Laboratório 1 - Características de Sistemas Populares GitHub

Davi Brandão Saldanha

Pontifícia Universidade Católica de Minas Gerais(PUC-MG)

Introdução:

Github é uma plataforma para versionamento de projetos muito usada por profissionais da área da informática, ela ajuda a organizar seus arquivos, armazená-los na nuvem e gerenciar alterações. Essa ferramenta é gratuita, embora existam features pagas também para fins comerciais de empresas.

O Github funciona a base de repositórios que podem ser públicos ou privados, esses repositórios que fazem a mágica da ferramenta acontecer, os arquivos do repositório podem ser acessados por você em qualquer PC do mundo, desde que haja conexão à internet.

Vários usuários podem ser contribuintes de um mesmo repositório, por isso a ferramenta é altamente recomendada para empresas, facilitando a integração entre os serviços de várias pessoas.

Para esse trabalho também é importante falar sobre a API do Github, que nos permite acessar dados dos repositórios afim de fazer medições e analisar os resultados obtidos.

Perguntas:

RQ 01. Sistemas populares são maduros/antigos?

RQ 02. Sistemas populares recebem muita contribuição externa?

RQ 03. Sistemas populares lançam releases com frequência?

RQ 04. Sistemas populares são atualizados com frequência?

RQ 05. Sistemas populares são escritos nas linguagens mais populares?

RQ 06. Sistemas populares possuem um alto percentual de issues fechadas?

Hipóteses:

1: Para um software ser considerado maduro decidi usar o período de 3 anos de idade, esse seria um tempo considerável da existência de um software, além de provavelmente já ter passado pelo seu processo de popularização, fazendo-o ter uma boa avaliação no Github.

2: Espera-se muita contribuição externa em projetos com tecnologia baseada em JS e o Linux, por serem tecnologias de código aberto bem populares e tem um bom engajamento da comunidade a fim de melhorar o desempenho

3: Uma junção das 2 métricas anteriores seria o adequado a se analisar aqui, tendo vista que um projeto mais maduro teria mais releases devido justamente ao seu tempo de existência. e o fato de um projeto que recebe muita ajuda externa, deveria constantemente soltar releases novas para integrar todas as “novidades”. Então sim, se espera um grande número de releases (500+).

4: Sistemas de código aberto sim, devem receber atualizações constantes devido ao engajamento da sua comunidade, já sistemas mais restritos e mais antigos não devem ter atualizações tão recentes, devido ao próprio ciclo de vida de um software.

5: Acho que a popularidade do sistema e a linguagem tenham uma relação direta, já que, inevitavelmente, a linguagem principal de um sistema atrai a atenção de desenvolvedores que trabalham naquela linguagem. E se a linguagem é popular, a tendência é existir mais desenvolvedores que trabalhem nela.

6: Espera-se que sim, já que correção de bugs encontrados e reportados, além da própria manutenção evolutiva são fatores vitais

para a popularidade de um sistema. Espera-se que a taxa de issues fechadas em relação à issues totais seja alta, não necessariamente o número bruto.

Metodologia:

Esta é uma pesquisa de cunho descritivo que realiza uma abordagem quantitativa. Foram minerados exatamente 1000 repositórios de software no GitHub (os mais bem avaliados) para análise, uma vez elaboradas as hipóteses. Foi elaborado um script Python com a finalidade de extrair e tratar os dados necessários via API do GitHub

Resultados:

1) Fora encontrado que a mediana da data de criação é a data 08/10/2014, totalizando 1982 dias, que são 6 anos e meio. Também percebemos que existem 843 repositórios com mais de 3 anos de criação, representando 84,3% das amostras.
2) Foi encontrado que a mediana de pull requests aceitas é de 281
3) Foi encontrado que a mediana de releases é de 7
4) Foi encontrado que a mediana de dias corridos desde a última atualização: 1 dias. Sendo encontrado apenas 3 resultados: (5/3/2020 ; 6/3/2020 e 7/3/2020)
5) Foram encontrados 586 repositórios que apresentam a linguagem principal entre as 10 mais populares.
6) Razão entre a mediana do número de issues fechadas e a mediana do número total de issues: 0,776119403 (77,6%)

Conclusão:

Comparando as hipóteses com os resultados, pode-se concluir que:

1) Repositórios populares são maduros.

2) Repositórios populares recebem uma taxa considerável de contribuição externa.

3) Repositórios populares têm poucas releases.

4) Esses repositórios são atualizados com muita frequência, quase diariamente, hipótese errada na parte do ciclo de vida do software.

5) Mais de metade dos sistemas populares estão escritos em linguagens de programação populares, o que é uma parcela muito significativa.
6) Repositórios populares têm um alto índice de issues fechadas, mais de 3/4.