

Desenho do Experimento

O estudo teve como objetivo investigar de forma empírica o esforço exigido de desenvolvedores ao implementarem consultas remotas usando GraphQL em comparação ao padrão REST. Para esse fim, a variável independente foi a tecnologia adotada (REST vs GraphQL) e a variável dependente foi o tempo gasto em minutos para implementar cada tarefa proposta. Formularam-se então duas hipóteses: a hipótese nula (H_0) que assume ausência de diferença estatisticamente significativa no tempo médio de desenvolvimento entre REST e GraphQL, e a hipótese alternativa (H_1) que prevê redução desse tempo ao usar GraphQL.

O experimento seguiu um desenho within-subject com contrabalanceamento de tarefas, de modo que cada participante implementou quatro consultas em REST e quatro em GraphQL, alternando a ordem para mitigar efeitos de aprendizado. As oito tarefas abrangeram cenários de busca de repositórios, busca de usuários e requisições específicas a um repositório ou usuário, todas acessando a API pública do GitHub. Após a execução, os tempos de conclusão foram comparados usando o teste de Wilcoxon com $p = 0,00055$, revelando uma redução da mediana de 9 para 6 minutos ao adotar GraphQL, com tamanho de efeito médio.

As principais ameaças à validade incluíram possíveis vieses de construct validity, mitigados por instruções claras para evitar interferências externas; threats à validade interna, atenuados pelo contrabalanceamento; e limitações de validade externa, decorrentes do uso de 22 estudantes e de um único serviço de API, o que recomenda cautela ao generalizar para desenvolvedores profissionais.

Preparação do Experimento

A preparação envolveu a montagem de um ambiente de desenvolvimento em Python, utilizando o IDE IDLE para a execução de um script que continha, em formato de comentários, a descrição de cada tarefa e variáveis do tipo string para armazenar as consultas (Listing 1). Ao rodar o script, o sistema validava automaticamente cada consulta, indicava seu sucesso ou falha e gravava registros de tempo com timestamps de início e fim, permitindo calcular a duração de cada implementação.

Para testar as consultas GraphQL, os participantes utilizaram o aplicativo GraphiQL, oficial do GitHub, que fornece suporte a auto-completar e warnings imediatos sobre parâmetros obrigatórios. Já as requisições REST puderam ser validadas diretamente em um navegador ou ferramenta de linha de comando, inspecionando os campos retornados e o cabeçalho Content-Length quando necessário.

Foi fornecido a todos acesso às documentações oficiais da API REST v3 e da API GraphQL v4 do GitHub, garantindo que cada desenvolvedor pudesse consultar endpoints, parâmetros e exemplos de payloads durante a execução das tarefas. Antes da coleta de dados, realizou-se um piloto com dois estudantes experientes em REST para ajustar o formato das tarefas e calibrar o script de validação, levando em conta comentários sobre clareza na instrução e adequação do log de execução.