

# Do GitHub para sua Pesquisa: extraindo dados usando a API GraphQL



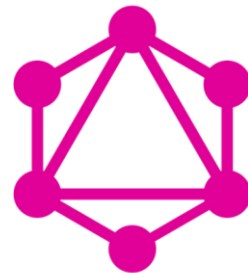
Carlos Eduardo de Carvalho Dantas

[carlooseduardodantas@iftm.edu.br](mailto:carlooseduardodantas@iftm.edu.br)

<https://carlooseduardoxp.github.io/>

Julyanara Rodrigues Silva

[julyasstudy@gmail.com](mailto:julyasstudy@gmail.com)



GraphQL

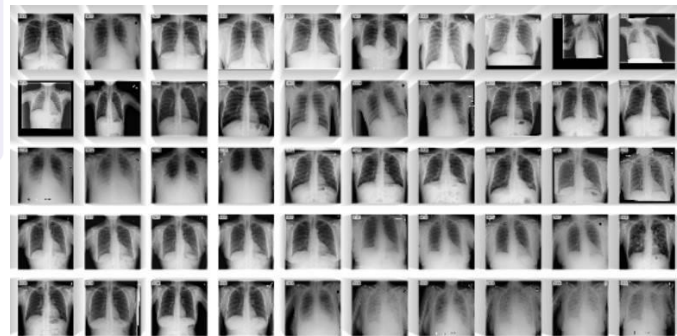
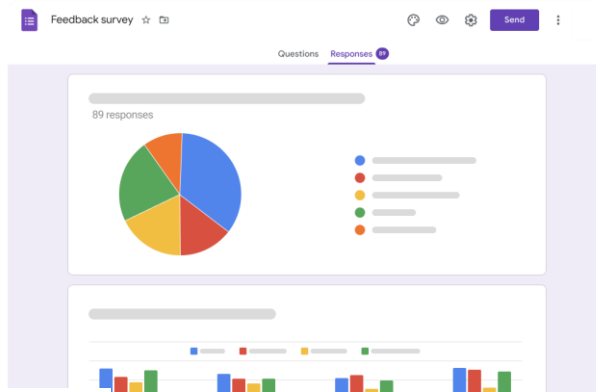
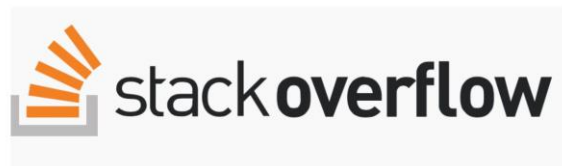
# O Caminho de uma Pesquisa de Impacto

---

- 1) **Identificação de um Problema ou Tema de Interesse:** O que você quer investigar?
- 2) **Formulação de Hipóteses:** Quais possíveis respostas ou teorias você quer testar?
- 3) **Definição de Perguntas de Pesquisa:** Que perguntas precisam ser respondidas para validar ou refutar as hipóteses?
- 4) **Construção ou Seleção do Dataset:** Como obter os dados necessários para responder às perguntas?
- 5) **Análise de Dados:** Como os dados serão interpretados para gerar resultados?
- 6) **Conclusões e Comunicação:** Quais são os insights e como eles serão apresentados?

# Conjunto de Dados (dataset)

- Fonte de onde serão extraídas as informações necessárias para responder às perguntas de pesquisa.



# Conjunto de Dados (dataset) – Exemplo 1



**XIII Congresso Brasileiro de  
Informática na Educação**

## **Aplicação de Métodos Ativos no Ensino de Análise e Projeto de Sistemas: Um Relato da Avaliação de Desempenho**

**Vitor de Souza Castro<sup>1</sup>, Sandro Ronaldo Bezerra Oliveira<sup>1</sup>**

<sup>1</sup>Programa de Pós-Graduação em Ciência da Computação – Instituto de  
Ciência Exatas e Naturais – Universidade Federal do Pará (UFPA) – Belém – PA – Brasil

vitor@unifesspa.edu.br, srbo@ufpa.br

**QP** Qual método ativo possui maior desempenho no ensino de análise e projeto de software?

### **4. Contexto do Experimento**

O experimento foi realizado na disciplina de Análise e Projeto de Sistemas de uma Instituição Federal de Ensino Superior da região norte do Brasil. A turma era composta por 38 alunos matriculados e 36 com frequência mínima de 75%.

# Conjunto de Dados (dataset) – Exemplo 2

## Rumo a uma Taxonomia de Observabilidade para Aplicações Baseadas em Microserviços

Francisco A. A. Gomes  
Universidade Federal do Ceará  
Crateús, Ceará, Brasil  
almada@crateus.ufc.br

Paulo A. L. Rego  
Universidade Federal do Ceará  
Fortaleza, Ceará, Brasil  
paulo@dc.ufc.br

Fernando A. M. Trinta  
Universidade Federal do Ceará  
Fortaleza, Ceará, Brasil  
fernando.trinta@dc.ufc.br




viços”. Portanto, este trabalho propõe uma **Questão de Pesquisa** a fim de delimitar o escopo do trabalho: *Quais são os diferentes domínios e categorias de observabilidade em microserviços?* Esta questão visa identificar o real propósito da pesquisa em como a observabilidade pode ajudar as aplicações baseadas em microserviços.

**Busca:** Dentro desta etapa, uma *String* de busca foi aplicado em quatro Mecanismos de Busca: *IEEE*, *ACM*, *Scopus* e *Web of Science*. Logo após, os trabalhos retornados foram coletados. Utilizamos palavras-chave, em inglês, mais relacionadas ao tema “Observabilidade em Microserviços” para gerar a *string* de busca. Após analisar

# Conjunto de Dados (dataset) – Exemplo 3



## CROKAGE: effective solution recommendation for programming tasks by leveraging crowd knowledge

Rodrigo Fernandes Gomes da Silva<sup>1</sup> · Chanchal K. Roy<sup>2</sup> ·  
Mohammad Masudur Rahman<sup>2</sup> · Kevin A. Schneider<sup>2</sup> · Klérisson Paixão<sup>1</sup>  
Carlos Eduardo de Carvalho Dantas<sup>1</sup> · Marcelo de Almeida Maia<sup>1</sup> 

### 3.1 Corpus Preparation

In order to deliver appropriate solutions from Stack Overflow against a programming task (i.e., query), we need to construct the domain specific knowledge base (Fig. 2a). We collect a total of 10,248,824 questions and answers from Stack Overflow Q&A site<sup>5</sup> related to three programming languages, as shown in Table 1. We separate posts related to each program-

# Por que construir um dataset usando dados do GitHub?

222 milhões de  
repositórios

## Total Public Repositories on GitHub [BETA]

Showing growth over time (Since we started tracking)

Current Total

222 623 029



Fonte: <https://gitcharts.com/>

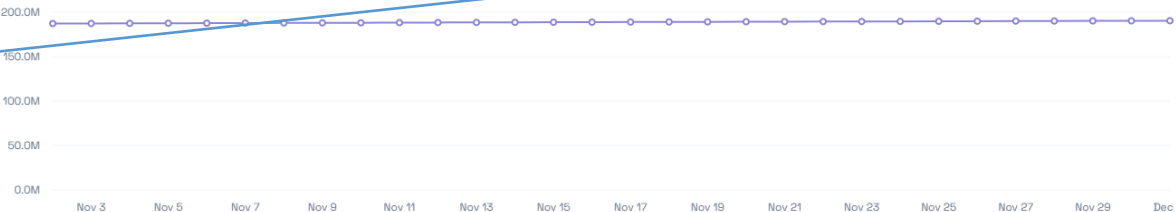
190 milhões de  
usuários

## Total Users on GitHub

Showing growth over time (Since we started tracking)

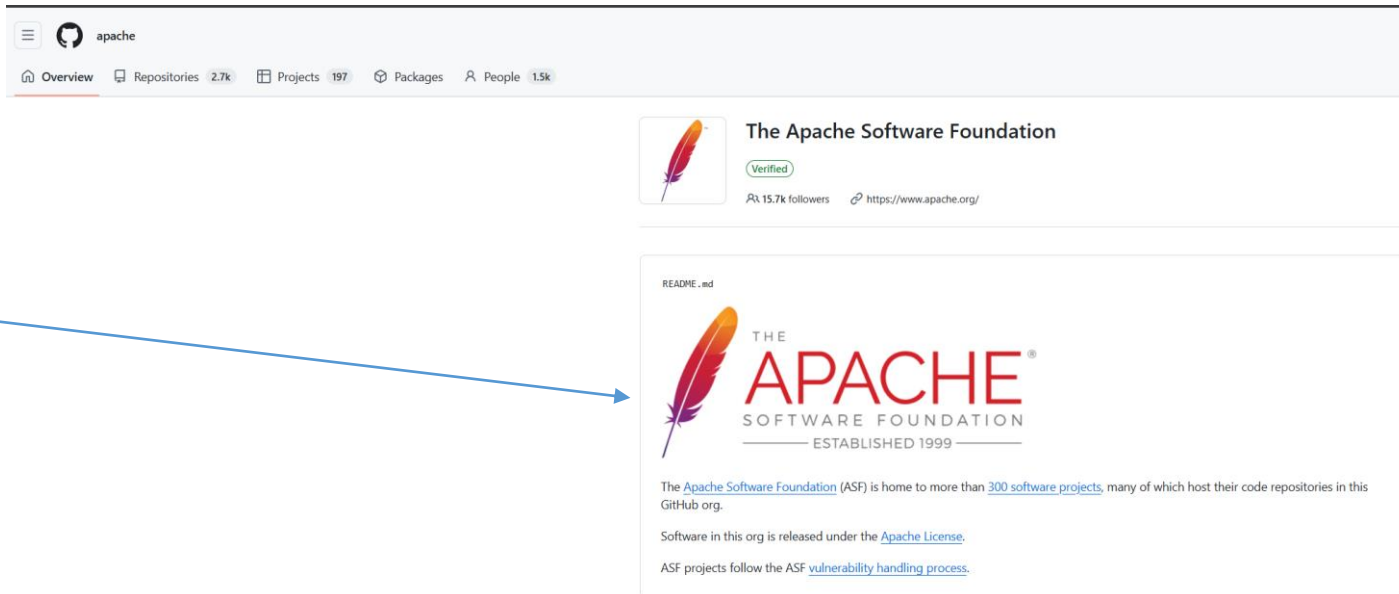
Current Total

190 407 007



# Por que construir um dataset usando dados do GitHub?

Muitas das principais ferramentas de código aberto estão hospedadas no Github, onde recebem atualizações constantes



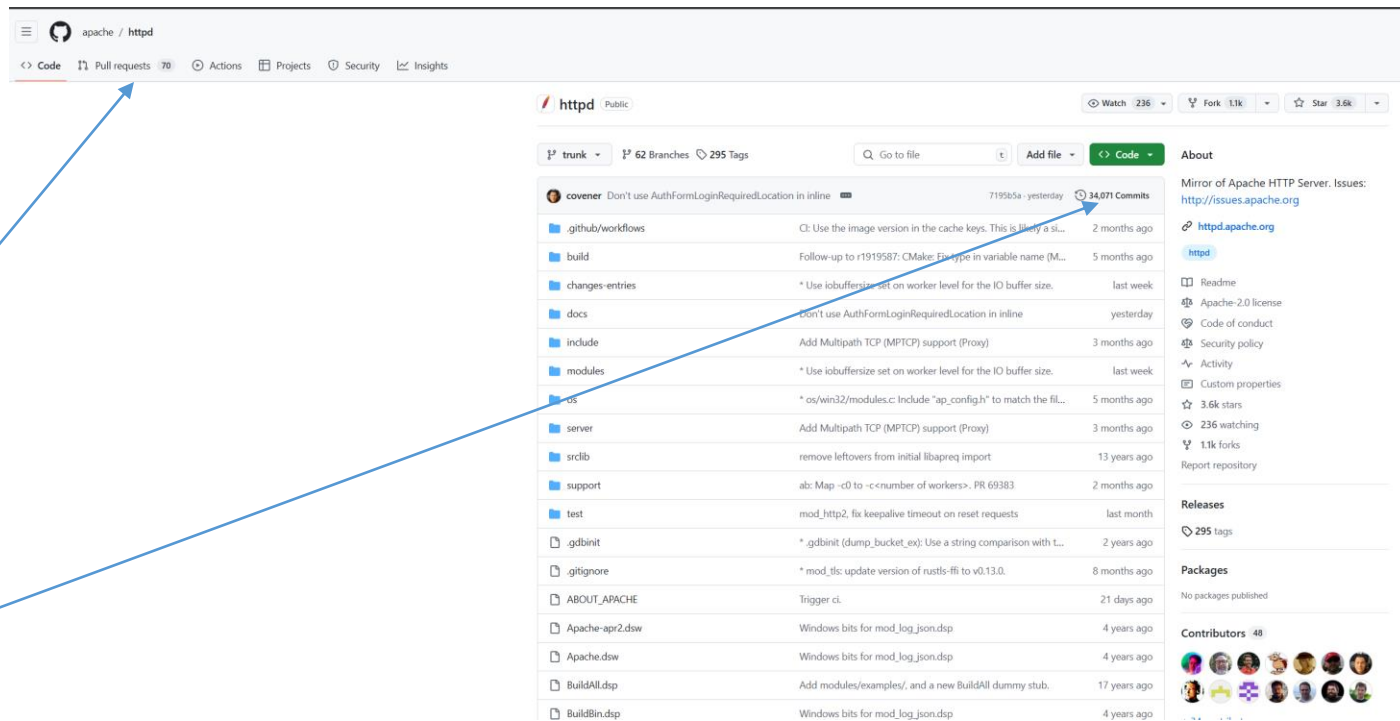


# Por que construir um dataset usando dados do GitHub?

Git se tornou um padrão para versionamento de código-fonte

**Pull Requests:**  
Fluxo colaborativo para revisão e integração de mudanças.

**Commits:**  
Registro estruturado e rastreável das alterações no código.



The screenshot displays the GitHub interface for the 'apache / httpd' repository. At the top, navigation tabs include 'Code', 'Pull requests' (with 70 items), 'Actions', 'Projects', 'Security', and 'Insights'. The repository name 'httpd' is shown as 'Public'. On the right, statistics indicate 236 watchers, 1.1k forks, and 3.6k stars. Below this, a table lists recent commits by user 'covener', including details like commit messages (e.g., 'Don't use AuthFormLoginRequired.Location in inline'), commit hashes (e.g., '7195b5a'), and commit dates (e.g., 'yesterday'). A blue arrow points from the 'Pull Requests' tab in the top navigation to the 'Pull requests' section on the right. Another blue arrow points from the 'Commits' section in the left sidebar to the commit list table.

# Conjunto de Dados (dataset) – Exemplo 4

## What Developers Ask to ChatGPT in GitHub Pull Requests? an Exploratory Study

Julyanara R. Silva<sup>1</sup>, Carlos Eduardo C. Dantas<sup>1</sup>, Marcelo A. Maia<sup>2</sup>

<sup>1</sup>Instituto Federal de Ciência e Tecnologia do Triângulo Mineiro (IFTM)  
Campus Uberlândia Centro – Uberlândia, MG – Brazil

<sup>2</sup>Universidade Federal de Uberlândia (UFU) – Uberlândia, MG – Brazil

12th Workshop on Software  
Visualization, Evolution and  
Maintenance

Co-located with CBSOFT, 30th of September 2024 (In Person)



### RQ #1) What do developers request on ChatGPT to solve Pull Requests?

The goal of this step is to identify potential merged PRs where ChatGPT was likely used to assist developers. We began by writing a script that utilized the GitHub GraphQL API<sup>2</sup> to query for non-forked merged PRs, mentioning the ChatGPT share link: “chat.openai.com/share”. We first executed this query in May 2024 to collect and work

# Conjunto de Dados (dataset) – Exemplo 5

## How do Developers Improve Code Readability? An Empirical Study of Pull Requests

1<sup>st</sup> Carlos Eduardo C. Dantas  
*Federal University of Uberlândia*  
Uberlândia, Brazil  
carlosoeduardodantas@iftm.edu.br

2<sup>nd</sup> Adriano M. Rocha  
*Federal University of Uberlândia*  
Uberlândia, Brazil  
adriano.rocha@ufu.br

3<sup>rd</sup> Marcelo A. Maia  
*Federal University of Uberlândia*  
Uberlândia, Brazil  
marcelo.maia@ufu.br



- **RQ #1)** What types of code readability improvements do developers describe and perform in Pull Requests (PRs)?

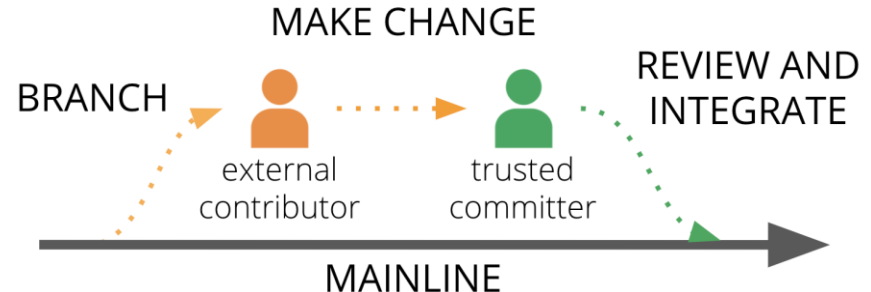
Then, we mined merged PRs approved by the reviewer(s) for each of these 4,179 Java engineering repositories, using GitHub APIs such as the GitHub REST API<sup>4</sup> and GitHub GraphQL API<sup>5</sup>. To identify merged PR candidates related to

# Que tipos de dataset podem ser criados a partir do GitHub?

## • Avaliação do Código-Fonte Produzido por Desenvolvedores

- **Qualidade do Código:** Clareza, organização e ausência de duplicações.
- **Manutenibilidade:** Facilidade de entender e modificar o código no futuro.
- **Performance:** Eficiência em termos de tempo de execução e uso de recursos.
- **Conformidade:** Adesão aos padrões de codificação e boas práticas.

- Snapshot específicos
- Antes ou depois de commits/pull requests



# Pra que avaliamos o código-fonte produzido por desenvolvedores?

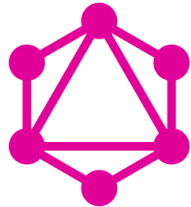
- Aperfeiçoar Sistemas de Recomendação
- Reconhecer padrões
- Identificar Melhorias no Processo de Desenvolvimento
- Prevenir e Resolver Bugs



# Como extrair dados do GitHub para construir o dataset?

---

- GraphQL é uma API que permite criar consultas personalizadas para buscar os dados desejados no GitHub
- Os dados extraídos podem ser armazenados em arquivos CSV ou inseridos em bancos de dados, possibilitando a criação de um dataset estruturado.
- **Link documentação:** <https://docs.github.com/en/graphql>



GraphQL

# Query 1: Exemplo Prático de Consulta com GraphQL - Repositories

Início de  
uma  
consulta  
graphql

```
query {  
  search(query: "language:Python stars:>50", type: REPOSITORY, first: 5) {  
    edges {  
      node {  
        ... on Repository {  
          name  
          owner {  
            login  
          }  
          stargazerCount  
          url  
        }  
      }  
    }  
  }  
}
```

Edge – lista de dados  
paginados, retornados  
pela consulta  
Node – dados que  
queremos acessar

Query específica. Retorna os primeiros  
5 repositórios Python com pelo menos  
50 estrelas

.. On Repository – significa  
que serão retornados campos  
que estão dentro de  
Repository, como seu nome,  
o owner, url e quantidade de  
estrelas.

# Executando a consulta do GraphQL no GraphQL Explorer

O github possui uma interface para executarmos as consultas GraphQL

GitHub GraphQL API

Signed in as carloseduardoxp. You're ready to explore! [Sign out](#)

Heads up! GitHub's GraphQL Explorer makes use of your real, live, production data.

Graph/QL

```
1 query {
2   search(query: "language:Python stars:>50", type: REPOSITORY) {
3     edges {
4       node {
5         ... on Repository {
6           name
7           owner {
8             login
9           }
10          stargazerCount
11          url
12        }
13      }
14    }
15  }
16 }
```

```
{
  "data": {
    "search": {
      "edges": [
        {
          "node": {
            "name": "public-apis",
            "owner": {
              "login": "public-apis"
            },
            "stargazerCount": 319119,
            "url": "https://github.com/public-apis/public-apis"
          }
        },
        {
          "node": {
            "name": "system-design-primer",
            "owner": {
              "login": "donnemartin"
            },
            "stargazerCount": 278113,

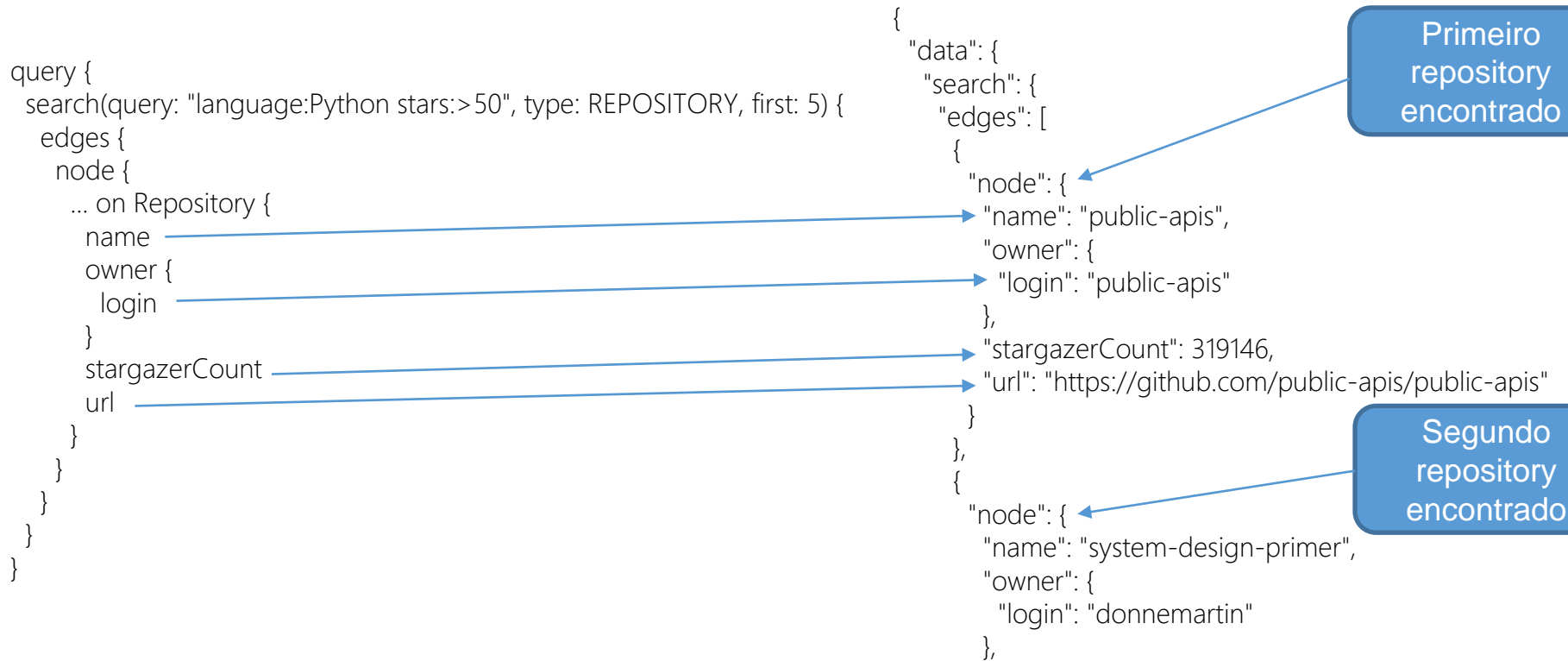
```

Variables Headers

Fonte: <https://docs.github.com/en/graphql/overview/explorer>



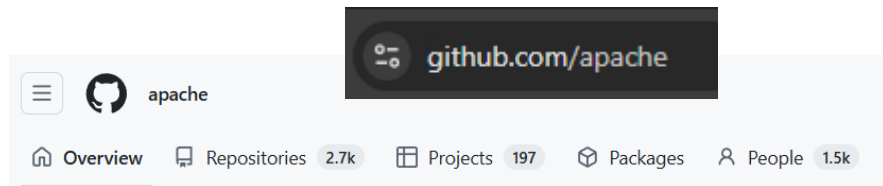
# Executando a consulta do GraphQL no GraphQL Explorer



## Query 2: Exemplo Prático de Consulta com GraphQL

Está buscando por issues abertas dentro dos 2.700 repositórios apache, e que contenham a string de busca “duplicate code”

```
query {  
  search(query: "is:issue is:open org:apache duplicate code", type: ISSUE, first: 5) {  
    edges {  
      node {  
        ... on Issue {  
          title  
          bodyText  
          url  
          createdAt  
          repository {  
            nameWithOwner  
          }  
        }  
      }  
    }  
  }  
}
```



Agora está retornando campos relacionados a Issues, como o título da issue, a data de criação, a url, o texto digitado no corpo da issue, etc.

# Executando a consulta do GraphQL no GraphQL Explorer

```
{
  "data": {
    "search": {
      "edges": [
        {
          "node": {
            "title": "[Improvement] too many duplicated code in Gravitino error handler",
            "bodyText": "What would you like to be improved?¶\nThere are many duplicated code in error handler in both server side and client side, we can move the general exception handling in BaseExceptionHandler¶\n    if (e instanceof IllegalArgumentException) {¶\n        return Utils.illegalArguments(errorMsg, e);¶\n    } else if (e instanceof NotFoundException) {¶\n        return Utils.notFound(errorMsg, e);¶\n    } else if (e instanceof NotInUseException) {¶\n        return Utils.notInUse(errorMsg, e);¶\n    } else {¶\n        return super.handle(op, credential, parent, e);¶\n    }¶\nHow should we improve?¶\nNo response",
            "url": "https://github.com/apache/gravitino/issues/5687",
            "createdAt": "2024-11-27T06:50:56Z",
            "repository": {
              "nameWithOwner": "apache/gravitino"
            }
          }
        }
      ]
    }
  },
}
```

[Improvement] too many duplicated code in Gravitino error handler

Open FANNG1 opened this issue last week · 4 comments



FANNG1 commented last week

Contributor · ...

What would you like to be improved?

There are many duplicated code in error handler in both server side and client side, we can move the general exception handling in `BaseExceptionHandler`

```
if (e instanceof IllegalArgumentException) {
    return Utils.illegalArguments(errorMsg, e);
} else if (e instanceof NotFoundException) {
    return Utils.notFound(errorMsg, e);
} else if (e instanceof NotInUseException) {
    return Utils.notInUse(errorMsg, e);
} else {
    return super.handle(op, credential, parent, e);
}
```

How should we improve?

No response



## Query 3: Exemplo Prático de Consulta com GraphQL

Está buscando por discussões que envolvam a “String” ChatGPT

```
query {  
  search(query: "is:discussion ChatGPT", type: DISCUSSION, first: 5) {  
    edges {  
      node {  
        ... on Discussion {  
          title  
          url  
          createdAt  
          repository {  
            nameWithOwner  
          }  
          category {  
            name  
          }  
        }  
      }  
    }  
  }  
}
```

Agora está retornando campos relacionados a discussões, como o título, url, a data de criação, o repositório, etc.

# Executando a consulta do GraphQL no GraphQL Explorer

```
{
  "data": {
    "search": {
      "edges": [
        {
          "node": {
            "title": "`chatgpt-tooling` discussions",
            "url": "https://github.com/edward-cates/chatgpt-tooling/discussions/1",
            "createdAt": "2024-12-03T01:20:41Z",
            "repository": {
              "nameWithOwner": "edward-cates/chatgpt-tooling"
            },
            "category": {
              "name": "Announcements"
            }
          }
        },
        {
          "node": {
            "title": "ChatGPT vs. You",
            "url": "https://github.com/prosyslab-classroom/cs424-program-reasoning/discussions/449",
            "createdAt": "2024-11-25T09:01:23Z",
```

## chatgpt-tooling discussions #1

edward-cates announced in Announcements



edward-cates 11 hours ago Maintainer

### Welcome!

We're using Discussions as a place to connect with other members of our community. We hope that you:

- Ask questions you're wondering about.
- Share ideas.
- Engage with other community members.
- Welcome others and are open-minded. Remember that this is a community we build together 🙌.

↑ 1

## ChatGPT vs. You #449

sujin0529 announced in Announcements



sujin0529 last week Maintainer

edited ...

안녕하세요.

여러분이 구현하신 그리고 구현하실 LIA 합성기와 SLIA 합성기를 ChatGPT로 대체한 결과를 Gradescope 순위표에 추가해두었습니다.

LIA 는 2 / 100 점을, SLIA 는 7 / 100 점을 받았습니다.

ChatGPT 이 해결하지 못한 LIA 합성 문제 1개를 공개하겠습니다.

ChatGPT 이 해당 문제에 대해 주어진 문법을 수정하지 않고 올바른 답변을 내도록 프롬프트를 작성해보세요.

성공하신 분에게는 소정의 상품을 드리도록 하겠습니다 🍷

성공하신분은 프롬프트와 ChatGPT의 답을 캡처하여 댓글에 달아주세요!

+ ) 성적에는 반영되지 않습니다!

# Higienização do dataset

Este exemplo destaca a importância de se realizar uma “higienização” na base de dados. Ou você entende o que está escrito?

## ChatGPT vs. You #449

sujin0529 announced in Announcements



sujin0529

last week

Maintainer

edited ...

안녕하세요.

여러분이 구현하신 그리고 구현하실 LIA 합성기와 SLIA 합성기를 ChatGPT로 대체한 결과를 Gradescope 순위표에 추가해두었습니다.

LIA 는 2 / 100 점을, SLIA 는 7 / 100 점을 받았습니다.

ChatGPT 이 해결하지 못한 LIA 합성 문제 1개를 공개하겠습니다.

ChatGPT 이 해당 문제에 대해 주어진 문법을 수정하지 않고 올바른 답변을 내도록 프롬프트를 작성해보세요.

성공하신 분에게는 소정의 상품을 드리도록 하겠습니다 🍷

성공하신분은 프롬프트와 ChatGPT의 답을 캡처하여 댓글에 달아주세요!

+) 성적에는 반영되지 않습니다!

Na documentação oficial é possível encontrar mais detalhes sobre os tipos de consultas que podem ser realizadas

## SearchType [↗](#)

Represents the individual results of a search.

### Values for SearchType

#### DISCUSSION

Returns matching discussions in repositories.

#### ISSUE

Returns results matching issues in repositories.

#### REPOSITORY

Returns results matching repositories.

#### USER

Returns results matching users and organizations on GitHub.

Fonte: <https://docs.github.com/en/graphql/reference/enums#searchtype>

# Ok, mas como vou construir o dataset?

Aqui temos 2 problemas:

- 1) Dataset não pode ser limitado a “first 5 registros”. Precisamos de todas as ocorrências possíveis. Entretanto o Graphql está limitado a retonar 100 registros

Rate limits and node limits for the GraphQL API  
The GitHub GraphQL API has limitations in place to protect against excessive or abusive calls to GitHub's servers.

## Node limit

To pass [schema](#) validation, all GraphQL API [calls](#) must meet these standards:

- Clients must supply a `first` or `last` argument on any [connection](#).
- Values of `first` and `last` must be within 1-100.
- Individual calls cannot request more than 500,000 total [nodes](#).

## Calculating nodes in a call

These two examples show how to calculate the total nodes in a call.

### 1 Simple query:

```
query {
  viewer {
    repositories(first: 10) {
      edges {
        repository: node {
          name

          issues(first: 10) {
            totalCount
            edges {
              node {
                title
                bodyHTML
              }
            }
          }
        }
      }
    }
  }
}
```



# Ok, mas como vou construir o dataset?

Aqui temos 2 problemas:

2) Embora o Graphl Explorer seja útil para validarmos o script, o formato de resposta em Json não é amigável. Precisávamos de algo mais tabulado como uma planilha.

O dataset do DevGPT foi construído para o Mining Challenge do MSR 2024, e possui 16,129 amostras de uso do ChatGPT no GitHub. Os dados estão no formato JSON

<https://2024.msrconf.org/track/msr-2024-mining-challenge?#Call-for-Mining-Challenge-Papers->

## DevGPT: Studying Developer-ChatGPT Conversations

Xiao, Tao<sup>1</sup>; Treude, Christoph<sup>2</sup>; Hata, Hideaki<sup>3</sup>; Matsumoto, Kenichi<sup>1</sup>

Show affiliations

DevGPT is a curated dataset which encompasses 16,129 prompts and ChatGPT's responses including 9,785 code snippets, coupled with the corresponding software development artifacts—ranging from source code, commits, issues, pull requests, to discussions and Hacker News threads—to enable the analysis of the context and implications of these developer interactions with ChatGPT.

### Files

DevGPT.zip	
DevGPT.zip	
README.md	10.4 kB
■ snapshot_20230727	
20230727_195816_hn_sharings.json	16.3 MB
20230727_195927_pr_sharings.json	21.5 MB
20230727_195941_issue_sharings.json	30.4 MB
20230727_195954_discussion_sharings.json	4.4 MB
20230727_200003_commit_sharings.json	15.5 MB
20230727_200102_file_sharings.json	224.0 MB
ChatGPT_Link_Sharing.csv	245.1 MB

Fonte: <https://zenodo.org/records/8248511>

# Ok, mas como vou construir o dataset?

---

## Soluções:

### **Solução para o Problema 1: Paginação dos Resultados**

A paginação é necessária para lidar com a quantidade limitada de registros (até 100) que podem ser retornados em uma única consulta à API GraphQL do GitHub. É importante ressaltar que a API do GitHub também impõe um limite de requisições por hora para cada usuário.

### **Solução para o Problema 2: Tabulação dos Dados**

Para organizar os dados retornados pela API, pode-se criar um script em Python (ou outra linguagem de programação) que tabule as informações conforme desejado. O script pode, por exemplo, exportar os dados para um arquivo CSV ou inseri-los em um banco de dados.

# Paginação dos Resultados

```
query {  
  search(query: "chat.openai.com/share is:pr is:merged in:title,body",type: ISSUE,first: 100) {  
    pageInfo {  
      endCursor  
      hasNextPage  
    }  
    issueCount  
    edges {  
      node {  
        ... on PullRequest {  
          url  
          title  
          createdAt  
          mergedAt  
          repository {  
            stargazerCount  
            isFork  
            primaryLanguage {  
              name  
            }  
          }  
        }  
      }  
    }  
  }  
}
```

issueCount  
retorna a  
quantidade total  
de registros

hasNextPage é  
um booleano  
informando se  
tem mais  
páginas

endCursor  
retorna uma  
string com um  
cursor para a  
próxima página

A query busca por pull requests  
em estado merged que possuem  
a string  
“chatgpt.openai.com/share” no  
título ou no corpo da pull request

# Paginação dos Resultados

```
{
  "data": {
    "search": {
      "pageInfo": {
        "endCursor": "Y3Vyc29yOjU=",
        "hasNextPage": true
      },
      "issueCount": 253,
      "edges": [
        {
          "node": {
            "url": "https://github.com/Mudlet/Mudlet/pull/7123",
            "title": "Fix a crash when double-clicking on a word to select it",
            "createdAt": "2024-02-07T19:29:39Z",
            "mergedAt": "2024-04-25T15:25:07Z",
            "state": "MERGED",
            "repository": {
              "stargazerCount": 740,
              "isFork": false,
              "primaryLanguage": {
                "name": "C++"
              }
            }
          }
        }
      ]
    }
  }
}
```

Existe uma próxima página,  
e deverá ser acessada com  
o cursor "Y3Vyc29yOjU="

A consulta da página  
anterior possui 253 pull  
requests de resultado

# Paginação dos Resultados

```
query {  
  search(query: "chat.openai.com/share is:pr is:merged in:title,body",type: ISSUE,first: 100, after:"Y3Vyc29yOjU=") {  
    pageInfo {  
      endCursor  
      hasNextPage  
    }  
    issueCount  
    edges {  
      node {  
        ... on PullRequest {  
          url  
          title  
          createdAt  
          mergedAt  
          repository {  
            stargazerCount  
            isFork  
            primaryLanguage {  
              name  
            }  
          }  
        }  
      }  
    }  
  }  
}
```

Para buscar os resultados da próxima página, deve-se usar o comando “after”  
Esse processo se repetirá até que  
hasNextPage = False

# Paginação dos Resultados

## Execução 1

```
{
  "data": {
    "search": {
      "pageInfo": {
        "endCursor":
          "Y3Vyc29yOjEwMA==",
        "hasNextPage":
          true
      },
      "issueCount":
        253,
      "edges": [
        {
          "node": {
            "url":
              "https://github.com/Mudlet/Mudlet/pull/7123"
          },
          "title":
            "Fix a crash when
            double-clicking on a
            word to select it",
```

## Execução 2

```
{
  "data": {
    "search": {
      "pageInfo": {
        "endCursor":
          "Y3Vyc29yOjIwMA==",
        "hasNextPage":
          true
      },
      "issueCount":
        253,
      "edges": [
        {
          "node": {
            "url":
              "https://github.com/Aligary/CS472_Group1/pull/55",
            "title":
              "Complex algorithm
              improved for
              readability and
```

## Execução 3

```
{
  "data": {
    "search": {
      "pageInfo": {
        "endCursor":
          "Y3Vyc29yOjI1Mw==",
        "hasNextPage":
          false
      },
      "issueCount":
        253,
      "edges": [
        {
          "node": {
            "url":
              "https://github.com/poki/netlib/pull/50",
            "title":
              "Postgress doesn't
              support LIMIT in
              DELETE",
```

# Tabulação dos dados

```
def run_query(query):  
    url = 'https://api.github.com/graphql'  
    headers = {'Authorization': f'Bearer {access_token}'}  
  
    response = requests.post(url, json={'query': query}, headers=headers)  
  
    if response.status_code != 200:  
        raise RuntimeError(  
            f"Falha na execução da query GraphQL.\n"  
            f"Status HTTP: {response.status_code}\n"  
            f"Resposta: {response.text}\n"  
            f"Query: {query}"  
        )  
  
    data = response.json()  
    if 'errors' in data:  
        raise RuntimeError(  
            f"Erros retornados pela API GraphQL.\n"  
            f"{data['errors']}\n"  
            f"Query: {query}"  
        )  
  
    return data
```

Para executar uma query na API do Github, precisamos de um `access_token`

# Criando um access token

The screenshot displays the GitHub user interface for a user named Carlos Eduardo. The left sidebar contains the user profile and navigation links. The main content area shows the 'Developer settings' page, which is highlighted with a red box. The 'Generate new token (classic)' option is highlighted with a red box, and a blue arrow points to it from the 'Tokens (classic)' link in the 'Personal access tokens' section. The 'Tokens (classic)' link is also highlighted with a red box. The 'Generate new token (classic)' modal is open, showing the 'Generate new token (classic)' option for general use.

carloseduardoxp  
Carlos Eduardo

Set status

Your profile

Your repositories

Your Copilot

Your projects

Your stars

Your gists

Your organizations

Your enterprises

Your sponsors

Try Enterprise (Free)

Feature preview

Settings

Security

Code security

Integrations

Applications

Scheduled reminders

Archives

Security log

Sponsorship log

Developer settings

Generate new token (Beta)

Fine-grained, repo-scoped

Generate new token (classic)

For general use

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens (Preview)

Tokens (classic)



# Criando um access token

## New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

### Note

What's this token for?

### Expiration \*

30 days ▾

The token will expire on Thu, Jan 2 2025

Generate token

Cancel

## Personal access tokens (classic)

Generate new token ▾

Tokens you have generated that can be used to access the [GitHub API](#).



Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp\_



Delete

pesquisa — copilot, project, repo, user, workflow

Expires on Thu, Jan 23 2025.

Last used within the last week

Delete

# Tabulação dos dados – Script Python

```
import csv, requests

access_token = 'seu_access_token'

def run_query(query):
    url = 'https://api.github.com/graphql'
    headers = {'Authorization': f'Bearer {access_token}'}

    response = requests.post(url, json={'query': query}, headers=headers)

    if response.status_code != 200:
        raise RuntimeError(
            f"Falha na execução da query GraphQL.\n"
            f"Status HTTP: {response.status_code}\n"
            f"Resposta: {response.text}\n"
            f"Query: {query}"
        )

    data = response.json()
    if 'errors' in data:
        raise RuntimeError(
            f"Erros retornados pela API GraphQL:\n"
            f"{data['errors']}\n"
            f"Query: {query}"
        )

    return data
```

Variável que irá armazenar o  
access\_token extraído do Github

# Tabulação dos dados – Script Python

```
def query_composer(cursor=None):
    cursor_part = f', after: "{cursor}"' if cursor else ""
    query = f"""
    query {{
      search(query: "chat.openai.com/share is:pr is:merged in:title,body",
        type: ISSUE,
        first: 100{cursor_part}) {{
        pageInfo {{
          endCursor
          hasNextPage
        }}
        issueCount
        edges {{
          node {{
            ... on PullRequest {{
              url
              title
              createdAt
              mergedAt
              repository {{
                stargazerCount
                isFork
                primaryLanguage {{
                  name
                }}
              }}
            }}
          }}
        }}
      }}
    }}
    """
    return query
```

Método que irá devolver a query  
GraphQL.

Irá adicionar a string “after:...”  
quando houver cursor

# Tabulação dos dados – Script Python

```
def get_samples():  
    cursor = None  
    has_next_page = True  
    prs = []  
  
    while has_next_page:  
        result = run_query(query_composer(cursor))  
        end_cursor = result["data"]["search"]["pageInfo"]["endCursor"]  
        has_next_page = result["data"]["search"]["pageInfo"]["hasNextPage"]  
  
        issue_count = result["data"]["search"]["issueCount"]  
  
        print(f"Occurrences: {issue_count}")  
  
        for pr in result["data"]["search"]["edges"]:  
            pr_url = pr["node"]["url"]  
            pr_title = pr["node"]["title"]  
            pr_created_at = pr["node"]["createdAt"]  
            pr_merged_at = pr["node"]["mergedAt"]  
            stars = pr["node"]["repository"]["stargazerCount"]  
            fork = pr["node"]["repository"]["isFork"]  
            language = ""  
            if pr["node"]["repository"]["primaryLanguage"] != None:  
                language = pr["node"]["repository"]["primaryLanguage"]["name"]  
  
            prs.append((pr_url, pr_title, pr_created_at, pr_merged_at, stars, fork, language))  
  
        cursor = end_cursor  
    return prs
```

Método que irá buscar as amostras e jogá-las em uma lista

Continuará executando enquanto existir página

Armazenará todos os campos retornados do GraphQL em uma lista

# Tabulação dos dados – Script Python

Método para escrever o csv

```
def write_samples(prs):  
    filename = 'Candidate samples.csv'  
  
    with open(filename, mode='w', newline="", encoding='utf-8') as file:  
        writer = csv.writer(file)  
        writer.writerow(['PR URL', 'PR Title', 'PR createdAt', 'PR mergedAt', 'stars', 'fork', 'language'])  
        writer.writerows(prs)
```

```
prs = get_samples()  
write_samples(prs)
```

Chamada dos métodos de buscar  
as amostras e escrevê-las em um  
csv

# Executando o script

	A	B	C	D	E	F	G	H
1	PR URL	PR Title	PR createdAt	PR mergedAt	stars	fork	language	
2	<a href="https://github.com/Mudlet/Mudlet/pull/7123">https://github.com/Mudlet/Mudlet/pull/7123</a>	Fix a crash wher	2024-02-07T19:2	2024-04-25T15:2	740	FALSE	C++	
3	<a href="https://github.com/Mudlet/Mudlet/pull/7120">https://github.com/Mudlet/Mudlet/pull/7120</a>	Fix: prevent an c	2024-02-03T19:4	2024-02-17T17:0	740	FALSE	C++	
4	<a href="https://github.com/HBO-i/ictresearchmethods.nl/pull/230">https://github.com/HBO-i/ictresearchmethods.nl/pull/230</a>	Update static-prc	2024-04-15T14:5	2024-07-30T19:0	9	FALSE	Svelte	
5	<a href="https://github.com/HBO-i/ictresearchmethods.nl/pull/96">https://github.com/HBO-i/ictresearchmethods.nl/pull/96</a>	[Maria Pizarro C	2024-04-22T07:5	2024-04-22T07:4	1	FALSE	JavaScript	
6	<a href="https://github.com/HBO-i/ictresearchmethods.nl/pull/96">https://github.com/HBO-i/ictresearchmethods.nl/pull/96</a>	Adding "Queryin	2024-05-16T21:5	2024-05-22T21:4	119	FALSE	TeX	
7	<a href="https://github.com/alshedivat/al-folio/pull/2059">https://github.com/alshedivat/al-folio/pull/2059</a>	Enable specifyin	2024-01-10T05:2	2024-05-28T00:7	11416	FALSE	HTML	
8	<a href="https://github.com/possee-org/genai-numpy/pull/13">https://github.com/possee-org/genai-numpy/pull/13</a>	Missing Docstrin	2024-05-08T23:5	2024-05-08T23:5	4	FALSE	Jupyter Notebook	
9	<a href="https://github.com/YongHyeonLeeKr/WebBasic/pull/80">https://github.com/YongHyeonLeeKr/WebBasic/pull/80</a>	BK-board-crud: [	2024-05-11T07:5	2024-05-12T01:0	0	FALSE	JavaScript	
10	<a href="https://github.com/pollen-robotics/rustypot/pull/50">https://github.com/pollen-robotics/rustypot/pull/50</a>	Adding xm devic	2024-04-30T09:4	2024-05-03T11:2	17	FALSE	Rust	
11	<a href="https://github.com/open-truss/open-truss/pull/171">https://github.com/open-truss/open-truss/pull/171</a>	Add graphql api	2024-05-09T17:0	2024-05-10T18:5	8	FALSE	TypeScript	
12	<a href="https://github.com/LIDR-academy/AI4Devs-intro-202404/pull/72">https://github.com/LIDR-academy/AI4Devs-intro-202404/pull/72</a>	Snake - Team 6	2024-04-16T17:5	2024-04-17T08:0	1	FALSE	JavaScript	
13	<a href="https://github.com/UNLV-CS472-672/2024-S-GROUP3-Barbell/pull/225">https://github.com/UNLV-CS472-672/2024-S-GROUP3-Barbell/pull/225</a>	153 frontend priv	2024-04-20T05:5	2024-04-25T00:0	4	FALSE	TypeScript	
14	<a href="https://github.com/UNLV-CS472-672/2024-S-GROUP3-Barbell/pull/225">https://github.com/UNLV-CS472-672/2024-S-GROUP3-Barbell/pull/225</a>	173 / Update cre	2024-04-19T23:5	2024-04-23T05:5	4	FALSE	TypeScript	

# Importando o csv no Google Planilha

	A	B	C	D	E	F	G	H
1	PR URL	PR Title	PR createdAt	PR mergedAt	stars	fork	language	
2	<a href="https://github.com/Mudlet/Mudlet/pull/7123">https://github.com/Mudlet/Mudlet/pull/7123</a>	Fix a crash when	2024-02-07T19:00:00Z	2024-04-25T15:00:00Z	740	FALSE	C++	
3	<a href="https://github.com/Mudlet/Mudlet/pull/7120">https://github.com/Mudlet/Mudlet/pull/7120</a>	Fix: prevent an c	2024-02-03T19:00:00Z	2024-02-17T17:00:00Z	740	FALSE	C++	
4	<a href="https://github.com/HBO-ictresearchmethods.nl/pull/230">https://github.com/HBO-ictresearchmethods.nl/pull/230</a>	Update static-pro	2024-04-15T14:00:00Z	2024-07-30T19:00:00Z	9	FALSE	Svelte	
5	<a href="https://github.com/HBO-ictresearchmethods.nl/pull/96">https://github.com/HBO-ictresearchmethods.nl/pull/96</a>	[Maria Pizarro C	2024-04-22T07:00:00Z	2024-04-22T07:00:00Z	1	FALSE	JavaScript	
6	<a href="https://github.com/HBO-ictresearchmethods.nl/pull/96">https://github.com/HBO-ictresearchmethods.nl/pull/96</a>	Adding "Queryin	2024-05-16T21:00:00Z	2024-05-22T21:00:00Z	119	FALSE	TeX	
7	<a href="https://github.com/alshedivat/al-folio/pull/2059">https://github.com/alshedivat/al-folio/pull/2059</a>	Enable specifying	2024-01-10T05:00:00Z	2024-05-28T00:00:00Z	11416	FALSE	HTML	
8	<a href="https://github.com/possee-org/genai-numpy/pull/13">https://github.com/possee-org/genai-numpy/pull/13</a>	Missing Docstrin	2024-05-08T23:00:00Z	2024-05-08T23:00:00Z	4	FALSE	Jupyter Notebook	
9	<a href="https://github.com/YongHyeonLeeKr/WebBasic/pull/80">https://github.com/YongHyeonLeeKr/WebBasic/pull/80</a>	BK-board-crud: [	2024-05-11T07:00:00Z	2024-05-12T01:00:00Z	0	FALSE	JavaScript	
10	<a href="https://github.com/pollen-robotics/rustypot/pull/50">https://github.com/pollen-robotics/rustypot/pull/50</a>	Adding xm devic	2024-04-30T09:00:00Z	2024-05-03T11:00:00Z	17	FALSE	Rust	
11	<a href="https://github.com/open-truss/open-truss/pull/171">https://github.com/open-truss/open-truss/pull/171</a>	Add graphql api	2024-05-09T17:00:00Z	2024-05-10T18:00:00Z	8	FALSE	TypeScript	
12	<a href="https://github.com/LIDR-academy/AI4Devs-intro-202404/pull/72">https://github.com/LIDR-academy/AI4Devs-intro-202404/pull/72</a>	Snake - Team 6	2024-04-16T17:00:00Z	2024-04-17T08:00:00Z	1	FALSE	JavaScript	
13	<a href="https://github.com/UNLV-CS472-672/2024-S-GROUP3-Barbell/pull/225">https://github.com/UNLV-CS472-672/2024-S-GROUP3-Barbell/pull/225</a>	153 frontend priv	2024-04-20T05:00:00Z	2024-04-25T00:00:00Z	4	FALSE	TypeScript	
14	<a href="https://github.com/UNLV-CS472-672/2024-S-GROUP3-Barbell/pull/223">https://github.com/UNLV-CS472-672/2024-S-GROUP3-Barbell/pull/223</a>	173 / Update cre	2024-04-19T23:00:00Z	2024-04-23T05:00:00Z	4	FALSE	TypeScript	

# Próximos passos?

---

## 1) Higienizar os dados

Next, we performed manual filtering to discard as many false positives as possible:

1. 97 merged PRs where the ChatGPT share link was not found in the PR title, body, comments, commit messages or code diffs.
2. 29 merged PRs with broken or invalid ChatGPT share links (e.g., error 404).
3. 25 merged PRs written in non-english languages.
4. 12 merged PRs without any reviewer (e.g., the developer themselves opened the PR and merged it without receiving any feedback on their modification)



# Próximos passos?

## 2) Analisar os dados e responder as perguntas de pesquisa

**Table 1. ChatGPT requests performed in Merged Pull Requests**

Category	ChatGPT Request	Occurrences	Total
Code Generation	How-To Code Snippets	30	61
	Task Automation Requests	20	
	Feature Addition to Existing Code	11	
Code Review	Fix Bugs and Warnings	18	46
	Optimization/Refactoring	13	
	Explain the Code	8	
	Test/Debug	5	
	Performance Analysis	2	
Information Request	Technical Explanation	24	42
	Technical Support	10	
	Coding Conventions	5	
	Policy	3	
Text Review	Grammar and Refinement	10	14
	Formatting	4	
<b>Total</b>			<b>163</b>

SILVA, Julyanara R.; DANTAS, Carlos Eduardo C.; MAIA, Marcelo A.. **What Developers Ask to ChatGPT in GitHub Pull Requests? an Exploratory Study.** In: VEM , 2024, Curitiba/PR.

Documentação oficial da API GraphQL: <https://docs.github.com/en/graphql>