PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Bacharelado em Engenharia de Software - Instituto de Ciências Exatas e Informática
Unidade Educacional Praça da Liberdade

Daniel Henrique Vargas
Davi Brandão Saldanha

Relatório Final
Análise Comparativa de Repositórios Python
Laboratório de Experimentação de Software

Belo Horizonte
2018

Sumário

1. Introdução

No processo de desenvolvimento de software, muitas vezes nos deparamos durante as etapas iniciais com o questionamento sobre qual linguagem seria a mais adequada para o projeto.

Quando o projeto requer um desenvolvimento mais ágil, com uma linguagem menos verbosa, mas ainda assim poderosa e eficiente, a linguagem Python surge como primeira opção de muitos.

Em visto disso, foi realizada a mineração e análise dos 1000 repositórios Python mais populares da plataforma GitHub, em comparação com os repositórios Python do Guido van Rossum, um dos idealizadores originais da linguagem, com o objetivo de analisar a qualidade de projetos desenvolvidos em Python.

Para alcançar este objetivo, esse trabalho visa responder às algumas questões de pesquisas, utilizando as seguintes métricas:

Objetivos	Questões de Pesquisa	Métricas
Analisar a qualidade de repositórios desenvolvidos na linguagem Python	Quais as características dos repositórios Python do Guido van Rossum?	● **Popularidade:** número de estrelas, número de watchers, número de forks ● **Tamanho:** linhas de código (LOC) ● **Atividade:** número de releases, frequência de releases (número de releases / dias) ● **Maturidade:** idade (em anos)
	Quais as características dos top-1000 repositórios Python mais populares?	
	Repositórios populares Python são de boa qualidade?	
	A popularidade influência nas características de repositórios Python?	

2. Metodologia

O processo de coleta e análise comparativa dos 1000 repositórios Python mais populares com os repositórios Python do Guido van Rossum foi realizado em duas etapas.

2.1.Definição da Baseline

A mineração dos dados foi feita por meio de uma requisição em forma de query realizada em uma API de GraphQL, disponibilizada pelo próprio GitHub, para obter informações sobre os repositórios Python do Guido van Rossum.

Entretanto, para poder obter os dados necessários de todos os repositórios, foi preciso fazer o uso de um sistema para repetir a requisição caso está falhasse.

Logo após realizar a consulta dos repositórios Python do Guido van Rossum, os dados foram enviados para um arquivo CSV, onde foram organizados, tratados e analisados.

Além disso, foram utilizadas as bibliotecas GitPython e Pygount para realizar a clonagem e a contagem de linhas de código (LOC), respectivamente, de todos os repositórios.

2.2.Seleção dos Repositórios Python

A mineração dos dados foi feita por meio de uma requisição em forma de query realizada em uma API de GraphQL, disponibilizada pelo próprio GitHub, para obter informações sobre os 1000 repositórios Python com mais de 100 estrelas da plataforma.

Entretanto, para poder obter os dados necessários de todos os 1000 repositórios, visto que o volume de informações era muito alto para a API obter de uma vez só, foi preciso fazer o uso de recursos como paginação e um sistema para repetir a requisição caso está falhasse.

Logo após realizar a consulta dos 1000 repositórios, os dados foram enviados para um arquivo CSV, onde foram organizados, tratados e analisados.

Além disso, foram utilizadas as bibliotecas GitPython e Pygount para realizar a clonagem e a contagem de linhas de código (LOC), respectivamente, de todos os 1000 repositórios.

2.3.Hipóteses

As seguintes hipóteses foram formuladas priori a análise dos resultados:

- **Hipótese 1 –** Dado o fato do Guido van Rossum ser um dos idealizadores originais da linguagem, é de se esperar que os repositórios Python dele sejam bem populares, não sejam tão grandes, tenham bastante atividade e sejam bem antigos.
- **Hipótese 2 –** Sendo Python uma das linguagens mais populares atualmente, é de se esperar que os repositórios Python mais populares tenham altos índices de popularidade, sejam razoavelmente grandes, tenham bastante atividade e sejam mais recentes.
- **Hipótese 3 –** Devido ao fato de Python ser uma linguagem pouco verbosa, podemos esperar um número de LOC relativamente baixo, já que escrevemos menos que outras linguagens populares utilizando o Python.
- **Hipótese 4 –** Já que Python é uma das linguagens mais populares atualmente, podemos esperar um número maior do que o normal de followers, releases, Stars, forks etc. Também podemos esperar que não demore para ocorrer atualizações, em menos de uma semana deve ocorrer uma atualização ao menos, justamente por ser uma linguagem muito usada e explorada nos dias de hoje.

3. Apresentação e Discussão dos Resultados

Após a organização, tratamento e análise dos dados, os seguintes resultados foram obtidos e comparados com as hipóteses originalmente formuladas.

3.1.Quais as características dos repositórios Python do Guido van Rossum?

Com base nos resultados obtidos, podemos perceber que os repositórios Python do Guido van Rossum não possuem índices de popularidade tão altos, realmente não são tão grandes, mas possuem praticamente nenhuma atividade e são bem mais recentes do

que o esperado. Todas as métricas foram avaliadas levando em consideração valores medianos.

Baseline						
Popularidade			Tamanho	Atividade		Maturidade
Nº de Estrelas	Nº de Watchers	Nº de Forks	LOC	Nº de Releases	Releases/dias	Idade (em anos)
49	6	3	798	0	0	3

3.2.Quais as características dos top-1000 repositórios Python mais populares?

Com base nos resultados obtidos, podemos perceber que os top-1000 repositórios Python mais populares realmente possuem índices de popularidade bem elevados e não são tão grandes. Contudo, possuem muito pouca atividade e são bem mais antigos do que o esperado. Todas as métricas foram avaliadas levando em consideração valores medianos.

Análise dos 1000 Repositórios Python Mais Populares						
Popularidade			Tamanho	Atividade		Maturidade
Nº de Estrelas	Nº de Watchers	Nº de Forks	LOC	Nº de Releases	Releases/dias	Idade (em anos)
4294,5	200	843	792	1	0,000671144	5

3.3.Repositórios populares Python são de boa qualidade?

Observando que a mediana de LOC dos repositórios de Python deu um resultado de 792 LOC, um número relativamente baixo, podemos concluir então que os repositórios Python são de qualidade pois atendem o que foi proposto e com poucas linhas de código, assim tendo uma boa relação de LOC por função.

3.4.A popularidade influência nas características de repositórios Python?

O número de releases não foi afetado pela popularidade, já que apresenta mediana de apenas uma release, já os outros dados ocorreram como o previsto, temos um grande número de followers, watchers e stars nas medianas, e menos de uma semana como a mediana para que uma atualização ocorra, portanto a popularidade é sim um motivador para que o repositório tenda a melhorar.

3.4.1. Análise do grupo Top

Top						
Popularidade			Tamanho	Atividade		Maturidade
Nº de Estrelas	Nº de Watchers	Nº de Forks	LOC	Nº de Releases	Releases/dias	Idade (em anos)
10573	435,5	2001,5	13137	2	0,001685729	5

3.4.2. Análise do grupo Bottom

Bottom						
Popularidade			**Tamanho**	**Atividade**		**Maturidade**
Nº de Estrelas	**Nº de Watchers**	**Nº de Forks**	**LOC**	**Nº de Releases**	**Releases/dias**	**Idade (em anos)**
2803,5	125,5	533	99,5	0	0	5