

GraphQL vs REST: Um Experimento Controlado

Bernardo C. Rohlfs¹, Leonardo A. P. do Carmo², Matheus B. S. Mello³, Tarcísio N. M. Filho⁴

¹Engenharia de Software – Pontifícia Universidade Católica de Minas Gerais (PUCMG)
Av. Brasil, 2024 - Funcionários, Belo Horizonte - MG, 30140-002

{mbmelo1421¹, tarcisioneym²}@gmail.com

Abstract. *The GraphQL query language, developed by Facebook, is an innovative alternative to traditional REST APIs, which are widely used for implementing web APIs. Unlike REST, which uses endpoints with predefined operations, GraphQL is graph-based and allows database queries through schemas, enabling more flexible and provider-defined queries.*

Although many systems have adopted GraphQL while maintaining compatibility with REST, the real benefits of this adoption are not yet clearly defined. This study proposed a controlled experiment to quantitatively evaluate these benefits compared to REST APIs.

The experiment's results showed that, in terms of response time and the size of returned data, GraphQL performed more satisfactorily than REST. Based on these varied query scenarios, it was concluded that GraphQL is more efficient and offers greater benefits.

For future studies, it is recommended to increase the number of query executions, diversify the types of queries, and extend the data collection period to gain a more in-depth and detailed understanding of the advantages of each API type.

Resumo. *A linguagem de consulta GraphQL, desenvolvida pelo Facebook, é uma alternativa inovadora às tradicionais APIs REST, amplamente utilizadas para implementação de APIs Web. Diferente do REST, que usa endpoints com operações pré-definidas, o GraphQL é baseado em grafos e permite consultas a bancos de dados através de schemas, possibilitando consultas mais flexíveis e definidas pelo fornecedor da API.*

Embora muitos sistemas tenham adotado o GraphQL mantendo a compatibilidade com REST, os benefícios reais dessa adoção ainda não são claramente definidos. Este laboratório propôs um experimento controlado para avaliar quantitativamente esses benefícios em comparação às APIs REST.

Os resultados do experimento mostraram que, em termos de tempo de resposta e tamanho dos dados retornados, o GraphQL apresentou desempenho mais satisfatório que o REST. Com base nesses cenários variados de consultas, concluiu-se que o GraphQL é mais eficiente e oferece maiores benefícios.

Para futuros estudos, recomenda-se aumentar o número de execuções das consultas, diversificar os tipos de consultas e prolongar o período de coleta de dados, visando um entendimento mais aprofundado e detalhado das vantagens de cada tipo de API.

1. Introdução

A linguagem de consulta GraphQL, desenvolvida pelo Facebook, surge como uma alternativa inovadora às tradicionais APIs REST, amplamente utilizadas na implementação de

APIs Web. GraphQL é baseada em grafos e permite que os usuários consultem bancos de dados através de schemas, possibilitando a exportação da base e a realização de consultas em um formato definido pelo fornecedor da API. Em contrapartida, as APIs REST utilizam endpoints com operações pré-definidas para consulta, deleção, atualização ou inserção de dados. Embora muitos sistemas tenham migrado para GraphQL, mantendo compatibilidade com REST, os reais benefícios dessa adoção ainda não estão claramente definidos.

Diante disso, este laboratório propõe um experimento controlado para avaliar quantitativamente os benefícios da adoção de uma API GraphQL em comparação a uma API REST. As questões de pesquisa específicas que guiam este estudo são:

1. Respostas a consultas GraphQL são mais rápidas que respostas a consultas REST?
2. Respostas a consultas GraphQL têm tamanho menor que respostas a consultas REST?

Para abordar essas questões, este estudo testa as seguintes hipóteses:

- **Hipótese Nula:** Não há diferença significativa na velocidade e no tamanho das respostas entre consultas GraphQL e consultas REST.
- **Hipótese Alternativa:** Consultas GraphQL são mais rápidas e geram respostas menores do que consultas REST para o mesmo conjunto de dados.

2. Metodologia

A metodologia utilizada neste experimento é detalhada a seguir, com o objetivo de permitir a reprodução e replicação dos resultados obtidos.

2.1. Desenho do Experimento

Variáveis Dependentes:

- Tempo de resposta (medido em segundos)
- Tamanho da resposta (medido em bytes)

Variáveis Independentes:

- Tipo de API (GraphQL vs REST)
- Tipo de consulta (quantidade de dados solicitados)

Tratamentos:

- Realização de diferentes tipos de requisições, focando em repositórios mais populares e repositórios Python mais populares.
- Cada consulta será realizada 25 vezes para garantir consistência e reduzir variabilidade nos resultados.

Objetos Experimentais:

- Consulta ao mesmo conjunto de dados utilizando tanto a API GraphQL quanto a API REST.

Tipo de Projeto Experimental:

- Projeto controlado, comparando consultas no mesmo conjunto de dados em ambas as APIs.

Quantidade de Medições:

- Cada consulta será executada 25 vezes, permitindo a obtenção de médias e medianas mais confiáveis.

2.2. Ambiente de Execução

Os trials foram realizados em um ambiente controlado para minimizar interferências externas. As consultas foram feitas em servidores com especificações semelhantes, garantindo que a infraestrutura não influencie nos resultados. As ferramentas e bibliotecas utilizadas para realizar as consultas e medir os tempos e tamanhos de resposta incluem:

- Ferramentas para realizar consultas HTTP (como ‘curl’ ou bibliotecas HTTP em Python)
- Ambientes de desenvolvimento idênticos (versão do Python, versões das bibliotecas)
- Infraestrutura de rede controlada para garantir consistência na latência e largura de banda

2.3. Procedimento Experimental

Preparação do Ambiente:

- Configuração dos servidores para hospedagem das APIs GraphQL e REST.
- Implementação de endpoints equivalentes em ambas as APIs para permitir comparações diretas.

Execução das Consultas:

- Para cada tipo de consulta definida, executar 5 requisições utilizando tanto a API GraphQL quanto a API REST.
- Registro dos tempos de resposta e tamanhos das respostas para cada requisição.

Análise dos Dados:

- Cálculo das médias e medianas dos tempos de resposta e tamanhos das respostas.
- Aplicação de testes estatísticos para avaliar a significância das diferenças observadas entre as duas abordagens.

Este experimento pretende fornecer uma análise quantitativa robusta sobre os benefícios potenciais da adoção de APIs GraphQL em comparação às APIs REST, contribuindo para uma compreensão mais clara das vantagens e limitações de cada abordagem.

3. Resultados

Essa seção tem o objetivo de apresentar os resultados das métricas que respondem cada pergunta proposta.

3.1. RQ1 - Respostas a consultas GraphQL são mais rápidas que respostas a consultas REST?

Para responder a primeira questão, foi utilizada duas *queries* diferentes, a primeira buscamos os 100 repositórios mais populares, com suas data de criação, o dono do projeto e o número de estrelas. Na segunda *query* foi buscado os mesmos parâmetros com a adição do número de *commits*. Dessa maneira, com as consultas revelou-se que as respostas em consultas *GraphQL* são mais rápidas do que o *REST* considerando o tempo gasto.

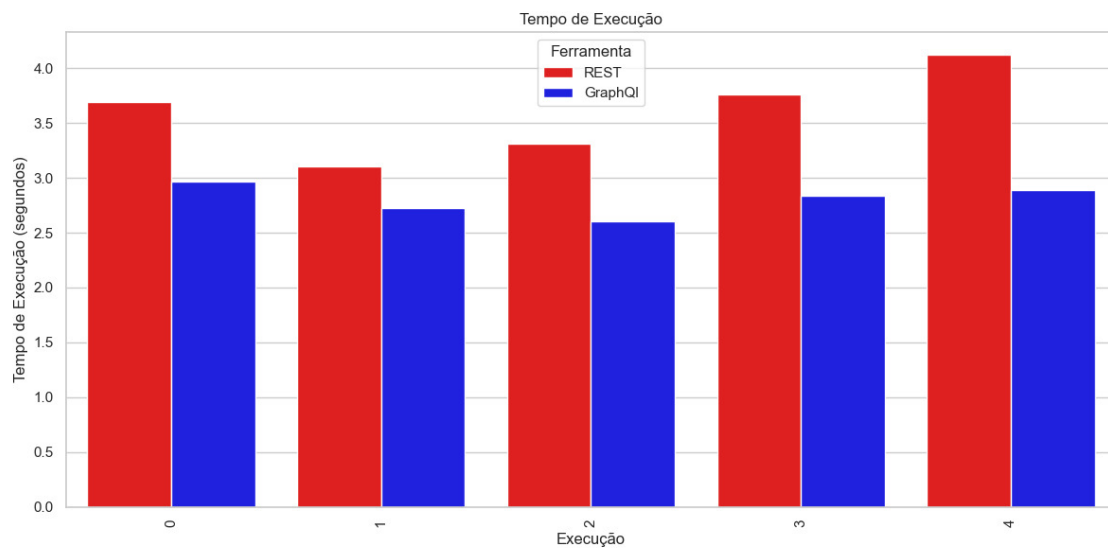


Figura 1. Query 1: Tempo de Execução

Conforme podemos identificar na Figura 1, nas cinco execuções da primeira *query* obtivemos um tempo maior nas consultas utilizando REST (em média 3,5 segundos) do que o GraphQL (em média 2,5 segundos)

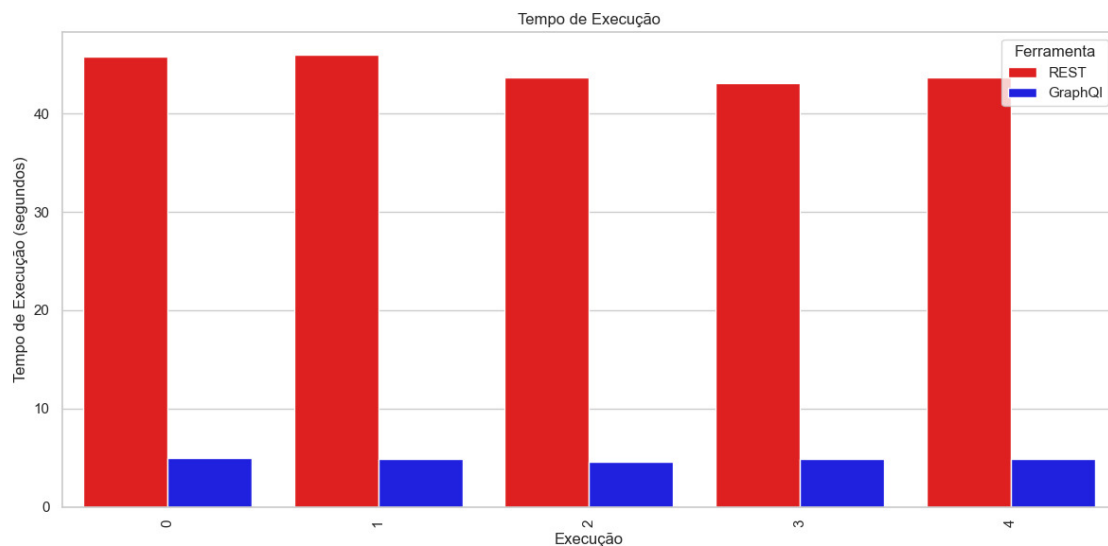


Figura 2. Query 2: Tempo de Execução

Com um comportamento diferente da primeira *query*, observa-se na Figura 2 que o tempo de execução das consultas GraphQL foram significativamente menores, em média 5 segundos, do que às realizadas para o REST, em média 44 segundos

3.2. RQ2 - Respostas a consultas GraphQL têm tamanho menor que respostas a consultas REST?

Para responder a segunda questão foi utilizada as mesmas *queries* da primeira pergunta. Dessa maneira, com as consultas realizadas revelou-se que as respostas em consultas *GraphQL* possuem tamanho menor do que o *REST* considerando o a quantidade de *bytes* consumidas na consultas.

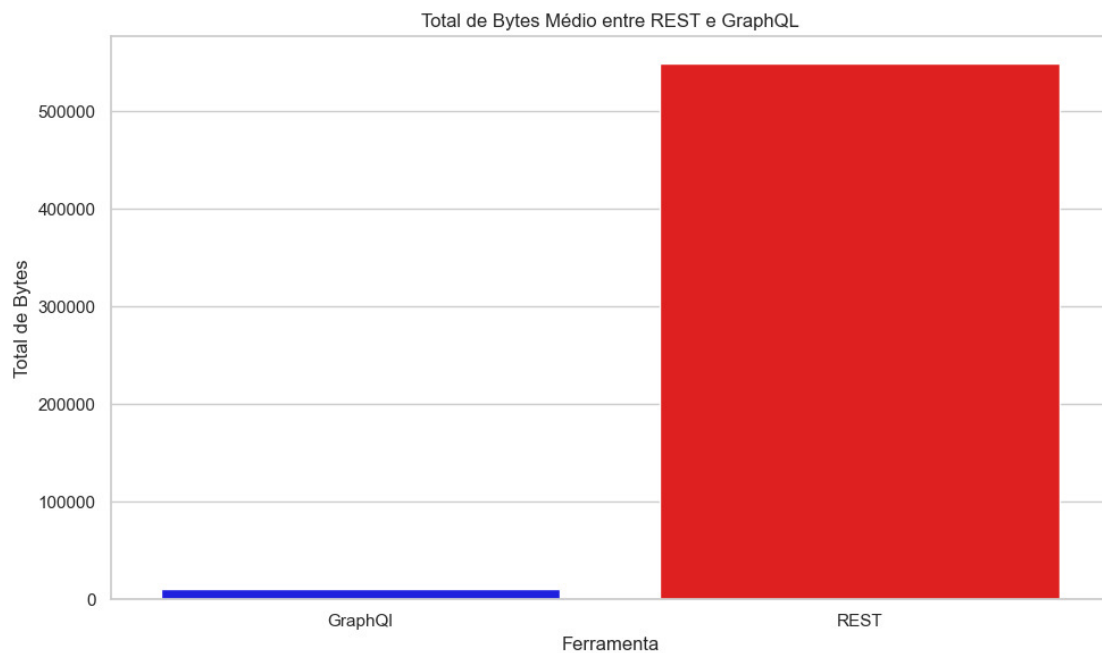


Figura 3. Query 1: Total de Bytes Médio

Conforme podemos identificar na Figura 3, na primeira *query* obtivemos um consumo de *bytes* do GraphQL (10194 bytes) significamente menor do que na consulta utilizando REST (549361 bytes)

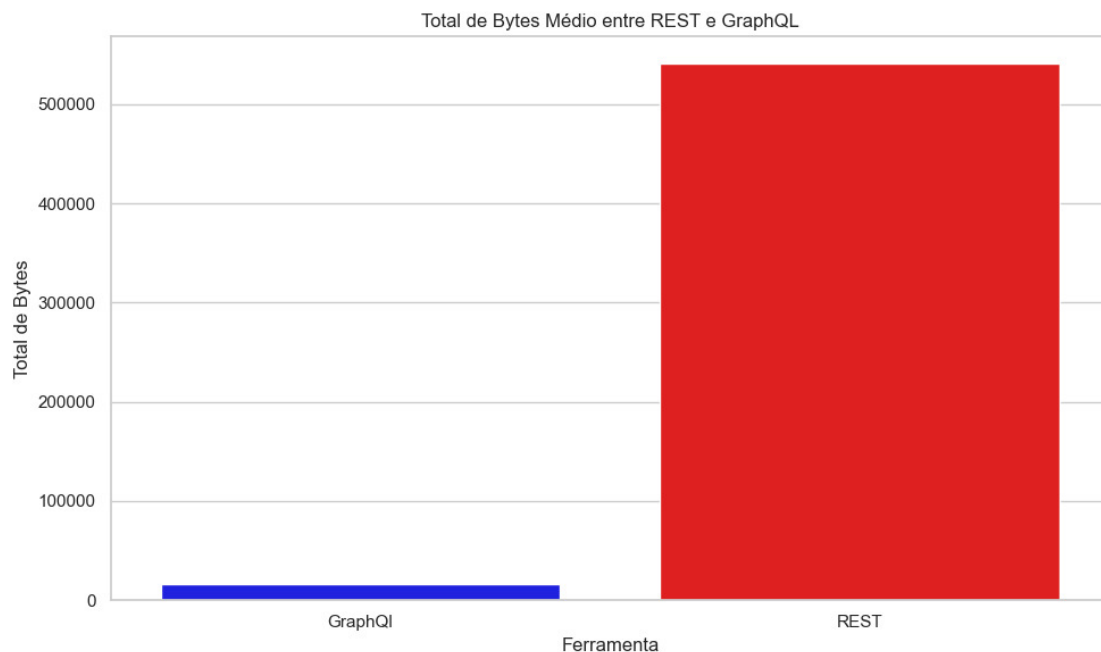


Figura 4. Query 2: Total de Bytes Médio

Conforme podemos identificar na Figura 4, obtivemos um comportamento semelhante à primeira *query*. Dessa maneira, obtivemos um consumo de *bytes* do GraphQL (16107 bytes) significativamente menor do que na consulta utilizando REST (541139 bytes)

4. Conclusão

Embora muitos sistemas tenham migrado para GraphQL, mantendo compatibilidade com REST, pode-se averiguar, quantitativamente, pelo experimento controlado os reais benefícios da adoção de uma API GraphQL em comparação a uma API REST. Nesse sentido, conforme os resultados obtidos, pode-se analisar, utilizando cenários variados de consultas, que tanto em termos de tempo quanto de tamanho o GraphQL apresentou resultados mais satisfatórios do que o REST, cooperando para a inferência de que realmente o GraphQL é mais eficiente e possui maiores benefícios do que o REST.

Ademais, seria positivo para trabalhos futuros realizar um maior número de execuções das *queries*, utilizar mais tipos de consultas e aumentar o período de coleta de dados. Assim, possibilitaria um conhecimento mais aprofundado e detalhado dos benefícios de uma API (Interface de Programação de Aplicação) perante o outro.

5. Pacote de Replicação

O pacote de replicação deste trabalho encontra-se disponível em: https://github.com/MatheusBelo/lab5_graphql_vs_rest.git

Referências