



UNIVERSIDADE FEDERAL DA BAHIA
FACULDADE DE FILOSOFIA E CIÊNCIAS HUMANAS
DEPARTAMENTO DE SOCIOLOGIA

GUSTAVO MENDEL QUEIROZ E SOUZA

**APRENDIZADO COM PROJETOS DE CÓDIGO ABERTO:
COMO A PARTICIPAÇÃO EM AMBIENTES DE
DESENVOLVIMENTO DE CÓDIGO ABERTO PODE CONTRIBUIR
PARA A FORMAÇÃO PROFISSIONAL DE UM ESTUDANTE DE
ENGENHARIA DE SOFTWARE**

Salvador

2022

GUSTAVO MENDEL QUEIROZ E SOUZA

**APRENDIZADO COM PROJETOS DE CÓDIGO ABERTO:
COMO A PARTICIPAÇÃO EM AMBIENTES DE
DESENVOLVIMENTO DE CÓDIGO ABERTO PODE CONTRIBUIR
PARA A FORMAÇÃO PROFISSIONAL DE UM ESTUDANTE DE
ENGENHARIA DE SOFTWARE**

Pré-projeto de pesquisa apresentado como pré-requisito para a conclusão da disciplina Metodologia e Expressão Técnico-científica.

Responsável
Prof. Dr. Felipe Padilha

Salvador

2022

Resumo

A indústria de Engenharia de Software vem sofrendo com a falta de profissionais devidamente qualificados desta área e a falta de atividades práticas e realistas nas disciplinas de Engenharia de Software pode ser a razão para isto. Já os Projetos de Código Aberto vêm se mostrando uma ótima alternativa para os estudantes e professores aproveitarem como método do aprendizado, já que estes projetos possuem um ótimo ambiente colaborativo formado por profissionais do mundo inteiro e conta também com um cenário totalmente realista para se praticar os conceitos da área. Neste âmbito, o principal objetivo deste trabalho é buscar entender como a participação do estudante em Projetos de Código Aberto pode contribuir para a formação profissional como engenheiro de software.

Palavras-chave

Educação da Computação, Projetos Open Source, Ambientes Colaborativos, Categorização de Projeto Código Aberto.

Sumário

1. Introdução	1
1.1 Motivação	2
1.2 Objetivos	3
2. Ambientes Colaborativos	4
2.1 Social Coding.....	5
2.2 Engenharia de Software	6
3. Metodologia.....	8
3.1 Cronograma	10
4. Referências Bibliográficas	11

1. Introdução

Projetos de código aberto (Open Source Software - OSS), também conhecidos como Software Livre (Free Software ou Libre Software), são aqueles softwares que possuem seu código fonte aberto para todos os usuários, respeitando a liberdade e o senso de comunidade (GNU, 2022). Basicamente, permite aos usuários executar, copiar, distribuir, estudar, realizar modificação e melhorar o software, sendo assim, a definição de Software Livre diz respeito à liberdade e não ao preço, confusão bastante comum motivada pela nomenclatura original em inglês, Free Software, que pode ser interpretada como Software de graça em vez de Software livre.

Sendo assim, o ambiente de desenvolvimento desse tipo de software tende a ser muito colaborativo entre a comunidade, chegando a ter centenas de colaboradores de diversas nacionalidades (ROCHA, 2007). Geralmente, esses ambientes são baseados em plataformas com soluções já prontas e pensadas para projetos de cunho colaborativo, como exemplo o GitHub, adquirida pela Microsoft em 2018, e a plataforma GitLab (que por si só é uma plataforma de código aberto), popular entre projetos Open Source bastante conhecidos como GNOME e KDE.

Logo, os requerimentos para a participação e colaboração desse tipo de ambiente de desenvolvimento tendem a ser muito relacionados com experiência real dos desenvolvedores em projetos similares ou pelo menos requer que eles se adaptem ao longo do tempo para satisfazer tais requerimentos. Contrastando-se, portanto, com a falta de profissionais engenheiros de software devidamente capacitados para o mercado de trabalho, previsto por Meyer (2001) e melhor desenvolvido posteriormente neste projeto.

Este projeto de pesquisa pretende entender como a participação e a colaboração em ambientes de desenvolvimento de projetos de código aberto podem impactar na carreira do estudante de Engenharia de Software, preenchendo a deficiência que os cursos da área têm em não preparar o estudante adequadamente para os projetos mais realísticos que recorrentemente irão participar ao se formarem profissionalmente.

1.1 Motivação

Os projetos de código aberto hoje em dia já estão presentes no dia a dia de todos nós. Por exemplos, como citado por Crowston, Wei, Howison e Siggins em 2008, o Kernel Linux é certamente o maior exemplo de Código Aberto atualmente, e provavelmente o software mais importante do mundo. O Linux está sendo executado em uma quantidade incontável de aparelhos, como dispositivos celulares a base de Android, servidores em grandes e pequenos Data Center, Supercomputadores, e mais recentemente, dispositivos Desktop.

Além disso, o servidor de internet Apache é o mais famoso e mais usado servidor web do mundo, e sua fundação Apache Software Foundation é a maior fundação de software aberto do mundo (ASF, 2022), assim como o navegador de Internet Mozilla Firefox, sob a Fundação Mozilla, é um dos principais softwares de navegação na internet atualmente e sua fundação vem liderando a política de Código Aberto em muitos âmbitos políticos atualmente.

Além dos grandes exemplos de projetos que carregam a filosofia de Código Aberto, temos inúmeros projetos menores recém-criados e que estão em constante crescimento, como o projeto de interface visual para ambientes desktop GNOME (ROCHA, 2007). E existem também os projetos muito antigos e pouco conhecidos, mas que ainda assim possuem até hoje uma base de usuários fiéis e motivados para uma constante manutenção e aprimoramento do código existente, como exemplo: VIM, GIMP e XML (CROWSTON, K.; WEI, K.; HOWISON, J.; WIGGINS, A, 2008).

Assim sendo, a comunidade e o ambiente de desenvolvimento de projetos de código aberto são gigantes e possuem como senso de comunidade e colaboração um dos principais pilares da filosofia de Código Aberto.

Além disso, espera-se que no decorrer das disciplinas de ES o estudante adquira e/ou aperfeiçoe por meio de experiência prática as habilidades de comunicação, trabalho em equipe, ética profissional, solução de problemas, além de habilidades técnicas adquiridas durante o desenvolvimento do Software (ACM AND IEEE, 2015). Portanto, é perceptível que a falta de vivência em projetos reais vem implicando em profissionais desta área pouco qualificados para o mercado de trabalho, tratando-se de uma profissão que requer tanta prática em seu currículo (ACM AND IEEE, 2015; ACM AND IEEE, 2013), e que pode, de alguma maneira, ser suprida pela colaboração em projetos de Código Aberto da parte do estudante. Justificando, portanto, a realização desse estudo.

1.2 Objetivos

Este trabalho conta com o seguinte objetivo geral:

- listar motivos para o estudante de ES do porquê ele deve (ou não) colaborar com projetos de código aberto durante o seu período de estudos, visando contribuir para a sua carreira profissional como engenheiro de software.

Além disso, este trabalho conta com 3 objetivos específicos, sendo eles:

- investigar como os principais pilares da filosofia de código aberto influenciam no senso de colaboração e comunidade das comunidades de código aberto;
- analisar como as soluções de plataformas de colaboração de código, como GitHub e GitLab, colaboram para o chamado Codificação Social (Social Coding, em inglês);
- revisar principais ementas da disciplina de ES, buscando os tópicos e habilidades esperadas do estudante da disciplina após a conclusão do curso.

2. Ambientes Colaborativos

Para entender melhor o fluxo de um ambiente colaborativo de desenvolvimento de software, primeiro analisa-se o ambiente de colaboração da Wikipedia, uma enciclopédia online gratuita criada e editada por voluntários ao redor do mundo (WIKIPEDIA, 2022), que possui filosofia de contribuição de conteúdo bastante relacionada com a contribuição de código, o objeto desse estudo.

A comunidade da Wikipedia segue uma abordagem bastante acolhedora para com os recém-chegados, como Bryant (2005) traz à tona, em síntese, os recém-chegados aprendem as convenções e regras de contribuição da comunidade da Wikipédia através da observação e orientação direta de usuários mais experientes. Relacionado a este conceito de mentoria está uma norma comunitária de "não morda os recém-chegados". A natureza das contribuições dos usuários também tende a mudar à medida que os recém-chegados se tornam mais socializados, desde a simples edição de artigos até a participação em discussões da comunidade, tarefas administrativas e tarefas "meta".

Para Dabbish, Stuart, Tsay e Herbsleb (2012), essa é uma abordagem bastante acolhedora das empresas e fundações visando a retenção dos recém-chegados em seus ambientes, para que no futuro se tornem bastante capacitados e colaborarem com o desenvolvimento dos seus projetos. Contudo, quando se trata do escopo de projetos de código aberto, não existirá esse estágio de mentoria tão cuidadoso para tratar dos recém-chegados. Grande parte das vezes, as comunidades de desenvolvimento já vão exigir certo conhecimento e capacidade de colaboração dos recém-chegados que estão propondo alguma modificação no código existente.

Ainda assim, a partir do momento que o desenvolvedor demonstra que tem mínima capacidade para estar ali no ambiente de desenvolvimento, de acordo com Crowston, Wei, Howison e Wiggins (2008), ele se vê na possibilidade de se aprofundar mais naquele projeto, participando de mais colaborações, estudando todas as características arquiteturais daquele software, se aventurando mais em áreas que ele não era familiarizado antes, como áreas de testes e padronização de código, e principalmente, fica exposto à revisões de outros desenvolvedores mais dizendo onde ele errou, onde acertou e onde pode melhorar.

2.1 Social Coding

Social Coding é um termo que remete ao desenvolvimento de código em um ambiente semelhante às redes sociais, em que é possível visualizar os códigos dos outros, colaborações, comentários, sugestões e demais atividades, como seguidores, curtidas e mais recentemente, apoiar demais colaboradores e projetos através de contribuições financeiras.

A principal plataforma atualmente que possui as soluções de mídias sociais integradas com o desenvolvimento de código certamente é o GitHub. No atual mês de novembro de 2022, a plataforma possui quase 100 milhões de desenvolvedores registrados e mais de 50 milhões de repositórios públicos, através do mecanismo de busca do GitHub (GITHUB, 2022).

Dessa maneira, o GitHub se tornou não só uma plataforma de hospedagem de código, mas também como uma espécie de rede social de desenvolvedores, em que outros desenvolvedores podem acessar o perfil de outro desenvolvedor e checar: as recentes contribuições dele, quais projetos ele esteve acompanhando, quais discussões ele esteve participando, além de todo o código produzido por aquele desenvolvedor postado no GitHub.

Como cita Tsay, Dabbish e Herbsleb (2014):

A natureza transparente do GitHub também levou os desenvolvedores a se conscientizarem de que suas ações de trabalho tinham um público. Essas pressões do público encorajaram os desenvolvedores a criar contribuições mais legíveis com mensagens de commit “rápidas” e código limpo, porque os usuários potencialmente usariam posteriormente o histórico de commit para fazer inferências como intenção e competência do desenvolvedor.

Tendo isso em vista, presume-se que a participação em colaborações de projetos de código aberto pode não só contribuir para as habilidades técnicas do desenvolvedor, mas também contribuir para a sua carreira profissional em um âmbito de networking. Já que ele irá passar a conhecer mais pessoas através do Social Coding, além de que ele terá uma espécie de portfólio montado em seu GitHub, com as suas colaborações servindo como prova de sua competência em codificar programas.

2.2 Engenharia de Software

A área de ES é um tanto diferente das demais engenharias, já que aqui o produto é o software, algo intangível. Destaca-se três das suas principais características: intangibilidade, complexidade e importância. Tendo isso em vista, o engenheiro de software precisa lidar também com a possibilidade de modificação constante durante o seu ciclo de vida.

A ES vem lidando com o problema da falta de capacitação de profissionais no mercado de trabalho, visto que grande parte dos engenheiros de software ainda são originários de cursos de ciências da computação e não necessariamente de cursos de engenharia de software. Nascimento (2017) vai trazer à tona que tais estudantes possuem conhecimento de projetos de compiladores, sistemas operacionais, inteligência artificial, dentre outros tópicos de interesse das ciências da computação, porém com pouca experiência em trabalhar em equipe, testes e práticas comuns do mercado, como controle de versão e gerenciamento de pacotes.

Logo, é neste vão que a abordagem utilizando projetos de código aberto entra como uma forma de suprir essa deficiência dos cursos de ciência da computação para com os engenheiros de software. Fazendo ligação com o Social Coding, Smith et al. (2014) acreditam que os desenvolvedores que participam dos projetos de código aberto possuem maior comprometimento em aplicar os princípios da engenharia de software, consequentemente produzir projetos cuidadosamente concebidos, construídos e mantidos, já que a maioria deles participa dos projetos código aberto para gerar reputação, autodesenvolvimento e altruísmo, como visto no Social Coding.

É sabido que projetos de código aberto possuem características que quase nenhum ambiente acadêmico pode prover para os estudantes e que são similares às encontradas na indústria: projetos de grande porte, ciclo de vida de décadas, requisitos de qualidade restritos, arquitetura complexa, usuários reais demandando correções em tempo real de desenvolvimento, entre outros (NASCIMENTO, 2017). Portanto, pressupõe-se que a colaboração e a participação do desenvolvedor em projetos de código aberto contribuem para a sua carreira profissional no sentido técnico, de vivência prática em projetos reais, ao mesmo tempo em que o colaborador desenvolve suas capacidades como engenheiro de software.

Entretanto, sabe-se também que nem todos os ambientes de colaboração de projetos de código aberto podem ser benéficos para o estudante, tendo em vista que em alguns deles, pode-se achar uma comunidade um tanto hostil e não tão receptiva como foi visto anteriormente. Este estudo busca também trazer os lados negativos à tona e recomendar, através da literatura obtida, projetos modelos e com uma ótima comunidade de desenvolvimento para o estudante, além de

maneiras de como lidar com projetos potencialmente maléficos para a carreira profissional dele, desde lidar com projetos de cunho ético duvidoso e que pode não ser bem-visto pelo mercado profissional até projetos em que não possuem uma qualidade de software assegurada.

3. Metodologia

O estudo se dará principalmente pela revisão da literatura e por levantamento de dados através dos métodos descritos a seguir.

Será realizado um levantamento bibliográfico e montagem da literatura produzida sobre o tema de artigos científicos clássicos e atuais, através da *internet* por meio de revistas científicas digitais. Este levantamento continuará até o final do trabalho.

Além disso, será feito uma listagem de repositórios de projetos de código aberto recomendados para o ensino da engenharia de software através do levantamento bibliográfico de autores consagrados nessa área e suas obras, como Smith (2014) e Meyer (2001), além de recomendações de projetos mais atuais feitos pela comunidade código aberto.

Posteriormente, será analisado os principais meios de comunicação, das ferramentas de socialização e colaboração, a fim de obter o histórico de contribuições dos projetos listados no passo anterior, relacionando-o com o Social Coding citado anteriormente. Isso se dará a partir da listagem *Issues*, *Pull Requests*, *Discussions* entre outras ferramentas das plataformas de colaboração, exemplo GitHub. Tendo em vista que os projetos são de código aberto, esses dados são públicos nas páginas dos respectivos repositórios, que possui ótimas ferramentas de pesquisa e filtragem para facilitar a listagem dos itens citados acima.

Também, serão listadas as características arquiteturais dos projetos, assim como as técnicas de engenharia de software usadas naquele projeto, utilizando como base os arquivos de documentação de cada projeto e através de uma análise do código em si, com base nas referências bibliográficas clássicas sobre o tema engenharia de software.

Em seguida, será realizado uma busca nas literaturas clássicas e atuais sobre os cursos de engenharia de software, visando obter as habilidades e capacidades que um engenheiro de software é suposto a ter após a finalização dos cursos. Para complementar, serão procurados os mesmos dados, porém em emendas de cursos de diversas faculdades e universidades sobre ciências da computação ou engenharia de software.

Logo após, será feito uma revisão das vagas mais recentes de engenharia de software em diversas empresas. Em partículas, aquelas mais conhecidas pelo ramo, como Google, Amazon, Facebook, além de outras vagas nacionais aleatórias selecionadas pelo LinkedIn, plataforma de mídia social voltada para os profissionais e divulgação de vagas de emprego. É possível, portanto, listar os requisitos que as empresas estão cobrando para os engenheiros de software,

sendo assim um norte para relacionar os projetos de código aberto com a atual demanda da indústria de software.

Após a coleta de todos os dados citados anteriormente, será realizado uma análise entre os resultados obtidos buscando listar as relações entre os projetos de código aberto analisados e os cursos de engenharia de software, além do mercado de trabalho e o que ele está esperando do profissional de engenharia de software atualmente, tendo como base as referências bibliográficas clássicas sobre o tema.

Ao término do estudo, serão elaborados os artigos para divulgação através da submissão em eventos e periódicos científicos relacionados ao tema, além de uma divulgação via universidades e cursos de engenharia de software visando o alcance aos estudantes e engenheiros de software, que são os impactados pelo resultado da pesquisa.

3.1 Cronograma

A seguir, é apresentada uma proposta para o cronograma de atividades a ser seguida no desenvolvimento do trabalho.

Ano 2023												
Etapas	Mês 1	Mês 2	Mês 3	Mês 4	Mês 5	Mês 6	Mês 7	Mês 8	Mês 9	Mês 10	Mês 11	Mês 12
1												
2												
3												
4												
5												
6												
7												
8												

Figura 1 - Cronograma

Etapas:

1. Coleta de material bibliográfico.
2. Levantamento de projetos de código aberto modelo para serem ensinados em engenharia de software, através do levantamento literário do tema.
3. Coleta de dados das plataformas de colaboração dos devidos projetos, através das *Issues*, *Pull Requests*, *Discussions*.
4. Busca detalhada das emendas de engenharia de software através de universidades e levantamento bibliográfico do tema.
5. Levantamento dos requisitos para as vagas de engenharia de software através da plataforma de socialização profissional LinkedIn.
6. Estudo, a partir da literatura obtida do tema, o relacionamento entre os dados obtidos até então.
7. Escrita do artigo final levando em consideração as conclusões obtidas durante a pesquisa.
8. Divulgação do resultado desde estudo em eventos e congressos científicos da área, assim como entre os estudantes de universidades dos cursos relacionados com engenharia de software.

Referências Bibliográficas

ACM AND IEEE. *Joint Task Force on Computing Curricula. Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. [S.l.], 2013. 514 p.

ACM AND IEEE. *Joint Task Force on Computing Curricula. Software Engineering 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*. [S.l.], 2015. 134 p.

BRYANT, S. L., FORTE, A. AND BRUCKMAN, A. *Becoming Wikipedian: transformation of participation in a collaborative online encyclopedia*. Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work (New York, NY, USA), 2005.

CROWSTON, K.; WEI, K.; HOWISON, J.; WIGGINS, A. *Free/Libre open-source software development: What we know and what we do not know*. ACM Comput. Surv. 44, 2 (Mar. 2008).

DABBISH, L., STUART, C., TSAY, J. AND HERBSLEB, J. *Social coding in GitHub: transparency and collaboration in an open software repository*. Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work (New York, NY, USA). 2012.

LUTZ, M. J.; NAVEDA, J. F.; VALLINO, J. R. *Undergraduate Software Engineering: Addressing the Needs of Professional Software Development*. Queue, ACM, v. 12, n. 6, p. 30, jun 2014. ISSN 1542-7730.

MEYER, B. *Software engineering in the academy*. Computer, IEEE Computer Society Press, v. 34, n. 5, p. 28–35, 2001.

NASCIMENTO, Debora Maria Coelho. *Educação em Engenharia de Software com a adoção de projetos de código aberto: uma análise detalhada*. 2017. 396p.: il. Tese de Doutorado - Universidade Federal da Bahia, Salvador, 2017.

ROCHA, L.A. *Software livre e produção colaborativa na Internet: um estudo de caso dos instrumentos de comunicação do Projeto GNOME*. 2007. 80 f. : il. Dissertação de Pós-graduação - Universidade Federal da Bahia, Salvador, 2007.

TSAY, J., DABBISH, L., HERBSLEB, J. *Influence of social and technical factors for evaluating contribution in GitHub*. Proceedings of the 36th international conference on Software engineering, 2014.

SMITH, T. M. et al. *Selecting open source software projects to teach software engineering*. SIGCSE '14 Proceedings of the 45th ACM technical symposium on Computer science education. [S.l.: s.n.], 2014.

Foundation Apache Project. Disponível em: <<https://www.apache.org/foundation>>. Acesso em: 30 de out. de 2022.

GitHub. Disponível em: <<https://github.com>>. Acesso em: 29 de out. de 2022.

GNOME GitLab homepage. Disponível em: < <https://gitlab.gnome.org/GNOME>>. Acesso em: 29 de out. de 2022.

What is Free Software? GNU Operating System. 2022. Disponível em: <<https://www.gnu.org/philosophy/free-sw.html>>. Acesso em: 28 de set. de 2022.

Wikipedia. Disponível em: <<https://www.wikipedia.org>>. Acesso em: 30 de out. de 2022.