL code-first con Nexus

Ventajas:

- Tipado extremo end-to-end con TypeScript.
- Único origen de verdad (schema.prisma).
- Generación automática de SDL y resolvers CRUD.
- Flexibilidad para que los clientes soliciten sólo los datos necesarios.
- Ecosistema maduro y comunidad activa.

Desventajas:

- Overhead de parseo y validación de GraphQL en runtime.
- Configuración inicial más compleja.
- Requiere infraestructura de caching y batching para escalar.

Apollo Server SDL-first

Ventajas:

- Sencillez conceptual: esquema definido en SDL.
- Amplio soporte de herramientas (Playground, plugins).
- Integración directa con resolvers manuales.

Desventajas:

- Duplicidad entre SDL y tipos TypeScript.
- Mayor riesgo de errores de tipado en resolvers.
- Mantenimiento manual de SDL y resolvers CRUD.

tRPC

Ventajas:

- Tipado extremo sin esquema separado.
- Cero overhead de parsing de esquemas.
- Batching de llamadas y validación automática con Zod.
- Muy ágil para full-stack TypeScript.

Desventajas:

- Clientes deben ser TS/JS.
- Ecosistema menos maduro y herramientas limitadas.
- No introspección pública de API.

REST/JSON-RPC

Ventajas:

- Máximo rendimiento y compatibilidad.
- Capa de caching HTTP nativa.
- Simplicidad en endpoints y rutas.

Desventajas:

- Sin tipado automático.
- Proliferación de endpoints con crecimiento de la API.
- Mayor duplicidad de lógica de validación.

gRPC

Ventajas:

- Alto rendimiento y streaming.
- Contratos definidos con Protocol Buffers.
- Multilenguaje nativo.

Desventajas:

- Complejidad de infraestructura (proxy para web).
- Curva de aprendizaje más pronunciada.
- Requiere gRPC-Web y proxies para navegador.

2. APIs de los servidores unitarios

REST ligeros

Ventajas:

- Sencillez e interoperabilidad.
- Implementación rápida con Next.js.
- Bajo overhead de librerías.

Desventajas:

- Sin tipado automático.
- Validación manual de payloads.
- Escalabilidad limitada de métodos.

GraphQL puntual

Ventajas:

- Unificación con stack GraphQL central.
- Introspección y Playground disponibles.
- Clientes solicitan sólo lo necesario.

Desventajas:

- Overhead de GraphQL para pocas operaciones.
- Configuración de servidor GraphQL.
- Parsing de esquemas en cada llamada.

tRPC mini

Ventajas:

- Tipado extremo en TypeScript.
- Validación integrada con Zod.
- Batching y resolución eficiente.

Desventajas:

- Solo clientes TS/JS.
- Boilerplate inicial.
- Ecosistema limitado.