Cada vez que tenemos un proyecto donde se necesita conseguir datos de distintas fuertes (Apis, o microservicios), entre otras cosas, nos encontramos con la difícil elección entre REST y GraphQL para poder estructurar y consumir estos datos.

Por un lado, REST utiliza HTTP para poder modificar y acceder a los recursos. Cada uno de los recursos tendrá su propia URL. /libros nos llevara los libros, /autores a los autores, y así con cada recurso.

Un modelo claro. Hacemos GET, POST, PUT y DELETE a un endpoint especifico y el servidor nos devuelve una respuesta a partir de ese ese endpoint.

Por otro lado, GraphQL es un lenguaje de consultas donde se puede hacer un pedido exactamente de los datos que necesitamos, sin necesidad de hacer una petición distinta para cada endpoint. En este caso todo se encuentra en un único endpoint desde donde accedemos a los recursos, teniendo la posibilidad de consultar múltiples recursos al mismo tiempo, de esta manera el cliente tiene mas control y flexibilidad sobre los datos que solicita .

Ahora, ¿Que pasa cuando trabajamos con muchos servicios externos?

Cada vez que hacemos una solicitud a una API externa con REST, nos encontramos con varias llamadas HTTP independientes. Luego podemos hacer una combinación de los datos desde el backend y mostrar al usuario la ifnormacion que requiere, pero puede pasar que se genere un overfetching ( donde recibimos mas datos que los que necesitamos) o un underfetching (Donde no tenemos los datos suficientes y tenemos que hacer una nueva llamada para completar los datos faltantes)

Cuando trabajamos con GraphQL se podría trabajar con un resolver que lo que hace es consultar a los servicios externos y combina los distintos recursos y devuelva una unica respuesta. El cliente solo pide los campos necesarios, haciendo que el rendimiento mejore mucho y se disminuya el ancho de banda necesario para realizar la petición.

En conclusión, así como REST es muy bueno cuando tenemos servicios independientes que no necesitamos combinar de manera eficiente, GraphQL es una mejor herramienta, mas optima para poder combinar la lógica de combinación de datos, siempre y cuando, por supuesto, los resolvers esten bien llevados a cabo.