

Identificação de Padrões de Comportamentos e Atividades de Usuários no GitHub

Vinícius Moreira Galvão¹, Hudson Silva Borges²

¹Faculdade de Computação (FACOM) – Universidade Federal do Mato Grosso do Sul (UFMS)
Campo Grande – MS – Brasil

`vinicius.galvao@ufms.br, hudson.borges@ufms.br`

Resumo. A plataforma do GitHub é uma das maiores plataformas atuais de hospedagem de código, utilizada por muitos usuários diariamente se tornou um elemento importante para o desenvolvimento de código. A plataforma foi muito estudada por pesquisadores na área de desenvolvimento de software, principalmente no processo evolutivo do repositório que hospeda o código-fonte desses softwares, com uma grande oportunidade de estudo na área de comportamento e de padrões dos usuários que utilizam essa plataforma. Com o intuito de compreender os comportamentos que os usuários apresentam nessa plataforma, neste trabalho foi realizado um estudo exploratório que visa analisar as atividades realizadas pelos usuários e identificar seus comportamentos mais comuns. O entendimento desses padrões de comportamentos permitirá identificar as atividades que os usuários normalmente realizam na plataforma e como esses padrões afetam o repositório. Com esse objetivo foi realizado uma análise sobre diversos dados extraídos com o uso da sua API (Application Programming Interface), para mostrar os comportamentos, a evolução das atividades e uma linha do tempo de um usuário, com o intuito de melhor compreender o comportamento desses usuários e como isso impacta na evolução do repositório e desenvolvimento de software. Os resultados obtidos a partir desse estudo, indicam que a maioria dos usuários realiza poucas atividades na plataforma do GitHub, sendo que a maioria das atividades realizadas por esses usuários envolvem somente a realização da atividade Starred, seguida da atividade de Reactions e da atividade de AuthorIssue, poucos usuários vão se envolver em atividades mais complexas como a atividade de AuthorPullRequest, e os usuários que contribuem com essas atividades de maior complexidade são também os usuários que possuem maior atividade no repositório e que realizam a maioria das contribuições para o crescimento do mesmo, mostrando que o crescimento do repositório está principalmente relacionado a um número pequeno de contribuidores e que eles são os principais responsáveis pela evolução do repositório e do crescimento do mesmo.

1. Introdução

A plataforma do GitHub é atualmente a principal plataforma para a hospedagem de código-fonte de programas, com mais de 10 milhões de repositórios, e seus usuários realizam diversas atividades diariamente nessa plataforma, como, por exemplo, a submissão de trechos de códigos para repositórios, reações a essas submissões, entre outros [Kalliamvakou et al. 2014]. Uma das razões da popularidade que envolve essa plataforma, é pelo fato de possuir diversas ferramentas que auxiliam os usuários a realizarem,

com maior facilidade, o controle de código na plataforma, bem como uma interface intuitiva para o usuário, e também a possibilidade da realização de interações sociais entre os usuários, auxiliando em todo processo de desenvolvimento de software e gerando diversas atividades que podem ser realizadas na plataforma, estimulando o desenvolvimento colaborativo que ocorre entre os mesmos [Dabbish et al. 2012].

Atualmente existe uma grande oportunidade de estudo sobre os comportamentos e padrões que esses usuários apresentam quando interagem nessa plataforma, assim um estudo realizando uma abordagem sobre esse assunto vai permitir compreender como os usuários se comportam normalmente em um repositório, influenciando como os responsáveis pelo repositório vão realizar o gerenciamento do mesmo e como os pesquisadores que estudam comportamentos sociais e práticas de desenvolvimento de software, podem compreender e melhorar a eficiência das práticas existentes, com isso e usando de referência o trabalho realizado por Cosentino *et al.* (2017), foi desenvolvido este artigo buscando compreender esses comportamentos e padrões dos usuários.

As diversas pesquisas analisadas sobre a plataforma do GitHub, como a pesquisa apresentada por Zhang *et al.* (2014), apresentavam um foco que envolvia principalmente o desenvolvimento de código e a evolução do repositório em si, com pouco foco no comportamento dos usuários, dessa forma buscou-se, com esse estudo, uma abordagem diferente focada nos padrões das atividades que esses usuários realizam e em estudar a tendência dessas atividades a partir da criação de uma linha do tempo.

Com o estudo desses comportamentos e padrões realizados pelos usuários sobre a plataforma do GitHub, permite-se uma melhor compreensão dos aspectos que envolvem o desenvolvimento de software e como esses comportamentos impactam sobre um repositório dentro dessa plataforma, podendo servir como uma forma de analisar as tendências que existem nesse desenvolvimento e possivelmente verificar formas de otimizar esse processo.

Dessa forma, visando encontrar essas tendências e comportamentos, realizou-se uma análise dos dados extraídos da plataforma do GitHub com o uso da sua API. Foi feito um estudo sobre 9.059 atividades realizadas por usuários, a um determinado repositório popular na plataforma do GitHub, buscando encontrar essas tendências e comportamentos, e verificar como ocorre a evolução dessas atividades ao longo do tempo.

Os resultados obtidos, permitiram a criação de uma linha do tempo mostrando como um usuário se comportou ao longo de um período, como também quais foram as atividades mais realizadas pelos usuários e os comportamentos que os usuários possuíam quando interagiam com o repositório, o que permitiu encontrar os tipos de usuários que interagiam com o repositório, e como esses comportamentos afetam na evolução do repositório.

Para a organização desse trabalho, foi realizado a divisão em 6 seções, nas quais são descritas a seguir. A Seção 1 (Introdução) que realiza a introdução deste trabalho. A Seção 2 (Trabalhos Relacionados), que comenta sobre os trabalhos analisados que serviram como base para esse estudo. Na Seção 3 (Background), ocorre a apresentação de informações e conceitos importantes para o entendimento desse estudo. Na Seção 4 (Metodologia), ocorre a explicação de como ocorreram as preparações e os processos realizados nesse artigo. A Seção 5 (Estudo exploratório) que analisa os dados obtidos e

mostra os resultados. A Seção 6 (Discussões) que realiza uma discussão sobre a análise realizada na seção anterior e nas dificuldades encontradas durante a produção desse artigo e a última Seção 7 (Conclusões) que finaliza o estudo apresentado nesse trabalho.

2. Trabalhos Relacionados

Conforme AlMarzouq *et al.* (2020), a obtenção de dados do GitHub através de sua API apresenta determinados desafios, porém é um fator importante para estudos na plataforma, que apresenta uma variedade de informações sobre projetos, usuários e organizações, demonstrando uma rica quantidade de dados de desenvolvimento armazenados que podem ser obtidos e utilizados por pesquisadores, porém tais dados devem ser obtidos com cuidado devido a limitações sobre a sua API e com cautela com relação à qualidade e a validação dos dados obtidos [AlMarzouq et al. 2020].

Segundo Onoue *et al.* (2013), cada repositório apresenta um tipo de desenvolvedor que possui um perfil com base nas atividades que realiza na plataforma do GitHub, tais características apresentam um impacto no tipo de contribuição que o desenvolvedor apresenta em um repositório. Os desenvolvedores que apresentam uma mistura de características tendem a serem considerados os melhores contribuintes, mostrando que em repositórios ativos na plataforma do GitHub existe uma variedade de perfis de desenvolvedores que realizam a manutenção e garantem a qualidade do repositório [Onoue et al. 2013].

Segundo Zhang *et al.* (2014) em plataformas como GitHub uma abundância de dados é disponível por ações de usuários realizados na plataforma e usar esses dados de maneira a obter as preferências destes permite analisar como um repositório pode se tornar de interesse a um usuário. Para isso foi usado uma *mineração de dados* com base em eventos como *Fork*, *Watch*, dentre outros de maneira a encontrar a relação dos eventos com o interesse do usuário ao repositório, tendo como resultados que diferentes tipos de projetos bem como tipos específicos de eventos como *Fork*, *Watch*, *Pull-Request* e *Member* afetam principalmente o interesse dos usuários [Zhang et al. 2014].

Conforme Consentino *et al.* (2017), para identificar como as práticas de desenvolvimento de software são influenciadas em plataformas de desenvolvimento social como o GitHub e como os usuários envolvidos podem otimizar seu método de colaboração, realizou um estudo exploratório analisando diversos artigos relacionados a plataforma do GitHub, filtrando os mais relevantes. Os resultados mostraram como ocorrem a organização dos repositórios, os eventos que ocorrem nos mesmos e como tais fatores impactam na produtividade do repositório [Cosentino et al. 2017].

Segundo Costa e Ponciano *et al.* (2018), para realizar a caracterização dos comportamentos dos programadores em relação a sua interação com os repositórios de software e para determinar como os comportamentos dos mesmos se relacionam ao sucesso do repositório é necessário definir objetivos nos quais envolvem: (i) definir as métricas de sucesso dos repositórios; (ii) coletar na plataforma do GitHub dados de amostra de um repositório; (iii) obter dados das interações dos programadores para identificar perfis de engajamento; e (iv) relacionar perfis de engajamento com os dados das amostras coletados [Costa and Ponciano 2018]. Seus resultados mostram que existem múltiplos perfis de programadores, sendo o principal tipo o chamado *Programador Codificador* baseado principalmente em enviar *Commits* que são pequenas modificações submetidas para o

repositório, e o perfil mais raro, porém o mais desejado, o chamado *Programador Socializador* que realizam as mais diversas atividades na plataforma do GitHub e buscam inovações.

3. Background

Para este estudo foram utilizadas ferramentas e diversos métodos para a extração dos dados, o tratamento dos mesmos, a análise de dados e para a identificação de padrões de comportamentos e atividades. As subseções a seguir apresentam os principais conceitos, métodos, e ferramentas usadas neste trabalho.

3.1. GitHub

O estudo foi realizada em cima de uma plataforma *Open Source* de colaboração de dados e controle de versão nomeada GitHub. O GitHub possui mais de 10 milhões de repositórios, sendo atualmente a principal plataforma para a hospedagem de código-fonte de programas [Kalliamvakou et al. 2014].

A Figura 1 apresenta a interface de um repositório chamado JDA (disponível no GitHub), nessa página pode ser observado diversas ferramentas e opções disponíveis ao usuário bem como o código-fonte armazenado no repositório em questão.

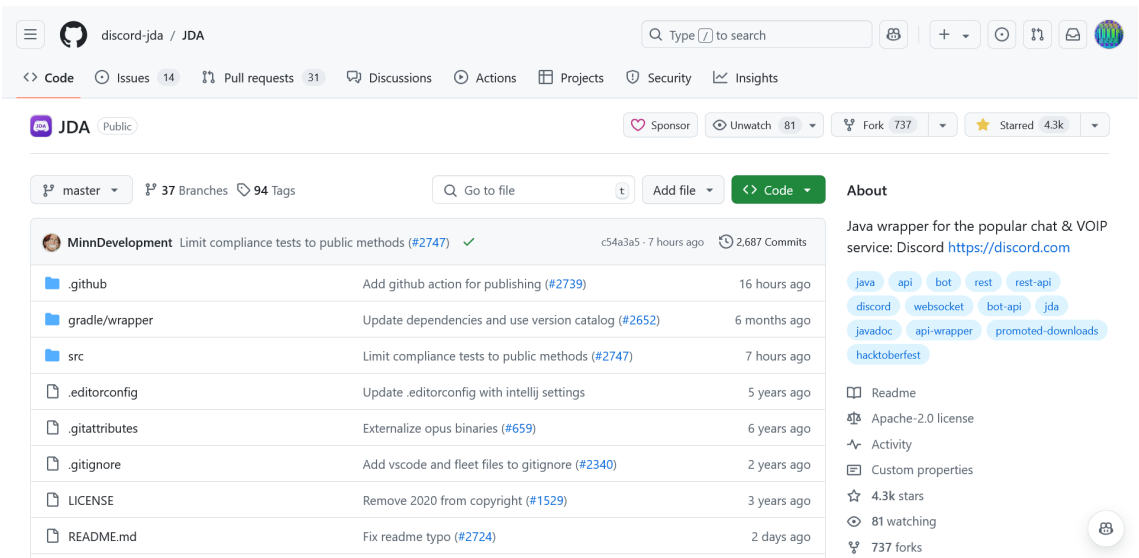


Figura 1. Imagem de um repositório no GitHub
Fonte: Autoria própria, 2024

O principal foco do GitHub consiste em permitir a seus usuários hospedarem artefatos de software, de forma que o usuário tenha controle dos recursos nessa aplicação ou repositório (conforme chamado dentro do GitHub). O GitHub oferece uma variedade de ferramentas que os usuários podem utilizar para controlar as alterações no seu código, auxiliar no desenvolvimento do mesmo e permitir que outras pessoas também realizem modificações, notifiquem sobre problemas encontrados e solicitem envios de código para

complementar o código-fonte do repositório, tudo com bastante controle e segurança disponibilizado tanto para o usuário responsável pelo repositório como para o usuário que está interagindo com o mesmo.

É possível procurar por outros repositórios que possam ser de determinado interesse para um usuário, bem como realizar diversas atividades dentro do GitHub que permitem *acompanhar* um determinado repositório, verificar mudanças, ser notificado de atualizações, auxiliar no desenvolvimento de outros repositórios, notificar problemas, procurar documentações relacionadas a um determinado repositório, dentre uma gama de ferramentas que possibilitam diversas interações ao usuário.

Além disso, temos as interações sociais na plataforma que permite ao usuário, interagir de forma colaborativa com diversas pessoas, bem como *seguir* outros usuários e ser notificado de diversas ações ou interações que acontecem nos repositórios ou em locais de discussão que existem na plataforma, assim como a possibilidade de analisar de forma gráfica os acontecimentos e atividades de ocorreram.

Devido à abundância de ferramentas disponíveis, muitos dados podem ser coletados na plataforma, sendo os mesmos obtidos através de sua API (Application Programming Interface) que permitem uma comunicação por meio de um conjunto de definições e protocolos. No caso do GitHub o uso da REST API permite uma comunicação entre servidor e cliente para a obtenção dos dados internos em conjunto com a GraphQL API, que flexibiliza a obtenção dos dados a partir de uma linguagem focada em realizar uma busca de dados com precisão e com menor complexidade, que pode ser utilizada para criar *scripts* e aplicativos que automatizam o processo de integração com o GitHub bem como a obtenção dos dados internos da plataforma, esses dados são o foco do estudo para a obtenção da análise e padrões de comportamentos e atividade dos usuários, sendo os dados classificados na plataforma como *eventos* permitindo categorizar as diversas atividades que ocorreram na mesma bem como as diversas atividades que os usuários realizaram.

3.2. Eventos

Na plataforma do GitHub, *eventos* estão relacionados as ações de usuários. Por exemplo, quando um usuário favorita um repositório por meio da funcionalidade *star*, é gerado um registro que pode ser posteriormente coletado por meio das APIs oferecidas pelo GitHub. Tais registros, que incluem diversas ações na plataforma, é o que denominamos *eventos* neste trabalho. Para o estudo em questão, podemos classificar majoritariamente os eventos de interesse como: *watchers*, *tags*, *stargazers*, *reactions*, *pull requests*, *issues* e *releases*. Segundo a documentação da plataforma¹, temos a seguinte definição para as funcionalidades que geram os eventos estudados:

Watchers: representado com o símbolo de um olho na página de repositórios no GitHub, permite sinalizar que o usuário apresenta interesse em ser notificado de atividades, como *issues*, *releases*, *pull requests*, *discussions* e alerta de seguranças que aconteceram no repositório. A Figura 2 apresenta a funcionalidade *watchers* para o repositório JDA. Este

¹Disponível em: <https://docs.github.com/en/rest?apiVersion=2022-11-28>
Acesso em: 5 de out, 2024.

repositório possui 81 *watchers* registrados, gerando assim a mesma quantidade de registro de eventos na API do GitHub.

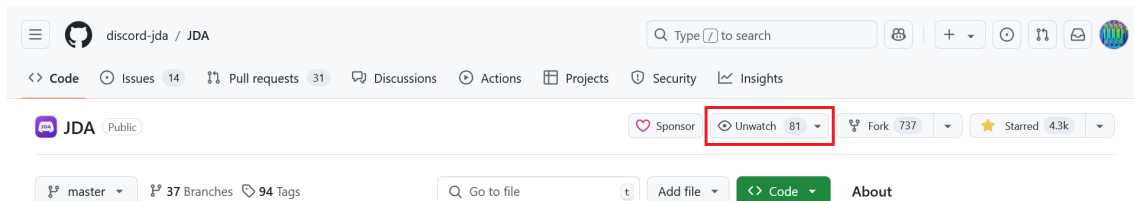


Figura 2. Em destaque a aba *watchers*

Fonte: Autoria própria, 2024

Tags: representado com o símbolo de uma etiqueta, é a marcação específica de um *commit*, uma alteração no código enviado para o repositório, muito usado para sinalizar uma versão do programa ou lançar alguma especificação que necessita ser nomeada. A Figura 3 demonstra quantidade de eventos do tipo *tags* que existe no repositório, para o repositório JDA a quantidade é de 94.

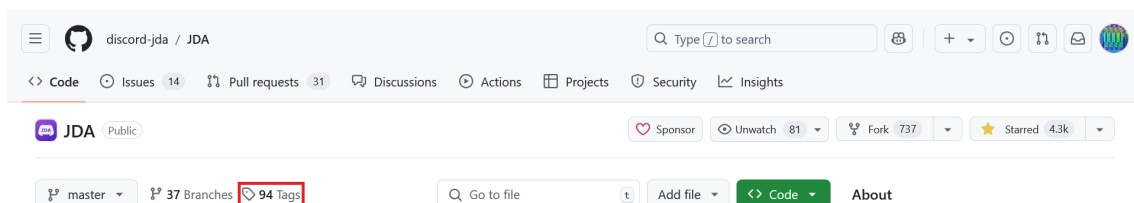


Figura 3. Em destaque as *tags* do repositório

Fonte: Autoria própria, 2024

Stargazers: representado com o símbolo de uma estrela numa aba em um repositório no GitHub, indica algo como *favoritar* determinado repositório ou tópico, permitindo fácil acesso ao mesmo e indicações de conteúdo ou interesse recomendados pelo GitHub em sua *dashboard* que é a página principal do GitHub. Além disso, é muito usado como parâmetro de referência para a indicação do sucesso e popularidade de um repositório. A Figura 4 indica a quantidade de pessoas que marcaram o repositório JDA como favorito simbolizado pela estrela preenchida, caso a pessoa não favorite o repositório o símbolo da estrela fica *vazio*.

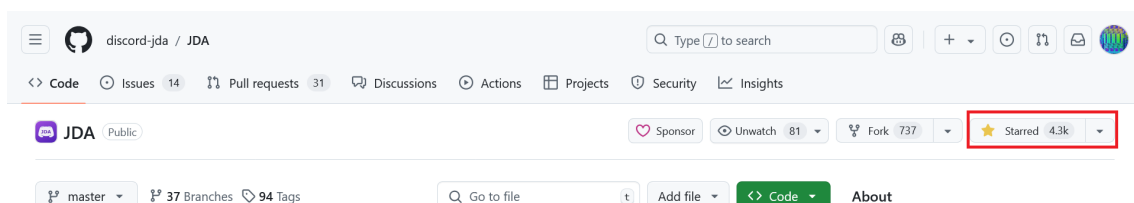
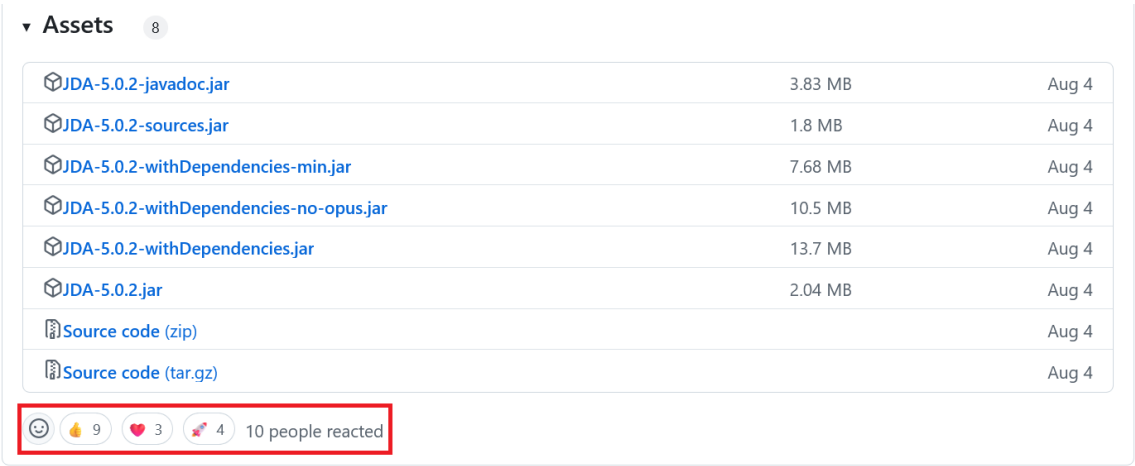


Figura 4. Em destaque um repositório que foi *starred*.

Fonte: Autoria própria, 2024

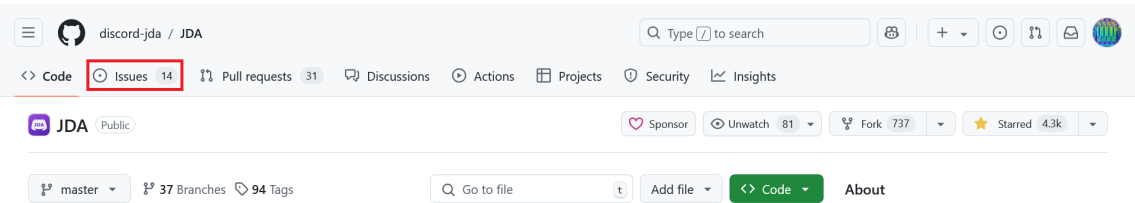
Reactions: demonstra o uso de emojis em algum outro evento, que pode ser um *release*, determinada *issue*, um comentário a respeito de um *commit* ou algum outro evento relacionado no GitHub, basicamente uma demonstração de uma reação social a determinado evento. A Figura 5 demonstra a quantidade de reações feitas em uma determinada *release* do repositório JDA.



▼ Assets 8		
JDA-5.0.2-javadoc.jar	3.83 MB	Aug 4
JDA-5.0.2-sources.jar	1.8 MB	Aug 4
JDA-5.0.2-withDependencies-min.jar	7.68 MB	Aug 4
JDA-5.0.2-withDependencies-no-opus.jar	10.5 MB	Aug 4
JDA-5.0.2-withDependencies.jar	13.7 MB	Aug 4
JDA-5.0.2.jar	2.04 MB	Aug 4
Source code (zip)		Aug 4
Source code (tar.gz)		Aug 4
👍 9 🍷 3 🧡 4 10 people reacted		

Figura 5. Uma *release* do repositório JDA que teve *reactions*.
Fonte: Autoria própria, 2024

Issues: representado por um símbolo de um círculo com um ponto no meio, sinaliza que algum usuário do GitHub identificou algum pequeno problema no código, teve alguma ideia com relação a determinada parte do código, ou serve como um indicador de problemas mais severos como falhas de segurança, servindo não somente como uma fonte de detectar problemas, mas como um histórico do progresso desenvolvido no repositório. A Figura 6 está mostrando a quantidade de *issues* em aberto no repositório, sendo um bom indicador do estado em que se encontra o código.



discord-jda / JDA		Type ↵ to search	
<> Code	Issues 14	Pull requests 31	Discussions
JDA (Public)		Sponsor	Unwatch 81
Fork 737		Starred 4.3k	
master 37 Branches 94 Tags		Go to file t Add file >> Code About	

Figura 6. Em destaque a aba *issues* com os problemas em aberto.
Fonte: Autoria própria, 2024

Pull Requests: representado por um símbolo de uma linha com duas circunferências nas extremidades sendo interceptada por uma seta, sinaliza o pedido de um envio de um grande conjunto de código ou *commits*, geralmente utilizado quando algum problema foi resolvido por um usuário do GitHub (normalmente colaboradores do repositório) ou quando algo novo vai ser implementado, evento bastante importante e complexo que deve

ser avaliado antes de ser incorporado ao repositório e ao código-fonte. A Figura 7 apresenta a informação sobre a quantidade de eventos de *pull requests* que estão atualmente disponíveis para serem realizados no repositório JDA.

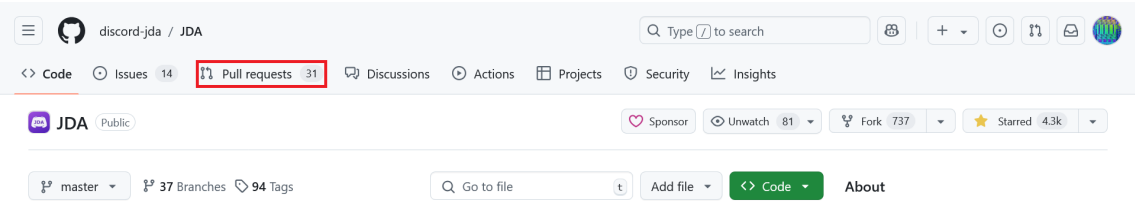


Figura 7. Em destaque uma aba que sinaliza os *pull requests* em aberto.
Fonte: Autoria própria, 2024

Releases: representado também com um símbolo de etiqueta, é a disponibilização de uma versão do repositório estável ou geralmente de uma grande quantidade de modificações feitas e prontas para serem distribuídas, é normalmente feito através da compilação do código-fonte e posteriormente distribuído para download com os arquivos compactados, tal evento é baseado em versões que se baseiam com o evento de *tags* que representam determinados marcos no desenvolvimento do código-fonte e do repositório. A Figura 8 está mostrando a *release* mais atual que foi lançada no repositório.



Figura 8. Imagem demonstrando *releases* do repositório.
Fonte: Autoria própria, 2024

3.3. Mineração de Padrões

Primeiramente é necessário definir o conceito da palavra *padrão*, no contexto deste trabalho padrão significa um conjunto de sequências que vão se repetir a partir das atividades ou eventos descritos na subseção anterior, assim como é importante definir o conceito de da palavra padrão *padrão*, a definição de *comportamento* também é importante e nesse contexto temos que a definição de *comportamento* é uma atividade que realizada em grande frequência por usuários.

Para melhor compreensão do conceito de padrão, podemos ilustrar alguns exemplos, suponhamos que temos alguns usuários que realizaram diversas das atividades mostradas subseção anterior, uma grande maioria desses usuários realizam as atividades de *stargazer* e *reaction* conjuntamente, dessa forma encontramos um padrão que envolve essas duas atividades.

Outro exemplo de forma mais numérica, supondo que tenhamos 100 usuários que realizaram variadamente atividades como *stargazer*, *reaction*, *authorpullrequest*, *authorissue*, dentro desse grupo de usuário 80 deles realizam as atividades de *authorpullrequest* e *authorissue*, com essa informação foi possível encontrar um padrão que envolve a realização dessas duas atividades.

Com esses exemplos é possível reforçar e compreender melhor o conceito de um *padrão*, que pode ser basicamente definido como um conjunto que vai se repetir em determinada frequência.

Todos os dados obtidos através desses eventos são importantes para o processo de extração dos padrões de comportamento. Após seu tratamento e padronização, tais dados serão a base para a extração dos padrões e atividades dos usuários, bem como para os estudos que vão ser possíveis após esta análise.

Com base nesses conceitos, temos que a mineração de dados é a busca em larga escala de forma repetitiva em um conjunto grande de dados para determinar padrões e tendências [Sidhu et al. 2014]. Outro conceito importante relacionado a mineração de dados, no qual serve como base para o algoritmo *FPGrowth* que foi usado nesse estudo, são as regras de associação, na qual se um elemento chamado de antecedente é encontrado nos dados e posteriormente o elemento antecedente é encontrado junto de outro elemento nomeado consequente no conjunto de dados, teremos então o que é chamado de associação.

A associação, de forma exemplificada, é um elemento que se encontra junto de outro, caso tenhamos uma atividade de *release* e posteriormente essa atividade está junta da atividade de *tags* então temos uma associação entre essas duas atividades.

Junto deste conceito temos os critérios de suporte e confiança, que envolvem respectivamente a indicação da frequência de um determinado elemento no conjunto de dados e o número de vezes que as regras de associação foram válidas, e que vão servir como indicadores na análise dos dados e na identificação de padrões.

Para o exemplo numérico citado anteriormente, por exemplo, seu suporte teria um valor 80% para a atividade conjunta de *authorpullrequest* e *authorissue*, com relação aos valores de confiança, a sua variação ia ocorrer em torno da ocorrência da regra de associação entre essas atividades proporcionalmente, um valor maior de ocorrências de associação indica um maior valor de confiança. Com isso podemos realizar a definição formal de suporte e confiança, com suporte sendo um conjunto no qual é definido a proporção de transações da base de dados que contém esse conjunto e confiança a regra na qual para ser encontrada utiliza-se do valor de suporte da interseção entre dois conjuntos diferentes, dividido pelo suporte do primeiro conjunto.

4. Metodologia

Neste estudo, é fundamental a disponibilidade de uma base de dados sólida e que possua o maior número de registros de eventos possíveis. Contudo, obter tais dados pode ser uma tarefa desafiadora, assim como diversos trabalhos na literatura reportam.

Os principais meios de obtenção de dados para estudos que fazem tais análises envolvem diretamente a coleta desse conjunto de dados das APIs do GitHub (i.e., Rest

API e GraphQL API) ou a obtenção de bases de dados pré-processadas e disponibilizadas por terceiros (e.g., GHTorrent ou GHArchive).

Embora muitas das bases pré-processadas ofereçam uma facilidade na obtenção, as mesmas são geralmente limitadas e com relação aos dados oferecidos, muitos são desatualizados, uma vez que as pessoas que realizam a manutenção sobre esses dados executam o processo de tempos em tempos [Borges et al. 2016]. Por outro lado, a coleta diretamente da API do GitHub pode ser desafiadora e demorada, já que existem mecanismos de proteção e regras de uso conscientes implementadas pela própria plataforma.

Neste trabalho optou-se pela utilização de uma ferramenta de coleta pública, que usa diretamente da API do GitHub. A ferramenta escolhida foi o GITTRENDS.APP², que faz a coleta dos dados e gera uma base de dados para cada projeto desejado, sendo que associado a cada projeto será disponibilizado os dados de diversos eventos em tabelas diferentes.

Além disso, considerando as características deste estudo, optou-se pela análise de somente um único projeto com muitas atividades e que envolvesse uma comunidade ativa, para ser possível entender melhor o desafio a ser enfrentado. Contudo, ressalta-se que o trabalho desenvolvido visa ser aplicado em quaisquer outros repositórios ou projetos por terceiros no futuro.

Portanto, foi definido como projeto de estudo o projeto chamado JDA (JAVA DISCORD API)³, sendo bastante utilizado no desenvolvimento de bots para a plataforma Discord. Além de sua importância, o repositório do projeto é bastante ativo, possui uma larga comunidade e oferece muitas possibilidades de análise pela grande quantidade de dados disponível.

4.1. Preparação dos Dados

Considerando que o ponto de maior interesse deste trabalho é a identificação de padrões de comportamento de usuários, o primeiro passo necessário após a obtenção dos dados, foi em realizar um processo de preparação dos dados, que inclui a busca, identificação, limpeza e organização dos mesmos com intuito de gerar linhas temporais de atividades de cada usuários. Para facilitar esse processo, foi criado um conjunto de *scripts* automatizados para tal processo. A seguir, são apresentados detalhes dos passos realizados.

4.1.1. Organização e Processamento de Dados

Nesta etapa, o objetivo foi de identificar usuários e suas atividades nas diferentes bases de dados. Inicialmente, foi feita uma análise exploratória manual usando de ferramentas de visualizações de bases de dados, como o PGADMIN⁴. Após a análise exploratória, foi desenvolvido um *script*⁵ para automatizar a identificação dos dados relevantes para o projeto, no link disponibilizado está o código do *script* criado e um pacote de replicação. A seguir, são listados as categorias de dados mais relevantes para o estudo e dados im-

²Disponível em: <https://github.com/gittrends-app> Acesso em: 5 de out, 2024.

³Disponível em: <https://github.com/discord-jda/JDA> Acesso em: 5 de out, 2024.

⁴Disponível em: <https://www.pgadmin.org/> Acesso em: 5 de out, 2024.

⁵Disponível em: <https://github.com/LinuxT25/TCC> Acesso em: 5 de out, 2024.

portantes que nos permitiram consolidar uma nova base de dados para a identificação dos padrões.

Actors: Na tabela de dados com informações de usuários, foi extraído todos os IDs nela disponível. Esses identificadores são importantes já que a base de dados inicial segue um modelo estruturado e as demais tabelas simplesmente referenciam tais IDs. No contexto deste trabalho, esses identificadores são muito importantes porque nos ajudam a identificar desenvolvedores que tiveram atividade no repositório para, posteriormente, construirmos suas respectivas linhas do tempo.

Issues: Na tabela contendo registros das *Issues* do repositório, foram extraídos os campos: *assignees*, retornando um valor booleano para indicar se o usuário foi designado para esta *issue*; *created_at*, retornando um valor em formato de data indicando quando foi realizado a atividade de criação de *issue* pelo usuário sinalizando implicitamente que ele foi o autor da *issue*; *last_edit_at*, retornando em formato de data a última vez que o usuário realizou a atividade de editar a *issue*; *participants*, retornando um valor booleano para indicar se o usuário foi um participante dessa *issue*.

Pull Requests: Na tabela contendo registros de *Pull Requests*, foram extraídos os campos: *assignees*, retornando um valor booleano para indicar se o usuário foi designado para esta atividade de *pull request*; *created_at*, retornando um valor em formato de data indicando quando foi realizado a atividade de criação de *pull request* pelo usuário; *last_edited_at*, retornando em formato de data a última vez que o usuário realizou a atividade de editar um *pull request*; *participants*, retornando um valor booleano para indicar se o usuário foi um participante dessa *pull request*; *merged_at*, retornando um valor em formato de data indicando quando foi realizado a atividade de *merge* pelo usuário; *suggested_reviewers*, retornando um valor booleano indicando se o usuário foi sugerido para revisar essa atividade de *pull request*.

Reaction: Na tabela contendo registros das *Reaction*, foram extraídos a partir do campo: *created_at*, no qual retornou um valor em formato de data, indicando quando foi realizado a atividade de *reaction* pelo usuário.

Releases: Na tabela contendo registros das *Releases*, foram extraídos a partir do campo: *created_at*, no qual retornou um valor em formato de data, indicando quando o usuário realizou alguma atividade de *release* para o repositório.

Stargazers: Na tabela contendo registros dos *Stargazers*, foram extraídos a partir do campo: *starred_at*, no qual retornou um valor em formato de data, indicando se o usuário realizou a atividade de *stargazer* no repositório.

Taggers: Na tabela contendo registros dos *Taggers*, foram extraídos a partir do campo: *tagger*, no qual retornou um valor em formato de data, indicando quando o usuário realizou a criação de alguma *tag* para o repositório.

Watchers: Na tabela contendo registros dos *Watchers*, foi-se verificado se o usuário existia na tabela, caso positivo foi retornado um valor do tipo booleano, indicando que o usuário realizou a atividade de *watcher* no repositório.

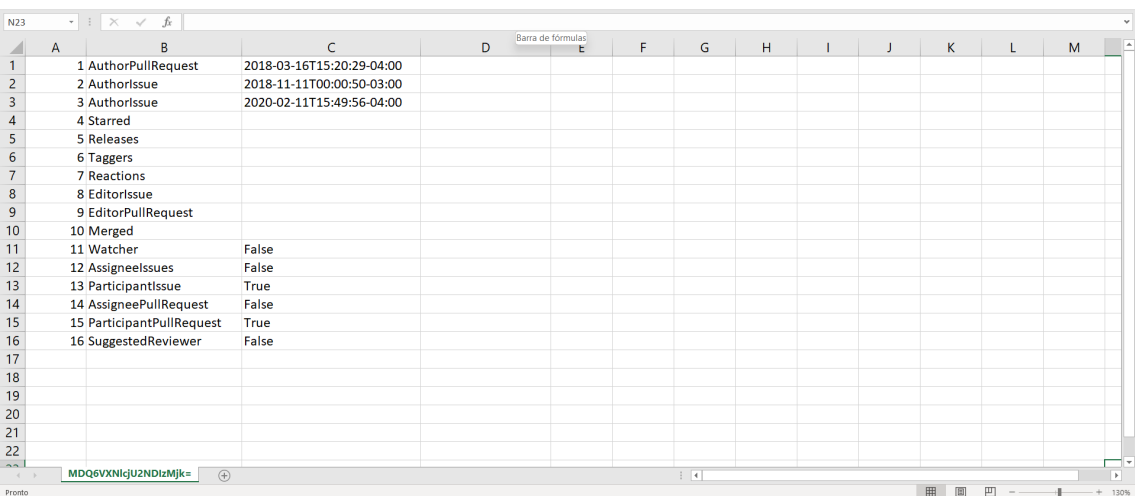
4.2. Recuperação da Linha do Tempo de Eventos

Após o termino do processo de extração dos dados, o nosso objetivo foi de reconstruir uma linha do tempo contendo os eventos de cada um dos usuários do projeto analisado. Para cada usuário, visamos organizar todos os eventos em uma lista ordenada pela data de ocorrência dos mesmos.

Antes da junção de todos os eventos em uma lista, que seria exportada, os dados foram filtrados para padronização, de forma que os campos que apresentavam formatos de data foram convertidos para o padrão ISO 8061, bem como os campos que possuíam ausência de dados foram categorizados como vazio. Após o fim de todo processo de filtragem, todas as tuplas que possuíam campo de data foram ordenadas de maneira crescente, ignorando os campos que apresentavam ausência de dados e por fim unidas as tuplas que apresentavam campo com formato booleano em uma única lista que representava o ID do usuário, com todas suas atividades realizadas.

Após o termino desse processo, os dados foram exportados para duas fontes complementares: (i) para um arquivo no formato *.csv* que possuía o nome do arquivo como o ID do usuário e dentro do arquivo suas atividades, e para (ii) uma base de dados no qual foi criado uma tabela contendo os campos: *user_id*, *activity*, *activity_date*, nos quais continham respectivamente o id do usuário, sua atividade realizada, e a data de realização das atividades. A primeira fonte foi utilizada para uma análise exploratória manual, enquanto a segunda fonte foi usada nos próximos passos deste trabalho.

Para ilustrar como foi armazenado a série temporal, será mostrado na Figura 9 seu formato armazenado em *.csv* e na Figura 10 no banco de dados, foi escolhido um usuário com poucas atividades para simplificação da ilustração.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	1	AuthorPullRequest	2018-03-16T15:20:29-04:00										
2	2	AuthorIssue	2018-11-11T00:00:50-03:00										
3	3	AuthorIssue	2020-02-11T15:49:56-04:00										
4	4	Starred											
5	5	Releases											
6	6	Taggers											
7	7	Reactions											
8	8	EditorIssue											
9	9	EditorPullRequest											
10	10	Merged											
11	11	Watcher	False										
12	12	AssigneeIssues	False										
13	13	ParticipantIssue	True										
14	14	AssigneePullRequest	False										
15	15	ParticipantPullRequest	True										
16	16	SuggestedReviewer	False										
17													
18													
19													
20													
21													
22													

Figura 9. Série temporal armazenada em formato *.csv*
Fonte: Autoria própria, 2024

Data Output Messages Notifications			
	user_id text	activity text	activity_date date
1	MDQ6VXNlcjU2NDIzMjk=	AuthorPullRequest	2018-03-16
2	MDQ6VXNlcjU2NDIzMjk=	AuthorIssue	2018-11-11
3	MDQ6VXNlcjU2NDIzMjk=	AuthorIssue	2020-02-11

Total rows: 3 of 3 Query complete 00:00:00.100 Ln 1, Col 81

Figura 10. Série temporal armazenada no banco de dados

Fonte: Autoria própria, 2024

4.3. Identificação de Padrões e Regras de Associação

Para identificar padrões de comportamento nas atividades dos usuários, foi utilizado o algoritmo *FPGrowth*. Neste trabalho, optamos por utilizar de uma solução existente e disponível no repositório *CHONYY/FPGROWTH_PY*⁶. Contudo, também fizemos algumas adaptações, que envolveram modificações na entrada, na saída e em alguns processos do algoritmo, para o mesmo poder atender a todas as necessidades do nosso trabalho.

Considerando que o algoritmo escolhido depende de informações parametrizadas, como suporte e confiança, neste estudo decidimos explorar um conjunto amplo de valores para identificação da melhor configuração para a base de dados. Para o suporte, foi definido valores com base na quantidade total de usuários que existiam no repositório (usando-se como referência o número de entradas que existiam na tabela com informações dos usuários). Assim, foi definido que o estudo deveria considerar os seguintes valores: 50%, 25%, 10%, 5%, 2% e 1%. Por exemplo, um valor de 50% implicaria que para ser considerado um padrão, o conjunto de atividades identificadas deveria ter sido realizado por pelo menos 50% da base dos usuários. Para a confiança, foi estabelecido um valor padrão de 10% com objetivo de obter mais regras, mas evitando regras com baixo nível de qualidade.

5. Estudo Exploratório

Nesta seção será feito uma abordagem em cima dos dados e dos resultados obtidos a partir da análise realizada sobre os mesmos, foi utilizado como base para essa análise os conceitos apresentados na Seção 4 (Metodologia) e as características que puderem ser obtidas em cima desses dados.

5.1. Sobre o Repositório JDA

O repositório JDA (Java Discord API) é um grande repositório focado no desenvolvimento de uma API com foco na implementação de bots para a plataforma Discord. Apresenta uma grande variedade de usuários (aproximadamente 19 mil usuários), de contribuidores (aproximadamente 150 contribuidores) e de *releases* (aproximadamente 90 lançamentos).

⁶Disponível em: https://github.com/chonyy/fpgrowth_py, Acesso em: 5 de out, 2024.

Devido à diversidade de dados que pode ser coletada desse repositório, foi extraído um conjunto base desses dados usado para o estudo apresentado, coletados em 16 de novembro de 2022, como o repositório JDA apresenta uma comunidade vastamente ativa e com inúmeros contribuintes, muitos novos dados foram gerados, mas não serão incluídos nesse estudo apresentado. Na Tabela 1, é apresentado a quantidade total de dados disponível por tabela utilizados para a realização dessa análise.

Na tabela apresentada é possível verificar todas as tabelas que foram extraídos os dados utilizados nesse estudo, o método de extração foi exemplificado na seção de metodologia, mais especificamente na Subseção 4.1.1 (Organização e processamento de dados), sendo elas: *actors* (indicando a quantidade de usuários que realizaram alguma atividade no repositório), *issues* (os *bugs* que foram identificados nesse repositório), *pull_requests* (as solicitações para a realização de *pull_request* feitas ao repositório), *reactions* (as reações realizadas no repositório), *releases* (a quantidade de *releases* que foi feita no repositório), *stargazers* (o número de estrelas dadas ao repositório), *tags* (a quantidade de tags geradas no repositório) e *watchers* (a quantidade de pessoas que se inscreveram para receber notificações do repositório).

Tabela 1. Tabelas Utilizadas para Realização da Análise

Nome da Tabela	Quantidade de Registros
actors	4.263
stargazers	3.385
reactions	1.559
pull_requests	1.389
issues	823
watchers	81
tags	72
releases	64

5.2. Construção da Linha do Tempo dos Usuários

Com os dados organizados iniciou-se então o processo da criação de uma linha do tempo das atividades de um usuário como exemplificado na Subseção 4.2 (Recuperação da linha do tempo de eventos), para a realização desse procedimento foi inicialmente analisado as atividades datadas ao longo do período em que os dados foram extraídos com a finalidade de verificar a evolução das mesmas.

Para ilustrar a linha do tempo das atividades foi então criado um gráfico contendo todas as atividades datadas ao longo do período em que os dados foram extraídos para ilustrar esse processo de evolução como mostrado na Figura 11.

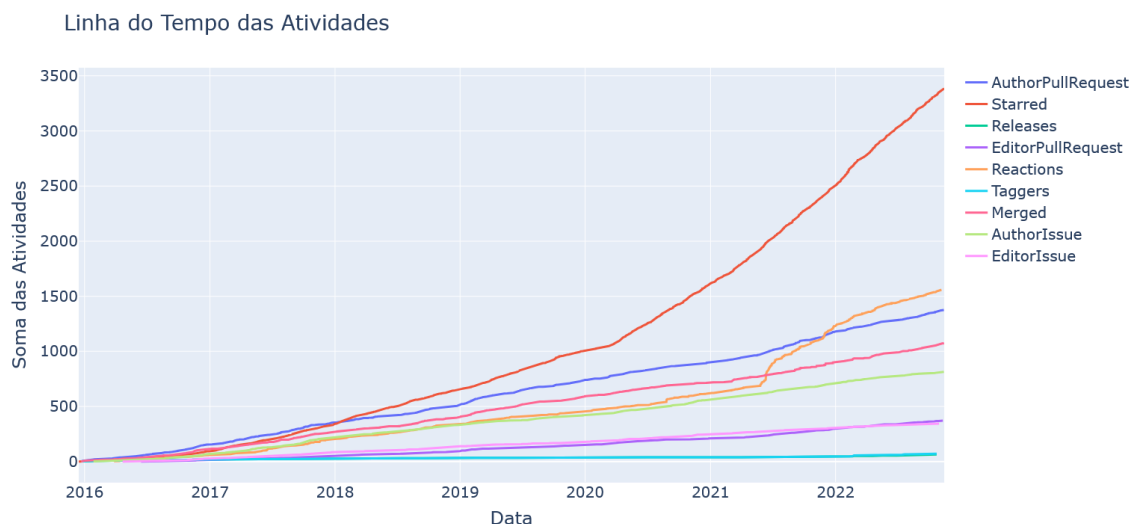


Figura 11. Evolução das atividades ao longo de um período

Fonte: Autoria própria, 2024

Analisando a figura temos o eixo horizontal, representado pela data limitada pelo período em que os dados foram extraídos, 16 de novembro de 2022, e temos o eixo vertical, representando a soma do número de atividades que ocorreu até determinada data. Dessa forma, teremos aparecendo na ordem ao longo desse período, *AuthorPullRequest* com o pico de 1.375 atividades, *Starred* com um pico de 3.385 atividades, *Releases* com um total de 64 atividades, *EditorPullRequest* com um total de 372 atividades, *Reactions* com 1.559 atividades, *Taggers* com 72 atividades, *Merged* com 1.074 atividades, *AuthorIssue* com 814 atividades e o *EditorIssue* com 344 atividades.

É possível observar com esses resultados, que a atividade que apresenta maior crescimento é a *Starred* algo que é esperado por ser uma atividade simples de ser realizada pelos usuários e demonstra que os mesmos apresentam determinado interesse no repositório, posteriormente a atividade de *Reactions* supera a partir do final de 2021 a atividade de *AuthorPullRequest* como a segunda atividade que mais cresceu ao longo do período, indicando que a partir de determinado crescimento do repositório existe um maior engajamento dos usuários em atividades do tipo social. Em terceiro no crescimento a atividade *AuthorPullRequest*, indicando um aumento nas solicitações de envio de código para a base do repositório, servindo também como um indicador do crescimento do mesmo com maiores usuários contribuindo para esse fator.

Logo em seguida é observável que a atividade de *Merged*, está geralmente relacionada a atividade de *AuthorPullRequest*, é possível concluir então que o crescimento dessas atividades seguem uma margem semelhante, isso não indica necessariamente que todas as atividades de *Merged* estão diretamente relacionadas a atividade de *AuthorPullRequest*, mas que existe uma correlação que acontece entre as duas. Com relação aos usuários, como a atividade de *Merged* é uma atividade que necessita de determinadas permissões para serem realizadas no repositório devido a medidas de segurança, e possível supor que os usuários que possuem essas permissões são ativos nesse repositório.

Em quinta posição de crescimento ao longo do gráfico, a atividade de *AuthorIssue*

indicando que os usuários notificaram problemas existentes no repositório, essa atividade está relacionada diretamente ao crescimento do mesmo, visto que quanto maior a quantidade de alterações e de atividades realizadas maior será a quantidade de problemas que vão aparecer em conjunto com essas modificações, servindo também como um indicador do engajamento dos usuários no repositório.

Posteriormente, é possível observar uma sequência de atividades com menor potencial de crescimento por serem mais específicas, que vão depender exclusivamente de outras atividades para serem realizadas, sendo elas: *EditorPullRequest*, *EditorIssue*, *Taggers* e *Releases*. Sendo *EditorPullRequest* uma atividade diretamente relacionada a *AuthorPullRequest* indicando que o usuário realizou modificações nessa atividade, e o mesmo irá ocorrer para a atividade de *EditorIssue* com a atividade de *AuthorIssue*, essas modificações mostram interesse dos usuários em aprimorar ou corrigir esses eventos, servindo como parâmetro para estudo do comportamento e indicando que usuários que realizam essas modificações geralmente contribuem para essas duas atividades em questão devido ao seu crescimento ser semelhante.

Por fim, as últimas duas atividades, *Taggers* e *Releases*, diretamente relacionadas entre si e com outras atividades como o *AuthorPullRequest* e *Merged*, essas últimas atividades apresentam um menor potencial de crescimento por serem mais complexas, visto que dependem exclusivamente de outras atividades e indicam grandes modificações que ocorreram no repositório, também só podem ser realizadas por usuários que possuem determinadas permissões, levando então ao seu baixo potencial de crescimento, e servindo como indicador mostrando que os usuários que possuem essas permissões apresentam um comportamento no qual eles contribuem ativamente neste repositório.

Concluindo essa análise da evolução de todas as atividades ao longo do período no qual os dados foram extraídos, podemos então realizar esta mesma análise sobre um usuário mostrando a linha do tempo das atividades que o mesmo realizou no repositório, como apresentado na Figura 12, no qual mostra ao longo do ano de 2022, atividades que um usuário realizou.



Figura 12. Linha do tempo de atividades de um usuário.

Fonte: Autoria própria, 2024

Analisando a imagem é possível observar que esse usuário realizou diversas interações de cunho social ao longo do ano de 2022, e também contribuiu com o repositório realizando atividades como *AuthorIssue* e *AuthorPullRequest*.

As atividades realizadas por esse usuário também condizem com a análise realizada sobre todas as atividades apresentada anteriormente, como pode ser observado na

figura esse usuário realizou uma atividade de *Starred* e diversas atividades de *Reactions*, contribuindo para a soma das atividades apresentadas anteriormente.

Além disso, é importante observar que a linha do tempo apresentada é uma simplificação, devido ao usuário ser relativamente novo em contribuições ao repositório e não apresentar um esquema complexo de atividades, algo que ocorre com diversos usuários, o que tornaria a visualização da linha do tempo dificultosa.

Outro fator que é importante ser observado, são que as atividades representadas e utilizadas para a criação dessas figuras e na qual vão ser utilizadas para a identificação de padrões são somente as atividades datadas, desta forma diversas atividades que foram extraídas, mas que não possuem data, não foram representadas, como as atividades de *Watcher*, *AssigneeIssues*, *ParticipantIssue*, *AssigneePullRequest*, *ParticipantPullRequest* e *SuggestedReviewer*.

Com isso é concluído a linha do tempo das atividades e exemplificado a linha do tempo de um usuário, com as observações obtidas da partir da análise feita sobre esses gráficos, será possível auxiliar no processo de identificação de padrões para tirar conclusões sobre os comportamentos dos usuários no GitHub, que será apresentado na próxima seção.

5.3. Identificação dos Padrões

Nesta seção foi realizado a identificação dos padrões e das regras de associação usando o algoritmo *FPGrowth*, da forma que havia sido apresentado na Seção 4.3 (Identificação de padrões e regras de associação), a partir do uso desse algoritmo foi possível verificar os tipos de comportamentos que os usuários apresentam com relação às atividades de determinado repositório.

Para evitar vícios e otimizar o algoritmo *FPGrowth*, foi desconsiderado atividades do mesmo tipo realizadas por um único usuário, dessa forma caso um usuário, por exemplo, tenha realizado 100 atividades de *Taggers* será considerado que ele realizou essa atividade somente uma vez para as identificações de padrões dos usuários.

A primeira análise foi realizada usando-se um valor de suporte de 50% em cima de um valor total de usuários de 4.263, resultando em um valor de 2.131,5, esse valor foi dividido pelo total de usuários que realizaram alguma atividade datada sendo esse valor de 4.109, obtendo um valor resultante de suporte igual a 0,5187 no qual foi aplicado no algoritmo.

O suporte com valores altos eliminou praticamente todas as atividades, com exceção da atividade *Starred* que apresentava um valor de suporte de aproximadamente 82,35%, indicando que a maioria dos usuários que realizam alguma atividade no repositório, acabam por realizar a atividade de *Starred*, mostrando perfis de usuários definidos e que apresentam impactos diferentes no repositório como foi apresentado na pesquisa realizada por Costa e Ponciano *et al.* (2018).

A fim de refinar os resultados na segunda análise decaiu-se o valor de suporte, para diversificar os resultados e obter melhores perspectivas. Usando então o valor de 25% em cima da quantidade total dos usuários apresentados anteriormente, obteve-se um valor de 0,2593 no qual foi inserido no algoritmo.

Para essa análise não foi obtido nenhuma mudança significativa no resultado, encontrando-se os mesmos valores obtidos anteriormente, os resultados da segunda análise permitiram verificar que existem uma grande diversidade entre as atividades que os usuários realizam, o que gera valores de suporte mais baixos, justificando os valores de suporte mais alto filtrarem grande parte das atividades.

Para a terceira análise, decaiu-se mais o valor de suporte usando um valor de 10%, obtendo-se um valor de 426,3 em cima do número de usuários que dividido novamente pela quantidade total de usuários que realizaram alguma atividade datada, foi possível de obter um valor de suporte igual a 0,1037 que foi inserido no algoritmo.

Para esse valor já foi possível de observar algumas mudanças, como o aparecimento de novas atividades como *AuthorIssue* apresentando aproximadamente 13,45% de valor de suporte e *Reactions* que apresentou aproximadamente 13,99% de valor de suporte, uma melhoria muito significativa com relação aos resultados obtidos anteriormente, porém ainda não foi o suficiente para apresentar regras de associações com confiança de 10%.

Reduzindo ainda mais o valor de suporte para essa quarta análise e usando o valor de 5%, foi obtido o resultado de 205,45, que dividido pela quantidade total de usuários resultou em um valor de suporte de 0,05 que foi inserido no algoritmo.

Para esse valor de suporte mudaram-se muitas coisas, foi encontrado novas atividades, como *EditorIssue* com valor de suporte de aproximadamente 5,32% e combinações de atividades como *EditorIssue* com *AuthorIssue* com valor de suporte de aproximadamente 5,3% e *Reactions* com *Starred* com valor de suporte de aproximadamente 5,91%, bem como o aparecimento das primeiras associações.

As associações obtidas nessa quarta análise, indicam que para usuários que realizam atividade de *EditorIssue* existe uma grande chance de aproximadamente 99,54%, que o usuário tenha realizado a atividade de *AuthorIssue*, curiosamente a associação inversa, ou seja, usuários que realizaram a atividade de *AuthorIssue* apresentam somente uma chance de aproximadamente 39,42%, de realizar a atividade de *EditorIssue*, levando a indicar que os usuários que editaram uma *issue* são geralmente os autores da mesma, porém nem todo autor de uma *issue* realiza alguma modificação nessa atividade.

Por fim, temos a última associação obtida a partir desse valor de suporte, indicando que usuários que realizaram a atividade de *Reactions* tem aproximadamente uma chance de 42,26%, de terem realizado a atividade de *Starred*, um indicativo interessante visto que não temos o inverso (usuários que realizaram a atividade de *Starred* com relação à atividade de *Reactions*), com esse valor de suporte e de confiança, servindo como um indicativo que se um usuário realizou alguma reação no repositório ele tem uma chance razoável de ter favoritado esse mesmo repositório.

Indo para a quinta análise e reduzindo o valor de suporte para o valor de 2%, é obtido o resultado de 85,26 em cima da quantidade total de usuários, esse valor quando dividido pela quantidade total de usuários resultou em um valor de suporte de 0,0207 usado no algoritmo.

A partir desse suporte temos um aumento considerável da lista de atividades e de suas combinações, que ocorreram devido a novas atividades que apareceram como

EditorPullRequest com valor de suporte de aproximadamente 2,38% e *AuthorPullRequest* com valor de suporte de aproximadamente 4,98% e novas combinações de atividades como *AuthorPullRequest* e *EditorPullRequest* com um valor de suporte de aproximadamente 2,38%, *AuthorPullRequest* e *Starred* com um valor de suporte de aproximadamente 2,43%, *Reactions* e *AuthorPullRequest* com valor de suporte de aproximadamente 2,45%, *Reactions* e *AuthorIssue* com valor de suporte de aproximadamente 3,45% e a última combinação de *AuthorIssue* e *Starred* com valor de suporte de aproximadamente 3,55%.

É notável que observando a quinta análise, não somente temos muitas combinações novas como também combinações com atividade que já haviam aparecido em análises mais iniciais, como ocorre entre *AuthorIssue* e *Starred*, indicando um número baixo da frequência dessas duas atividades por um usuário com relação ao número massivo de atividades do tipo *Starred* que foi realizada.

Temos também novas regras de associação, indicando que usuários que realizam a atividade de *AuthorPullRequest* tem aproximadamente 47,80% de chance de realizarem a atividade de *EditorPullRequest*, um valor bem diferente ocorre caso seja o inverso, indicando que um usuário que realizou a atividade de *EditorPullRequest* tem 100% de chance de realizar a atividade de *AuthorPullRequest*, uma regra de associação muito parecida com o que havia acontecido com as atividades de *AuthorIssue* e *EditorIssue*.

Outra regra de associação, envolve a atividade de *AuthorPullRequest* e a atividade de *Starred*, indicando que se o usuário realizou a atividade de *AuthorPullRequest* então temos uma chance de aproximadamente 48,78% de chance que ele realizará a atividade de *Starred*, um indicativo que caso um usuário tenha feito um *PullRequest* então ele terá aproximadamente 50% de chance de favoritar esse repositório.

Outro conjunto de regras de associação que foi obtido envolve a atividade de *Reactions* e *AuthorPullRequest*, com um indicativo de 17,56% de chance de um usuário que realizou a atividade de *Reactions* ter realizado a atividade de *AuthorPullRequest*, e seu inverso já ocorre uma variação significativa sendo ela de 49,26%, indicando que um usuário que realiza a atividade de *Reactions* tem uma chance pequena de fazer uma atividade de *AuthorPullRequest*, mas que um usuário que realiza a atividade de *AuthorPullRequest* tem chances significativamente maiores de realizar uma atividade de *Reaction*.

Mais um conjunto de regra de associação observável envolve as atividades de *Reactions* e *AuthorIssue*, muito parecido com a regra apresentada anteriormente, teremos que usuários que realizam a atividade de *Reactions* vão apresentar uma chance de 24,69% de realizarem a atividade de *AuthorIssue* e seu inverso será de 25,67%, uma chance relativamente equilibrada.

Por fim a última regra de associação, que foi possível de ser obtida nessa quinta análise, envolve as atividades de *AuthorIssue* e *Starred*, servindo como um indicativo que se um usuário realizou a atividade de *AuthorIssue* ele possui aproximadamente 26,40% de chance de ter realizado a atividade de *Starred*, essa regra não possui inverso para o suporte e confiança apresentados.

Para a sexta e última análise foi utilizado o valor de suporte de 1% sobre os usuários, obtendo-se um valor de 42,63 que dividido pelo total de usuários gerou um valor de suporte de 0,0103 no qual foi aplicado no algoritmo.

Esse valor de suporte é o mais importante para o estudo atual, representando todas as atividades analisadas anteriormente e todas as regras de associação que vão definir os padrões e comportamentos dos usuários no GitHub.

Para ilustrar todas as atividades que já foram mencionadas e para a simplificar a apresentação dos resultados será mostrado na tabela abaixo, os resultados obtidos na última análise, ilustrando na Tabela 1 as diversas combinações de atividades com seus respectivos valores de suporte e quantidade.

Tabela 2: Atividades e seus Valores de Suporte

Atividades Realizadas	Quantidade	Valor de Suporte
<i>Starred</i>	3384	82,35%
<i>Reactions</i>	575	13,99%
<i>AuthorIssue</i>	553	13,45%
<i>Starred, Reactions</i>	243	5,91%
<i>EditorIssue</i>	219	5,31%
<i>EditorIssue, AuthorIssue</i>	218	5,30%
<i>AuthorPullRequest</i>	205	4,98%
<i>Starred, AuthorIssue</i>	146	3,55%
<i>Reactions, AuthorIssue</i>	142	3,45%
<i>AuthorPullRequest, Reactions</i>	101	2,45%
<i>AuthorPullRequest, Starred</i>	100	2,43%
<i>EditorPullRequest</i>	98	2,38%
<i>AuthorPullRequest, EditorPullRequest</i>	98	2,38%
<i>AuthorPullRequest, AuthorIssue</i>	76	1,84%
<i>EditorIssue, Reactions</i>	74	1,80%
<i>EditorIssue, Reactions, AuthorIssue</i>	73	1,77%
<i>Starred, Reactions, AuthorIssue</i>	71	1,72%
<i>AuthorPullRequest, Reactions, Starred</i>	64	1,55%
<i>Starred, EditorIssue</i>	63	1,53%
<i>Starred, EditorIssue, AuthorIssue</i>	62	1,50%
<i>Reactions, EditorPullRequest</i>	58	1,41%
<i>AuthorPullRequest, Reactions, EditorPullRequest</i>	58	1,41%
<i>Starred, EditorPullRequest</i>	54	1,31%
<i>Starred, AuthorPullRequest, EditorPullRequest</i>	54	1,31%
<i>AuthorPullRequest, Reactions, AuthorIssue</i>	52	1,26%
<i>AuthorPullRequest, AuthorIssue, Starred</i>	49	1,19%
<i>AuthorIssue, EditorPullRequest</i>	45	1,09%
<i>AuthorPullRequest, AuthorIssue, EditorPullRequest</i>	45	1,09%

Com a Tabela 1, é bem observável que a maioria das atividades como havia sido anteriormente analisado envolvem a atividade de *Starred*, logo em seguida com as atividades de *Reactions* e *AuthorIssue*, algo esperado por serem as atividades mais comuns que um usuário pode realizar no GitHub, e as atividades que envolvem grandes combinações

como *AuthorPullRequest*, *AuthorIssue* e *EditorPullRequest*, sendo as menos realizadas pelos usuários.

Com relação às regras de associação, para simplificar será mostrada na Tabela 2 somente as que possuem no mínimo 40% de chance de ocorrer, de maneira a identificar comportamentos mais constantes dos usuários para mostrar padrões de atividades.

Tabela 3: Regras de Associação

Atividades	Atividade Relacionada	Ocorrência	Porcentagem
<i>AuthorIssue, EditorPullRequest</i>	<i>AuthorPullRequest</i>	45	100%
<i>Starred, EditorPullRequest</i>	<i>AuthorPullRequest</i>	54	100%
<i>Reactions, EditorPullRequest</i>	<i>AuthorPullRequest</i>	58	100%
<i>EditorPullRequest</i>	<i>AuthorPullRequest</i>	98	100%
<i>EditorIssue</i>	<i>AuthorIssue</i>	219	99,54%
<i>EditorIssue, Reactions</i>	<i>AuthorIssue</i>	74	98,64%
<i>Starred, EditorIssue</i>	<i>AuthorIssue</i>	63	98,41%
<i>AuthorPullRequest, AuthorIssue</i>	<i>Reactions</i>	76	68,42%
<i>AuthorPullRequest, AuthorIssue</i>	<i>Starred</i>	76	64,47%
<i>AuthorPullRequest, Starred</i>	<i>Reactions</i>	100	64,00%
<i>AuthorPullRequest, Reactions</i>	<i>Starred</i>	101	63,36%
<i>AuthorPullRequest, AuthorIssue</i>	<i>EditorPullRequest</i>	76	59,21%
<i>EditorPullRequest</i>	<i>Reactions</i>	98	59,18%
<i>EditorPullRequest</i>	<i>AuthorPullRequest, Reactions</i>	98	59,18%
<i>AuthorPullRequest, EditorPullRequest</i>	<i>Reactions</i>	98	59,18%
<i>AuthorPullRequest, Reactions</i>	<i>EditorPullRequest</i>	101	57,42%
<i>EditorPullRequest</i>	<i>Starred</i>	98	55,10%
<i>EditorPullRequest</i>	<i>Starred, AuthorPullRequest</i>	98	55,10%
<i>AuthorPullRequest, EditorPullRequest</i>	<i>Starred</i>	98	55,10%
<i>Starred, AuthorPullRequest</i>	<i>EditorPullRequest</i>	100	54,00%
<i>AuthorPullRequest, Reactions</i>	<i>AuthorIssue</i>	101	51,48%
<i>Reactions, AuthorIssue</i>	<i>EditorIssue</i>	142	51,40%
<i>Reactions, AuthorIssue</i>	<i>Starred</i>	142	50,00%
<i>AuthorPullRequest</i>	<i>Reactions</i>	205	49,26%
<i>AuthorPullRequest, Starred</i>	<i>AuthorIssue</i>	100	49,00%
<i>AuthorPullRequest</i>	<i>Starred</i>	205	48,78%
<i>Starred, AuthorIssue</i>	<i>Reactions</i>	146	48,63%
<i>AuthorPullRequest</i>	<i>EditorPullRequest</i>	205	47,80%
<i>EditorPullRequest</i>	<i>AuthorIssue</i>	98	45,91%
<i>EditorPullRequest</i>	<i>AuthorPullRequest, AuthorIssue</i>	98	45,91%
<i>AuthorPullRequest, EditorPullRequest</i>	<i>AuthorIssue</i>	98	45,91%
<i>Starred, AuthorIssue</i>	<i>EditorIssue</i>	146	42,46%
<i>Reactions</i>	<i>Starred</i>	575	42,26%

Analisando a tabela vemos que muitas atividades que se relacionam entre si, envolvem atividades que dependem uma das outras como *EditorPullRequest* e *AuthorPull-*

Request, um fator interessante é que os usuários que realizam essas atividades mesmo em conjunto com outras acabam também por realizar outras atividades, indicando uma maior participação do usuário no repositório, como é possível perceber nos primeiros elementos da tabela.

Temos também a tendência que usuários, que realizam atividades que envolvem maior complexidade como *AuthorPullRequest*, *AuthorIssue*, acabam por ter uma chance maior de se envolver em diversas atividades, e que usuários que realizam atividades mais simples como *Starred* e *Reactions*, geralmente não se envolvem com outras atividades.

Um indicativo que o comportamento que os usuários apresentam maiores interações sociais vão contribuir de forma mais ativa no repositório e os usuários que simplesmente acompanham o repositório, tendem a não se envolver ou contribuir diretamente, sendo eles sua grande maioria.

6. Discussões

Utilizando das análises realizadas nas seções anteriores, foi possível verificar que repositórios ativos (que apresentam um grande número de usuários que contribuem com atividades) apresentam uma tendência de crescimento, esse crescimento pode ser caracterizado relativamente de forma constante e linear, a depender muito das interações dos usuários que vão ocorrer conforme a popularidade do repositório, e nos eventos realizados dentro desse repositório conforme Zhang *et al.* (2014) havia mencionado em seus estudos.

Esses comportamentos dos usuários vão apresentar algumas variações, como foi possível verificar na Seção 5.3 (Estudo Exploratório) em Identificação dos Padrões, a tendência é que os usuários no geral realizem atividades de menor complexidade como *Starred* ou *Reactions*, tendo um menor potencial para o desenvolvimento do crescimento do repositório, outra parcela dos usuários signficante possui um determinado padrão que envolve realizar atividades de *AuthorIssue* no repositório e somente uma minoria dos usuários realiza atividades mais complexas como *AuthorPullRequest*, e esses usuários acabam por ter a tendência também de realizar diversas outras atividades, se encaixando no perfil do *Programador Socializador*, contribuindo para a popularidade do repositório como havia sido apontado por Costa e Ponciano *et al.* (2018) e por Onoue *et al.* (2013), sendo que essas variações na realizam das atividades permitem essas grande variedade de perfis de usuários que existem nos repositórios.

Aprofundando um pouco mais nos comportamentos dos usuários, percebemos que quando comparamos esses resultados com os obtidos por Costa e Ponciano *et al.* (2018), também conseguimos caracterizar os resultados com os perfis apresentados pelo estudo desenvolvido por Costa e Ponciano, indicando que as tendências e comportamentos que foram obtidas nos resultados apresentados, vão seguir determinada tendência sobre os repositórios.

Esses comportamentos analisados, levam a determinadas implicações, sobre principalmente o repositório, como a maioria dos usuários somente realiza atividades simples que contribuem para o crescimento do repositório, mas não contribuem com atividades como o envio de *pullrequests*, deixando essa atividade específica para um número bem menor de usuários com relação à quantidade total de usuários que interagiram com o

repositório, teremos um risco no qual se esses usuários que contribuem deixarem de contribuir o crescimento do repositório pode cair e ficar estagnado.

Além disso, temos também que a popularidade do repositório afeta bastante como a comunidade que está em torno desse repositório, interage com o mesmo, e como consequência isso afeta as contribuições em atividades realizadas pelos usuários dentro do repositório, com os resultados obtidos percebemos que uma pequena parcela de usuários vai realizar diversas contribuições e obter esses usuários é fundamental para ser possível aumentar o crescimento do repositório.

Todas essas análises foram feitas sobre uma vasta gama de dados, que sofre muitas alterações com o passar do tempo podendo resultados diferentes, mas com uma tendência semelhante, existe determinada dificuldade em obter os padrões principalmente devido vasta maioria dos usuários não realizarem múltiplas atividades, além disso, nos dados existem muitas atividades que tem potencial de serem exploradas para gerar mais padrões, porém geram mais complexidade no estudo como apontado por AlMarzouq *et al.* (2020).

Todo esse estudo sobre os padrões de comportamento dos usuários, se baseiam no funcionamento atual do GitHub que também sofre mudanças e atualizações com o decorrer do tempo, abrindo novas atividades e como consequência abrindo possibilidade de novas metodologias e novos estudos, mas é algo que deve ser filtrado como realizado por Consentino *et al.* (2017).

7. Conclusões

Foi apresentado neste estudo as tendências e os comportamentos dos usuários que interagem com a plataforma do GitHub, especificamente com um repositório. Em uma primeira abordagem, foi padronizado os dados e iniciou-se um processo de análises e discussões sobre os dados extraídos das atividades sobre essa plataforma, buscou-se criar uma linha temporal ilustrando a evolução das atividades e a linha temporal da evolução as atividades de um usuário, para mostrar o crescimento e a evolução no repositório. Posteriormente foi realizado uma verificação usando o algoritmo *FPGrowth* com o intuito de verificar os comportamentos dos usuários, no qual foi possível identificar que uma grande parcela desses usuários somente realizou atividades que exigem pouco esforço e praticamente nenhuma interação social sendo ela a atividade de *Starred*, e que uma parcela intermediária realizou atividades de *Reaction* e *AuthorIssue*, somente poucos usuários vão realizar atividades de maior complexidade como a atividade de *AuthorPullRequest* e essa pequena parcela de usuários são os usuários mais ativos e que mais realizam contribuições para o repositório, sendo chamados de *Programador Socializador*, por Costa e Ponciano *et al.* (2018). Com relação a tendências temos que usuários que realizam atividades mais complexas e que envolvem maior interação social, tendem a realizar as demais atividades. Com essa abordagem foi possível verificar essas tendências e comportamentos que os usuários apresentam, e esse estudo apresentado abre uma grande margem para novos estudos que possam incluir outras atividades e ampliar essa margem de tendências.

Referências

- [AlMarzouq et al. 2020] AlMarzouq, M., AlZaidan, A., and AlDallal, J. (2020). Mining github for research and education: challenges and opportunities. *International Journal of Web Information Systems*, 16(4):451–473.

- [Borges et al. 2016] Borges, H., Coelho, J., Carvalho, P., Fernandes, M., and Valente, M. T. (2016). Como pesquisadores usam o dataset ghtorrent. In *5th Brazilian Workshop on Software Visualization, Evolution and Maintenance (VEM)*, pages 1–8.
- [Cosentino et al. 2017] Cosentino, V., Izquierdo, J. L. C., and Cabot, J. (2017). A systematic mapping study of software development with github. *Ieee access*, 5:7173–7192.
- [Costa and Ponciano 2018] Costa, V. and Ponciano, L. (2018). Minerando padrões de interação de programadores com repositórios na plataforma github.
- [Dabbish et al. 2012] Dabbish, L., Stuart, C., Tsay, J., and Herbsleb, J. (2012). Social coding in github: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, pages 1277–1286.
- [Kalliamvakou et al. 2014] Kalliamvakou, E., Gousios, G., Blincoe, K., Singer, L., German, D. M., and Damian, D. (2014). The promises and perils of mining github. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, pages 92–101.
- [Onoue et al. 2013] Onoue, S., Hata, H., and Matsumoto, K.-i. (2013). A study of the characteristics of developers’ activities in github. In *2013 20th Asia-Pacific Software Engineering Conference (APSEC)*, volume 2, pages 7–12.
- [Sidhu et al. 2014] Sidhu, S., Meena, U. K., Nawani, A., Gupta, H., and Thakur, N. (2014). Fp growth algorithm implementation. *International Journal of Computer Applications*, 93(8).
- [Zhang et al. 2014] Zhang, L., Zou, Y., Xie, B., and Zhu, Z. (2014). Recommending relevant projects via user behaviour: an exploratory study on github. In *Proceedings of the 1st International Workshop on Crowd-based Software Development Methods and Technologies*, pages 25–30.