

Investigando fatores que influenciam a participação de colaboradores em projetos de código aberto

Relatório final

Guilherme Montes de Luca (guilherme.montes@unesp.br)
Orientador: Higor Amario de Souza (higor.amario@unesp.br)

Resumo

Este estudo investiga os fatores que influenciam a participação de colaboradores em projetos de código aberto, com foco na popularidade dos projetos, número de contribuidores e contextos de desenvolvimento. O problema abordado é a dificuldade em manter uma alta taxa de contribuições em repositórios complexos e de grande volume. A relevância da pesquisa está na importância dos projetos open source para a inovação tecnológica e na necessidade de estratégias que incentivem a participação contínua dos desenvolvedores.

A pesquisa utilizou mineração de dados de *commits* e *pull requests* de repositórios como Go, Dask, Pandas e Ember.js, aplicando ferramentas como PyDriller e Git CLI. Os resultados mostram que projetos populares atraem mais contribuições, mas enfrentam desafios para manter consistência nas *pull requests*. Concluiu-se que práticas de desenvolvimento mais acessíveis e melhor gerenciamento das contribuições podem aumentar o engajamento em projetos de código aberto.

1 Introdução

A colaboração em projetos de código aberto (*open source*) é um tema que vem sendo estudado ao longo dos últimos anos (Tymchuk et al., 2014; Puentes, 2015; Steinmacher et al., 2019). A contribuição nesse tipo de projeto é voluntária e depende da capacidade de engajamento de pessoas interessadas em contribuir com um determinado projeto (Chen et al., 2022).

A comunidade de desenvolvimento de software de código aberto é conhecida por sua capacidade de colaboração e inovação. Muitas organizações e indivíduos dependem do trabalho desses projetos para impulsionar seus negócios e iniciativas pessoais. Assim, a participação ativa em projetos de código aberto está intrinsecamente ligada ao sentimento de comunidade e conexão dos colaboradores com a produção de software. Além disso, a conexão emocional com o projeto e com a comunidade pode levar os colaboradores a se dedicarem mais intensamente, contribuindo ativamente com ideias de novas funcionalidades, correções de defeitos (*bugs*) e melhorias no código. Portanto, o sentimento de comunidade e conexão desempenha um papel fundamental no desenvolvimento de software de código aberto (Steinmacher et al., 2019), impulsionando a excelência técnica e a evolução contínua dos projetos. Além disso, os projetos *open source* proporcionam diversos benefícios para os desenvolvedores colaboradores, que podem melhorar seus conhecimentos sobre programação em projetos de grande porte, aprendem novos métodos e ferramentas, além de interagirem com desenvolvedores mais experientes (Constantino et al., 2020).

Geralmente, projetos de código aberto tem um grupo de colaboradores mantenedores, que controlam os envios de modificação do código para evitar que sejam introduzidos defeitos no programa, código mal escrito ou incompleto. Os colaboradores encaminham requisições de envio de código (*pull requests*) e esses códigos são avaliados pelos mantenedores, podendo ser aceitos ou não. Dadas as características voluntárias desses projetos, os processos de *pull requests* podem demorar. Essa demora pode ser uma das razões que desmotivam desenvolvedores a colaborar

com um projeto. Além disso, existem outras possíveis barreiras que dificultam a colaboração, tais como a demora na resposta de perguntas feitas pelos colaboradores ou a falta de conhecimento sobre testes automatizados (Steinmacher et al., 2019). Fatores indicam que a colaboração em projetos é um fenômeno efêmero, o que pode não ser sustentável a longo prazo Tymchuk et al. (2014).

Este relatório referente ao projeto de Iniciação Científica intitulado "*Investigando fatores que influenciam a participação de colaboradores em projetos de código aberto*" tem como objetivo explorar as variáveis que afetam o engajamento de indivíduos em iniciativas de software de código aberto. O autor, Guilherme Montes de Luca, apresenta um estudo estruturado em várias etapas, desde uma revisão bibliográfica até a aplicação de ferramentas como PyDriller e Git CLI para extração de dados de *commits* e *pull requests* de repositórios selecionados.

A pesquisa examina repositórios de diversos projetos *open source* conhecidos, como Go, Dask, Pandas, Ember.js, e Godot, buscando entender os padrões de colaboração e contribuição. A análise inclui uma investigação sobre as barreiras que podem limitar a participação dos colaboradores e o impacto de fatores como a popularidade dos projetos e os níveis de atividade ao longo do tempo.

O trabalho se destaca por aplicar técnicas de mineração de repositórios e ciência de dados para gerar gráficos e visualizações, revelando a dinâmica de *commits*, *pull requests* e o comportamento de colaboradores em diferentes contextos de desenvolvimento de software.

1.1 Objetivos

O objetivo desta pesquisa é investigar os fatores que influenciam a participação de colaboradores em projetos de código aberto, buscando entender como aspectos relacionados a popularidade dos projetos, quantidade de contribuidores e se o tipo de projeto afeta o engajamento e as contribuições dos desenvolvedores.

Esta pesquisa parte das seguintes hipóteses:

- A popularidade de um projeto está diretamente relacionada ao nível de participação dos colaboradores.
- Repositórios com maior número de contribuidores ativos apresentam uma maior frequência de *commits* e *pull requests*.
- A diversidade de tipos de projetos e de ferramentas usadas influencia a quantidade e a qualidade das contribuições.

1.2 Objetivos Específicos

Os objetivos específicos desta pesquisa são:

- Coletar e analisar dados de *commits* e *pull requests* de diversos repositórios de código aberto;
- Identificar padrões de participação em diferentes projetos e contextos;
- Analisar o impacto da popularidade e da área de atuação dos projetos na contribuição dos desenvolvedores;
- Propor recomendações para aumentar o engajamento de colaboradores em projetos de código aberto.

2 Materiais e métodos

A metodologia utilizada neste estudo para investigar os fatores que influenciam a participação de colaboradores em projetos de código aberto foi estruturada em diversas etapas relacionadas. O trabalho começou com um levantamento bibliográfico, com o intuito de identificar abordagens mais adequadas e as ferramentas disponíveis para realizar a análise. Foram revisadas várias fontes sobre mineração de repositórios de software e ciência de dados, resultando na escolha de ferramentas como o PyDriller, utilizado para a extração de dados dos commits. Primeiramente, o PyDriller foi utilizado para realizar testes e extrair informações dos commits de um repositório de código aberto. Contudo, encontrou-se limitações nessa ferramenta, pois ela fornecia apenas dados relacionados aos commits, sem fornecer informações sobre pull requests, que eram necessários para a análise mais completa da contribuição dos desenvolvedores.

Diante dessas limitações, a ferramenta Git CLI foi selecionada para a coleta de dados mais abrangentes. No início, o uso do Git CLI apresentou desafios, especialmente pela falta de documentação clara sobre como utilizá-lo de maneira eficiente. No entanto, após inúmeras tentativas, a utilização da ferramenta auxiliou a extração dos dados relacionados aos pull requests, como a caracterização de cada usuário e as taxas de aprovação.

A seleção dos repositórios de código aberto foi realizada com base em critérios específicos, como a popularidade dos projetos, o número de contribuidores e a área de atuação. Os repositórios escolhidos incluem projetos como Go, Dask, Pandas, Ember.js, GnuCash, I-Educar, Puter, Godot, Docusaurus e Reflex. A diversidade desses repositórios visou garantir que a amostra de projetos analisada fosse variada, permitindo um estudo das colaborações em diferentes contextos. As análises detalhadas estão disponíveis no seguinte link: [GitHub - Iniciação Científica](#).

- **Go:** Linguagem de programação de código aberto desenvolvida pelo Google em 2007 e lançada em código aberto em 2009. É fortemente tipada e compilada, sendo considerada uma potencial substituta do Java no futuro, já que seu código é executado diretamente pelo sistema operacional ou pelo processador.
- **Dask:** Biblioteca Python de código aberto para computação paralela, que expande o código Python de máquinas locais para grandes clusters distribuídos na nuvem.
- **Pandas:** Biblioteca Python para manipulação e análise de dados, fornecendo estruturas e operações para tabelas numéricas e séries temporais.
- **Ember.js:** Framework JavaScript para criar aplicações web de página única, baseado na arquitetura Model–view–viewmodel.
- **GnuCash:** Software multiplataforma de gestão de finanças, voltado tanto para uso pessoal quanto empresarial.
- **I-Educar:** Sistema de gestão escolar distribuído como software livre, com foco na administração eficaz de escolas, especialmente na esfera pública.
- **Puter:** Sistema operacional de internet avançado, projetado para ser rico em recursos, rápido e altamente extensível.
- **Godot:** Motor de jogos de código aberto que suporta exportação para diversas plataformas, desenvolvido inicialmente na América Latina.
- **Docusaurus:** Ferramenta que facilita a construção de sites de documentação de projetos de código aberto, proporcionando uma maneira simples de criar e manter páginas de documentação.
- **Reflex:** Framework de código aberto que capacita desenvolvedores a construir aplicações web full-stack inteiramente em Python puro — sem necessidade de JavaScript ou experiência em desenvolvimento web.

Esses repositórios foram escolhidos com base em suas características diversas, cobrindo uma ampla gama de áreas de atuação e tipos de colaboração em projetos open source. Os dados coletados passaram por um processo de limpeza e tratamento, que incluiu a remoção de informações duplicadas, alguns erros nas datas e a organização dos dados sobre pull requests e commits, além da padronização dos formatos para facilitar análises posteriores.

Na etapa de análise, foram empregadas técnicas estatísticas e de visualização de dados para explorar os padrões emergentes. Foram gerados gráficos que ajudaram a examinar a relação entre variáveis importantes, como a quantidade dos pull requests e commits ao longo do tempo, e o nível de atividade dos colaboradores ao longo do tempo. Essas visualizações permitiram a identificação de barreiras potenciais à participação dos colaboradores nos projetos open source.

3 Resultados e discussão

Nesta seção, apresentaremos gráficos que ilustram o comportamento dos repositórios analisados neste projeto. Esses gráficos oferecem uma visão detalhada das dinâmicas de contribuição e participação, permitindo uma melhor compreensão dos padrões de desenvolvimento observados em cada repositório.

3.1 Análise do Projeto Dask

A Figura 1 mostra um pico de atividade no início do projeto, seguido por uma estabilização na taxa de commits, indicando que, após a fase inicial, o ritmo de desenvolvimento se tornou mais constante. A análise das pull requests, revelada pela mesma figura, também mostra uma certa regularidade, com algumas variações que apontam períodos de maior colaboração, alternando com momentos mais estáveis. Já a Figura 2 destaca um número elevado de pull requests com o status **Commented**, incluindo um pico específico, enquanto os tipos **Approved**, **Changes Requested** e **Dismissed** se mantêm constantes ao longo do tempo. Por fim, a Figura 3 ilustra a participação significativa de membros do projeto, enquanto colaboradores externos e contribuidores apresentam um nível de atividade mais equilibrado e uniforme. Essa análise integrada sugere que o projeto passou por uma fase inicial intensa, que se estabilizou com o tempo, contando com um núcleo de membros ativos e um fluxo constante de contribuições externas.

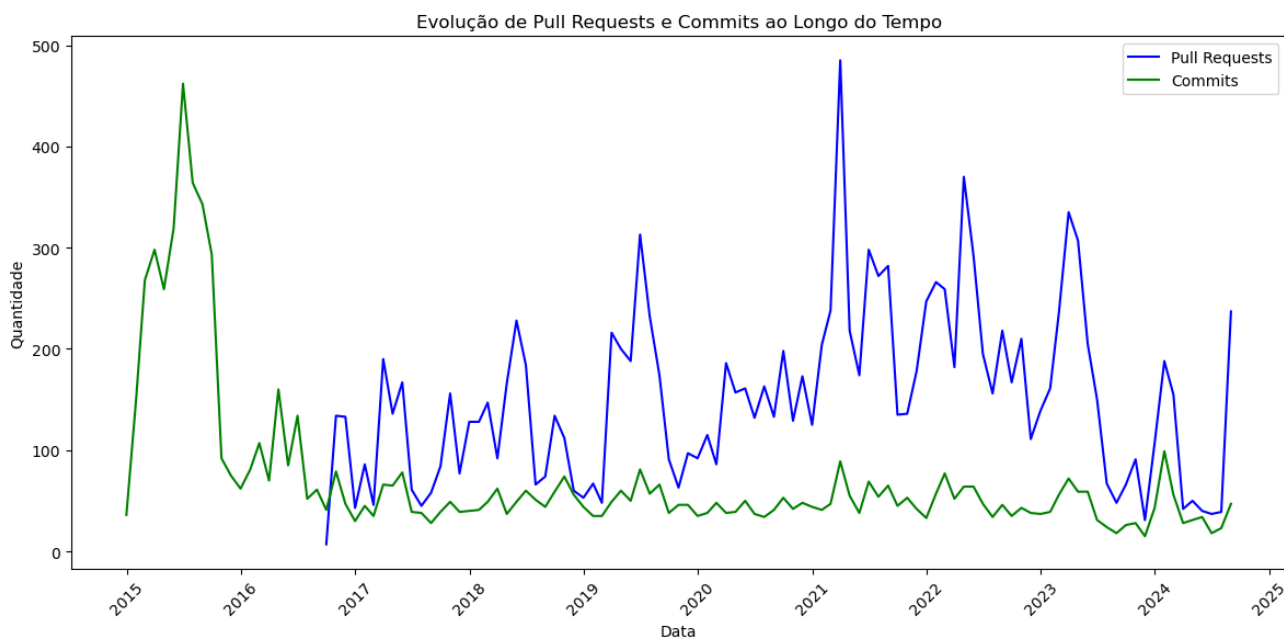


Figura 1: Comparação entre Commits e Pull Request ao Longo do Tempo no Repositório Dask

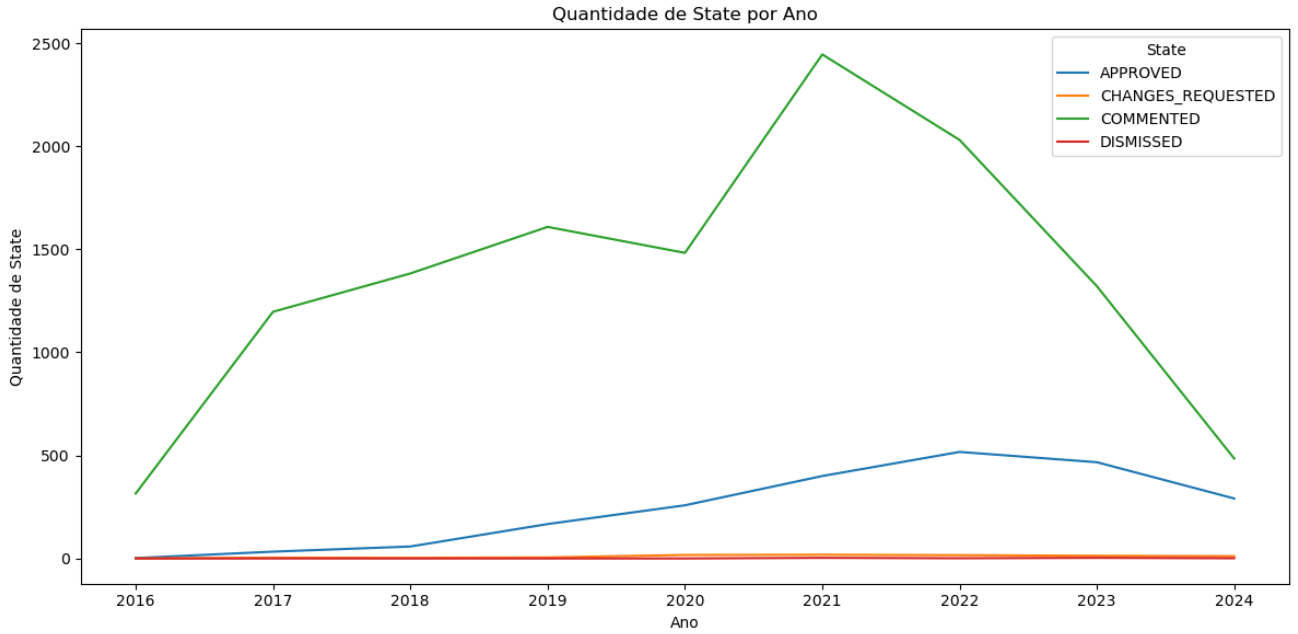


Figura 2: Quantidade de Pull Request classificados ao Longo do Tempo no Repositório Dask

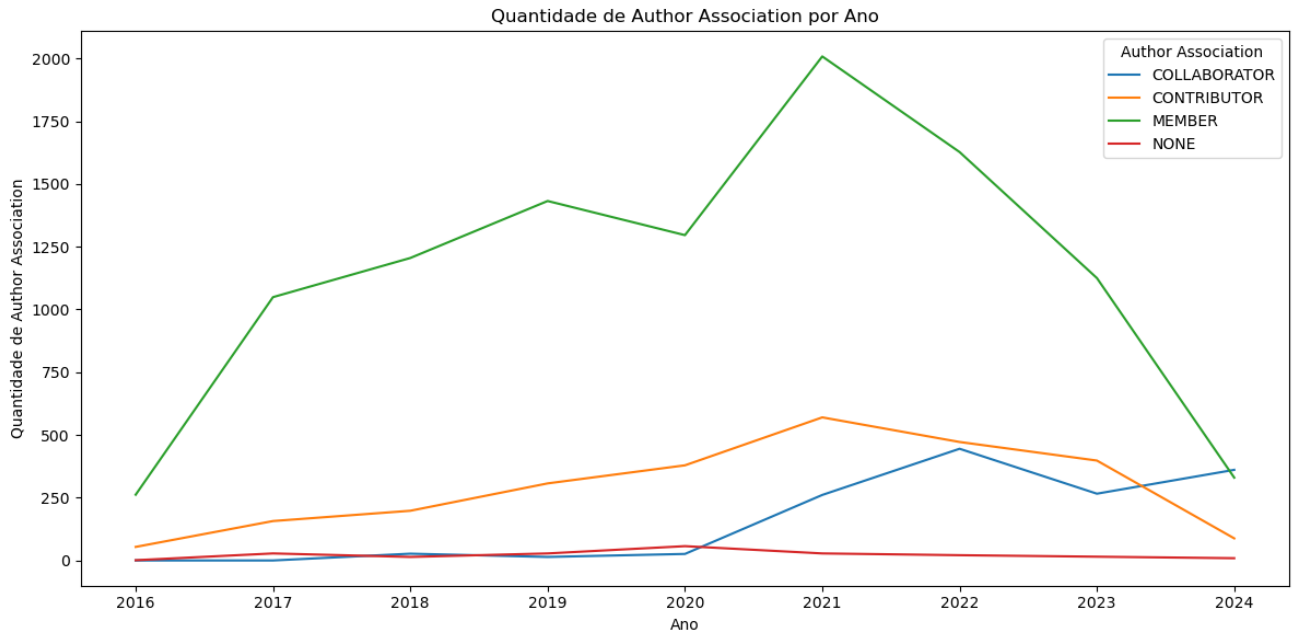


Figura 3: Quantidade de Pull Request com classificação de Autores ao Longo do Tempo no Repositório Dask

3.2 Análise do Projeto Docusauro

O projeto apresenta um padrão de commits irregular, com picos de atividade em determinados períodos, indicando fases de alta atividade seguidas por momentos mais calmos, como ilustrado na Figura 4. No início, a frequência de pull requests foi alta, com picos significativos de atividade; no entanto, observou-se uma queda acentuada no número de propostas recentemente, conforme mostrado na mesma figura. Além disso, o projeto registrou um número expressivo de pull requests do tipo **Commented**, mas não incluiu registros do tipo **Dismissed**, como ilustrado na Figura 5. Em relação à participação dos contribuintes, observou-se um pico inicial que diminuiu com o tempo, enquanto o número de colaboradores aumentou de forma constante até atingir outro pico, seguido por uma queda, como demonstrado na Figura 6. Membros e *owners* mantiveram uma presença mais baixa e estável ao longo do tempo, conforme observado na mesma figura. A Figura 4 destaca picos claros de atividade, que possivelmente coincidem com

grandes lançamentos ou o desenvolvimento de novas funcionalidades. Existem momentos em que a atividade diminui drasticamente, sugerindo um ritmo de desenvolvimento não contínuo. Esses picos são, provavelmente, associados a "sprints" de desenvolvimento ou grandes atualizações, seguidos por períodos de menor atividade.

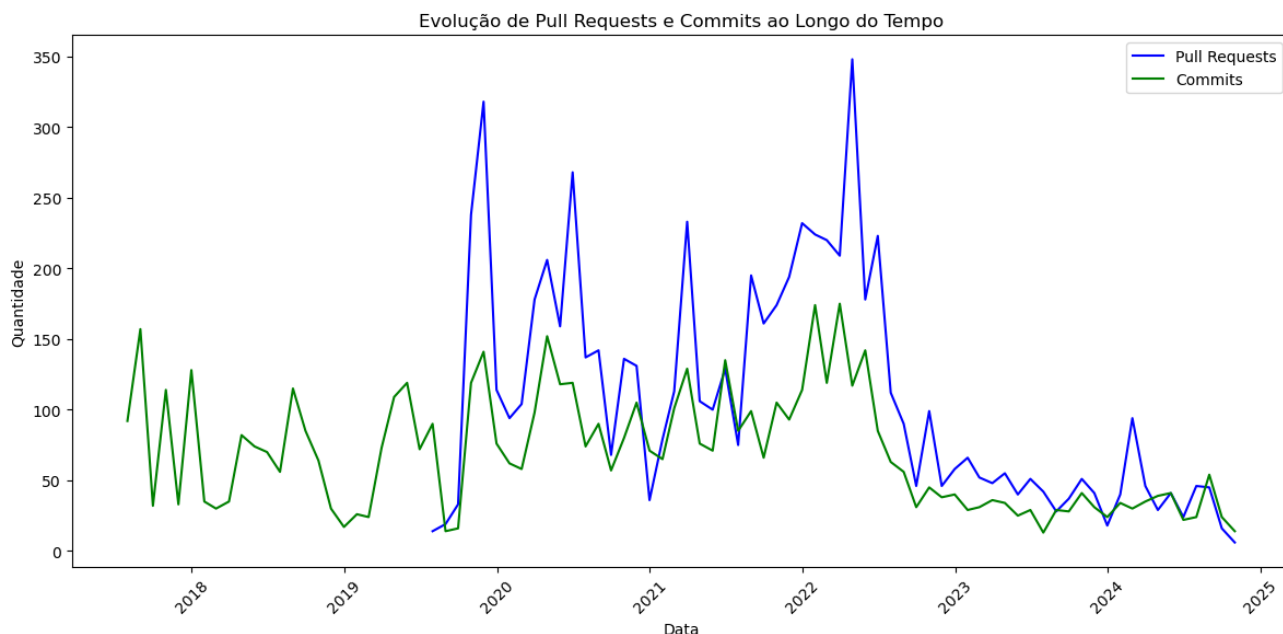


Figura 4: Comparação entre Commits e Pull Request ao Longo do Tempo no Repositório Docusauro

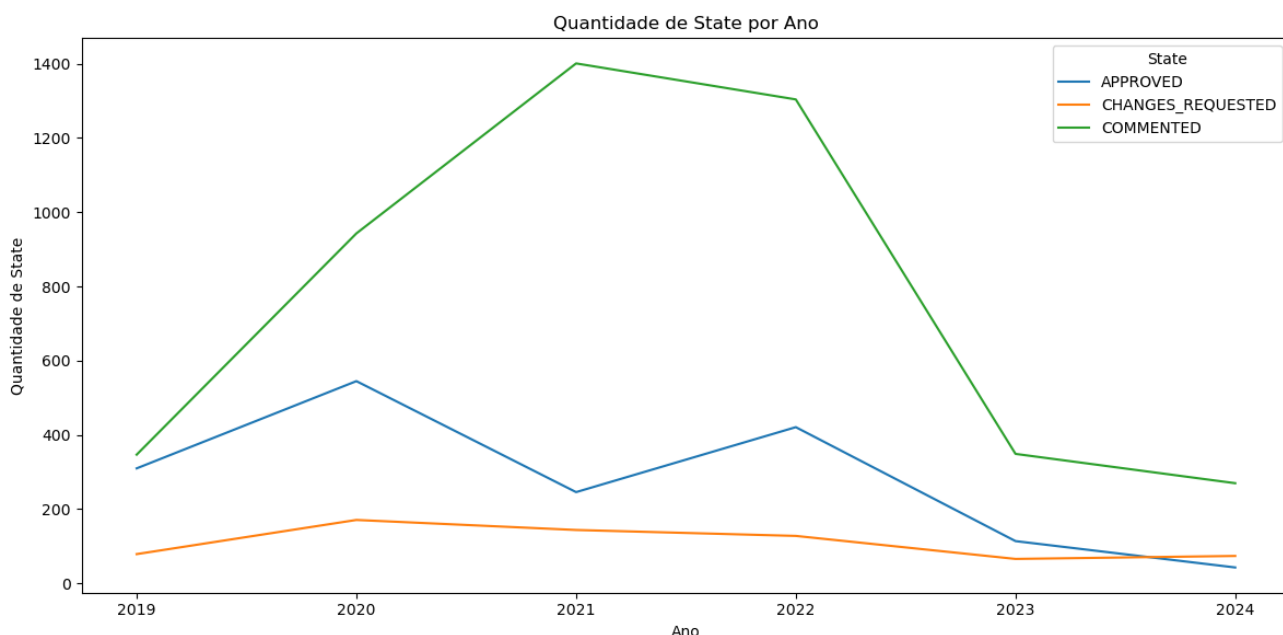


Figura 5: Quantidade de Pull Request classificados ao Longo do Tempo no Repositório Docusauro

3.3 Análise do Projeto EmberJS

O projeto apresenta uma atividade constante, com alguns picos ocasionais, o que sugere um fluxo de desenvolvimento estável, conforme mostrado na Figura 7. Inicialmente, houve uma alta taxa de pull requests, acompanhada de diversos picos de contribuição; no entanto, o ritmo diminuiu progressivamente ao longo do tempo, como ilustrado na mesma figura. Além disso, observa-se

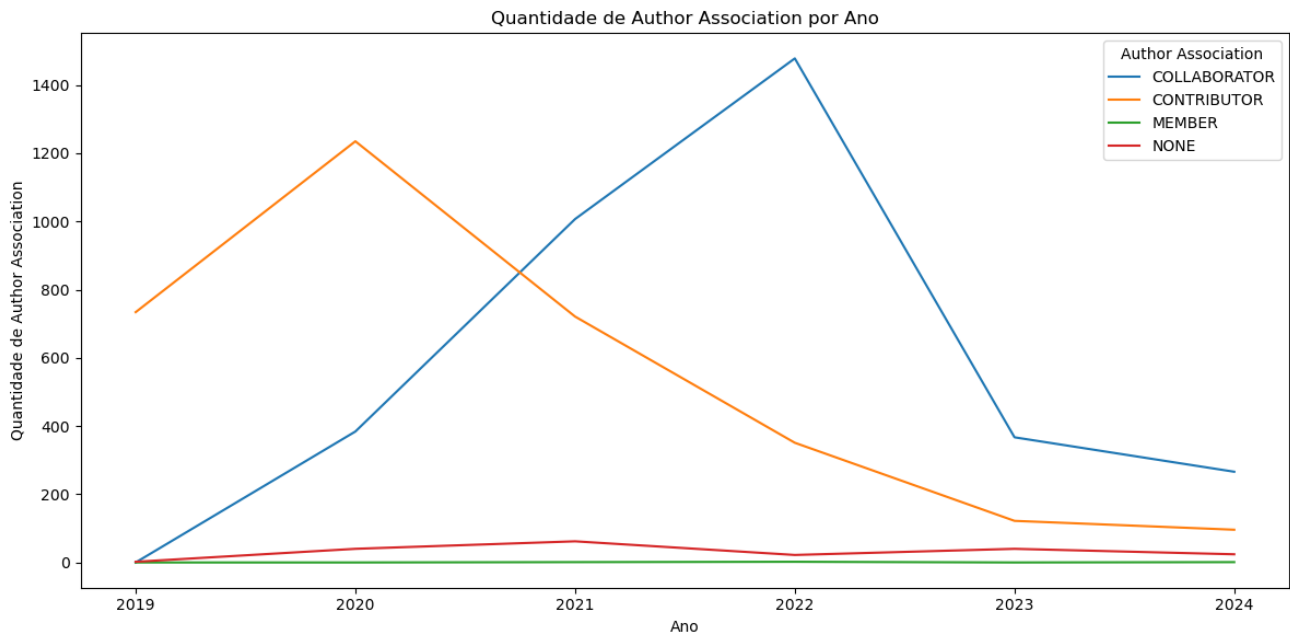


Figura 6: Quantidade de Pull Request com classificação de Autores ao Longo do Tempo no Repositório Docusauro

uma constância nos tipos de pull requests, especialmente os classificados como **Commented** e **Approved**, sendo os **Commented** sempre mais numerosos, como evidenciado na Figura 8. A participação de membros e colaboradores também se destaca, apresentando uma proporção estável ao longo do período analisado, como mostrado na Figura 9. Esses dados indicam que, embora o projeto mantenha uma atividade regular, os picos ocasionais estão provavelmente associados a lançamentos significativos ou eventos importantes. Em geral, a continuidade e estabilidade na atividade de pull requests sugerem que o projeto segue em um desenvolvimento ativo e contínuo, conforme ilustrado na Figura 7.

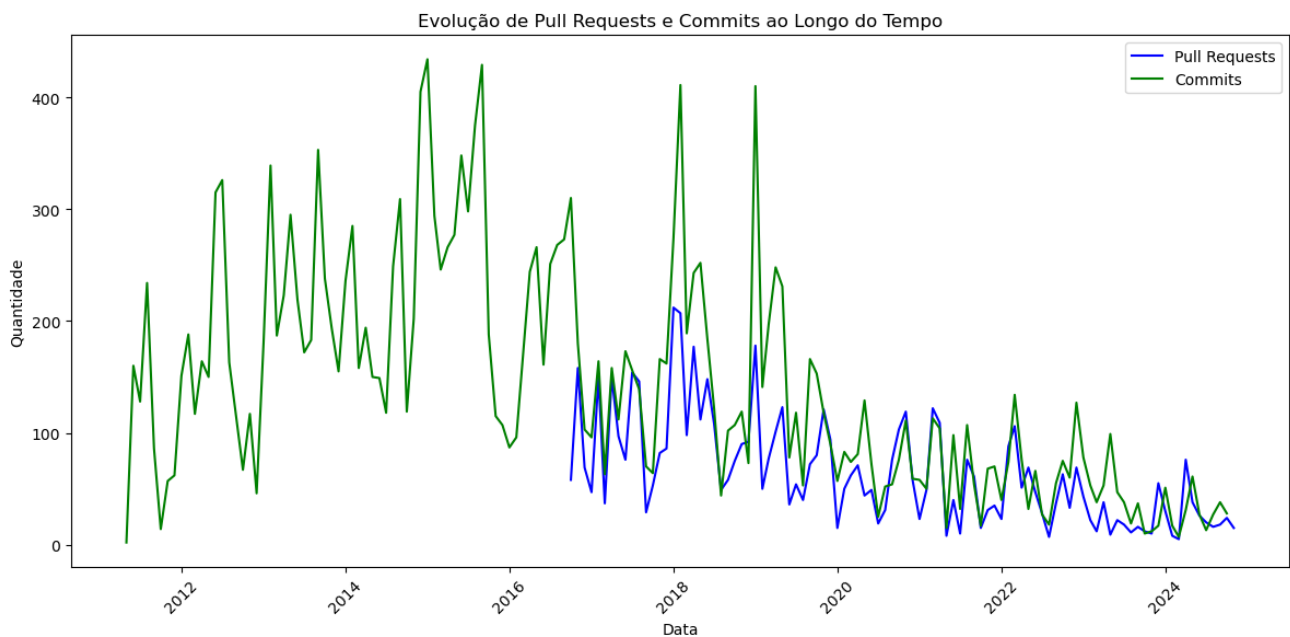


Figura 7: Comparação entre Commits e Pull Request ao Longo do Tempo no Repositório EmberJS

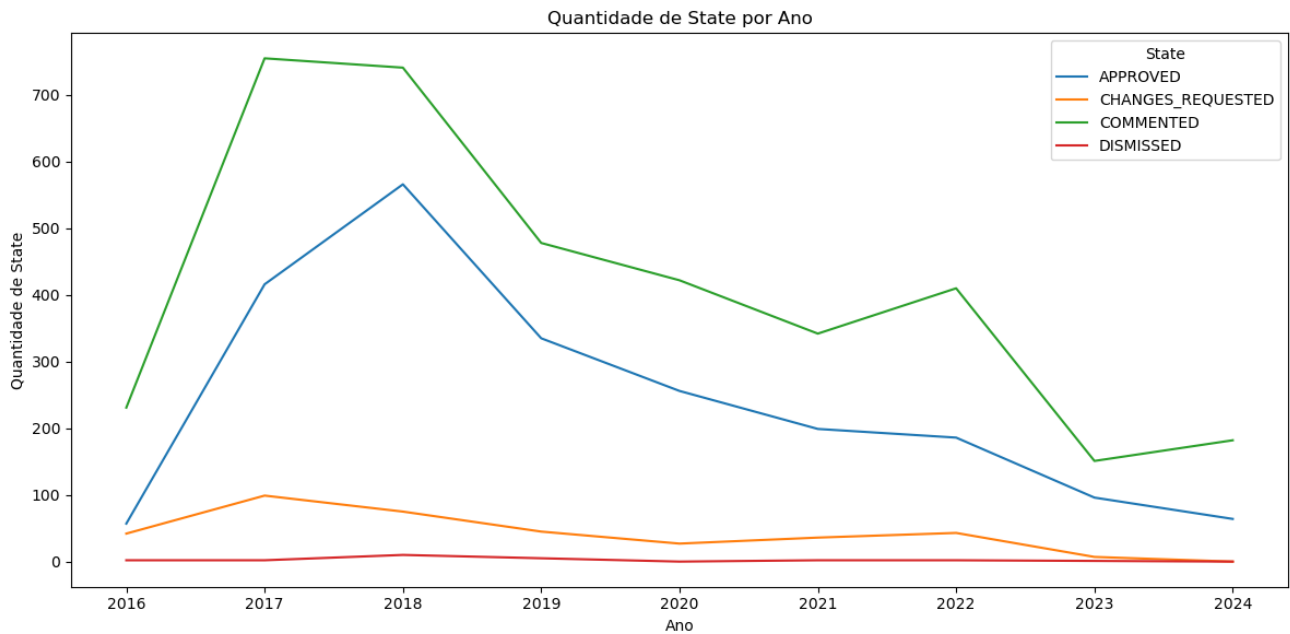


Figura 8: Quantidade de Pull Request classificados ao Longo do Tempo no Repositório EmberJS

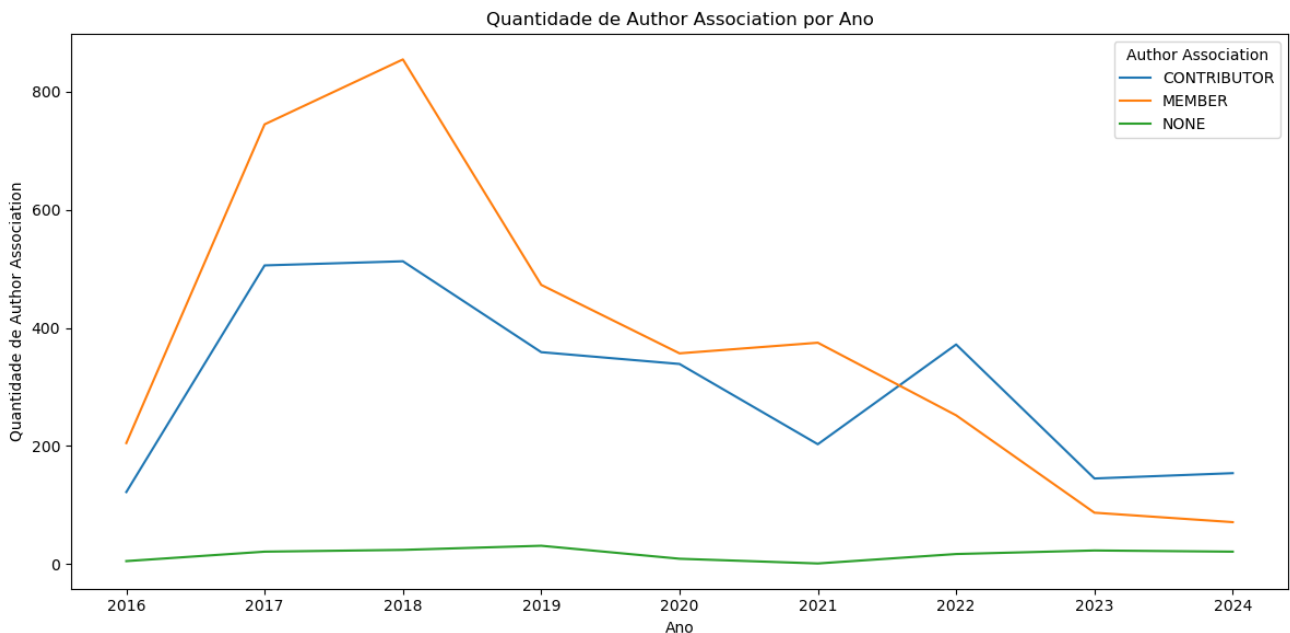


Figura 9: Quantidade de Pull Request com classificação de Autores ao Longo do Tempo no Repositório EmberJS

3.4 Análise do Projeto Gnucash

O desenvolvimento do projeto segue um padrão intermitente, com períodos de alta e baixa atividade, conforme ilustrado na Figura 10. Inicialmente, a taxa de pull requests era baixa, mas ao longo do tempo, o número de contribuições aumentou, com picos notáveis seguidos por períodos de maior constância na frequência de pull requests. Esse padrão sugere que o desenvolvimento do projeto ocorre em ciclos, com marcos de versões ou fases de planejamento mais longas. A análise dos tipos de pull requests mostra uma distribuição equilibrada entre as categorias **Approved**, **Changes Requested**, **Dismissed** e **Commented**, com uma quantidade significativa de pull requests **Commented**, conforme mostrado na Figura 11. Essa distribuição reflete um processo de revisão ativo e detalhado ao longo do tempo. Além disso, tanto membros quanto contribuidores do projeto mantiveram níveis de atividade semelhantes, com membros sempre superando os contribuidores em número, como evidenciado na Figura 12. Esse comportamento é consistente

com o padrão observado em outros projetos, como o Docusauro, que também apresenta picos de atividade seguidos por períodos de menor intensidade. Esse ciclo de desenvolvimento parece refletir um foco em grandes atualizações, seguidas de fases de manutenção ou planejamento, como ilustrado na Figura 10.

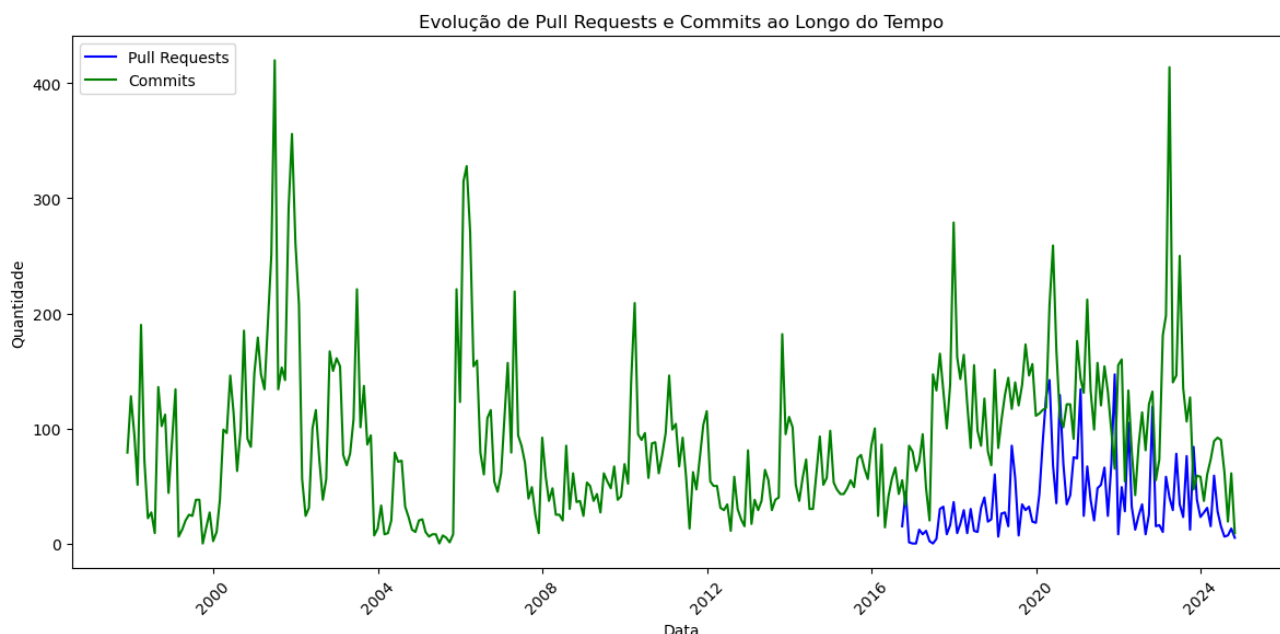


Figura 10: Comparação entre Commits e Pull Request ao Longo do Tempo no Repositório Gnocash

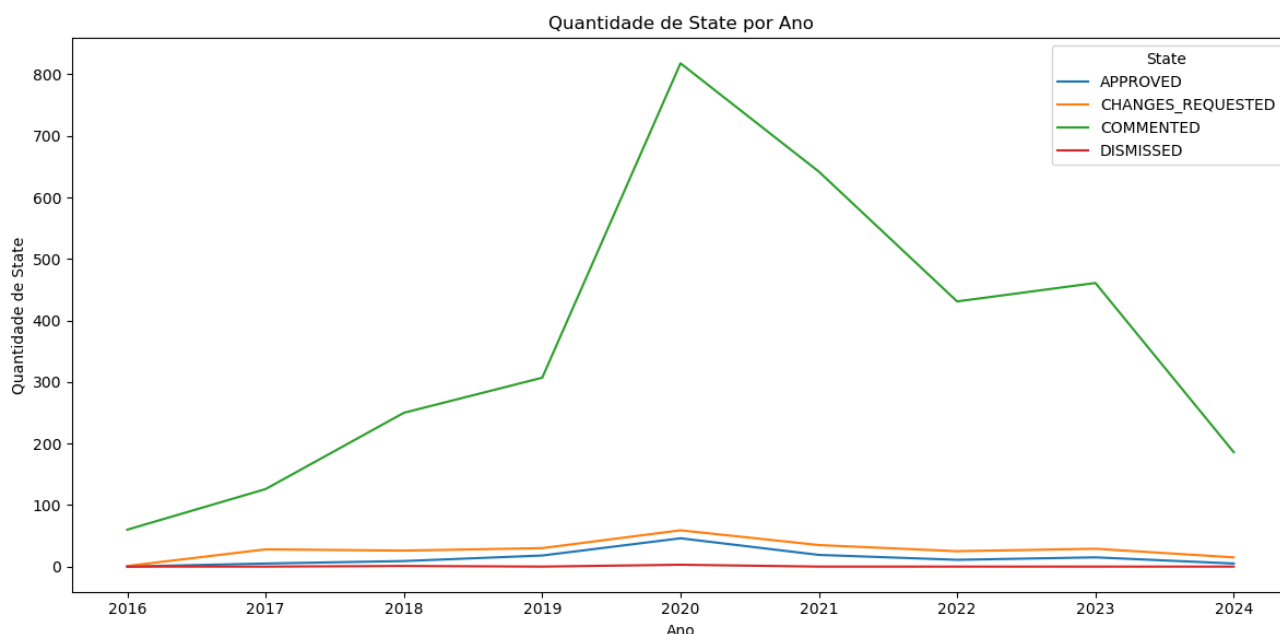


Figura 11: Quantidade de Pull Request classificados ao Longo do Tempo no Repositório Gnocash

3.5 Análise do Projeto Go

O projeto demonstra uma alta intensidade de commits de forma constante, refletindo um desenvolvimento ativo e regular, conforme ilustrado na Figura 13. Inicialmente, a atividade foi intensa, mas o ritmo de pull requests diminuiu ao longo do tempo, indicando uma desaceleração nas contribuições, como mostrado na mesma figura. Além disso, observou-se um elevado

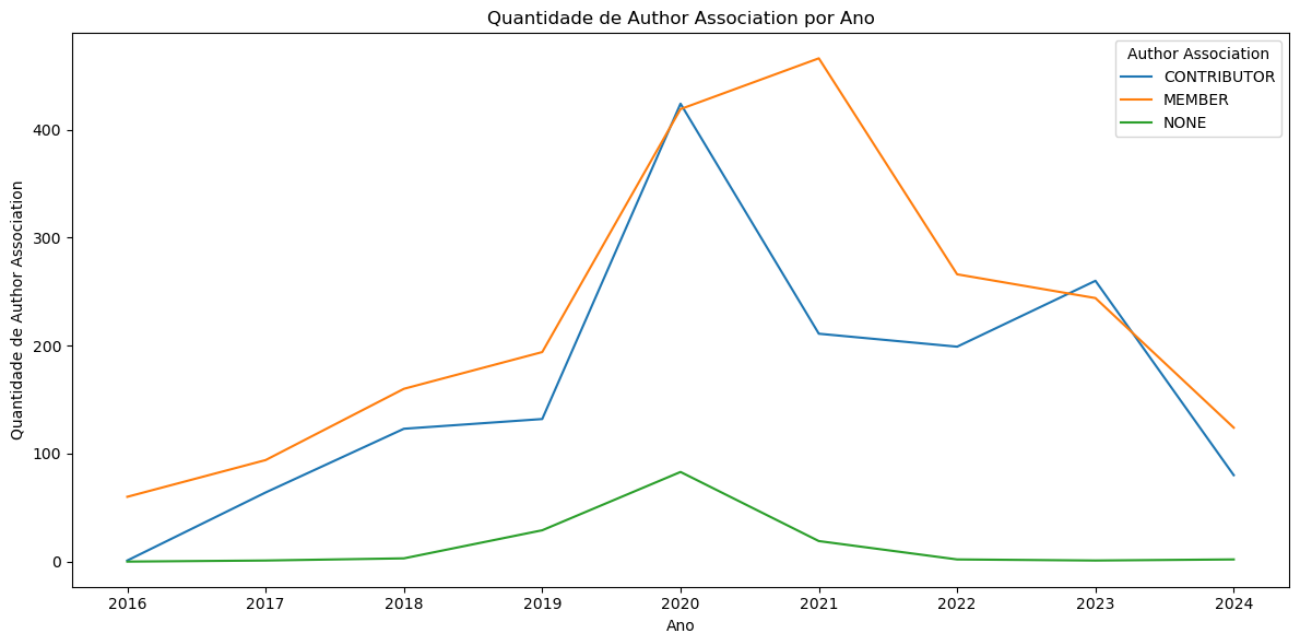


Figura 12: Quantidade de Pull Request com classificação de Autores ao Longo do Tempo no Repositório Gnucash

número de pull requests **Commented**, com algumas quedas, enquanto os pull requests **Approved** apresentaram um aumento gradual ao longo do tempo. Não foram registrados pull requests **Dismissed**, conforme ilustrado na Figura 14. A maior parte das pull requests foi enviada por autores classificados como **none** (sem um papel claramente definido no repositório), como pode ser visto na Figura 15. i Esse padrão sugere que, embora o ritmo de pull requests tenha diminuído, o projeto segue com um desenvolvimento intenso e contínuo, mantendo um alto número de commits sem grandes variações ao longo do tempo. Isso reflete um projeto com uma equipe sólida e um ritmo constante de evolução, como demonstrado na Figura 13.

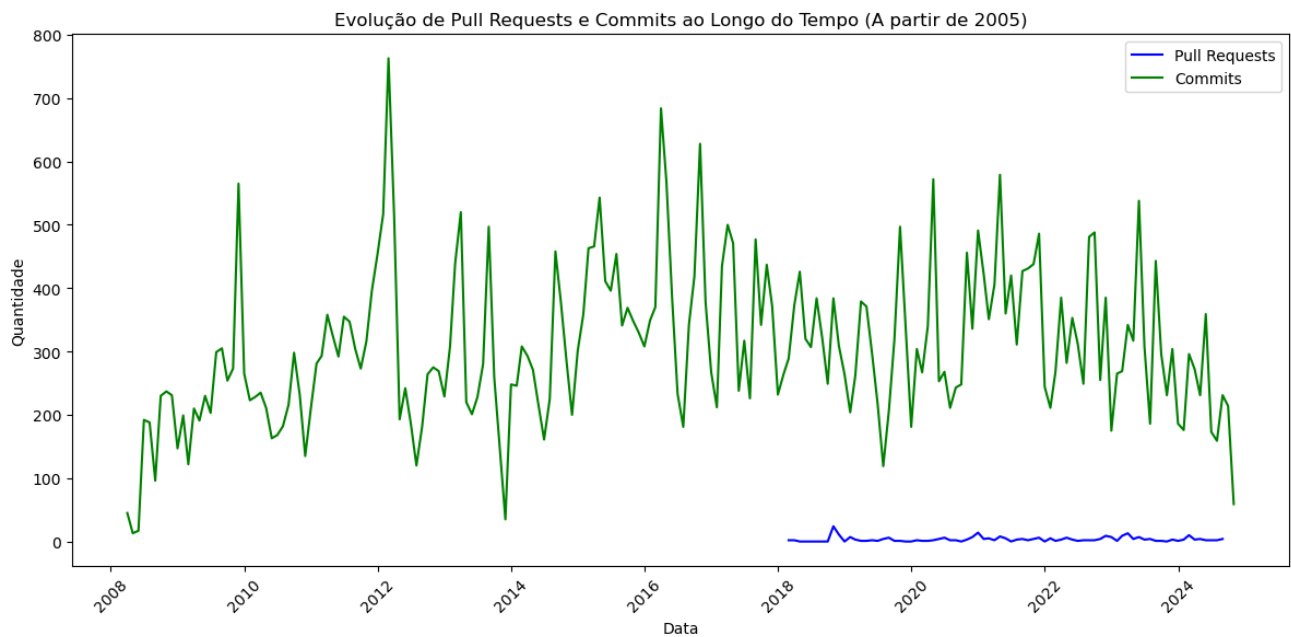


Figura 13: Comparação entre Commits e Pull Request ao Longo do Tempo no Repositório Go

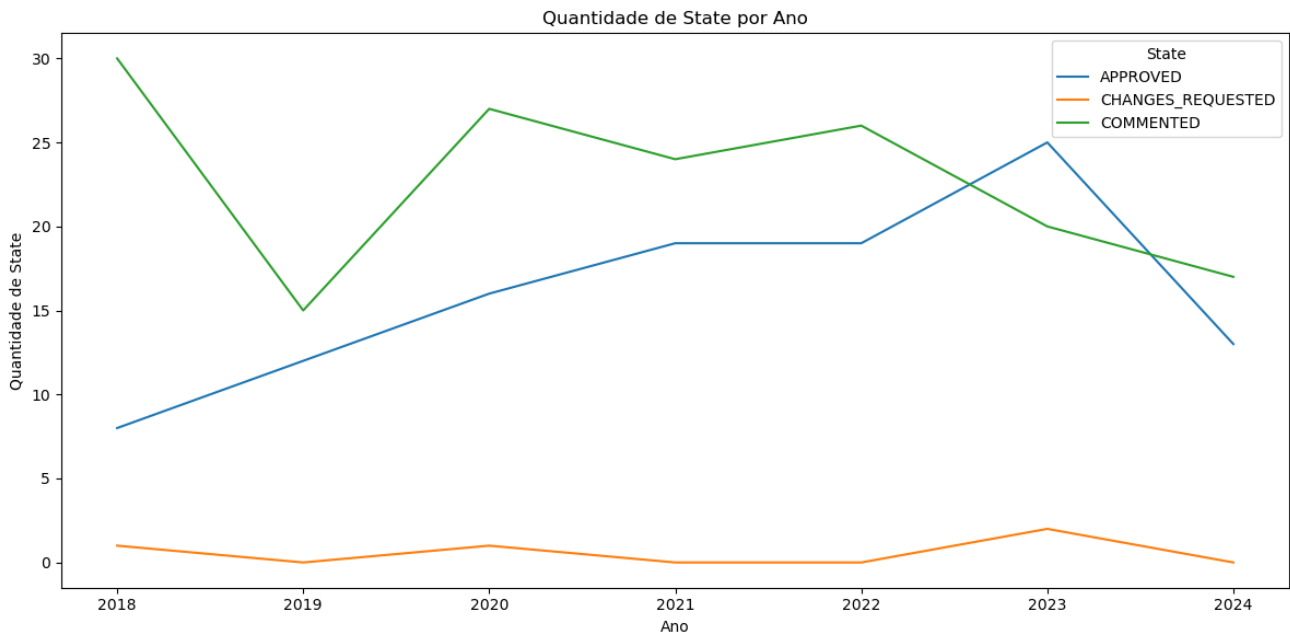


Figura 14: Quantidade de Pull Request classificados ao Longo do Tempo no Repositório Go

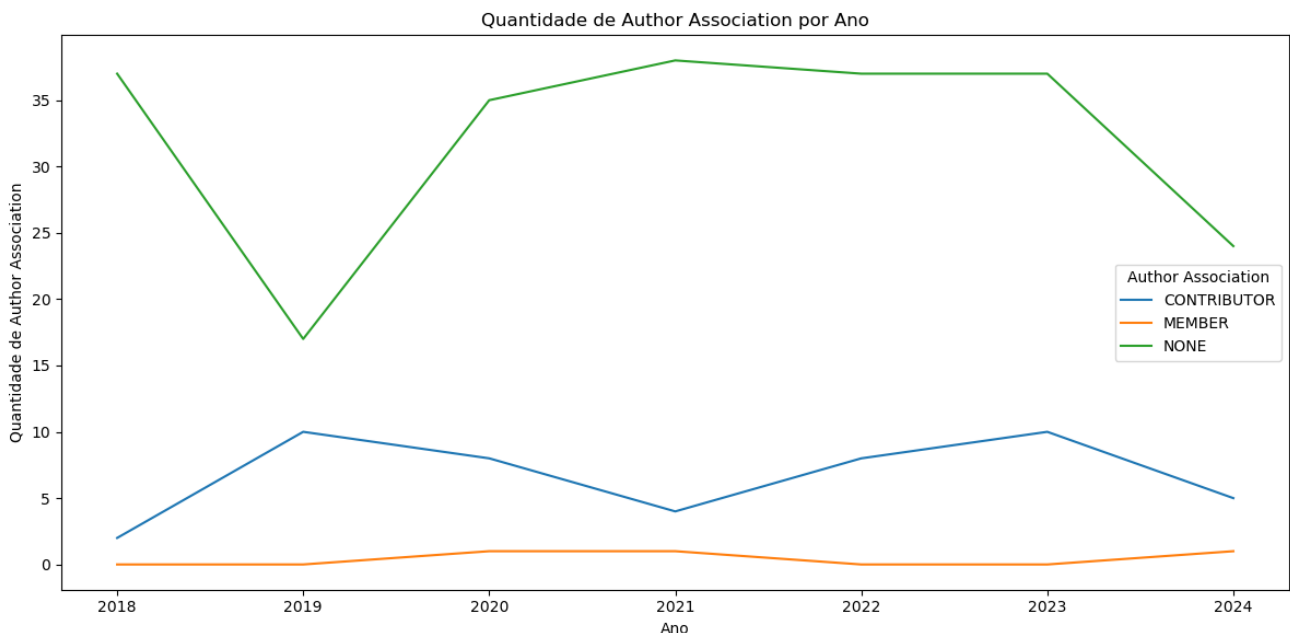


Figura 15: Quantidade de Pull Request com classificação de Autores ao Longo do Tempo no Repositório Go

3.6 Análise do Projeto Godot

O projeto inicialmente apresentou poucos picos de atividade, mas, após um período inicial de inatividade, manteve-se constante, sugerindo que o desenvolvimento ganhou ritmo ao longo do tempo, como ilustrado na Figura 16. A atividade, embora tenha mostrado uma queda recente, manteve uma alta e constante frequência de pull requests, evidenciando um ritmo estável de contribuições. Essa constância é refletida nos tipos de pull requests, com destaque para os tipos **Commented** e **Approved**, sendo **Commented** o mais frequente, conforme mostrado na Figura 17. Além disso, a análise por tipos de autores revela que membros do projeto apresentaram um alto índice de atividade, seguidos pelos contribuidores, que também desempenharam um papel expressivo, como mostrado na Figura 18. Em resumo, após um começo de baixa atividade e poucos picos, o projeto ganhou tração e se estabilizou em um ritmo regular de desenvolvimento, conforme evidenciado nas Figuras 16 e 17.

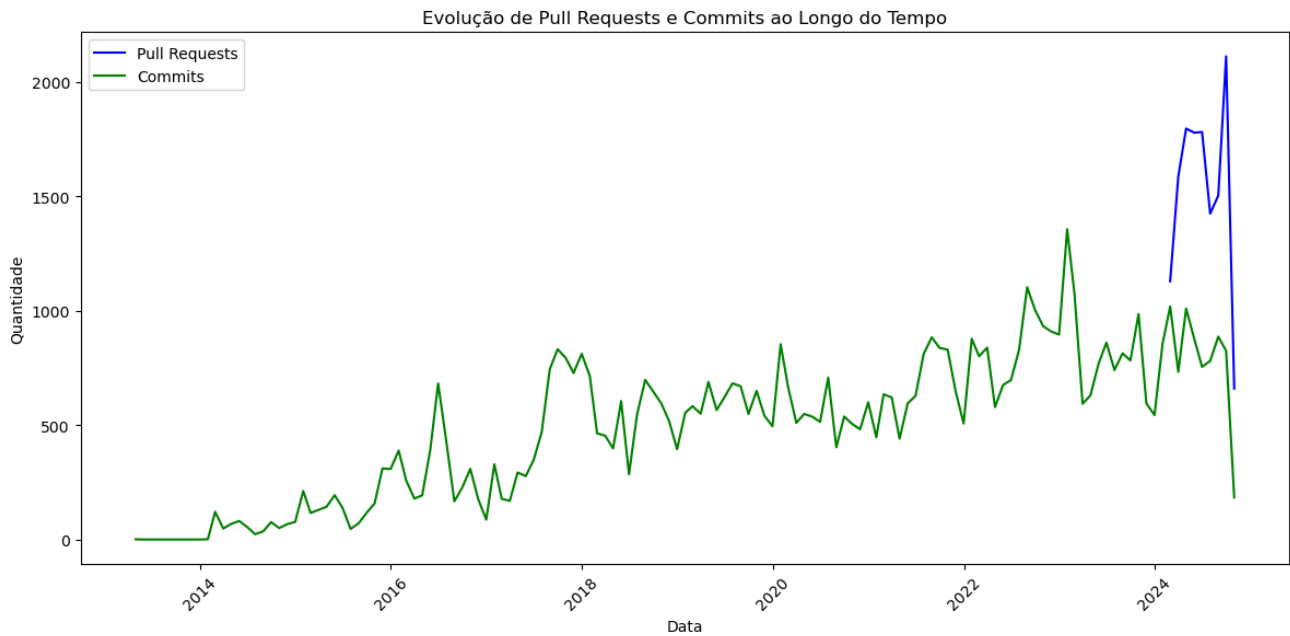


Figura 16: Comparação entre Commits e Pull Request ao Longo do Tempo no Repositório Godot

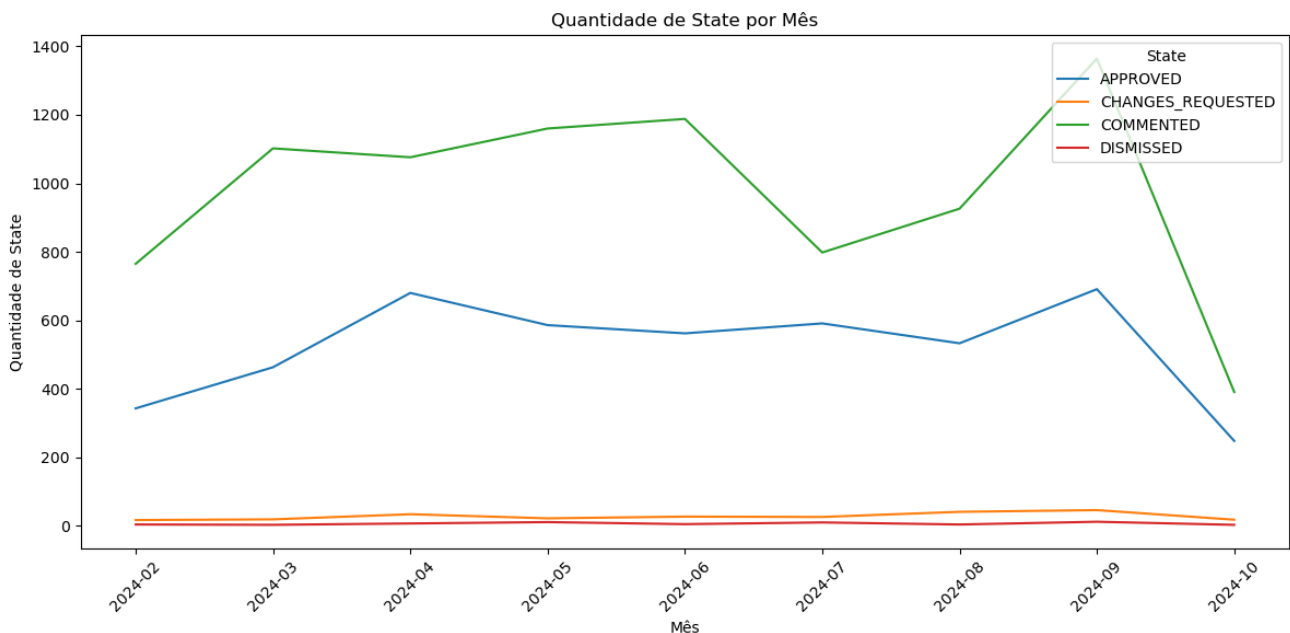


Figura 17: Quantidade de Pull Request classificados ao Longo do Tempo no Repositório Godot

3.7 Análise do Projeto IEducar

O padrão de atividade observado no projeto exibe uma variação significativa, com diversos picos de atividade, refletindo períodos de desenvolvimento intensivo intercalados com momentos de menor produtividade, como mostrado na Figura 19. Inicialmente, houve uma alta taxa de pull requests, mas logo ocorreu uma queda acentuada, estabilizando-se em um ritmo mais constante, conforme ilustrado na mesma figura. A distribuição das classificações dos pull requests revela um equilíbrio relativo entre os tipos **Commented**, **Approved** e **Changes Requested**, sem registros de **Dismissed**, como pode ser visto na Figura 20. Além disso, observou-se um declínio contínuo no número de contribuidores, membros e *owners*, embora os contribuidores tenham alcançado um pico elevado no início do período, conforme mostrado na Figura 21. Esses padrões inconstantes de atividade, com picos seguidos de períodos mais tranquilos, podem estar associados a fases intensas de desenvolvimento, como sprints ou lançamentos importantes, seguidos por momentos de menor atividade, como também evidenciado na Figura 19.

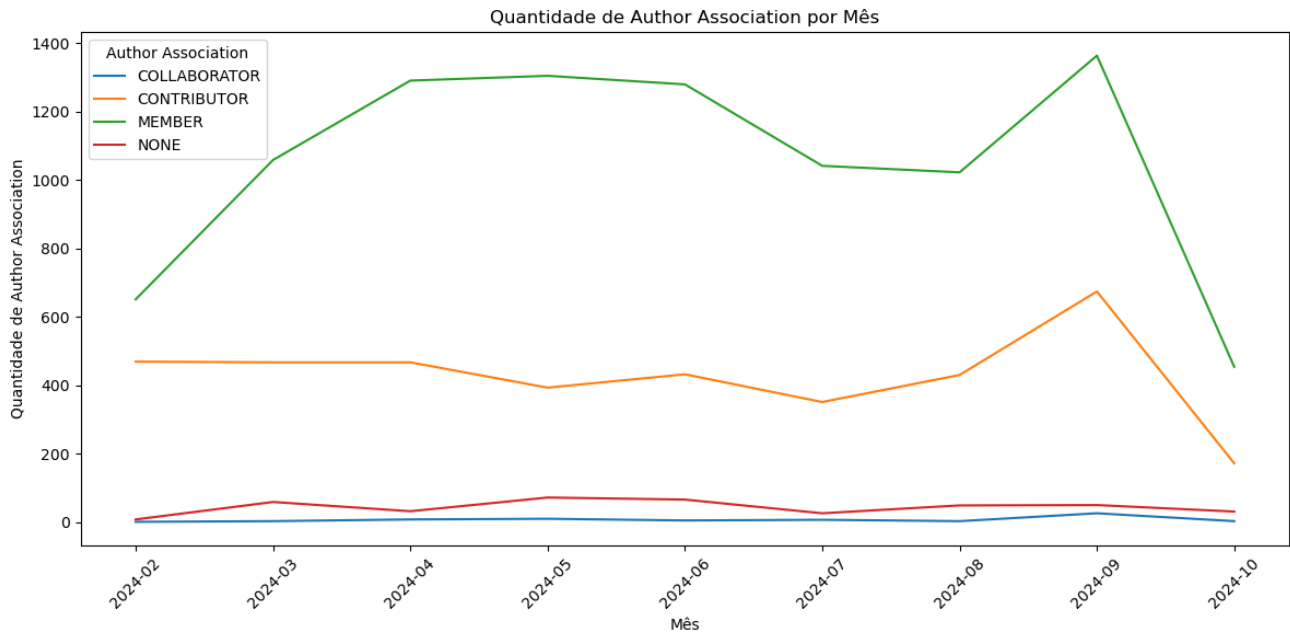


Figura 18: Quantidade de Pull Request com classificação de Autores ao Longo do Tempo no Repositório Godot

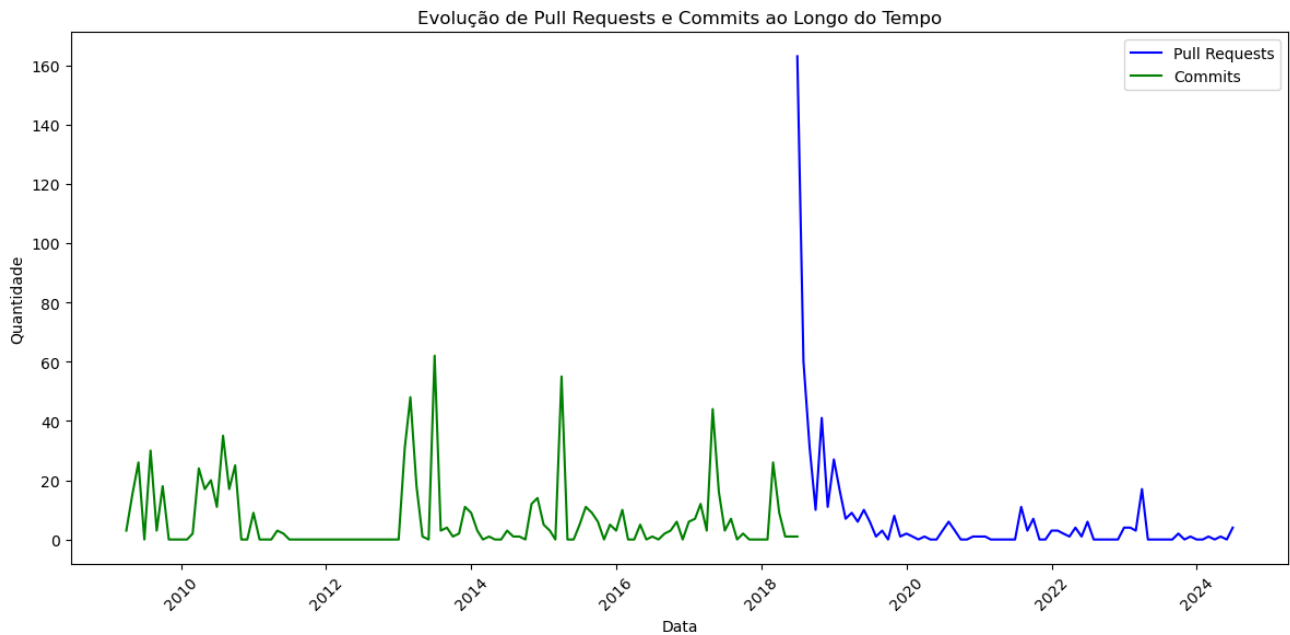


Figura 19: Comparação entre Commits e Pull Request ao Longo do Tempo no Repositório IEducar

3.8 Análise do Projeto Pandas

O projeto apresenta uma atividade consistente, com picos frequentes de contribuição, indicando que a comunidade ou equipe envolvida é ativa e realiza contribuições constantes, como mostrado na Figura 22. Observa-se uma queda contínua no número de pull requests **Commented**, enquanto o número de pull requests **Approved** se mantém constante, representando o segundo tipo mais frequente. As categorias **Changes Requested** e **Dismissed** também estão presentes, mas em menor número, conforme ilustrado na Figura 23. Além disso, nota-se uma redução contínua no número de membros ao longo do tempo, enquanto o número de contribuidores e *owners* permaneceu estável, como evidenciado na Figura 24. Essa estabilidade na atividade, com contribuições frequentes, sugere que o projeto conta com uma equipe ou comunidade robusta e engajada, o que sustenta o padrão constante de contribuição, conforme mostrado na Figura 22.

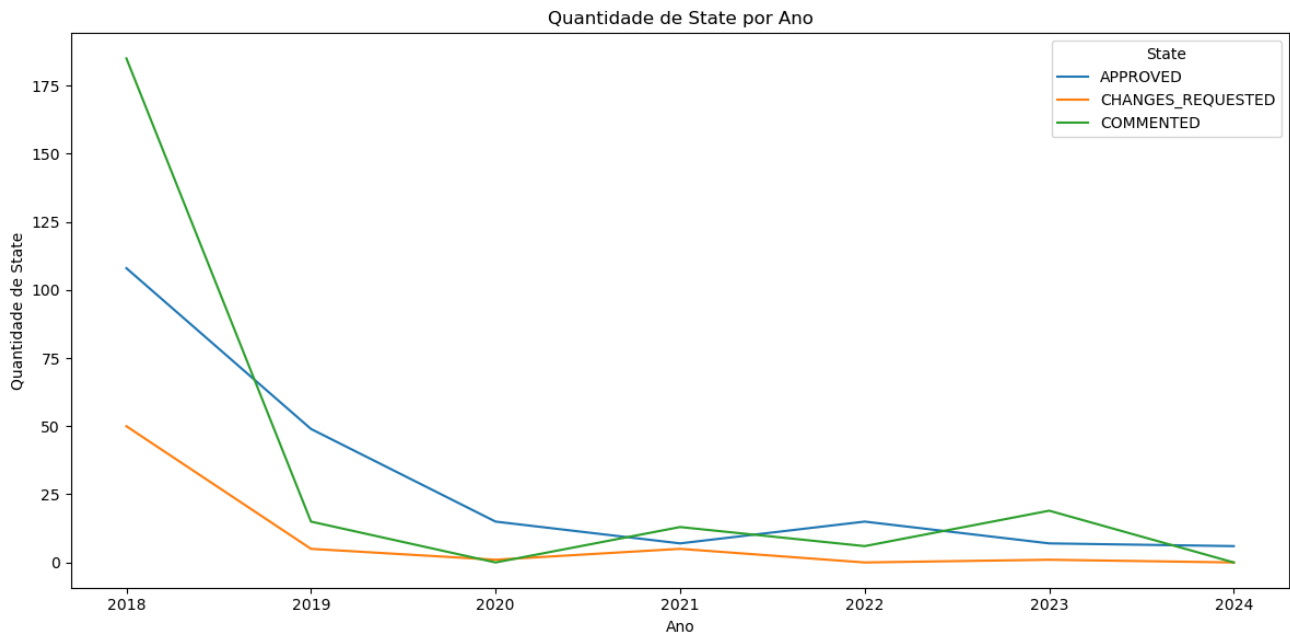


Figura 20: Quantidade de Pull Request classificados ao Longo do Tempo no Repositório IEducar

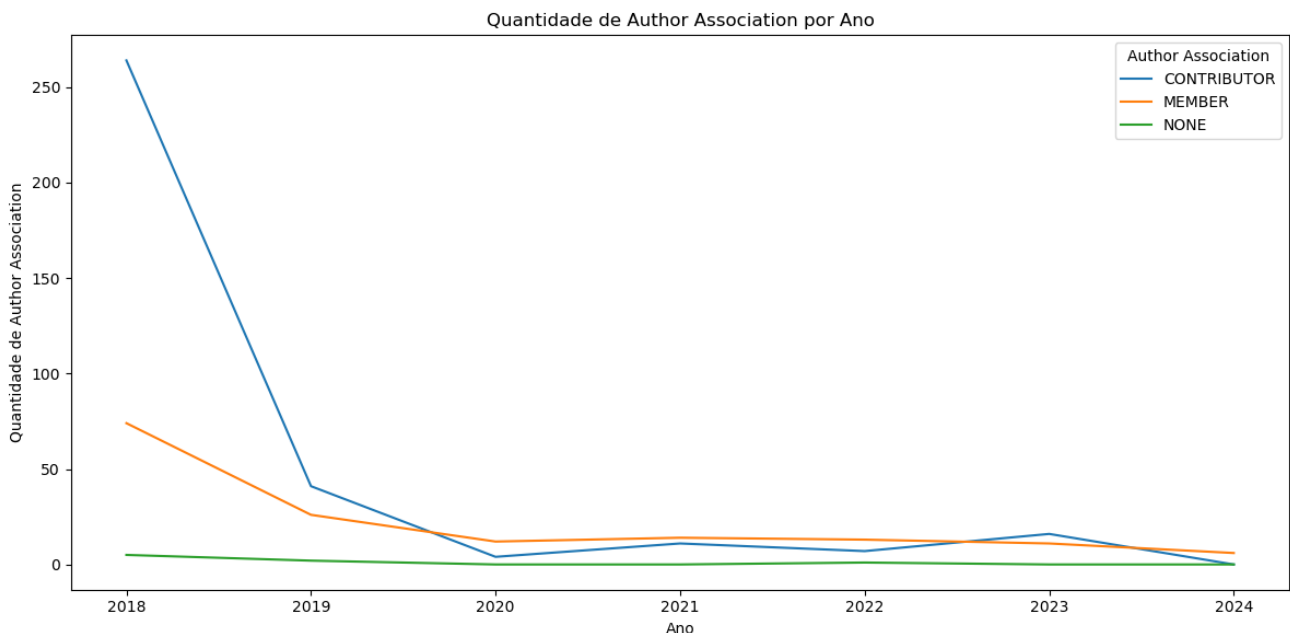


Figura 21: Quantidade de Pull Request com classificação de Autores ao Longo do Tempo no Repositório IEducar

3.9 Análise do Projeto Puter

O projeto apresenta uma atividade irregular de commits, com inúmeros picos e baixa consistência, como mostrado na Figura 25. Embora tenha havido uma alta atividade de pull requests no início, o ritmo de desenvolvimento diminuiu progressivamente ao longo do tempo. A constância de atividades entre os tipos de pull request **Commented**, **Approved** e **Changes Requested** foi semelhante, sem registros de pull requests **Dismissed**, conforme ilustrado na Figura 26. Inicialmente, o projeto contou com um número elevado de contribuidores, mas a presença de colaboradores e *owners* se manteve constante ao longo do tempo, conforme mostrado na Figura 27. Apesar de diversos picos de atividade, não há um padrão constante, o que sugere um desenvolvimento irregular, possivelmente influenciado por contribuições pontuais ou fases intensas de trabalho seguidas por longos períodos de inatividade, conforme evidenciado na Figura 25.

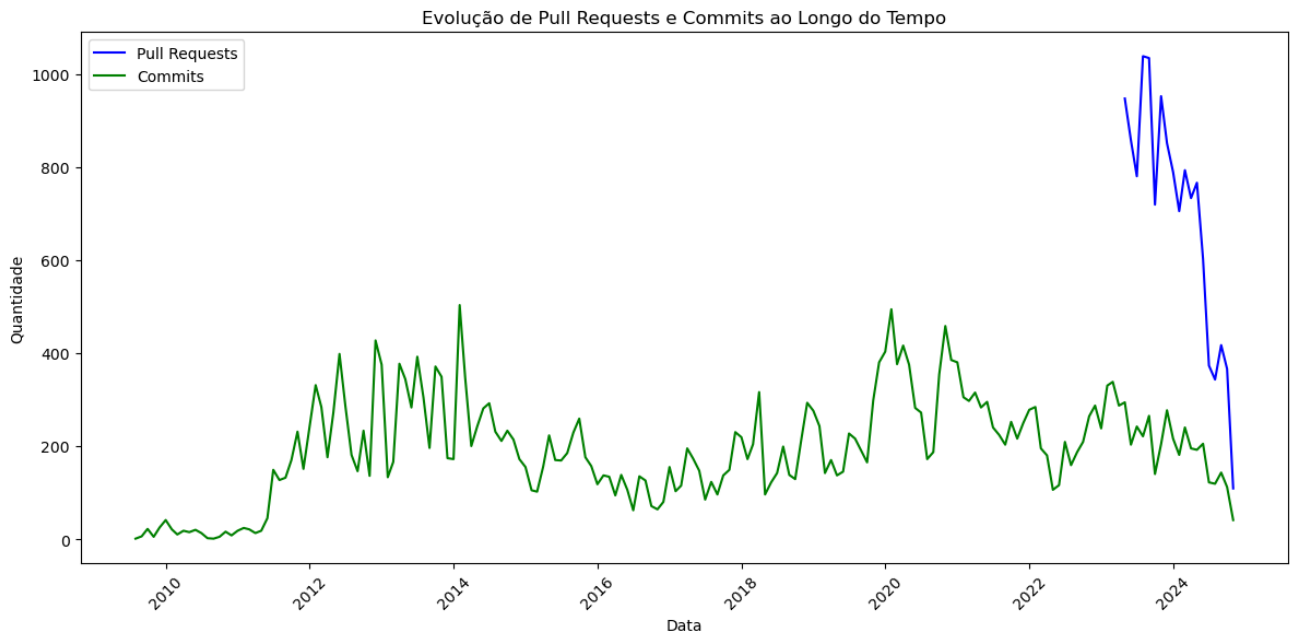


Figura 22: Comparação entre Commits e Pull Request ao Longo do Tempo no Repositório Pandas

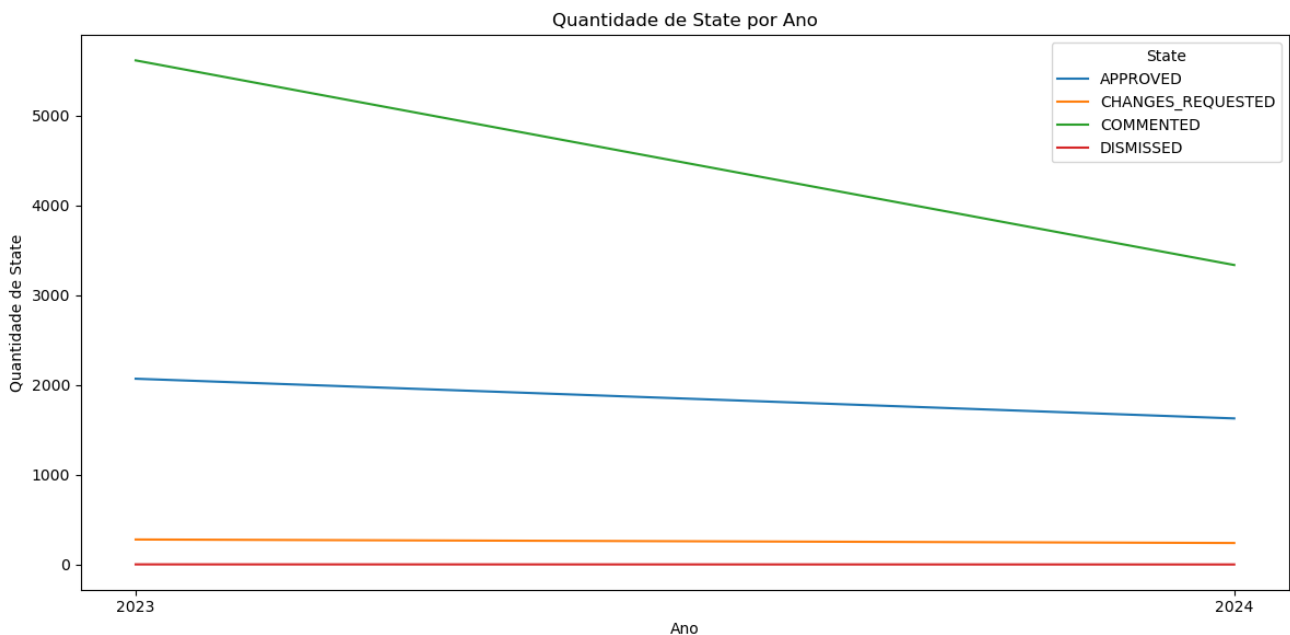


Figura 23: Quantidade de Pull Request classificados ao Longo do Tempo no Repositório Pandas

3.10 Análise do Projeto Reflex

O projeto apresenta um padrão constante de commits, com um pico significativo apenas ao final do período analisado, como mostrado na Figura 28. Inicialmente, a taxa de pull requests foi baixa, mas houve um crescimento contínuo, estabilizando-se em uma frequência mais regular após esse período de aumento, conforme ilustrado na Figura 28. A frequência dos tipos de pull requests **Commented** e **Approved** seguiu um padrão linear crescente ao longo do tempo. Por outro lado, os tipos **Dismissed** e **Changes Requested** mantiveram uma frequência relativamente estável, conforme demonstrado na Figura 29. Além disso, observou-se um crescimento constante no número de colaboradores, enquanto os contribuintes apresentaram um pico, que diminuiu em seguida. A atividade de membros e *owners* foi baixa e constante, conforme mostrado na Figura 30. De forma geral, o gráfico apresenta um padrão majoritariamente constante, com um pico acentuado no final. Este pico pode coincidir com uma grande release ou uma atualização

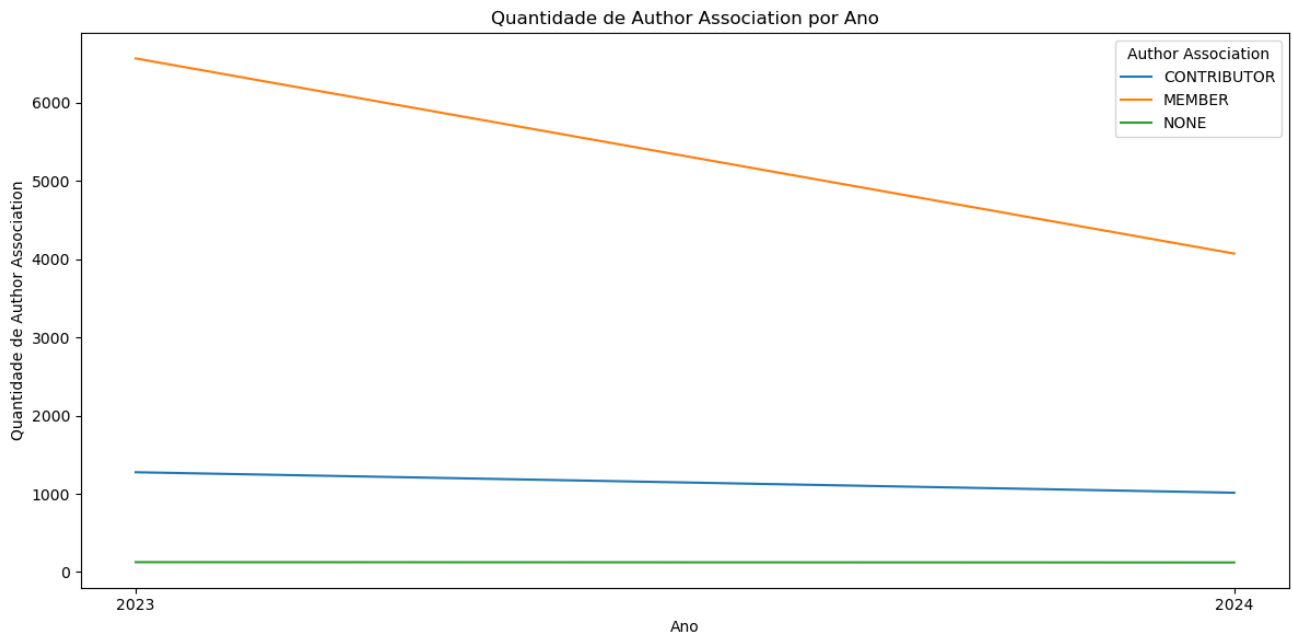


Figura 24: Quantidade de Pull Request com classificação de Autores ao Longo do Tempo no Repositório Pandas

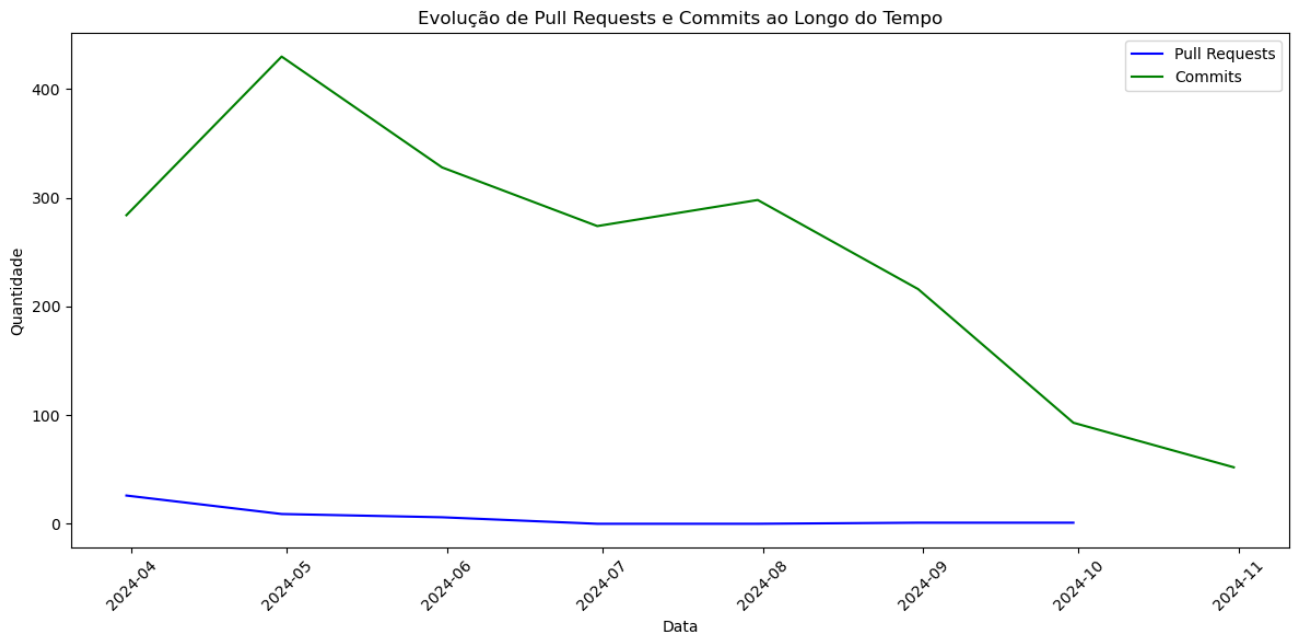


Figura 25: Comparação entre Commits e Pull Request ao Longo do Tempo no Repositório Puter

significativa, enquanto o comportamento constante ao longo do tempo sugere uma manutenção regular do projeto, conforme ilustrado na Figura 28.

4 Conclusão

O presente estudo investigou os fatores determinantes para a participação de colaboradores em projetos de código aberto, explorando como variáveis como popularidade dos projetos, número de contribuidores e contexto de desenvolvimento afetam o engajamento. A pesquisa identificou padrões de colaboração e barreiras que impactam diretamente a participação, fornecendo uma visão detalhada do comportamento dos contribuidores ao longo do tempo.

Projetos com alta popularidade, como Go e Pandas, apresentaram atividade de contribuição mais constante, o que sugere que a visibilidade e o reconhecimento do projeto influenciam

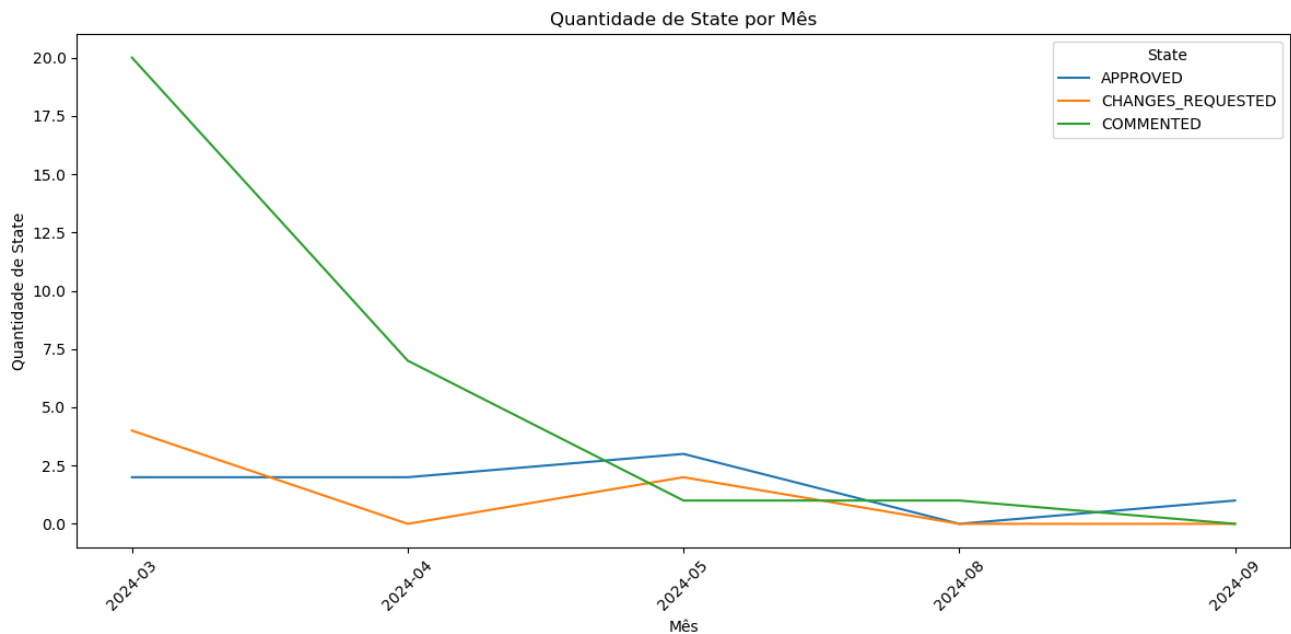


Figura 26: Quantidade de Pull Request classificados ao Longo do Tempo no Repositório Puter

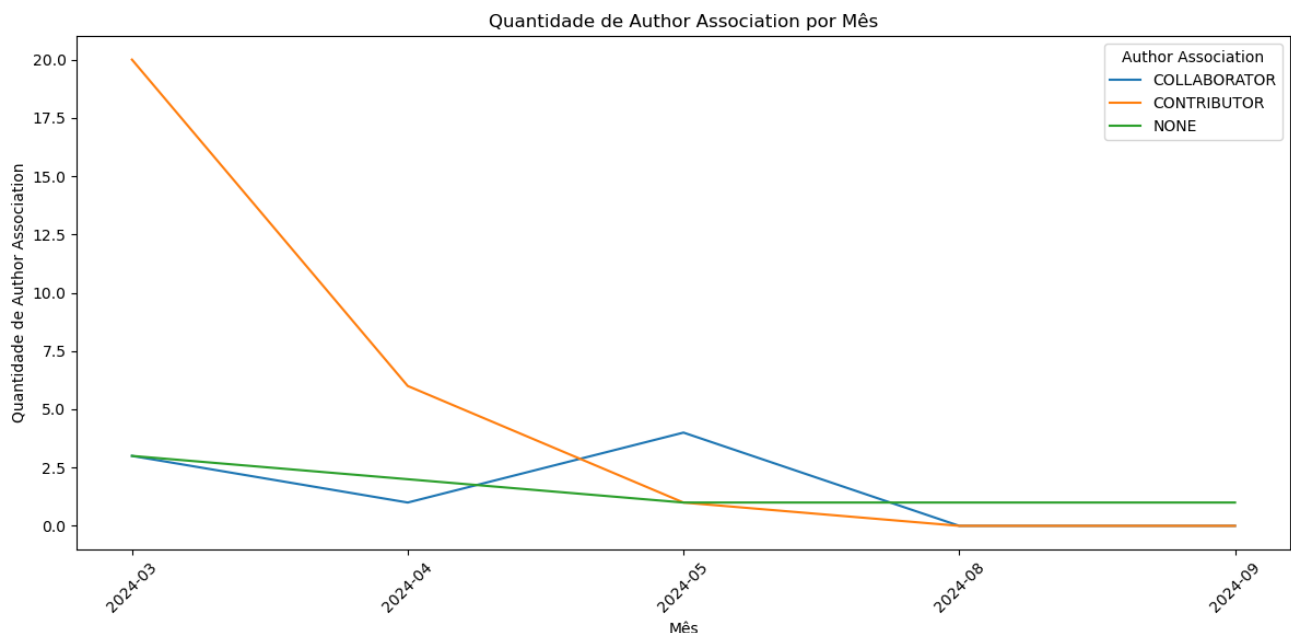


Figura 27: Quantidade de Pull Request com classificação de Autores ao Longo do Tempo no Repositório Puter

positivamente na frequência das contribuições. Em contrapartida, projetos como Docusaurus mostraram oscilações com picos de atividade específicos, possivelmente relacionados a grandes atualizações. Essa variação pode indicar que projetos menos conhecidos ou com atualizações esporádicas dependem de momentos específicos para captar maior engajamento.

Com base nos achados, recomenda-se que gerentes de projetos *open source* considerem a implementação de práticas de resposta ágil nas *pull requests*, além de uma moderação estruturada para incentivar feedback construtivo, especialmente para novos colaboradores. Tais práticas podem contribuir para a retenção e motivação dos desenvolvedores, tornando o processo de contribuição mais acessível e eficiente. Adotar uma estrutura de feedback simplificada e promover um ambiente acolhedor pode aumentar a continuidade nas contribuições e reforçar o engajamento da comunidade.

Entre as limitações deste estudo, destaca-se o foco em um número restrito de projetos e a utilização de ferramentas que, embora eficazes para a mineração de dados, oferecem restrições

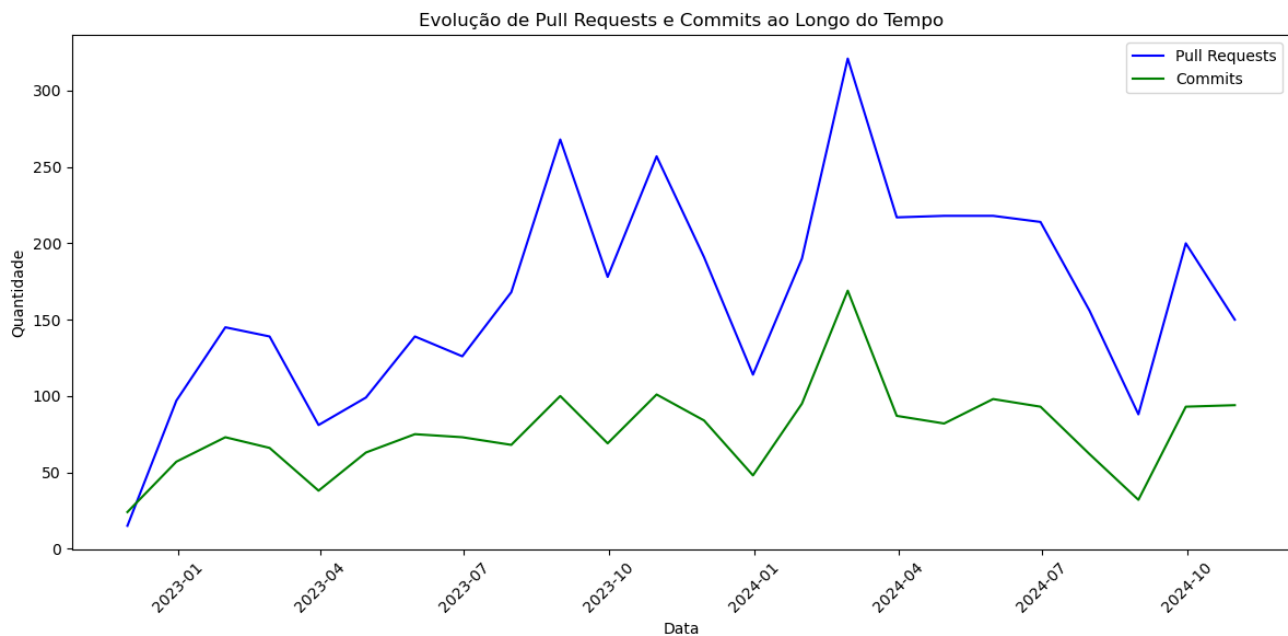


Figura 28: Comparação entre Commits e Pull Request ao Longo do Tempo no Repositório Reflex

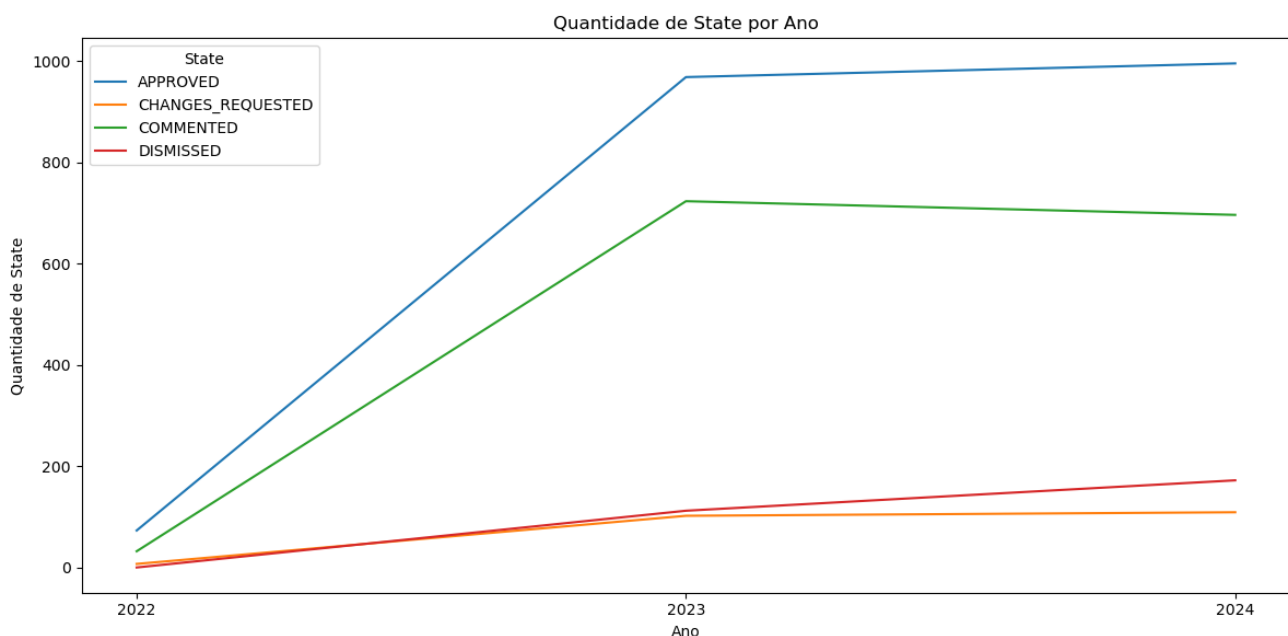


Figura 29: Quantidade de Pull Request classificados ao Longo do Tempo no Repositório Reflex

em termos de análise qualitativa. Trabalhos futuros poderiam explorar uma maior variedade de projetos ou incluir algoritmos de aprendizado de máquina para prever padrões de contribuição, bem como analisar o impacto de diferentes estruturas de governança na retenção dos colaboradores.

Em um cenário onde os projetos de código aberto se tornaram pilares da inovação tecnológica, entender e aprimorar os fatores que sustentam a participação contínua de colaboradores é essencial para o avanço dessas comunidades. Para promover a sustentabilidade dos projetos *open source*, é fundamental adotar estratégias que incentivem a participação contínua dos desenvolvedores e reforcem a cultura colaborativa, garantindo o sucesso e a evolução constante desses projetos.

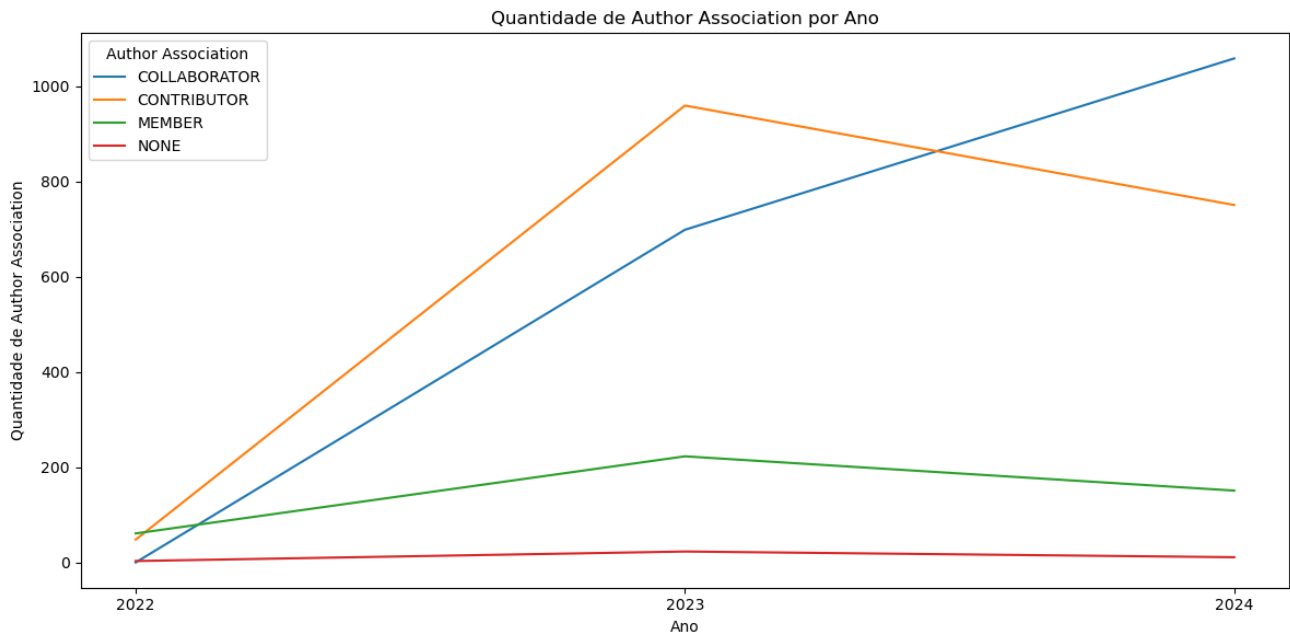


Figura 30: Quantidade de Pull Request com classificação de Autores ao Longo do Tempo no Repositório Reflex

Referências

- Zhifei Chen, Wanwangying Ma, Lin Chen, and Wei Song. Collaboration in software ecosystems: A study of work groups in open environment. *Information and Software Technology*, 145: 106849, 2022. doi: <https://doi.org/10.1016/j.infsof.2022.106849>.
- Kattiana Constantino, Shurui Zhou, Mauricio Souza, Eduardo Figueiredo, and Christian Kästner. Understanding collaborative software development: An interview study. In *2020 ACM/IEEE 15th International Conference on Global Software Engineering, ICGSE'20*, pages 55–65, 2020.
- Juan Pablo Puentes. Recommendations for the adoption of libre software in the public sector in south america. Master's thesis, Tilburg University, 2015.
- Igor Steinmacher, Marco Gerosa, Tayana U Conte, and David F Redmiles. Overcoming social barriers when contributing to open source software projects. *Computer Supported Cooperative Work (CSCW)*, 28:247–290, 2019.
- Yuriy Tymchuk, Andrea Mocci, and Michele Lanza. Collaboration in open-source projects: Myth or reality? In *Proceedings of the 11th Working Conference on Mining Software Repositories*, pages 304–307, 2014.