

GRAPHQL

- O que é GraphQL?
- Por que usar GraphQL?
- Como começar com GraphQL?

- GraphQL é uma linguagem de consulta para APIs (semelhante ao *REST*)
- Permite que os clientes solicitem exatamente os dados de que precisam, evitando o excesso ou a falta de informações.
- Funciona com um sistema de tipos forte para definir a estrutura dos dados disponíveis.

GraphQL

- Consulta flexível
- Recupera apenas os dados necessários
- Sistema de tipos **forte**
- Documentação *automática*
- Endpoint único

REST

- Estrutura fixa
- Pode retornar dados desnecessários
- Sem sistema de tipos forte
- Documentação *manual* (depende do desenvolvimento)
- Múltiplos endpoints

- Eficiência na recuperação de dados
 - Reduz múltiplas chamadas de API
- Flexibilidade para os clientes
 - Solicitam o que precisam
- Evolução da API sem versões
 - Adicionar novos campos sem afetar os existentes
- Ferramentas poderosas para desenvolvimento e depuração
 - Clientes embutidos com ferramentas de desenvolvimento

Orientado a operação! No lado do servidor, as operações são definidas, e o cliente apenas as invoca. Então não temos endpoints de operação, e sim chamadas de “função”.

Todas as operações tem um tipo, e estes tipos são fortes, e o cliente deve levar estes em consideração ao fazer uma chamada, ou ao definir como vai consumir.

O retorno é sempre em JSON, e o cliente deve interpretar este JSON de acordo com o que foi definido na operação.

Schema Define os tipos e relações dos dados disponíveis.

Queries Solicitações feitas pelos clientes para recuperar dados.

Mutations Operações para modificar dados no servidor.

Definição do esquema (schema)

Define tipos e relações

```
type User {  
  id: ID!  
  name: String!  
  email: String!  
}
```


Consultas (Queries)

Solicitam dados específicos

```
{
  getUserById(id: "1") {
    name
    email
  }
}
```

Quem define a operação de busca é o servidor.

Alterações (Mutations)

Modificam dados no servidor

```
mutation {  
  createUser(name: "Pedro Pedreira", email: "pedro.pedreira@exemplo.com") {  
    id  
    name  
    email  
  }  
}
```

Quem define a operação da Mutation é o servidor. Não existem verbos HTTP para isso.

O GraphQL pode conviver com o REST, permitindo uma transição gradual.

O GraphQL é agnóstico em relação à linguagem de programação e pode ser implementado em várias linguagens, tanto o cliente quanto o servidor. *Nisso ele é similar ao REST.*

- <https://spring.io/projects/spring-graphql>
- <https://pt.quarkus.io/guides/smallrye-graphql>
- <https://www.apollographql.com/docs/kotlin> (cliente, mas só funciona com Kotlin)

Vamos à prática!