



Bem-vindos!



A linguagem de consulta e manipulação de dados para APIs





Nome: Doris Helena Nunes

Empresa: Compass

GitHub: <https://github.com/DrisHel>

Linkedin: <https://www.linkedin.com/in/dris/>





O que é GraphQL?





Como ou quando surgiu?





Conceitos básicos...





Onde posso usar?





Quem usa?





Quais as Vantagens?

- ✓ Busca de dados declarativa
- ✓ Soluciona o over fetching, under fetching
- ✓ GraphQL é versátil



Exemplo 1 overfetching:
<http://localhost:3000/users>
DB(usuário, endereço)

Exemplo 2 underfetching:

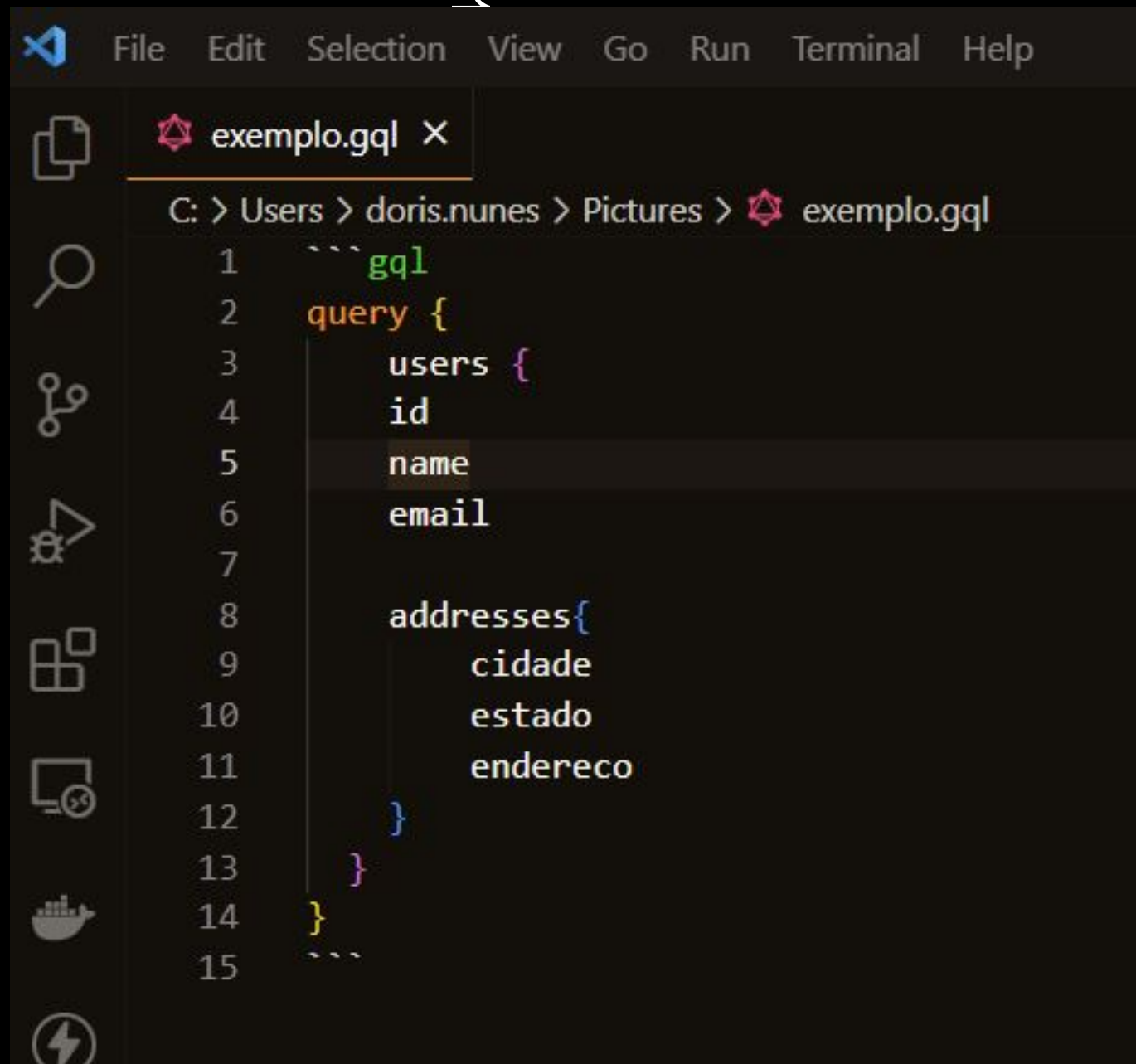
<http://localhost:3000/users>

DB(usuário)

<http://localhost:3000/addresses>

Rota graphql:

[http://localhost:3000/
graphql](http://localhost:3000/graphql)



```

1  `` gql
2  query {
3      users {
4          id
5          name
6          email
7
8          addresses {
9              cidade
10             estado
11             endereco
12         }
13     }
14 }
15 ``
  
```



<https://api.com/cheeseburger/>



```
query getCheeseburger ($vegan: Boolean) {
  cheeseburger {
    bun
    lettuce
    patty
    bun
    cheese @skip(if: $vegan)
  }
}
```







Quais as Desvantagens?

- ✓ Problemas para lidar com consultas complexas
- ✓ Não é a melhor opção para pequenas aplicações
- ✓ Cache é complexo
- ✓ Curva de aprendizado íngreme





HTTP Methods

GET



Receive information
about an API resource

POST



Create an
API resource

PUT



Update an
API resource

DELETE

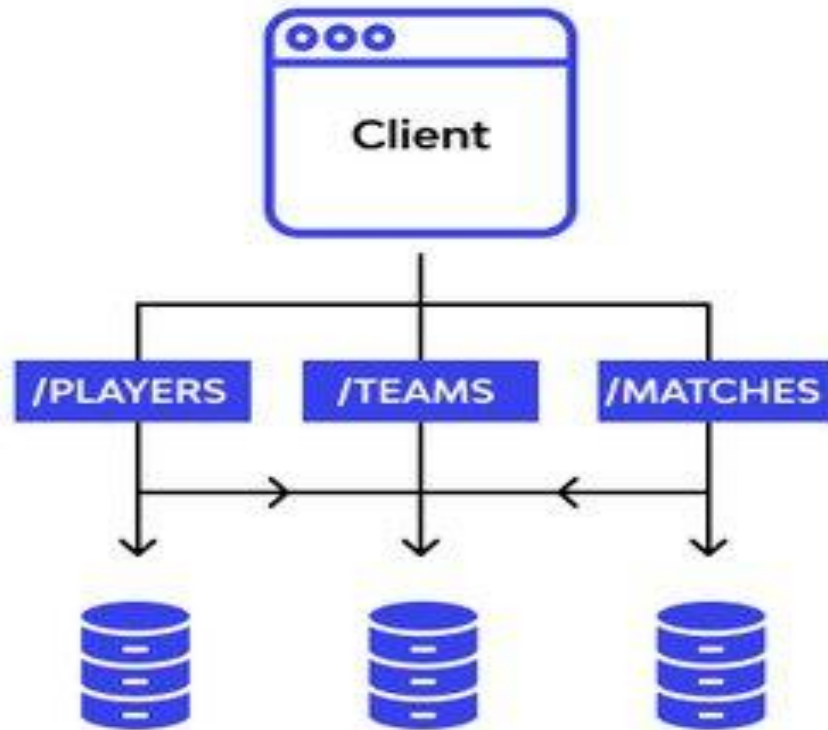


Delete an
API resource

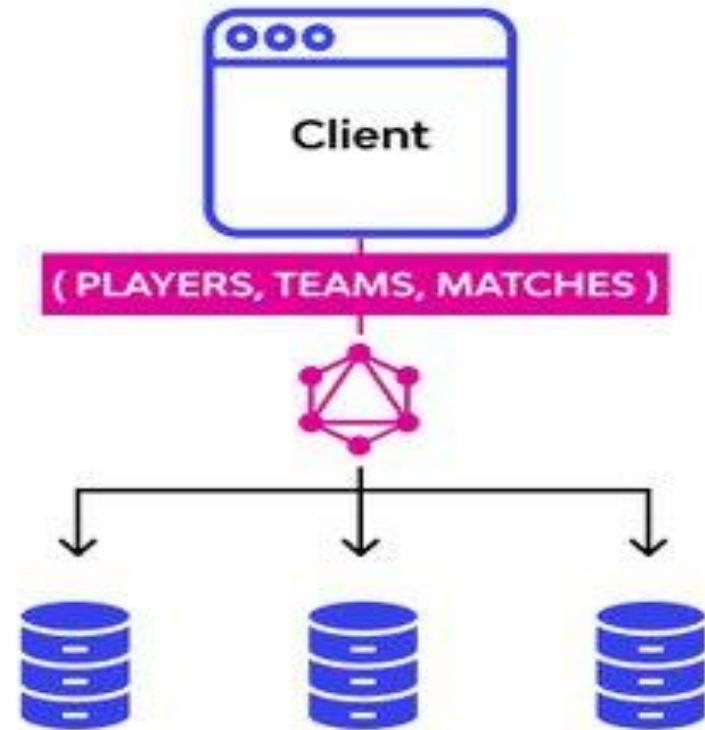




Rest API



GraphQL API





The screenshot shows the GraphQL Playground interface. At the top, there's a tab labeled 'customer' with a close button and a plus sign to add more tabs. Below the tabs are two buttons: 'PRETTIFY' and 'HISTORY', followed by the URL 'http://localhost:5000/graphql'. The main area is split into two panels. The left panel contains a query editor with the following text:

```
1 # Write your query or mutation here
2 {
3   customer(id: "6e2eccf5-0204-45c5-b519-a49944fa08c8") {
4     id
5     name
6     email
7   }
8 }
```

The right panel shows the JSON response, which is expanded to show the details of the 'customer' object. A large play button icon is positioned between the two panels.

```
{
  "data": {
    "customer": {
      "id": "6e2eccf5-0204-45c5-b519-a49944fa08c8",
      "name": "Marie Doe",
      "email": "mariedoe@doe.com"
    }
  },
  "extensions": {}
}
```





A tabela abaixo mostra as diferenças mais fundamentais entre GraphQL e REST.

	GraphQL	REST
Tecnologia	Nova	Madura
Ideal para	Apps maiores	Apps pequenos e medios
Arquitetura	Arquitetura orientada para o cliente	Arquitetura baseada em servidor
Operacoes	Mutation, query, subscription	CRUD
Vantagens	Sem overfetching	Os dados não estão vinculados a recursos ou métodos
Tipo de organização	Schema	Endpoints
Data fetching	Chamada de API única para busca de dados específicos	Várias chamadas de API com dados fixos
Comunidade	Crescendo	Grande
Performance	Desempenho rápido	Mais tempo necessário para várias chamadas
Velocidade de desenvolvimento	Desenvolvimento rápido	Desempenho lento
Complexidade de aprendizagem	Alta	Baixa
Autodocumentado	Sim	Nao
Opção de upload de arquivo	Nao	Sim
Opção de cache da web	Só com bibliotecas	Sim
Estabilidade	Menos erros, validação automática e verificação de tipo	Mais adequado para consultas complexas





Canais oficiais:

Twitter: @graphql

GitHub: github.com/graphql

Discord discord.graphql.org

Grupo do Facebook: fb.com/groups/graphql.community/

Google Agenda: calendar.graphql.org

Blog: graphql.org/blog





Extras: <https://www.youtube.com/watch?v=xbLpIhCsIdg>

<https://www.youtube.com/watch?v=6SZOPKs9SUg&t=1506s>

https://www.youtube.com/watch?v=1dz48pReq_c&t=109s

https://blog.back4app.com/pt/graphql-vs-rest-quais-sao-as-diferencas/#Quais_sao_os_beneficios_do_GraphQL





"O sucesso é a soma de pequenos
esforços **repetidos** dia após dia."

• Robert Collier



Obrigada pela participação!