



UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIENCIAS EXATAS E APLICADAS
DEPARTAMENTO DE COMPUTAÇÃO E SISTEMAS

WALACE MARTINS SANTOS JUNIOR

**ESTUDO DE CASO: INTRODUÇÃO À COMUNIDADE E DESENVOLVIMENTO
COLABORATIVO DA FERRAMENTA DE SOFTWARE LIVRE FREEDOMBOX**

João Monlevade

2022

WALACE MARTINS SANTOS JUNIOR

**Estudo de Caso: Introdução à Comunidade e Desenvolvimento
Colaborativo da Ferramenta de Software Livre FreedomBox**

Monografia apresentada ao curso de Engenharia de Computação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Orientador: Igor Muzetti Pereira

João Monlevade

2022



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS
DEPARTAMENTO DE COMPUTAÇÃO E SISTEMAS



FOLHA DE APROVAÇÃO

Wallace Martins Santos Júnior

Estudo de Caso: Introdução à comunidade e desenvolvimento colaborativo da ferramenta de software
livre FreedomBox

Monografia apresentada ao Curso de Engenharia de Computação da Universidade Federal
de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Engenharia de Computação

Aprovada em 4 de Novembro de 2022

Membros da banca

Mestre Igor Muzetti Pereira - Orientador (Universidade Federal de Ouro Preto)
Mestre Euler Horta Marinho (Universidade Federal de Ouro Preto)
Doutor Diego Zuquim Guimarães Garcia (Universidade Federal de Ouro Preto)

Igor Muzetti Pereira, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital
de Trabalhos de Conclusão de Curso da UFOP em 17/11/2022



Documento assinado eletronicamente por **Igor Muzetti Pereira, PROFESSOR DE MAGISTERIO SUPERIOR**, em 17/11/2022, às 19:39, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0427759** e o código CRC **309A6D4C**.

Referência: Caso responda este documento, indicar expressamente o Processo nº 23109.015519/2022-01

SEI nº 0427759

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: (31)3808-0819 - www.ufop.br

RESUMO

Este trabalho tem por objetivo apresentar um estudo de caso etnográfico sobre uma comunidade colaborativa de software livre através da participação no projeto FreedomBox. Com a consolidação de diversos projetos de código aberto e livre, as práticas abordadas em seu ecossistema têm se tornado um interesse de estudo, visando utilização de seus métodos no desenvolvimento de software, de forma benéfica. O projeto FreedomBox, lançado em 2010, possui uma comunidade colaborativa onde este estudo é aplicado. Com a participação ativa na comunidade e metodologia etnográfica aplicada a Engenharia de Software, pôde-se traçar um perfil comportamental da estrutura colaborativa, práticas utilizadas e instigar o ingresso de novos colaboradores em projetos FLOSS, além de discutir as dificuldades vivenciadas. Como membro da comunidade, foi possível contribuir com o desenvolvimento através de atividades vivenciando o fluxo utilizado pela mesma, obtendo como resultado um passo de iniciação à uma comunidade FLOSS, uma perspectiva das dificuldades enfrentadas no primeiro contato, sendo uma delas a barreira imposta pela comunicação interna, bem como a utilidade de um filtro bem refinado acerca do projeto a se ingressar.

Palavras-chave: Software livre, Código aberto, FLOSS, FreedomBox.

ABSTRACT

This work aims to present an ethnographic case study about a collaborative free software community through participation in the FreedomBox project. With the consolidation of several open and free source projects, the practices addressed in its ecosystem have become an interest of study, aiming to use its methods in software development, in a beneficial way. The FreedomBox project, launched in 2010, has a collaborative community where this study is applied. With active participation in the community and ethnographic methodology applied to software engineering, it was possible to trace a behavioral profile of the collaborative structure, practices used and instigate the entry of new collaborators in FLOSS projects, in addition to discussing the difficulties experienced. As a member of the community, it was possible to contribute to the development through activities experiencing the flow used by it, resulting in an initiation step to a FLOSS community, a perspective of the difficulties faced in the first contact, one of them being the barrier imposed by communication. internal, as well as the usefulness of a well-refined filter about the project to be entered.

Keywords: Free Software, Open Source Software, FLOSS, FreedomBox.

LISTA DE ABREVIATURAS

FLOSS	–	FREE/LIBRE OPEN SOURCE SOFTWARE
S.O	–	SISTEMA OPERACIONAL
M.R	–	MERGE REQUEST
GQM	–	GOAL QUESTION METRIC

SUMÁRIO

1 INTRODUÇÃO	14
1.1 PROBLEMA	15
1.2 OBJETIVOS	15
1.2.1 Objetivo geral	15
1.2.2 Objetivos específicos.....	16
1.3 JUSTIFICATIVA	16
1.4 ESTRUTURA DO TRABALHO.....	16
2 CONCEITOS GERAIS E REVISÃO DA LITERATURA.....	17
2.1 ESTUDO ETNOGRÁFICO	18
2.2 OBSERVAÇÃO PARTICIPANTE	18
2.3 DEBIAN.....	19
2.4 COMUNIDADES FLOSS.....	19
2.5 COMUNIDADE FREEDOMBOX	20
3 METODOLOGIA	22
3.1 ESTRATÉGIA DE PESQUISA	22
3.2 DESENVOLVIMENTO.....	22
3.3 CONTRIBUIÇÕES	25
3.4 PARTICIPAÇÃO DA COMUNIDADE	29
4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS.....	30
5 CONCLUSÕES	36
REFERÊNCIAS.....	38

1 Introdução

Software livre, por conceito, é o software que respeita a liberdade e senso de comunidade dos usuários (GNU, 2022), onde os mesmos possuem a liberdade de:

- Liberdade 0 - Total liberdade de execução para qualquer fim;
- Liberdade 1 - Liberdade para estudar e modificar o código fonte;
- Liberdade 2 - Liberdade de redistribuir cópias com intenção de colaborar com outras pessoas;
- Liberdade 3 - Liberdade de distribuir cópias alteradas

Essas liberdades têm por objetivo garantir que os usuários estejam livres de restrições e com total direito sobre o software.

Um exemplo de software livre consolidado é o sistema operacional Debian GNU/Linux, que possui uma estrutura comunitária e colaborativa que o aprimora e o mantém. O Debian é uma organização formada exclusivamente de voluntários dedicada ao desenvolvimento de software livre e a promover as ideias da comunidade de Software Livre.

O Debian foi lançado em 1993 e desde então tem se mantido no cenário de sistemas operacionais e atraindo mais colaboradores e usuários. A crescente popularidade de projetos de software livre levanta interesses em como a estrutura funciona e o que pode ser adaptado, dentro e fora do contexto FLOSS para beneficiar a Engenharia de Software. Existem muitos projetos FLOSS que são integrados ou utilizados no Debian. Dentre estes, tem-se o FreedomBox.

O FreedomBox, lançado em 2010, é uma ferramenta sem fins lucrativos que possibilita a criação de servidores privados para usuários com pouca experiência, conhecimento e desempenho de hardware. O código fonte, escrito em Python, fica disponível para acesso a todos aqueles que tem interesse em estudar ou colaborar com o projeto.

Este trabalho visa identificar aspectos estruturais, comportamentais e práticas utilizadas em comunidades FLOSS através de um estudo de caso etnográfico com participação ativa no FreedomBox. Ao final deste estudo, foi possível descrever a comunidade e suas práticas, como uma fonte de informações atuais sobre a metodologia empregada em projetos FLOSS, a partir da vivência e análise. Buscamos responder as seguintes questões a partir dos resultados obtidos:

1. Como fazer parte de uma comunidade?
2. Quais são os principais desafios encontrados?
3. O que se tem na literatura sobre FLOSS que não condiz com observado no FreedomBox?

1.1 Problema

Com esse projeto visa-se colaborar e participar efetivamente do movimento FLOSS a partir do desenvolvimento e estudo do código fonte. Atualmente existem 449 necessidades, descritas como *Issues* no *Gitlab*, que abrangem bugs, melhorias e testes. Essas *Issues*¹ possuem algumas classificações que orientam a sua execução, objetivo e categorizam o nível de expertise presumidamente necessária para ser atendida. Das 449, 347 estão em aberto, sem início de desenvolvimento. A resolução dessas *Issues* irá trazer benefícios para o projeto como, por exemplo, maior segurança, compatibilidade, aumento do número de utilizadores, qualidade e melhor experiência para os usuários.

1.2 Objetivos

De acordo com Osterlie and Jaccheri (2007), citado por Wen (2021), "a literatura da Engenharia de Software apresenta pouco sobre o que é executado pelos profissionais, quais atividades eles fazem e com que frequência acontecem".

O objetivo deste projeto é colaborar com a ferramenta de software livre "FreebomBox", com sua utilização no sistema operacional Debian. A colaboração é feita em comunidade, com o objetivo de aprimorar a ferramenta, desenvolvendo novas funcionalidades e correção de erros. As oportunidades de atuação/colaboração são identificadas por usuários e outros desenvolvedores através de seu *Gitlab*, onde são descritas as necessidades e melhorias a serem atendidas. Com a colaboração visa-se ter dados para descrever, sob a perspectiva de um novo colaborador, a comunidade e suas práticas, além de fornecer um guia para iniciantes/novos colaboradores em projetos FLOSS. Ao fim deste estudo espera-se ter um material que sirva de base inicial e guia para futuros estudos e indivíduos interessados a iniciar projetos de software livre. Descrevendo aqui os processos, os passos iniciais, a estrutura e ecossistema a partir de uma visão ativa em uma comunidade específica.

1.2.1 Objetivo geral

Observar a dinâmica do processo de desenvolvimento distribuído do software FreedomBox na perspectiva de um novo contribuidor da comunidade e comparar com conhecimento adquirido na literatura científica contemporânea sobre comunidades FLOSS em geral.

¹ <https://salsa.debian.org/freedombox-team/freedombox/-/issues>

1.2.2 Objetivos específicos

A consecução do objetivo geral determina a necessidade de atendimento aos seguintes objetivos específicos:

- Localizar no FreebomBox *issues* com os quais contribuir no *GitHub*.
- Criar solicitações de MR.
- Implementar melhores práticas para se comunicar com mantenedores do projeto e realizar revisões de código.

1.3 Justificativa

Ao longo dos anos tecnologias de software livre têm atraído cada vez mais interesses, sendo esses interesses de empresas privadas até mesmo de desenvolvedores. A busca por software livre levanta a questão acerca de como é o estado de arte atual do desenvolvimento FLOSS e como funciona o acesso a este ambiente. Dessa forma, o estudo dessa metodologia se justifica como área de interesse da engenharia de computação/software.

O interesse na utilização de software livre permite que comunidades de software livre e de desenvolvedores possam cooperar entre si, compartilhando experiências e conhecimentos, com o intuito de que os projetos produzam aplicações cada vez melhores e mais eficientes. O projeto consiste em descrever os processos, os passos iniciais, a estrutura e o ecossistema a partir de uma visão ativa em uma comunidade específica.

1.4 Estrutura do trabalho

No capítulo 2 é apresentado conceitos gerais e revisão de literatura. No capítulo 3 são descritos a metodologia de desenvolvimento composta neste trabalho. No capítulo 4 é descrito a análise dos resultados. O capítulo 5 relata as conclusões sobre o trabalho e apresenta propostas de pesquisas futuras.

2 Conceitos gerais e revisão da literatura

Há diversos estudos sobre comunidades de projetos FLOSS, que analisam diferentes perspectivas sobre o assunto, que vão desde o ecossistema ao menor indivíduo.

Wen (2021) faz um estudo etnográfico e compreensivo sobre o modelo atual de desenvolvimento do kernel Linux, onde busca preencher o distanciamento de informações entre o que se tem na literatura acadêmica com o cenário real, a partir da participação na comunidade com contribuições e de referências na literatura cinzenta, fornecendo também uma terceira perspectiva no âmbito observador. Sua pesquisa busca responder três questões principais:

- Como os estudos da Engenharia de Software e publicações da comunidade Linux descrevem o atual modelo de desenvolvimento Linux?
- Quais as técnicas de pesquisa que podem ser utilizados para examinar um projeto FLOSS por meio de suas publicações da comunidade?
- Quais são as possíveis lacunas e oportunidades para pesquisa acadêmica e tópicos de desenvolvimento FLOSS?

Como resultado desse trabalho supracitado, os autores obtiveram as respostas das perguntas direcionais bem como o mapeamento da dinâmica interna do modelo de desenvolvimento do kernel Linux, além de mapear problemas que poderiam ser mais discutidos dentro da literatura formal. O trabalho ainda levanta questões a serem analisadas em trabalhos futuros, com por exemplo, uma investigação dos conceitos para identificar desafios e benefícios de alterar o modelo de manutenção para suportar escalabilidade e mitigar gargalos e sobre a dificuldade na transferência de conhecimento e a curva de aprendizagem de novos membros.

Já Trinkenreich et al., (2021) abordam a perspectiva do conceito de sucesso dentro de comunidades FLOSS, voltado para a visão dos membros. Os autores buscam nesse estudo classificar e identificar os principais motivadores dos colaboradores de FLOSS dentro do cenário e o que eles entendem como sucesso. A pesquisa se dá a partir de uma série de entrevistas em que as perguntas giram em torno da carreira e atuação nas comunidades. Com os dados obtidos é feita uma triangulação dos resultados, gerando um modelo capaz de mapear o indivíduo de acordo com seu objetivo. Como conclusão, os autores afirmam que o sucesso é multifacetado e vai além do objetivo de se tornar um membro principal e com contribuições relacionadas a código. Ainda concluem também que a percepção de sucesso individual determina o futuro tanto do indivíduo como o dos projetos, uma vez que pode acarretar na saída do contribuidor. O resultado final espera-se ser utilizado como um suporte para evitar o abandono das comunidades.

Uma outra perspectiva sobre o assunto pode ser observada nos trabalhos científicos Steinmacher et. Al. (2013) e Steinmacher et. Al. (2014) ambos contendo como autor Igor Steinmacher. As duas pesquisas trazem um olhar sobre como funciona o ecossistema de comunidade para novos membros e relatam as principais causas que os fazem abandonar bem como as principais barreiras enfrentadas no início do processo.

No estudo apresentado por Steinmacher (2013), foram analisadas as barreiras encontradas pelos recém-chegados à uma comunidade de software livre. Como resultado, o estudo oferece um modelo hierárquico que se baseia em cinco categorias de barreiras: encontrar uma maneira de começar, interações sociais, problemas de código, problemas de documentação e conhecimento dos recém-chegados. As barreiras mais evidenciadas são as habilidades técnicas prévias dos recém-chegados, receber resposta da comunidade, centralidade dos contatos sociais e encontrar a forma adequada de começar a contribuir.

2.1 Estudo Etnográfico

Um estudo etnográfico é um método de pesquisa comportamental sobre um grupo ou indivíduo que fazem parte de uma comunidade. A análise é construída sobre os aspectos funcionais, comportamentais e crenças dessa comunidade.

O objetivo deste tipo de pesquisa é se obter dados para se conhecer o objeto de estudo, de maneira a conseguir extrair informações, padrões, planos e inovações que beneficiem os interesses acerca da pesquisa.

Existem três métodos de coleta de dados que são frequentemente utilizados na pesquisa etnográfica: observação participante, entrevistas e análise de documentação (Sarmiento, 2011). Neste trabalho, coletamos dados do projeto Freedombox examinando os documentos produzidos pela comunidade e com observação participante.

2.2 Observação Participante

Quando consideramos um contexto de estudo etnográfico, existem formas de obtenção dos materiais utilizados para análise e obtenção de resultados diferentes. Uma maneira de se obter é a observação participante. A observação participante consiste de uma metodologia de pesquisa na qual o observador interage com o ecossistema do observado de forma a se misturar, tornando-se um membro participante do grupo imerso ao ecossistema a ser analisado.

2.3 Debian

O Debian foi lançado em 16 de agosto de 1993 por Ian Murdock. Através do Manifesto Debian, escrito por Murdock, foi criada uma distribuição GNU/Linux² a ser mantida de uma maneira livre, segundo o espírito do GNU.

Os sistemas Debian atualmente usam o kernel Linux ou o kernel FreeBSD. O Linux é um software criado inicialmente por Linus Torvalds com a colaboração de milhares de programadores de diversas nacionalidades. O Debian se destaca pela grande quantidade de pacotes de software. O Debian destaca-se também pelo compromisso com a estabilidade do sistema. Apesar de possuir uma versão instável para testes, sua versão principal só é lançada quando considerada livre de problemas de estabilidade.

2.4 Comunidades FLOSS

O movimento Free Libre Open Source Software - FLOSS são caracterizados pela existência de uma cultura própria tendo como base a cooperação e distribuição em rede de código-fonte.

O FLOSS desenvolve novas formas de produção e de circulação do conhecimento que divergem com as práticas consolidadas e limites estabelecidos à liberdade de circulação e apropriação da informação e da tecnologia, representada pelas licenças jurídicas e interesses privados, sendo configurado como um movimento de caráter político. Nesse sentido, Coleman afirma que o desenvolvimento do software livre não se limita à produção de novas tecnologias, mas estão inseridas em um processo mais amplo de debate sobre “liberdades civis, propriedade e software, afirmando de novas maneiras que código é uma forma de expressão” (Coleman, 2010, p.420 apud Araújo, 2018, p.32)

Um dos principais resultados dessa cultura são as licenças jurídicas abertas ou Copyleft, que estabelecem um novo código de conduta para a circulação e a apropriação de conhecimentos embasados nas liberdades de uso, modificação e compartilhamento da informação.

As disputas conceituais e políticas que marcam a história do Software Livre e do movimento de Código Aberto também são semelhantes às aquelas envolvendo abertura e privatização da ciência. Raymond, em “A Catedral e o Bazar”, descreve as diferenças entre os modelos aberto e fechado na produção do software. Raymond, observando os projetos de

² <https://www.debian.org/index.pt.html>

software livre, os separou em dois conjuntos, com as características encontradas em um projeto pessoal e em projetos como Linux e GCC, “A Catedral e o Bazar”.

Catedral: o conceito é de que nesse método de desenvolvimento de software, o código fonte é disponibilizado em cada release do sistema, sendo restrito a um grupo exclusivo de desenvolvedores.

Algumas características desse método:

- Ambiente fechado (altamente hierarquizado);
- Diversos líderes e desenvolvedores;
- Desenvolvimento centralizado;
- Versões estáveis e/ou betas;
- Metodologia clássica.

Bazar: o conceito é de que o código fonte do sistema é desenvolvido de forma totalmente aberta e pública, proporcionando que o desenvolvimento seja compartilhado e colaborativo.

Algumas características desse método:

- Ambiente aberto (todos podem participar);
- Diversos líderes e desenvolvedores;
- Desenvolvimento cooperativo;
- Várias versões;
- Metodologia indefinida.

A tese central do ensaio de Raymond define que “*Dado um número de olhos suficiente, todos os erros são triviais*”. Essa afirmação sugere que se o código fonte de um software está disponível para testes e experimentação pública, é fato que os possíveis erros serão descobertos com mais rapidez e assertividade, agilizando a entrega com qualidade e proporcionando a melhoria contínua.

2.5 Comunidade FreedomBox

O Freedombox³ é uma comunidade de software livre, que teve seu conceito cunhado por Eben Moglen, fundador e diretor do Software Freedom Law Center , em 2010, onde ele lançou um desafio para os desenvolvedores tornarem o Debian mais simples, com a criação de um servidor privado para utilização por indivíduos com pouca experiência (não-técnicos). Esse servidor é capaz de gerenciar diversas outras aplicações e funcionalidades como

³ <https://freedombox.org/>

armazenamento de arquivos, gerenciamento de proxys, rede compartilhada entre dispositivos e compartilhamento de mídias visuais, sendo tudo proveniente de software livre.

O FreedomBox é composto de: um sistema de software livre e hardware sempre ativo, barato e com baixo consumo de energia, do tamanho de um dicionário de bolso. O sistema de software é 100% gratuito e de código aberto e está disponível para download sem nenhum custo. FreedomBox é pré-carregado com mais de 20 aplicativos e recursos projetados para proteger a liberdade, privacidade e direitos do usuário.

A comunidade do FreedomBox conta com 34 membros de diferentes nacionalidades. Tendo como principais líderes de desenvolvimento Sunil Mohan Adapa e James Valleroy, que controlam as entregas de novos códigos. Não existe uma hierarquia de membros dentro da comunidade e possibilita a todos total liberdade de caminho a se seguir, desde cadastro de tarefas até execução das mesmas. Possui uma corrente de e-mails formais sobre versionamento e informações importantes, além de uma reunião periódica com os membros que ocorre no domingo a cada 15 dias no horário de 17:00 UTC.

3 Metodologia

Neste capítulo serão apresentados os métodos utilizados para atender aos objetivos propostos por este trabalho, bem como as atividades que foram realizadas para este fim. Utilizando metodologia de estudo etnográfico para analisar o atual cenário de desenvolvimento FLOSS, comparando os resultados obtidos através de observação participante com trabalhos correlatos e o que se tem de conhecimento dentro da literatura cinza.

3.1 Estratégia de Pesquisa

Um ecossistema FLOSS, em sua grande maioria, possui uma comunidade que atua no desenvolvimento do projeto ao qual está associada, sendo essa grande fonte de conhecimento e informações para se analisar o cenário atual deste sistema e como o mesmo têm evoluído.

Como discutido no capítulo anterior, têm-se uma gama de informações que podem ser retiradas desse fluxo de desenvolvimento que trazem uma perspectiva ao que se tem pré-estabelecido no meio acadêmico. Assim é necessário filtrar o escopo e quais informações serão úteis ao objetivo da pesquisa.

Dessa forma, foram traçadas três questões que limitam e tornam mais claros quais são os dados que serão necessários para o estudo e guiaram este capítulo:

1. Como fazer parte de uma comunidade?
2. Quais são os principais desafios encontrados?
3. O que se tem na literatura sobre FLOSS que não condiz com o mundo real?

3.2 Desenvolvimento

A primeira etapa de um estudo etnográfico é fazer parte da comunidade que se quer estudar. Como a forma de coleta de dados é a metodologia de observação participante, foi necessária para a realização deste projeto a participação do autor como membro contribuinte da comunidade.

Como apresentado anteriormente, a comunidade escolhida para o estudo foi a referente ao FreedomBox. A escolha se deu a partir de uma busca por comunidades que possuíam membros ativos, que houvesse contribuições provindas do Brasil visando facilitar tanto a introdução como a comunicação. O FreedomBox é um software livre e que tem seu desenvolvimento voltado ao Debian, sendo este também um software livre com sua própria comunidade e que tem em sua composição apenas softwares livres. Optou-se por esta

comunidade devido à proximidade da área de desenvolvimento escolhida pelo autor, bem como a diversidade de natureza de contribuições que poderiam ser efetuadas, além de atentar aos requisitos descritos anteriormente.

Para se tornar um membro, existe um protocolo a ser seguido que pode ser acessado pela página da comunidade, ou pode ser obtido através de comunicação com um dos integrantes. Com esse intuito, foi feita uma comunicação primária com um dos contribuidores de maior hierarquia e que poderia auxiliar no ingresso à comunidade. O e-mail pode ser obtido através da página do fórum do FreedomBox⁴.

Após a comunicação inicial, é necessário possuir uma conta no sistema do *Gitlab* e solicitar acesso ao repositório do software utilizado pela comunidade. Esse processo leva algum tempo visto que tem a necessidade de aprovação por um dos contribuidores que possuem cargo suficiente para esta ação. Uma vez que tenha obtido o acesso, o iniciante tem à sua disposição informações desde a missão, visão da comunidade, manual que mostra requisitos e passos para se ingressar e começar a contribuir, bem como relatos de pessoas que já colaboram com a comunidade.

A escolha pela participação observante, trouxe a necessidade de fazer parte da comunidade e voltado para o desenvolvimento, dessa forma, o passo seguinte é idealizar como conseguir obter os dados de forma assertiva. Sendo assim, foram escolhidas atividades de desenvolvimento que foram executadas com o objetivo de vivenciar o fluxo de contribuição. Portanto, utilizando o manual de contribuição que é acessível no repositório do software, foi executada a etapa de configuração de máquina e ambiente de desenvolvimento e de testes. Apesar do Freedombox ser um software puramente Debian, é possível configurar o ambiente de desenvolvimento em qualquer outro Sistema Operacional (S.O). Com isso em mente, foram feitas duas configurações distintas uma no Debian Bullseye e uma no S.O Windows 10, optando pela utilização no Debian devido a uma maior compatibilidade. No repositório encontra um documento intitulado HACKING.md, o qual descreve as diferentes necessidades de configuração para cada S.O. Mesmo com esse documento foi observado uma gama de dificuldades para executar os passos descritos e configurar o ambiente conforme o esperado.

Dentro da página do *Gitlab* da comunidade existe um tópico acessível que permite ver os projetos, atividade e correção de falhas que estão planejados. Esse tópico, chamado de *Issue*, contempla todas as tarefas priorizadas para atuação dos contribuidores. Existe uma hierarquia interna à comunidade que auxilia novos membros no ingresso do ecossistema, marcando o nível necessário de conhecimento do funcionamento do código, conhecimento da linguagem e ainda de *expertise* pessoal do contribuinte, como é possível observar na Figura

⁴ <https://discuss.freedombox.org/>.

1, onde estão as principais *labels* utilizadas. Dessa maneira as tarefas que foram escolhidas para contribuição são aquelas que possuíam nível iniciante, marcadas com a descrição de “*beginner*”, sendo possível utilizar um filtro para listar apenas as que possuem essa categoria, como observado na Figura 2, que exibe a listagem de atividades recomendadas para iniciantes. Inicialmente foram traçadas quatro contribuições para serem efetuadas. Estas e suas atividades correspondentes serão listadas na próxima seção.

Prioritized Labels

Drag to reorder prioritized labels and change their relative priority.

Contributor Invite	FreedomBox / FreedomBox	Issues · Merge requests · Prioritized label	★ ✎ ⋮	Subscribe
Next Release	FreedomBox	Issues · Merge requests · Prioritized label	★	Subscribe ▼
Beginner	Suitable for first time contributors of the project. FreedomBox developers are willing to guide/motor contributors to this issue. FreedomBox	Issues · Merge requests · Prioritized label	★	Subscribe ▼
Help Wanted	Recognized as important, but not a project priority yet. Contributions are especially welcome. FreedomBox	Issues · Merge requests · Prioritized label	★	Subscribe ▼

Figura 1: Categoria de marcação das atividades. Fonte: <https://salsa.debian.org/freedombox-team/freedombox/-/labels>

The screenshot shows a web browser displaying the 'Issues' page of the FreedomBox project, filtered by the 'Beginner' label. The interface includes a top navigation bar with 'Open 24', 'Closed 182', and 'All 206' counts, along with buttons for 'Edit issues' and 'New issue'. A search bar and a filter dropdown set to 'Beginner' are visible. The main content area lists 11 issues, each with a title, a brief description, a list of labels (e.g., Beginner, Enhancement, UI, UX, Bug, Cleanup, Documentation, no coding needed), and an 'updated' timestamp. The issues are as follows:

Issue ID	Created	Author	Labels	Updated
#2161	10 months ago	Sunil Mohan Adapa	Beginner, Enhancement, UI, UX, ready for implementation	updated 8 months ago
#2146	11 months ago	Sunil Mohan Adapa	Beginner, Cleanup, ready for implementation	updated 11 months ago
#2126	1 year ago	Fioddor Superconcentrado	Beginner, Bug, ready for implementation	updated 5 months ago
#2121	1 year ago	Sunil Mohan Adapa	Beginner, Cleanup, ready for implementation	updated 1 month ago
#2115	1 year ago	Altadil	Beginner, Enhancement, UI, UX	updated 10 months ago
#1990	1 year ago	Joseph Nuthalapati	Beginner, Enhancement	updated 1 year ago
#1983	1 year ago	Joseph Nuthalapati	Beginner, Documentation, no coding needed	updated 1 year ago
#1968	2 years ago	Sunil Mohan Adapa	Beginner, Enhancement, ready for implementation	updated 2 years ago
#1850	2 years ago	Sunil Mohan Adapa	Beginner	updated 1 year ago
#1837	2 years ago	James Valleroy	Beginner, UI	updated 1 year ago
#1734	2 years ago	Joseph Nuthalapati	Beginner, Enhancement	updated 1 year ago

Figura 2: Lista de atividades filtrada pela marcação de "Beginner". Fonte:

<https://salsa.debian.org/freedombox-team/freedombox/-/issues>

3.3 Contribuições

Contribuições são as atividades executadas pelos membros de uma comunidade de software livre que auxiliam no crescimento do produto. Existem diversas formas de se contribuir que vão desde tradução, desenvolvimento até mesmo a atividades não relacionadas à código, como por exemplo, burocracias, documentações, publicidade e etc. Contribuição é o fator principal e mais desejado, sendo elas que caracterizam as comunidades e seus comportamentos, contribuição é colaborar em conjunto.

Neste trabalho o foco primário de contribuições é com relação ao desenvolvimento. O filtro utilizado para a escolha das tarefas é voltado à uma análise do fluxo de sistema. Dentro da comunidade FreedomBox a atribuição de tarefas ocorre de forma livre, ou seja, o contribuidor escolhe com o que deseja executar. Com isso, as tarefas que foram escolhidas são aquelas que possuíam âmbitos de desenvolvimento bem como a identificação de iniciante.

As atividades executadas tiveram os temas abaixo como objetivo:

1. Refatoração

Refatoração é um processo de melhoria de código sem envolver a criação de novas *features*, para transformar um código mal elaborado ou confuso em código limpo e com *design* simples, melhorando sua legibilidade e eficiência. Refatorar é a melhora do entendimento do código, facilitando a manutenção e a resolução dos *bugs*.

2. Correção de *bug*

Bug se refere às falhas inesperadas que ocorrem ao executar algum software ou usar um hardware. Esses erros imprevisíveis que prejudicam o funcionamento correto de alguma tecnologia e podem desencadear problemas como travamentos e acessos às informações sigilosas. As falhas inesperadas são uma porta de entrada para os crimes cibernéticos.

Correção de *bug* é uma alteração em um sistema ou produto projetado. Uma das aplicações mais comuns de correções de bugs é um protocolo técnico, usado para identificar os tipos de bugs. A documentação exige o arquivamento de quaisquer alterações até que seja resolvido com uma correção de *bug*. Essa manutenção de registros ajuda a impedir que as empresas de tecnologia se envolvam em questões técnicas que podem afetar um produto ou sistema durante sua jornada, desde o desenvolvimento inicial até o lançamento final.

3. Documentação

Documentação é uma das ferramentas mais importantes em qualquer projeto de software, pois é ela quem auxilia em desenvolvimentos compartilhados direcionando o que se tem já construindo e seu comportamento funcional, dá suporte para novos membros e no entendimento do produto final pelos usuários. Por isso, é necessário se ter e manter a documentação clara e atualizada.

As atividades possuem uma marcação da sua finalidade, além das descritas na seção 3.2, que é possível observar na imagem a seguir:

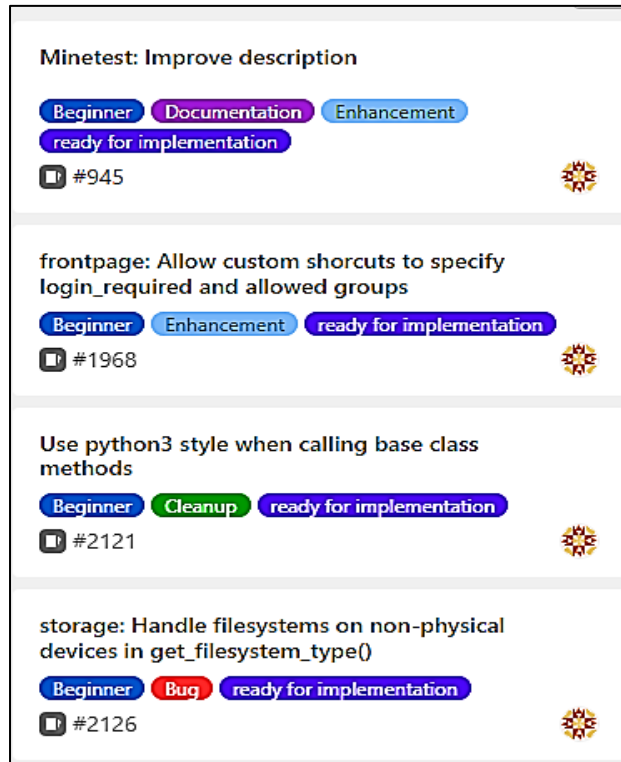


Figura 3: Atividades selecionadas para contribuição. Fonte: https://salsa.debian.org/freedombox-team/freedombox/-/boards?assignee_username=tunebes

A atribuição é feita pelo próprio membro na tarefa em que deseja executar. A Figura 3 mostra as atividades atribuídas ao autor.

Com o ambiente configurado, conforme seção 3.2, e as atividades selecionadas deu-se então o início ao fluxo de desenvolvimento. As etapas de desenvolvimento são descritas nos arquivos CONTRIBUTING.md e HACKING.md que se encontram no repositório do projeto. No primeiro encontra-se o padrão de nomenclatura de *branches*, que são as versões locais do código, e no segundo o protocolo de execução do código na máquina do contribuidor, execução dos testes e protocolo de entrega.

✚ CONTRIBUTING.md	CONTRIBUTING: Require flake8 compliance	2 years ago
✚ COPYING.md	doc: Update the correct license for documen...	3 years ago
✚ HACKING.md	HACKING: Improve documentation on how t...	5 months ago
✚ INSTALL.md	d/rules: vagrant: INSTALL.md: Fix installing d...	2 years ago
✚ README.md	*: Update links to repository and project page	2 years ago

Figura 4: Arquivos de fluxo de desenvolvimento. Fonte: <https://salsa.debian.org/freedombox-team/freedombox/-/tree/master>

Para uma contribuição ser aceita, tem-se um protocolo a ser seguido. Este começa na criação de uma *branch* com a nomenclatura correta a partir da versão mais atual do código (master). Uma *branch* nada mais é do que uma ramificação de uma versão do código, é uma cópia que pode ser modificada sem alterar o original. Essa política está diretamente relacionada com versionamento de código para controle do que entra no repositório principal e o que não e obter um controle e rastreo do código.

Após a criação da *branch*, pode-se efetivamente iniciar o desenvolvimento da tarefa no código. A primeira atividade, que também pode ser chamada de “*Issue*”, foi a de número #2126. Essa *issue* descreve a necessidade de corrigir um bug encontrado no código que não consegue identificar dispositivos não físicos e causava uma interrupção da execução. Com a correção realizada, o próximo passo foi a execução dos testes da nova versão. O fluxo de testes é dividido em três caminhos que devem ser seguidos:

- Teste automatizado: o teste automatizado é a criação de roteiros de fluxo de execução do software que serão executados de forma automática pela aplicação, a fim de validar que nenhum dos fluxos parou de funcionar com a alteração feita no código. Este tipo de teste ajuda a prevenir que *bugs* não previstos entrem na versão final.
- Teste de cobertura: o teste de cobertura é a validação se existe algum fluxo que não foi mapeado para o teste automatizado. Esse ajuda na identificação de novos fluxos que foram inseridos no código.
- Teste funcional: o teste funcional é a execução da aplicação e validação de fluxos funcionais ou não funcionais em tempo de execução, como exemplo: testes de telas, validação de tempo de resposta, comportamento de botões e etc. O teste funcional é utilizado também para validar a compilação do código e se a alteração está agindo conforme o esperado.

Com todos os fluxos de testes validados com sucesso, o próximo passo é a criação de um *Merge Request*. Um MR é uma solicitação de inclusão do novo código proveniente da *branch* previamente criada no repositório final. O processo de MR é um processo muito cuidadoso e este necessita da aprovação de membros mais experientes, para controle do que vai entrar na versão final. Nesta etapa, um membro

mais experiente analisa todo o código que foi solicitado para inclusão e validar se o padrão de desenvolvimento está de acordo com o protocolo.

O FreedomBox faz um versionamento da aplicação, ou seja, ele atualiza as versões com uma frequência determinada evoluindo o produto, seguindo padrões ágeis. Dessa forma é discutido quais códigos irão entrar na próxima versão, interferindo na aprovação dos MRs. As contribuições relacionadas ao código se encontram na etapa de avaliação e pendente de aprovação do *merge request*. A Figura 5 representa de forma ilustrativa o fluxo que se seguiu para o ingresso na comunidade e primeira contribuição, podendo ser utilizado como um guia resumido do processo.

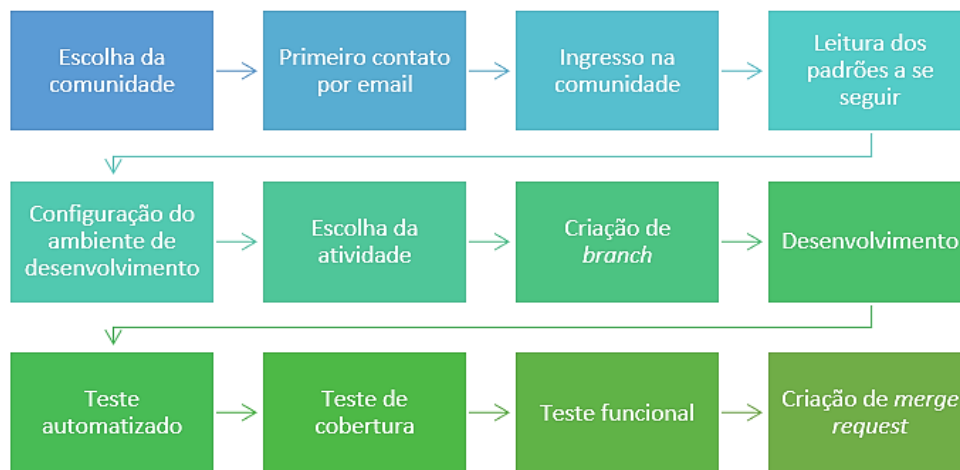


Figura 5. Ilustração representativa do fluxo de desenvolvimento no FreedomBox

3.4 Participação na comunidade

Fazer parte de uma comunidade vai além de contribuições e um ponto importante é fazer parte da rotina e do meio social. Dessa forma, aderiu-se à utilização da ferramenta de comunicação Internet Relay Chat, comumente chamada de IRC da comunidade do FreedomBox. O IRC é um sistema de bate-papo baseado em texto que permite discussões entre qualquer número de participantes. Além do acesso ao IRC, outras formas de comunicação foram utilizadas, como e-mail e o fórum que a comunidade mantém.

Outra forma de participação é a presença nas chamadas quinzenais que ocorrem aos domingos, para divulgação de informações e discussão dos tópicos mais relevantes pré-estabelecidos pelos líderes da equipe. De maneira a obter dados de como a comunidade se organiza e comporta.

4 Apresentação e análise dos Resultados

Neste capítulo é apresentada uma análise dos dados coletados, a partir da investigação realizada por meio da metodologia aplicada e uma comparação dos resultados obtidos com os encontrados nos estudos apresentados no referencial teórico. A atuação nas atividades, participação das reuniões, leitura do fórum e comunicações com demais membros trouxe uma gama de dados que são utilizados para responder as perguntas propostas pelo trabalho.

O primeiro resultado a ser levantado é o de dificuldades encontradas. No texto “Barriers Faced by Newcomers to Open Source Projects: A Systematic Review” de I. Steinmacher et al. (2014) o resultado obtido categoriza as principais dificuldades encontradas em seu estudo, sendo elas: interações sociais, encontrar um caminho para iniciar, problemas nas documentações, dificuldades com o código e o conhecimento do iniciante.

Com a pesquisa aplicada à comunidade do FreedomBox, foram encontradas dificuldades similares, nas quais foram classificadas em quatro grupos:

1. Documentação não clara

Um dos pontos encontrados na pesquisa foram os documentos de suporte para configuração do ambiente de desenvolvimento. Os documentos apesar de bem escritos não são claros o suficiente para novos membros o que acabou gerando um grande desgaste e múltiplas configurações afim de se obter todo o ambiente configurado, e consequentemente tomando um tempo superior ao realmente necessário. Durante o início do processo encontrou-se dificuldades para configuração do setup de desenvolvimento (instalados Debian 11, VirtualBox, vagrant python além da configuração Github). Foram feitas tentativas em sistemas operacionais diferentes, obtendo-se o mesmo resultado. Em busca de solucionar, houve várias tentativas de contato com a comunidade, principalmente a partir de e-mail. O que ocasionou numa reformulação do cronograma. Outro ponto a se ressaltar é a falta de uma descrição das dependências de desenvolvimento, como por exemplo, o que é necessário se ter para que consiga executar a aplicação e qual a versão das bibliotecas são utilizadas.

Problema na documentação, foi uma barreira no início do projeto muito grande dado que existem problemas na documentação de testes funcionais, documentação de configuração de ambiente e na documentação de padrões de nomenclatura e criação de branch. Na Figura 6 é exibido uma captura de tela do documento de HACKING.MD que possui a descrição dos testes de forma genérica, e sendo este a causa de um problema no qual a execução os testes não estavam sendo executados, e posteriormente após a resolução, a execução gerou arquivos de modificação no código e não há descrição do que são essas modificações e se as mesmas devem ser consideradas artefatos de desenvolvimento. Na

Figura 7 mostra como o contato é debilitado em relação aos membros mais experientes, onde houve uma tentativa de solicitação de apoio e não houve réplicas por mais de 5 meses.

Testing

Running Tests

To run all the standard unit tests in the container/VM:

```
guest$ py.test-3
```

To run a specific test function, test class or test module, use pytest filtering options. See pytest documentation for further filter options.

Examples:

```
# Run tests in a directory
guest$ py.test-3 plinth/tests

# Run tests in a module
guest$ py.test-3 plinth/tests/test_actions.py

# Run tests of one class in test module
guest$ py.test-3 plinth/tests/test_actions.py::TestActions

# Run one test in a class or module
guest$ py.test-3 plinth/tests/test_actions.py::TestActions::test_is_package_manager_busy
```

Some tests are skipped by default:

- tests that need root privileges,
- functional tests (they need additional preparation to run. See next section),
- tests that take much time to run.

Use `sudo` to run the ones that need root access:

```
guest$ sudo py.test-3
```

To force functional tests and tests that take long to run, set the environment variable `EXTENDED_TESTING=1`:

```
guest$ EXTENDED_TESTING=1 py.test-3
```

To really run all tests, combine `sudo` with `EXTENDED_TESTING`:

```
guest$ sudo EXTENDED_TESTING=1 py.test-3
```

Figura 6. Documentação de execução de testes. Fonte: Capturada pelo autor.



Walace Martins @tunebes · 5 months ago

Developer





I've been having a lot of trouble with running it for development testing purposes. I followed the guide described in HACKING.md but still I couldn't run. Would you have any tips? I'm using Windows, but I've also tried using Debian 11.



Walace Martins @tunebes · 5 months ago

Developer





When I ran the test guide described in HACKING.md there were 180 file changes but no error was found. However, should I commit these changes caused by the test?

Edited by Wallace Martins 5 months ago

Figura 7. Duvidas a cerca do fluxo de testes automatizados. Fonte: <https://salsa.debian.org/freedombox-team/freedombox/-/issues/2126>

2. Socialização com demais membros

A comunidade do FreebomBox apresenta membros de diversos países, isso leva com que se tenha um padrão de comportamento e linguagem. A língua utilizada para registro, documentação e contatos é o inglês, e isto acaba gerando uma necessidade de conhecimento da linguagem em um nível mais avançado para a participação e socialização. É muito comum que comunidades FLOSS utilizem o inglês como linguagem de comunicação por se tratar de uma língua mundial. Outra dificuldade é em relação à comunicação entre as partes envolvidas para um melhor andamento do projeto causado pela divergência de fuso horário. Uma vez que a comunidade como um ecossistema utiliza fuso UTC para seus ritos. A comunidade possui fóruns e também um grupo no IRC que tem o objetivo de facilitar a sociabilização, porém não soluciona o problema, uma vez que os contatos são curtos e de réplica também demorada, levando em média 3 meses.

Ainda que como membro, a dificuldade de socialização interfere na evolução do iniciante e no sentimento de pertencimento à comunidade. Uma sugestão do autor seria um acompanhamento por parte dos membros mais antigos para realmente criar nos novos contribuidores um melhor senso de comunidade. Um exemplo prático dessa dificuldade pode ser visto no tempo para se obter um retorno sobre uma das atividades selecionadas nesta análise, que levou 1 mês. Na Figura 8 mostra um contato efetuado acerca do MR de uma das contribuições e a necessidade de melhoria a ser codificada, vale ressaltar que esse MR esteve em avaliação por mais de um mês.

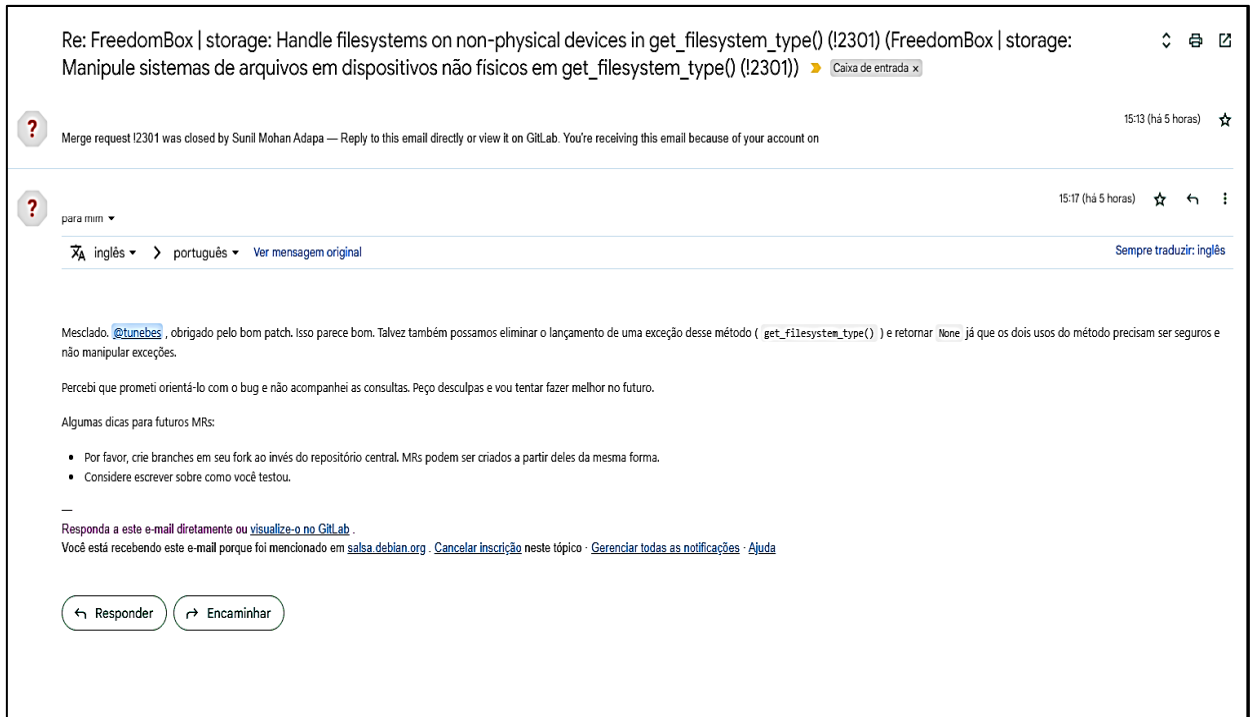


Figura 8. E-mail de um dos líderes do FreedomBox a respeito de merge request.

3. Dificuldades com o código

Como membro iniciante, o primeiro contato com o código desenvolvido foi uma grande barreira a ser enfrentada. O padrão de desenvolvimento aplicado, as integrações sistêmicas e o padrão de código são pontos muito importantes em todo projeto de software e o primeiro contato demanda um certo tempo para entendimento e compreensão, o que acaba impactando o desenvolvimento. Apesar de ter sido um ponto observado em meio ao ecossistema do FreedomBox, o que durou cerca de um mês, esse é um problema fortemente encontrado em outros ambientes de desenvolvimento, principalmente naqueles em que se tem uma rotatividade muito grande de desenvolvedores.

É interessante ressaltar novamente que a comunidade FreedomBox permite o ingresso de membros independente de sua instrução ou qualificação profissional e isso agrega que novos membros não necessariamente têm níveis de conhecimento tecnológicos voltados a programação bem estabelecidos.

4. Gestão de tempo

Como em qualquer projeto, gerenciar o tempo destinado à sua evolução é crucial, e uma dificuldade para membros iniciantes é saber como lidar com esse gerenciamento durante as contribuições e participações. Esse problema não afeta somente o novo membro, mas é possível observar também em membros mais experientes. Quando se fala em comunidade FLOSS, a maioria das contribuições são não remuneradas, o que acaba sendo

um trabalho a parte o desenvolvimento no sistema. Além disso, é necessário se ter uma dinâmica onde os horários coincidam para ser obter ajuda, por parte do iniciante, uma vez que essas comunidades apresentam membros em fuso horários diferentes, portanto ter essa sincronia entre quando efetuar o contato para ser obter retorno pode acabar se tornando um malabarismo. Um exemplo disso pode ser visualizado na Figura 6, onde um dos líderes se desculpa por não ter conseguido guiar o membro iniciante em suas dúvidas, e é notório que esse contato levou 5 meses para um respaldo.

Contrapondo o resultado de I. Steinmacher et al..., na comunidade FreedomBox não foram encontradas as dificuldades com “caminho a se iniciar”. Como descrito na seção 3.2, existe uma categorização das tarefas que consideram conhecimento do código e experiência do contribuidor. Dado a todas as dificuldades, a programação de execução do cronograma de contribuições acabou sendo impactado.

A observação participante permitiu que a pesquisa fosse mais que apenas uma leitura e sim uma vivência do ecossistema FLOSS, e essa vivência revela que o que se tem documentado hoje na literatura cinza se encontra divergente do contexto atual do FLOSS. Um exemplo disso pode ser observado no texto de RAYMOND(1999), onde ele descreve o movimento FLOSS categorizado em duas formas a catedral e o bazar, porém é válido observar que desde sua análise, o movimento cresceu muito e ao mesmo tempo se modificou muito. É possível ainda relatar que cada comunidade hoje segue um fluxo que a distância dos padrões que são comumente relatados na literatura cinza. Voltando à comunidade do FreedomBox, seu padrão estrutural não se remete nem à catedral nem ao bazar, por mais que seja notório que existam práticas que sejam compartilhadas por esses modelos não é adequado ter essa associação.

WEN(2021) tem esse mesmo resultado obtido a partir de sua análise sobre a comunidade do Kernel Linux, onde a autora conclui que o movimento FLOSS teve muita alteração causada pelo seu crescimento e aumento de interesse, ela ainda afirma que grandes mudanças ocorreram quando empresas privadas começaram a olhar para o movimento FLOSS e ver um grande potencial, conclui também que o mesmo projeto que foi responsável pelo conhecimento que temos na literatura cinza não é mais o mesmo. Hoje Wen desassocia o Kernel Linux do bazar e o descreve como uma ditadura controlada por um ditador.

Como em todo projeto, existem membros que abandonam e aqueles que fortemente permanecem contribuindo para o seu crescimento. É difícil traçar as razões que fazem que contribuidores continuem ativos nas comunidades, como conclui B. Trinkenreich et al. (2021) é a percepção de sucesso individual determina o seu futuro na comunidade. E na comunidade do FreedomBox não é diferente, desde sua origem a comunidade conta com 34 membros, analisando profundamente a participação dos mesmos, temos como resultado que temos

apenas 14 ainda permanecem ativos contribuindo com a ferramenta e participando efetivamente, o que é um número bem baixo e talvez tenha relação com as dificuldades que foram descritas nesse texto. E ainda vale ressaltar que desses 14 membros, 7 deles ingressaram recentemente, de 2020 até os dias atuais, como é possível observar na Figura 9. Essa baixa de membros ativos e membros mais experientes na ferramenta causa uma barreira muito difícil de ser quebrada para a permanência de novos membros e na sobrevivência da comunidade.

Account	Source	Access granted	Max role	Expiration	Created on	Last activity
	FreedomBox	9 months ago by Sunil Mohan Adapa	Developer	Expiration date 	26 Dec, 2021	22 Oct, 2022
 Wallace Martins @tunebes 	Direct member	5 months ago by Sunil Mohan Adapa	Developer	Expiration date 	12 Oct, 2021	22 Oct, 2022 
	FreedomBox	9 months ago by Sunil Mohan Adapa	Developer	Expiration date 	26 Sep, 2021	22 Oct, 2022
	FreedomBox	1 year ago	Developer	Expiration date 	26 Apr, 2021	9 Sep, 2021
	FreedomBox	1 year ago by James Valleroy	Developer	Expiration date 	27 Mar, 2021	24 Aug, 2021
	FreedomBox	2 years ago	Developer	Expiration date 	8 Oct, 2020	6 Feb, 2022
	Direct member	2 years ago by James Valleroy	Maintainer	Expiration date 	15 Mar, 2020	17 Aug, 2022
	FreedomBox	2 years ago	Developer	Expiration date 	7 Dec, 2019	3 Feb, 2021

Figura 9. Membros ativos mais recentemente. Fonte: https://salsa.debian.org/freedombox-team/freedombox/-/project_members?sort=recent_created_user

5 Conclusões

Este trabalho analisou a comunidade FreedomBox como um estudo de caso sobre comunidades FLOSS, com resultados parcialmente satisfatórios. Com os dados coletados é possível responder as perguntas que guiaram esta análise, anteriormente descritas na seção 3.1. A resposta à **“1. Como fazer parte de uma comunidade FLOSS?”**, surge das experiências provenientes da observação participante, fazer parte de uma comunidade é mais complexo do que efetivamente entrar em uma. Para entrar em uma comunidade é necessário apenas seguir o protocolo que cada uma segue, a complexidade está em se manter e ser produtivo. Dado a isto, é necessário se ter uma pesquisa prévia de como ela funciona prevenindo futuras frustrações, uma vez que cada comunidade tem suas próprias características. Trazendo a pergunta para a comunidade FreedomBox, fazer parte se torna um desafio, o que levanta a hipótese de uma má escolha para uma análise mais assertiva sobre o ecossistema FLOSS, uma vez que para fins de análise ou conhecimento do contexto FLOSS ela não se mostra muito receptiva e num cenário de primeiro contato com esse ambiente pode não ser muito proveitoso/produtivo, gerando uma primeira percepção negativa sobre experiência em comunidade. Contudo, isso mostra que é necessário se ter um filtro mais bem elaborado quando se deseja adentrar no universo FLOSS, dado a gama de dificuldades que foram levantadas. O primeiro passo é pesquisar sobre o contexto FLOSS, posteriormente encontrar um projeto que um novo contribuidor realmente veja benefícios em seu desenvolvimento e assim contribuir da melhor maneira possível.

Com relação a pergunta **“2. Quais são os principais desafios encontrados?”**, a mesma teve sua resposta durante as análises dos resultados, porém na visão do autor, a principal são a frequência de comunicação e a experiência. Por mais que na comunidade do FreedomBox as tarefas não tem uma data de entrega marcada ou expectativa de conclusão, a comunicação defasada somada à inexperiência com o código acabam gerando muitos gargalos e consequentemente uma desmotivação em se manter ativo.

E quanto a questão **“3. O que se tem na literatura sobre FLOSS que não condiz com o observado no FreedomBox?”**, pode-se concluir que em alguns aspectos existe uma divergência entre o que se tem na literatura acadêmica e o que se tem no contexto real. O movimento FLOSS está crescendo e a cada dia atraindo mais interesse, isso faz que essas comunidades sejam uma grande fonte de informações e novidades. Existem diversos estudos acerca desses ecossistemas e todos com uma perspectiva que pode vir a somar à Engenharia de Software, sejam elas padrões de projeto, documentação e até mesmo estudos sociais. Por mais que o movimento esteja se tornando cada vez maior, ainda sim muito pouco se é comentado fora do nicho específico, muito dos materiais produzidos por essas comunidades

ficam retidas ao meio que as produz. É interessante que novas perspectivas a respeito do movimento FLOSS sejam analisadas e que este tema consiga cada dia mais crescer fora do seu meio. E tem se tornado viável dado o número de pesquisadores começando a olhar para o movimento, buscando entendê-lo de dentro para fora que inclusive tiveram suas próprias perspectivas e resultados sobre esse ecossistema de maneira correlata à esta pesquisa.

Embora os resultados obtidos através desta análise tenham sido diferentes do esperado e descreverem um ambiente de difícil adaptação aos que almejam se tornarem contribuidores FLOSS, ainda é um ambiente restrito à uma comunidade, e não deve ser generalizado. A comunidade do FreedomBox não possui uma frente suficientemente apropriada para membros menos experientes tanto em nível pessoal quanto em nível de conhecimento da aplicação e isto fica claro nas dificuldades encontradas, além disso houve uma grande baixa de membros ativos durante o desenvolvimento da pesquisa o que acaba gerando um cenário de análise mais restrito, diminuindo o número de amostra das quais dados poderiam ser retirados.

Este trabalho mostra que o ambiente FLOSS como um todo é uma grande fonte de análise e como todo desenvolvimento de software também tem seus percalços a se ajustar. Como trabalho futuro, pode-se pensar nas dificuldades observadas dentro da comunidade e em formas de saná-las e também postular uma forma de filtro para encontrar comunidades que possuam uma maior amostragem de membros. Ainda é possível, a partir dessa pesquisa iniciar uma nova análise a respeito do movimento FLOSS, através de uma perspectiva diferente ou ainda uma correlata.

Referências

- ADAMS, Paul; CAPILUPPI, Andrea. **“Preenchendo a lacuna entre abordagens de software ágil e livre: o impacto da corrida”**. In: International Journal of Open Source Software and Processes (IJOSSP) 1 (Jan. 2009), pp. 58–71.
- ARAÚJO, Daniela de. **Feminismo e cultura hacker: intersecções entre política, gênero tecnologia**. 2018. 147f. Tese (doutorado em Política Científica e Tecnológica) – Instituto de Geociências, Universidade Estadual de Campinas, Campinas, 2018.
- B. Trinkenreich, M. Guizani, I. Wiese, T. Conte, M. A. Gerosa, A. Sarma, and I. Steinmacher, **“The pot of gold at the end of the rainbow: What is success for open source contributors?”**. Disponível em: <https://www.researchgate.net/publication/351709487_The_Pot_of_Gold_at_the_End_of_the_Rainbow_What_is_Success_for_Open_Source_Contributors>, 05 2021. Acesso em: novembro de 2021.
- COLEMAN, Gabriella. **Revoluções Silenciosas: o irônico surgimento do software livre e de código aberto e a constituição de uma consciência legal hacker**. In: COLEMAN, Gabriella. Do regime de propriedade intelectual: estudos antropológicos. PortoAlegre: Tomo Editorial, 2010.
- DEBIAN, “Wiki debian.” Freedombox Contribute. Disponível em: <<https://wiki.debian.org/FreedomBox/Contribute>>. Acesso em: dezembro de 2021.
- DELFANTI, Alessandro. **Biohackers: The Politics of Open Science**. Londres: Pluto Press, 2013. Beams of particles and papers. How digital preprint archives shape authorship and credit. In: arXiv:1602.08539 [physics], 26 fev. 2016. Disponível em: <<https://arXiv.org/pdf/1602.08539.pdf>>. Acesso em: outubro 2022.
- FREEDOMBOX, “FreedomBox Forum”, FreedomBox team. Disponível em: <<https://discuss.freedombox.org/>>. Acesso em: dezembro 2021.
- GITLAB, “Salsa debian.” Freedombox team. Disponível em: <<https://salsa.debian.org/freedombox-team/freedombox/-/blob/master/HACKING.md>>. Acesso em: dezembro de 2021.

GNU. GNU Operating System, 2022. "What is Free Software?". Disponível em: <<https://www.gnu.org/philosophy/free-sw.html.en>>. Acesso em: julho 2022.

RAYMOND, Eric. S. *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol: O'Reilly Media, 1999.

ROBSON, Colin E MCCARTAN. Colin Robson e Kieran McCartan. **Pesquisa do mundo real: um recurso para usuários de métodos de pesquisa social em ambientes aplicados**. 2016, pág. 533. isbn: 111874523X.

SARMENTO, Manuel Jacinto (2011) "**O Estudo de Caso Etnográfico em Educação**" In N. Zago; M. Pinto de Carvalho; R. A. T. Vilela (Org.) *Itinerários de Pesquisa - Perspectivas Qualitativas em Sociologia da Educação* (137 - 179). Rio de Janeiro: Lamparina (2ª edição)

STALLMAN, Richard. **Sobre o sistema operacional GNU**. Disponível em: <<https://www.gnu.org/gnu/about-gnu.html>>. Acesso em: dezembro 2021.

STEINMACHER, Igor; Graciotto Silva, Marco Aurélio & Gerosa, Marco Aurelio. (2014). **Barriers Faced by Newcomers to Open Source Projects: A Systematic Review**. IFIP Advances in Information and Communication Technology. 427. 10.1007/978-3-642-55128-4_21.

STEINMACHER, Igor; WIESE, Igor; CHAVES, Ana Paula e GEROSA, Marco Aurélio. "**Why do newcomers abandon open source software projects?**" in 2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), pp. 25–32, 2013.

WEN, Melissa Shihfan Ribeiro, "**What happens when bazaar grows.**" https://www.teses.usp.br/teses/disponiveis/45/45134/tde-07092021-041136/publico/dissertacao_mestrado_final_melissa_wen_10662130.pdf, 04 2021. Acesso em: dezembro de 2021.