Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

BRUNO VIEIRA

MONOGRAFIA EM SISTEMAS DE INFORMAÇÃO I

COMO OS ARQUIVOS README EVOLUEM AO LONGO DO TEMPO EM PROJETOS DO GITHUB?

Belo Horizonte
2019/2

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

COMO OS ARQUIVOS README EVOLUEM AO LONGO DO TEMPO EM PROJETOS DO GITHUB?

por

BRUNO VIEIRA

Monografia em Sistemas de Informação I

Apresentado como requisito da disciplina de Monografia em Sistemas de Informação I do Curso de Bacharelado em Sistemas de Informação da UFMG

Prof. Dr. Andre Hora
Orientador

Belo Horizonte
2019/2

RESUMO

O Github é, atualmente, uma das maiores plataformas de hospedagem de código-fonte e outros arquivos importantes para um projeto de desenvolvimento, hospedando mais de 100 milhões de repositórios públicos e privados em seus servidores. Dentro destes projetos do Github podemos encontrar o arquivo README, uma forma de documentação do repositório que contém informações sobre o projeto, tais como, formas de utilizá-lo e como contribuir para o seu desenvolvimento.

Neste trabalho, o objetivo está centrado na investigação de como estes arquivos README evoluem ao longo do tempo nestes repositórios. São utilizados 50 repositórios públicos com mais de 10.000 estrelas (métrica do Github para popularidade de um repositório) e outros 50 repositórios públicos com 1.000 ou menos estrelas. A partir deste conjunto de dados estabelecido inicialmente, elaborou-se algumas estatísticas descritivas que caracterizam estes arquivos README e, com o auxílio de scripts em bash e bibliotecas de manipulação e visualização de dados em Python, gerou-se mapas de calor que indicam, em média, a posição nestes arquivos que mais sofrem alterações ao longo do tempo, por meio de commits.

Palavras-chave: README. Github. Repositórios de software. Evolução do README. Documentação de repositórios.

ABSTRACT

Nowadays, the Github is one of the largest platforms of code source hosting and other important files for a development project, hosting over 100 million public and private repositories on its servers. Inside these Github projects we can find the README file, a way to document the repository which contains information about the project, like, manners to use it and how to contribute to its development.

In this paper, the objective is to investigate how these README files evolve over time in these repositories. 50 public repositories with over 10,000 stars (Github metric for popularity of a repository) and 50 other public repositories with 1,000 stars or less are used. From this initially established dataset, we developed some descriptive statistics that characterize these README files and, with the help of bash scripts and Python data manipulation and visualization libraries, heat maps were generated that indicate, in average, the position in these files that change most over time through commits.

Keywords: README. Github. Software Repositories. README Evolution. Repositories Documentation.

LISTA DE FIGURAS

Figura 1 - Boxplots dos conjuntos de dados……15
Figura 2 - Mapa de calor das alterações por posição do arquivo README no conjunto de dados 1 (escala em 11 categorias).....16
Figura 3 - Mapa de calor das alterações por posição do arquivo README no conjunto de dados 1 (escala em 101 categorias).....16
Figura 4 - Mapa de calor das alterações por posição do arquivo README no conjunto de dados 2 (escala em 11 categorias).....17
Figura 5 - Mapa de calor das alterações por posição do arquivo README no conjunto de dados 2 (escala em 101 categorias).....17

LISTA DE TABELAS

Tabela 1 - Conjunto de dados 1……..11
Tabela 2 - Conjunto de dados 2……..12
Tabela 3 - Categorias de mudanças nos arquivos README……14
Tabela 4 - Frequência das categorias de mudanças nos arquivos README…….18

SUMÁRIO

RESUMO……...3
ABSTRACT……...4
LISTA DE FIGURAS……... 5
LISTA DE TABELAS……... 6
1 INTRODUÇÃO……..8
2 TRABALHOS RELACIONADOS……....9
3 DESENVOLVIMENTO DO TRABALHO……. 10
3.1 Coleta de Dados……... 10
3.2 Extração de Informação Quantitativa……12
3.3 Extração de Informação Qualitativa……...13
4 RESULTADOS E DISCUSSÕES……15
5 CONCLUSÃO E TRABALHOS FUTUROS……....19
6 REFERÊNCIAS…….. 20

1 INTRODUÇÃO

O Github é uma plataforma de hospedagem de código-fonte e outros arquivos importantes para um projeto de desenvolvimento, que se tornou popular entre empresas e a comunidade de desenvolvedores, em geral. Atualmente, existem mais de 100 milhões de repositórios de software hospedados no serviço, segundo dados da própria rede, dentre repositórios públicos e privados.

Com a popularização do serviço, muitos projetos de código aberto passaram a ser hospedados, incentivando a colaboração entre desenvolvedores de todo o mundo. Dentro destes repositórios existe a possibilidade de se utilizar o README, um arquivo de texto que contém informações sobre o projeto, tais como, um resumo e como utilizá-lo. Esse arquivo é importante por se tratar de uma forma de documentar atualizações no código para usuários e para desenvolvedores interessados, como explicado por Shonei et al.

Existem indícios de que há uma correlação entre a documentação de um projeto e a sua popularidade no Github (AGGARWAL, Karan; HINDLE, Abram; STROULIA, Eleni). Sendo o README uma forma de documentar, é relevante entender como esse arquivo evolui em diferentes repositórios conforme um projeto é atualizado.

Neste trabalho são analisados os arquivos README de 100 projetos do Github, separados em dois conjuntos de teste que representam os repositórios mais populares e os menos populares. Para os mais populares, foram selecionados 10 projetos com mais de 10.000 estrelas, ordenados de forma decrescente, para cada uma das 5 linguagens de programação mais utilizadas. Em adição a isso, para os menos populares, também foram selecionados 10 projetos com menos de 1.000 estrelas, ordenados de forma decrescente, para cada uma das 5 linguagens de programação mais utilizadas, seguindo este critério. O objetivo é verificar o que muda no arquivo ao longo do tempo de sua existência no repositório, analisando por meio dos commits o que foi alterado e onde a alteração foi feita.

A partir dos dados coletados, pode-se perceber o comportamento de mudança nos arquivos README dos repositórios selecionados por meio de mapas de calor, que informam os locais onde mais houveram alterações. Dessa forma, é possível fazer uma comparação entre os dois conjuntos de dados, demonstrando semelhanças entre eles.

2 TRABALHOS RELACIONADOS

Alguns trabalhos sobre os arquivos README e outras formas de documentação de projetos do Github já foram conduzidos, seguindo diferentes perspectivas. Ikeda et al., por exemplo, investigaram o conteúdo dos READMES de projetos escritos em Javascript, analisando apenas documentos escritos em inglês, com o intuito de saber (i) o que os desenvolvedores costumam escrever nos arquivos README e (ii) Se o tipo de projeto afeta como os desenvolvedores escrevem os arquivos README.

Prana et al. seguiram um caminho diferente no estudo do README, realizando uma pesquisa qualitativa com o intuito de classificar as seções do arquivo em diferentes categorias, automaticamente. Estas categorias geradas foram então mostradas para vinte profissionais da área de software que avaliaram a qualidade da categorização. Esse estudo se mostrou importante para melhorar a qualidade da documentação e entendimento dos arquivos README do Github.

Ainda se tratando da exploração dos arquivos README, mas se afastando um pouco do objetivo desta pesquisa, HASSAN e WANG desenvolveram um trabalho com a intenção de extrair do README as informações necessárias para a construção automática de softwares desenvolvidos em Java. Este tipo de ferramenta facilitaria muito o trabalho de desenvolvedores e pesquisadores, mostrando assim o potencial do arquivo README como ferramenta de documentação e alvo de estudos para melhor entendimento e aprimoramento.

Com isso, este trabalho se apoia nos estudos anteriores, realizados sobre o mesmo tema, e tem como foco a exploração da evolução dos arquivos README ao longo do tempo, mostrando como o arquivo costuma ser alterado. A partir dessa exploração, espera-se adquirir mais conhecimento que contribua para o entendimento dos arquivos README, assim como os outros trabalhos da área.

3 DESENVOLVIMENTO DO TRABALHO

3.1 Coleta de Dados

Para a realização das análises deste trabalho, foram montados dois conjuntos de dados seguindo alguns critérios de inclusão. O conjunto de dados 1 é formado por repositórios com maior popularidade no Github, enquanto o conjunto de dados 2 segue o caminho contrário, sendo composto por repositórios menos populares. De forma geral, os repositórios selecionados são de sistemas, *APIs* e bibliotecas de sistemas, desconsiderando-se qualquer outro tipo de repositório. Além disso, os repositórios selecionados devem estar escritos em Inglês ou Português. Outros critérios detalhados para escolha dos repositórios são explicados a seguir.

Conjunto de dados 1: Os 10 primeiros repositórios, ordenados de forma decrescente, com mais de 10.000 estrelas, para cada uma das 5 linguagens de programação mais populares no Github que atendem à este critério de busca. Para a construção deste conjunto de dados foi utilizada a seguinte string de busca no Github:

stars:>10000

Javascript	Python	Java	GO	C++
bootstrap	django	dubbo	etcd	bitcoin
react	flask	elasticsearch	frp	electron
vue	keras	glide	gin	swift
axios	thefuck	guava	go	tensorflow
create-react-app	youtube-dl	MPAndroid Chart	gogs	terminal
d3	ansible	okhttp	hugo	nw.js
node	httpie	retrofit	kubernetes	opencv
react-native	requests	RxJava	moby	protobuf
angular.js	scikit-learn	spring-boot	prometheus	pytorch
Font-Awesome	scrapy	spring-framework	syncthing	x64dbg

Tabela 1 - Conjunto de dados 1

Conjunto de dados 2: Os 10 primeiros repositórios, ordenados de forma decrescente, com número de estrelas menor ou igual a 1.000, para cada uma das 5 linguagens de programação mais populares no Github que atendem à este critério de busca. Para a construção deste conjunto de dados foi utilizada a seguinte string de busca no Github:

stars:<=1000

Javascript	Java	Python	HTML	Ruby
elevatezoom	BitHub	android-plus-plus	1pass	cache-money
forms	droid-fu	BicaVM	clowncar	event_ calendar
react-meteor	HorizontalVariableListView	cgt	geomicons-open	rocket_pants
run-sequence	android-player	cozy-setup	tent.io	facebooker
timbre.js	baasbox	Django-Socialauth	t.js	select2-rails
Notification	android-pedometer	hilbert	jquery-dropdown	stamp
connect-assets	CircularProgressDrawable	LINE	f8Developer Conference App	vestal_ versions
botwillacceptanything	Bubble-Notification	PEPS	kraken	json_spec
grunt-shell	Account Authenticator	python-beaver	rowGrid.js	webistrano
garkit	TeleHash	workflow	RWDPerf	cloud_crowd

Tabela 2 - Conjunto de dados 2

3.2 Extração de Informação Quantitativa

Após a seleção e clonagem dos repositórios para o estudo, a extração da informação foi feita utilizando, inicialmente, comandos GIT alocados em scripts Shell, para que todo o processo fosse feito de forma automatizada e replicável para qualquer outro conjunto de dados. Para a construção destes scripts, necessitou-se tratar casos específicos em que o arquivo README não possuía a extensão padrão ".md". Toda as variações de extensão para este arquivo são listadas a seguir:

- .md
- .rst
- .textile
- .adoc
- .rdoc
- .markdown
- .txt

Com exceção dos arquivos README com a extensão ".txt", todos os outros seguem o padrão de linguagem de marcação para compor uma formatação adequada e diferentes maneiras de exibição e padronização do documento, para melhor visualização da informação.

Os scripts em shell executados fazem uso dos comandos: *git log*, para levantamento do número de commits em cada repositório e para coleta dos IDs de cada um destes commits; *git diff*, para comparar o que houve de mudança entre um commit e outro e; *git show*, para exibir o tipo de mudança ocorrida e número da linha onde ocorreu esta mudança.

A partir destas informações extraídas, foi feita uma organização e tratamento, para melhor visualização dos dados, utilizando as bibliotecas Numpy, Pandas, Matplotlib, RegEx e Seaborn do Python, por meio do ambiente de programação por células Jupyter Notebook. Com o auxílio dessas ferramentas, foram gerados os mapas de calor e os boxplots dos commits que serão mostrados e discutidos na seção de resultados.

3.3 Extração de Informação Qualitativa

Em adição aos dados quantitativos coletados, foi feita uma análise manual, de cunho qualitativo, para analisar quais são os tipos de mudanças que ocorrem nos arquivos

README ao longo da sua história de commits. Para isso, foi realizado um processo de criação de categorias de mudança, seguindo um processo iterativo baseado no trabalho de PRANA et al.

Para essa etapa, foi utilizado apenas o conjunto de dados 1, por possuir uma quantidade suficiente de commits para realização desta tarefa. O processo seguido foi o de varredura manual de cada um dos 50 repositórios presentes no conjunto de dados e a leitura diagonal dos 5 últimos commits dos arquivos README de cada um desses repositórios. Por meio dessa leitura, foram criadas 13 categorias para classificar o tipo de mudança ocorrida no arquivo.

Categoria	**Descrição**
Atualização de links	Mudança de algum link que já estava presente no documento anteriormente.
Correção ortográfica/gramatical	Correção de pequenos erros de inglês ou concordância entre palavras.
Inserção de nova informação	Adição de uma nova informação que não estava presente no documento anteriormente.
Atualização de versionamento	Atualização da informação de versão do sistema ou de dependências do sistema.
Atualização de informação	Atualização de uma informação que já estava presente no documento.
Alteração de texto	Mudança em um texto explicativo, atualizando frases sem mudar o sentido geral do trecho.
Remoção de informação	Remoção total de alguma informação do documento.
Alteração de exemplo de código	Alteração em algum código utilizado na documentação para aplicação de uso ou instalação da ferramenta.
Formatação	Alteração na forma como a informação é distribuída e mostrada no documento, conforme regras de formatação.
Adição de link	Adição de novo link para alguma dependência importante do projeto.

Remoção de link	Remoção de um link presente anteriormente no documento.
Alteração na estrutura do código	Alteração na estrutura do código de marcação utilizado para compor o arquivo README.
Alteração de posição	Mudança de posição de alguma informação presente no documento.

Tabela 3 - Categorias de mudanças nos arquivos README

4 RESULTADOS E DISCUSSÕES

Inicialmente, foram gerados boxplots para cada um dos conjuntos de dados, a fim de caracterizá-los levando em conta o número de commits dos repositórios. Estes boxplots são exibidos a seguir, com breve explicação dos resultados encontrados.

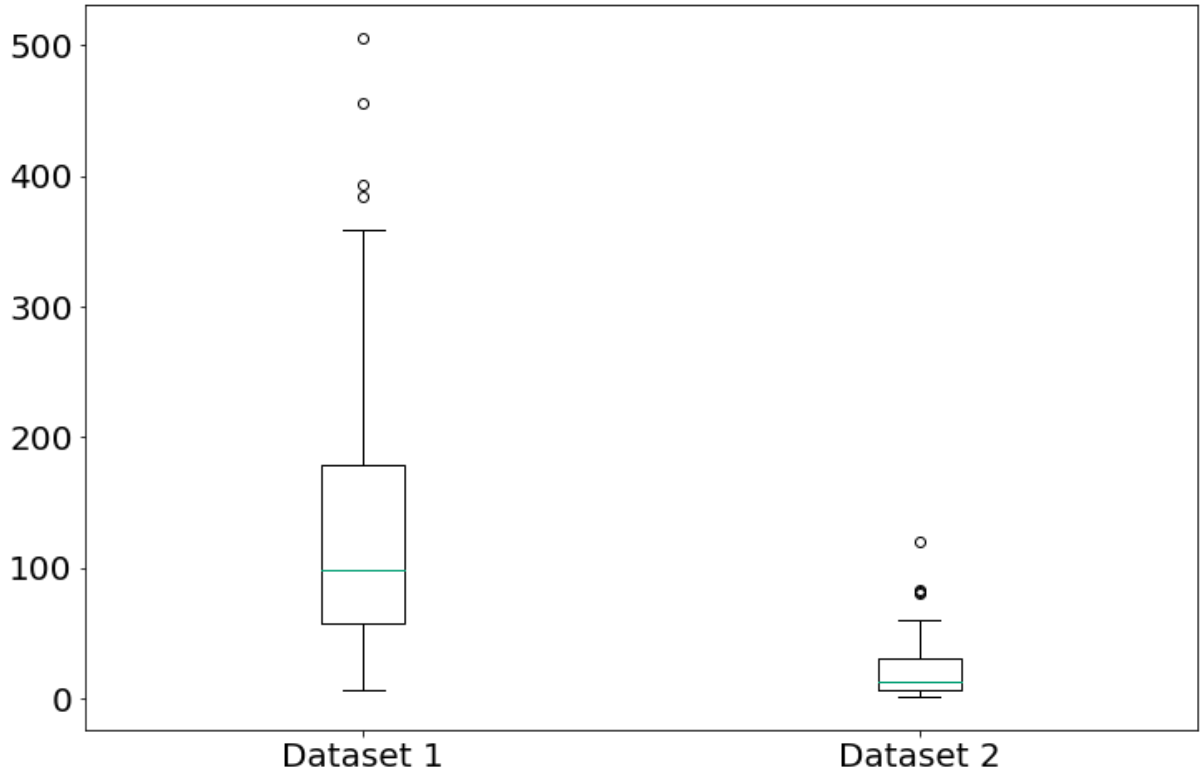

Figura 1 - Boxplots dos conjuntos de dados

Como visto na Figura 1, o número de commits presentes no primeiro conjunto de dados, para cada um dos 50 repositórios, é superior ao do segundo conjunto, com mediana centrada próxima a 100 commits em comparação à mediana centrada em aproximadamente 15 commits do dataset 2. Um número maior de commits por repositório é mais interessante, por disponibilizar mais informação a ser explorada estatisticamente e indicar com mais acurácia a informação presente nos mapas de calor dispostos a seguir.

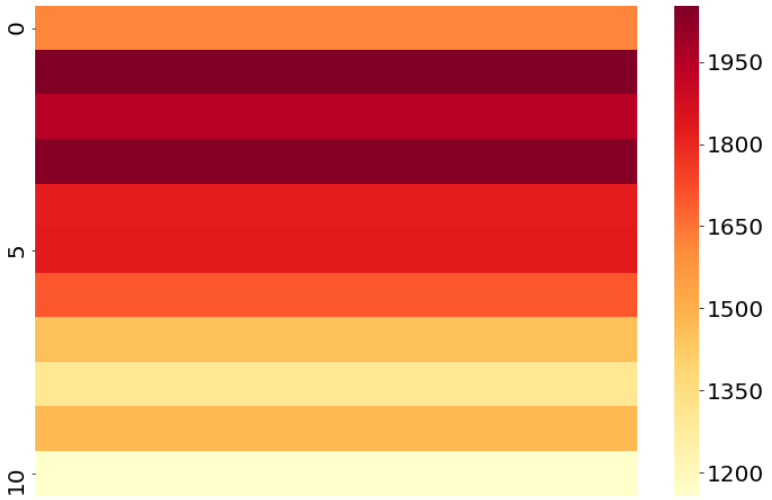

Figura 2 - Mapa de calor das alterações por posição do arquivo README no conjunto de dados 1 (escala em 11 categorias)

Figura 3 - Mapa de calor das alterações por posição do arquivo README no conjunto de dados 1 (escala em 101 categorias)

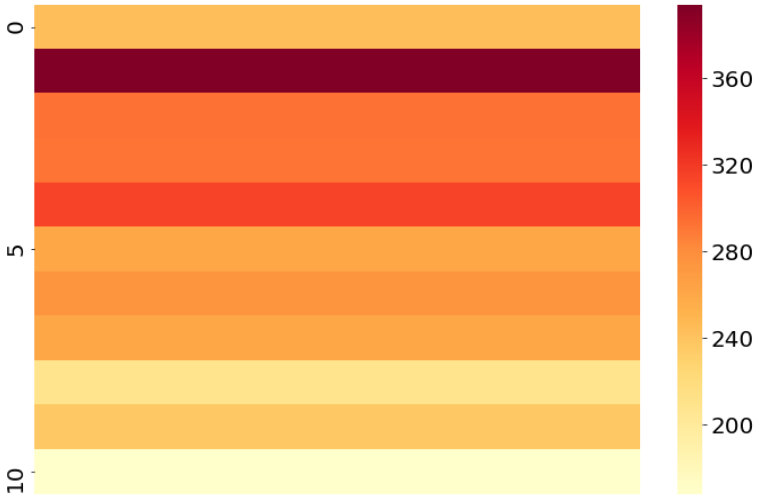

Figura 4 - Mapa de calor das alterações por posição do arquivo README no conjunto de dados 2 (escala em 11 categorias)

Figura 5 - Mapa de calor das alterações por posição do arquivo README no conjunto de dados 2 (escala em 101 categorias)

Na Figura 2 e na Figura 3 podemos ver o comportamento geral nos arquivos README do conjunto de dados dos repositórios mais populares. É possível observar que a maioria das mudanças tende a ocorrer entre o topo e o meio dos arquivos. Se levarmos em consideração o mapa de calor com mais categorias (Figura 3), fica mais claro que a maioria das mudanças está alocada no topo dos arquivos.

Na Figura 4 e na Figura 5, apesar de diferenças sutis, podemos observar, basicamente, o mesmo comportamento nos arquivos README dos repositórios menos populares, com concentração forte de mudanças no topo dos arquivos. Este resultado é interessante, primeiramente por demonstrar que existe um certo padrão independente do conjunto de repositórios utilizados. Em segundo lugar, apesar deste conjunto de dados ser mais pobre em informações, ele foi capaz de nos retornar dados coerentes com o que se esperava, conforme os resultados do conjunto de dados 1.

Categoria	**Frequência**
Atualização de links	21,6%
Correção ortográfica/gramatical	14,8%
Inserção de nova informação	13,6%
Atualização de versionamento	9,6%
Atualização de informação	9,2%
Alteração de texto	6,8%
Remoção de informação	6,4%
Alteração de exemplo de código	4,8%
Formatação	3,6%
Adição de link	3,2%
Remoção de link	2,8%
Alteração na estrutura do código	2,0%
Alteração de posição	1,6%

Tabela 4 - Frequência das categorias de mudanças nos arquivos README

Em complemento, podemos ver nos dados presentes na Tabela 4, quais os tipos de mudanças estão ocorrendo nos arquivos README, em paralelo aos locais de mudança, já demonstrados nos mapas de calor. Com base na frequência

apresentada, percebemos que as principais mudanças são causadas por Atualização de links e por correções ortográficas e/ou gramaticais. Os dados apresentados nesta tabela são apenas uma referência no contexto deste trabalho. Dado o número baixo de commits avaliados para cada um dos repositórios, não seria correto generalizar estas frequências para todos os projetos presentes no Github.

Todo o código e ferramentas necessárias para replicação deste trabalho podem ser encontrados no link: <https://github.com/brunoa15/monografia-1>

5 CONCLUSÃO E TRABALHOS FUTUROS

Conforme demonstrado na seção de resultados, em relação à posição dos arquivos README, estes costumam evoluir mais a partir do topo e meio. Esta informação por si só foi bastante representativa para o contexto deste trabalho e coincidiu para ambos os conjuntos de dados utilizados. Porém, como trabalho futuro, é ideal propor que esta informação seja validada, por meio de testes de hipótese, podendo assim, ser melhor generalizada para outros contextos de repositórios no Github.

Além disso, os tipos de categorias de mudanças geradas, podem, também, ser expandidas para um número maior de commits e de repositórios, buscando representar mais o universo de estudo. Esta tarefa poderia, inclusive, ser facilitada por meio do uso de ferramentas de aprendizado de máquina.

Por fim, caso os resultados encontrados neste trabalho fossem generalizados por meio de ferramentas estatísticas mais robustas, a investigação de possíveis fatores para este tipo de comportamento nos repositórios de software poderiam ser um tema importante de aprofundamento. Com base nos fatores levantados, seria possível melhorar a qualidade dos arquivos README dos repositórios de software do Github, contribuindo para projetos melhor documentados e estruturados.

6 REFERÊNCIAS

Thank you for 100 million repositories - The GitHub Blog. Disponível em: <https://github.blog/2018-11-08-100m-repos/> Acesso em: 4 Set. 2019

GitHub's products.
Disponível em: <https://help.github.com/en/articles/githubs-products> Acesso em: 4 Set. 2019

IKEDA, Shohei et al. An Empirical Study of README contents for JavaScript Packages. IEICE TRANSACTIONS on Information and Systems, v. 102, n. 2, p. 280-288, 2019.

PRANA, Gede Artha Azriadi et al. Categorizing the content of GitHub README files. Empirical Software Engineering, v. 24, n. 3, p. 1296-1327, 2019.

AGGARWAL, Karan; HINDLE, Abram; STROULIA, Eleni. Co-evolution of project documentation and popularity within GitHub. In: Proceedings of the 11th Working Conference on Mining Software Repositories. ACM, 2014. p. 360-363.

BORGES, Hudson; HORA, Andre; VALENTE, Marco Tulio. Understanding the factors that impact the popularity of GitHub repositories. In: 2016 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, 2016. p. 334-344.

HASSAN, Foyzul; WANG, Xiaoyin. Mining readme files to support automatic building of java projects in software repositories: Poster. In: Proceedings of the 39th International Conference on Software Engineering Companion. IEEE Press, 2017. p. 277-279.