PUC Minas

INSTITUTO DE CIÊNCIAS EXATAS E INFORMÁTICA
UNIDADE EDUCACIONAL PRAÇA DA LIBERDADE
Bacharelado em Engenharia de Software
Laboratório de Medição de Software

Aluno: Lucas Branco Laborne Tavares
Professor: Humberto Torres

Caracterização de Repositórios populares do GitHub utilizando a API GraphQL

Introdução

Este trabalho tem como objetivo analisar dados sobre repositórios populares do GitHub, através de um experimento utilizando a tecnologia GraphQL e a linguagem Python, para investigar questões de pesquisa, objetivando ampliar a percepção sobre o comportamento de sistemas open-source em larga escala, a utilização de tecnologias modernas, e o uso da experimentação como instrumento para tomada de decisões mais eficazes. O repositório deste trabalho está disponível em: https://github.com/lbltavares/trabalho-lab-github

GitHub

O GitHub é uma plataforma de armazenamento e versionamento de código-fonte baseada em git, amplamente utilizada atualmente. Repositórios hospedados no GitHub, se autorizados devidamente, podem ser visíveis para qualquer pessoa, o que promove a contribuição e a divulgação de trabalhos entre desenvolvedores do mundo inteiro. Além disso, o GitHub disponibiliza gratuitamente dados sobre todos os repositórios públicos a partir de uma API de consulta. Por estes motivos, o GitHub foi selecionado como plataforma deste experimento.

API GraphQL

A GraphQL é uma linguagem de consulta para APIs criada pelo Facebook, que fornece uma descrição mais compreensível e utilização mais simples, uma vez que ela pode especificar quais dados o cliente necessita buscar, reduzindo o custo da transferência de dados e garantindo tempos de resposta mais rápidos. O GitHub fornece um serviço de API GraphQL, podendo este ser utilizado via programação ou ainda a partir de uma interface gráfica, via navegador. Por estes motivos, o GraphQL foi utilizado como linguagem de consulta para este experimento.

Questões a serem investigadas

O enfoque deste experimento é responder às seguintes perguntas de pesquisa (RQ, ou *Research Questions*), bem como explorar e discutir os seus resultados:

1. Sistemas populares são maduros/antigos?
2. Sistemas populares recebem muita contribuição externa?
3. Sistemas populares lançam releases com frequência?
4. Sistemas populares são atualizados com frequência?
5. Sistemas populares são escritos nas linguagens mais populares?
6. Sistemas populares possuem um alto percentual de issues fechadas?

Hipóteses

Antes de executar o experimento e obter os resultados, foram elaboradas as seguintes hipóteses informais para cada RQ:

1. *Sistemas populares são maduros/antigos?*
   Considerando que repositórios antigos/maduros possuem **mais de cinco anos de idade**, minha hipótese é a de que **os sistemas populares** (definidos a partir do número de estrelas) **são maduros**, uma vez que leva um tempo até a divulgação e popularização do mesmo.

2. *Sistemas populares recebem muita contribuição externa?*
   Minha hipótese é a de se verifique **mais de 5000 pull-requests aceitas em sistemas populares**, o que os caracteriza como repositórios que recebem muita contribuição externa.

3. *Sistemas populares lançam releases com frequência?*
   Considerando **apenas os repositórios que lançam releases pelo Github**, minha hipótese para esta pergunta é a de que **os sistemas populares lançaram mais de 100 releases, sendo a última release lançada em menos de 1 mês (30 dias)**, devido ao grande número de contribuidores e à grande demanda que estes sistemas possuem.

4. *Sistemas populares são atualizados com frequência?*
   Minha hipótese para esta pergunta é a de que **os sistemas populares receberam a última atualização em menos de 7 dias**, também devido ao grande número de usuários/contribuidores e à grande demanda que estes sistemas possuem.

5. *Sistemas populares são escritos nas linguagens mais populares?*
   Minha hipótese é a de que as linguagens têm grande influência sobre a popularidade de um repositório, portanto, seguindo como critério a pesquisa realizada pelo StackOverflow, **espera-se observar a maioria das linguagens mais populares nos sistemas analisados**.

6. *Sistemas populares possuem um alto percentual de issues fechadas?*
   Desconsiderando os repositórios que não possuem issues ou que utilizam outro sistema de relatório de erros, minha hipótese é que **os repositórios possuam pelo menos 70% de issues fechadas**, levando em conta o grande número de contribuidores destes repositórios.

7. *Sistemas escritos em linguagens mais populares recebem mais contribuição externa, lançam mais releases e são atualizados com mais frequência?*
   Considerando que a linguagem na qual um repositório é escrito influencia na sua popularidade, minha hipótese é a de que **repositórios que são escritos em linguagens populares tendem a receber mais contribuição externa, lançarem mais releases e serem atualizados com mais frequência do que os demais**.

Metodologia e Tecnologias utilizadas

Este experimento foi realizado a partir de dados coletados de **1000 repositórios do GitHub**, utilizando a tecnologia **GraphQL** para realizar as queries, e a linguagem **Python** para executar o algoritmo de coleta e análise dos dados. Foi utilizada a biblioteca *requests* para realizar as solicitações HTTP, a biblioteca *matplotlib* para plotar os gráficos presentes neste documento, e a biblioteca *numpy* como ferramenta para facilitar cálculos matemáticos. As métricas utilizadas para responder cada questão foram, respectivamente:

RQ1. Idade do repositório (calculado a partir da data de sua criação)
RQ2. Total de pull requests aceitas
RQ3. Total de releases e última release criada. Repositórios que não possuem releases serão desconsiderados.
RQ4. Tempo até a última atualização (calculado a partir da data de última atualização)
RQ5. Linguagem primária de cada um desses repositórios, e comparação com a pesquisa realizada pelo StackOverflow: https://insights.stackoverflow.com/survey/2019/#technology
Repositórios que não utilizam nenhuma linguagem de programação serão excluídos da análise.
RQ6. Razão entre número de issues fechadas pelo total de issues. Repositórios que não possuem Issues serão desconsiderados.
RQ7. Valores medianos de porcentagens de pull-requests aceitas, número de releases e dias desde a última atualização.

Resultados

Os resultados para este experimento foram obtidos no dia **16/09/2020**, a partir de dados coletados de um arquivo *.csv* contendo todas as informações sobre os **1000 repositórios mais populares do GitHub**.

1. *Sistemas populares são maduros/antigos?*
   Os resultados obtidos para responder esta pergunta mostraram que a maior parte dos repositórios coletados tem entre **6 a 8 anos de idade**, e possuem:

   Média: 5.9 anos
   Mediana: 5.8 anos
   Desvio-Padrão: 2.5 anos
   Repositórios analisados: 1000

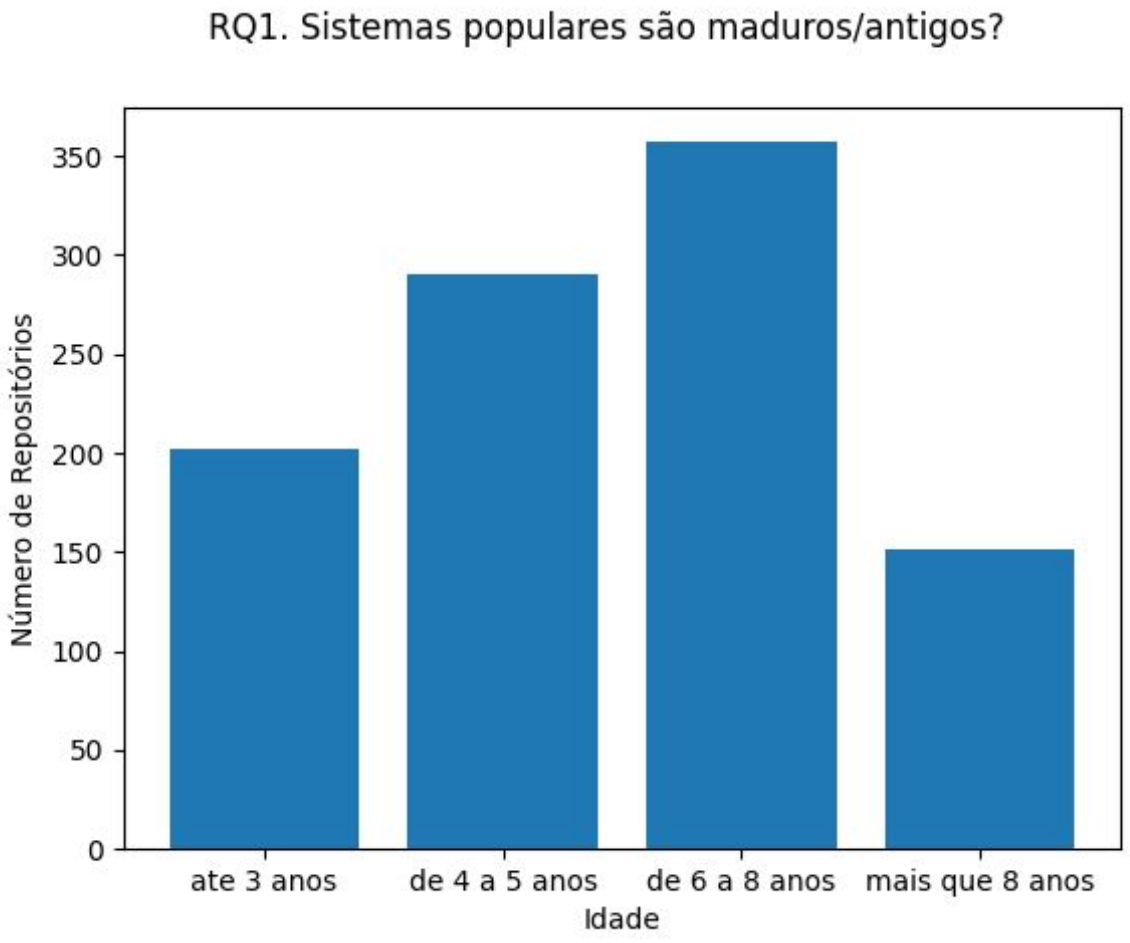

Figura 1.1: A maior parte dos sistemas mais populares do GitHub possuem entre 6 a 8 anos de idade

O histograma abaixo ilustra a distribuição de frequências de idades dos 1000 repositórios mais populares do GitHub:

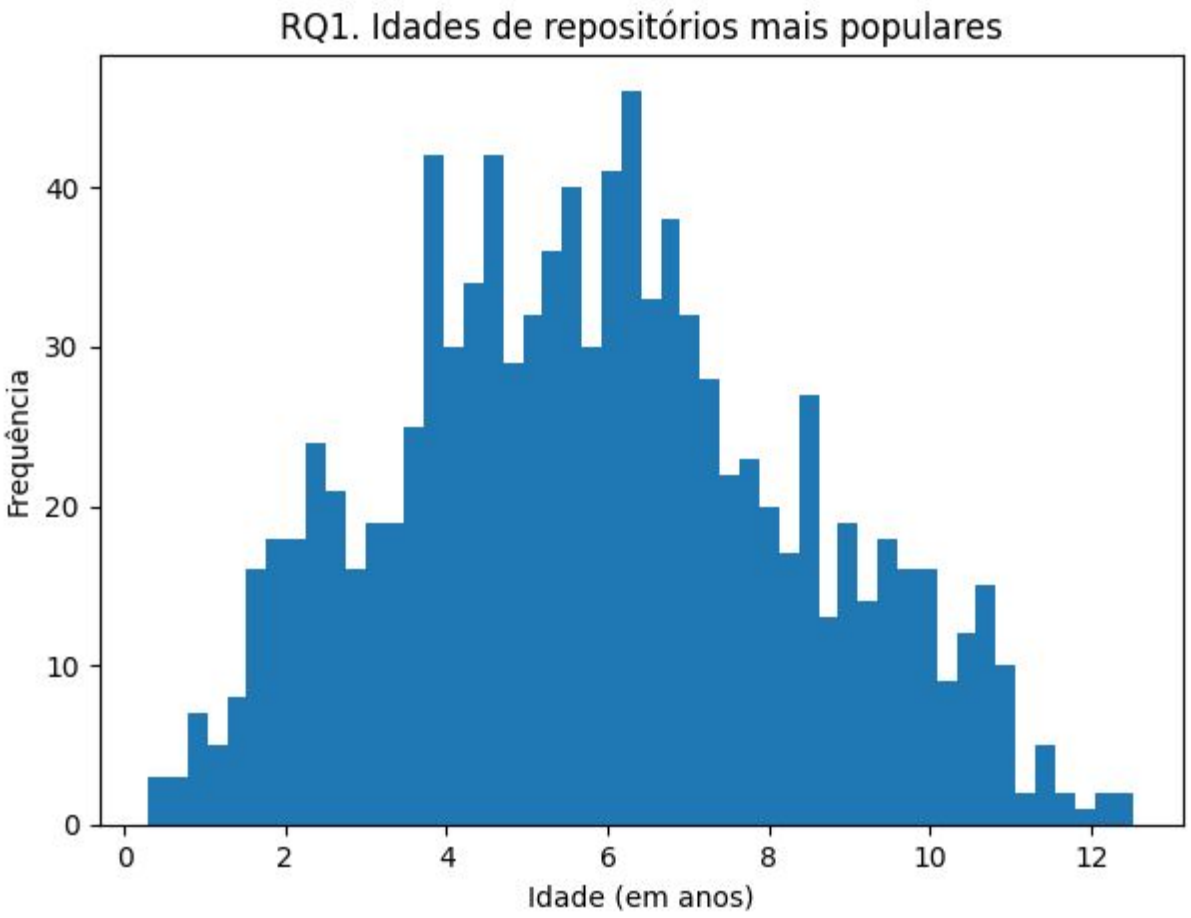

Figura 1.2: Distribuição de frequência de idades (em anos) de repositórios mais populares do GitHub

2. *Sistemas populares recebem muita contribuição externa?*
   A análise dos dados coletados para responder esta pergunta mostraram os seguintes resultados:

   Repositórios analisados: 1000
   Média: 1519.4 pull-requests aceitas
   Mediana: 314.5 pull-requests aceitas
   Desvio-Padrão: 4465.8 pull-requests aceitas

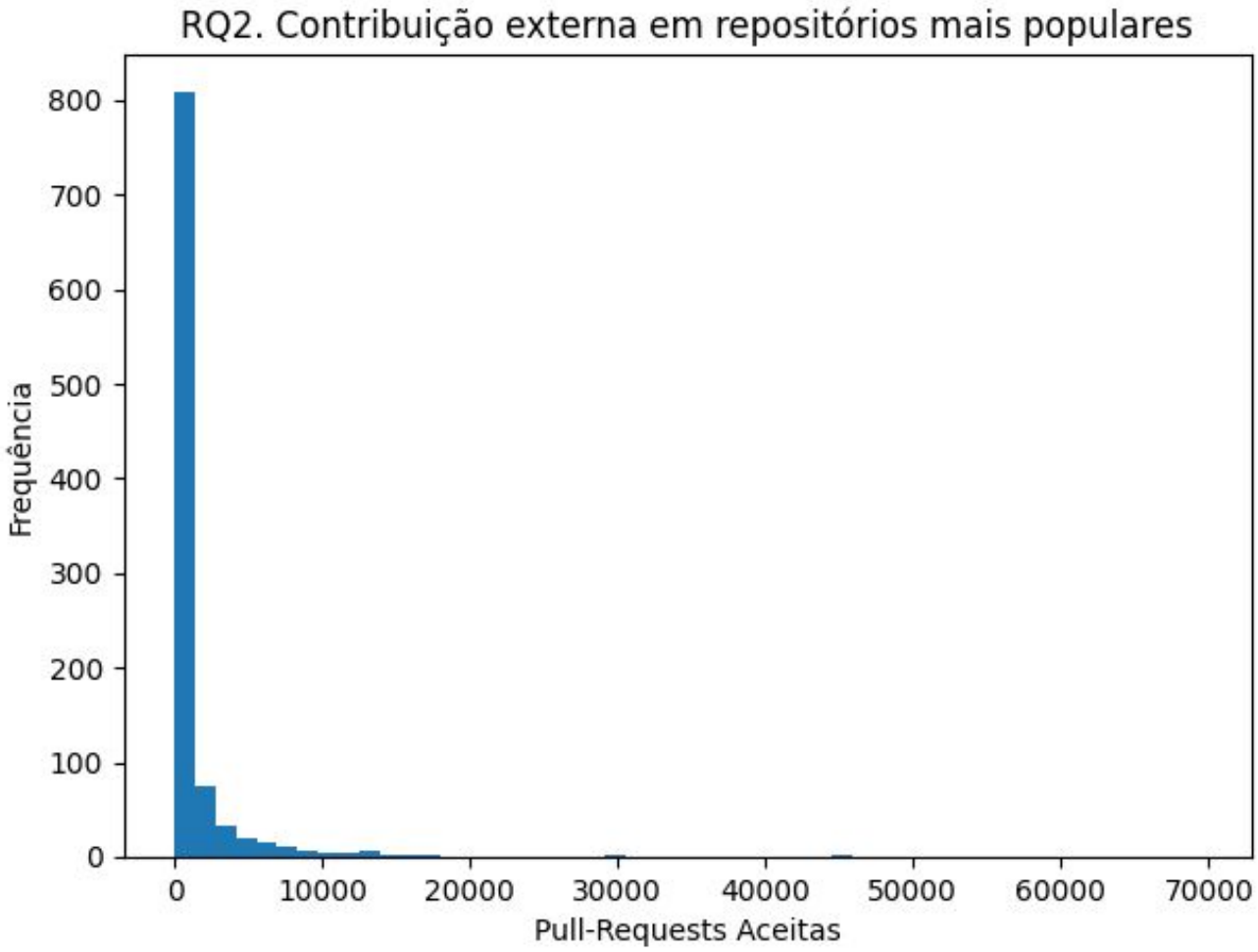

Figura 2.1: Distribuição de frequência de pull-requests aceitas

3. *Sistemas populares lançam releases com frequência?*
   Para responder esta RQ, primeiramente foram considerados os repositórios que lançavam releases pelo GitHub (desconsiderando os que não possuiam releases). Os resultados obtidos foram os seguintes:

   Repositórios analisados: 622
   Média: 68.9 releases
   Mediana: 32.0 releases
   Desvio-Padrão: 97.0 releases

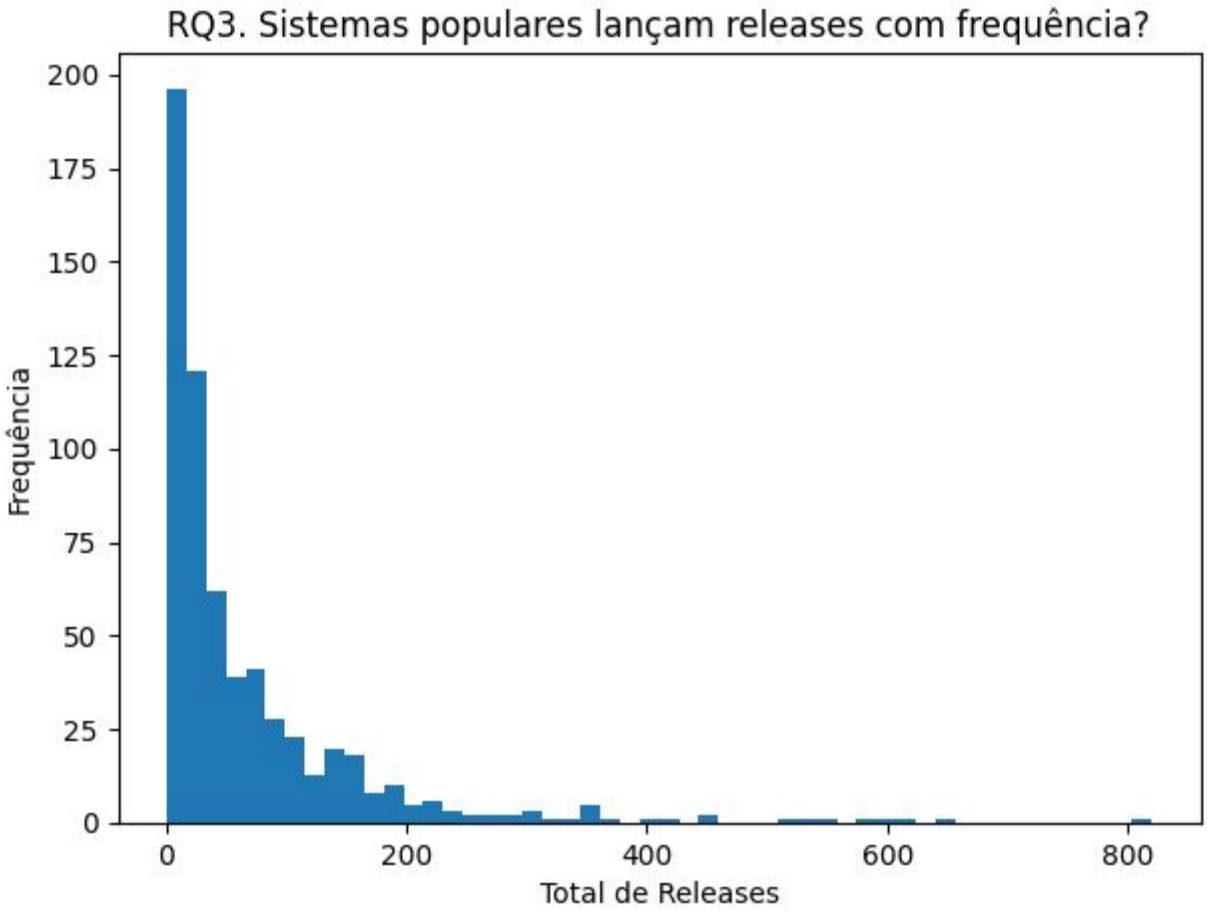

Figura 3.1: Distribuição de frequência do número de releases por repositório

Em seguida, um novo filtro foi criado para obter os repositórios que possuiam mais de 1 release, e foi calculado o **tempo médio entre cada release (em dias)**, por repositório. Os resultados obtidos foram os seguintes:

Repositórios analisados: 598
Média: 78.6 dias
Mediana: 41.5 dias
Desvio-Padrão: 116.9 dias

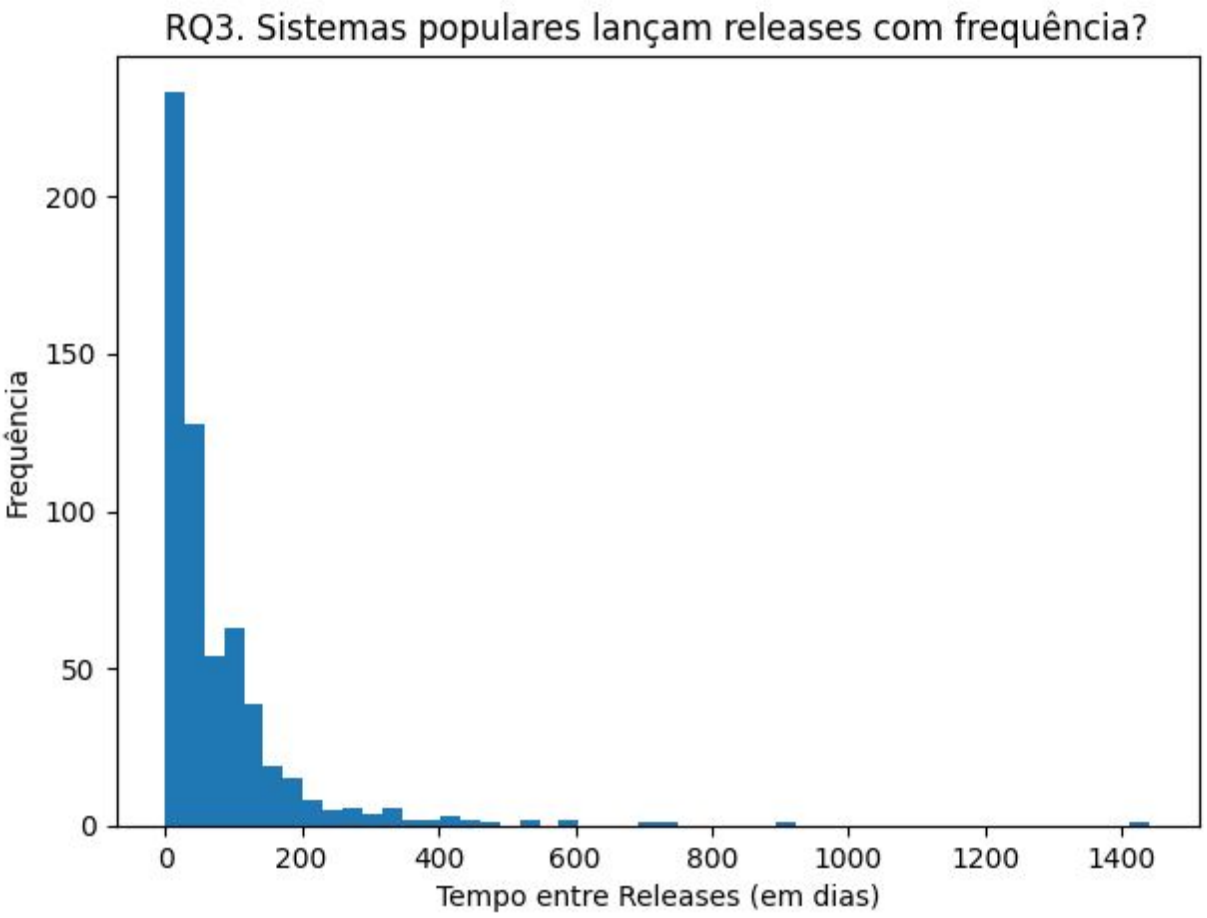

Figura 3.2: Distribuição de frequência do tempo entre releases (em dias) de cada repositório

4. *Sistemas populares são atualizados com frequência?*
   Foram coletados dados sobre os tempos decorridos desde a última atualização de cada repositório, considerando a **data de hoje (16/09/2020)**. Os resultados obtidos foram os seguintes:

   Repositórios analisados: 1000
   Média: 2.4 horas
   Mediana: 0.0 horas
   Desvio-Padrão: 9.6 horas

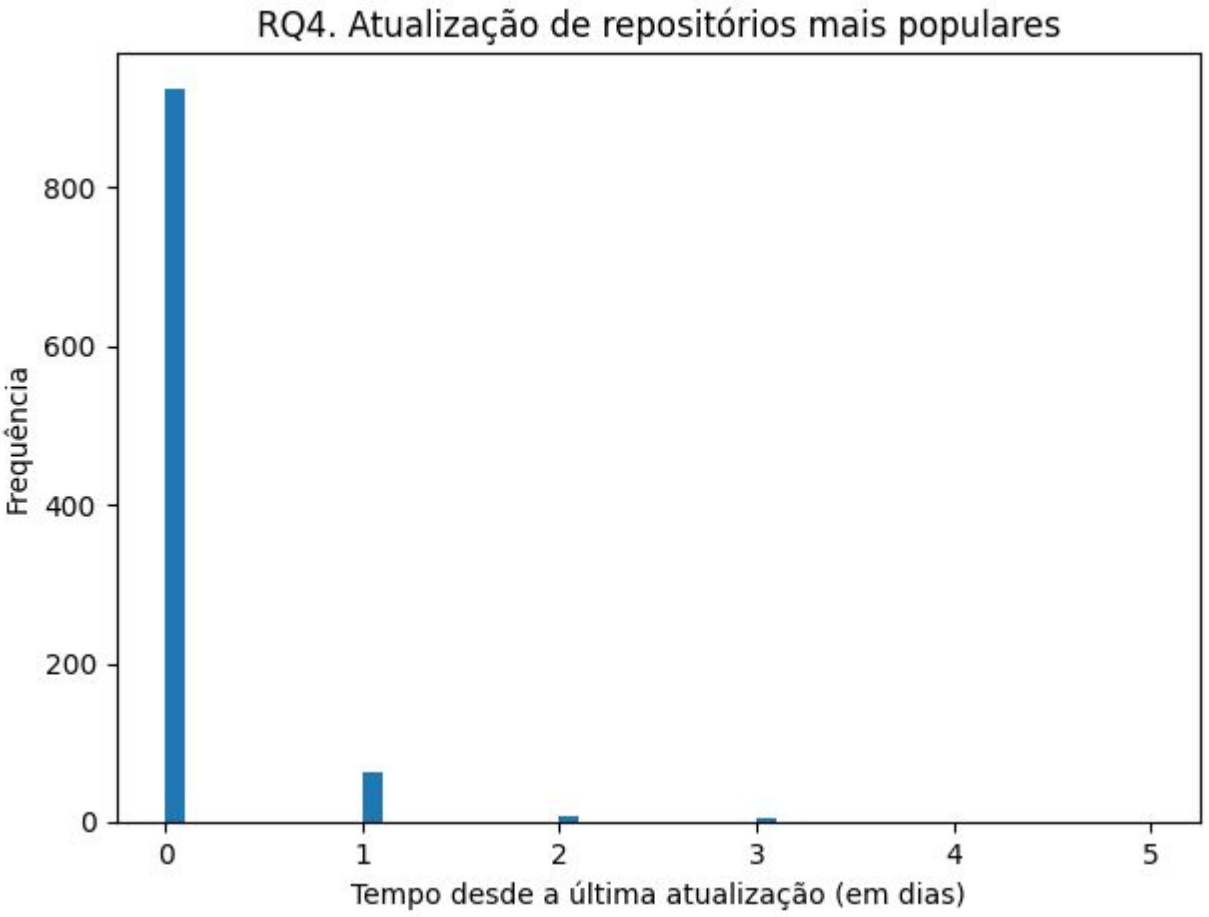

Figura 4.1: Distribuição de frequência dos tempos desde a última atualização (em dias) de repositórios mais populares

5. *Sistemas populares são escritos nas linguagens mais populares?*
   Para responder esta pergunta, foi tomado como parâmetro para a popularidade de uma linguagem a sua posição no ranking de tecnologias mais populares, criado a partir de uma pesquisa realizada pelo StackOverflow em 2019. A figura abaixo ilustra as frequências observadas de 25 linguagens mais presentes nos repositórios mais populares do GitHub:

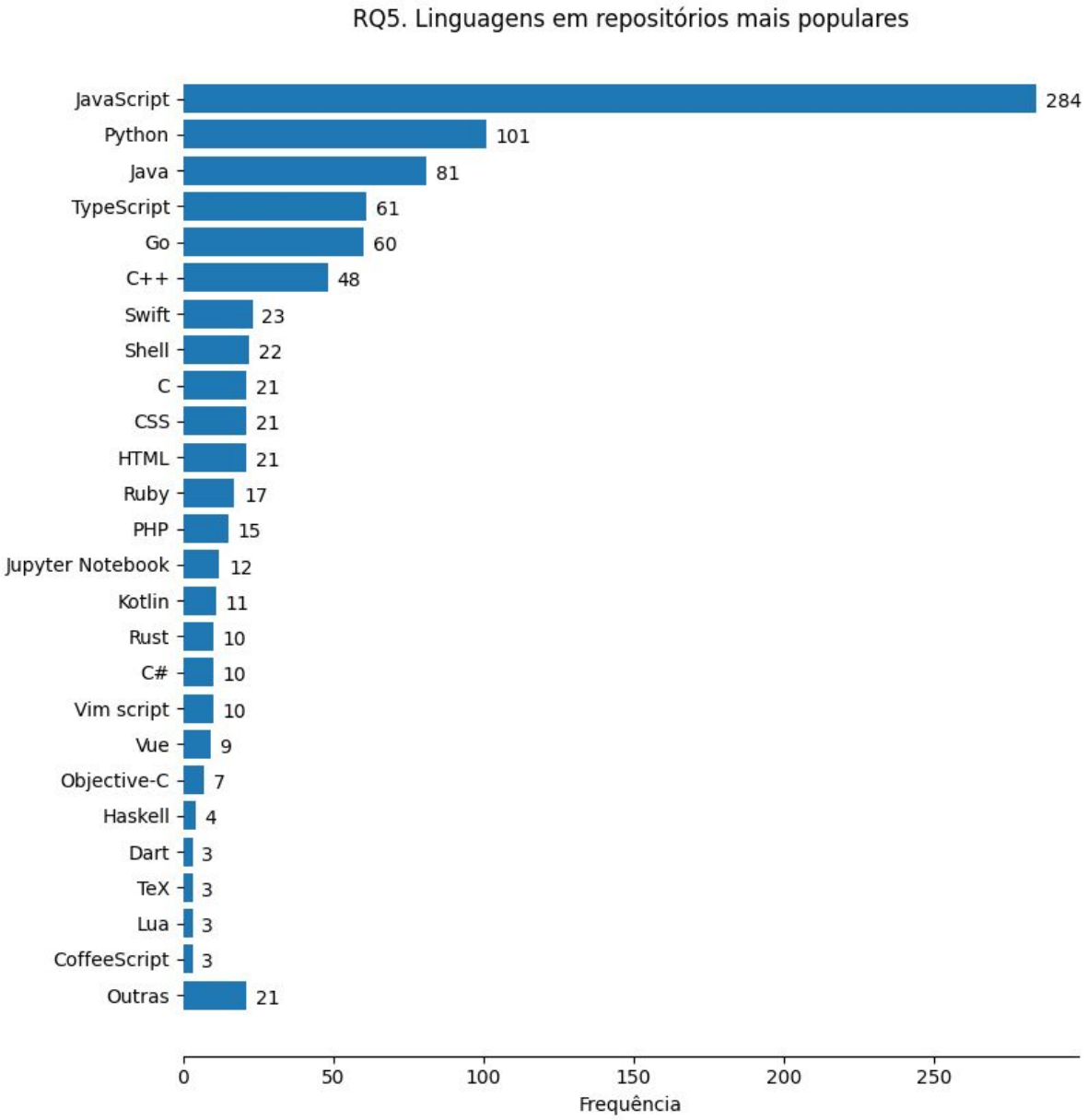

Figura 5.1: Linguagens mais utilizadas em repositórios mais populares do GitHub

Foi verificado que **81.5% das linguagens mais populares estão presentes nos repositórios mais populares**. Também foi verificado que **os repositórios que utilizam estas linguagens representam 93.53% do total dos repositórios**.

A Tabela abaixo mostra as frequências de todas as linguagens observadas. É importante destacar que **119 dos 1000 repositórios não possuiam linguagens de programação, portanto eles foram excluídos desta análise**:

Linguagem	Frequência
JavaScript	*284*
Python	*101*
Java	*81*
TypeScript	*61*
Go	*60*
C++	*48*
Swift	*23*
Shell	*22*
C	*21*
CSS	*21*
HTML	*21*
Ruby	*17*
PHP	*15*
Jupyter Notebook	*12*
Kotlin	*11*
Rust	*10*
C#	*10*
Vim script	*10*
Vue	*9*
Objective-C	*7*

Linguagem	Frequência
Haskell	*4*
Dart	*3*
TeX	*3*
Lua	*3*
CoffeeScript	*3*
Clojure	*2*
Assembly	*2*
Scala	*2*
Makefile	*2*
Elixir	*2*
Dockerfile	*1*
Julia	*1*
Batchfile	*1*
OCaml	*1*
Emacs Lisp	*1*
Objective-C++	*1*
V	*1*
Rich Text Format	*1*
Standard ML	*1*
Rascal	*1*
Crystal	*1*
Total	***881***

6. *Sistemas populares possuem um alto percentual de issues fechadas?*
   Para responder esta pergunta, foram **desconsiderados os repositórios que não possuem Issues**, e foram analisados os **percentuais de Issues fechadas em cada repositório**. Os resultados obtidos foram os seguintes:

   Repositórios analisados: 952
   Média: 80.7% issues fechadas
   Mediana: 86.4% issues fechadas
   Desvio-Padrão: 18.5% issues fechadas

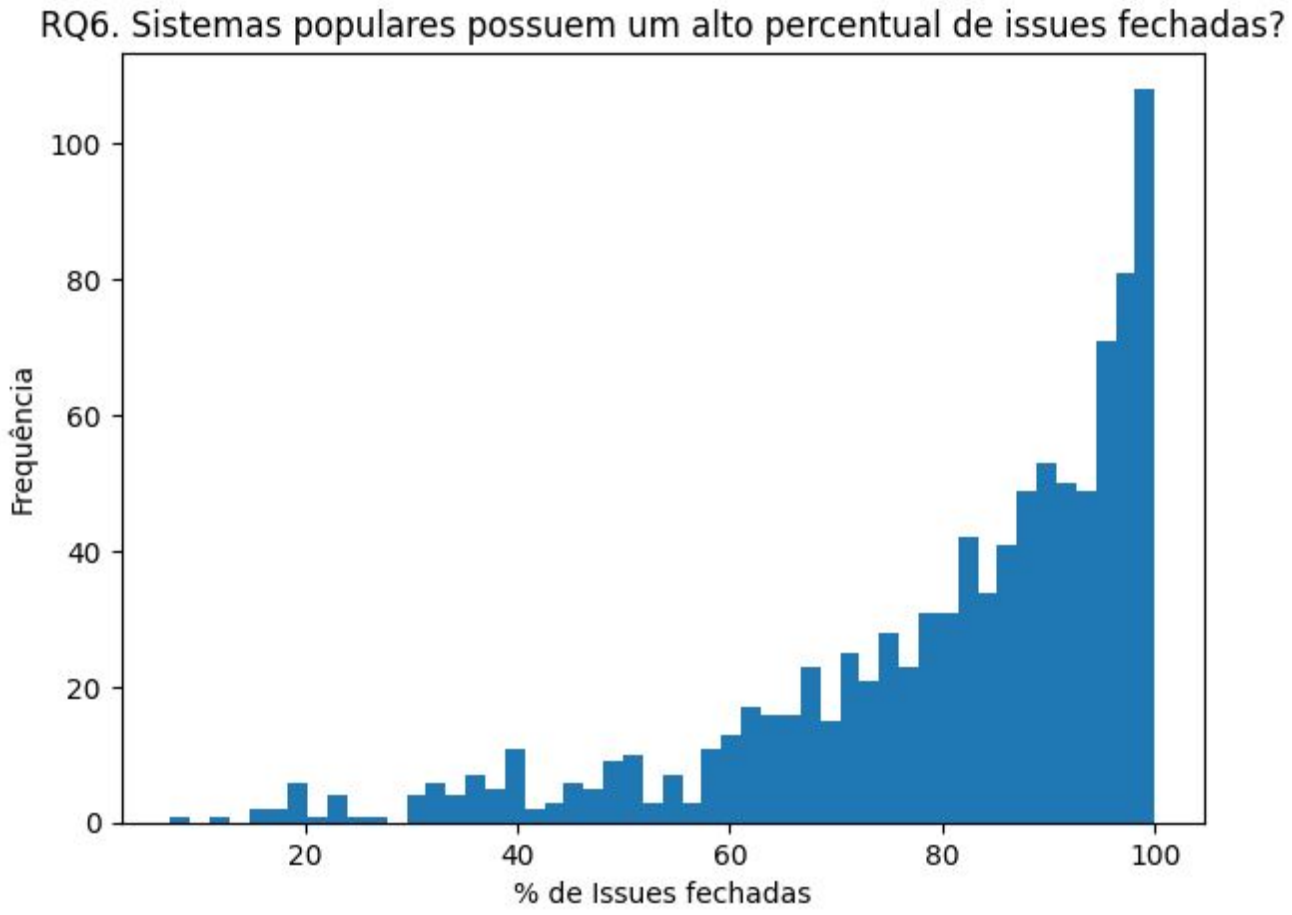

Figura 6.1: Distribuição de frequência de percentuais de Issues fechadas em repositórios mais populares do GitHub

7. *Sistemas escritos em linguagens mais populares recebem mais contribuição externa, lançam mais releases e são atualizados com mais frequência?*
   Para responder esta pergunta, foram considerados repositórios que foram escritos nas linguagens mais populares (segundo a pesquisa do StackOverflow) e repositórios que não foram escritos nestas linguagens. Em seguida, foi feita a comparação dos valores das medianas calculadas de cada um. Os resultados foram os seguintes:

Valores Medianos	**Escritos em Linguagens populares**	**Escritos em outras Linguagens**	**Diferença**
Nº de PRs Aceitas	1367	138	1229
Nº de Releases	80	27	53
Horas desde o último update	0.85	3.32	2.47

Figura 7.1: Comparação entre valores medianos de repositórios que foram escritos em linguagens mais populares, e repositórios que foram escritos em outras linguagens

---

Conclusão

1. *Sistemas populares são maduros/antigos?*
   Os resultados coletados **confirmaram a hipótese** de que os repositórios mais populares possuem, em média, mais de 5 anos, portanto podem ser considerados repositórios maduros/antigos.

2. *Sistemas populares recebem muita contribuição externa?*
   Os resultados coletados **negaram a hipótese** de que os repositórios mais populares teriam mais de 5.000 pull-requests aceitas. Os valores da média e mediana obtidos mostraram ainda que os repositórios mais populares tendem a ser mais rigorosos ao aceitar contribuição externa, possuindo menos de 2000 pull-requests aceitas.

3. *Sistemas populares lançam releases com frequência?*
   Os resultados coletados **negaram a hipótese** de que os sistemas populares possuíam mais de 100 releases, e que a última release teria sido feita em menos de 1 mês. Os valores medianos obtidos foram de 32 releases, e 41.5 dias, respectivamente.

4. *Sistemas populares são atualizados com frequência?*
   Os resultados coletados **confirmaram a hipótese** de que os sistemas populares são atualizados em menos de 7 dias, sendo constatado um valor médio de 2.4 horas desde a última atualização.

5. *Sistemas populares são escritos nas linguagens mais populares?*
   Os resultados coletados **confirmaram a hipótese** de que a maioria dos sistemas mais populares são escritos nas linguagens mais populares. Foi verificado que 81.5% das linguagens mais populares estão presentes nos repositórios mais populares, e que 93.53% dos repositórios analisados utilizam as linguagens mais populares. **A linguagem Javascript é destacada como a mais popular, sendo observada em cerca de 32% dos repositórios analisados**.

6. *Sistemas populares possuem um alto percentual de issues fechadas?*
   Os resultados coletados **confirmaram a hipótese** de que os sistemas mais populares possuem pelo menos 70% de Issues fechadas. Os valores da média e mediana calculados foram, respectivamente: 80.7% e 86.4%. Estes resultados refletem o engajamento dos contribuidores e desenvolvedores nos repositórios mais populares.

7. *Sistemas escritos em linguagens mais populares recebem mais contribuição externa, lançam mais releases e são atualizados com mais frequência?*
   Os resultados coletados **confirmaram a hipótese** de que os sistemas mais populares escritos em linguagens mais populares recebem mais contribuição externa, lançam mais releases e são atualizados com mais frequência.