Características de Sistemas Populares GitHub

Henrique Ramos

13 de Março de 2020

1 Introdução

Neste primeiro laboratório de experimentação de software, foi proposto aos alunos analizar as principais características de populares repositórios open-source. Para realizar a tarefa, foi necessário coletar dados sobre os **1.000 repositórios com o maior número de estrelas no GitHub** utilizando a linguagem de preferência. Este trabalho foi desenvolvido utilizando **Python**, linguagem sugerida pelo professor Laerte e vastamente utilizada na área de ciência de dados. Ao iniciar a pesquisa, houve a formulação de uma conjectura informal à respeito dos resultados que viriam a ser obtidos:

> Os repositórios atualmente mais utilizados têm mais de 3 anos, recebem bastante contribuição externa, lançam *releases* frequentes — em razão da grande quantidade de contribuintes —, são regularmente atualizados, escritos nas linguagens mais populares, mas não possuem um alto percentual de issues fechadas.

2 Metodologia

Para responder às questões de pesquisa, utilizou-se a *scripts* na linguagem Python. Cada programa foi desenvolvido em um arquivo separado para cada pergunta, assim como para realizar a consulta e salvar os dados. Após isso, foram realizados cálculos utilizando, na maioria das vezes, média e mediana.

3 Resultados

1. **Sistemas populares são maduros/antigos?**

   Sim. Repositórios populares possuem, em média, **5 anos e meio de criação**.

   Fórmula: $\frac{\text{Timestamp data atual} - \text{Timestamp data de criação media}}{\text{Segundos em um ano}}$

2. **Sistemas populares recebem muita contribuição externa?**

   Sim. Possuem, em média, 1322 Pull requests.

   Fórmula: $\frac{\text{Soma de PRs nos repositórios}}{\text{quantidade de repositórios}}$

3. **Sistemas populares lançam releases com frequência?**

   Não. sistemas populares lançam, em média, 37 releases no total. Sabendo que, em média, foram criados há 5 anos e meio, calcula-se que lançam, em média, uma release a cada dois meses.

   Fórmula: $\frac{\text{total de releases}}{\text{número de repositórios} \times 5.5 \times 12}$

4. **Sistemas populares são atualizados com frequência?**

   Sim. No momento que o código foi executado, os repositórios haviam sido atualizados, em média, há **8 horas e 30 minutos**.

   Fórmula: $\frac{\text{Timestamp data atual} - \text{Timestamp data de atualização media}}{60 \times 60}$

5. **Sistemas populares são escritos nas linguagens mais populares?**

   Sim, dos repositórios populares, 302 são escritos em *JavaScript*, 9 em *Rust*, 45 em *C++*, 22 em *Shell*, 71 em *Java*, 48 em *TypeScript*, 23 em *C*, 94 em *Python*, 12 em *Jupyter Notebook*, 59 em *Go*, 25 em *CSS*, 19 em *PHP*, 10 em *Vue*, 8 em *C#*, 17 em *Ruby*, 22 em *HTML*, 11 em *Kotlin*, 10 em *Vim script*, 23 em *Swift*, 12 em *Objective-C*

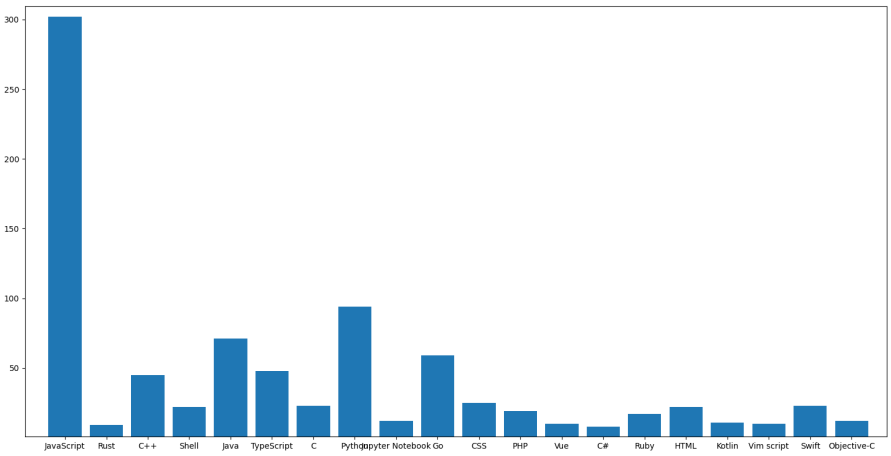

Figura 1: Gráfico de linguagens mais utilizadas nos repositórios mais populares

6. **Sistemas populares possuem um alto percentual de issues fechadas?**

   Sim. Sistemas populares possuem, em média **86% das issues fechadas**.

   Fórmula: $\frac{\text{Nº total de issues fechadas}}{\text{Nº total de issues}}$

4 Discussão sobre os resultados

Apesar da maioria dos dados aparentemente confirmarem as hipóteses previamente levantadas, os sistemas populares **possuem um alto percentual de issues fechadas** e **Não lançam *releases* com frequência**.

A primeira pode ser explicada em conjunto com a conclusão da questão 2 — sistemas populares recebem muita contribuição externa —, pois a grande quantidade de contribuintes que podem auxiliar na resolução de *issues* é alta e, por isso, muitas vezes a própria comunidade realiza este trabalho de "suporte".

Já a segunda afirmativa possivelmente ocorre em razão da popularidade destes repositórios: lançar novas versões de um sistema frequentemente, adicionando novas funcionalidades e realizando alterações naquelas já existentes, pode torná-lo de difícil uso por desenvolvedores em produção, que terão problemas de refatoração e reaprendizado.