

LABORATÓRIO DE EXPERIMENTAÇÃO DE SOFTWARE



# GraphQL vs. REST: Um experimento controlado

Grupo: Ana Luiza Machado, Lucas Barros e  
Raquel Calazans



# Introdução & Objetivo

## REST

Estilo arquitetônico para APIs que usa HTTP e múltiplos endpoints.

## GraphQL

Linguagem de consulta para APIs que permite pedir exatamente os dados desejados em um único endpoint.

O nosso objetivo é **comparar GraphQL vs REST** em termos de desempenho e tamanho de payload através de um experimento controlado.

# Metodologia

O experimento segue as etapas: **(1)** desenho, preparação, **(2)** execução, **(3)** análise e relatório.

A coleta consistiu em executar consultas equivalentes REST e GraphQL contra a API do GitHub em diferentes tratamentos:

- cache ***cold/warm***
- níveis de carga **1/10/50**
- tipos de consulta ***simple, nested*** e ***aggregated***.

01

Definição das RQs, variáveis e fatores; implementação dos scripts REST/GraphQL; configuração do ambiente e dos cenários de teste.

02

Realização automatizada das consultas em todas as combinações de fatores, coleta do tempo de resposta, payload e registros brutos.

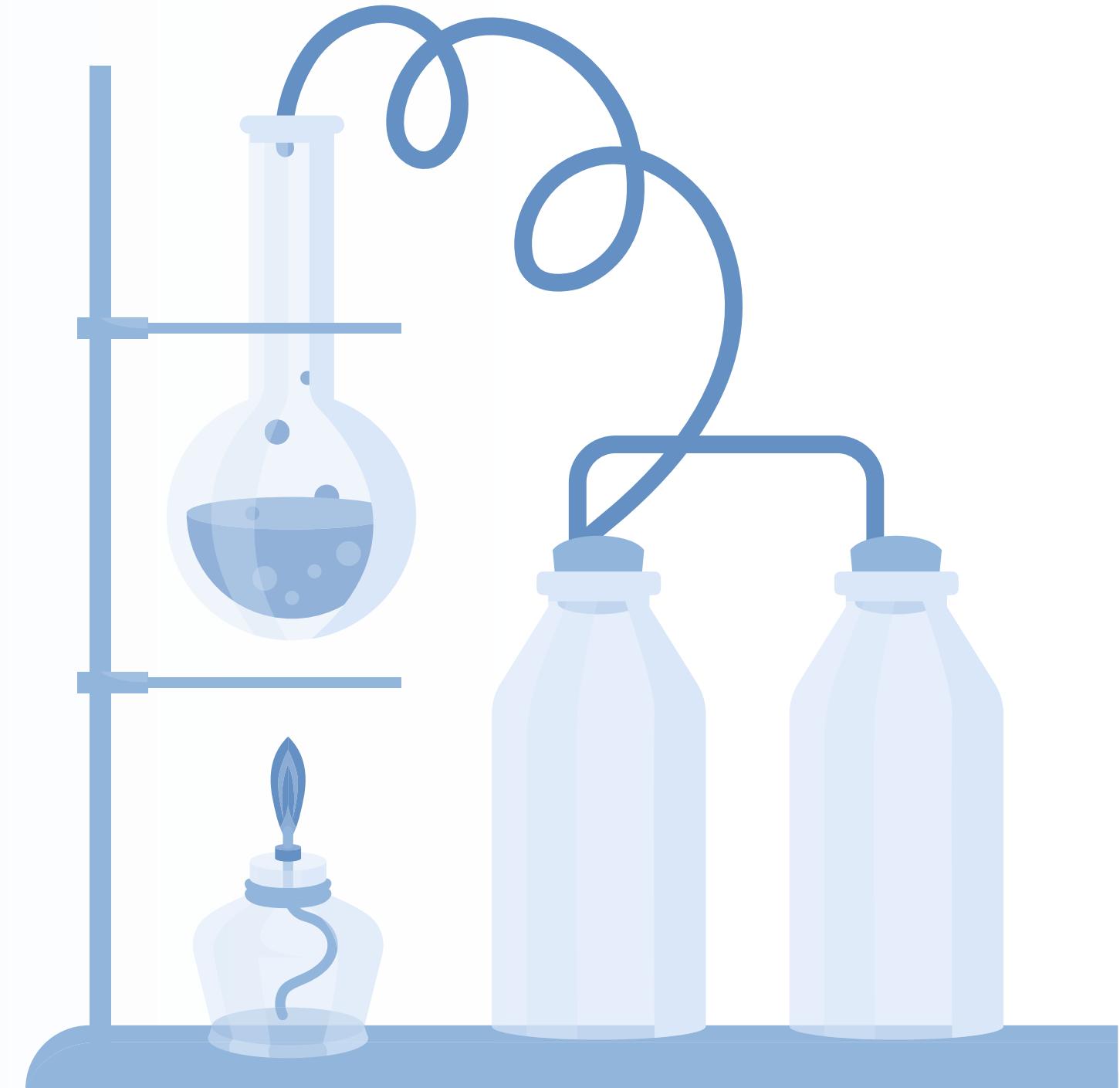
03

Cálculo das métricas, geração de gráficos, testes estatísticos e documentação dos resultados, incluindo conclusões e limitações.

# Tecnologias e Ferramentas

Ferramentas usadas no experimento:

- **Python 3.x** (bibliotecas: requests, gql, pandas, seaborn, etc.)
- **Google Looker Studio** (Dashboard de visualização)
- **APIs do GitHub** (REST e GraphQL)



# RQs e Hipóteses

As questões de pesquisa foram definidas para guiar a investigação e estruturar a análise dos dados coletados:

- **RQ 01: Respostas GraphQL são mais rápidas que REST?**
  - **H<sub>0</sub> (RQ1):** Não há diferença significativa nos tempos de resposta entre GraphQL e REST.
  - **H<sub>1</sub> (RQ1):** GraphQL apresenta tempos de resposta significativamente menores do que REST.

---
- **RQ 02: Respostas GraphQL têm tamanho de payload menor do que REST?**
  - **H<sub>0</sub> (RQ2):** Não há diferença significativa no tamanho dos payloads entre GraphQL e REST.
  - **H<sub>1</sub> (RQ2):** GraphQL apresenta payloads significativamente menores do que REST.

# Desenho do Experimento

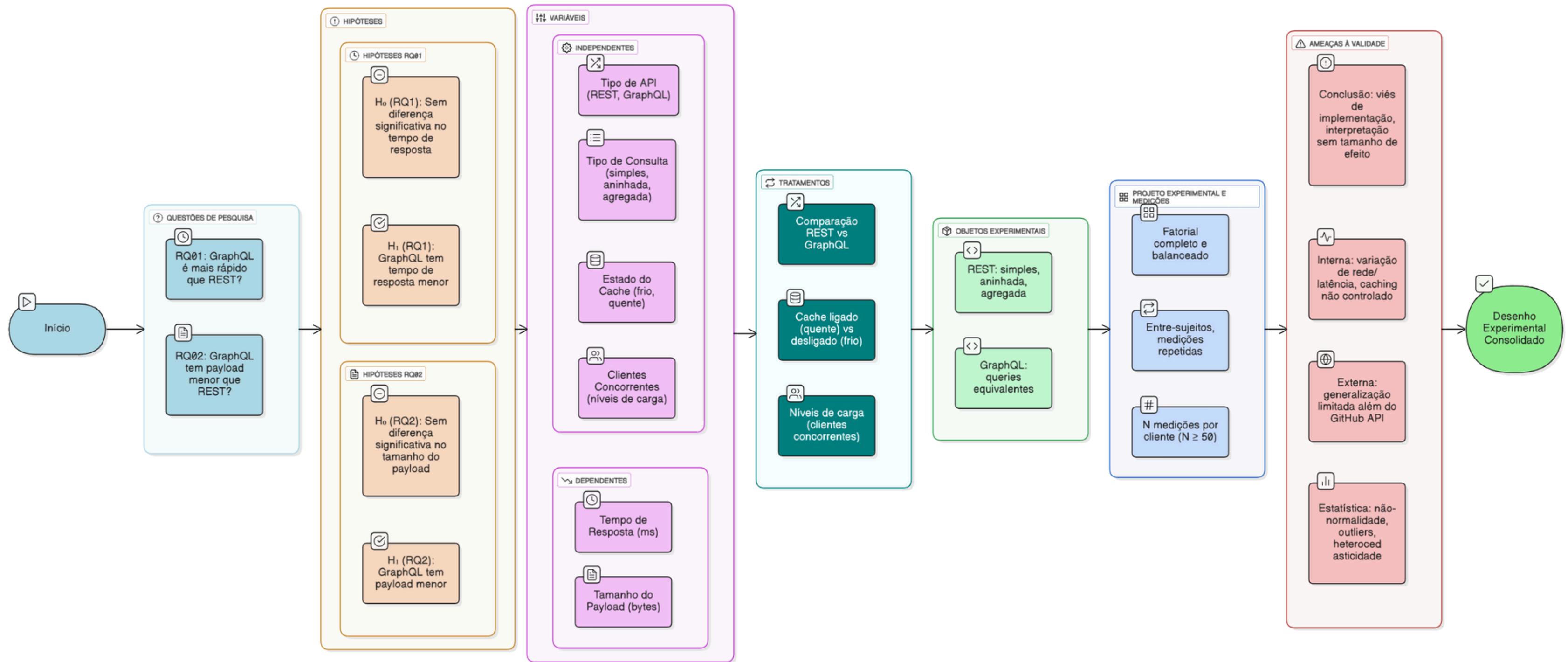


Figura 1 - Diagrama ilustrativo do desenho do experimento

# Métricas Avaliadas

Código	Métrica	Descrição
M01	<i>response_time_ms</i>	Tempo de resposta observado em milissegundos
M02	<i>payload_size_bytes</i>	Tamanho do payload observado em bytes
F01	<i>status_code</i>	Código HTTP de retorno
F02	<i>cache_state</i>	Estado de cache: <i>cold/warm</i>
F03	<i>concurrent_clients</i>	Número de clientes concorrentes
F04	<i>query_type</i>	Tipo de consulta: <i>simple/nested/aggregated</i>
F05	<i>api_type</i>	Tipo de API: REST/GraphQL

Também tivemos duas métricas adicionais: **(A01)** tamanho de efeito (Delta de Cliff) para medir quão grande é a diferença entre ambos os grupos e **(A02)** distribuição de Shapiro para verificação da distribuição dos dados.

# Cálculos e Testes Estatísticos

01

## Comparação REST vs GraphQL:

- Usamos t-test quando os dados eram normais; caso contrário, usamos Mann-Whitney, que é o equivalente não paramétrico.

02

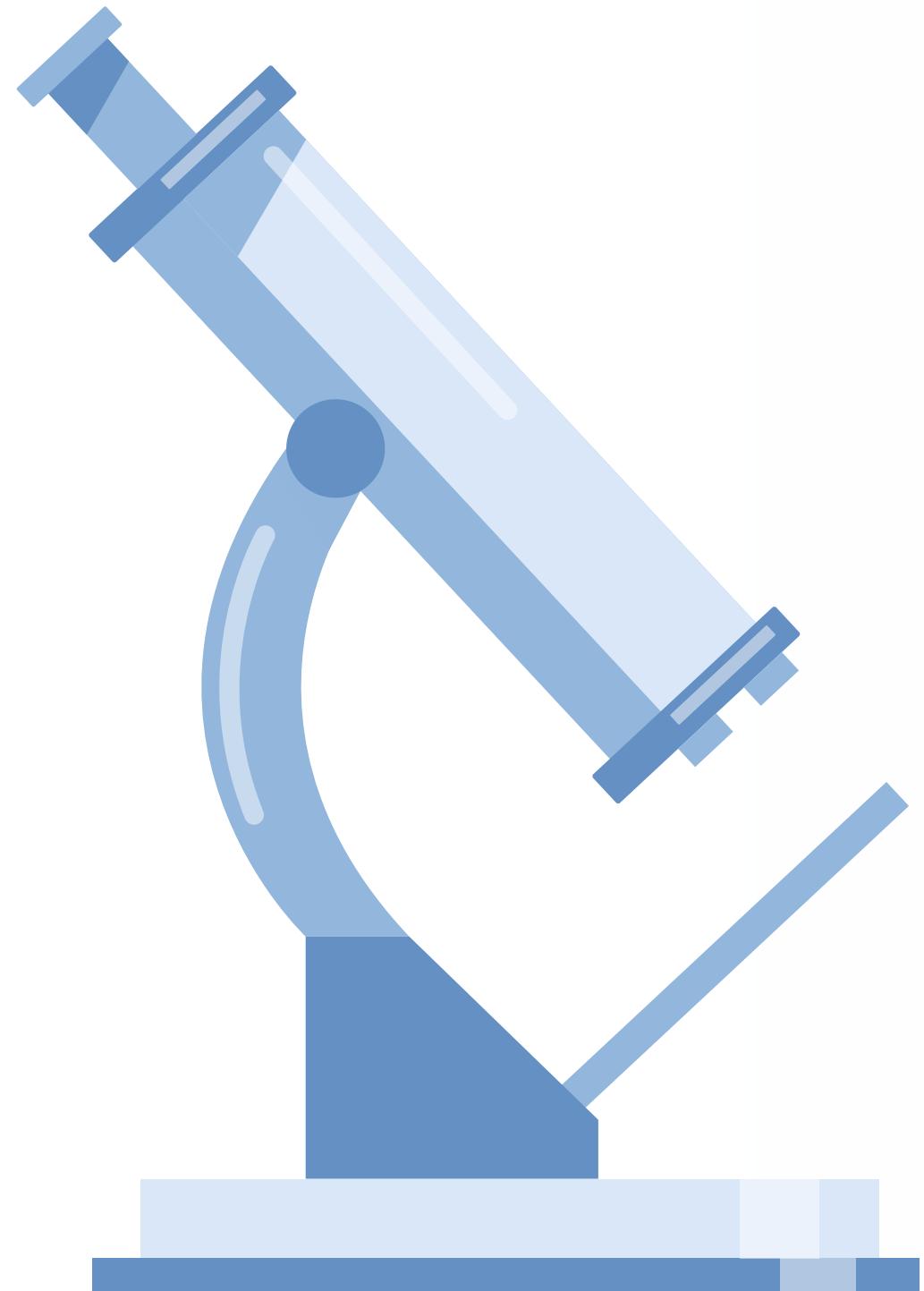
## Estatísticas analisadas:

- Observamos média, mediana, desvio padrão e percentis para entender o comportamento geral dos dados.

03

## Tamanho do efeito:

- Calculamos o Delta de Cliff, que indica quão grande é a diferença entre os grupos (não só se ela existe).

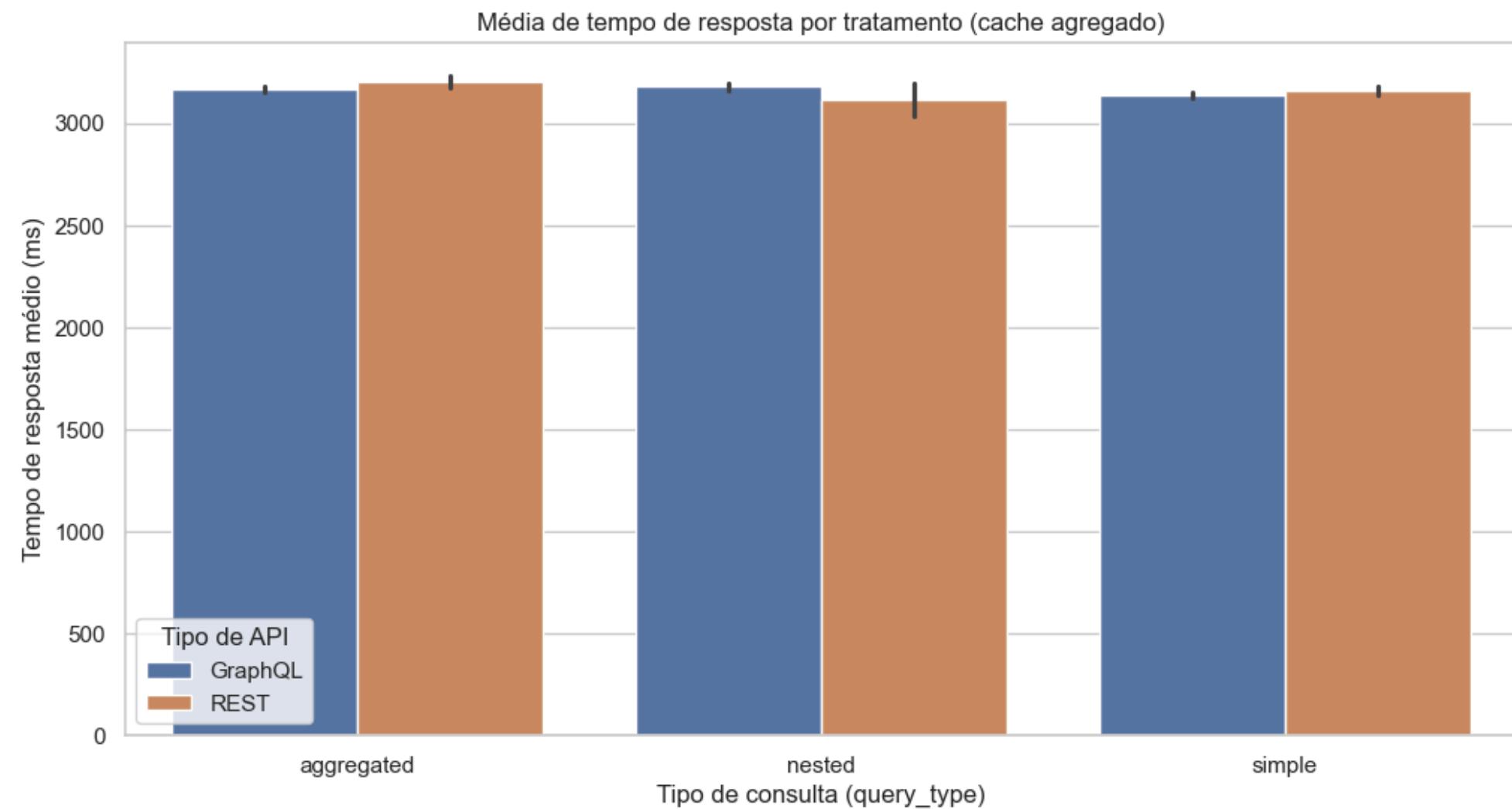


# Resultados

- Analisamos mais de **73 mil requisições REST vs. GraphQL**.
- Consideramos o *tempo de resposta* (RQ1) e *tamanho de payload* (RQ2).
- Usamos testes estatísticos + Mann-Whitney e tamanho de efeito (Cliff's Delta).

# Gráfico 1: Médias por Tipo de Consulta

- Os tempos médios são **muito próximos** entre REST e GraphQL, na faixa de 3,1 a 3,2 segundos.
- Variação **pequena** entre *simple*, *nested* e *aggregated*.
- Intervalos de erro se sobrepõem: **não há diferença visível de desempenho**.

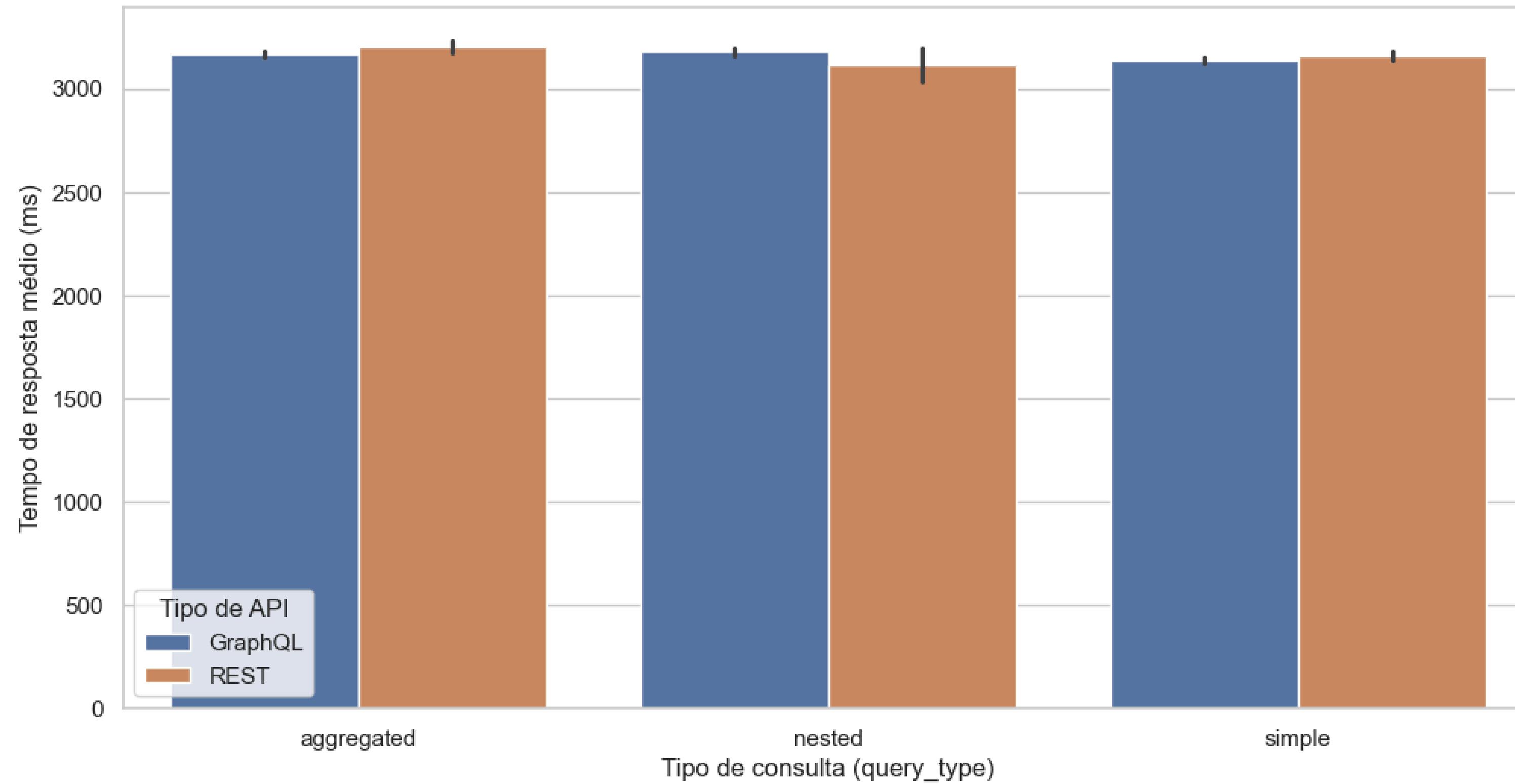


Tipos de consulta:

- *Simple* — consulta direta a um único recurso.
- *Nested* — consulta que inclui relacionamentos (ex.: repositório → issues).
- *Aggregated* — consulta que agrupa ou combina múltiplas informações em uma única chamada.

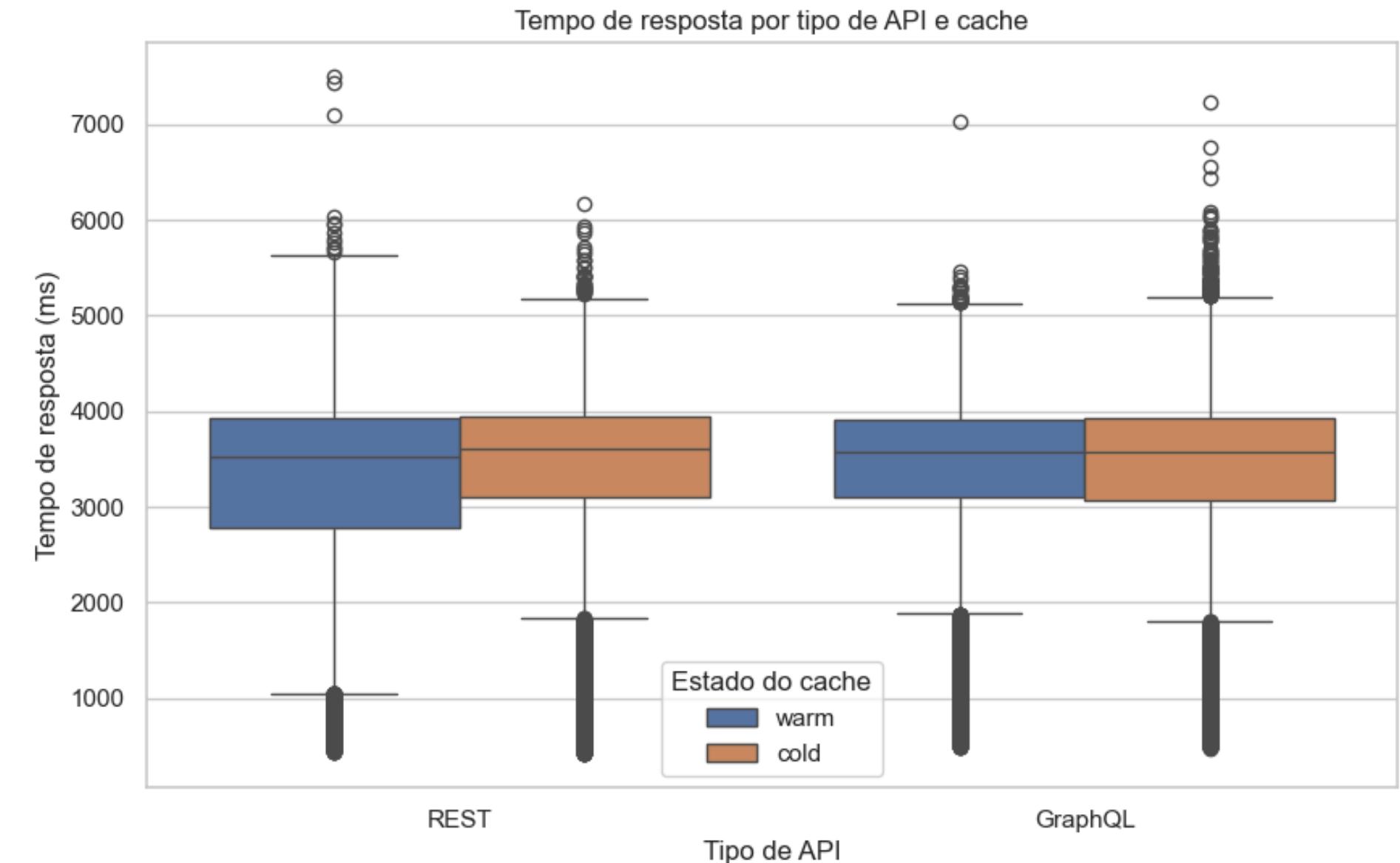


Média de tempo de resposta por tratamento (cache agregado)



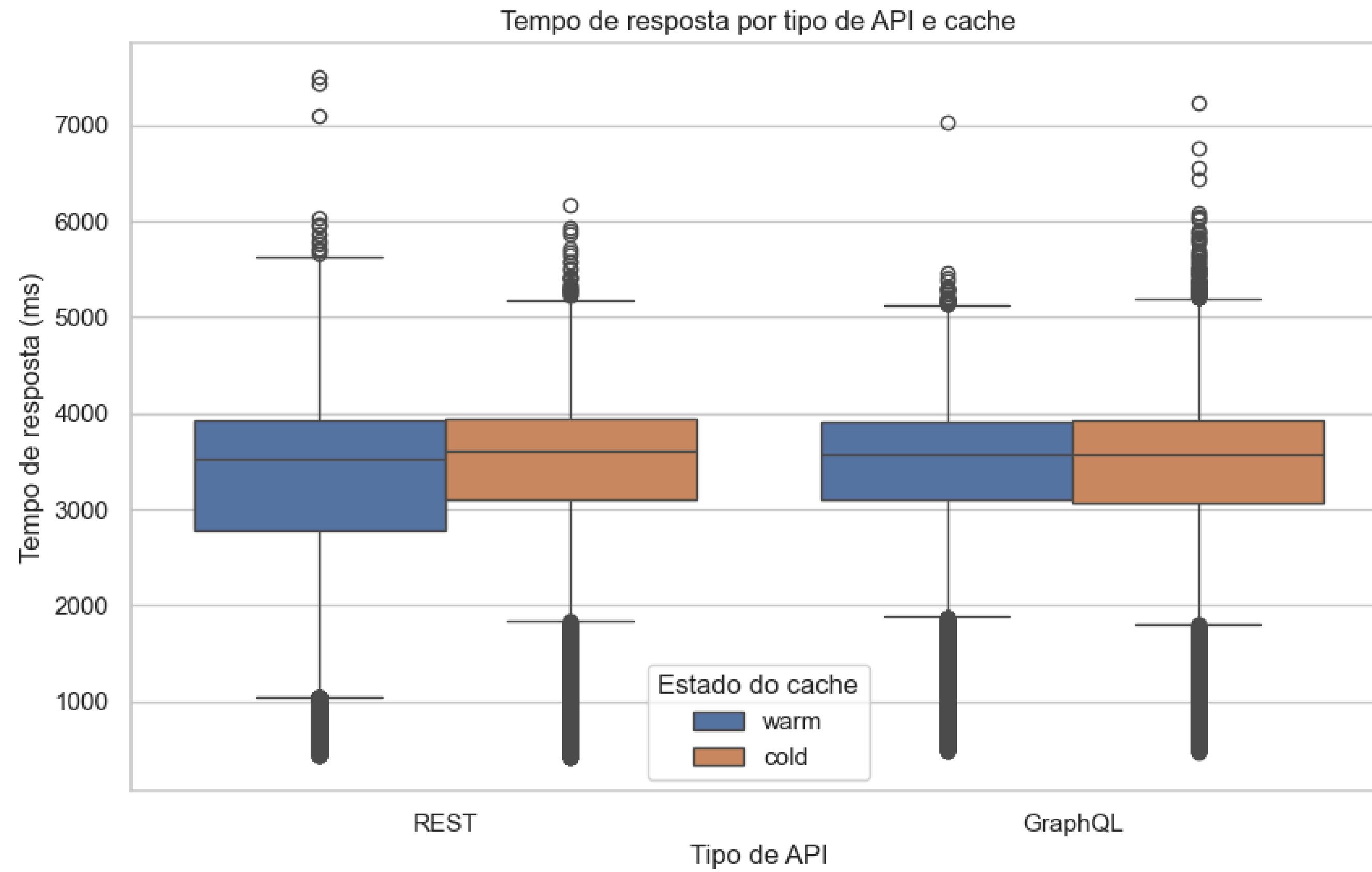
## Gráfico 2: Distribuição dos Tempos de Resposta

- Distribuições de REST e GraphQL **são praticamente iguais**.
- Warm vs. cold também não muda significativamente.
- Muitos outliers mostram **alta variabilidade externa** da API do GitHub.



O boxplot reforça que a latência do GitHub domina o comportamento: há grande sobreposição entre REST e GraphQL, e os estados de cache mudam pouco o resultado. Outliers altos existem em todos os cenários.

!



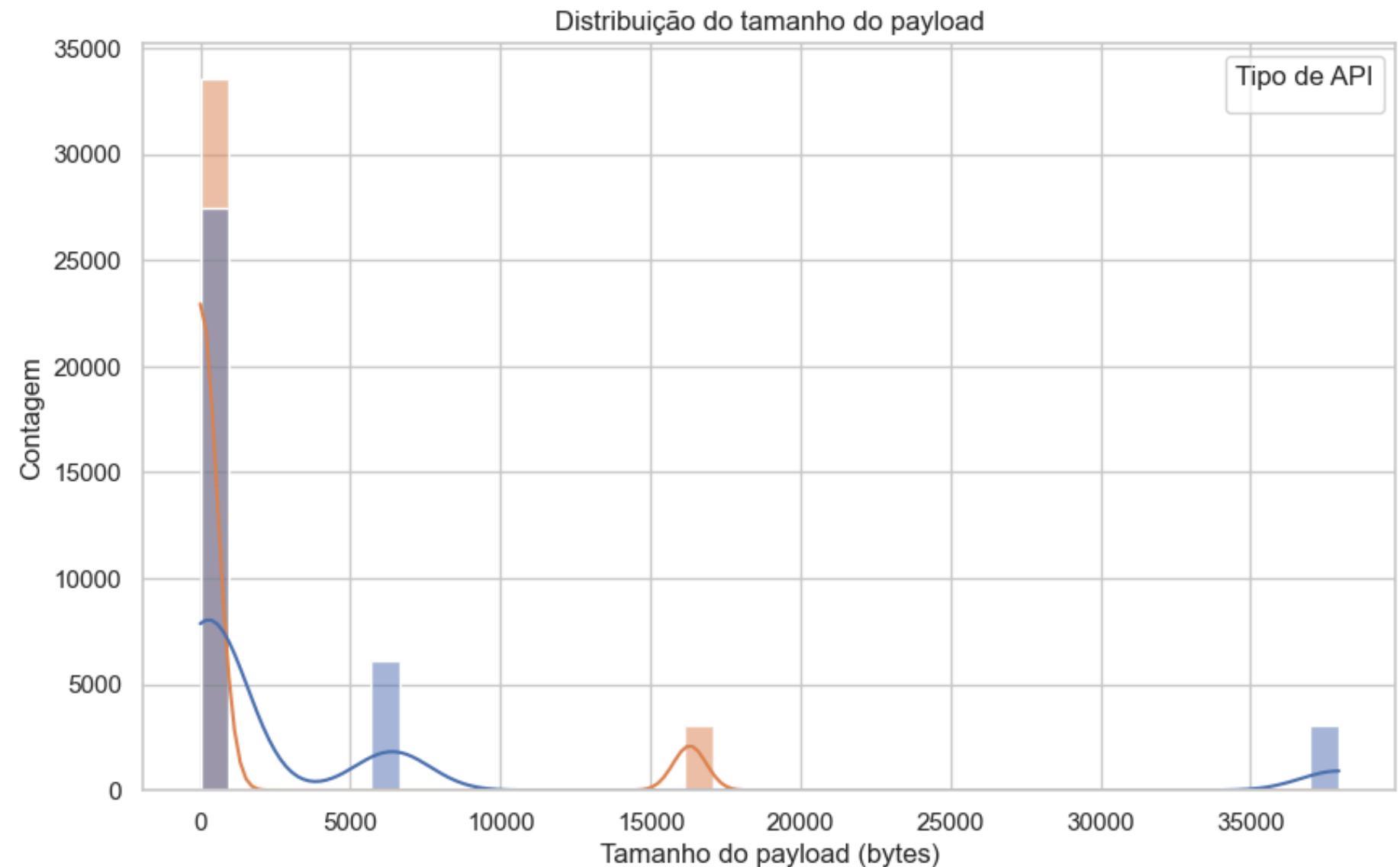
# Resultado Estatístico para RQ1

- Teste usado: **Mann-Whitney (dados não normais)**.
- Resultado: **p = 0.9907 → não rejeita  $H_0$** .
  - O p-valor alto indica que os tempos de REST e GraphQL não diferem de forma confiável;
- Tamanho de efeito: **Cliff's delta ≈ 0.03** (diferença desprezível).
  - O tamanho de efeito quase zero confirma que, na prática, eles são igualmente rápidos.
- **Conclusão: Não há evidência de que GraphQL seja mais rápido.**

A hipótese nula  $H_0$  foi **mantida** enquanto a hipótese alternativa  $H_1$  foi **recusada**.

## Gráfico 3: Distribuição do Tempo de Payload

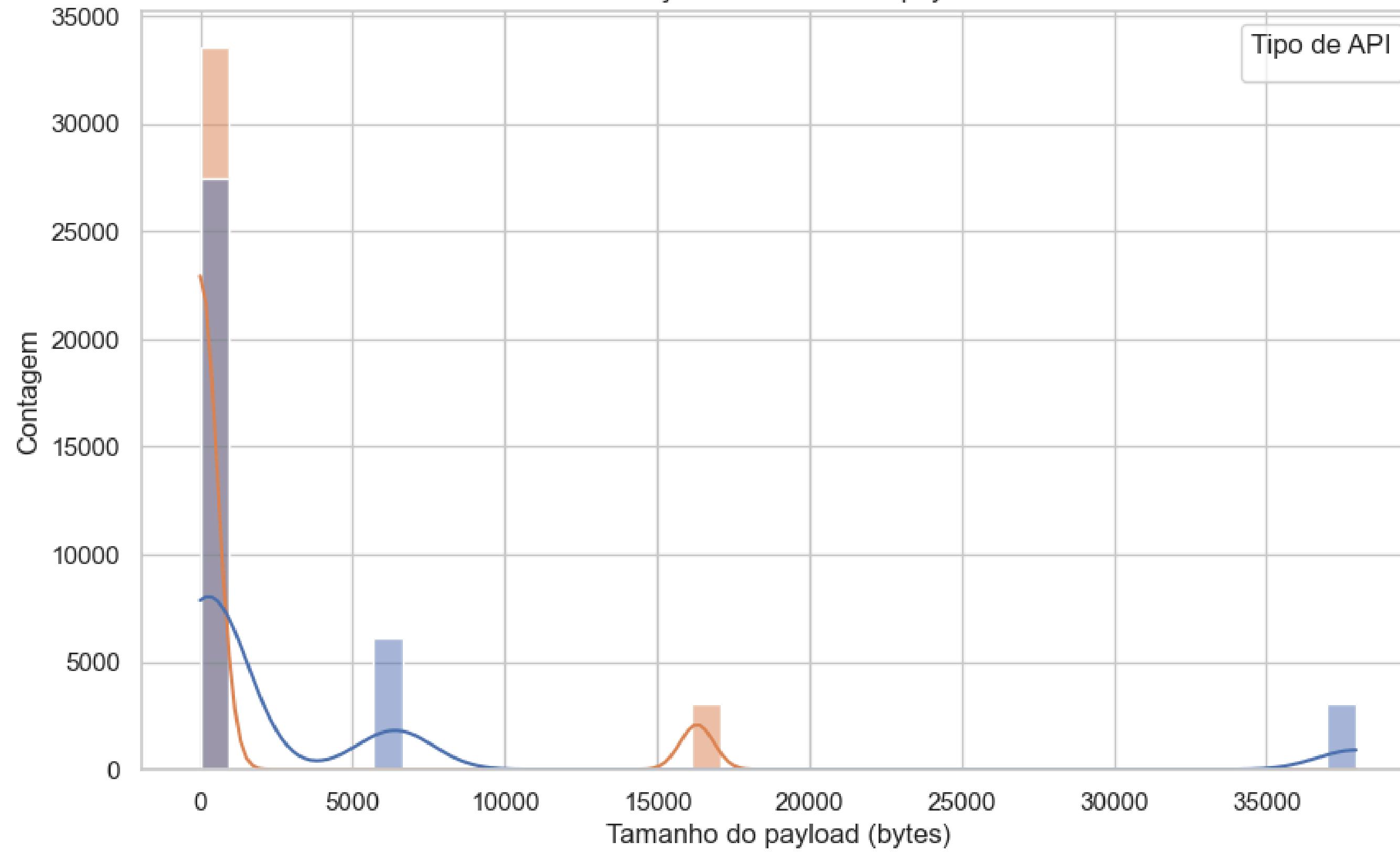
- GraphQL tende a gerar payloads menores.
- REST apresenta caudas longas de respostas muito maiores.
- Mediana de payload é 0 B, mas média é puxada por casos extremos.



A maior parte das respostas tem payload pequeno ou zero, mas o REST possui casos com respostas bem maiores, o que amplia a média. A distribuição do GraphQL é mais concentrada em tamanhos reduzidos.



Distribuição do tamanho do payload

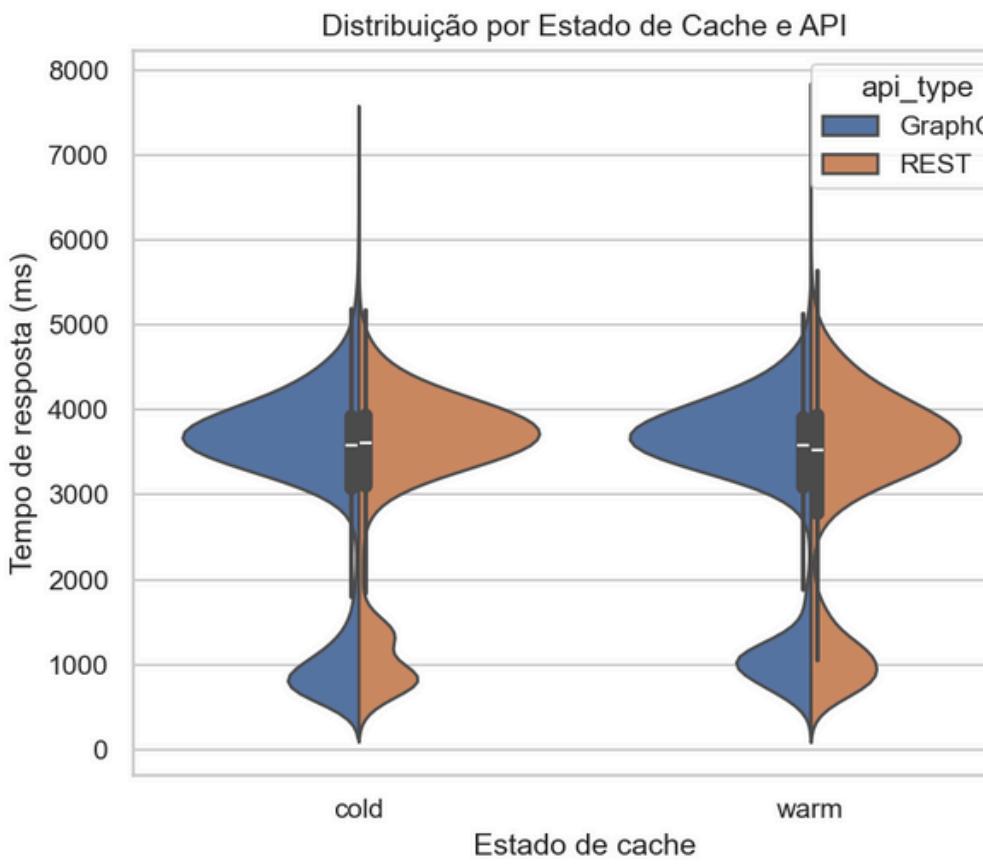
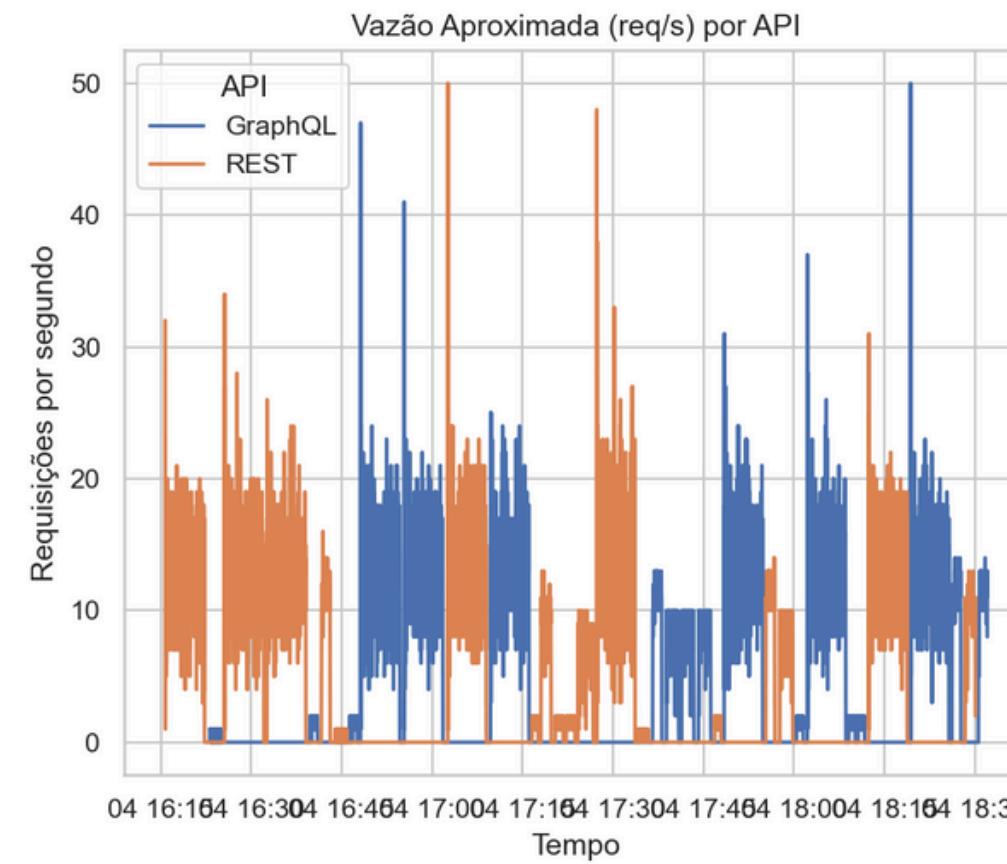
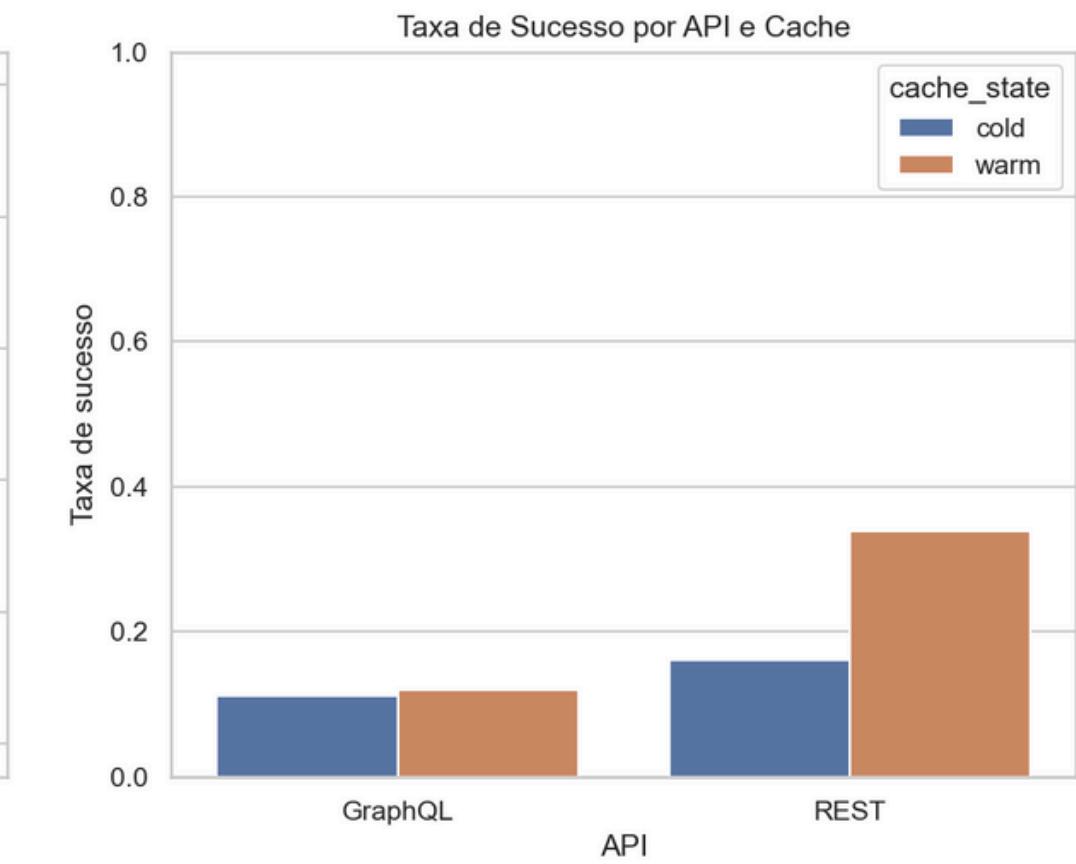
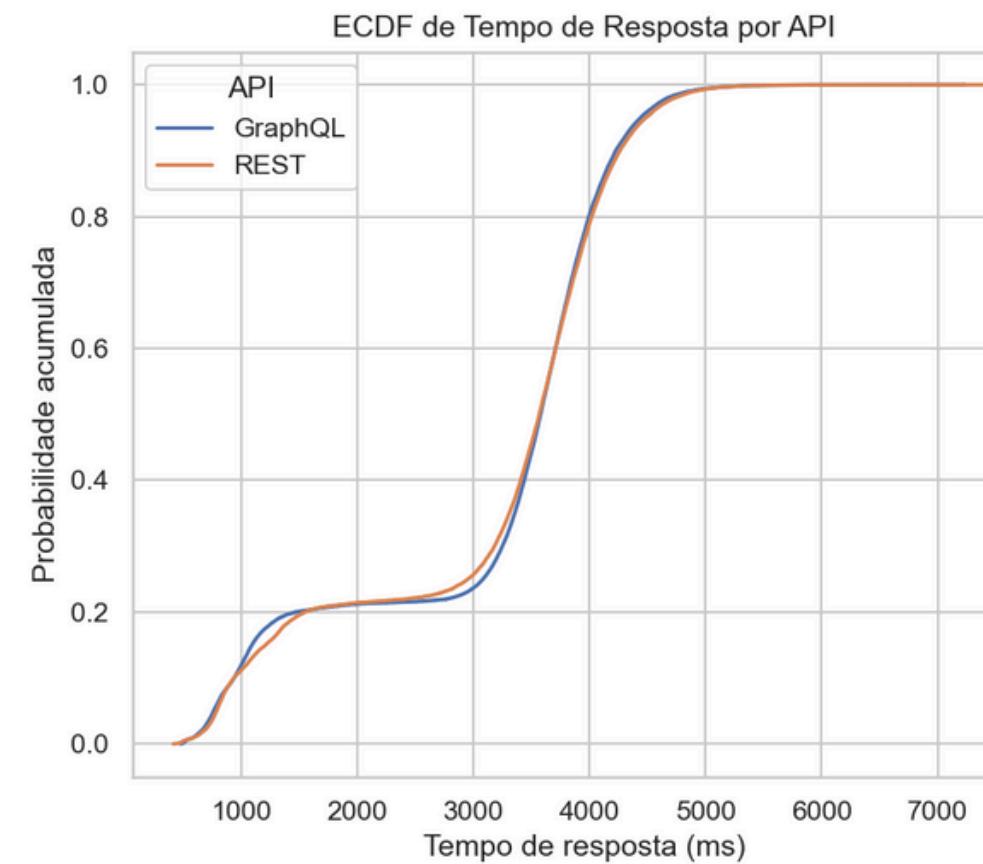
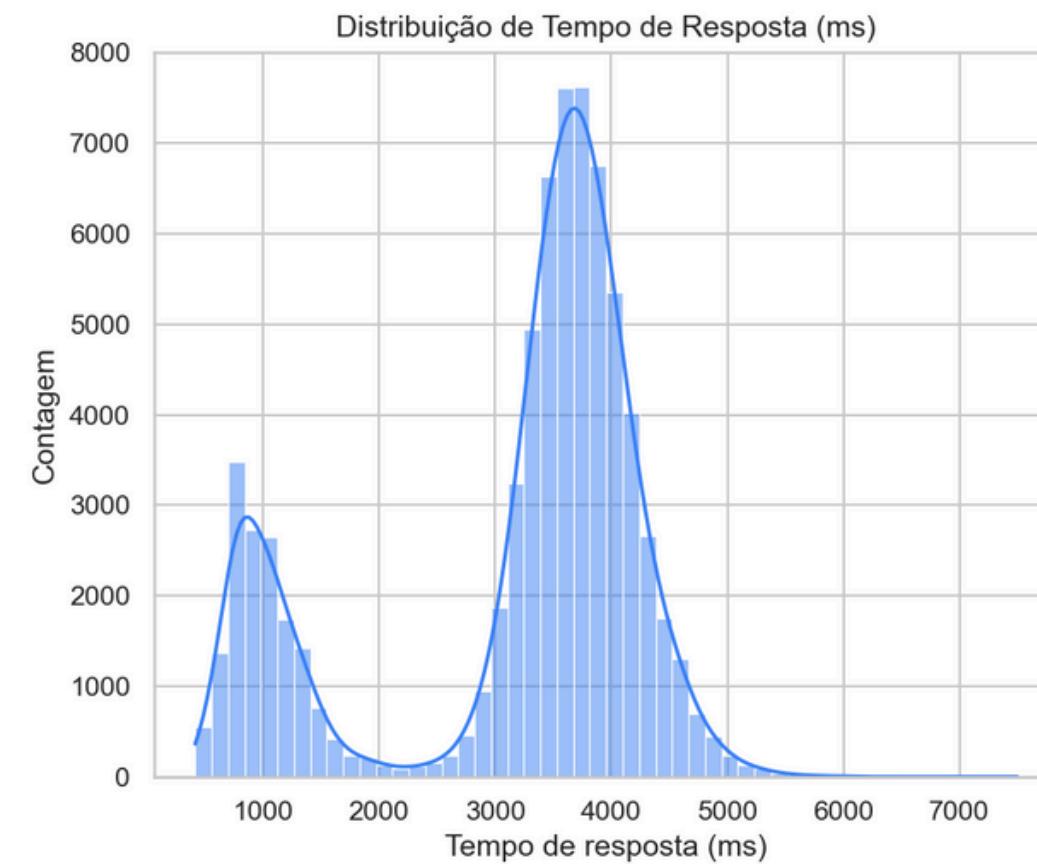


# Resultado Estatístico para RQ2

- Teste usado: **Mann-Whitney**.
- Resultado:  **$p \ll 0.001 \rightarrow$  rejeitamos  $H_0$** .
  - *O p-valor muito baixo mostra que a diferença é real.*
- Tamanho de efeito: **Cliff's delta  $\approx 0.53$  (moderado a forte)**.
  - O tamanho de efeito moderado/forte indica que GraphQL envia significativamente menos dados que REST.
- **Conclusão: GraphQL retorna menos dados na maior parte das requisições.**

A hipótese nula  $H_0$  foi **rejeitada** enquanto a hipótese alternativa  $H_1$  foi **suportada**.

# Gráficos Adicionais



\***ECDF = (Função de Distribuição Cumulativa Empírica)** - A análise da ECDF do tempo de resposta indica que GraphQL e REST apresentam distribuições muito semelhantes. Observa-se uma leve vantagem do GraphQL nos percentis intermediários (p50–p90), sugerindo tempos de resposta marginalmente menores para a maioria das requisições. Entretanto, as caudas das distribuições convergem, indicando comportamento similar nos piores casos.

# Conclusão

- **RQ1:** O tempo total de resposta é controlado muito mais pela própria infraestrutura do GitHub (rede, servidores, latência interna) do que pelo fato de ser REST ou GraphQL.
- **RQ2:** GraphQL se mostrou mais eficiente em dados, principalmente pela capacidade de reduzir a quantidade de dados transferidos
- **Cache e concorrência não alteram as conclusões principais.**

**REST e GraphQL** mostraram desempenho **semelhante** em **tempo de resposta**, mas GraphQL apresenta **menor uso de banda** em muitos cenários. A escolha prática depende da necessidade de **economia de dados** ou **flexibilidade**.

# Obrigado!