

UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"

FACULDADE DE CIÊNCIAS - CAMPUS BAURU

DEPARTAMENTO DE COMPUTAÇÃO

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ALBERTO AZEVEDO MARTINEZ

**ANÁLISE DO PERFIL DOS COLABORADORES DE PROJETOS DE
CÓDIGO ABERTO EM PLATAFORMAS DE REPOSITÓRIOS DE
SOFTWARE**

BAURU

2025

ALBERTO AZEVEDO MARTINEZ

**ANÁLISE DO PERFIL DOS COLABORADORES DE PROJETOS DE
CÓDIGO ABERTO EM PLATAFORMAS DE REPOSITÓRIOS DE
SOFTWARE**

Trabalho de Conclusão de Curso do Curso
de Ciência da Computação da Universidade
Estadual Paulista “Júlio de Mesquita Filho”,
Faculdade de Ciências, Campus Bauru.
Orientador: Prof. Dr. Higor Amario de Souza

M385a Martinez, Alberto Azevedo
Análise do perfil dos colaboradores de projetos de código aberto em plataformas de repositórios de software / Alberto Azevedo Martinez. -- Bauru, 2025
38 f. : il., tabs., mapas + 1 CD-ROM

Trabalho de conclusão de curso (Bacharelado - Ciência da Computação) - Universidade Estadual Paulista (UNESP), Faculdade de Ciências, Bauru
Orientador: Higor Amario de Souza

1. Mineração de dados. 2. Programadores de computador. 3. Python. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp.
Dados fornecidos pelo autor(a).

Alberto Azevedo Martinez

Análise do perfil dos colaboradores de projetos de código aberto em plataformas de repositórios de software

Trabalho de Conclusão de Curso do Curso de Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.

Banca Examinadora

Prof. Dr. Higor Amario de Souza

Orientador

Universidade de São Paulo

Escola Politécnica

Departamento de Engenharia de Computação e
Sistemas Digitais

Prof. Dra. Simone das Graças Domingues Prado

Universidade Estadual Paulista "Júlio de
Mesquita Filho"

Faculdade de Ciências

Departamento de Computação

Prof. Dr. André Luis Debiasso Rossi

Universidade Estadual Paulista "Júlio de
Mesquita Filho"

Faculdade de Ciências

Departamento de Computação

Bauru, 12 de Novembro de 2025.

Agradecimentos

Começo esse texto agradecendo principalmente ao Prof. Dr. Higor Amario de Souza e Prof. Dra. Simone das G. Domingues por me auxiliarem por todo esse ano em diversos momentos em que me foi necessário, retirando dúvidas, me auxiliando e trazendo novas possibilidades a serem analisadas para meu TCC. Agradeço a minha família que, apesar de não entenderem sobre o que eu estava pesquisando, se mostravam animados sempre que eu apresentava nem que fosse um pequeno progresso. Muitos amigos próximos merecem meus agradecimentos por, mesmo que por breves momentos, tornarem o processo do TCC em algo mais tolerável, obrigado a todo o pessoal do Caosnesp e outros grupos de amigos, obrigado Leonel, Rhayssa e Julia pelos novos vícios e pela companhia, Nicole pelas discussões e aos meus irmãos pela motivação de querer apresentar este trabalho para eles, amo vocês Henrique, Julia, Larissa e Laura. Por fim, aqueles que merecem os meus mais sinceros agradecimentos e amor, aqueles que me ajudaram durante todo o ano nos meus melhores e piores momentos pessoais e acadêmicos, muito obrigado Agatha, Akhan, Cape, Celso, Mavi e, não menos importante, Vinchenzzo, só sou capaz de entregar este TCC pois tive vocês ao meu lado, muito obrigado, eu amo muito todos vocês.

*Debí tirar más foto' de cuando te tuve. Debí darte más beso' y abrazo' las veces que pude.
Ojalá que los míó' nunca se muden. Y si hoy me emborracho, pues que me ayuden*
Bad Bunny

Resumo

Projetos de código aberto fazem parte importante do ecossistema de programas, com representantes performando entre opções viáveis no mercado, porém não possuem tantas pesquisas tendo enfoque específico em sua formação, ou sua comunidade. Essa pesquisa foi realizada com o objetivo de encontrar o perfil de colaboradores de projeto de código aberto e enriquecer as análises disponíveis para esses projetos, auxiliando assim os mantenedores desses projetos a tomarem melhores decisões e ações pelo bem da comunidade. Utiliza-se mineração de repositórios para conseguir os dados dos colaboradores de um grupo de projetos específico, sendo feito tratamento e diferentes análises gráficas dos dados posteriormente. A ferramenta desenvolvida durante este trabalho permite analisar a quantidade de contribuições feitas por país, a porcentagem de contas em atividade, assim como suas idades e contribuições feitas por idade de conta, possibilitando também uma análise visual da distribuição dos colaboradores. Desta forma, a ferramenta desenvolvida consegue auxiliar na criação de perfis relacionados aos colaboradores de projetos de código aberto.

Palavras-chave: Mineração. Dados. Programadores. Python.

Abstract

Open-source projects are an important part of the software ecosystem, with representatives performing among viable options in the market, but there is not much research specifically focused on their formation or their community. This research was conducted with the objective of finding the profile of contributors to open-source projects and enriching the analyses available for these projects, thus helping the maintainers of these projects to make better decisions and actions for the good of the community. Repository mining is used to obtain data on contributors from a specific group of projects, followed by data processing and different graphical analyses. The tool developed during this work allows for the analysis of the number of contributions made by country, the percentage of active accounts, as well as their ages and contributions made by account age, also enabling a visual analysis of the distribution of contributors. In this way, the developed tool can assist in the creation of profiles related to contributors to open-source projects.

Keywords: Mining. Data. Programmers. Python.

Lista de figuras

Figura 1 – Código responsável por fazer o tratamento das datas dos dados coletados. . . .	20
Figura 2 – Imagem do estado final do mapa de calor.	20
Figura 3 – Imagem do código funcionando com o <i>IPyWidgets</i>	22
Figura 4 – Imagem do mapa de calor geral.	24
Figura 5 – Imagem do mapa de calor do projeto Tasmota.	24
Figura 6 – Imagem do mapa de calor do projeto vscode.	25
Figura 7 – Contribuições gerais feitas por cada país.	26
Figura 8 – Contribuições para o projeto Tasmota feitas por cada país.	27
Figura 9 – Contribuições para o projeto vscode feitas por cada país.	27
Figura 10 – Porcentagem geral de contas ainda em atividade.	28
Figura 11 – Porcentagem do projeto Tasmota de contas ainda em atividade.	29
Figura 12 – Porcentagem do projeto vscode de contas ainda em atividade.	29
Figura 13 – Gráfico geral das idades das contas.	30
Figura 14 – Gráfico do projeto Tasmota sobre as idades das contas.	31
Figura 15 – Gráfico do projeto vscode sobre as idades das contas.	31
Figura 16 – Gráfico geral das contribuições por idades das contas.	32
Figura 17 – Gráfico do projeto Tasmota sobre as contribuições por idades das contas.	32
Figura 18 – Gráfico do projeto vscode sobre as contribuições por idades das contas.	33

Lista de quadros

Quadro 1 – Representação de parte do arquivo projetos.csv.	16
Quadro 2 – Representação de parte do arquivo colaboradores.csv.	17
Quadro 3 – Representação de parte do arquivo dados.csv.	18

Sumário

1	INTRODUÇÃO	11
2	FUNDAMENTOS	12
2.1	Ciência de Dados	12
2.2	Projetos de Código Aberto	13
2.2.1	Visão Geral	13
2.2.2	Licenças	14
3	METODOLOGIA	15
3.1	Coleta	15
3.2	Tratamento	18
3.3	Mapa	20
3.4	Gráficos	21
4	RESULTADOS E DISCUSSÃO	23
4.1	Mapa de calor	23
4.2	Gráficos	25
4.2.1	Contribuições feitas por cada país	25
4.2.2	Porcentagem de perfis abandonados	28
4.2.3	Idades das contas	29
4.2.4	Contribuições por idade das contas	30
5	CONCLUSÃO	34
	REFERÊNCIAS	36

1 Introdução

Projetos de código aberto são aqueles em que o código fonte é acessível de forma livre para toda a comunidade de usuários e desenvolvedores (FUGGETTA., 2003). Normalmente advindos de esforços colaborativos de colaboradores em torno do mundo (CORBLI, 2014). Outro aspecto importante de projetos de código aberto são suas licenças, como a licença pública geral ou a licença artista, que diferenciam sua distribuição (WU; LIN, 2001). São exemplos desse tipo de projeto, programas como o Linux, o Blender, o VS Code ou o Android, todos *softwares* de código aberto que se mantém na atualidade.

Apesar de sua variedade e importância, dificilmente foram encontrados trabalhos que tenham como foco apenas os colaboradores específicos deste tipo de projeto. Plataformas de repositório de *softwares* são pesquisadas e têm seus usuários analisados, mas não possuem um recorte específico para a comunidade (BLINCOE et al., 2016). Outras pesquisas se focam em parte da comunidade, mas não possuem uma visão mais ampla da mesma (QIU et al., 2023). Desta forma, havendo uma ausência de pesquisas e ferramentas que esboçem o perfil de colaboradores de código aberto.

Possuir uma base de colaboradores saudável é essencial para a manutenção de projetos de código aberto, sendo necessário atrair, treinar e manter os mesmos (GUIZANI et al., 2022). Esse tipo de projeto pode perder 80% dos seus novos colaboradores, ou possuir majoritariamente colaboradores que contribuem apenas uma vez (GUIZANI et al., 2022). Desta forma, se os mantenedores de projetos de código aberto não souberem o perfil de sua comunidade, podem tomar decisões que a prejudiquem como um todo.

Ciência de dados, assim como suas ferramentas, são a base para o desenvolvimento do trabalho realizado, visando auxiliar as comunidades de código aberto. O uso da mineração de repositórios possibilita a extração de diferentes tipos de dados valiosos que podem estar presentes no repositórios minerado (HASSAN, 2008). Atualmente, graças a plataformas de repositórios de software, como o GitHub, o processo de mineração se torna mais facilitado devido ao grande número de usuários que o site possui e as ferramentas que disponibiliza (KALLIAMVAKOU et al., 2014).

Considerando o que foi abordado, este trabalho tem como principal objetivo desenvolver uma ferramenta capaz de auxiliar os mantenedores de projeto de código aberto a realizar diferentes tipos de análises referentes a suas próprias comunidades. Possibilitando de forma simples que, utilizando a ferramenta, eles consigam criar mapas de calor, gráficos referentes a contribuições por país dos colaboradores, gráficos de porcentagem de colaboradores ainda ativos, gráficos referente a idade de suas contas e suas contribuições.

2 Fundamentos

Neste capítulo, será feita a definição dos principais temas deste trabalho, especificando a situação atual e conceitos por trás da ciência de dados, bem como seus usos e técnicas utilizadas. Posteriormente será definido outro conceito central: os projetos de código aberto, repositórios de softwares e os projetos selecionados para serem alvos deste trabalho.

2.1 Ciência de Dados

Ciência de Dados é definida como o estudo científico da criação, validação e transformação de dados para criar algo com significado, sendo ela a base de toda a pesquisa realizada ([ASSOCIATION, 2025](#)). Dentre as diversas opções que a ciência de dados nos possibilita escolher, a principal técnica utilizada para este trabalho é a mineração de repositórios, sendo através da mesma que foi obtido os dados a serem usados nesta pesquisa.

A mineração de repositórios de *software* é um campo dedicado a analisar e cruzar ligações de dados presentes em repositórios de código e desta forma encontrar informações interessantes sobre os sistemas e projetos ([HASSAN, 2008](#)). Esse campo busca transformar repositórios estáticos em ferramentas ativas que auxiliem no processo de tomada de decisões em projetos de *software* atuais ([HASSAN, 2008](#)). Exemplos de repositórios são os históricos, os de tempo de execução e de código, cada um contendo tipos de dados diferentes que podem gerar diferentes análises como o histórico de um erro, identificar anomalias de execução ou identificar padrões de uso de biblioteca ([HASSAN, 2008](#)).

No começo da história do campo de mineração de repositórios eram encontrados uma série de problemas, sendo alguns deles o fato de haver um acesso limitado a repositórios de *software*, geralmente advindos de projetos ou sistemas acadêmicos mas não possuindo uma riqueza de dados ([HASSAN, 2008](#)). Outro dos problemas era a complexidade para conseguir extrair os dados dos repositórios, visto que até então eles não eram construídos tendo a mineração de dados em mente, tornando o processo mais complicado([HASSAN, 2008](#)).

Atualmente tais problemas foram diminuídos graças a plataformas de repositório de código, sendo o GitHub a que será abordada neste trabalho, entre outras coisas ele possui uma gama de ferramentas que auxiliam para a mineração de repositórios ([KALLIAMVAKOU et al., 2014](#)). Sua interface de programação de aplicações de transferência de estado representacional (API REST) será responsável por transmitir de forma fácil e pratica todas as informações a serem coletadas.

2.2 Projetos de Código Aberto

Sendo projetos de código aberto uma das bases deste trabalho, nesta seção serão abordados dois tópicos diferentes acerca deste assunto. O primeiro tópico é uma visão geral da comunidade de código aberto como um todo, como estudos e sua importância. O segundo tópico a ser abordado é referente aos diferentes tipos de licenças que este tipo de código pode possuir, sendo necessário para ter uma visão mais objetiva do tipo de projeto trabalhado.

2.2.1 Visão Geral

Segundo Corbly (2014), é possível definir código aberto como o tipo de programa que o detentor dos direitos possibilita a terceiros o direito de estudar, modificar ou distribuir sem maiores problemas. Ele ainda define alguns outros critérios para dizer se um programa é de código aberto ou não, entre eles os dois que são mais interessantes para esse trabalho envolvem o fato de que eles têm seu código base disponível para que programadores possam acessar e, caso queiram, alterar (CORBLY, 2014). São exemplos desse tipo de projeto, programas como o Linux, o Blender ou o VS Code, que ainda se fazem presentes no mercado, mesmo com outras opções com códigos proprietários, o que demonstra que iniciativas de código aberto tem sido cada vez mais usadas sendo ferramentas como o DeepSeek, Android e bibliotecas de linguagens de programação, bons exemplos de projetos de código aberto na atualidade.

Ao procurar trabalhos focando nos usuários de repositórios de código que contribuem em projetos de código aberto, foi constatada uma ausência de materiais sobre esse tópico em específico. Existem diversos trabalhos que avaliam o perfil de colaboradores, mas não atrelados a cada projeto específico (BLINCOE et al., 2016). Além disso, esses trabalhos costumam conferir a influência ou o comportamento dos usuários e seus projetos na plataforma (KALLIAMVAKOU et al., 2016). Outros trabalhos que também poderiam ser citados são aqueles que possuem como foco um recorte em específico da comunidade, como a porcentagem de colaboradores mulheres e não-binários (QIU et al., 2023).

A vontade de aprender se mostra como um dos fatores importantes para contribuir em projetos de código aberto, desta forma, é necessário ter conhecimento do perfil geral dos colaboradores, para além de visões mais específicas (GEROSA et al., 2021). Considerando o que foi definido até então, conhecer o estado dessas comunidades é essencial para o seu desenvolvimento e manutenção, pois um projeto deste tipo necessita de uma base saudável de contribuidores, além de conseguir atrair, treinar e manter os mesmos (GUIZANI et al., 2022).

No entanto, como evidenciado anteriormente, há uma ausência de pesquisas, tanto recentes quanto mais antigas, neste tópico, demonstrando a necessidade de trabalhos neste âmbito. Esses fatos, somados à importância de projetos de código aberto, motivaram este trabalho que visa criar uma ferramenta que ajude a descrever o perfil de seus desenvolvedores para uma melhor manutenção dessas comunidades.

2.2.2 Licenças

O conceito de código aberto foi criado em 1984 por Richard Stallman, originalmente como Fundação de Código Livre, tendo como ideia principal que o usuário final pudesse utilizar o programa livremente, permitindo que o mesmo execute, modifique e distribua o programa de forma a beneficiar a comunidade (WU; LIN, 2001). Posteriormente o termo de código aberto é criado tendo como foco uma tentativa de expressar o mesmo conceito, mas de forma mais prudente e palatável para a comunidade, dessa forma impedindo a ideia de que códigos livres necessariamente seriam gratuitos (FUGGETTA., 2003).

Portanto, código aberto pode ser considerado um termo abrangente que engloba diferentes tipos de programas, sendo sua licença específica um fator diferenciador entre projetos. Algumas das licenças requerem que códigos derivados sigam as mesmas regras nas quais os programas originais foram adquiridos, enquanto outras permitem que as modificações sejam consideradas de propriedade do desenvolvedor que as implementou (COLAZO; FANG, 2009).

A maioria das licenças de código aberto podem ser categorizadas em três tipos diferentes, sendo elas as licenças de *copyleft* forte, *copyleft* fraco e as sem *copyleft* (SEN; SUBRAMANIAM; NELSON, 2009). As licenças de *copyleft* forte exigem que qualquer trabalho derivado siga a mesma licença do programa original (SEN; SUBRAMANIAM; NELSON, 2009). Por sua vez, projetos que sigam licenças de *copyleft* fraco permitem que derivações tenham suas próprias licenças sob certas restrições (SEN; SUBRAMANIAM; NELSON, 2009). Por fim, aquelas sem *copyleft* permitem que trabalhos derivados não precisem seguir nenhum tipo de licença que o programa original segue (SEN; SUBRAMANIAM; NELSON, 2009).

3 Metodologia

Neste capítulo será descrito com maiores detalhes todo o processo envolvendo a coleta, tratamento e manipulação dos dados, dando enfoque nas bibliotecas usadas, problemas encontrados, soluções para o mesmo, assim como trechos do código e seus respectivos usos.

3.1 Coleta

Levando as definições de um código aberto como base para este trabalho, foram separados trinta e um projetos de código aberto diferentes, com diversos tamanhos e funcionalidades. O objetivo principal é possuir uma grande variedade de projetos para serem trabalhados, desta forma realizando análises generalizadas utilizando-se da ferramenta a ser desenvolvida e almejando que ela consiga lidar com diferentes comunidades.

Os projetos selecionados podem ser separados em três tipos diferentes de conjuntos: aqueles que possuem menos de mil colaboradores, aqueles que possuem entre mil e dois mil colaboradores, e por fim aqueles que possuem mais de dois mil. O propósito desta separação está relacionado com a necessidade de uma maior variedade de projetos e evitar que a amostra trabalhada apresentasse uma tendência a tipos de projetos específicos ao invés de uma abordagem mais abrangente.

A começar pelos projetos com menos de mil colaboradores, os escolhidos foram: **Tasmota**, um *firmware* alternativo para microcontroladores; **localstack**, um *framework* de desenvolvimento em nuvem; **youtube-music**, *framework* com *plugins* customizáveis para o YouTube Music; **n8n**, plataforma de automação de fluxo de trabalho com recursos IA nativos; **discord.js**, módulo de Node.js que permite uma fácil interação com a Discord API; **python-telegram-bot**, biblioteca em Python que oferece uma interface para a API do Bot do Telegram; **ccxt**, biblioteca para negociação de criptomoedas e e-commerce; **starship**, *prompt* minimalista e personalizável para qualquer *shell*; **nvm**, gerenciador de versão para node.js; **qBittorrent**, cliente *bittorrent* programado em C++; e **TypeScript**, é uma linguagem de programação de código aberto desenvolvida pela Microsoft.

Os projetos selecionados por terem entre mil e dois mil colaboradores foram: **flutter**, uma SDK do Google para criar uma experiência rápida e bonita para usuários; **serverless**, um *framework* para gerenciamento de nuvem; **flink**, *framework* para transmissão e processamento de dados; **babel**, compilador de JavaScript; **electron**, um *framework* para construção de aplicativos multi-plataforma; **neovim**, um editor de texto baseado em Vim; **bootstrap**, uma ferramenta para desenvolvimento *web*; **supabase**, plataforma de desenvolvimento de banco de dados; **docusaurus**, projeto para construção, desenvolvimento e manutenção de sites mais

facilmente; e **matplotlib**, biblioteca Python para criar estatísticas, animações e visualizações.

Por fim, estes são os projetos selecionados por possuírem mais de dois mil contribuidores: **react-native**, *framework* para construir aplicações nativas usando React; **vscode**, um editor de código; **tensorflow**, uma biblioteca de *software* de aprendizagem de máquina; **kubernetes**, uma plataforma de organização de contêineres; **django**, uma *framework web* para Python; **scikit-learn**, biblioteca de aprendizagem de máquina em Python; **angular**, *framework* para a criação de aplicativos web; **transformers**, modelo de aprendizagem de máquina; **ansible**, uma plataforma para gerenciamento de servidores; e **pandas**, biblioteca Python para análise e manipulação de dados.

Uma vez que os projetos estão selecionados, foi necessário criar um arquivo contendo o nome do projeto no GitHub e seus respectivos donos, como é possível visualizar no Quadro 1, esse arquivo sendo a base de toda a coleta. Em versões iniciais do arquivo projetos.csv, o primeiro registro, que possui a coluna “Dono” e “Projeto” marcados ambos como “Geral” e que pode ser visto no Quadro 1, não foi incluída devido não possuir necessidade em uma período inicial do trabalho, sendo adicionado posteriormente para ser utilizado em outros momentos do código.

Quadro 1 – Representação de parte do arquivo projetos.csv.

	Dono	Projeto
1	Geral	Geral
2	arendst	Tasmota
3	localstack	localstack
4	th-ch	youtube-music
5	n8n-io	n8n
6	discordjs	discord.js
7	python-telegram-bot	python-telegram-bot
8	ccxt	ccxt
9	starship	starship
10	nvm-sh	nvm
11	qbittorrent	qBittorrent
12	microsoft	TypeScript
13	flutter	flutter
14	serverless	serverless
15	apache	flink

Fonte: Elaborado pelo autor.

A principal ferramenta utilizada para a coleta foi a biblioteca *request* do Python, que permite fazer requisições para determinados sites. Ela foi primeiramente utilizada para a coleta de todos os colaboradores que eram possíveis ter acesso. As primeiras informações coletadas foram os nomes de colaboradores, o número de contribuições e o seu projeto de origem, gerando dessa forma o arquivo bruto usuarios.csv que pode ser visto no Quadro 2.

Quadro 2 – Representação de parte do arquivo colaboradores.csv.

	login	contributions	id_projeto
1	arendst	8879	Tasmota
2	s-hadinger	3214	Tasmota
3	Jason2866	2139	Tasmota
4	ascillato	886	Tasmota
5	gemu2015	592	Tasmota
6	Staars	306	Tasmota
7	stefanbode	302	Tasmota
8	curzon01	206	Tasmota
9	barbudor	199	Tasmota
10	andrethomas	197	Tasmota
11	pcdiem	122	Tasmota
12	ascillato2	114	Tasmota
13	bovirus	113	Tasmota
14	hallard	99	Tasmota
15	ajithvasudevan	84	Tasmota

Fonte: Elaborado pelo autor.

Já nessa primeira coleta alguns problemas se destacaram, como a paginação das requisições e o limite de requisições por hora. O GitHub possui um limite de colaboradores que ele exibe, dessa forma, mesmo tendo acesso a quinhentos colaboradores, numa primeira requisição apenas os cem primeiros são apresentados, esse problema foi superado se utilizando de múltiplas requisições, apenas mudando qual página deveria ser consultada. O segundo problema, por sua vez, se dá como um mecanismo de defesa do GitHub para evitar ataques distribuídos de negação de serviço (DDoS), limitando para apenas sessenta requisições não identificadas por hora, porém se utilizando de um token de acesso associado a uma conta no site, é possível aumentar esse limite para quinze mil, solucionando o problema de atingir um limite de requisições.

Uma segunda coleta inicial envolvia informações sobre os donos e os projetos em si, apesar de ambos terem sido coletados com sucesso e sem maiores problemas, pelo fato de não possuírem uma grande quantidade de dados para se trabalhar, não foram utilizados posteriormente neste trabalho, apesar dos arquivos ainda poderem ser acessados em donos.csv e repositorios.csv.

Na parte final da coleta de dados foi utilizado o arquivo usuarios.csv como base, o arquivo gerado na primeira parte da coleta contendo nome, contribuições e projeto de cada usuário coletado. Ao todo foram registrados onze mil seiscentos e doze perfis, nos quais nessa parte da coleta tiveram os seus dados registrados individualmente.

Assim como no primeiro momento é feita a requisição para o site do GitHub, desta vez solicitando as informações do perfil, entre uma grande gama de dados disponíveis foram

gravadas a localização do perfil, a quantidade de repositórios públicos, quantidade de seguidores, quantidade de pessoas que o perfil segue, quando a conta foi criada e a última vez que a conta foi atualizada. Juntamente dessas informações também foi colocado num único arquivo os dados já previamente coletados, como nome de usuário, projeto de origem e quantidade de contribuições, desta forma dando origem ao arquivo usuarios.csv, que pode ser parcialmente visto no Quadro 3, sendo a versão bruta dos dados que posteriormente são trabalhados. Ainda nessa coleta os dados de dois colaboradores foram perdidos devido a erros que aconteceram durante o processo, ademais, nenhum outro problema se manifestou.

Quadro 3 – Representação de parte do arquivo dados.csv.

	login	id_projeto	location	public_repos	followers
1	arendst	Tasmota	Netherlands	30	1627
2	s-hadinger	Tasmota	NULL	26	39
3	Jason2866	Tasmota	Germany	198	375
4	ascillato	Tasmota	Canada	155	56
5	gemu2015	Tasmota	NULL	4	34
6	Staars	Tasmota	Germany	55	15
7	stefanbode	Tasmota	Coloogne, Germany	17	45
8	curzon01	Tasmota	Germany	30	12
9	barbudor	Tasmota	France, Parias south suburbs	42	24
10	andrethomas	Tasmota	In a 408km orbit...	55	11
11	pcdiem	Tasmota	IL, US	5	2
12	ascillato2	Tasmota	NULL	0	16
13	bovirus	Tasmota	Milano, Italy	47	99
14	hallard	Tasmota	Poitiers, France	116	472
15	ajithvasudevan	Tasmota	Bangalore	31	19

Fonte: Elaborado pelo autor.

Todas as informações coletadas do GitHub foram adquiridas no período entre os dias doze e dezessete de junho de 2025. Outro ponto que merece destaque é o fato da plataforma disponibilizar para consulta apenas os dados dos quinhentos colaboradores com mais contribuições de cada projeto. Isso significa que para projetos com até, aproximadamente, quinhentos colaboradores, as análises são para o projeto como um todo, enquanto para aqueles com mais, a análise se torna parcial e focada naqueles que mais contribuíram.

3.2 Tratamento

Para o tratamento dos dados, foi utilizada a API do MapBox, uma plataforma de sistema de posicionamento global (GPS) utilizada por empresas como BMW, Toyota e CNN. A escolha se deve ao fato do mesmo possuir um grande número de requisições mensais gratuitas para geocodificação, cem mil, sendo dez vezes a quantidade oferecida pela Google, dez mil, que também cobra cinco dólares para cada mil requisições acima do limite. A facilidade de uso

para geocodificação foi outro fator crucial para destacar essa API em comparação a outras semelhantes, como a OpenStreetMap, que, mesmo sendo gratuita, não apresentou facilidade de utilização, culminando na escolha do MapBox para a realização deste trabalho. Outra ferramenta importante é a biblioteca Pandas do Python que é utilizada para leitura e gravação dos dados, neste momento sendo também responsável por tratar as informações relacionadas às datas das contas.

Os dois maiores problemas que necessitavam de tratamento eram em relação às localizações dos colaboradores e as datas de criação e atualização das suas contas. O primeiro não estava padronizado, pois como era um campo de texto, os colaboradores poderiam escrever o que quisessem e da maneira que quisessem, resultando num campo com uma mesma localização escrita de diversas formas diferentes, sem uma localização real, uma piada ou apenas vazio, sendo necessário tratá-lo para que uma análise seja possível. Já o segundo, as datas, estavam padronizadas, mas todas as informações estavam aglutinadas, sendo necessário separá-las em campos específicos para uma maior facilidade de análise posterior.

O tratamento das localizações pode ser dividido em duas partes, a primeira é uma filtragem manual de textos que se destaquem como piadas, símbolos, possuindo formatações únicas ou localizações confusas, a segunda parte se utiliza da API REST do MapBox para coletar e padronizar as localizações, registrando o nome do país e as coordenadas geográficas exatas do ponto. A separação é necessária pois sem um tratamento prévio, quando uma localização inexistente é consultada na API do MapBox, a mesma pode apresentar erros, ou, em caso de localizações confusas, apontar erroneamente para algum lugar.

Algo a se destacar sobre o tratamento das localizações é o fato de que algumas delas que seriam válidas tiveram que ser alteradas ou desconsideradas. Para locais reais não especificados ou plausíveis, foram considerados como nulos, vale citar exemplos de localizações como “Oceania” ou “Antártida”, que foram consideradas como localizações nulas e alocadas as coordenadas zero e zero. A escolha de tal ponto se deve ao fato de que serviços de localização que apresentam erros durante seu funcionamento acabam alocando a posição não definida no ponto zero e zero, em um local comumente conhecido como Ilha Nula, para manter esse comportamento e possibilitar uma fácil visualização desses colaboradores, foi mantido o mesmo padrão (JUHÁSZ; MOONEY, 2022). Outros pontos que tiveram uma abordagem específica foram nomes populares que passaram para o nome oficial da cidade, estado ou país e a situação de pessoas que se localizaram como presentes na Europa, que possuíam um valor considerável e que usando a API seriam coladas em um ponto no meio da França, para evitar erros, as coordenadas das mesmas foram destinadas ao centro de Mônaco, um local que não possui qualquer outro usuário na amostra.

O tratamento das datas ocorreu de forma muito mais simples, pois com a biblioteca Pandas, foi necessário apenas ler os campos de criação e atualização em formato de datas e criar novos campos específicos de dia, mês e ano para cada um, o que pode ser observado na

Figura 1. Desta forma o arquivo dadosFinais2.csv é criado contendo não apenas as datas de forma conjunta, como também de forma separada, facilitando as análises.

Figura 1 – Código responsável por fazer o tratamento das datas dos dados coletados.

```
#####Tratamento das datas#####
df = pd.read_csv( filepath_or_buffer: 'CSVs/Dados Finais/dados.csv', sep=';')

df['Atualizado_em'] = pd.to_datetime(df['Atualizado_em'], format='%Y-%m-%d').dt.date

df['Atualizado_em_Ano'] = pd.to_datetime(df['Atualizado_em']).dt.year
df['Atualizado_em_Mes'] = pd.to_datetime(df['Atualizado_em']).dt.month
df['Atualizado_em_Dia'] = pd.to_datetime(df['Atualizado_em']).dt.day

df['Criado_em'] = pd.to_datetime(df['Criado_em'], format='%Y-%m-%d').dt.date

df['Criado_em_Ano'] = pd.to_datetime(df['Criado_em']).dt.year
df['Criado_em_Mes'] = pd.to_datetime(df['Criado_em']).dt.month
df['Criado_em_Dia'] = pd.to_datetime(df['Criado_em']).dt.day

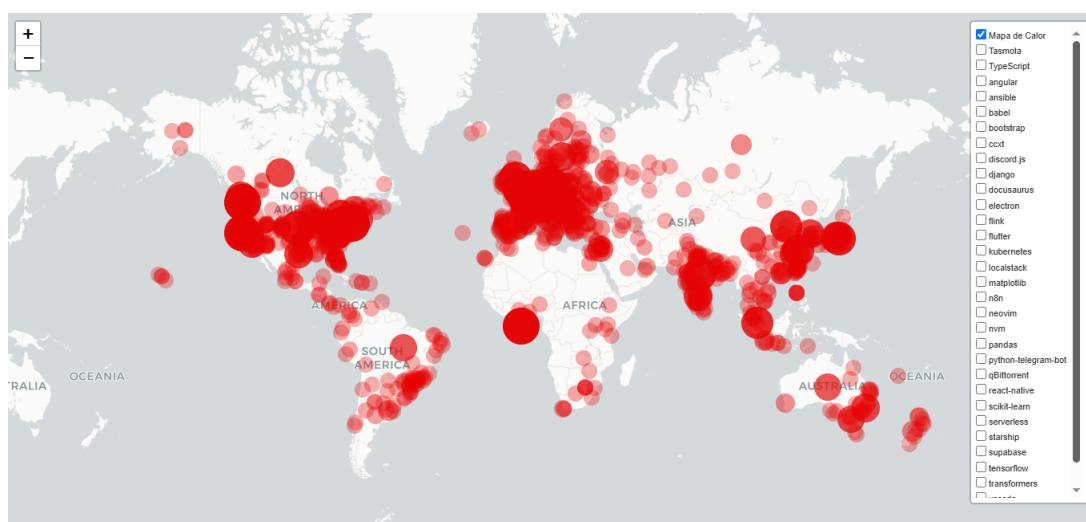
df.to_csv( path_or_buf: 'CSVs/Dados Finais/dadosFinais2.csv', sep=';', index=False)
```

Fonte: Elaborada pelo autor.

3.3 Mapa

Uma vez que os dados estão devidamente tratados, é possível formar os itens de análise que serão usados posteriormente, sendo o mapa a ferramenta mais interativa dentre eles, pois permite visualizar e comparar de forma gráfica as diferenças em questão de quantidades de colaboradores de cada projeto ou em quesito geral, se utilizando de diferentes cores, tamanhos e opacidades para cada quantidade de colaboradores em determinados pontos, na Figura 2 é possível ver o estado final de como ele ficou.

Figura 2 – Imagem do estado final do mapa de calor.



Fonte: Elaborada pelo autor.

A principal biblioteca utilizada para a criação do mapa de calor foi a *Folium* do Python por possuir uma série de diferentes mapas disponíveis, sendo possível criar e manipular muitos

tipos de formas e ainda ter diversas ferramentas que melhoram a experiência de quem estiver visualizando o mapa de calor.

Algumas funcionalidades utilizadas para melhorar a visualização do mapa de calor envolvem a escolha do mapa base por si só, que está com os nomes dos países em inglês para melhor entendimento, a redundância do mesmo, que possibilita navegar por ele de forma mais fluida sem ter que lidar com limites, e o uso de caixas de textos nos pontos que explicitam a quantidade exata de colaboradores em cada local.

Para a criação dos pontos, foi utilizada a função *CircleMarker*, essa que por sua vez desenha no mapa um círculo em coordenadas específicas. Seu tamanho e opacidade são definidos pela quantidade de colaboradores daquele ponto, possuindo um valor máximo e mínimo. As cores dependem de qual grupo o ponto faz parte, sendo possível um ponto estar em diferentes grupos, possuindo diferentes tamanhos, opacidades e cores. Outra possibilidade que o uso dessa função permite é que caso diversos pontos estejam muito próximos uns dos outros, suas opacidades e cores são somadas, dando um destaque maior para aquela região, mas caso o usuário se aproxime daquela área, os pontos vão se tornando mais precisos e separados.

3.4 Gráficos

Para cobrir as diferentes análises idealizadas para esse trabalho, foram utilizadas diferentes bibliotecas Python. A biblioteca *Matplotlib* foi usada para produzir a maioria dos gráficos. Outras bibliotecas usadas foram a *Plotly*, que possibilitou uma criação gráfica única, e a *IPyWidgets*, que tornou interativo o programa responsável pela geração de gráficos e análises, fazendo com que a consulta fosse feita de forma muito mais simples e rápida.

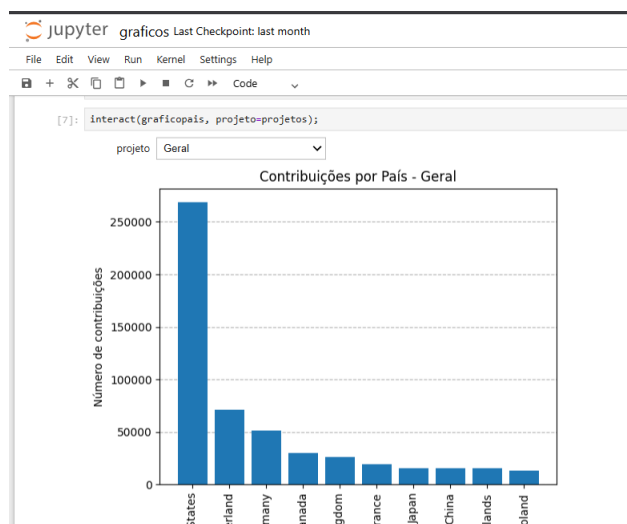
A biblioteca *matplotlib* foi responsável por três dos quatro gráficos criados para este trabalho, sendo utilizadas funções para montar gráficos de barras e gráficos de pizza. A maior dificuldade presente para se utilizar essa biblioteca foi em relação à organização das informações, juntando, somando ou separando dados específicos a depender do tema do gráfico e se o mesmo representa uma análise global do que foi coletado ou de algum projeto em especial. Desta forma toda a construção gráfica acabou sendo muito mais uma parte que se utilizava da biblioteca Pandas e de código puro do que especificamente de funcionalidades do *matplotlib*, uma vez que as informações estavam organizadas, elas eram facilmente colocadas em gráficos.

Para a última análise, a responsável pelas contribuições por idade das contas, foi utilizada a biblioteca *Plotly*, graças ao seu acesso a um tipo de gráfico chamado *treemap* que transmite as informações de forma mais clara e simples. Apesar de ser uma biblioteca diferente, os problemas citados anteriormente persistem, fazendo com que a estrutura de parte fosse bem semelhante à anterior.

A responsável por tornar o código em algo mais interativo e de fácil utilização foi

a biblioteca *IPyWidgets*. Utilizando-se da função *interact* da mesma e mudando de simples arquivos Python para um arquivo *notebook* foi possível realizar análises para todos os projetos ou filtrar somente para um projeto específico. Devido a mudança de abordagem, parte do código precisou ser adaptado para se adequar às novas demandas, sendo sua principal mudança a necessidade de gerar apenas um gráfico específico, ao invés de gerar todos. Na Figura 3 é possível observar a versão final do código em funcionamento.

Figura 3 – Imagem do código funcionando com o *IPyWidgets*.



Fonte: Elaborada pelo autor.

4 Resultados e Discussão

Para este capítulo será feita a análise de tudo que foi gerado pela ferramenta durante a elaboração deste trabalho, analisando os gráficos, mapas e dados conseguidos durante este estudo. Para uma base comparativa e para mostrar uma visão global dos dados coletados, serão utilizados como exemplo os gráficos gerais do projeto Tasmota, que possui menos de quinhentos contribuidores. Já o projeto vscode possui mais de dois mil colaboradores no GitHub e nos dará uma visão mais específica dos seus principais colaboradores.

4.1 Mapa de calor

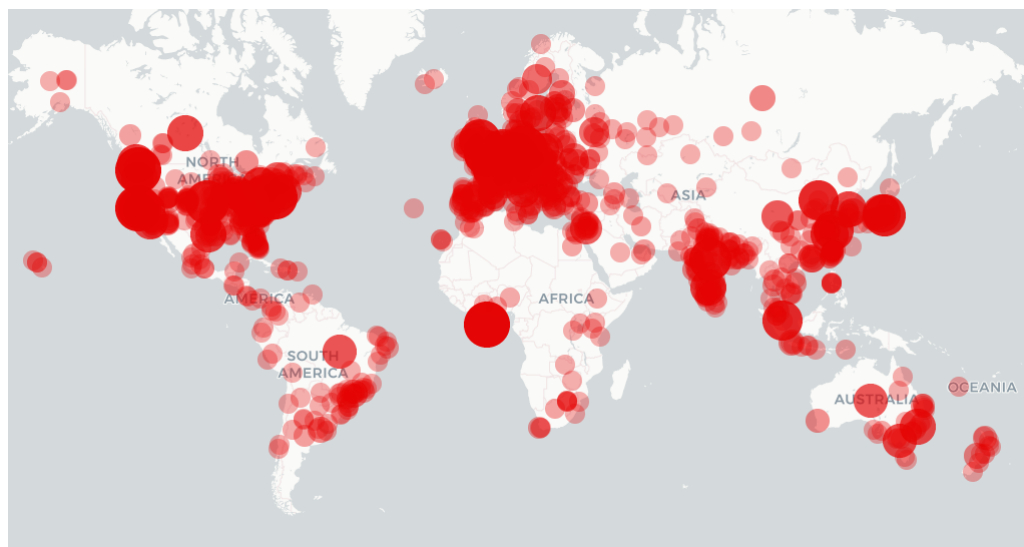
A análise através do mapeamento possibilita vermos de forma direta os locais nos quais os colaboradores estão concentrados, comparando entre mapas os locais com maior quantidade de pontos e o que os mesmos dizem.

Ao começar pelo mapa de geral, que está sendo ilustrado na Figura 4, é perceptível uma grande quantidade de pontos localizados na latitude 0 e longitude 0, pontos espalhados por todos os Estados Unidos da América (EUA) e Europa, além de vários pontos na Ásia em países como China, Índia e Japão. Esses pontos acompanham características interessantes, uma delas é que eles são países muito populosos, países como EUA, China e Índia constam entre os países com a maior quantidade de pessoas ([UNIDAS, 2024](#)). Já locais que não possuem uma grande população, mas ainda contam com uma grande quantidade de pontos se devem ao fato de possuírem uma grande taxa de escolaridade. Dessa forma, diversos países da Europa, a Austrália, o Japão e os EUA se destacam ([ECONÔMICO, 2023](#)).

Ao analisar os pontos referentes ao projeto Tasmota, que podem ser vistos na Figura 5, é possível notar que, assim como no mapa geral, há uma grande presença de colaboradores nulos e diversos pontos espalhados pelos EUA e pela Europa. Esse padrão acaba se repetindo novamente quando é analisado o mapa referente ao vscode, sendo possível perceber com a Figura 6 que locais como Seattle possuem alguns dos maiores pontos do mapa.

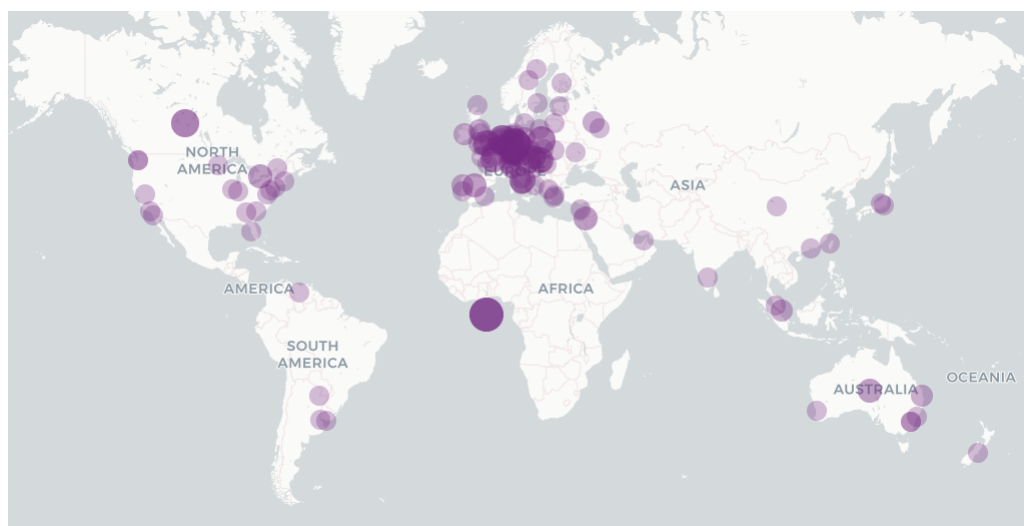
Após a análise dos mapas de calor é possível afirmar que, entre outras coisas, a maioria dos colaboradores opta por não definir seu país de origem, fazendo com que os pontos localizados na latitude 0 e longitude 0 se destaquem, sendo ao todo quatro mil, cento e sessenta colaboradores que não identificaram uma localização, tornando essa a moda da amostra coletada. Outra coisa que pode ser percebida é o fato que escolaridade parece ser um fator importante para definir a presença ou não de colaboradores em um determinado país, podendo afirmar isso vide que os EUA possuem pontos em Nova Iorque, São Francisco e Seattle com aproximadamente duzentos colaboradores cada, sendo algo que se repete em países da

Figura 4 – Imagem do mapa de calor geral.



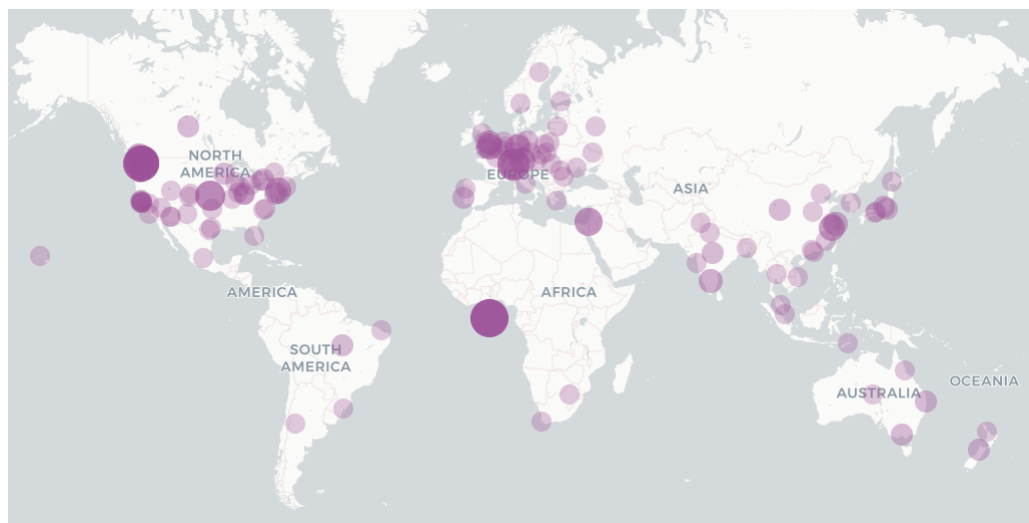
Fonte: Elaborada pelo autor.

Figura 5 – Imagem do mapa de calor do projeto Tasmota.



Fonte: Elaborada pelo autor.

Figura 6 – Imagem do mapa de calor do projeto vscode.



Fonte: Elaborada pelo autor.

Europa como Alemanha, França e Reino Unido que possuem pontos com uma alta densidade de colaboradores, além de diversos outros pontos menores espalhados por seus respectivos territórios. Por sua vez, países como a China ou a Índia, que possuem uma alta população, ainda se destacam, mas não possuem uma densidade que permita se assemelhar aos locais previamente citados. Vale ressaltar que a junção de uma grande população com uma alta escolaridade faz com que os EUA seja um dos países com maior quantidade de pontos de calor, senão, o maior.

4.2 Gráficos

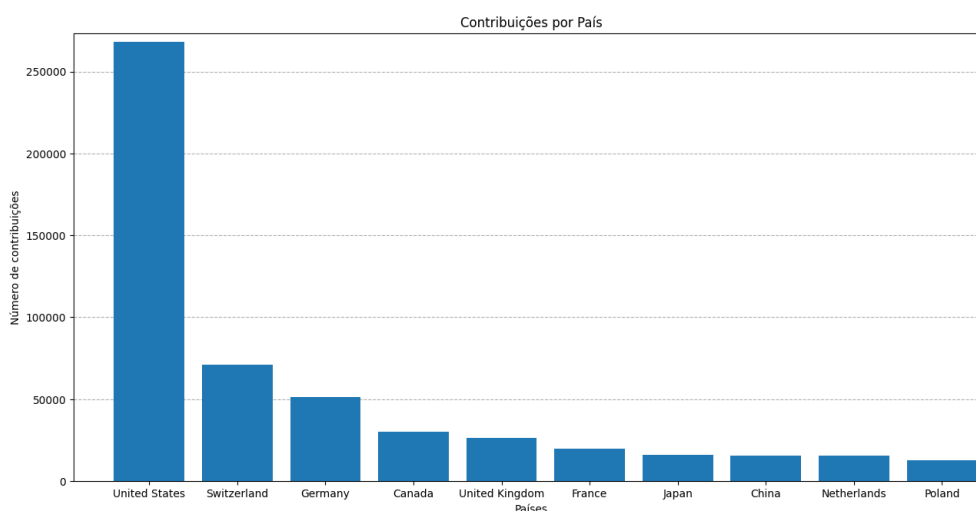
Diferentes análises gráficas foram realizadas, cada uma com um objetivo que ajuda a entender melhor o cenário dos projetos de código aberto, sendo essas análises das contribuições por país, porcentagem de perfis abandonados, idades das contas e colaboração por idade. Nesta seção será feita uma análise dos resultados obtidos.

4.2.1 Contribuições feitas por cada país

O objetivo desta análise é conferir os países com mais contribuições, independente do número de colaboradores de cada um, sendo possível perceber na Figura 7 que EUA, Alemanha e Reino Unido, que estão entre os países com maior quantidade de colaboradores, também se fazem presentes entre aqueles com mais contribuições, assim como Suíça e Canadá, que mesmo não possuindo uma grande quantidade de colaboradores, possuem muitas contribuições. Outros países que estão entre os maiores contribuidores também apresentam este padrão, evidenciando que o número de colaboradores não necessariamente se converte em contribuições. Vale destacar a necessidade de tirar as contribuições feitas por colaboradores que não especificaram sua

localização, uma vez que eles representam por volta de 35% dos colaboradores e o valor das suas contribuições destoa muito das demais, além do fato que sua presença neste gráfico não acrescentaria informações relevantes vide que não diz a distribuição de contribuição dos colaboradores.

Figura 7 – Contribuições gerais feitas por cada país.

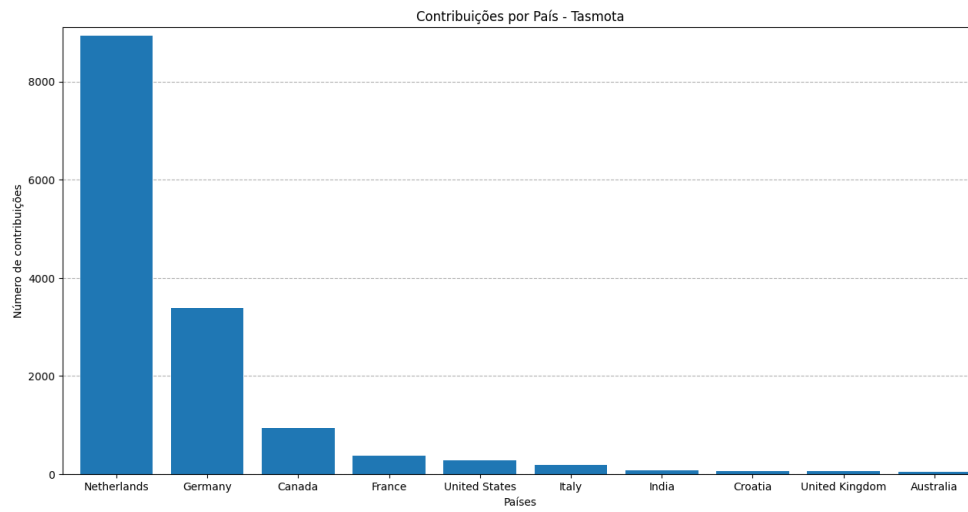


Fonte: Elaborada pelo autor.

Saindo de uma análise geral e aprofundando nas contribuições de cada país por projeto é possível ver na Figura 8 que o projeto Tasmota conta com a presença de países com as maiores contribuições globais, tais quais como Alemanha, Canadá, França. Há casos como os Países Baixos, que na classificação global está em nono, mas para o projeto Tasmota é o país com mais colaborações, sendo o local onde o criador do projeto, Theo Arends, está localizado, demonstrando uma possível relação entre o país com mais contribuições e o país do criador do projeto. É importante destacar que Croácia, Itália e Índia também estão presentes apesar de não serem os países com os maiores números de contribuições globais. Esse comportamento se repete quando é analisado o gráfico na Figura 9, que é referente ao projeto vscode, países como os EUA, Suíça, entre outros se destacam como grandes contribuintes, tal qual no gráfico geral, porém ainda há presença de países que não estão entre aqueles que mais contribuíram, como a Austrália e a Índia. Vale ressaltar que a ordem de grandeza dos dois projetos é bem destoante, com o gráfico referente ao Tasmota não chegando a dez mil contribuições, enquanto o do vscode tem um país com mais de sessenta mil.

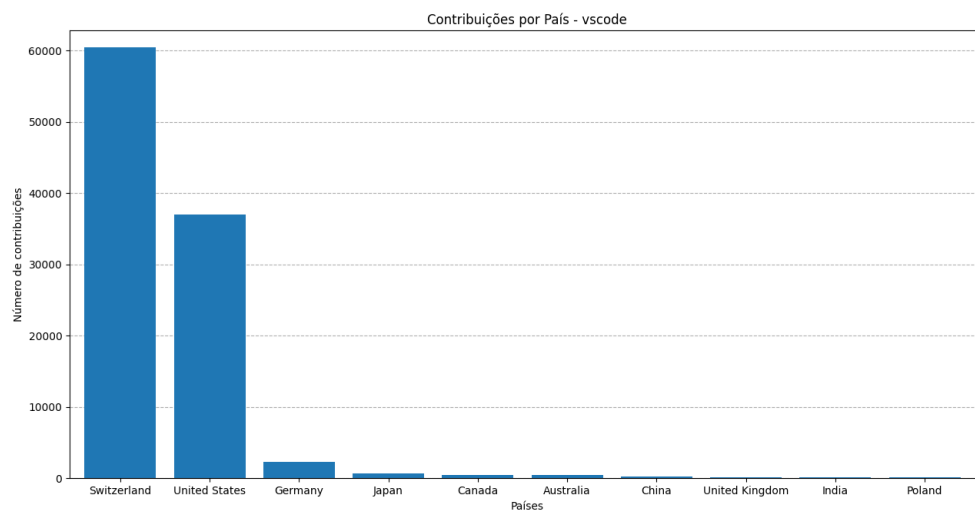
Analisando os gráficos como um todo, é possível afirmar que a quantidade de colaboradores e o número de contribuições possuem uma forte relação, vide que os países com mais colaboradores, como Alemanha, EUA e Reino Unido performam entre aqueles com maiores quantidades de contribuições, porém isso não é uma máxima, pois mesmo países que possuem relativamente menos colaboradores ainda conseguem se destacar com seus números de con-

Figura 8 – Contribuições para o projeto Tasmota feitas por cada país.



Fonte: Elaborada pelo autor.

Figura 9 – Contribuições para o projeto vscode feitas por cada país.



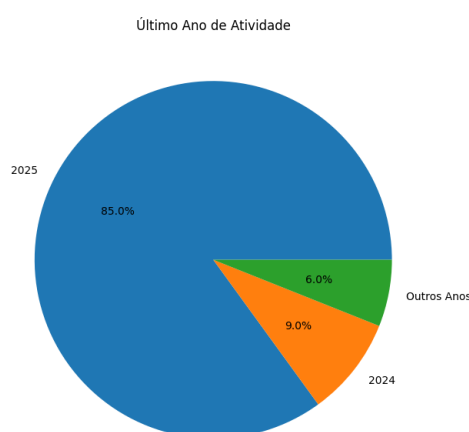
Fonte: Elaborada pelo autor.

tribuições, como é o caso do Canadá, Polônia e Suíça. Outra coisa que merece destaque é o comportamento de rápida queda no número de contribuições, onde independente do gráfico analisado, um país se destaca por possuir o maior número e os próximos já apresentam uma queda considerável na quantidade de contribuições, evidenciando que, mesmo sendo um projeto livre para qualquer um contribuir, a maioria das contribuições acabam sendo de uma mesma nacionalidade.

4.2.2 Porcentagem de perfis abandonados

Considerando os principais colaboradores de projetos de código aberto, é necessário analisar o quanto dos mesmo ainda estão em atividade, sendo possível ver na Figura 10 que, num cenário geral, 85% dos colaboradores ainda mantêm suas contas em atividade, apenas 9% não tiveram atividade desde 2024 até a data da coleta e os 6% restantes estão sem atualizações em seus perfis há pelo menos dois anos no mesmo cenário citado anteriormente.

Figura 10 – Porcentagem geral de contas ainda em atividade.

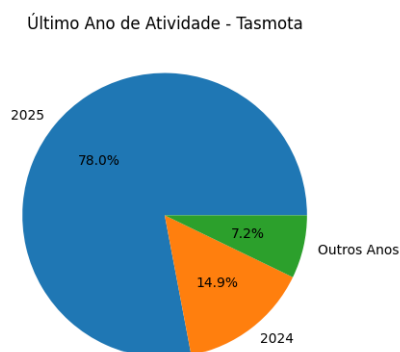


Fonte: Elaborada pelo autor.

Aprofundando a análise para as porcentagem específicas de projetos, é visível que o gráfico do projeto Tasmota e do vscode, que estão representados respectivamente pelas Figuras 11 e 12, também se assemelham com o gráfico geral, possuindo em média 82,7% de contas em atividade, 11,3% de colaboradores inativos a pelo menos um ano e 6% de colaboradores que abandonaram a pelo menos 2 anos, sendo porcentagens muito próximas, senão iguais às aferidas na análise global.

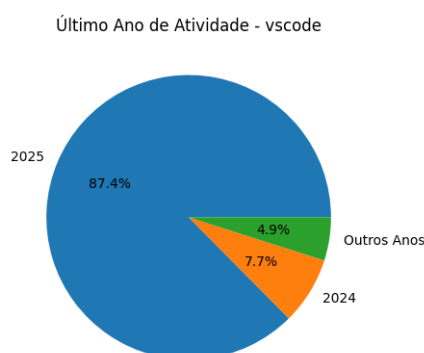
Levando em consideração os gráficos gerados, é possível dizer que, em geral, os colaboradores de projetos de código aberto tendem a permanecer em atividade, sendo uma maioria relevante, além de, diferentemente dos gráficos analisados anteriormente, o resultados

Figura 11 – Porcentagem do projeto Tasmota de contas ainda em atividade.



Fonte: Elaborada pelo autor.

Figura 12 – Porcentagem do projeto vscode de contas ainda em atividade.



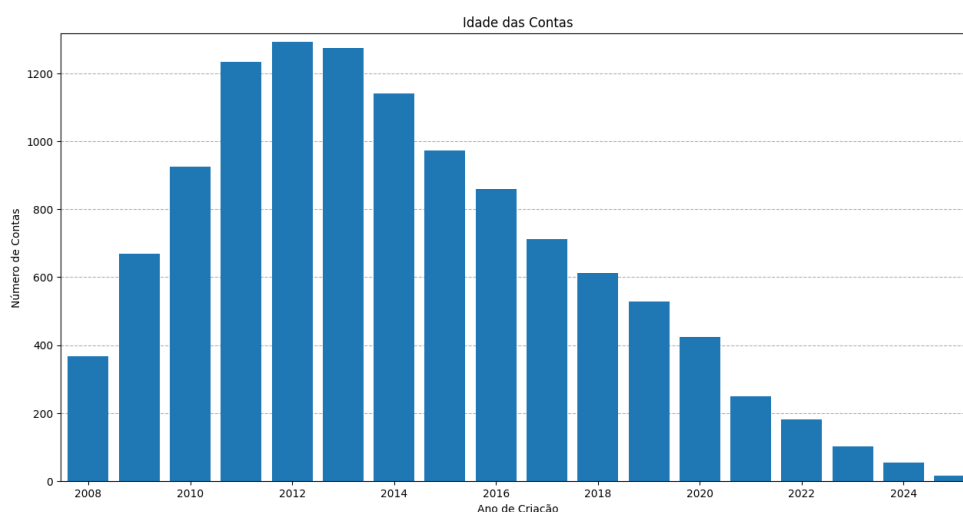
Fonte: Elaborada pelo autor.

encontrados numa análise geral se mantiveram semelhantes aos obtidos em um olhar mais aprofundado sobre os projetos, coisa que não aconteceu antes pois diversos países poderiam estar ou não entre os maiores contribuidores analisados.

4.2.3 Idades das contas

Um dos últimos gráficos gerados trata da distribuição etária das contas dos colaboradores. Conforme é perceptível na Figura 13, é possível perceber um crescimento relativamente rápido que tem seu ápice nos anos de 2011 a 2013 e logo depois tem uma queda gradual no número de contas que foram criadas naquele ano colaborando com os projetos. Graças a essa análise, é possível dizer que os colaboradores que se fazem mais presentes são aqueles que possuem de dez a quinze anos de conta no GitHub, além de que contas mais novas vão ficando gradualmente mais raras nesse cenário.

Figura 13 – Gráfico geral das idades das contas.



Fonte: Elaborada pelo autor.

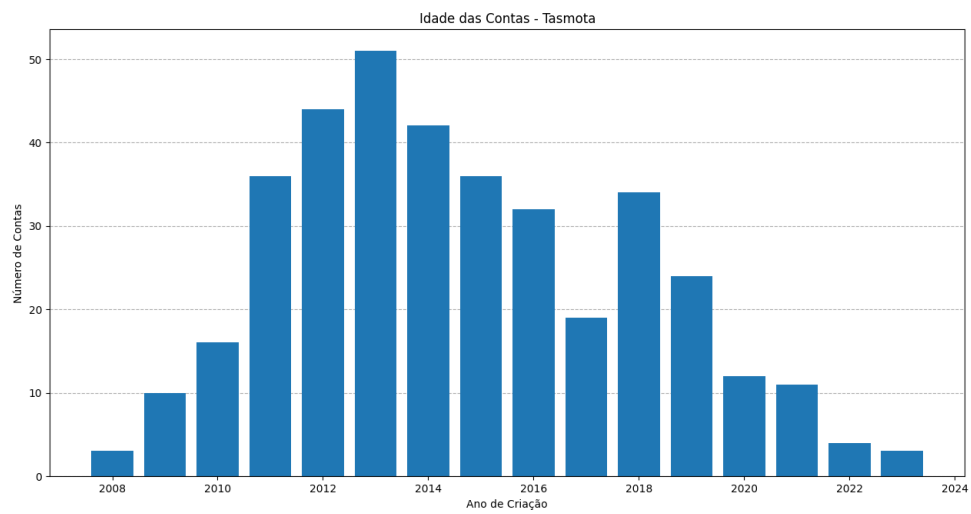
Olhando novamente para os projetos selecionados como exemplos, é visível que nenhum dos dois segue o mesmo padrão do gráfico anterior, na Figura 14, que é referente ao projeto Tasmota, tem um crescimento mais lento até 2010, possuindo um aumento rápido nos anos de 2011 a 2013, assim como, posteriormente, uma redução um pouco maior a cada ano que é interrompida em 2018 por um aumento significativo no número de contas antes de voltar a ter uma queda, chegando a não possuir colaboradores que criaram as contas em 2024 e 2025. Paralelamente, na Figura 15, que se refere ao projeto vscode, o crescimento inicial é um pouco mais simples até a chegada de 2012 com um pico de contas criadas nesse anos e imediatamente depois ele volta a seguir um ritmo mais lento, quando a quantidade de colaboradores começa a diminuir novamente, já não há um padrão evidente, apenas que a tendência é diminuir, podendo ou não possuir pequenos aumentos nas quantidades de contas criadas em determinados anos.

Levando em consideração o que foi apresentado pelos gráficos é possível afirmar que a maior quantidade de contas que colaboram em projetos de código aberto possuem por volta de dez a quinze anos, porém o crescimento ou decaimento da quantidade de colaboradores não possui um padrão claro, sendo assim, provavelmente seu maior valor de contas estará por volta de 2012, porém os anos anteriores e posteriores podem acabar possuindo valores arbitrários, tal qual o caso do gráfico do projeto vscode.

4.2.4 Contribuições por idade das contas

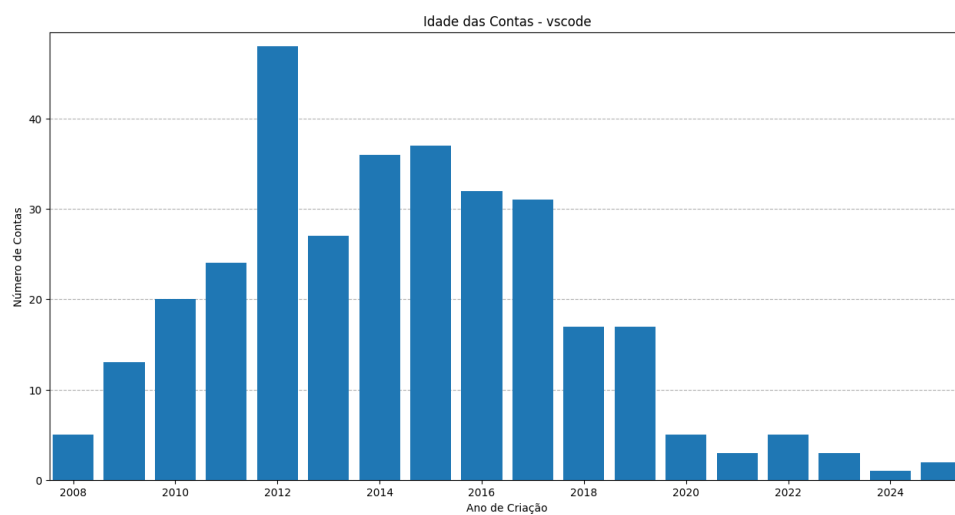
Por fim, a última análise gráfica feita por este trabalho tem como objetivo verificar a quantidade de contribuições por idade independente do número de contas que a mesma possui, na Figura 16 é visível o *treemap* dos valores gerais e é possível perceber que mesmo anos de

Figura 14 – Gráfico do projeto Tasmota sobre as idades das contas.



Fonte: Elaborada pelo autor.

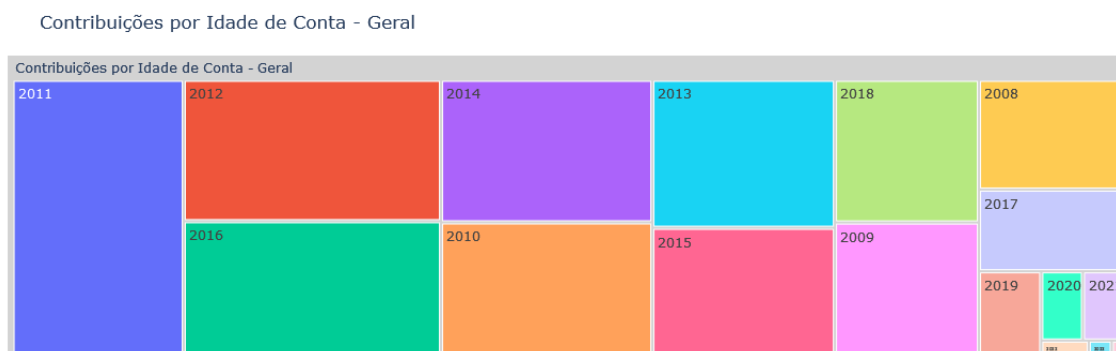
Figura 15 – Gráfico do projeto vscode sobre as idades das contas.



Fonte: Elaborada pelo autor.

criação que não estão entre aqueles com maior quantidade de colaboradores, como é o caso de contas criadas em 2016, se fazem presentes entre os que possuem grandes quantidades de contribuições. Um ponto interessante é o fato de que outras idades de contas possuem um valor considerável de contribuições, tais quais aquelas criadas em 2009 ou 2018, não o bastante para se destacarem demais, mas grande o suficiente para ainda serem percebidas juntas às demais.

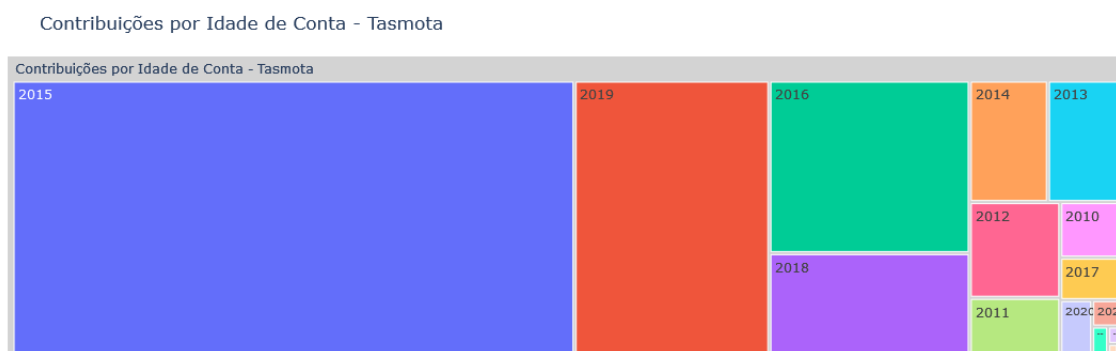
Figura 16 – Gráfico geral das contribuições por idades das contas.



Fonte: Elaborada pelo autor.

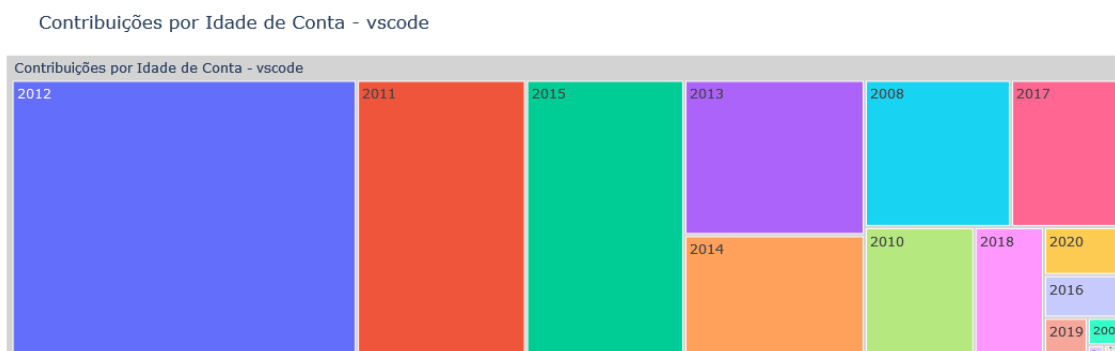
Analisando uma última vez os exemplos selecionados, é perceptível que no projeto Tasmota, que pode ser visto na Figura 17, há uma dominância de contribuições feitas por contas criadas em 2015 e 2019, sendo interessante não só a diferença numérica dessas contas para as demais, mas também pelo fato de nenhuma das duas estarem entre as três idades mais comuns do projeto. Por sua vez, na Figura 18, que é referente ao projeto vscode, é visível que anos como 2012 e 2015 estão presentes tanto como o ano de criação das contas com a maior quantidade de colaboradores, como o ano com o maior número de contribuições, chegando ao ponto de suprimir contas mais novas no gráfico, sendo efetivamente o oposto do que acontece no projeto Tasmota.

Figura 17 – Gráfico do projeto Tasmota sobre as contribuições por idades das contas.



Fonte: Elaborada pelo autor.

Figura 18 – Gráfico do projeto vscode sobre as contribuições por idades das contas.



Fonte: Elaborada pelo autor.

Uma vez que as análises de contribuições por idade da conta estão feitas, é possível chegar em algumas conclusões, a primeira sendo que, apesar da boa distribuição geral de contribuições, isso não se reflete para valores específicos de cada projeto, com os dois gráficos que foram usados como exemplos possuindo dois ou três anos de criação preenchendo pelo menos metade da área, um segundo ponto que também se revela é o fato de que não necessariamente mais contas de uma certa idade estejam associadas com mais contribuições para aquela idade, vide que tirando o vscode, o ano com mais contribuições não era o ano com mais colaboradores. Outro ponto interessante é o fato de que os projetos não seguem um padrão em seus principais contribuidores, ou seja, mesmo que anos como 2010 estejam entre os cinco com mais contribuições, ele não aparece com muitas contribuições em nenhum gráfico pego de exemplo.

5 Conclusão

Este trabalho realizou uma extensa pesquisa por meio de coleta de mais de onze mil colaboradores de projetos de código aberto. Foram realizados tratamentos de dados relativos a locais e horários. Foram gerados resultados usando diferentes tipos de bibliotecas gráficas e outras ferramentas da linguagem Python. Com isto, utilizando-se da ferramenta desenvolvida neste projeto, foi possível analisar o perfil dos colaboradores de projetos de código aberto em plataformas de repositórios de *software*, sendo listadas a seguir algumas das principais características encontradas nesses colaboradores.

As primeiras características que merecem destaque é que grande parte desses colaboradores estão localizados nos Estados Unidos da América ou na Europa, tal fato pode trazer a hipóteses da existência da relação entre a taxa de escolaridade e a presença desses desenvolvedores. Outras hipóteses que podem ser criadas analisando os resultados é se a população de um país acaba não possuindo muito efeito sobre a presença de colaboradores, se casos como os Estados Unidos da América, o país com mais desenvolvedores que colaboraram com projetos de código aberto, se destaca por sua população ou por outros fatores.

Ademais, os Estados Unidos da América também é o país com o maior número de contribuições nos projetos analisados, ultrapassando a marca de duzentas e cinquenta mil. Apesar da maioria das contribuições e dos desenvolvedores serem desse país, não é garantido que ele sempre possuirá uma grande influência no projeto, vide que geralmente as maiorias das contribuições vêm de dois ou três países que podem ou não incluir os Estados Unidos da América. Vale ressaltar que grande parte dos colaboradores não colocam uma localização exata em seus perfis, o que faz com que tudo que foi definido até então envolva apenas os desenvolvedores que compartilharam sua localização.

Outras características que se fazem presentes em contribuidores de programas de código aberto é o fato de que grande parte de suas contas possuem de dez a quinze anos de idade, onde 85% dos usuários coletados ainda estão em atividade. Essas características se mantêm iguais mesmo quando analisadas em projetos específicos, só não havendo uma constância exata para a quantidade de contas por idade. Quanto às contribuições realizadas por essas contas organizadas por idade, é possível perceber uma divisão relativamente equilibrada, com destaque para contas criadas em 2011 e 2012, porém, tal qual anteriormente, quando é analisado em projetos específicos, não é constatada uma constância exata.

Portanto, o perfil dos colaboradores de projetos de código aberto em plataformas de repositórios de *software* se apresenta sendo majoritariamente dos Estados Unidos da América, com uma vasta quantidade de contribuições, possuindo contas criadas por meados de 2012, com 85% deles ainda estando ativos. Esse perfil encontrado possui uma vasta importância

para a manutenção das comunidades de projetos de código aberto, pois com ele é possível saber o comportamento e distribuição dos colaboradores do mesmo, auxiliando na tomada de decisões ou em processos que tenham como enfoque os seus colaboradores, como traduções e ferramentas que os mantenedores dos processos possam querer implementar para seus colaboradores.

Para trabalhos futuros, o principal objetivo é responder às hipóteses levantadas durante a pesquisa, tais como a relação entre o IDH dos países e sua quantidade de contribuidores, se o perfil encontrado se mantém o mesmo caso uma amostragem maior ou análises diferentes que podem ser feitas. Os trabalhos futuros também devem focar em tornar a ferramenta desenvolvida mais acessível e automática, removendo a necessidade de interagir com o código, a obrigatoriedade de execução de múltiplos programas, assim como acelerar sua execução. Enfim, novos trabalhos devem focar no desenvolvimento de ferramentas para a manutenção e pesquisa da comunidade de colaboradores de programas de código aberto, auxiliando no crescimento e entendimento da mesma.

Referências

- ASSOCIATION, D. S. *Data science code of professional conduct*. 2025. Disponível em: <<<https://dev.datascienceassn.org/code-conduct>>>. Acesso em 14 de Outubro de 2025.
- BLINCOE, K.; SHEORAN, J.; GOGGINS, S.; PETAKOVIC, E.; DAMIAN, D. Understanding the popular users: Following, affiliation influence and leadership on github. *Information and Software Technology*, v. 70, p. 30—39, 2016.
- COLAZO, J.; FANG, Y. Impact of license choice on open source software development activity. *JASIST*, v. 60, p. 997–1011, 05 2009.
- CORBLY, J. E. The free software alternative: Freeware, open-source software, and libraries. *Information Technology and Libraries*, v. 33, p. 65—75, 2014.
- ECONÔMICO, O. para a Cooperação e D. *Population with tertiary education*. 2023. Disponível em: <<<https://www.oecd.org/en/data/indicators/population-with-tertiary-education.html#indicator-chart>>>. Acesso em 27 de Outubro de 2025.
- FUGGETTA., A. Open source software—an evaluation. *Journal of Systems and Software*, v. 66, p. 77–90, 2003.
- GEROSA, M.; WIESE, I.; TRINKENREICH, B.; LINK, G.; ROBLES, G.; TREUDE, C.; STEINMACHER, I.; SARMA, A. The shifting sands of motivation: Revisiting what drives contributors in open source. In: *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. [S.l.: s.n.], 2021. p. 1046–1058.
- GUIZANI, M.; ZIMMERMANN, T.; SARMA, A.; FORD, D. Attracting and retaining oss contributors with a maintainer dashboard. In: *2022 IEEE/ACM 44th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. [S.l.: s.n.], 2022. p. 36–40.
- HASSAN, A. E. The road ahead for mining software repositories. *Frontiers of Software Maintenance*, p. 48–57, 2008.
- JUHÁSZ, L.; MOONEY, P. “i think i discovered a military base in the middle of the ocean”—null island, the most real of fictional places. *IEEE Access*, v. 10, p. 84147–84165, 2022.
- KALLIAMVAKOU, E.; GOUSIOS, G.; BLINCOE, K.; SINGER, L.; GERMAN, D. M.; DAMIAN, D. The promises and perils of mining github. In: *Proceedings of the 11th Working Conference on Mining Software Repositories*. New York, NY, USA: Association for Computing Machinery, 2014. (MSR 2014), p. 92–101. ISBN 9781450328630. Disponível em: <<https://doi.org/10.1145/2597073.2597074>>.
- KALLIAMVAKOU, E.; GOUSIOS, G.; BLINCOE, K.; SINGER, L.; GERMAN, D. M.; DAMIAN, D. An in-depth study of the promises and perils of mining github. *Empir Software Engineering*, v. 21, p. 2035—2071, 2016.

QIU, H. S.; ZHAO, Z. H.; YU, T. K.; WANG, J.; MA, A.; FANG, H.; DABBISH, L.; VASILESCU, B. Gender representation among contributors to open-source infrastructure : An analysis of 20 package manager ecosystems. In: *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. [S.l.: s.n.], 2023. p. 180–187.

SEN, R.; SUBRAMANIAM, C.; NELSON, M. Determinants of the choice of open source software license. *J. of Management Information Systems*, v. 25, p. 207–240, 01 2009.

UNIDAS, O. das N. *World Population Prospects 2024*. 2024. Disponível em: <<<https://population.un.org/wpp/>>>. Acesso em 27 de Outubro de 2025.

WU, M.-W.; LIN, Y.-D. Open source software development: an overview. *Computer*, v. 34, n. 6, p. 33–38, 2001.