
1. Desenho do Experimento

O experimento foi elaborado visando assegurar a validade dos resultados e responder às questões de pesquisa propostas. Foram definidas hipóteses, variáveis dependentes e independentes.

A. Hipóteses Nula e Alternativa

- **Hipótese Nula (H0):** Não há diferença significativa entre GraphQL e REST em termos de tempo de resposta e tamanho das respostas para as consultas.
- **Hipótese Alternativa (H1):** GraphQL apresenta uma diferença significativa (tempo de resposta menor ou tamanho da resposta menor) quando comparado ao REST.

B. Variáveis Dependentes

- **Tempo de Resposta:** Medido em milissegundos, representa a rapidez com que a API responde a uma consulta.
- **Tamanho da Resposta:** Medido em bytes, representa a quantidade de dados trafegados em uma consulta, o que impacta a eficiência da comunicação de rede.

C. Variáveis Independentes

- **Tipo de API:** Serão utilizados GraphQL e REST como comparativos.
- **Tipo de Consulta:** Variar entre consultas de dados específicos e múltiplas entidades.
- **Volume de Dados:** O número de registros ou objetos retornados nas consultas será modificado para simular diferentes cargas de dados.

D. Tratamentos

Para cada variável independente (tipo de API e tipo de consulta), será aplicado um tratamento envolvendo a realização de consultas similares em ambas as APIs. Exemplos de consultas incluem:

- Busca por um registro específico.
- Busca por múltiplos registros.

E. Objetos Experimentais

Neste experimento, serão utilizados dados reais de repositórios populares do GitHub, que poderão ser consultados tanto pela API REST quanto pela GraphQL. Estes dados serão estruturados de maneira similar em ambas as APIs para assegurar a comparação.

F. Tipo de Projeto Experimental

Será realizado um experimento com dois grupos: consultas GraphQL e consultas REST. O experimento será estruturado como um estudo comparativo controlado, com delineamento de medidas repetidas. Esse delineamento permitirá que as mesmas consultas sejam executadas repetidamente em ambas as APIs, sob condições controladas, para reduzir variáveis de confusão e garantir a confiabilidade dos resultados.

As consultas serão realizadas em três cenários experimentais:

1. **Consulta Simples:** Retorno de um único registro ou item específico.
2. **Consulta Complexa:** Retorno de múltiplos registros ou itens agregados.

G. Quantidade de Medições

Cada tipo de consulta será repetido pelo menos 5000 vezes para cada API, a fim de controlar variâncias e identificar dados atípicos.

H. Ameaças à Validade

- **Ameaças à Validade Interna:** Variabilidade de desempenho do servidor, rede ou cachê de dados. Será necessário garantir condições estáveis durante o experimento.
 - **Ameaças à Validade Externa:** A aplicação dos resultados é limitada ao ambiente e API específica do GitHub. Resultados podem não se generalizar para outros tipos de APIs.
 - **Controle de Influências Externas:** O experimento será realizado em um ambiente controlado, minimizando o impacto de variáveis externas.
-

2. Metodologia

2.1 Criação do Dataset

Este experimento compara o desempenho entre as APIs REST e GraphQL do GitHub, utilizando consultas similares ao trabalho anterior "Análise de Repositórios Populares no GitHub - [Lab01]". Os dados serão coletados por ambas as APIs e estruturados de forma semelhante, garantindo uma comparação justa. Serão realizadas 5000 requisições para cada API para permitir uma análise estatística robusta.

2.2 Consultas

As consultas retornarão informações semelhantes, como nome do repositório, data de criação, URL, número de estrelas, issues, pull requests, releases, e linguagem principal. Serão realizadas 30 requisições para cada consulta, totalizando 1000 repositórios analisados.

5. Resultados

- **Fig. 1 - Dados Obtidos**

Em uma análise comparativa entre REST e GraphQL, observou-se que o tempo médio de resposta foi significativamente menor para REST, com um tempo de 32,300 milissegundos, enquanto GraphQL apresentou um tempo muito superior, atingindo 736,036 milissegundos.

- **Tamanho dos Dados Transferidos**

REST transferiu um volume maior de dados, com uma média de 5.447.890 bytes, comparado a GraphQL, que transferiu 179.496 bytes em média.

- **Conclusões**

Embora GraphQL seja considerado mais eficiente por design, neste experimento REST apresentou melhor desempenho, devido a timeouts na API do GitHub, que afetaram negativamente o desempenho do GraphQL.