



Ciencia de la Computación

Desarrollo Basado en Plataformas

Docente Renzo Hernán Medina Zeballos

GraphQL

Entregado el 05/06/2024

Briceño Quiroz Anthony Angel

Semestre III

2024-1

¿Qué es GraphQL?

GraphQL es un lenguaje de consulta para API y un tiempo de ejecución para completar esas consultas con sus datos existentes. GraphQL proporciona una descripción completa y comprensible de los datos de su API, brinda a los clientes el poder de solicitar exactamente lo que necesitan y nada más, facilita la evolución de las API con el tiempo y habilita potentes herramientas para desarrolladores.

Conceptos:

- **Schemas:** El schema en GraphQL es una definición de todos los tipos de datos que puede manejar la API, incluyendo las relaciones entre estos tipos. El schema también define las operaciones que se pueden realizar (queries y mutations). Es como el contrato entre el cliente y el servidor que especifica qué datos están disponibles y cómo se pueden consultar o modificar.
- **Queries:** Las queries en GraphQL son las solicitudes que los clientes hacen para obtener datos. A diferencia de REST, donde hay múltiples endpoints, en GraphQL, todas las solicitudes se realizan a un solo endpoint, pero pueden solicitar diferentes datos según lo definido en el schema.
- **Resolver:** Un resolver es una función que se encarga de obtener los datos solicitados en una query. Cuando una query es ejecutada, GraphQL llama a los resolvers correspondientes para obtener los datos necesarios y construir la respuesta. Cada campo en una query puede tener su propio resolver.

GraphQL vs. REST APIs:

- **Endpoint único vs múltiples endpoints:** GraphQL utiliza un único endpoint para todas las solicitudes, mientras que REST utiliza múltiples endpoints para diferentes recursos.
- **Selección de datos:** En GraphQL, el cliente puede especificar exactamente qué datos necesita, mientras que en REST, el servidor define qué datos se devuelven por cada endpoint.
- **Agregación de datos:** GraphQL permite a los clientes solicitar datos agregados de múltiples recursos en una sola solicitud. En REST, esto a menudo requiere múltiples solicitudes.

Ventajas de GraphQL:

- **Precisión en la solicitud de datos:** Los clientes pueden pedir exactamente los datos que necesitan, lo que puede reducir el tráfico de red.
- **Menos endpoints:** Un solo endpoint para todas las operaciones puede simplificar la gestión de la API.

- **Desarrollos más rápidos:** Al definir el schema, los desarrolladores tienen una clara documentación de lo que la API ofrece, lo cual acelera el desarrollo y mantenimiento.
- **Mejora en la agregación de datos:** GraphQL permite obtener datos relacionados en una sola solicitud, evitando múltiples llamadas a la API.

Desventajas de GraphQL:

- **Complejidad inicial:** La configuración inicial de GraphQL puede ser más compleja en comparación con REST.
- **Overfetching:** Aunque GraphQL evita el problema del overfetching al permitir solicitudes precisas, si no se usa correctamente, puede resultar en subfetching o solicitudes ineficientes.
- **Caché:** Las estrategias de caché en GraphQL pueden ser más complicadas que en REST, donde el caché HTTP es más straightforward.
- **Sobrecarga en el servidor:** Las consultas muy complejas pueden sobrecargar el servidor si no se gestionan adecuadamente.