

INSTITUTO FEDERAL FLUMINENSE
PROGRAMA DE PÓS-GRADUAÇÃO EM PROPRIEDADE INTELECTUAL E
TRANSFERÊNCIA DE TECNOLOGIA PARA INOVAÇÃO

VITOR DA SILVA GONÇALVES

FATORES INFLUENCIADORES DO SUCESSO EM PROJETOS DE SOFTWARE

LIVRE:

UM ESTUDO DE CASO DA DESCONTINUAÇÃO DO FIREFOX OS

Campos dos Goytacazes

2023

VITOR DA SILVA GONÇALVES

FATORES INFLUENCIADORES DO SUCESSO EM PROJETOS DE SOFTWARE

LIVRE:

UM ESTUDO DE CASO DA DESCONTINUAÇÃO DO FIREFOX OS

Dissertação apresentada como requisito parcial para obtenção do título de Mestre Programa de Pós-Graduação em Propriedade Intelectual e Transferência de Tecnologia para Inovação - PROFNIT-Ponto Focal Instituto Federal Fluminense - IFF

Orientador: Dr Samuel Alex Coelho Campos

Campos dos Goytacazes

2023

Dados Internacionais de Catalogação na Publicação (CIP)

G635f *Gonçalves, Vitor da Silva*

Fatores influenciadores do sucesso em projetos de software livre: um estudo de caso da descontinuação do Firefox OS / Vitor da Silva Gonçalves - 2023.

116[22] f.

Orientador: Samuel Alex Coelho Campos

Dissertação (mestrado) – Instituto Federal de Educação, Ciência e Tecnologia Fluminense, Curso de Mestrado Profissional de Propriedade Intelectual e Transferência de Tecnologia para a Inovação (PROFNIT), Campos dos Goytacazes, RJ, 2023.

1. Software livre. 2. Firefox OS. 3. Código aberto. 4. Android. I. Campos, Samuel Alex Coelho, orient. II. Título.

VITOR DA SILVA GONÇALVES

FATORES INFLUENCIADORES DO SUCESSO EM PROJETOS DE SOFTWARE

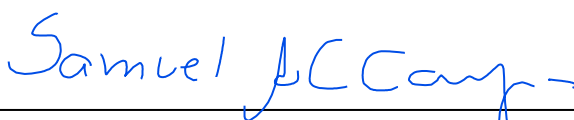
LIVRE:

UM ESTUDO DE CASO DA DESCONTINUAÇÃO DO FIREFOX OS

Dissertação apresentada como requisito parcial para obtenção do título de Mestre Programa de Pós-Graduação em Propriedade Intelectual e Transferência de Tecnologia para Inovação - PROFNIT-Ponto Focal Instituto Federal Fluminense - IFF

Aprovada em 12/04/2023

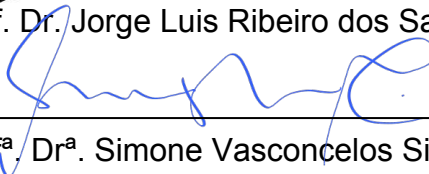
BANCA EXAMINADORA



Prof. Dr. Samuel Alex Coelho Campos



Prof. Dr. Jorge Luis Ribeiro dos Santos



Prof^a. Dr^a. Simone Vasconcelos Silva

Prof. Me. Felipe Do Espirito Santo

DEDICATÓRIA

Dedico este trabalho a todos que contribuem para o bem comum da humanidade por meio da disponibilização de suas obras sob licenciamento aberto.

RESUMO

O *software* livre tem alcançado relevante sucesso, com a maioria das empresas utilizando e contribuindo em algum grau com o seu desenvolvimento. Entretanto, quando se observa individualmente esse tipo de projeto, nota-se que muitos falham em se estabelecer no mercado. Nesse sentido, este trabalho realizou uma revisão sistemática sobre fatores que influenciam no sucesso de projetos de *software* livre, assim como realizou um estudo de caso sobre a descontinuação do sistema operacional Firefox OS, comparativamente com o projeto Android. A revisão sistemática foi baseada na metodologia PRISMA, partindo de 1062 artigos das bases de dados Scopus e Web of Science, que foram filtrados com o apoio da ferramenta AsReview para um total de 16 incluídos na pesquisa. Já o estudo de caso, foi realizado com dados documentais e bibliográficos, dados do repositório Git dos projetos, dados de interesse de busca da plataforma Google Trends, e comentários da Rede social Reddit escritos entre novembro de 2007 a dezembro de 2018. Os comentários foram processados por meio de técnica de análise de sentimento VADER, que analisou um total de 2142706 comentários, sendo os 200 de sentimento mais intenso selecionados para análise manual, o que permitiu coletar e analisar a opinião do público em relação a cada projeto. Como resultados, identificou-se que os principais fatores motivadores do sucesso em projetos de *software* livre estão relacionados a fenômenos de rede, execução de tarefas, tipo de licença, portabilidade, engajamento da comunidade, traduções, utilização de linguagens de programação populares, diversificação dos papéis, e nicho de mercado. Identificou-se ainda que, dentre estes fatores, o Firefox OS teve bom desempenho, com exceção da portabilidade, mas que o fator crítico para sua descontinuidade foi a disparidade entre sua missão com o projeto e o produto físico desenvolvido, além da não observância do descontentamento dos usuários.

Palavras-Chave: Software livre; Código aberto; Fatores de sucesso; Firefox OS; Android.

ABSTRACT

Free and open source software has achieved significant success, with most companies using and contributing to its development to some extent. However, when individual open source software projects are examined, it's noticed that many of them fail to establish their product into the market. In this regard, this work conducted a systematic review on factors that influence the success of open source software projects, as well as a case study on the discontinuation of the Firefox OS operating system, comparatively with the Android project. The systematic review was based on the PRISMA methodology, starting from 1062 articles from the Scopus and Web of Science databases, which were filtered with the support of AsReview to a total of 16 included in the research. The case study was conducted with documentary and bibliographic data, data from the Git repository of the projects, data of search interest from the Google Trends platform, and comments from the Reddit social network written between November 2007 and December 2018. The comments were processed using the VADER sentiment analysis technique, which analyzed a total of 2,142,706 comments. The 200 with the most intense sentiment values were selected for manual analysis, allowing the collection and analysis of public opinion regarding each project. As a result, it was identified that the main motivating factors for success in open source software projects are related to network phenomena, task execution, license type, portability, community engagement, translations, use of popular programming languages, role diversification, and market niche. It was also identified that, among these factors, Firefox OS performed well, except for portability, but the critical factor for its discontinuation was the discrepancy between its mission with the project and the physical product developed, as well as the failure to observe user dissatisfaction.

Keywords: Free software. Open source. Success factors. Firefox OS. Android

LISTA DE FIGURAS

Figura 1 - Diagrama de fluxo da metodologia PRISMA utilizada na revisão sistemática.....	31
Figura 2 - Quantidade de artigos publicados por ano sobre fatores motivadores do sucesso em projetos de <i>software</i> livre.....	37
Figura 3 - Critérios de sucesso mais utilizados na literatura com a quantidade de artigos revisados que os utilizam.....	40
Figura 4 - Fatores estudados na literatura por quantidade de artigos em que foram identificados.....	41
Figura 5 - Evolução dos tipos de licença utilizados no repositório Github entre 2008 a 2015.....	47
Figura 6 - Representação esquemática das camadas de tecnologias do sistema operacional Firefox OS.....	65
Figura 7 - Quantidade de <i>commits</i> mensais na base de código do projeto Firefox OS.....	68
Figura 8 - Número de <i>commits</i> mensais nas bases de código do Firefox OS e do Android.....	70
Figura 9 - Número de novos autores mensais nas bases de código do Firefox OS e do Android.....	71
Figura 10 - Quantidade acumulada de autores nas bases de código do Firefox OS e do Android.....	72
Figura 11 - Quantidade percentual de <i>commits</i> por autor nas bases de código do Android e do Firefox OS.....	73
Figura 12 - Percentual de <i>commits</i> por grupos de autores de contribuições mais frequentes.....	74
Figura 13 - Número de <i>apps</i> publicados na loja de aplicativos do Firefox OS entre outubro de 2013 a fevereiro de 2014.....	76
Figura 14 - Interesse de pesquisa no buscador Google para os sistemas operacionais Firefox OS e Android.....	78
Figura 15 - Quantidade de comentários no Reddit mencionando o Firefox OS para cada 1 milhão de comentários na rede.....	80
Figura 16 - Intensidade média mensal de sentimento nos comentários sobre o	

Firefox OS.....	81
Figura 17 - Proporção mensal de comentários sobre o Firefox OS por tipo de sentimento.....	82
Figura 18 - Quantidade de comentários no Reddit mencionando o Android para cada 1 milhão de comentários na rede.....	83
Figura 19 - Intensidade média mensal do sentimento nos comentários mencionando o Android.....	84
Figura 20 - Proporção mensal de comentários sobre o Android por tipo de sentimento.....	85
Figura 21 - Quantidade absoluta de comentários no Reddit mencionando os sistemas Android e Firefox OS (escala logarítmica).....	86
Figura 22 - Evolução da participação no mercado de diferentes sistemas operacionais móveis.....	93
Figura 23 - Tela principal do <i>software</i> , exibindo as opções disponíveis.....	136
Figura 24 - Imagem da tela principal com os dados preenchidos.....	138
Figura 25 - Imagem demonstrando um exemplo de saída de dados.....	139
Figura 26 - Exemplo do segundo arquivo de saída, com os números de comentários de cada tipo e as médias calculadas para cada arquivo de entrada.....	139

LISTA DE TABELAS

Tabela 1 - Relação de fatores encontrados e artigos que os abordaram.....	42
Tabela 2 - Relação de fatores influentes do sucesso no projeto Firefox OS.....	97

LISTA DE SIGLAS E ABREVIATURAS

API	Application Programming Interface
B2G	Boot to Gecko
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CSV	Comma-separated values
FSF	Free Software Foundation
JSON	JavaScript Object Notation
MWC	Mobile World Congress
NLTK	Natural Language Toolkit
OSI	Open Source Initiative
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
PROFNIT	Programa de Pós-Graduação em Propriedade Intelectual e Transferência de Tecnologia para a Inovação
RAM	Random Access Memory
RIS	Research Information Systems
SVM	Support-Vector Machine
SQL	Structured Query Language
SRDA	SourceForge Research Data Archive
VADER	Valence Aware Dictionary for Sentiment Reasoning

SUMÁRIO

1	INTRODUÇÃO.....	12
2	JUSTIFICATIVA.....	15
2.1	ADERÊNCIA, IMPACTO, APLICABILIDADE, INOVAÇÃO E COMPLEXIDADE.....	17
3	OBJETIVOS.....	19
3.1	OBJETIVO GERAL.....	19
3.2	OBJETIVOS ESPECÍFICOS.....	19
4	REVISÃO DA LITERATURA.....	20
4.1	DEFININDO SUCESSO EM PROJETOS DE SOFTWARE LIVRE.....	20
4.2	O SUCESSO EM PROJETOS DE SOFTWARE LIVRE.....	23
4.3	DADOS DE CONTROLE DE VERSÃO, INTERESSE DE PESQUISA E ANÁLISE DE SENTIMENTO COMO RECURSOS METODOLÓGICOS. .	24
5	METODOLOGIA.....	28
6	RESULTADOS.....	37
6.1	REVISÃO SISTEMÁTICA.....	37
6.1.1	Detalhamento dos fatores identificados.....	44
6.1.2	Síntese dos fatores motivadores do sucesso.....	55
6.2	ESTUDO DE CASO.....	58
6.2.1	Apresentação.....	58
6.2.2	Histórico e recepção na mídia digital.....	59
6.2.3	Tecnologia e desenvolvimento do projeto.....	64
6.2.4	Indicadores técnicos.....	67
6.2.5	Indicadores mercadológicos.....	75
6.2.6	O desafio do Firefox OS diante do crescimento do Android.....	92
6.2.7	Sumarização.....	96
7	CONSIDERAÇÕES FINAIS.....	99
8	ENTREGÁVEIS DE ACORDO COM OS PRODUTOS DO TCC.....	103
	REFERÊNCIAS.....	104
	APÊNDICE A – MATRIX FOFA (SWOT).....	118
	APÊNDICE B – SCRIPTS UTILIZADOS NA METODOLOGIA DO TRABALHO.....	119

APÊNDICE C – CÓDIGO-FONTE DO PRODUTO DESENVOLVIDO...	128
APÊNDICE D – DOCUMENTAÇÃO DO PRODUTO DESENVOLVIDO	133

1 INTRODUÇÃO

O movimento do *software* livre se originou como uma ação de preservação da então conhecida “cultura hacker”, caracterizada pela dedicação ao desenvolvimento e modificação de *software* e *hardware* para implementar novas funcionalidades, além do cultivo de uma cultura de compartilhamento de código entre seus pares (RAYMOND, 1999a; WILLIAMS, 2002). Essa cultura se perdia, à medida que as restrições jurídicas em torno do *software* se intensificavam, acompanhando a tendência que se dava em torno da propriedade intelectual como um todo. Este tipo de restrição jurídica levou parte dos autores a se questionarem quanto à possibilidade de novas formas de gerenciamento da sua propriedade intelectual, permitindo acesso mais distribuído e facilitando a reutilização para o desenvolvimento de novas obras (BOLLIER; HELFRICH, 2012).

A mesma linha de questionamento se deu entre desenvolvedores de *software*, que construíram um movimento cuja formalização se deu em 1984, guiada por Richard Stallman, que definiu as quatro liberdades essenciais para um *software* ser livre, sendo elas a liberdade de copiar, executar, modificar e redistribuir (STALLMAN; GAY, 2009). O movimento se difundiu pelo mundo, agregando uma base de obras produzidas, desenvolvedores ativos e usuários, formando o que se conhece como uma comunidade de desenvolvimento e compartilhamento de *software* (DULONG DE ROSNAY; STALDER, 2020). Este movimento levou à criação da “Free Software Foundation” (FSF), uma fundação de apoio, responsável por guiar e estruturar o movimento.

Um movimento paralelo surgiu na década seguinte, baseado nas ideias de que o teor ideológico do movimento do *software* livre não era atraente para o mercado, e se objetivando a levar o conceito de *software* livre para o mundo corporativo. Este novo movimento cunhou o termo “open source”, ou código aberto, como uma forma de remover a ambiguidade do termo “free software” na língua inglesa (RAYMOND, 1999b). Assim como no movimento do *software* livre, o movimento do código aberto também levou à criação de uma organização, a “Open Source Initiative” (OSI), que se envolve com a comunidade, mantém sua própria

definição de código aberto e um conjunto de licenças compatíveis com sua ideologia.

O principal ponto de discordância entre os dois movimentos se dá devido ao que Stallman (2009) descreveu como o movimento do código aberto se tratar de uma metodologia de desenvolvimento de *software* de forma colaborativa, enquanto o *software* livre não se trata apenas de uma metodologia, mas de um movimento social. O autor ainda enfatiza que, apesar dos desacordos ideológicos, não há um radicalismo ou rivalidade entre os movimentos e que suas visões levam, em muitos casos, às mesmas ações práticas. Já a OSI ([s.d.]), declarou que os termos *software* livre e código aberto são dois termos similares, apenas se diferenciando historicamente e focando em diferentes audiências. Desse modo, considerando a convergência dos termos em seu sentido prático, ambos foram utilizados indistintamente ao longo deste trabalho.

Projetos de *software* livre se mostram importantes para a sociedade no sentido que promovem a liberdade de executá-lo, estudá-lo, modificá-lo e distribuí-lo sem restrições. Estes aspectos levam a uma maior transparência e colaboração, incentivando o compartilhamento de conhecimento e permitindo que comunidades de desenvolvedores trabalhem juntas para melhorar o *software*. São ainda importantes para a democratização do acesso à tecnologia, uma vez que sua distribuição é comumente realizada gratuitamente e permite que os usuários os adaptem às suas condições de vida.

O desenvolvimento e uso de *software* livre se tornaram práticas populares em decorrência de ambos movimentos apoiadores, a ponto de se tornarem dominantes no mercado, como apontado por Wang *et al.* (2012). Os autores apresentaram indicadores da presença do *software* livre no mercado, demonstrando seu sucesso, mas ponderaram que, apesar deste sucesso do *software* livre como um todo, quando se observa os projetos a um nível individual, muitos deles não se tornam bem-sucedidos.

A dificuldade do desenvolvimento bem-sucedido de projetos de *software* livre se torna evidente quando se observa que a maioria dos projetos criados não são continuados após alguns meses da sua publicação (BERKHOLZ, 2014), e que, dentre os projetos ativos, 85% possuem baixo ou nenhum nível de atividade

(CAPILUPPI; LAGO; MORISIO, 2003). A baixa taxa de sucesso em projetos de *software* livre é ainda identificada por Coelho e Valente (2017), que selecionaram os 5000 projetos mais populares da plataforma Github e, ainda assim, identificaram um percentual de 13% de projetos sem atividade no último ano pesquisado.

Estas características vêm chamando a atenção de desenvolvedores e pesquisadores da área como um sinal de alerta, incentivando a busca pela compreensão do fenômeno. Com isso, se tornam úteis estudos que busquem identificar fatores que motivem o sucesso de projetos de *software* livre, como uma forma de compreender quais elementos podem ser observados e/ou manipulados a fim de elevar as chances de êxito desses projetos.

Nesse sentido, o presente trabalho propõe a identificação de fatores que motivam o sucesso de projetos de *software* livre, por meio de uma revisão sistemática da literatura. O trabalho propõe ainda a realização de um estudo de caso do projeto de *software* livre descontinuado¹ Firefox OS, realizado de forma comparativa em relação ao projeto Android, por este ser um *software* livre para o mesmo segmento e um caso de sucesso, uma vez que é o sistema operacional mais utilizado em todo o mundo (STATCOUNTER, 2022a). Desse modo, o trabalho busca compreender de que forma os fatores identificados foram trabalhados no projeto, além da identificação de possíveis fatores ainda não suficientemente explorados na literatura. Para isso, foram levantados dados e informações por meio da revisão da literatura, indicadores técnicos obtidos a partir de repositórios Git, indicadores mercadológicos obtidos a partir de comentários em rede social e dados da plataforma Google Trends,

¹Nesse contexto, a descontinuidade se deu devido à inviabilização da sua continuidade, e não apenas por decisão dos seus desenvolvedores.

2 JUSTIFICATIVA

Em um mundo no qual os meios de comunicação, ferramentas de trabalho, equipamentos, veículos e até mesmo sistemas críticos, como sistemas médicos, financeiros, governamentais, entre outros, funcionam com base em *software*, este pode ser considerado um dos pilares da sociedade contemporânea. Isto o torna parte fundamental da inovação, como apontado por Quinn *et al.* (1997). Os autores discutiram sobre a presença do *software* em todas as etapas do processo inovativo e, já no fim da década de 1990, sendo de uso majoritário para a inovação.

Este pensamento foi reforçado por Sousa *et al.* (2014), que apontaram o progresso técnico como vital para a economia e consideraram, para essa finalidade, o uso de *software* livre como uma forma mais eficiente na geração de crescimento tecnológico e econômico. Os autores mostraram que o *software* livre possui um fator multiplicador na disponibilização de produtos no mercado e oferece um custo mais baixo de produção, ou seja, uma associação de fatores que leva a uma maior eficiência em disponibilizar produtos no mercado. Os autores chamaram ainda a atenção para o benefício dado às pequenas e médias empresas, uma vez que elas podem, com o reaproveitamento de código livre, atender a mercados que não poderiam, caso precisassem desenvolver todo o código necessário. Já Andersson *et al.* (2021), observaram uma mudança no mercado na direção de um uso intensivo de *software* na inovação, mas cuja extensão e variação ainda possuem pouca evidência empírica.

A importância do *software* para a sociedade contemporânea e a sua capacidade de facilitar a inserção de novos produtos no mercado o tornam uma importante ferramenta para o desenvolvimento tecnológico. Uma parcela do *software* desenvolvido pela indústria já é licenciado por meio de licenças abertas, como destacado por Christof (2007). O autor dissertou que, apesar do *software* livre ter sido visto como uma “comoditização” e competição aos produtos proprietários, tornou-se uma relevante fonte de inovação em várias dimensões, com ênfase nas de processo, tecnologia, qualidade, arquitetura, padrões, modelos de negócio e marketing. Já Clark (2014), relatou o amplo uso de *software* livre nas corporações,

de forma que a maioria das grandes empresas trabalha, em algum grau, com *software* livre.

Além da questão econômica, há um fator social no *software* livre, já que, com base em sua importância para a vida das pessoas, e com o auxílio das proteções jurídicas no âmbito da propriedade intelectual, o *software*, em seu formato proprietário, pode gerar desigualdades sociais e econômicas ou amplificar as existentes. A desigualdade gerada por tecnologias digitais é identificada no acesso à tecnologia, na exigência de habilidades do usuário, e até em sua forma de uso (ROBINSON *et al.*, 2020), sendo as três categorias intensificadas pelo licenciamento proprietário, uma vez que este restringe o acesso, a participação e o desenvolvimento de habilidades do usuário.

Nesse sentido, o *software* livre se transforma, não apenas em uma subcultura de compartilhamento de código entre desenvolvedores, mas em uma ferramenta de importância social e econômica. Isto se dá uma vez que permite acesso igualitário de cidadãos e países menos favorecidos à tecnologia e também ao desenvolvimento destas, já que permite partir de uma base comum compartilhada. O estudo realizado por Bhatt *et al.* (2016) demonstrou que o *software* livre pode auxiliar no progresso social de países em desenvolvimento, por permitir reduzir as barreiras socioculturais para pessoas em locais marginalizados pela digitalização e criar oportunidades de desenvolvimento tecnológico e econômico, tanto para o indivíduo, quanto para a comunidade na qual ele está inserido.

O *software* livre é ainda considerado um fator positivo para o desenvolvimento econômico e uma ferramenta que auxilia na independência tecnológica das nações, sendo este um bem público (BLIND *et al.*, 2021). Quanto à inovação, Goldman e Gabriel (2005) destacaram que nenhuma empresa ou grupo fechado possui maior capacidade inovadora do que o restante dos profissionais de uma área, podendo a inovação ser impulsionada e amplificada por meio das práticas de colaboração de *software*.

Com base no apresentado, o *software* livre se mostra uma importante ferramenta para a sociedade e para o desenvolvimento de tecnologias, com potencial para transformar o mercado de *software* e todas as áreas que deste dependem, além de possibilitar uma aceleração e democratização na inovação

tecnológica. Entretanto, para que os benefícios do uso de *software* livre para a sociedade sejam devidamente alcançados, é preciso que as iniciativas tenham sucesso em suas implementações.

Nesse sentido, criar um projeto de *software* livre inovador se apresenta uma tarefa complexa e desafiadora, mas que, se amparada por estudos científicos e técnicas baseadas nestes, pode ter sua expectativa de sucesso elevada. Com isso, é possível que mais projetos livres atuem competitivamente no mercado e que mais cidadãos se beneficiem de um acesso mais igualitário à tecnologia.

2.1 ADERÊNCIA, IMPACTO, APLICABILIDADE, INOVAÇÃO E COMPLEXIDADE

O movimento do *software* livre baseia-se nos conceitos de propriedade intelectual, a temática principal do programa de mestrado do PROFNIT. Esse movimento confronta a ideia de que as restrições jurídicas são fundamentais para o incentivo à criação de novas obras e propõem um modelo baseado no compartilhamento e na colaboração. Ao trabalhar esta temática, o presente trabalho se enquadra na linha de propriedade intelectual do PROFNIT. Do mesmo modo, ao utilizar de metodologias inovadoras para esta área de estudo, o trabalho pode também se enquadrar na linha de inovação.

O assunto já foi trabalhado no programa de mestrado, no estudo realizado por Ferreira (2020), que construiu diretrizes para escolha de licenças nos projetos de *software* livre. Desse modo, o presente trabalho pode ampliar a base de estudos da área no PROFNIT, uma vez que é possível ampliar as diretrizes, encontrando relações ente o adequado licenciamento de um projeto e seu sucesso no mercado.

Este estudo pode ser classificado, segundo os critérios do programa de mestrado, como uma produção de médio teor inovativo, uma vez que combina conhecimentos presentes na literatura e agrega novos conhecimentos, advindos dos resultados obtidos. Considerando a complexidade e diversidade dos projetos de *software* livre, as áreas potencialmente impactadas pelo projeto são as áreas tecnológicas, sociais e econômicas, mas não se limitando a estas.

O trabalho visa atender a uma demanda espontânea, advinda da escassez de embasamento técnico-científico disponível para a tomada de decisão em projetos de *software* livre, mencionada previamente, de forma que os gestores de projetos possam utilizar dos conhecimentos resultantes deste trabalho para elevar suas chances de sucesso, configurando assim, sua aplicabilidade.

Atualmente, o repositório de projetos de *software* livre Github consta com mais de 200 milhões de repositórios (GITHUB, 2022a), e a tendência ao uso e desenvolvimento de *software* livre nas empresas já foi identificado por especialistas, como LeClair (2016), que apontou um percentual de 65% das empresas contribuindo para o *software* livre, e o relatório anual da Red Hat (2021), que apontou um percentual de 90% dos líderes da área de tecnologia de informação trabalhando com código aberto. Desse modo, estima-se um potencial de abrangência ampla para os resultados da pesquisa, permitindo sua utilização por quaisquer indivíduos que estejam a frente de um projeto de *software* livre, desde projetos de menor porte a grandes projetos.

A complexidade da pesquisa pode ser classificada, conforme os critérios do PROFNIT, como de complexidade média, uma vez que a produção resultou de combinação de conhecimentos preestabelecidos com as novas informações obtidas. É importante destacar que, utilizando da metodologia de estudo de caso e estudando entidades dinâmicas, uma vez que muitos dos projetos seguem em atividade, a complexidade da pesquisa variou ao longo da sua elaboração. Dese modo, a partir da obtenção inicial de resultados, se apresentou necessária a utilização de novas abordagens e do envolvimento de novos atores. Isto levou a uma considerável elevação da complexidade estimada.

3 OBJETIVOS

3.1 OBJETIVO GERAL

Analisar os fatores que influenciam o sucesso de projetos de *software* livre e observar seus efeitos sobre a descontinuidade do projeto Firefox OS, identificando a relevância de cada um e a possibilidade da existência de fatores não explorados na literatura científica.

3.2 OBJETIVOS ESPECÍFICOS

- Realizar uma revisão sistemática da literatura, visando identificar e agregar os fatores facilitadores ou dificultadores para o sucesso em projetos de *software* livre;
- Realizar um estudo de caso sobre o projeto Firefox OS, buscando identificar que fatores levaram à sua descontinuidade;
- Comparar os dados coletados sobre o Firefox OS com dados do projeto Android;
- Verificar se os fatores abordados na literatura científica foram decisivos para o Firefox OS;
- Verificar se houve fatores decisivos para o Firefox OS não explorados na literatura pesquisada.

4 REVISÃO DA LITERATURA

4.1 DEFININDO SUCESSO EM PROJETOS DE SOFTWARE LIVRE

O termo sucesso é usado em diferentes contextos e possui algum nível de subjetividade, de tal forma que, para uma pesquisa científica, é necessário defini-lo para haver consistência ao longo do trabalho e, posteriormente, para a comparação de resultados de trabalhos diferentes.

Agarwal e Rathod (2006) observaram que as definições de sucesso de projetos de *software* utilizadas no mercado e na academia costumavam se baseavam na capacidade deste em atingir um determinado nível de qualidade em margens aceitáveis de tempo e custo. Contudo, os autores apontaram que estes critérios são de difícil mensuração, tornando estes tipos de definição inconsistentes. Em seu trabalho, os autores realizaram uma pesquisa com desenvolvedores, gestores e usuários, para identificar que critérios consideravam importantes para o sucesso de um projeto de *software*. Os resultados obtidos apontaram que os principais critérios são o custo, tempo e escopo, além de apontarem alguns critérios considerados de menor relevância, como a satisfação dos usuários e prioridades específicas do projeto.

Esses principais critérios observados tratam-se do que é conhecido como “triângulo de ferro”, dentre outras nomenclaturas. Nesse triângulo, são colocadas as 3 principais características que definem o sucesso de um projeto: custo, qualidade e tempo. As características estão inter-relacionadas, de forma que há um contrabalanço na otimização de cada uma dessas áreas, sendo encontrar o equilíbrio adequado, um desafio para a gestão. O modelo é conhecido desde meados do século XX e se tornou de amplo uso na área de gerenciamento de projetos, o que pode explicar sua aplicação em projetos de *software*. Seu uso recebeu críticas devido ao seu foco na entrega do projeto, tornando-se, desse modo, um modelo limitado (ATKINSON, 1999).

Baseados nestas definições, foram elaborados alguns modelos de medição

de sucesso em projetos de *software*, sendo um dos mais relevantes o elaborado, no início da década de 1990, por DeLone e McLean (2003), e revisado posteriormente pelos mesmos autores. Trata-se de um modelo multidimensional e baseado em considerações causais e de processo, propondo 6 dimensões inter-relacionadas de sucesso, sendo elas a qualidade do sistema; a qualidade informacional; o uso; a satisfação dos usuários; o impacto individual e o impacto organizacional. Além de definir as dimensões, o modelo propôs uma relação linear entre elas, partindo das qualidades para o uso e satisfação e seguindo para os impactos. O modelo foi amplamente utilizado em artigos científicos no período desde sua primeira publicação até sua revisão, segundo os mesmos autores, como pode ser observado em Wei *et al.* (2009), Němec (2011) e Seliana *et al.* (2020).

Na revisão do modelo, tanto as dimensões, quanto suas relações, foram reestruturadas, sendo então formado pelas dimensões de qualidade informacional; qualidade de sistema; qualidade de serviço; uso/intenção de uso; satisfação dos usuários e benefícios em rede. A estrutura de inter-relações já não mais foi apresentada em um modelo linear, mas com diferentes caminhos e realimentações.

O modelo descrito foi ainda estendido por Westhuizen e Fritzgerald (2005). Os autores buscaram, na teoria de sucesso de projetos, fatores que não foram considerados no modelo de DeLone e McLean, o incrementando e construindo um novo modelo, que adiciona o fator temporal e a separação entre gerenciamento de projeto e produto, além de trazer novas dimensões como critérios de sucesso. Esse novo modelo foi elaborado para satisfazer os critérios da literatura para um modelo de sucesso em projetos. Segundo os autores, o novo modelo adicionou mais complexidade, sendo constituído de 10 dimensões, e alguns critérios tiveram que ser desconsiderados, para que não o tornasse inviável.

Embora as definições de sucesso em projetos de *software* baseadas no modelo do triângulo de ferro sejam amplamente utilizadas, tanto no mercado quanto na academia, quando se trata de projetos de *software* livre, há algumas características que os diferenciam de projetos convencionais, inviabilizando o uso deste tipo de definição.

Projetos de *software* livre nem sempre são desenvolvidos comercialmente. Por isso, seus custos e tempo são, muitas vezes, relativizados, além de possuírem

modelos de colaboração, permitindo contribuições externas e constante entrada de novos desenvolvedores, com possibilidade de novos grupos darem continuidade ao projeto, caso os desenvolvedores originais o abandonem, ou até mesmo a manutenção de projetos paralelos utilizando a mesma base de código. Essas características destacam a necessidade de uma definição diferente para o sucesso de um projeto de *software* livre.

Visando a elaboração de um modelo para medição de sucesso em projetos de *software* livre, Crowston *et al.* (2003) realizaram uma pesquisa com desenvolvedores e usuários de projetos de código aberto, questionando-os sobre quais aspectos consideravam que podem ser utilizados como indicadores do sucesso deste tipo de projeto. Foram identificadas 7 categorias de critérios, sendo elas: usuário, produto, processo, desenvolvedores, uso, reconhecimento e influência. Dentro destas categorias, foram identificados 22 critérios, com um destaque para os de reconhecimento, envolvimento dos usuários e portabilidade.

Ghapanchi *et al.* (2011) realizaram uma pesquisa de revisão baseada na literatura científica e identificam categorias apresentadas como critérios para medição do sucesso em projetos de *software* livre, separando-as entre sucesso do produto (qualidade e interesse dos usuários) e do projeto (eficiência, eficácia, atividade e interesse dos usuários). Os autores enfatizaram que os critérios identificados possuem um aspecto mais genérico, pois os critérios específicos estabelecidos podem variar entre projetos, já que estes possuem grande diversidade. Entretanto, apresentaram alguns critérios práticos que podem ser utilizados como guias.

Já Sen *et al.* (2012), utilizaram um modelo de medição do sucesso baseado unicamente na base de desenvolvedores e na base de seguidores, sendo esta última composta por pessoas interessadas que acompanham o projeto (não necessariamente usuárias do mesmo). O modelo, embora utilize de uma menor quantidade de variáveis, pode ser mensurado objetivamente e apresentou-se como um modelo viável para o estudo.

Desse modo, compreende-se que o sucesso em projetos de *software* livre é de complexa definição, necessitando da realização de simplificações para a sua operacionalização, para permitir medir o nível de sucesso de um projeto. Diferentes

estudos tomam diferentes abordagens, normalmente evitando a realização de classificações entre bem sucedidos e mal sucedidos, mas sim comparando projetos como mais sucedidos ou menos sucedidos. Entretanto, alguns casos são de mais fácil classificação, principalmente quando há o abandono ou a descontinuidade de um projeto.

O presente trabalho, portanto, utiliza como critério para o sucesso de um projeto, a sua manutenção estável no mercado, enquanto para o insucesso, a sua descontinuidade. Estes critérios foram adotados devido à sua compatibilidade com diferentes definições e a sua objetividade.

4.2 O SUCESSO EM PROJETOS DE SOFTWARE LIVRE

Durante os primeiros anos de popularização do *software* livre, havia ceticismo sobre a capacidade desta então nova forma de desenvolvimento e licenciamento de *software* para a criação e manutenção de um projeto em si. Ou seja, havia ceticismo em relação à sua viabilidade como método. Desta forma, os primeiros trabalhos relacionados tenderam a discutir a possibilidade de desenvolver um projeto de *software* livre bem-sucedido, como demonstrado no trabalho de Hars e Ou (2000), no qual os autores investigaram os fatores que levam uma pessoa a colaborar com projetos abertos. No modelo construído pelos mesmos, foram apresentadas as fontes de motivação identificadas: motivação intrínseca, satisfação de necessidades pessoais e expectativa de retorno futuro.

Até o início dos anos 2000, o *software* livre levantava questionamentos a respeito da sua praticabilidade econômica. Bonaccorsi e Rossi (2003) realizaram uma análise da motivação, coordenação e difusão do *software* livre. Os autores destacaram que a transição do *software* de um bem privado para público era um novo desafio para as teorias econômicas e as mesmas não conseguiam explicar alguns fenômenos, como as pessoas que realizam trabalhos considerados desagradáveis sem incentivo para tal. Contudo, os autores enfatizaram que o *software* livre é um fenômeno do mundo real, que cresce e independia da sua

compreensão teórica para seu sucesso, sendo o uso de sistemas híbridos (parcialmente livre e parcialmente proprietário) um caminho utilizado por muitas empresas e que provava sua viabilidade econômica. Também em uma linha ceticista em relação à viabilidade do *software* livre, Dreiling *et al.* (2005), abordaram o seu uso em sistemas *enterprise* e discutiram de que forma o *software* livre poderia se consolidar nessa indústria, além dos possíveis cenários para o seu desenvolvimento.

Foi somente a partir dos meados da década de 2000 que os estudos passaram a tratar o sucesso de projetos de *software* livre de uma forma diferente, não questionando a possibilidade de criar um projeto bem-sucedido em si, mas analisando os fatores que poderiam contribuir para o sucesso dos mesmos. Nesse sentido, foram desenvolvidos trabalhos em que autores experientes em grandes projetos de *software* livre discutiram os fatores de sucesso com base em suas experiências pessoais, como no trabalho de Fogel (2005). No meio científico, já se desenvolvia, nesse mesmo período, trabalhos sistematizados para identificar tais fatores, como os realizados por Grewal *et al.* (2006) e Wray *et al.* (2009).

Esses trabalhos contribuíram para a área, no sentido de fornecer critérios para a metrificação do sucesso de um projeto, assim como contribuíram para a construção de diretrizes para iniciativas de *software* livre. Contudo, os obstáculos enfrentados por novos projetos (STEINMACHER; GRACIOTTO SILVA; GEROSA, 2014) e suas baixas taxas de sucesso (COELHO; VALENTE, 2017) indicam que a área ainda carece de estudos, principalmente em relação a projetos que não foram bem-sucedidos, no sentido de encontrar os fatores críticos que contribuíram para o seu desfecho.

4.3 DADOS DE CONTROLE DE VERSÃO, INTERESSE DE PESQUISA E ANÁLISE DE SENTIMENTO COMO RECURSOS METODOLÓGICOS

O uso de sistemas de controle de versão para desenvolvimento de *software* acarreta geração de metadados que podem ser explorados metodologicamente para a obtenção de informações e construção de indicadores sobre um projeto. A

literatura científica vem abordando esse tema no sentido de encontrar novas e mais eficientes formas de extrair informações de repositórios de *software*.

Em um dos trabalhos pioneiros da área, Weissgerber *et al.* (2007) desenvolveram técnicas de visualização que permitem identificar padrões de desenvolvimento dos programadores e o particionamento de tarefas entre eles. Sendo pioneiro, esse estudo ainda se baseava principalmente em avaliar e construir métodos de análise manual.

Em uma abordagem automatizada, alguns trabalhos analisaram ou desenvolveram técnicas de mineração em repositórios de código para auxiliar no desenvolvimento do próprio *software*, identificando falhas, como no trabalho realizado por Mahmood *et al.* (2021). Os autores analisaram a eficiência das regras de associação, propondo um modelo e o verificando, para demonstrar sua capacidade de auxiliar gestores de projetos de *software* a encontrarem falhas no mesmo.

Com um foco maior no campo científico, Yang *et al.* (2016) apresentam uma estrutura para análise de repositórios que utilizam o sistema de controle Git e extração de dados referentes ao produto, o processo, e às pessoas envolvidas. Essa estrutura consiste da elaboração de *scripts* Python para a extração de dados do repositório e seu armazenamento em banco de dados do tipo SQL. Após armazenados, os dados podem ser acessados e analisados utilizando consultas SQL.

Uma ferramenta livre que implementa uma estrutura similar à elaborada por Yang *et al.* (2016), é a MergeStat (2022). A ferramenta abstrai o repositório Git para o usuário e oferece acesso direto às informações por meio de *queries* SQL, facilitando o trabalho para pesquisadores, que passam a não precisar interagir diretamente com os repositórios ou se adentrar no funcionamento dos mesmos. Com isso, o MergeStat foi adotado como a ferramenta a ser utilizada na metodologia deste trabalho para análise de repositórios Git.

Dados de interesse de pesquisa, por sua vez, permitem observar o quanto as pessoas pesquisam por determinados termos em buscadores da *internet*, o que está relacionado ao interesse sobre aquele assunto durante um período e em uma região geográfica. A principal plataforma que disponibiliza publicamente esse tipo de dado

é a Google Trends (2022).

Os dados da Google Trends vêm sendo utilizados cientificamente em diferentes contextos, principalmente na área da saúde, como nos trabalhos de Carneiro e Mylonakis (2009) e Pelat *et al.* (2009). Esses trabalhos demonstram como o interesse de pesquisa na internet pode auxiliar na identificação de surtos de doenças em alguma região, tendo sido até mesmo utilizadas como um recurso para predição de casos de COVID-19 (AYYOUBZADEH *et al.*, 2020).

Os campos de aplicação para dados de interesse de busca são diversos, e incluem o interesse do público em produtos, como estudado por Vosen e Schmidt (2011), que compararam os dados de interesse de pesquisa na *internet* com dados obtidos por aplicação de questionários para a predição de padrões de consumo. Foi observado que os resultados obtidos pelo Google Trends se demonstraram superiores em relação aos questionários aplicados.

Em uma temática semelhante, Yakubu e Kwong (2021) investigaram o potencial do uso dos dados da Google Trends em conjunto com análise de sentimento de avaliações de produtos online para auxiliar na predição de características de produtos relevantes para os consumidores no futuro. Foi realizado um comparativo com 3 métodos baseados em lógica *fuzzy* e obtidos resultados com maior acurácia. A combinação de dados de interesse de pesquisa na *internet* e análise de sentimento é comum na literatura. Foi também utilizada por Wijnhoven e Plant (2017), para a predição de preços de carros, e por Díaz e Henríquez (2021), para avaliar o sentimento social dos moradores de um município em relação às restrições de confinamento impostas como resposta à COVID-19.

A análise de sentimento pode ser compreendida como uma família de técnicas que permitem a classificação de um texto em relação ao sentimento expressado pelo autor, categorizando-o como positivo ou negativo (STINE, 2019). Sua aplicação também se dá em diferentes áreas de estudo, como categorizar avaliações de produtos (YAKUBU; KWONG, 2021; YANG *et al.*, 2020) ou obter *feedback* em processos educacionais (KECHAOU; BEN AMMAR; ALIM, 2011). Contudo, um campo de destaque da análise de sentimento é o de processamento de dados de redes sociais, como abordado por Neri *et al.* (2012), que compararam a classificação de 1000 postagens de rede social em relação a um tema com uma

pesquisa realizada por um instituto de pesquisa. Os resultados obtidos confirmaram a eficácia da análise de sentimento como metodologia para a identificação da percepção do público em relação a um assunto.

Sharma *et al.* (2020) realizaram um estudo de revisão comparando diferentes métodos para análise de sentimento e identificaram que os geradores de melhores resultados, sendo utilizados na literatura como referência, são os métodos SVM e Naive Bayes, mas que os métodos baseados em léxico também se destacaram ao apresentar bons resultados, enquanto consomem menos recursos computacionais e são mais escaláveis e fáceis de aplicar. Um resultado na mesma direção foi encontrado por Pandya e Mehta (2020), que também analisaram a literatura científica e identificaram SVM, Naive Bayes e redes neurais artificiais como os métodos com maior precisão, mas com um destaque para os métodos baseados em léxico, principalmente o VADER, que ofereceu precisão comparável aos métodos mais eficientes.

O método VADER utiliza de uma abordagem baseada em regras, considerando um léxico validado por humanos e regras relacionadas a gramática e sintaxe. O modelo calcula o que se conhece como a intensidade de sentimento de um texto, permitindo a classificação em 3 categorias: sentimento negativo, neutro e positivo (HUTTO; GILBERT, 2014). Sua relação entre precisão nos resultados e eficiência computacional são características de destaque e que levaram à utilização deste método na metodologia deste trabalho.

5 METODOLOGIA

O ponto de partida deste trabalho foi a identificação do estado da arte sobre fatores que motivam o sucesso de projetos de *software* livre. Para isso, foi realizada uma revisão sistemática com a finalidade de encontrar, na literatura científica, artigos que analisaram a influência de diferentes características de projetos e o desempenho de indicadores de sucesso.

A revisão sistemática consistiu da aquisição e seleção sistematizada de obras da literatura científica, que foram lidas e, seus resultados relevantes ao trabalho, tabulados e sumarizados, utilizando as diretrizes metodológicas PRISMA como base. Essas diretrizes tratam-se de um conjunto de 27 itens padronizados e revisados por pares para o detalhamento do processo de realização de uma revisão sistemática, visando uma maior transparência e reprodutibilidade no processo (PAGE *et al.*, 2021). Dentre os itens indicados pelas diretrizes PRISMA e utilizados no trabalho, tem-se a especificação dos critérios de inclusão e exclusão de estudos, dos bancos de dados pesquisados, da estratégia de busca, métodos utilizados para decidir se um estudo atingiram os critérios da revisão e método de coleta de dados. Nem todos os itens indicados se demonstraram aplicáveis ou relevantes para esta pesquisa. As diretrizes PRISMA apresentam ainda uma série de etapas demonstradas na Figura 1, que consistem da seleção de documentos, a triagem destes e, por fim, a inclusão dos documentos no trabalho. As etapas são detalhadas a seguir.

Foram utilizadas as bases de dados Scopus e Web of Science para a busca e obtenção dos documentos, utilizando credenciais de autenticação fornecidas pelo Instituto Federal Fluminense, por meio do sistema Cafe, mantido pela CAPES.

O termo de busca, elaborado após testes com cada palavra-chave individual, foi “project* AND ("free software" OR "open source software" OR foss OR floss) AND (success OR fail* OR obstacle* OR factor*)”, que objetivou encontrar artigos que investigaram projetos de *software* livre e seu sucesso/insucesso. A busca foi realizada em ambas bases de dados, utilizando os campos de título, *abstract* e palavras-chave, limitando-se a artigos de periódicos. A

base de dados Scopus retornou 653 resultados e a Web of Science, 409. Foi utilizada a opção de exportar os resultados como arquivo RIS, presente em ambas bases.

Em seguida, os arquivos exportados foram unidos em um único arquivo e importados para a ferramenta ASReview LAB², que identificou 19 resultados duplicados. Foi utilizado o modo “Oracle” e os parâmetros padrões para a versão utilizada (“TF-IDF”, “Naive Bayes”, “Maximum”, “Dynamic resampling”). Os parâmetros sugeridos como padrão foram estudados por Ferdinands *et al.* (2020) e apresentados como o conjunto que apresentou o melhor equilíbrio entre qualidade dos resultados e tempo de processamento. Segundo os autores, o método de extração de feições “TD_IDF” é um método comumente utilizado em processamento de texto, que associa um valor a cada palavra em cada documento, relativo ao número de documentos em que ocorreu, de forma a identificar sua raridade, como uma maneira de descobrir feições baseadas na escolha de palavras nos textos. Já o método Naive Bayes, é um algoritmo supervisionado utilizado para a classificação de texto, que considera todos os atributos como independentes. A estratégia de seleção “Maximum” escolhe para próximo documento a ser exibido para o usuário aquele com maior probabilidade de ser relevante. Por fim, o método de reamostragem “Dynamic resampling” realiza uma sobreamostragem dos documentos relevantes e subamostragem dos não relevantes, a fim de evitar um problema discutido pelos autores, chamado desbalanço de classes.

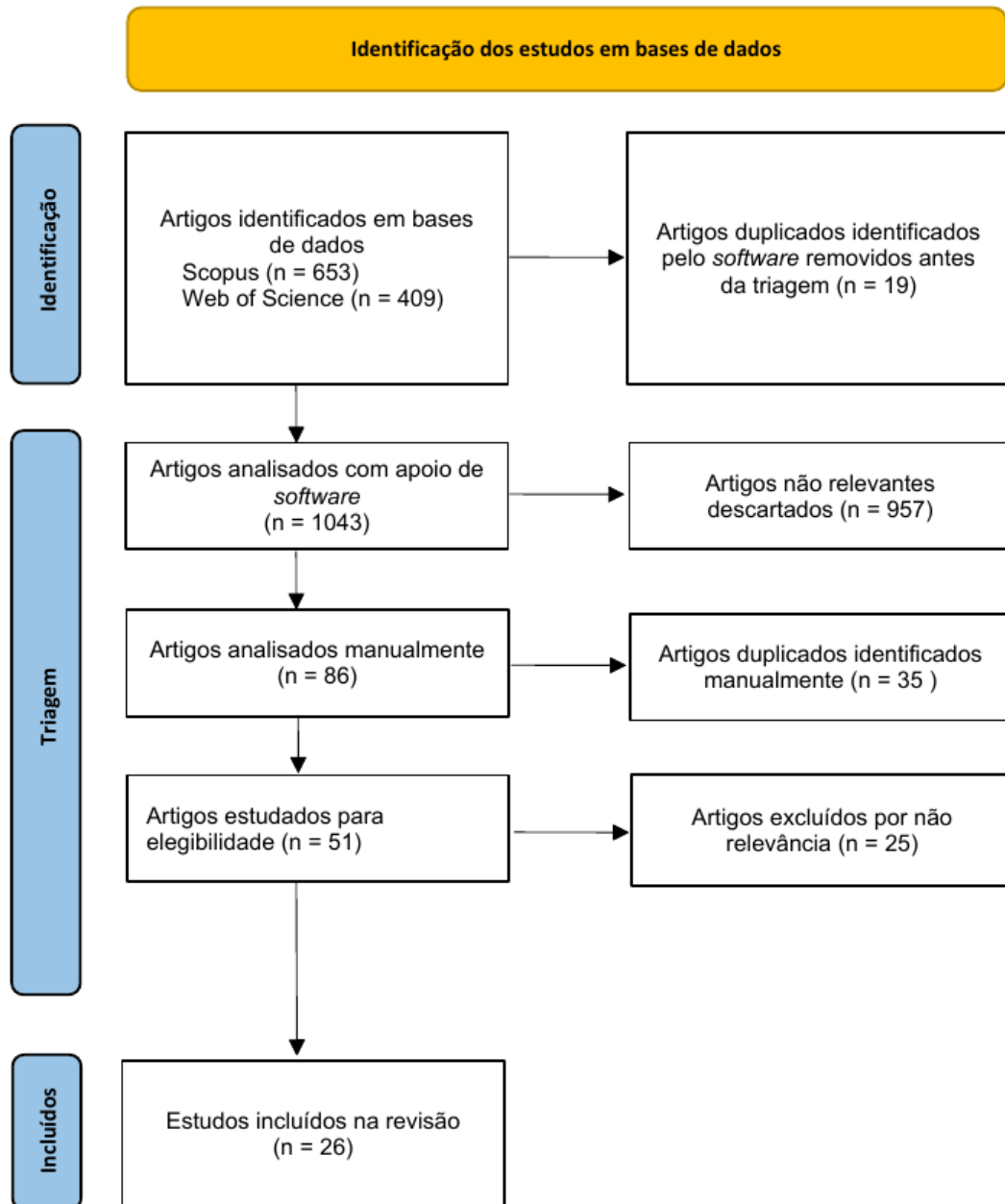
O critério para a classificação de um artigo como relevante foi a abordagem, no campo de *abstract*, da temática do sucesso em projetos de *software* livre, e, para sua exclusão, a não abordagem da mesma. O modelo foi treinado inicialmente com a identificação manual de 14 registros, sendo destes, 7 relevantes. A seleção posterior de artigos relevantes foi realizada assistidamente pelo *software*, com critério de parada estabelecido como a obtenção de 20 artigos não relevantes consecutivos.

Foram obtidos 86 artigos relevantes com o uso da ferramenta, exportados em arquivo CSV e manipulados manualmente, sendo identificados e excluídos 35

²ASReview LAB é um *software* livre projetado para acelerar o processo de revisão sistemática da literatura. Ele utiliza algoritmos de aprendizagem de máquina para auxiliar os pesquisadores a avaliar a relevância dos artigos em suas pesquisas, ao mesmo tempo em que não toma decisões automáticas, mantendo o controle do processo nas mãos do pesquisador.

artigos duplicados. A etapa seguinte consistiu da leitura dos artigos, identificando quais estudos analisaram fatores motivadores do sucesso em projetos de *software* livre, sendo estes incluídos na revisão, enquanto os artigos que trabalharam a temática, mas sem estudar o efeito de fatores individuais, foram descartados. No total desta etapa, foram descartados 25 artigos e incluídos 26, os quais foram utilizados na etapa de avaliação.

Figura 1 - Diagrama de fluxo da metodologia PRISMA utilizada na revisão sistemática



Fonte: (PAGE *et al.*, 2021) adaptado pelo autor, 2022

A avaliação dos artigos foi realizada manualmente, identificando quais fatores foram abordados em cada um dos estudos e se seus efeitos, tanto positivos quanto negativos, foram comprovados. Os dados foram extraídos e tabulados. Em seguida, foi realizado um agrupamento em categorias para os fatores obtidos.

Por fim, os resultados dos textos foram sumarizados e discutidos,

apresentando os principais fatores influenciadores do sucesso em projetos de *software* livre, seus embasamentos, e realizada a intertextualidade entre os diferentes artigos pesquisados.

Com os fatores motivadores do sucesso em projetos de *software* livre mais estudados na literatura científica tabulados, tornou-se viável a realização do estudo de caso sobre a descontinuação do Firefox OS, assim como a validação dos fatores identificados em um projeto de *software*, permitindo avaliar a sua relevância e identificar possíveis fatores relevantes, mas ainda pouco explorados cientificamente.

Conforme as diretrizes apresentadas por Ventura (2007), o estudo de caso é uma metodologia apropriada quando há uma grande quantidade de fatores e relacionamentos que podem ser diretamente observados e não existem leis básicas para determinar quais são importantes, sendo particularmente útil para o estudo de novos comportamentos ou descobertas, devido à sua capacidade para gerar hipóteses. Nesse sentido, o estudo de caso se demonstrou uma metodologia apropriada, porque o sucesso de um projeto de *software* livre é um fenômeno indissociável do seu meio, não possuindo clara distinção entre os limites entre o fenômeno e o contexto, além de possuírem inúmeras variáveis de interesse e fontes de informação.

Devido à complexidade e diversidade do objeto de estudo, se tornou necessário a obtenção de dados e reunião de informações de diferentes fontes, incluindo pesquisa bibliográfica e documental nas páginas do projeto Firefox OS, *sites* de notícias, bases de dados de artigos científicos, aquisição de dados a partir de bases de código de *software* livre, base de dados de cotas de mercado de sistemas operacionais, dados de interesse de pesquisa no motor de busca Google, e comentários de usuários da rede social Reddit.

A pesquisa se iniciou por meio de consulta pelas páginas oficiais do Firefox OS em motores de busca da *internet*. A partir das páginas encontradas, foram realizadas buscas nas mesmas, quando presente o recurso de pesquisa, para encontrar novas páginas relevantes, ou seguidos *links* presentes nas páginas. Cabe ressaltar que, tratando-se de um projeto descontinuado, muitas páginas não se encontram mais disponíveis para acesso *online* e, para contornar esta indisponibilidade, foi utilizado o serviço “Waybackmachine”, fornecido pela Internet

Archive (2022). Esse serviço arquiva capturas instantâneas de diversos *sites* na internet em diferentes datas, para permitir a realização de pesquisas históricas no acervo de documentos da *internet*.

Para buscar informações a respeito da recepção e repercussão do projeto na mídia digital e auxiliar na elaboração de uma linha do tempo, foram realizadas pesquisas também por meio de motores de busca da *internet*, mas dessa vez buscando sites de notícias e artigos de opinião contendo anúncios e resenhas sobre o Firefox OS. A relevância dessas fontes de informação se dá devido à sua ligação com a recepção do projeto no mercado, como a forma com que foi apresentada pela mídia, e com a opinião do público quanto a este. As informações obtidas foram agregadas e organizadas temporalmente. Em seguida, foram analisadas e discutidas, elaborando um histórico do projeto.

Para a obtenção de informações a respeito das atividades técnicas do projeto, foram obtidos dados da base de código arquivada do Firefox OS, desenvolvido pela Mozilla ([s.d.]), e, para a realização de comparações, foram obtidos dados da base de código do Android (2022). O repositório do Firefox OS foi clonado em massa por meio da API do Github, sendo obtidos apenas os repositórios criados diretamente pela Mozilla, e não os identificados nos metadados fornecidos pela API como *forks*³ de outros projetos, para reduzir a contabilização de *commits*⁴ criados por autores de outros projetos, sem a finalidade de desenvolver para o Firefox OS, principalmente a atividade do *kernel* Linux, utilizado no sistema, que possui um nível de atividade elevado e poderia interferir nos dados coletados. Já para o projeto Android, o código é fornecido por um sistema próprio, denominado “repo”, que permite a aquisição de toda a base de código a partir dos comandos descritos na página.

Como as bases de código de ambos projetos contém excesso de arquivos não relevantes para a pesquisa, a clonagem dos repositórios foi realizada utilizando o sufixo “--bare” do comando “git”, que permite a aquisição apenas das informações de controle de versão.

³No sistema de controle de versão Git, um *fork* é o nome dado a uma cópia de um repositório criada por um desenvolvedor em sua própria conta, permitindo que se trabalhe em uma versão separada do repositório original sem afetá-lo.

⁴Um *commit* é uma operação que registra uma nova versão de um arquivo ou conjunto de arquivos no repositório Git. Cada *commit* é identificado por um *hash* exclusivo, que permite que se retorne a um estado anterior do código, caso necessário. Ao se realizar um *commit*, é criado um registro permanente de alterações no código, incluindo informações sobre o autor, a data e hora da operação e uma mensagem de descrição das alterações realizadas.

A análise dos repositórios Git foi realizada utilizando a ferramenta MergeStat (2022), que permite a aquisição de informações dos repositórios a partir de consultas com sintaxe SQL. Foi realizada uma lista de consultas⁵, obtendo os dados de quantidade de *commits* mensais, novos autores por mês e quantidade de *commits* por autor. Os dados brutos foram processados por meio de planilha eletrônica para análise e elaboração de gráficos.

Foram ainda obtidos comentários da rede social Reddit⁶, por meio da iniciativa de arquivamento e análise de dados Pushshift (2022). Esta escolha se deu devido à possibilidade de se obter gratuitamente o conjunto completo de dados históricos desta rede, e processá-lo por meio de *software*, sem limites comumente impostos por APIs de acesso, que tornariam a coleta de dados inviável. Os conjuntos de dados obtidos é oferecido pela plataforma de arquivamento em formato JSON e organizado para cada mês. Foram baixados os conjuntos a partir de novembro de 2007, mês de anúncio público do projeto Android, até o último mês anterior à realização da pesquisa (outubro de 2022). Entretanto, devido a limitações técnicas, os dados foram processados até a data de dezembro de 2018, o que se demonstrou suficiente para a análise.

Os dados foram processados utilizando um *script* Python desenvolvido para o trabalho⁷, buscando as palavras-chave “firefox os” e “firefoxos”, sem sensibilidade de caixa, para obtenção dos comentários que discutiam o projeto Firefox OS, e as palavras-chave “Android” e “phone” simultaneamente, sem diferenciação de posição ou caixa, para obter comentários a respeito do sistema operacional Android. A restrição adicional da palavra “phone” se deu devido ao fato da palavra “android” se tratar de uma palavra comum na língua inglesa e, dessa forma, evitar a coleta de comentários sobre assuntos não relacionados ao tema.

Os 2151017 comentários extraídos do conjunto de dados foram então processados por meio de *scripts* Python, realizando a totalização de comentários do tema e utilizando a biblioteca de processamento de linguagem NLTK (2022) para a realização de análise de sentimento.

⁵Disponível em: <https://github.com/vitorsgoncalves/tcc-profnit>

⁶O Reddit é uma plataforma de rede social que permite aos usuários enviar *posts* e os discutir, por meio de comentários textuais, em comunidades temáticas denominadas “subreddits”. Os usuários podem comentar e votar, positiva ou negativamente, cada postagem ou comentário, resultando em uma classificação por popularidade e relevância.

⁷Disponível em: <https://github.com/vitorsgoncalves/tcc-profnit>

A análise de sentimento foi realizada utilizando o modelo VADER, que utiliza de uma abordagem baseada em regras, considerando um léxico validado por humanos e regras relacionadas à gramática e sintaxe. O modelo associa, a cada tipo de palavra ou elemento textual, por meio de uma espécie de tabela, um valor positivo ou negativo, e calcula, a partir da soma de todos os elementos do texto, o que se conhece como a intensidade de sentimento, permitindo a classificação em 3 categorias: sentimento negativo, neutro e positivo (HUTTO; GILBERT, 2014).

Foram então calculados os valores “compound”, ou agregado, para a intensidade de sentimento de cada comentário e calculadas as médias mensais, exportados em arquivo CSV para análise posterior. Foram então agrupados e contabilizados os comentários classificados como sentimento positivo (intensidade > 0,05), neutro ($-0,05 \leq \text{intensidade} \leq 0,05$) e negativo (intensidade < 0,05), também exportados para arquivo CSV.

Em seguida, os arquivos CSV foram processados por meio de planilha eletrônica, obtendo-se os dados de quantidade de comentários sobre o tema por mês e quantidade mensal de comentários positivos, negativos e neutros. Os dados foram analisados e, para facilitação da discussão, utilizados para a elaboração de gráficos.

A partir dos *scripts* construídos para o desenvolvimento desta dissertação, foi criado um *software* para a realização de análise de sentimento em fontes de dados no formato CSV, entregue como produto gerado pelo trabalho⁸. Este *software* foi desenvolvido com a finalidade de facilitar e agilizar o uso deste tipo de metodologia, assim como de facilitar a elaboração de novos trabalhos e permitir que pesquisadores de áreas de estudo não relacionadas à computação possam realizar análises de sentimento sem a necessidade da criação de código. O *software* consiste em uma única tela, na qual se introduz o caminho onde estão armazenados os dados textuais a serem classificados, em formato CSV, o caminho para os arquivos de saída, o nome da coluna contendo os textos no arquivo e palavras-chave de busca opcionais. Mais detalhes sobre o produto elaborado são apresentados no Apêndice D, na forma de documentação de *software*.

Para a realização de uma análise mais profunda e identificação não apenas do sentimento carregado pelos comentários, mas a identificação dos principais

⁸Disponível em: <https://github.com/vitorsgoncalves/tcc-profnit>

assuntos, foram selecionados os 50 comentários classificados com sentimento mais positivo e os 50 de sentimento mais negativo de cada projeto para leitura e análise manuais, totalizando 200 comentários.

Como dados auxiliares, foram obtidos os dados de interesse de busca no motor de pesquisa Google, por meio da plataforma Google Trends, e dados de fatias de mercado de sistemas operacionais móveis, fornecidos pela StatCounter. Os dados da Statcounter são gerados a partir da coleta de informações de dispositivos que acessam os mais de 1,5 bilhão de *sítes* utilizadores do serviço da plataforma, com localização geográfica variada (STATCOUNTER, 2022b). Já os dados do Google Trends, são elaborados a partir de uma amostra normalizada de termos de busca, data e localização geográfica das pesquisas realizadas no buscador Google, resultando em uma escala de valores entre 0 e 100, que representa a proporção de pesquisas do tópico em relação a todas as pesquisas de todos os tópicos (GOOGLE, 2022).

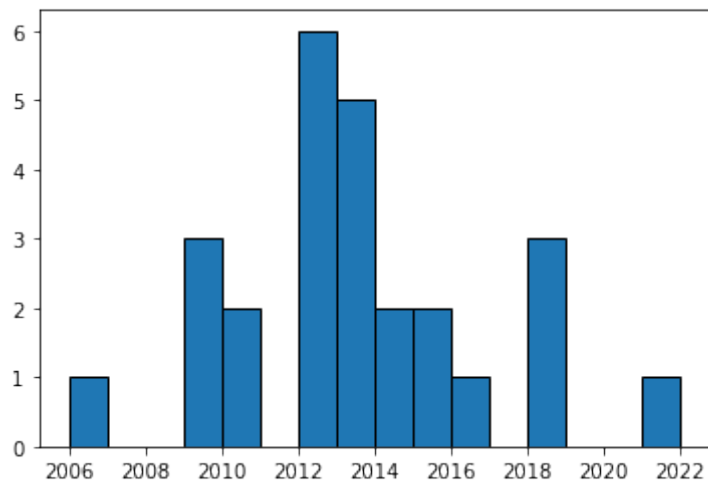
Por fim, os dados obtidos foram organizados, interpretados e discutidos, com o objetivo de compreender de que forma os fatores buscados influenciaram na descontinuidade do projeto Firefox OS.

6 RESULTADOS

6.1 REVISÃO SISTEMÁTICA

Foram analisados um total de 26 artigos, publicados entre 2006 a 2021. Uma das primeiras informações que se destacaram foi a ocorrência de um crescimento no número de publicações no tema, com um auge no biênio de 2012 a 2013, mas seguido de uma redução nos anos seguintes, retornando aos níveis de publicação anteriores, conforme apresentado na Figura 2.

Figura 2 - Quantidade de artigos publicados por ano sobre fatores motivadores do sucesso em projetos de *software* livre



Fonte: Elaborado pelo autor, 2022

Este fenômeno foi investigado e uma possível causa foi identificada ao se analisar as fontes de dados utilizadas nas pesquisas. Dentre os 26 trabalhos, 25 utilizaram amostras coletadas do repositório de *software* SourceForge. O repositório de *software* livre foi um dos mais importantes no planeta, com auges de atividade entre 2006 a 2008 e entre 2010 a 2011 (SQUIRE, 2017).

Houve uma parceria entre o *site* e um grupo de pesquisa do departamento de Ciência da Computação e Engenharia da Universidade de Notre Dame, na qual os

pesquisadores recebiam *snapshots* mensais do banco de dados *backend* do SourceForge (incluindo dados que não eram fornecidos ao público por meio do *site*). Os dados eram armazenados, formando uma base temporal, e disponibilizados a pesquisadores que entrassem em contato. Este projeto foi chamado “SourceForge Research Data Archive” (SRDA), e utilizado por mais de 100 pesquisadores em todo o mundo (ANTWERP; MADEY, 2008).

A partir dessa base de dados relacional, com mais de 100 tabelas, foi possível que pesquisadores obtivessem dados de participação de desenvolvedores e usuários, como número de *downloads* diários, acessos a páginas, frequências de contribuição, colaboradores únicos, categorias de projeto, estado de desenvolvimento, entre diversas outras. O projeto coletou dados a partir de 2003, até setembro de 2014, quando houve uma reorganização no banco de dados do SourceForge. Os dados coletados permaneceram disponíveis para pesquisadores, mesmo após o encerramento da coleta (SOURCEFORGE RESEARCH DATA ARCHIVE, 2016).

Contudo, houve uma rápida mudança nas preferências da comunidade de *software* livre quanto às opções de hospedagem de projetos, por volta do ano de 2011. Foi um cenário no qual o portal SourceForge se envolveu em controvérsias, tais quais apontadas por Hall (2015), como interferência nas páginas de *downloads* de projetos, oferecendo *links* para *downloads* de *software* patrocinado, o que levou desenvolvedores de *software* livre populares a buscarem outras plataformas. Foi no mesmo período em que a plataforma Github, ainda nova no mercado, ultrapassou o SourceForge em número de *commits*, segundo publicação de Finley (2011), e foi apontada por O'Grady (2011) como um novo “centro de gravidade” do *software* livre, crescendo a uma taxa de 4500 projetos por dia (NEATH, 2011). Desde então, a plataforma Github cresceu amplamente em número de projetos hospedados, possuindo atualmente mais de 200 milhões de repositórios de *software* e se apresentando como a maior comunidade de desenvolvimento de código aberto do mundo (GITHUB, 2022b, 2022a).

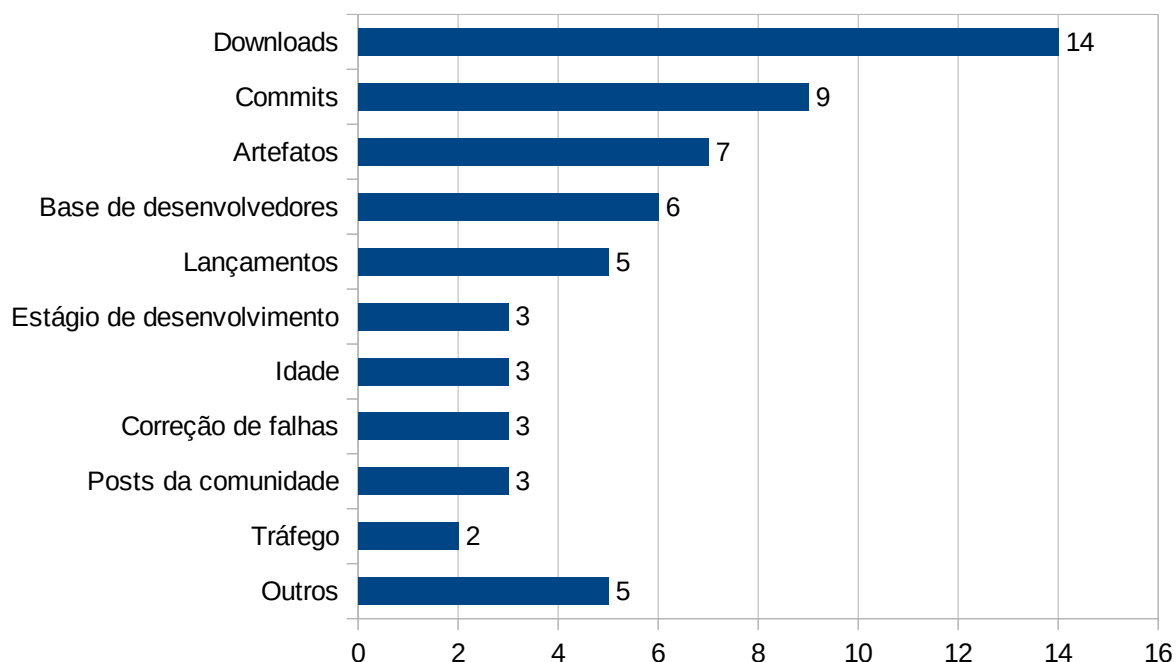
Essa mudança de plataforma de hospedagem preferida pela comunidade reduziu a relevância dos dados extraídos do SourceForge para pesquisas e, com o encerramento da aquisição de dados pela iniciativa SRDA, não houve um novo

projeto de pesquisa que possuísse parceria com o maior repositório de *software* livre da atualidade, o Github, que, por sua vez, não torna público e não compartilha dados internos do seu banco de dados com pesquisadores da mesma forma. Com isso, a principal fonte de dados utilizadas nas pesquisas relacionadas ao sucesso de projetos de *software* livre perdeu sua relevância, enquanto nenhuma alternativa foi criada.

Ainda após o encerramento da coleta de dados do SRDA, em 2014, algumas pesquisas foram realizadas com dados antigos do Sourceforge, o que se torna uma limitação das mesmas, já que não observaram as mudanças recentes no desenvolvimento de *software* livre (DANIEL *et al.*, 2018; DANIEL; STEWART, 2016; GHAPANCHI, 2015; GHAPANCHI; TAVANA, 2015; SUTANTO; JIANG; TAN, 2021; WANG; HUANG; LIU, 2018). Apenas 1 trabalho foi realizado com dados mais recentes, obtidos da plataforma Github, mas com limitações, obtendo apenas os dados disponíveis pela API oferecida pela plataforma para os projetos listados em sua página “featured” (SONG; KIM, 2018).

Em prosseguimento, foram observados os critérios de operacionalização do sucesso dentre os trabalhos, apresentados na Figura 3. Foram identificados como critérios mais comuns o número de *downloads* e *commits* de códigos. Estes são seguidos de outros critérios comuns, de estágio de desenvolvimento do projeto, frequência de lançamentos, correção de falhas, tráfego de rede, *posts* da comunidade, tarefas, base de desenvolvedores, base de usuários e idade do projeto. Apenas 5 critérios identificados foram encontrados em apenas 1 dos artigos, agrupados na categoria “outros”.

Figura 3 - Critérios de sucesso mais utilizados na literatura com a quantidade de artigos revisados que os utilizam



Fonte: Elaborado pelo autor, 2022

As justificativas para os critérios se deram em torno do sucesso técnico (*commits*, tarefas, correção de falhas, lançamentos e estágio de desenvolvimento), sucesso mercadológico (*downloads*, base de usuários e tráfego), sustentabilidade (base de desenvolvedores e idade) e engajamento da comunidade (base de usuários, posts e criação de tarefas).

A maioria das pesquisas utilizou diretamente as variáveis como indicadores de sucesso, enquanto os trabalhos de Wray *et al.* (2009) e Ghapanchi (2012a, 2012b) utilizaram de parâmetros calculados a partir das variáveis medidas, como uma forma de construir um indicador multivariável e simplificar o processo de análise. Entretanto, o indicador elaborado pelos autores utilizou de métricas dependentes de dados que eram fornecidos pela plataforma SourceForce, inviabilizando a utilização desses indicadores na atualidade.

Os fatores identificados como influentes para o sucesso dos projetos foram categorizados e podem ser observados na Figura 4, em que se observa um total de 20 fatores diferentes, com uma concentração de estudos em torno dos efeitos de rede e execução de tarefas.

Figura 4 - Fatores estudados na literatura por quantidade de artigos em que foram identificados



Fonte: Elaborado pelo autor, 2022

Dentre os fatores identificados, 7 não constam em mais de um estudo e outros 13 são encontrados em pelo menos 2 trabalhos. A maioria dos projetos aponta efeitos baseados em rede na comunidade de desenvolvedores e/ou usuários como um fator influente. Esses efeitos incluem quantidade de conexões internas ou externas, nível de “enbeddedness”⁹ e efeito “mundo pequeno”¹⁰. Outros fatores mais frequentes foram a eficiência na execução de tarefas, como correção de falhas e inserção de novas funcionalidades, o tempo de atividade do projeto (idade) e o nível de coordenação entre a equipe desenvolvedora e/ou entre colaboradores. Outros fatores menos frequentes foram o nicho de mercado do projeto, tipo de usuário alvo, diversidade de papéis entre os desenvolvedores, popularidade, linguagem de programação utilizada, existência de traduções, portabilidade e engajamento da comunidade.

⁹“Network embeddedness” é um conceito da teoria de redes sociais que se refere à forma como um indivíduo ou organização está integrado em uma rede social específica, incluindo a natureza dos laços sociais estabelecidos

¹⁰“Small world effect” é um fenômeno da teoria de redes sociais no qual, mesmo em redes grandes e complexas, a maioria dos nós pode ser alcançada por um pequeno número de etapas ou conexões intermediárias.

A relação de fatores identificados na literatura e seus respectivos artigos que os estudaram é apresentada na Tabela 1. É importante destacar que nem todos os artigos utilizaram a mesma nomenclatura para tratar do mesmo fator, então os termos encontrados nos artigos podem variar.

Tabela 1 - Relação de fatores encontrados e artigos que os abordaram

Fator estudado	Artigos
Efeitos de rede	Daniel e Stewart (2016) Grewal <i>et al.</i> (2006) Méndez-Duron e García (2009) Peng <i>et al.</i> (2013) Singh (2010) Sutanto <i>et al.</i> (2021) Sutanto <i>et al.</i> (2014) Wang (2012) Wang <i>et al.</i> (2012) Wen <i>et al.</i> (2013)
Execução de tarefas	Ghapanchi (2015) Ghapanchi (2013) Ghapanchi e Aurum (2012a) Ghapanchi e Tavana (2015) Méndez-Duron e García (2009)
Tipo de licença	Midha e Palvia (2012) Santos <i>et al.</i> (2013) Sen <i>et al.</i> (2012) Wang (2012)
Coordenação de equipe/colaboradores	Ghapanchi e Aurum (2012) Ghapanchi <i>et al.</i> (2014) Jiau e Kao (2009) Midha e Palvia (2012)
Tempo de projeto	Chengalur-Smith <i>et al.</i> (2010) Ghapanchi e Tavana (2015) Peng <i>et al.</i> (2013) Santos <i>et al.</i> (2013)
Tipo de usuário	Santos <i>et al.</i> (2013) Sen <i>et al.</i> (2012) Wang (2012) Wang <i>et al.</i> (2012)
Portabilidade	Ghapanchi e Tavana (2015) Sen <i>et al.</i> (2012) Wang <i>et al.</i> (2012)
Engajamento da comunidade	Jiau e Kao (2009) Song e Kim (2018)
Traduções	Ghapanchi e Tavana (2015) Midha e Palvia (2012)
Linguagem de programação utilizada	Ghapanchi e Tavana (2015) Sen <i>et al.</i> (2012)
Popularidade	Ghapanchi <i>et al.</i> (2014) Jiau e Kao (2009)

Diversidade de papéis	Midha e Palvia (2012) Daniel <i>et al.</i> (2018) Daniel <i>et al.</i> (2013)
Nicho de mercado	Chengalur-Smith <i>et al.</i> (2010) Santos <i>et al.</i> (2013)
Outros fatores	Chengalur-Smith <i>et al.</i> (2010) Daniel <i>et al.</i> (2018) Ghapanchi <i>et al.</i> (2014) Midha e Palvia (2012) Peng <i>et al.</i> (2013) Santos <i>et al.</i> (2013)

Fonte: Elaborado pelo autor, 2022

A literatura não apresentou distinção entre o que é considerado um fator que influencia o sucesso de um projeto de *software* livre e o que é um indicador do próprio sucesso, uma vez que alguns fatores identificados como exercendo influência positiva no sucesso de um projeto de *software* livre são, em outras pesquisas, identificados como indicadores do sucesso propriamente dito, como o tempo de projeto, popularidade, execução de tarefas, entre outros. Esta observação corresponde com os resultados obtidos por Peng *et al.* (2013), que identificaram inter-relações entre diferentes fatores e indicadores de sucesso, de tal forma que um indicador, como a idade do projeto, exerce influência em outros indicadores, como o número de *downloads* diários e o tamanho do projeto, podendo ser utilizado em alguns contextos como indicador e, em outros, como um fator motivador.

Tendo em vista essa dualidade entre causa e efeito de alguns fatores e critérios indicadores do sucesso, abordados em alguns trabalhos ora como fatores motivadores (causa), ora como indicadores de sucesso (efeito), este trabalho considerou como indicadores do sucesso aqueles nos quais os gestores do projeto não podem exercer influência direta e, como fatores motivadores do sucesso, aqueles em que é possível exercer influência direta. Portanto, os fatores de tempo de projeto e popularidade, devido a não permitirem exercício de ação direta sobre, foram considerados indicadores.

Em geral, identificou-se uma predominância de estudos sobre o nível de produção técnica dos projetos e sobre os efeitos de rede entre seus participantes. Outros fatores abordados em múltiplos estudos, vide Tabela 1, incluem o nicho de mercado ocupado pelo *software* desenvolvido, a quantidade de desenvolvedores no

projeto, a diversidade de papéis entre os participantes, a coordenação das equipes, a linguagem de programação utilizada, quantidade de traduções, portabilidade do *software*, tipo de licença utilizada e interação entre os participantes. Já outros fatores foram menos abordados na literatura, constando em apenas um estudo, como o grau de acoplamento do *software* (DANIEL; STEWART, 2016), modularidade e complexidade do código (MIDHA; PALVIA, 2012), quantidade de processos jurídicos (WEN; FORMAN; GRAHAM, 2013) e tamanho do projeto (WRAY; MATHIEU; TEETS, 2009).

É importante destacar que a recorrente abordagem de alguns fatores na literatura não implica em maior importância dos mesmos, assim como para uma menor abordagem, podendo sua recorrência estar relacionada à facilidade de obtenção dos dados a partir dos projetos estudados. Os dados relacionados a alguns fatores amplamente estudados, como o número de *commits* dos projetos e conexões entre os participantes, podiam ser extraídos mais facilmente da base de dados utilizada na maioria dos estudos, enquanto os dados relacionados a fatores como quantidade de traduções e coordenação de equipes, possuem maiores obstáculos para sua obtenção. Na seção seguinte, são detalhados e discutidos os principais fatores influenciadores do sucesso em projetos de *software* livre identificados na literatura.

6.1.1 Detalhamento dos fatores identificados

Projetos de *software* são complexos e envolvem diferentes questões em seu desenvolvimento, penetração e manutenção no mercado¹¹. Projetos de *software* livre, por sua vez, possuem uma complexidade ainda maior, já que possuem um tipo de licenciamento diferenciado e precisam coordenar não apenas sua equipe de trabalho, mas uma comunidade de colaboradores. Esta complexidade e multiplicidade de necessidades, se reflete na variedade de fatores encontrados na

¹¹Nesse contexto, entende-se mercado como os meios de distribuição de *software* para o público e as demandas do mesmo, não necessariamente significando um mercado comercial, uma vez que muitos projetos distribuem seus produtos gratuitamente.

literatura com algum grau de influência identificado em relação ao sucesso dos projetos, que envolvem aspectos relacionados ao seu planejamento, execução, distribuição e na forma como interagem seus participantes.

Desde a fase de planejamento do *software*, algumas decisões podem impactar seu sucesso futuro. Uma destas, e de importância fundamental, sendo preferencialmente definida antecipadamente, é o tipo de licença a ser utilizada no projeto. É sua licença que o tornará um *software* livre, oferecendo as 4 liberdades de *software* aos usuários e permitindo que estes também participem ativamente do projeto. A importância da escolha do tipo de licença o mais antecipadamente possível se dá pelo fato de que, à medida que colaboradores externos contribuem com código, se torna cada vez mais complexa a possibilidade da alteração da licença. Dessa forma, esta se torna, na maioria das vezes, uma característica permanente do projeto, necessitando ser escolhida após uma cuidadosa análise.

Dentre os artigos pesquisados, quatro identificaram relações entre o tipo de licença utilizada nos projetos de *software* livre e algum indicador de sucesso. Midha e Palvia (2012) formularam a hipótese de que projetos com licenças mais permissivas¹² possuem maiores taxas de sucesso mercadológico, enquanto há uma influência oposta no sucesso técnico. Esta hipótese foi elaborada a partir da ideia de que licenças permissivas atraem mais facilmente usuários e pessoas interessadas em criar projetos derivados, mas que geram menos incentivo aos seus próprios desenvolvedores em escrever código. Os autores consideraram a escala temporal, separando os projetos por estágios de desenvolvimento, e seus resultados confirmaram sua hipótese de que o uso licenças mais permissivas, de fato, contribui para o sucesso dos projetos, mas apenas durante o lançamento da primeira versão do *software*, enquanto o impacto do uso de licenças restritivas foi identificado apenas nos estágios mais avançados de desenvolvimento.

O impacto das licenças restritivas no sucesso técnico é reforçado pelos resultados da pesquisa de Santos *et al.* (2013), que identificaram relação positiva com a atividade e efetividade na resolução de tarefas. A pesquisa também avaliou a possibilidade de impactos causados pelo uso de licenças permissivas, não

¹²São as licenças de *software* livre que determinam o mínimo de limitações para o seu relicenciamento, permitindo até mesmo o uso só *software* licenciado para desenvolver *software* derivado de forma não livre, como as licenças do tipo BSD, a MIT e a APACHE.

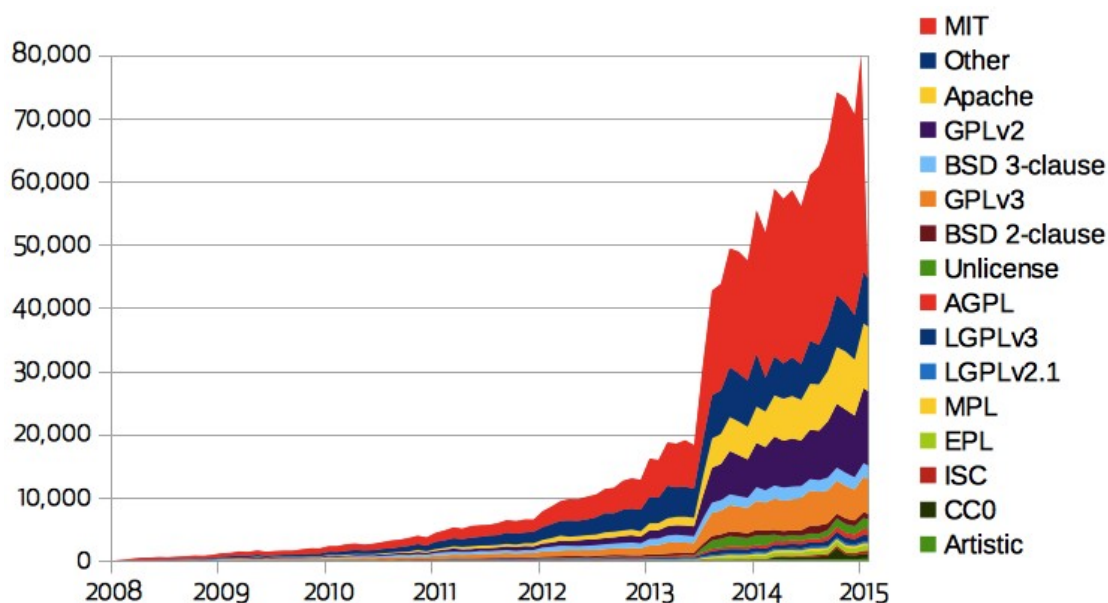
encontrando esta relação. Porém, sua metodologia não utilizou indicadores de sucesso baseados em popularidade ou separou os projetos por estágio temporal de desenvolvimento. Outra informação obtida no trabalho foi a relação do uso de licenciamento duplo na popularidade do projeto, medida por tráfego e número de *downloads*.

Já Sen *et al.* (2012), investigaram a taxa de seguidores e desenvolvedores nos projetos por tipos de licença, separadas em 3 categorias, sendo elas a permissivas, fracamente restritivas e fortemente restritivas. Sua pesquisa se difere das demais, por diferenciar os níveis de restrição em licenças restritivas. Foram encontradas taxas de seguidores inferiores em 21,7%, e de desenvolvedores, inferior em 18,4%.

Wang (2012) encontrou uma relação marginalmente positiva entre a restritividade mais alta de uma licença e a sobrevivência dos projetos nos estágios iniciais de desenvolvimento, o que pode se apresentar contraditório aos resultados das demais pesquisas e carece de maiores investigações.

É importante considerar que os estudos foram realizados com dados do SourceForge, que, como já mencionado, entrou em declínio de uso aproximadamente a partir de 2011 e que houve, desde então, um aumento no uso de licenças permissivas, principalmente em projetos de menor porte, como pode ser observado nos dados de uso de licenças no Github, disponibilizado por Balter (2015) e apresentado na Figura 5. Na figura, se observa um forte crescimento no uso da licença MIT, um tipo de licença permissiva, se tornando a mais utilizada na plataforma no período, com um total de 44,69% dos projetos.

Figura 5 - Evolução dos tipos de licença utilizados no repositório Github entre 2008 a 2015



Fonte: (BALTER, 2015)

O crescimento no uso da licença MIT está ligado ao seu uso como licença padrão para projetos no Github. O aumento do seu uso em projetos, junto do crescimento nas contribuições de código no Github, pode representar uma mudança na cultura da comunidade de *software* livre em relação ao licenciamento, com a possibilidade da redução do receio de contribuir com código para projetos sob licenciamento permissivo. Há uma carência de estudos mais recentes na área para identificar este possível fenômeno.

Outro fator normalmente determinado durante as etapas de planejamento do *software*, mas com algum grau de flexibilidade para mudanças futuras, é a linguagem, ou as linguagens, de programação utilizadas no projeto. Esta escolha muitas vezes se baseia em características técnicas, como as plataformas em que se deseja que o *software* funcione ou requisitos de desempenho e segurança, mas também está relacionada com preferências pessoais da equipe fundadora do projeto.

A escolha de uma linguagem de programação de maior popularidade no mercado foi relacionada a uma maior atratividade de desenvolvedores para o projeto em seus estágios iniciais de desenvolvimento, assim como a uma maior penetração

no mercado, tanto nos estágios iniciais, quanto nos estágios mais avançados de desenvolvimento (GHAPANCHI; TAVANA, 2015). Contudo, as preferências de linguagens de programação dos contribuidores de projetos de *software* livre podem diferir das preferências gerais dos programadores. Essa diferenciação foi apresentada por Sen *et al.* (2012), que identificou uma maior atratividade de desenvolvedores em projetos de *software* livre quando os projetos utilizam linguagens de programação da família C, em relação aos que utilizam outros tipos de linguagens, o que os autores associaram a questões culturais e históricas. Entretanto, o estudo foi realizado com dados entre 1999 a 2005 e, desde então, ocorreram mudanças nas preferências de linguagem de programação na comunidade de *software* livre, tendendo a uma maior utilização de linguagens interpretadas e independentes de plataforma para execução, como Python e JavaScript, apesar da presença ainda expressiva da linguagem C++ (ZAPPONI, 2022).

Dentre os fatores identificados cujo controle se dá principalmente durante as etapas de planejamento, há ainda o nicho de mercado de atuação do *software*, ou seja, o público-alvo do projeto e qual papel de utilidade este deseja desenvolver. Alguns tipos de *software*, devido à sua própria natureza técnica, possuem aplicação em diversas áreas, com possibilidade de uso generalizado, como editores de texto ou navegadores *web*, enquanto outros possuem funções mais especializadas, com áreas de uso mais restritas, o que se constitui como um nicho de mercado. Mesmo quando os projetos possuem funções mais generalistas, é ainda possível a sua inserção em nichos por meio de diferenciação da sua forma de uso, interface gráfica, tipo de licença, entre outras formas.

Na literatura, não há concordância quanto ao papel do nicho de mercado ocupado em relação ao sucesso dos projetos de *software* livre. Chengalur-Smith (2010) sugeriu uma maior capacidade de atrair desenvolvedores em projetos de segmentos com uma maior número de organizações atuantes. Já Santos *et al.* (2013), encontraram relações entre o segmento de atuação do *software* e sua atividade de desenvolvimento, com projetos de áreas mais populares, como *internet*, sendo mais ativos, enquanto projetos de áreas menos populares, como ciência e filosofia, sendo menos ativos. Sen *et al.* (2012) identificaram uma relação entre a

atratividade de desenvolvedores e o foco do projeto em usuários da área de tecnologia de informação, mas Wang (2012) observou essa relação apenas como marginalmente positiva.

Ainda que de forma não muito clara, a literatura sugere que projetos de *software* livre tendem a ser mais sucedidos quando exploram áreas já bem estabelecidas no mercado e em segmentos com maiores quantidades de usuários, conseguindo uma maior atratividade de desenvolvedores quando os usuários do *software* também são profissionais de tecnologia.

As pesquisas de Sen *et al.* (2012) e Wang (2012) identificaram relação entre as taxas de sucesso dos projetos e o tipo de usuário-alvo, sendo os projetos direcionados a profissionais de tecnologia da informação, como desenvolvedores e administradores de sistemas, mais frequentemente bem-sucedidos. O trabalho de Santos *et al.* (2013), por sua vez, realizou uma análise mais detalhada dos tipos de usuários, separando a categoria de profissionais de tecnologia da informação em duas, uma para desenvolvedores e outra para administradores de sistemas. A relação positiva foi encontrada apenas para o grupo de desenvolvedores e para os critérios de atratividade e tempo para entregar tarefas. A mesma pesquisa encontrou uma relação positiva entre o direcionamento a usuários não técnicos e a atratividade do projeto, mas uma relação negativa entre o direcionamento a esse mesmo grupo e o nível de atividade e a efetividade na resolução das tarefas. Já para o grupo de “outros”, foi encontrada relação negativa com a atratividade.

A relação entre o sucesso dos projetos de *software* livre e seu direcionamento a usuários desenvolvedores pode estar ligada ao fato destes terem maiores capacidades técnicas para contribuir de volta com código para os projetos, na forma de colaboradores, além de maiores habilidades na identificação de problemas e envio de sugestões mais realistas. O engajamento deste público-alvo com projetos de código aberto pode ser observado em uma pesquisa realizada pela Tidelift (2019), com profissionais da área. Quando questionados sobre a frequência com que contribuem com projetos de *software* livre, 62% dos entrevistados responderam que contribuem pelo menos 1 vez ao mês, enquanto apenas 3% responderam que nunca contribuíram.

O artigo de Midha e Palvia (2012) trouxe dois fatores não abordados pelos

demais estudos: a complexidade do projeto e sua modularidade, temas da engenharia de *software*, também fortemente relacionados às etapas de planejamento inicial, mas com consequências a longo prazo. Os autores mediram a modularidade utilizando a quantidade de agrupamentos de funções de membro, incluindo classes, *namespaces* e interfaces, e a complexidade, utilizando o método da complexidade ciclomática sobre a penúltima versão disponível do *software*. A complexidade foi observada pelos autores como influenciando negativamente o sucesso técnico, medido por número de *commits*, enquanto a modularidade apresentou relação positiva para a mesma métrica.

A partir da execução e manutenção do projeto, há uma maior quantidade de fatores influenciadores do seu sucesso estudados na literatura, com um destaque para a manutenção de alta atividade de desenvolvimento e para as conexões entre os participantes do projeto e com projetos externos.

Projetos de *software* são dinâmicos e inseridos em um cenário tecnológico de constantes mudanças, o que faz com que aqueles com pouca atividade de desenvolvimento percam seu atrativo, à medida que não apresentam novas funcionalidades e compatibilidades com novos padrões do mercado. Por isso, é importante que se mantenham com desenvolvimento ativo, além da constante necessidade de correções de falhas, principalmente quando relacionadas a questões de segurança. Tratando-se de projetos de *software* livre, há ainda a necessidade de manter relações com a comunidade de usuários e contribuidores externos, o que requer um nível mínimo de atividade constante.

Os resultados encontrados na literatura solidificam a base deste pensamento. Seis dos estudos pesquisados associam critérios relacionados à atividade de desenvolvimento, como número de *commits*, número de *bugs* resolvidos e execução de tarefas, a indicadores de sucesso, principalmente do sucesso técnico, como lançamento de versões. Os critérios são também relacionados a indicadores relacionados ao sucesso mercadológico, como o número de *downloads* e tráfego nas páginas dos projetos (GHAPANCHI, 2013, 2015; GHAPANCHI; AURUM, 2012a, 2012b; GHAPANCHI; TAVANA, 2015; GHAPANCHI; WOHLIN; AURUM, 2014). O fator relacionado à atividade de desenvolvimento mais frequentemente estudado foi a execução de tarefas no projeto, sendo também utilizado em outros artigos como

um indicador do próprio sucesso do *software* livre. Esta categoria abrange criação de código e gerenciamento de tarefas, sendo um fator de fácil medição, uma vez que as plataformas de hospedagem ou sistemas de controle de versão oferecem estes recursos, cujas variáveis podem ser obtidas quantitativamente. Jiau e Kao (2009) também investigaram os efeitos de atividades técnicas de desenvolvimento nos projetos, mas em relação aos estágios de desenvolvimento deste, confirmando sua hipótese de que projetos de *software* livre com maiores níveis de atividade atingem estágios de maturidade mais rapidamente. Os próprios estágios de desenvolvimento, são apontados por Santos *et al.* (2013) como um fator com influência sobre a quantidade e efetividade da atividade do projeto, sendo os estágios mais finalizados de desenvolvimento, positivamente influentes.

As conexões dos participantes dos projetos foram abordadas em oito dos estudos pesquisados, aplicando teorias de rede e buscando identificar a presença de diferentes efeitos da literatura de rede e suas possíveis relações com a rede de participantes dos projetos de *software* livre.

Os autores Daniel e Stewart (2016) apontaram que, para o desenvolvimento de *software*, é necessário um elevado nível de recursos de conhecimento, que ele denominou como “know-how” e “know-what”, que podem ser entendidos como o saber como fazer, e o saber o que fazer, além da atenção estruturada do desenvolvedor. Os autores argumentaram que projetos tradicionais de *software* são realizados em organizações que podem prover esses recursos, além de manter os desenvolvedores focados no projeto. Enquanto isso, projetos de *software* livre nem sempre são desenvolvidos em organizações, necessitando de formas alternativas de obter esses recursos e de engajar os desenvolvedores, que são, por muitas vezes, voluntários.

Com essa justificativa, os autores identificaram que a integração entre os participantes do projeto e a centralidade de rede influenciam positivamente no desenvolvimento de código, principalmente quando o acoplamento de código é mais baixo. Ou seja, se a rede de desenvolvedores e colaboradores não for muito fragmentada, mas centralizada, com mais interação e discussões entre os membros, o projeto possui maiores chances de ser bem-sucedido tecnicamente. Foi ainda observado que a externalidade de rede dos desenvolvedores, ou uma mais elevada

participação destes em outros projetos, pode anular as vantagens da centralidade de rede do projeto.

Esta relação entre recursos de conhecimento e redes concorda com o trabalho prévio de Méndez-Durón e García (2009), que observaram as conexões entre contribuidores que trazem seu conhecimento para o projeto, assim como as entre membros que levam conhecimento para outros projetos, é um fator motivador do sucesso, medido pelo nível de *downloads* por base de tempo.

Peng *et al.* (2013) observaram os efeitos da coparticipação de membros entre diferentes projetos de *software* livre e identificaram que esta se traduz em uma importante forma de transferência de conhecimento entre projetos, elevando as chances de sucessos de cada um.

Já Grewal *et al.* (2006) mediram o que denominaram como “network embeddedness”, que pode ser compreendido como o nível de integração da rede, baseada no número de projetos nos quais os gestores participam, no número de nós entre as pontas da rede e os gestores, e a participação dos gestores em projetos importantes. O conceito foi ainda expandido pelos autores para análise do grau de integração não apenas de pessoas, mas entre projetos de *software* livre. Foi identificado que o grau de integração do projeto possui influência no seu sucesso técnico e que o grau de integração de gestores é mais complexo e se diferencia entre projetos mais novos e mais antigos. Também foi identificada a influência da integração do projeto em seu sucesso comercial, mas em nível inferior, se comparado com o sucesso técnico. O nível de integração de rede entre participantes do projeto também foi identificado como um fator motivador para o sucesso por Sutanto *et al.* (2014), Wang *et al.* (2018) e por Sutanto *et al.* (2021), tendo estes últimos identificado o mesmo efeito positivo para a integração entre redes de projetos de *software* livre.

Wang *et al.* (2012) identificaram uma relação positiva com o sucesso no tamanho das redes internas e externas, tanto para usuários quanto para desenvolvedores, além de identificar que não apenas o tamanho, mas a qualidade das redes é um fator motivador para o sucesso. A relação entre o tamanho da rede e o sucesso do projeto também foi observada por Wang (2012).

Singh (2010) observou as características do efeito “mundo pequeno”, sendo

elas o *clustering*, comprimento médio de percurso, coeficiente “small-world”, e tamanho da rede, identificando relação positiva entre todos eles e o sucesso do projeto. O efeito de rede de mundo pequeno foi descrito pela primeira vez por Watts e Strogatz (1998) para definir uma classe de redes com características de alta clusterização e pequenos caminhos de rede, ainda não muito estudada na época e com características de alto desempenho em alguns campos de aplicação.

Outros tipos de fatores são relacionados à forma de interação dos participantes do projeto, se destacando a diversidade de papéis conduzidos por estes e a coordenação dos participantes do projeto.

Além do desenvolvimento de código-fonte, existem muitas tarefas envolvidas no desenvolvimento de um *software* livre, de forma a permitir a participação de diferentes pessoas, cumprindo com diferentes papéis, como programadores, *designers*, gestores, testadores, tradutores, documentadores, entre outros. Nesse sentido, Daniel *et al.* (2018) encontraram relação positiva entre a diversidade de papéis e a diversidade de contribuições dos participantes com o sucesso do projeto, de forma que projetos com maiores diversidades tendem a ser mais bem sucedidos. Não somente a diversidade de papéis dos integrantes das equipes possui influência sobre o sucesso dos projetos, como também a diversidade cultural das equipes e usuários contribui para o sucesso mercadológico do *software* livre (DANIEL; AGARWAL; STEWART, 2013).

As razões pelas quais a diversidade de papéis contribui para o sucesso de um projeto de *software* livre podem incluir uma maior divisão de tarefas e emprego de profissionais especializados nas suas áreas de atuação, além de incentivar uma maior participação da comunidade como um todo. Com isso, é possível ainda que haja uma maior execução de tarefas não relacionadas diretamente com o código-fonte, que poderiam ser negligenciadas por equipes constituídas majoritariamente por programadores. A contribuição da diversidade cultural, por sua vez, pode ser explicada, conforme os autores supracitados, em decorrência da facilitação do desenvolvimento da aplicação para diferentes contextos. Outra possível explicação pode ser devido a uma maior diversidade de culturas trazer uma maior variedade de ideias, contribuindo para a inovação do projeto.

A coordenação dos participantes do projeto, sejam os desenvolvedores,

sejam os usuários, se mostrou um fator relevante para o sucesso de iniciativas de *software* livre. Ghapanchi e Aurum (2012b), Midha e Palvia (2012) e Ghapanchi *et al.* (2014) observaram uma relação positiva entre a coordenação do trabalho cooperativo e a eficiência no desenvolvimento do projeto, enquanto Jiau e Kao (2009) identificaram uma relação positiva entre a coordenação das atividades, tanto de desenvolvedores, quanto de usuários, e uma mais rápida maturação do projeto.

A popularidade é utilizada como indicador do sucesso de projetos de *software* livre, mas poucos fatores estudados são ligados diretamente a ela. Dois fatores relacionados são a presença de traduções no projeto e a portabilidade do *software* para diferentes sistemas operacionais. Ghapanchi e Tavana (2015) identificaram relação positiva de ambos os fatores com o número de *downloads* do projeto e Midha e Palvia (2012) identificaram resultados concordantes para as traduções. Os artigos não encontraram relação entre os fatores mencionados e o sucesso técnico, utilizando seus critérios de medição. Já Sen *et al.* (2012) observaram relação positiva entre a portabilidade do *software* livre e a atração de tanto usuários quanto desenvolvedores, assim como Wray *et al.* (2009) encontraram relação positiva entre este mesmo fator e o indicador “project score” da base de dados do SourceForge. Este indicador era um valor calculado semanalmente a partir do número de postagens nos fóruns, tarefas, *bugs*, *patches*, itens rastreados, *commits*, lançamentos e *downloads*. O valor era disponibilizado publicamente pela plataforma e utilizado como comparativo entre os projetos hospedados.

Quanto ao engajamento da comunidade, é observado que, quanto mais intenso, mais rapidamente o projeto é desenvolvido (JIAU; KAO, 2009). Contudo, Song e Kim (2018) identificaram não apenas uma relação entre quantidade de contribuições de usuários e o sucesso do projeto, mas que a presença de “heavy users”, usuários que realizam quantidades de contribuições muito acima da média e com alta frequência, é um relevante fator em projetos bem-sucedidos. Esses usuários geralmente possuem um nível de conhecimento mais avançado e maior experiência em relação ao *software*, se tornando peças centrais na comunidade.

Estrategicamente, uma das principais vantagens de se desenvolver um *software* livre, em relação a um *software* proprietário, é a possibilidade de engajar a comunidade em seu desenvolvimento, recebendo diferentes tipos de contribuições,

e, até mesmo, reduzindo a demanda de profissionais. Nesse sentido, seria esperado que um maior engajamento da comunidade elevasse as chances de sucesso do projeto. Entretanto, compreendendo o papel dos “heavy users”, é possível se obter um melhor direcionamento em como melhor incentivar a comunidade a contribuir com o projeto.

Outros fatores foram menos abordados na literatura e, por isso, não incluídos no estudo, já que as informações não são suficientes para uma análise mais densa, ou para a comparação entre diferentes abordagens.

6.1.2 Síntese dos fatores motivadores do sucesso

Segundo as informações obtidas na literatura, foi possível identificar alguns fatores motivadores do sucesso dos projetos, observados de diferentes pontos de vista, sendo estes o sucesso técnico, baseado na capacidade da equipe e dos colaboradores de produzir um produto de boa qualidade e capaz de atender às demandas do mercado e dos usuários; o sucesso mercadológico, baseado na popularidade do projeto em seu meio de atuação; e a sustentabilidade, baseada na atividade do projeto e em sua capacidade de manter-se ativo no futuro.

Para cada perspectiva, diferentes indicadores são utilizados como métricas. Para o sucesso técnico, são normalmente baseadas nos metadados dos sistemas de controle de versão e plataformas de desenvolvimento, como número de *commits* e aberturas e fechamento de *bugs* e sugestões. Para o sucesso mercadológico, são normalmente baseadas nas estatísticas dos *sites*, como número de acessos, e *downloads*. Já para a sustentabilidade, as métricas são mais complexas, necessitando uma observação holística, considerando a base de usuários, desenvolvedores e a atratividade do projeto para a manutenção dos colaboradores e atração de novos interessados.

Dentre os fatores identificados, alguns são relacionados a decisões tomadas ainda nas etapas de planejamento do projeto, como a escolha da licença, a linguagem de programação e modularidade do *software*. Esses fatores são de difícil

manipulação em estágios mais avançados de desenvolvimento e muito custosos, caso necessário, mas possuem influência em todas as etapas do projeto. Isto chama a atenção para a necessidade de um bom planejamento inicial e visão a longo prazo dos gestores durante as fases iniciais.

Já os demais fatores, são mais flexíveis e podem ser mais facilmente trabalhados em estágios avançados de desenvolvimento, permitindo um maior espaço para novas estratégias, principalmente em projetos mais antigos. Dentre esses fatores, destacam-se as traduções, frequência de lançamentos, correção de falhas e atendimento de demandas.

Com base nas informações encontradas na literatura, pode-se sumarizar o caminho para o desenvolvimento de um projeto de *software* livre com maiores chances de sucesso como:

- Desenvolver o *software* com o objetivo de resolver um problema já conhecido;
- Buscar, quando possível, direcionar o *software* para categorias com bastantes usuários;
- Caso se deseje captar uma maior quantidade de desenvolvedores, considerar direcionar o projeto a um público-alvo de usuários da área de tecnologia da informação, já que tendem a contribuir de volta como desenvolvedores em maior frequência;
- Escolher cautelosamente o tipo de licença a ser utilizada, considerando suas consequências jurídicas e possíveis efeitos na atratividade de desenvolvedores a longo prazo. O uso de licenças mais permissivas, como a MIT e Apache, tende a tornar o projeto mais atrativo ao público durante a maturação do *software*, enquanto o uso de licenças mais restritivas, como a GPL, tende a incentivar os desenvolvedores a escrever mais código;
- Buscar, quando possível, utilizar linguagens de programação mais populares na comunidade de desenvolvedores de *software* livre, sendo recomendado verificar por publicações como a de Zapponi (2022), que apresentam as tendências para cada linguagem;

- Utilizar de boas práticas de engenharia de *software*, tornando o projeto mais facilmente modificável por pessoas externas, e com maior modularidade, permitindo, simultaneamente, o reúso de código em projetos externos e a inserção de código externo;
- Construir uma equipe competente, bem integrada, com variedade de papéis entre os membros e com boa comunicação entre si, de forma que haja uma centralidade deste grupo no desenvolvimento do *software*;
- Ter boa e frequente comunicação com participantes externos, facilitando a captação de contribuições;
- Traduzir o *software* para diferentes idiomas, ampliando a base de usuários para um público internacional;
- Portar, sempre que possível, o *software* para diferentes plataformas, conquistando assim novos usuários;
- Identificar e estimular membros de alta atividade (“heavy users”) na comunidade a se manterem no projeto;
- Manter o projeto ativo, com frequentes inserções de código, correção de falhas e lançamentos de novas versões;

O conjunto de fatores sumarizado neste trabalho pode servir como um guia de boas práticas na elaboração e condução de um projeto de *software* livre, permitindo que estes tenham maiores chances de sucesso.

6.2 ESTUDO DE CASO

Os fatores motivadores do sucesso de projetos de *software* livre encontrados na literatura foram construídos, em suas respectivas pesquisas, a partir da significância estatística da correlação com os diferentes indicadores do sucesso dos projetos. Este tipo de abordagem permite identificar a capacidade de um fator influenciar no resultado dos indicadores, mas não permite compreender sua dinâmica nos projetos, se sua relação é de causa ou indireta, como atuam ou como as diferentes partes envolvidas o experienciam.

Neste sentido, o estudo de caso permite aplicar uma visão mais aprofundada a cada fator levantado, observando seu comportamento e contribuições em um projeto. Foi realizado um estudo de caso do projeto Firefox OS, buscando identificar a dinâmica dos fatores estudados, estratégias de medição, e a possibilidade de existência de fatores relevantes ainda pouco abordados pela literatura científica. O estudo de caso usou, como comparativo, o projeto Android, devido a este também ser licenciado abertamente, atuar no mesmo segmento, e por ter se demonstrado um caso de sucesso.

6.2.1 Apresentação

O Firefox OS foi um projeto responsável pela criação de um sistema operacional para *smartphones*, desenvolvido pela Mozilla, uma fundação sem fins lucrativos criada em 1998 e conhecida pelo desenvolvimento do navegador *web* Firefox, que atua, junto da sua subsidiária, a corporação Mozilla, para desenvolver tecnologias para a *internet* e lutar pelos direitos dos usuários (MOZILLA, [s.d.]).

O projeto iniciou em 2011, com a proposta de um sistema operacional para dispositivos móveis em que toda a interface do usuário fosse construída na forma de

aplicativo *web*, que, por sua vez, conseguisse exibir e executar outros aplicativos *web*, trazendo assim, todas as funcionalidades de um sistema operacional móvel e flexibilidade para criação de novos aplicativos (MOZILLA DEVELOPER NETWORK, 2015).

A missão da Mozilla com o Firefox OS não era apenas a de criar um sistema operacional para *smartphones*, mas a de criar um sistema aberto, livre de tecnologias proprietárias e controle de fabricantes, baseado em tecnologia *web*. Isto permitiria que seus aplicativos pudessem ser desenvolvidos utilizando ferramentas já disponíveis no mercado e permitissem a sua execução em qualquer sistema operacional e plataforma, podendo o mesmo aplicativo ser utilizado em computadores e *smartphones*, assim como distribuído livremente, sem a necessidade de terceiros (JANCZUKOWICZ, 2013; MOZILLA DEVELOPER NETWORK, 2015). Sua missão incluía desenvolver um celular acessível, que permitisse auxiliar na inclusão digital das pessoas que não tinham acesso a *smartphones*. Para isso, buscava desenvolver um sistema operacional e ambiente de execução otimizados para *hardware* com poucos recursos e baixo custo. (MOZILLA, 2012)

O projeto ainda oferecia outros recursos inovadores, como busca contextual profunda, e aplicativos sob demanda (sem a necessidade de serem instalados no dispositivo). Entretanto, o projeto foi descontinuado pela Mozilla em setembro de 2016, sob a justificativa de que, desde o final do ano de 2015, a liderança da Mozilla havia chegado à conclusão de que o Firefox OS não traria o retorno esperado (MCCRACKEN, 2016).

6.2.2 Histórico e recepção na mídia digital

O projeto teve seu primeiro anúncio em julho de 2011, sob o nome de “Boot to Gecko”, com o objetivo de construir um sistema operacional completo e independente para a *web* aberta, como uma forma de lutar contra o monopólio dos fabricantes sobre as plataformas de aplicativos, permitindo que todos pudessem ter

acesso a esta (NYMAN, 2011). A primeira prova de conceito foi apresentada em fevereiro de 2012, no congresso MWC, em Barcelona. No evento, foi apresentada uma demonstração funcional, executando experimentalmente sobre *hardware* existente no mercado, além da apresentação da parceria firmada com a empresa de telecomunicações Telefónica e o interesse manifestado por outras empresas como a Adobe, Qualcomm e Deutsche Telekom em oferecer suporte para o projeto (FRANCIS, 2017).

A partir deste anúncio, a Mozilla atraiu a atenção do público para o projeto, ganhando espaço na mídia ao longo do ano de 2012, enquanto atraíam desenvolvedores e se preparavam para o lançamento da versão 1.0, a primeira a ser comercializada. A mídia apresentou o projeto com destaque para sua tecnologia baseada em padrões *web* e sua abertura, o que permitiria que os dispositivos se tornassem mais baratos, acessíveis, e que qualquer pessoa pudesse desenvolver aplicativos para essa plataforma, sem que se prendessem a *hardware* e *software* específicos, o que foi considerado um potencial para “mudar o mundo” (GOODWIN, 2012; MIES, 2012). Entretanto, a experiência de uso do protótipo apresentado foi relatada como não muito se diferenciando da experiência de uso das plataformas já existentes no mercado da época (MIES, 2012), mas, ainda assim, a plataforma foi apresentada como promissora e sua recepção pelo público, descrita como positiva, principalmente em relação à sua fluidez de desempenho em dispositivos de baixo custo, o que não era alcançado pelas plataformas disponíveis (BARROS, 2012; MCCANN, 2012; NAST, 2012; SILVA, 2012).

Nesse mesmo período, a Mozilla (2012) firmou parcerias com a ALCATEL e a ZTE, fabricantes mundiais de celulares, e com empresas operadoras de telefonia de diferentes países, como a Deutsche Telekom, Etisalat, Smart, Sprint, Telecom Italia, Telefónica e Telenor. O objetivo era ganhar suporte na indústria para o lançamento do FirefoxOS para o início de 2013, com o primeiro lançamento previsto para o Brasil. Este já não foi apresentado apenas como um *smartphone*, mas como um ecossistema móvel baseado em tecnologias *web*.

O anúncio da primeira versão comercial, foi realizado durante a MWC de 2013, apresentando o produto prestes a ser lançado aos consumidores no Brasil, Colômbia, Hungria, México, Montenegro, Polônia, Sérvia, Espanha e na Venezuela,

além de destacar a parceria com mais um fabricante, a LG, e a possibilidade de parceria com a Huawei (MOZILLA, 2013). Os dispositivos apresentados no evento foram o “Alcatel One Touch Fire” e o “ZTE Open”, ambos desenvolvidos especificamente para o Firefox OS e com especificações de *hardware* abaixo da média do mercado, já que o sistema era otimizado para um menor consumo de recursos e com a finalidade de reduzir os custos dos dispositivos para alcançar os mercados dos países em desenvolvimento (TRENHOLM, 2013; ZTE CORPORATION, 2013).

Dessa vez, o Firefox OS foi divulgado nos veículos digitais de mídia de forma mais objetiva e menos especulativa, com um foco nas melhorias apresentadas no evento, nos dispositivos a serem lançados no mercado e nas novas parcerias firmadas com fabricantes, distribuidores e desenvolvedores de aplicativos (HEATER, 2013; ROWINSKI, 2013; WIRED STAFF, 2013). É importante destacar que, na edição de 2013 do evento, outros projetos de sistemas operacionais para *smartphones* foram apresentados, como o Tizen, desenvolvido pela Samsung e o Ubuntu Touch, um projeto também de *software* livre, desenvolvido pela Canonical. A diversidade de opções dividiu o foco da atenção do público, reduzindo o destaque do Firefox OS como inovação no mercado de *smartphones* (GRUSH, 2013; JENKINS, 2013; SAWH, 2013). O criticismo em torno do Firefox OS após o evento centrou na experiência do usuário, considerada não tão fluida quanto antecipada, com tempos de carregamento de aplicativos podendo chegar a 5 segundos, e na existência de *bugs* (DONNELLY, 2013; WILLIAMS, 2013). Entretanto, apesar da experiência criticada, o sistema era considerado competitivo e com um objetivo que seria benéfico para toda a indústria *web* e de *smartphones* (ZIEGLER, 2013).

Uma importante observação pode ser feita sobre as apresentações públicas do Firefox OS: o projeto tinha como principal público-alvo pessoas de países emergentes com dificuldades de inclusão digital, mas seu principal veículo de apresentação ao público se dava no congresso MWC, que, apesar de ser considerado o principal evento da área no mundo, é realizado na Espanha e voltado ao público da Europa e América do Norte. Dessa forma, os primeiros utilizadores e divulgadores de opinião não pertenciam ao público-alvo do produto e, com isso, exerciam um viés em suas análises, comparando os produtos anunciados com

produtos “high end”, e não com *smartphones* “low end”, pertencentes ao segmento de mercado objetivado. Uma análise do Firefox OS considerando o seu público-alvo foi realizada por Shankland (2013), que o descreveu como atrás dos grandes atores do mercado, mas adequado para o seu nicho, sendo uma boa opção para usuários de “feature phones”.

A participação da Mozilla em eventos de tecnologia nos países onde o Firefox OS seria lançado foi mais restrita. Um dos eventos no qual o Firefox OS foi apresentado foi o Campus Party, no qual foi demonstrado o sistema e como era possível a criação de aplicativos para o mesmo, além da execução de um “hackathon” (AZEVEDO, 2014). O evento possui uma natureza mais fechada, portanto, não se encontram informações jornalísticas sobre como decorreu essa participação da Mozilla, assim como sua repercussão.

A Mozilla (2014a, tradução nossa) iniciou o ano seguinte reafirmando sua missão com o Firefox OS, declarando em seu blog oficial que:

(...) acredita que deve existir uma alternativa verdadeiramente aberta às duas plataformas dominantes de sistemas operacionais para *smartphones*. Uma alternativa que entregue uma experiência incrível ao usuário e provenha os desenvolvedores com a oportunidade de construir os *apps* dos seus sonhos, enquanto colocando o poder nas mãos de cada indivíduo.

A fundação ainda mencionou, no mesmo anúncio, que, diferentemente das outras plataformas, o conteúdo adquirido pelos usuários era pertencente a estes, sem estarem acoplados a serviços específicos, e que os desenvolvedores eram encorajados a distribuir aplicativos da forma como quisessem, até mesmo com a possibilidade de criação de lojas de aplicativos concorrentes.

Além desta reafirmação, o *post* apresentou o ano encerrado de 2013 como marcado pelo lançamento dos primeiros dispositivos com o sistema, já em 14 mercados no mundo, apoiados sobre parcerias com empresas de telecomunicação e fabricantes de *hardware*, além da criação de um grupo objetivado a padronizar o ecossistema de *web* aberta no mundo.

Em fevereiro de 2014, ocorreu mais uma edição do evento MWC e a Mozilla

(2014b) realizou nova exibição. Desta vez, foram apresentadas as conquistas do ano de 2013, considerado um ano de sucesso, e demonstradas suas novidades, com 7 novos dispositivos de configurações de *hardware* mais elevadas do que os do ano anterior (MOZILLA, 2014c).

A principal repercussão na mídia sobre o evento foi em relação ao lançamento do dispositivo “Alcatel Spreadtrum”, cujo custo estava projetado para 25 dólares, sendo divulgado por alguns veículos como possivelmente o *smartphone* mais barato do mundo (MACHADO, 2014; RIVERA, 2014). A novidade era considerada capaz de democratizar o acesso à informação, e o dispositivo, considerado bom dentro de suas limitações (RIVERA, 2014), mas a experiência geral de uso era ainda considerada pouco fluida, com travamentos e lentidão (SAVOV, 2014). Outros dispositivos e iniciativas foram também apresentados, chamando a atenção para o progresso alcançado desde o ano anterior, mas o panorama era considerado de um grande desafio frente à concorrência (MCCASKILL, 2014; SHANKLAND, 2014).

O ano de 2014 foi marcado pela entrada do Firefox OS em novos mercados e pela criação do *smartphone* de 25 dólares, o que a fundação considerava como a exploração de um novo mercado. A Mozilla descreveu o ano como um período de expansão e ganho de momento, demonstrando um sentimento positivo em relação ao projeto (MOZILLA, 2014e).

Já no MWC 2015, a Mozilla trouxe sua nova empreitada: levar o Firefox OS para televisores (MOZILLA, 2015a). A repercussão foi reduzida, limitada, em sua maioria a veículos de divulgação de menor porte e de pouco alcance global. O foco da divulgação foi a expansão do ramo de *smartphones* para o de dispositivos conectados, principalmente televisores, em parceria com a Panasonic (CUNNINGHAM, 2015; LOVERIDGE, 2015). Contudo, a usabilidade dos *smartphones* seguiu como objeto de intensas críticas (HIGA, 2015).

O ano de 2015 foi marcado por instabilidade e mudanças de estratégias no projeto. O diretor-executivo da Mozilla relatou o grande esforço desempenhado no Firefox OS e declarou que não observaram tração suficiente no dispositivo de 25 dólares. O diretor apresentou ainda a nova iniciativa da Mozilla, chamada de “ignite”, ou ignição. A iniciativa se baseava em criar uma experiência unificada de produto,

focada em sua qualidade e profundidade, sem buscar expansões para novos dispositivos antes de ganhar tração suficiente para tal, e criando um produto que as pessoas buscariam pela experiência, e não pelo preço (SHANKLAND, 2015). Apesar da declaração realizada em maio do mesmo ano, a Mozilla realizou, no mesmo ano, uma iniciativa de expansão para o mercado de dispositivos conectados, utilizando tecnologia do Firefox OS (MOZILLA, 2015b).

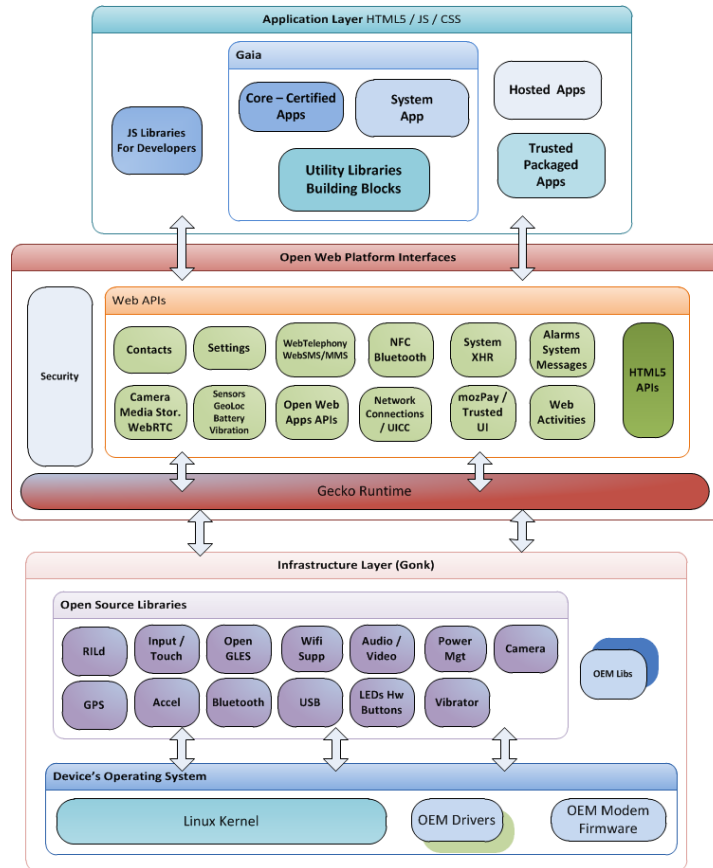
Em fevereiro de 2016, a Mozilla já anunciou a descontinuidade do desenvolvimento do Firefox OS para *smartphones*, mas afirmou que seguiria com a tecnologia desenvolvida para o ramo de dispositivos conectados (ROTTER, 2016). Apenas em setembro do mesmo ano que foi anunciada a descontinuidade definitiva do projeto (MCCRACKEN, 2016).

6.2.3 Tecnologia e desenvolvimento do projeto

A principal tecnologia integrante era o Boot to Gecko (B2G), um projeto anunciado pela Mozilla em 2011, cujo objetivo era criar um sistema operacional completo e independente para a *web*, de forma a eliminar as pilhas de tecnologias proprietárias para desenvolvimento de aplicações, que seriam desenvolvidas utilizando as tecnologias *web* abertas e já existentes (FRANCIS, 2017). O B2G consistia do ambiente de execução Gecko, já previamente desenvolvido pela Mozilla para o seu navegador Firefox, junto de uma série de APIs *web*, que serviam de blocos para a construção de aplicativos.

O B2G não era um sistema operacional por si só. Sua arquitetura, apresentada na Figura 6, é composta de mais 2 projetos: uma camada de infraestrutura, utilizando um *kernel* Linux, uma camada de abstração de *hardware* e um conjunto de bibliotecas básicas, chamado Gonk, além de uma camada de aplicativos com uma interface gráfica construída sobre os padrões *web* e que permitia a execução de aplicativos sobre ela, assim como o seu gerenciamento (MOZILLA, 2014d).

Figura 6 - Representação esquemática das camadas de tecnologias do sistema operacional Firefox OS



Fonte: (MOZILLA, 2014d) Elaborado pelo autor, 2022

O desenvolvimento do código do projeto foi realizado de forma pública, com o código-fonte disponível e possibilidade de participação da comunidade, por meio de sistema de controle de versão. O projeto-piloto do B2G constava com 36 colaboradores no desenvolvimento de código, enquanto no seu encerramento, segundo Shankland ([s.d.]), em uma matéria publicada na CNET, a Mozilla empregava cerca de 50 funcionários unicamente no projeto, de um total de cerca de 1000 na organização.

O projeto usava um modelo de lançamento denominado “train model”, no qual um período fixo de lançamento era definido e as versões lançadas nas datas previstas, removendo as funcionalidades ou correções que não conseguiram acompanhar o calendário. Trata-se de um modelo ágil para desenvolvimento. Eram adotadas ainda práticas de rastreamento de problemas, controle de versões e integração contínua (LIN, 2013).

Um importante aspecto do projeto é que, se tratando de um sistema operacional para *smartphones*, não basta que se desenvolva o sistema em si, mas, para que este apresente utilidade para o público, é necessário fornecer ferramentas para desenvolvimento de aplicativos e atrair desenvolvedores para criarem projetos compatíveis com o sistema.

Este era um diferencial do Firefox OS, pois não necessitava do uso de ferramentas de desenvolvimento próprias ou do aprendizado de novas tecnologias. Este permitia que fossem empregadas as ferramentas já existentes para desenvolvimento *web*, apenas recorrendo às suas APIs de acesso a recursos de *hardware* e de permissões do sistema.

Grønli *et al.* (2014) realizou um trabalho em que desenvolveu um mesmo aplicativo para as plataformas dominantes do mercado da época e para o Firefox OS, que os autores consideraram como representante de uma nova categoria de sistemas operacionais móveis. Na comparação realizada, destacaram a agilidade que a plataforma oferecia aos desenvolvedores para a publicação de seus aplicativos e o tamanho de arquivo reduzido destes em comparação com as demais plataformas. Contudo, os autores relataram encontrar limitações nas ferramentas de desenvolvimento do Firefox OS, considerando-as imaturas e ainda não boas o suficiente para serem adequadamente comparadas com as dos demais sistemas operacionais.

Já Romero-Olmo (2015), realizou o desenvolvimento sistematizado de um aplicativo para o Firefox OS, com o uso de técnicas de engenharia de *software*. O autor relatou o ambiente de desenvolvimento como simples, completo e de fácil utilização, permitindo a criação de aplicações em tempo ágil, mesmo para desenvolvedores ainda não familiarizados com a plataforma. Foram ainda discutidas a lenta adoção no mercado e a missão do projeto de desenvolver uma plataforma aberta e acessível. É importante destacar que o trabalho foi desenvolvido no ano seguinte ao trabalho de Grønli *et al.* (2014), e o projeto Firefox OS se encontrava em um estágio mais avançado de desenvolvimento, o que pode explicar a experiência mais positiva com as ferramentas de desenvolvimento oferecidas.

Outro importante aspecto de um sistema operacional é a sua segurança oferecida aos usuários, no sentido de proteger o sistema, seus dados e recursos

contra ameaças e ataques externos, assim como garantir a privacidade e a integridade das informações. Verez e Hugues (2013) apresentaram o que consideram como as principais implementações de segurança do sistema, sendo elas um controle de permissões, restrição de acesso às APIs, o processo B2G, o processo de conteúdo e sua metodologia de atualizações do sistema.

O Firefox OS implementava um sistema de restrição de uso de APIs no qual os aplicativos eram categorizados, em níveis de acesso, como aplicativos privilegiados, certificados, e *webapps* (GADYATSKAYA; MASSACCI; ZHAUNIAROVICH, 2014). Os aplicativos privilegiados eram aqueles pré-instalados pelo fabricante, com acesso a funções do sistema, como recursos de telefonia. Já os certificados, eram aqueles oferecidos pelo *marketplace* da Mozilla e com código revisado, recebendo acesso elevado às APIs do sistema. Os *webapps*, por sua vez, eram aplicativos sem acesso especial a recursos do sistema e passíveis de serem distribuídos e obtidos a partir de qualquer fonte.

Defreez *et al.* (2014) apontaram que o fato do Firefox OS aproximar a experiência do dispositivo móvel e da *web* o colocava numa posição de transição, se apoiando em bibliotecas baseadas no modelo de segurança de *desktops*. Porém, o modelo de um navegador de *internet* em operação constante entrava em conflito com a noção de ciclo de vida de aplicativos, o que trazia uma nova classe de *bugs* ainda pouco explorados e com potenciais riscos para a segurança.

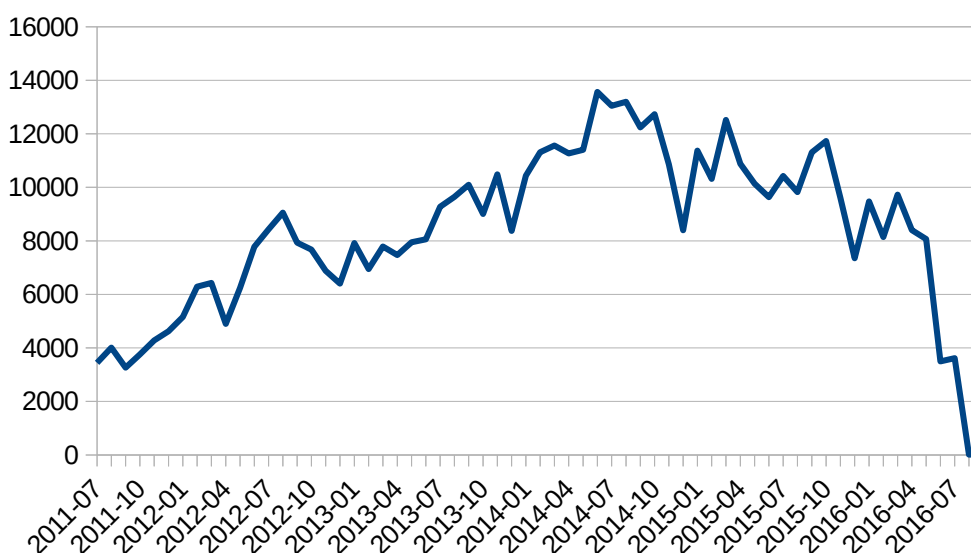
6.2.4 Indicadores técnicos

O nível de atividade de desenvolvimento no projeto foi medido por meio do número de *commits* na base de código associada, obtida a partir de repositórios arquivados da Mozilla ([s.d.]) disponíveis no repositório Github. Os dados representam a atividade tanto de membros da Organização, quanto colaboradores externos, e até mesmo autores, cujos códigos foram importados a partir de outras bases de código livre, como uma forma de contribuição indireta.

Para compreender o engajamento dos desenvolvedores, os números de

commits obtidos a partir do código-fonte do B2G foram agregados por mês, a partir de julho de 2011, quando houve o primeiro anúncio do projeto, até setembro de 2016, quando foi anunciado o seu encerramento. Os dados estão dispostos na Figura 7, em que se observa um crescimento no número de *commits* mensais a partir do primeiro anúncio, até o final de 2014, quando as atividades se reduziram, em ritmo moderado, até meados de 2016, quando houve anúncio do encerramento e os números cessaram.

Figura 7 - Quantidade de *commits* mensais na base de código do projeto Firefox OS

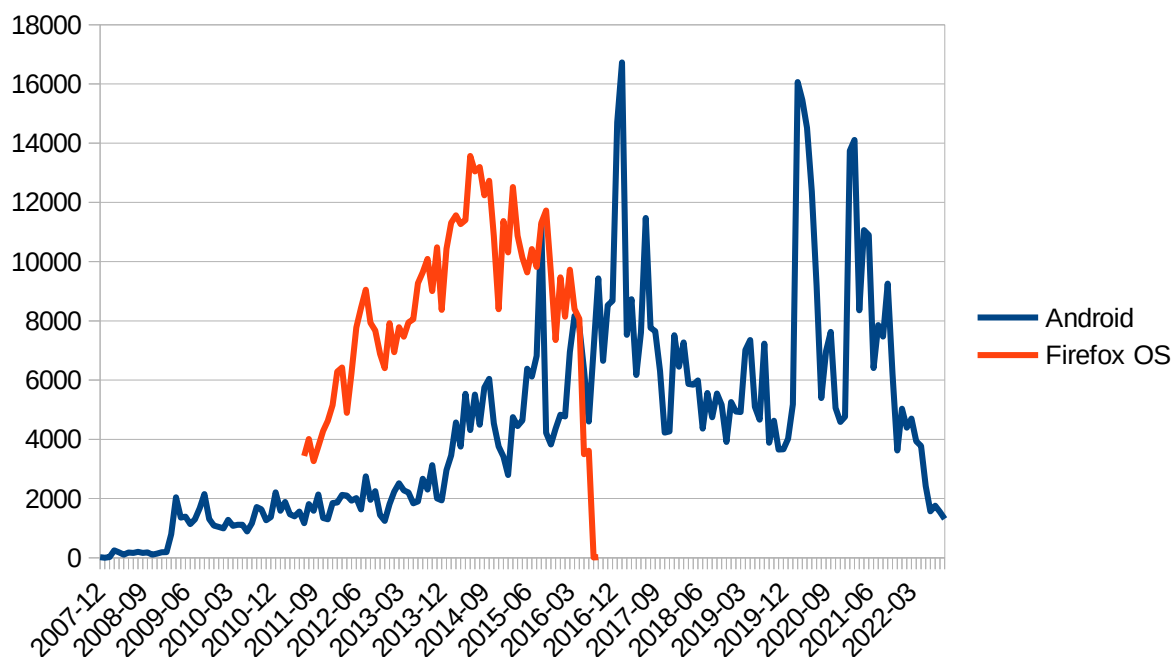


Fonte: Elaborado pelo autor, 2022

As atividades na base de código demonstram o engajamento dos desenvolvedores do Firefox OS, principalmente nos períodos da primeira apresentação pública no WMC 2012, no mês de fevereiro do mesmo ano do evento, quando o número de *commits* ultrapassou pela primeira vez a marca de 6000 mensais, seguido de um pico de atividade de 6428 no mês de março, e, posteriormente, em julho do mesmo ano, ultrapassando a marca de 9000 *commits* mensais. O mês de fevereiro do ano de 2013, marcado pela apresentação do primeiro dispositivo com o Firefox OS, coincidiu com o início de um novo crescimento na atividade de desenvolvimento de código, que se sustentou até o final do ano de 2014.

Como comparativo, a Figura 8 apresenta os números de *commits* na base de código do Android, sobrepostos do número de *commits* na base de código do Firefox OS. Nela se observa como, desde sua apresentação para o público, em 2011, o Firefox OS contava com maior quantidade de *commits* em seu repositório, em torno do dobro apresentado para a base de código do Android no mesmo período. O crescimento da quantidade de *commits* se deu de forma mais acelerada para o Firefox OS até o final do ano de 2014, quando a quantidade de *commits* mensais começou a reduzir, enquanto no projeto Android, seguia em crescimento.

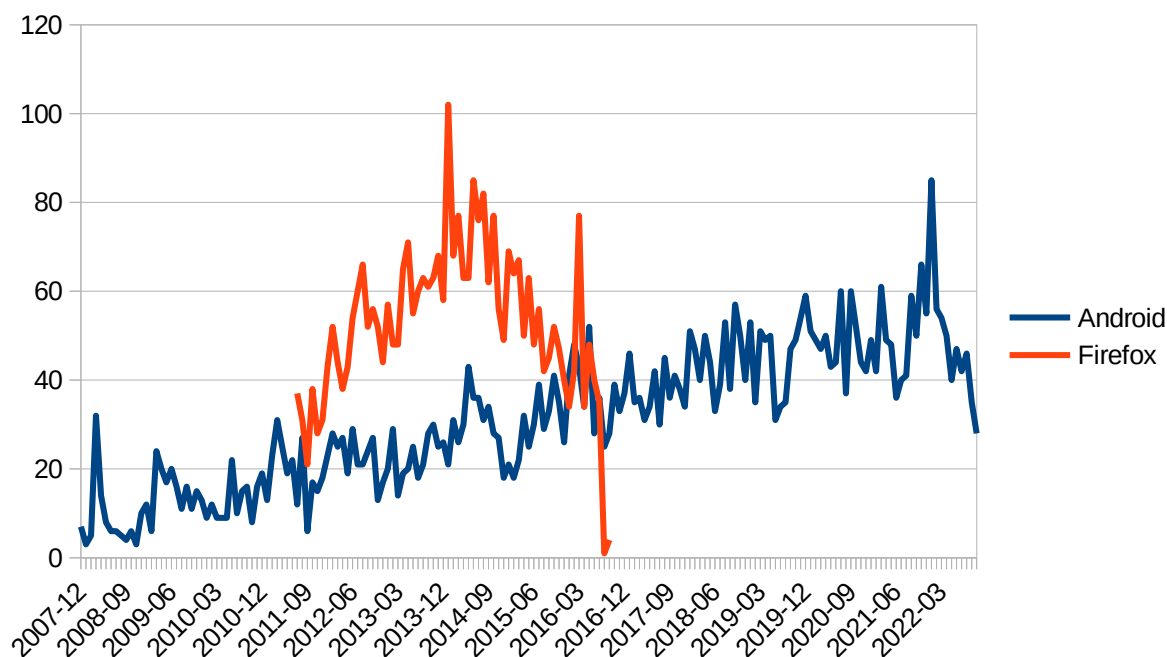
É esperado que projetos de *software* apresentem maior atividade de desenvolvimento de código em suas etapas iniciais de desenvolvimento e uma menor atividade durante sua manutenção a longo prazo. É esperado ainda que, durante a preparação de novas versões com significativa quantidade de novos recursos, um projeto apresente uma elevação em sua atividade, seguida de nova redução após o lançamento (GONZALEZ-BARAHONA *et al.*, 2014). Entretanto, a base de código do Android levou aproximadamente 10 anos até sofrer uma redução consistente em seu nível de atividade, o que se apresenta como indicativo de maturação do código, enquanto a base de código do Firefox OS apresenta uma queda no desenvolvimento ainda no ano de 2014, quando o sistema era colocado no mercado e novas parcerias eram realizadas. Esta queda precoce na atividade de desenvolvimento de código pode representar uma redução no interesse de alguns participantes do projeto.

Figura 8 - Número de *commits* mensais nas bases de código do Firefox OS e do Android

Fonte: Elaborado pelo autor, 2022

Foram também analisadas as quantidades de autores no repositório Git do projeto Firefox OS, um indicador utilizado por Midha e Ghapanchi (2015), que o consideram uma função da legitimidade do projeto. O número de novos autores por mês nas bases de código do Android e do Firefox OS são apresentados na Figura 9. Na figura, se observa um crescimento gradual para o Android, desde 2007 até 2022, enquanto o Firefox OS apresenta um padrão de crescimento acelerado entre 2011 a meados de 2014, com quantidades de novos autores mais elevada do que apresentada no projeto Android durante a maior parte do período, atingindo um pico de mais de 70 novos autores no mês de janeiro de 2014, enquanto o Android seguia com menos de 40. Assim como a quantidade de *commits* na base de código do Firefox OS tendeu a reduzir a partir de 2014, a quantidade de novos autores também se reduziu, com ritmo semelhante ao que se elevou previamente. É importante observar que, apesar das semelhanças com o gráfico da Figura 8, o decréscimo precedeu a redução na quantidade de *commits* criados, o que é coerente, uma vez que mais desenvolvedores tendem a gerar maiores níveis de atividade em um repositório de código.

Figura 9 - Número de novos autores mensais nas bases de código do Firefox OS e do Android



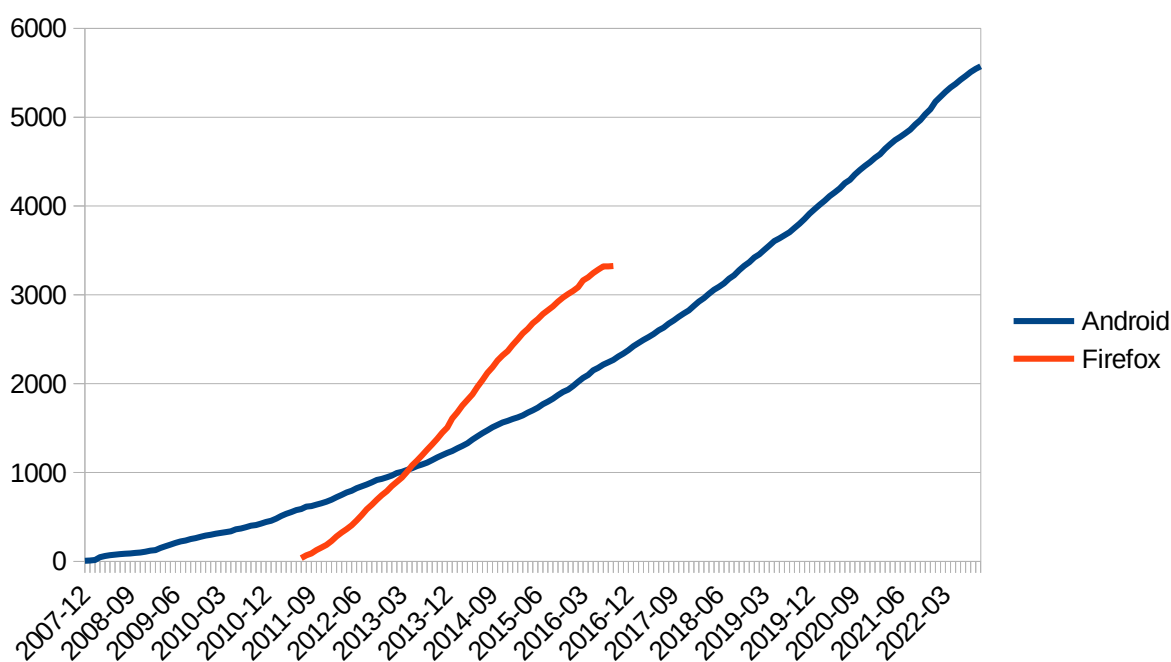
Fonte: Elaborado pelo autor, 2022

A diferença na quantidade de novos autores mensais entre os dois projetos é expressiva. A base de código do projeto Android apenas cruzou a marca de 40 novos autores por mês após mais de 7 anos desde o primeiro anúncio público do sistema operacional, o que a Mozilla conseguiu atingir com apenas 7 meses a partir do primeiro anúncio do seu projeto. O Firefox OS seguiu em crescimento no número de autores e, após a apresentação do seu primeiro *smartphone* conceitual, em fevereiro de 2013, atraiu novos desenvolvedores em quantidades superiores a 50 mensais, chegando a um pico de 102 novos contribuidores em janeiro de 2014, uma marca nunca atingida no repositório Git do Android.

A Figura 10 apresenta a quantidade acumulada de autores para ambos projetos, permitindo uma visualização do crescimento das suas bases de colaboradores. Nela se observa uma curva mais suave para o Android, e de maior crescimento para o Firefox OS. As curvas se cruzam em abril de 2013, quando os projetos se igualaram em quantidade de desenvolvedores de código, 2 meses após a primeira apresentação do *smartphone* funcional com Firefox OS. Já ao ser descontinuado, em setembro de 2016, o projeto constava com 3325 desenvolvedores, enquanto o Android apresentava 2268 para o mesmo momento.

Somente em novembro de 2018 que o Android atingiu a mesma quantidade de desenvolvedores de código.

Figura 10 - Quantidade acumulada de autores nas bases de código do Firefox OS e do Android



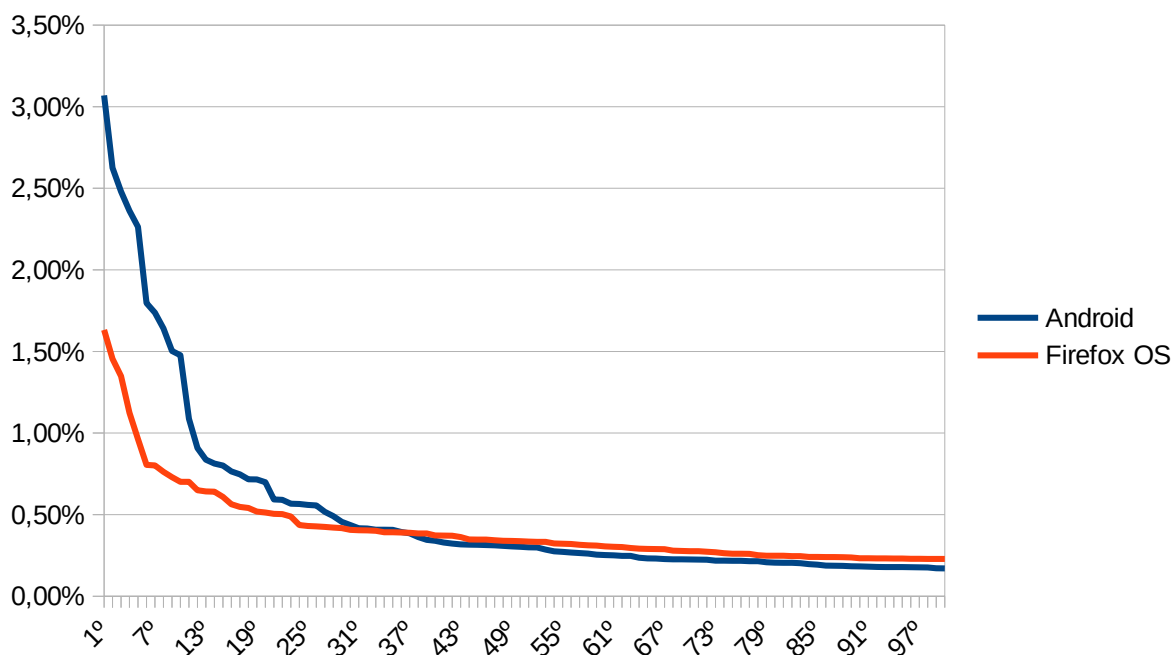
Fonte: Elaborado pelo autor, 2022

Estes dados representam uma maior eficiência na captação de desenvolvedores no projeto Firefox OS durante os meses anteriores à sua entrada no mercado, mas que não se sustentou após o início de 2014. A captação de novos desenvolvedores, ou a atração do projeto, foi apresentada por alguns autores, como Sen *et al.* (2012) e Ghapanchi e Tavana (2015), como um indicador de sucesso em projetos de *software* livre. Sua redução no projeto da Mozilla, imediatamente após a apresentação do seu produto comercializável, um evento em que se espera causar uma maior atratividade, aponta para um fator crítico ocorrido nesse lançamento.

As quantidades totais de *commits* por autor permitem observar se houve uma concentração anormal de tarefas entre os contribuidores do projeto. Excluindo-se *commits* gerados por *bots* automatizadores de tarefas, as quantidades de *commits* por autor são apresentadas na Figura 11. Nela, se observa que o usuário com maior frequência de contribuição no Firefox OS foi responsável por 1,63% do total de *commits* da base de código e este número desce progressivamente, caindo para a metade já na sexta posição, com 0,8% do total de *commits* e se reduzindo em ritmo

mais lento nas posições seguintes, caindo novamente pela metade na 32ª posição, com 0,4% dos *commits*, e chegando à centésima posição ainda acima de 0,2%.

Figura 11 - Quantidade percentual de *commits* por autor nas bases de código do Android e do Firefox OS



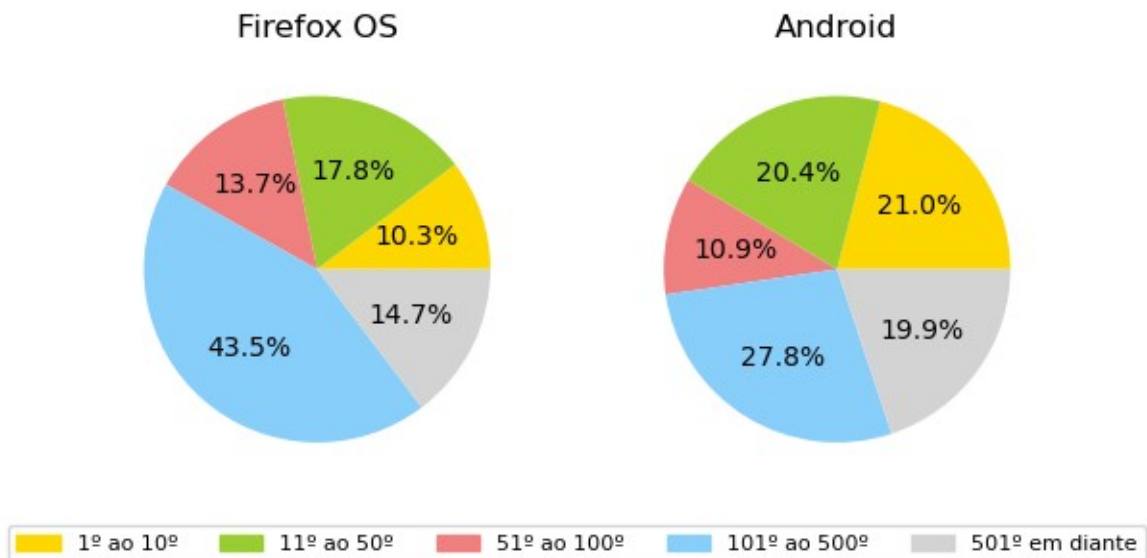
Fonte: Elaborado pelo autor, 2022

Já na base de código do Android, o autor com maior frequência de *commits* foi responsável por 3% do total, caindo para a metade na 9ª posição, com 1,5% e, novamente, na 17ª posição e chegando à centésima posição com 0,17%. Observa-se, portanto, que os autores mais frequentes criaram mais *commits* na base de dados do Android, se comparada com a do Firefox OS, e que este número também decaiu mais rapidamente, chegando à centésima posição em valores semelhantes nos 2 projetos. Nesse sentido, é possível identificar uma concentração de tarefas mais elevada no projeto Android.

Os mesmos dados podem ser observados em maior nível de detalhe ao se agregar as contribuições em grupos de desenvolvedores mais ativos, como apresentado na Figura 12. Dos 10 desenvolvedores do Firefox OS com mais *commits*, estes foram responsáveis por 10,3% do total da base de código, enquanto do 11º ao 50º contribuidor mais frequente, há um total de 17,8% dos *commits*, do 51º ao 100º, um total de 13,7%, do 101º ao 500º, um total de 43,5% e os demais

contribuidores, totalizando 14,7% dos *commits*.

Figura 12 - Percentual de *commits* por grupos de autores de contribuições mais frequentes



Fonte: Elaborado pelo autor, 2022

Observa-se também que na base de código do Android, os 10 usuários com mais *commits* são responsáveis por 21% do total, um valor próximo do dobro do encontrado a base de código do Firefox OS. Já se agregando do 11º contribuidor mais frequente ao 50º, há um total de 20,4% dos *commits*, e do 51ª ao 100º, um total de 10,9%, valores mais próximos dos encontrados no Firefox OS. Entretanto, o agregado do 101º ao 500º contribuidor mais frequente foi responsável por um total de 27,75% dos *commits*, um valor 15,6% menor do que o encontrado no Firefox OS. Já os demais contribuidores, formam 19,90% dos *commits*.

É esperado que haja alguma concentração de *commits* em alguns usuários, e que a maioria dos demais seja responsável por pequenas contribuições (MAJUMDER *et al.*, 2022). Isto se dá porque contribuidores externos não trabalham ativamente no projeto e, muitas vezes, realizam apenas uma ou algumas contribuições esporádicas. Contudo, uma concentração muito elevada de atividade de desenvolvimento em um número muito reduzido de pessoas poder ser um indicador negativo para um projeto de grande porte, mas o mesmo não foi

observado na base de código do Firefox OS. Isto significa haver uma maior distribuição de tarefas entre os desenvolvedores de código do Firefox OS, ou seja, o projeto demonstrou eficiência na coordenação dos seus participantes, o que foi identificado na literatura (GHAPANCHI; AURUM, 2012a; GHAPANCHI; WOHLIN; AURUM, 2014; JIAU; KAO, 2009; MIDHA; PALVIA, 2012) como um fator positivamente influente no sucesso em iniciativas de *software* livre.

Parte dessa diferença pode se dar devido ao Firefox OS ter sido um projeto com um formato de desenvolvimento mais aberto ao público, se aproximando do modelo descrito por Raymond (2001) como bazar, enquanto o projeto Android possui um modelo de desenvolvimento menos aberto a contribuições externas, mais se aproximando do modelo catedral¹³.

Com a análise dos dados da base de código do Firefox OS, pode-se identificar um projeto ativo, de comunidade de desenvolvedores crescente e engajada, demonstrando uma boa saúde nos critérios técnicos, que não podem explicar a sua descontinuação por parte da Mozilla. Este resultado chama a atenção para o fato de que parte significativa da literatura aponta os fatores técnicos como importantes identificadores do sucesso de um projeto de *software* livre, mas observa-se que, isoladamente, este critério pode ser falho.

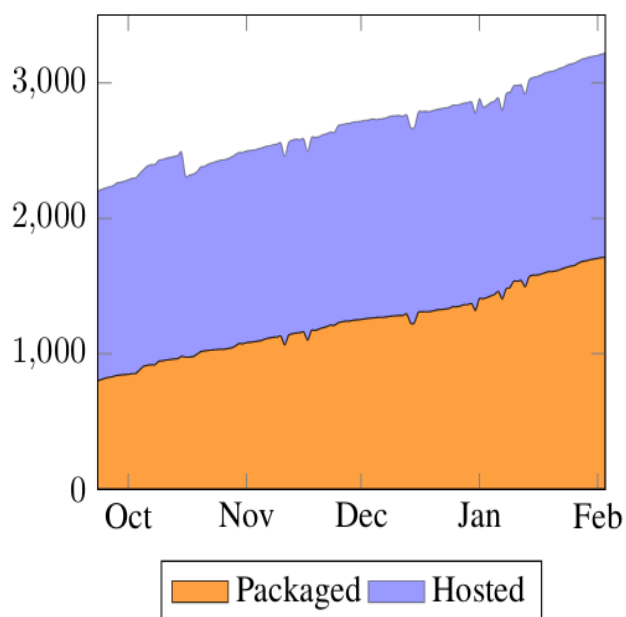
6.2.5 Indicadores mercadológicos

O sucesso mercadológico é uma importante face do sucesso geral de um projeto de *software* livre, normalmente medido por número de *downloads* e acessos às páginas *web* do projeto. Como o número de *downloads* não se aplica ao projeto estudado, uma vez que foi distribuído juntamente de uma plataforma de *hardware*, foram buscados outros indicadores para a absorção do sistema operacional no mercado, buscando uma compreensão multifacetada do seu desempenho.

¹³Os modelos catedral e bazar foram descritos por Raymond (2001) como um duas formas distintas de se desenvolver *software* livre. O modelo catedral refere-se a uma abordagem centralizada e hierárquica no desenvolvimento de *software* livre, liderada por um pequeno grupo de desenvolvedores-chave, enquanto o modelo bazar trata-se de uma abordagem descentralizada e colaborativa, em que não há restrições para que qualquer pessoa possa contribuir com o projeto.

Dada a natureza do projeto, sua inserção no mercado se dava em duas frentes: a de vendas para o usuário final e a de captação de desenvolvedores de aplicativos móveis para que criassem aplicativos *web* e os distribuíssem aos usuários finais. Não se encontra dados suficientes sobre as vendas de dispositivos, e poucos dados sobre o desenvolvimento de aplicativos para o Firefox OS permanecem acessíveis, uma vez que as fontes de aplicativos não se encontram mais disponíveis. Entretanto, Defreez *et al.* (2014) contabilizaram a quantidade de aplicativos disponíveis na loja de aplicativos do Firefox OS, identificando pouco mais de 2000 aplicativos disponíveis em outubro de 2013, crescendo para em torno de 3000 aplicativos em fevereiro de 2014.

Figura 13 - Número de *apps* publicados na loja de aplicativos do Firefox OS entre outubro de 2013 a fevereiro de 2014



Fonte: Defreez *et al.* (2014)

Em contrapartida, a loja de aplicativos oficialmente disponível para o Android, a Play Store, atingiu uma marca de 30000 aplicativos em seu primeiro ano de atividade, ou seja, um número 10 vezes superior ao máximo atingido pela loja disponível para o Firefox OS. Em 2013, quando o Firefox OS foi lançado comercialmente, a loja de aplicativos do Android atingiu a marca de 1 milhão de

aplicativos publicados, um número que cresceu para mais de 2 milhões no ano do encerramento do projeto da Mozilla (STATISTA, 2022). Este crescimento da base de aplicativos do Android demonstra o interesse dos desenvolvedores em publicar para a plataforma, o que não foi observado em proporção semelhante para o Firefox OS.

Um dos indicadores utilizados para medir interesse do público em um produto é o Google Trends (JUN; YOO; CHOI, 2018), que permite a obtenção de métricas de interesse de pesquisa por tópicos no seu mecanismo de busca. Os dados disponibilizados são organizados considerando as diferentes possibilidades de palavras-chave pesquisadas para o mesmo tópico, e permitem observar ao longo de toda a base de tempo, a partir da qual foram coletados. O indicador de interesse de buscas é calculado a partir da quantidade total de buscas no período em relação ao máximo histórico de buscas pelo tópico, variando de 0 a 1, sendo o máximo histórico sempre 1.

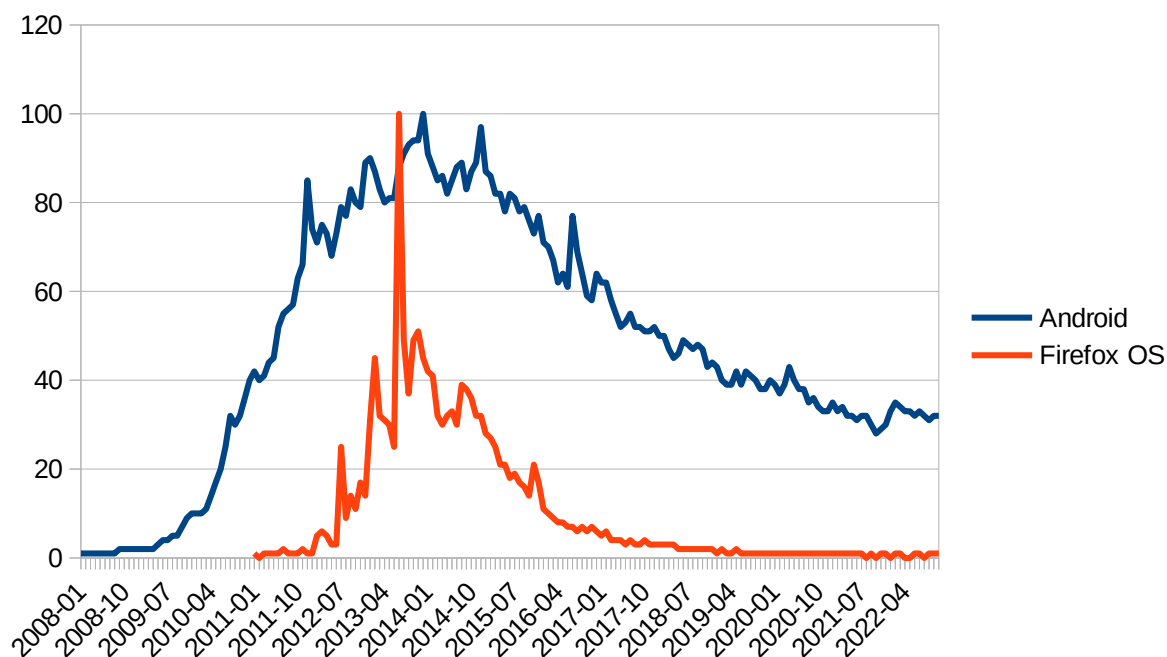
A Figura 14 apresenta o interesse de pesquisa no buscador Google para os sistemas operacionais Firefox OS e Android. Observa-se um crescimento de interesse de busca para o sistema da Mozilla a partir do primeiro lançamento da prova de conceito na MWC 2012, seguido de um forte crescimento no ano seguinte, no mês da MWC 2013, mas ainda em crescimento, até o máximo histórico, que ocorreu em julho de 2013, quando a Mozilla lançou seus primeiros dispositivos no mercado para o consumidor final. A partir dessa data, os interesses de busca se reduziram já para 49% no mês seguinte, e progressivamente, até valores próximos a 0. Essa rápida perda de interesse após seu lançamento pode se apresentar como um indicativo de insatisfação com o produto apresentado. Quando houve o anúncio do fim do projeto, em setembro de 2016, não se identifica atividade mais elevada nas buscas, um indicativo de uma fraca atenção do público ao projeto, já que este fato relevante não gerou interesse de busca acima do usual.

A curva de interesse de busca para o sistema Android apresenta formato semelhante, mas com duas diferenças. O ponto máximo levou mais de 5 anos para ser atingido, enquanto o máximo do Firefox OS foi atingido em menos de 3 anos, indicando uma perda de interesse prematura para este último. A segunda diferença se dá no comportamento da curva após atingir seu ponto máximo. O interesse de busca para o Android decresceu gradualmente, tendendo a uma estabilidade em

torno de 30% do máximo histórico, seis meses após sua ocorrência. Já o interesse para o Firefox OS, decaiu mais rapidamente, chegando a 30% do máximo histórico em apenas 1 ano e seguindo em decréscimo, se apresentando em 7% do ponto máximo no mês do anúncio do fim do projeto.

É esperado que o público pesquise mais por um sistema operacional quando este ainda é menos conhecido e carrega uma característica de novidade, e menos quando este se torne uma tecnologia comum, no caso de um projeto bem-sucedido. Tal característica é identificada na curva de interesse de busca para o Android, mas não para o Firefox OS, indicando que este falhou em se manter no interesse dos usuários após deixar de ser uma novidade no mercado.

Figura 14 - Interesse de pesquisa no buscador Google para os sistemas operacionais Firefox OS e Android



Fonte: Elaborado pelo autor, 2022

A elevação do interesse de busca para o Firefox OS se deu durante as fases de conceitualização e desenvolvimento do projeto. Porém, se reduziu abruptamente após as primeiras concretizações do produto e entrada no mercado, o que pode ser um fator indicativo de uma aprovação da ideologia e da proposta do projeto, mas uma rejeição da sua concretização como produto.

Foram ainda analisados os dados de interesse de busca para algumas regiões geográficas do mercado alvo do projeto, observando-se um padrão semelhante, mas com o máximo de interesse de busca ocorrendo em momento posterior ao máximo global. Esses países foram o México, Brasil, Chile e Índia, onde os máximos de atividade ocorreram, respectivamente, em novembro de 2013, novembro de 2013, abril de 2014 e setembro de 2014. Na Venezuela, o máximo ocorreu em agosto de 2013, mas houve um segundo crescimento elevado no interesse de buscas em fevereiro de 2015, a partir de quando houve o decréscimo final. Esse atraso na perda de interesse nas buscas sobre o projeto podem representar que houve uma maior receptividade e interesse pelo produto nos países em desenvolvimento, em relação aos países desenvolvidos, o que é esperado, pois eram o público-alvo do projeto.

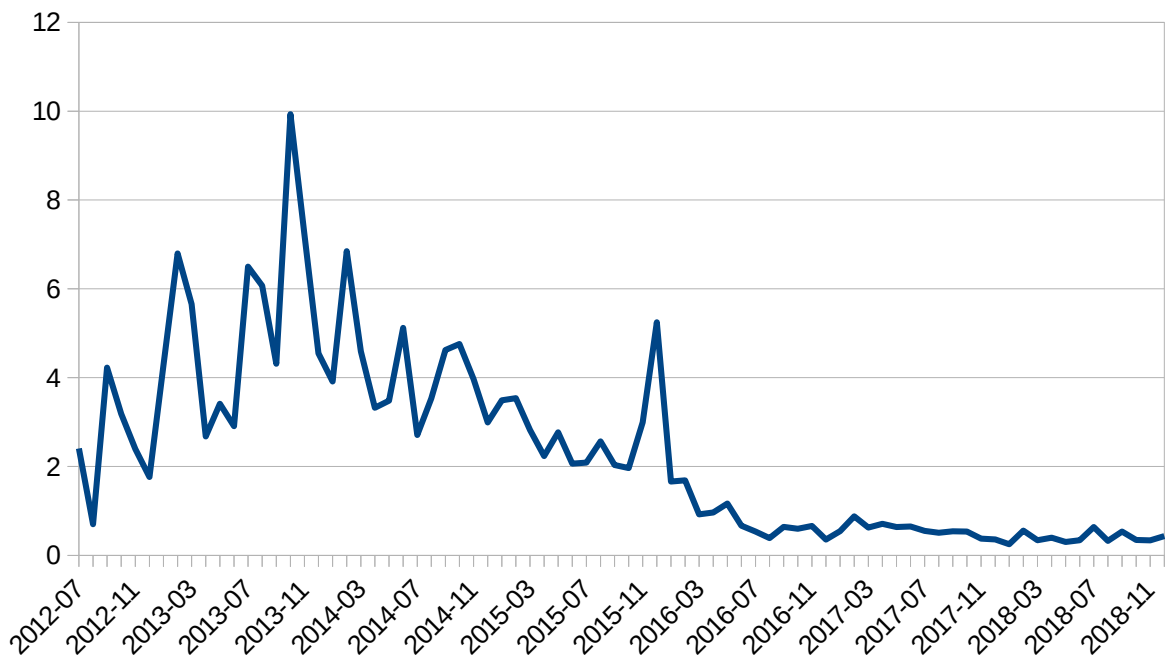
Dados de pesquisa em mecanismos de busca na *internet* podem indicar o interesse das pessoas por obter informação acerca de um tema, mas não captam suas reações a este. Objetivando-se compreender como as pessoas receberam o projeto e que tipos de opiniões formaram sobre o mesmo, foram coletados e estudados comentários da rede social Reddit.

A partir do conjunto de dados obtido, foi possível selecionar comentários que mencionam o projeto Firefox OS e, a partir destes, organizá-los e obter indicadores. O primeiro foi a quantidade de comentários mensais sobre o projeto em relação a todos os comentários realizados na rede, apresentada na Figura 15, na forma de razão de comentários mencionando o Firefox OS para cada 1 milhão de comentários totais no Reddit.

Se observa um padrão semelhante ao encontrado nos dados do Google Trends, demonstrando um crescimento acelerado, com um pico de comentários em fevereiro de 2013, quando foi anunciado os primeiros dispositivos comerciais. Esta alta é seguida de mais um pico em julho de 2013, quando os dispositivos foram lançados, porém, em menor grandeza, seguido de um máximo de atividade em outubro de 2013, para então decrescer. Ocorreram também alguns picos menos elevados de atividade ao longo da redução, principalmente o ocorrido em novembro de 2015, quando a Mozilla apresentou sua mudança de planos para o projeto. O crescimento após o lançamento, até o mês de outubro, se justifica, uma vez que os

usuários tendem a discutir as experiências de uso do dispositivo após seu lançamento. Entretanto, sua redução ainda no mesmo ano reforça a possibilidade de uma insatisfação dos usuários com o produto.

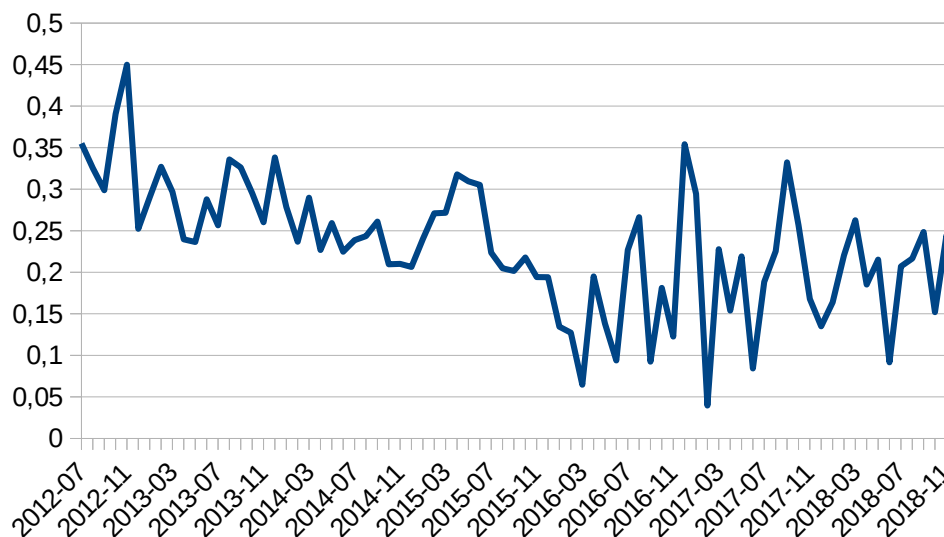
Figura 15 - Quantidade de comentários no Reddit mencionando o Firefox OS para cada 1 milhão de comentários na rede



Fonte: Elaborado pelo autor, 2022

Em seguida, para obter informações não apenas da atividade de interações na rede, mas da forma como as pessoas comentavam sobre o projeto, foi realizada uma análise de sentimento para cada comentário no Reddit que mencionou o sistema operacional Firefox OS, identificados por palavra-chave. A Figura 16 apresenta a intensidade média do sentimento dos comentários na rede social para cada mês. Valores acima de 0,05 representam sentimento positivo, abaixo de -0,05, sentimento negativo e entre ambos valores, neutro.

Figura 16 - Intensidade média mensal de sentimento nos comentários sobre o Firefox OS



Fonte: Elaborado pelo autor, 2022

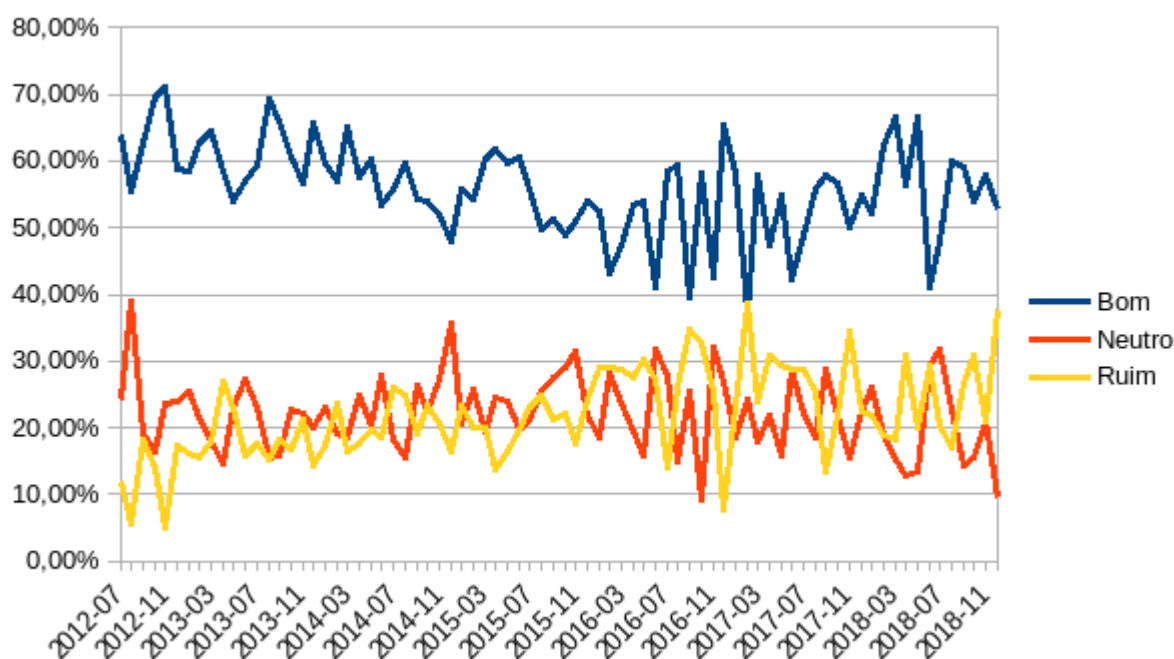
Observa-se um sentimento médio positivo para todo o período analisado, mas cuja intensidade se reduziu ao longo do tempo, até março de 2016, quando o sentimento médio foi apenas marginalmente positivo, com intensidade aproximadamente 7 vezes inferior à medida no início do projeto. Após esse período, o sentimento sofreu alguma elevação, mudando o seu perfil para um de maior oscilação, provavelmente porque a quantidade total de comentários se reduziu consideravelmente nesse mesmo período, levando a um maior peso para cada comentário individual.

Estes dados se destacam em relação aos demais, pois se observa uma redução na intensidade dos comentários já desde 2012, quando foi apresentada a prova de conceito do produto. Esta redução demonstra que, apesar dos usuários comentarem positivamente sobre o sistema, vieram o expressando em intensidade reduzida. Em outras palavras, os usuários seguiram falando bem sobre o projeto, mas não com tanta positividade como antes, o que já indicava que possivelmente havia algo no projeto a reduzir a positividade da percepção do público.

Foram também contabilizadas as quantidades de comentários classificados como carregando sentimentos positivos, negativos, e neutros, apresentadas na Figura 17. É possível observar que a maioria dos comentários possui sentimento positivo, mas que houve uma ligeira redução destes desde o início do projeto,

seguidos de uma maior oscilação a partir de outubro de 2015, similar ao gráfico da Figura 16. Já os comentários com sentimentos neutros e ruins se mantiveram em percentuais mais baixos e em valores próximos entre si, com seus picos abaixo dos 40%, exceto para o mês de março de 2017.

Figura 17 - Proporção mensal de comentários sobre o Firefox OS por tipo de sentimento



Fonte: Elaborado pelo autor, 2022

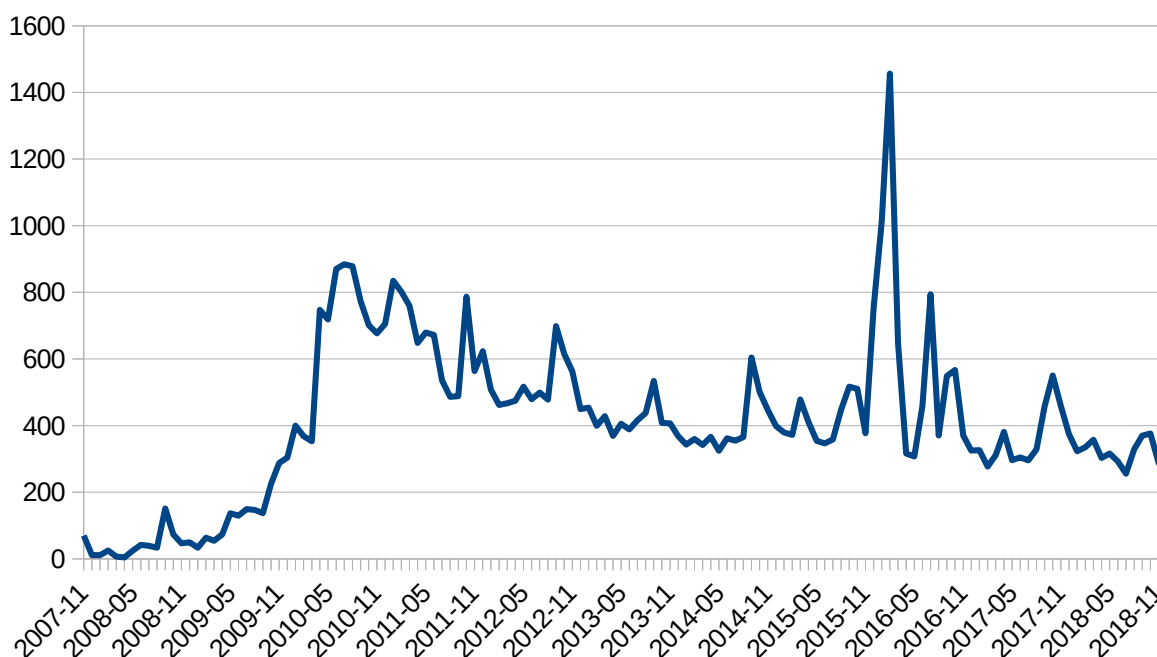
Um sentimento predominantemente positivo pode representar uma boa receptividade do público em relação ao projeto. Entretanto, essa informação se contrasta com a redução no número relativo de comentários na rede e redução da intensidade média de comentários. Isto demonstra um desinteresse do público, à medida que deixaram de conversar sobre o projeto, mas não houve um crescimento de sentimento negativo acompanhando, nem mesmo após o encerramento do desenvolvimento do Firefox OS, com o sentimento se mantendo predominantemente positivo. Além disso, o fato da redução na proporção de comentários positivos não acompanhar a redução da intensidade média de sentimento, conforme Figura 16, significa que esta redução se deu devido à redução da intensidade dos próprios comentários positivos, e não por uma elevação de comentários neutros ou

negativos, reforçando a ideia previamente discutida.

Novamente, os dados foram comparados com os do projeto Android. No gráfico da Figura 18, são apresentadas as quantidades relativas de comentários na rede, em que se pode perceber uma escala numérica muito superior à escala do gráfico da Figura 15, demonstrando uma quantidade muito mais elevada de comentários mencionando o sistema operacional Android, um indicador de maior alcance e engajamento de usuários.

No mesmo gráfico, pode-se observar um crescimento moderado, seguido de uma elevação mais brusca por volta de março de 2010, a partir de quando a quantidade de comentários decaiu gradualmente. Porém, sofreu uma alta de atividade no ano de 2016, quando a Google preparou a versão 7 do Android, com mudanças significativas no comportamento do sistema e da API (GOOGLE, 2020), além da preparação do sistema para o lançamento do seu *smartphone* próprio, o Google Pixel (RAKOWSKI, 2016). A partir de 2017, a quantidade de *commits* seguiu na tendência anterior.

Figura 18 - Quantidade de comentários no Reddit mencionando o Android para cada 1 milhão de comentários na rede

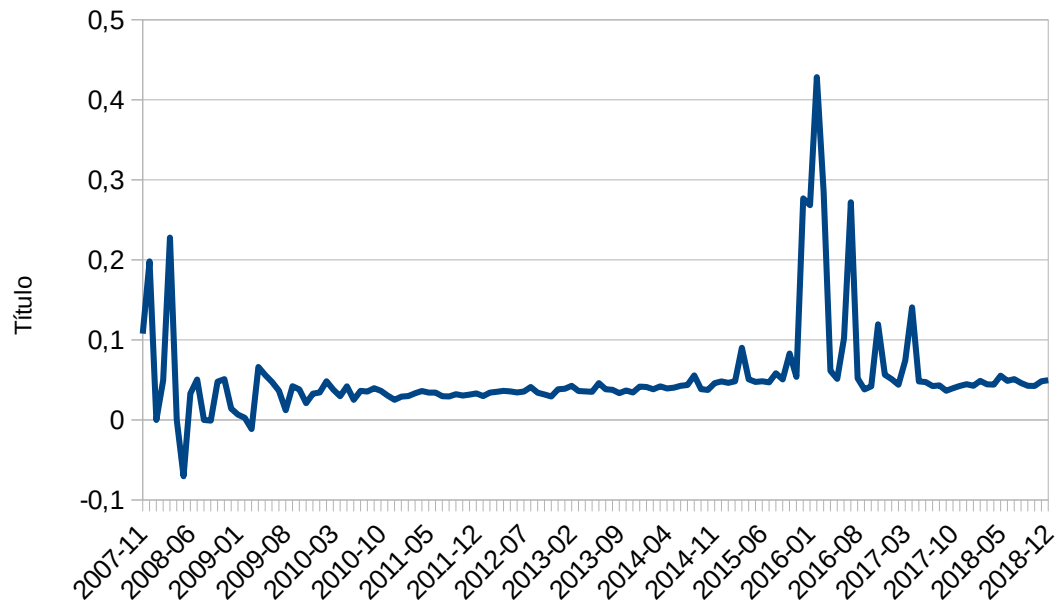


Fonte: Elaborado pelo autor, 2022

Já a intensidade do sentimento nos comentários, sofreu oscilações nos

primeiros meses do projeto, até se estabilizar, apenas sofrendo novas oscilações no ano de 2016, até o início de 2017, com elevação da intensidade dos sentimentos.

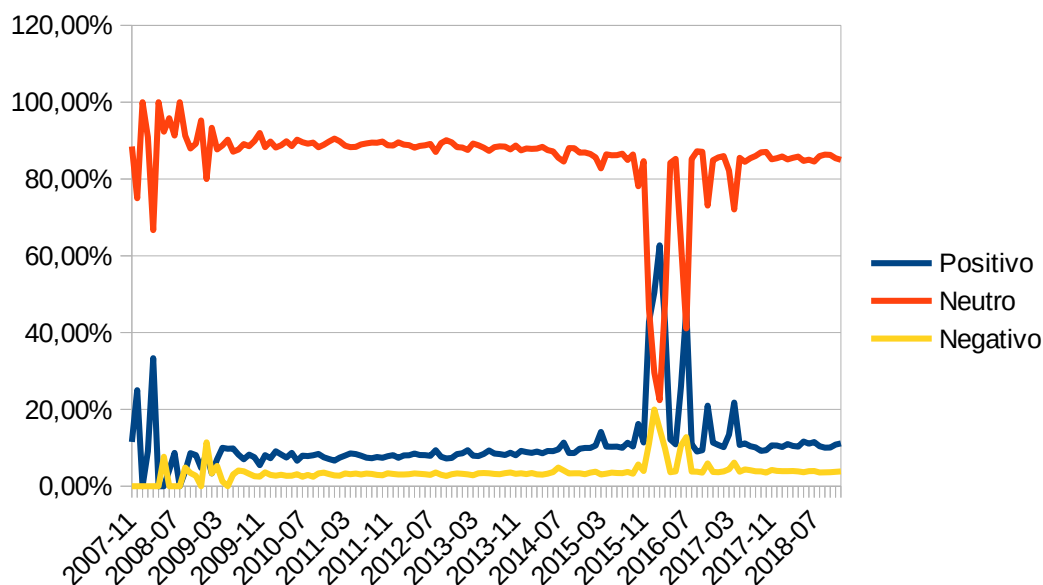
Figura 19 - Intensidade média mensal do sentimento nos comentários mencionando o Android



Fonte: Elaborado pelo autor, 2022

Um contraste em relação ao Firefox OS pode ser identificado na Figura 20, em que são mostradas as quantidades de sentimentos classificados como positivos, negativos e neutros. Para o sistema Android, a maioria dos comentários carrega sentimento neutro, e não positivo como no Firefox OS, sendo os comentários de sentimento positivo e negativo em quantidades mais próximas, mas com uma quantidade ligeiramente superior de comentários com sentimento positivo.

Figura 20 - Proporção mensal de comentários sobre o Android por tipo de sentimento



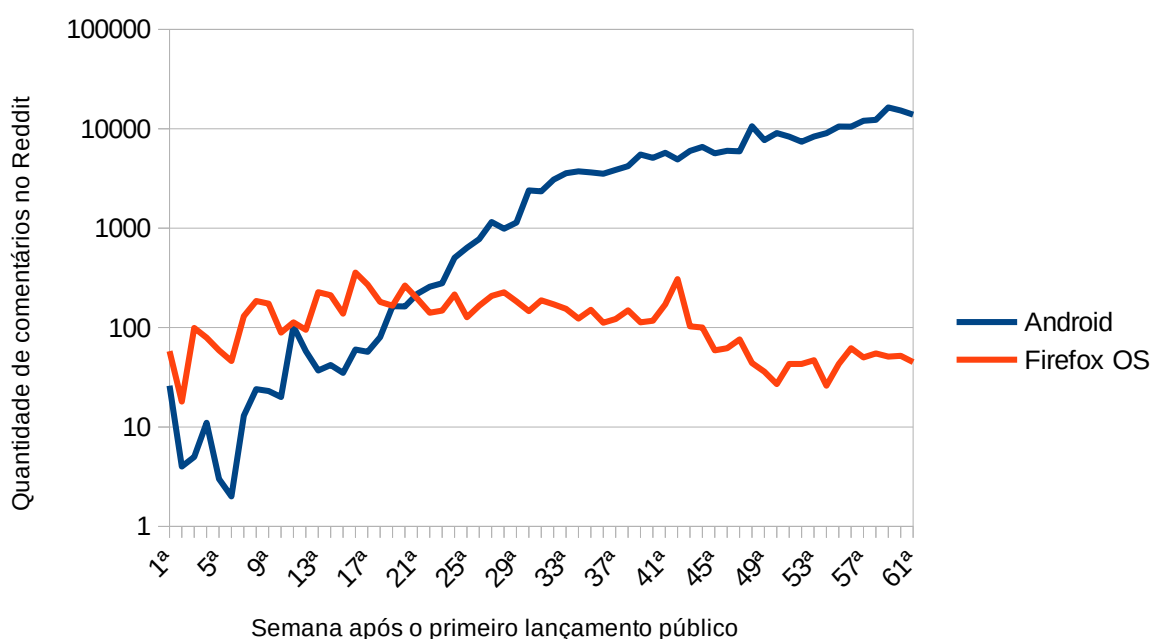
Fonte: Elaborado pelo autor, 2022

Embora a média da intensidade de sentimento nos comentários para o Android seja mais baixa, e a predominância seja de comentários neutros, ao invés de positivos, há uma quantidade mais elevada de comentários relacionados ao sistema, com um total de 2142706, enquanto há apenas 8311 comentários mencionando o Firefox OS. Dessa forma, apesar de possuir uma proporção menos elevada de comentários com sentimento positivo, conforme Figura 20, seu total é de 342887 comentários, o que supera até mesmo a totalidade de comentários sobre o Firefox OS. Esta disparidade sugere a importância de se ter uma ampla base de usuários em um projeto de um sistema operacional, o que não foi alcançado pelo Firefox OS. Observa-se ainda, a elevação do sentimento positivo em 2016, resultante dos mesmos lançamentos já discutidos, o que evidencia a boa recepção das novidades apresentadas no período.

Contudo, ao tomar-se como referência o tempo após o primeiro anúncio público de cada projeto, e comparar a quantidade absoluta de comentários no Reddit entre eles, como demonstrado na Figura 21, observa-se que até o vigésimo mês subsequente do conhecimento público, o Firefox OS apresentou uma maior quantidade de comentários em comparação com o Android, principalmente nos primeiros meses de conhecimento público do projeto. Porém, o Firefox OS não

apresentou o mesmo ritmo de crescimento nas discussões da rede social, decaindo gradualmente a partir do vigésimo primeiro mês, enquanto o Android cresceu substancialmente, ultrapassando os 10000 comentários mensais.

Figura 21 - Quantidade absoluta de comentários no Reddit mencionando os sistemas Android e Firefox OS (escala logarítmica)



Fonte: Elaborado pelo autor, 2022

A data em que a quantidade de comentários sobre o Firefox OS iniciou tendência de redução coincide com o segundo evento MWC, no qual a Mozilla apresentou o produto, em fevereiro de 2014. Neste evento, foi levado ao público os dispositivos com sistema Firefox OS a serem comercializados para o consumidor final, com um destaque para o *smartphone* de 25 dólares. A redução de comentários a partir dessa data pode representar uma perda de interesse do público diante do produto apresentado.

A partir dos agrupamentos de comentários positivos e negativos sobre o sistema operacional Firefox OS, foram selecionados aqueles com maior intensidade positiva e negativa para leitura manual, visando identificar como o projeto era abordado, e em quais contextos.

Foi identificado que, em geral, os comentários com sentimento positivo em relação ao projeto da Mozilla abordavam suas questões ideológicas, como a

proposta de *web* aberta, da possibilidade de criar aplicativos que funcionassem em qualquer dispositivo, do incentivo à inclusão digital, entre outras, enquanto os comentários com sentimento negativo, em geral, abordavam diretamente o produto comercializado no mercado, relatando insatisfações com o *hardware*, travamentos, falhas no *software*, falta de aplicativos para utilizar, entre outras questões.

Em um dos comentários de maior intensidade de sentimento positivo, um usuário com conta excluída (2012, tradução nossa) disse:

Firefox OS é, na minha opinião, uma das melhores coisas a acontecer no cenário de sistemas operacionais móveis. É um sistema operacional verdadeiramente de código aberto, desenvolvido de forma aberta, com uma barreira de entrada muito menor, o que permitirá dispositivos muito mais acessíveis, atualizações muito mais rápidas e impulsionará a *web* aberta que tem sido atacada recentemente com a dominância do Android e iOS.

O comentário sumariza a missão do projeto Firefox OS e mostra o alinhamento do(a) usuário(a) em relação à mesma, o que se observa na maioria dos comentários de sentimento positivo. Este sentimento em relação à missão do projeto não se extinguiu com sua descontinuação, mas persistiu, como pode ser observado no comentário do(a) usuário(a) “brianpeiris” (2015, tradução nossa):

Há um futuro para a plataforma da *web* e são os *apps* progressivos, e o Firefox OS fez as primeiras explorações nesse espaço. Seja como um sistema operacional para celular que pode ser instalado, ou algo menos intrusivo (como o b2gdroid), o Firefox OS continua sendo uma plataforma bastante poderosa para essas "explorações" (e um recurso valioso para futuros produtos, como os da plataforma de dispositivos conectados).

Já os comentários negativos, tenderam a ser mais diretos e focados no produto comercializado, como pode ser observado no comentário de outro usuário com conta excluída (2016, tradução nossa):

Eu gosto dele na teoria, mas a RAM é pequena o suficiente para que muitas

vezes ele trava, e ele ocasionalmente se recusa a permitir que eu desligue o alarme ou atenda o telefone até que eu tenha inserido meu PIN.

Muitos dos comentários de sentimento negativo demonstraram interesse na plataforma, mas uma frustração ao utilizar o produto, como se observa no comentário do(a) usuário(a) “slayerboy” (2014, tradução nossa):

Eu estava realmente esperando que o Firefox OS fosse um sucesso para começar lentamente minha migração para uma alternativa do Google e do Android, mas o telefone ZTE Open vendido pelo eBay é uma piada desastrosa que nem é engraçada e isso arruinou minha impressão do Firefox OS.

Dentre os comentários de sentimento negativo, se observa ainda algumas nuances, como usuários desejando utilizar o *software*, mas não interessados no *hardware* comercializado, mas sim em utilizá-lo em seus próprios equipamentos, o que não era possível em termos práticos. Um exemplo desse tipo de comentário foi realizado pelo(a) usuário(a) “milksteaksonthehouse” (2014, tradução nossa).

Eu estou meio surpreso que o Firefox OS seja tão importante para a diretoria. Parece ser uma baixa prioridade para alguém de fora. Eu não tenho nenhum celular que possa executar o Firefox OS. Ele só suporta modelos de celular que eu nunca vi ou ouvi falar. É difícil fazer as pessoas se animarem com uma plataforma quando você não tem como executá-la.

Este comentário traz à luz um fator abordado na literatura, a portabilidade do *software*, demonstrando que, quando este depende de plataformas específicas, pode reduzir a quantidade de usuários interessados no projeto.

Outro comentário, realizado pelo(a) usuário(a) “sQtWLgK” (2014, tradução nossa), já aborda a percepção de uma incoerência entre a missão da Mozilla com o projeto e o produto comercializado:

Na minha humilde opinião, ambos compreenderam erroneamente o código aberto: código aberto não é apenas deixar as pessoas corrigirem bugs para você, mas também permitir que você modifique o código-fonte e o execute. Os dispositivos com FirefoxOS e Android são vendidos "não-rooteados" e com bloqueios de bootloader que impedem que você modifique seu *software*. As pessoas ainda são capazes de fazê-lo apenas porque os mantenedores do sistema operacional são descuidados o suficiente para deixar exploits não corrigidos.

A incoerência se dá a partir do fato de que a Mozilla se objetivava a desenvolver uma plataforma aberta, livre de restrições provenientes de *software* proprietário, de forma que qualquer indivíduo pudesse desenvolver sobre ela e adaptá-la às suas necessidades. Entretanto, o(a) usuário(a) "sQtWLgK" identificou o que ele(a) considera uma incoerência entre o produto desenvolvido e a missão do projeto, uma vez que, apesar do código-fonte ser livre, não se podia construir e executar versões modificadas do Firefox OS, já que a plataforma de *hardware*, necessária para a execução do *software*, por sua vez, não era aberta, não permitindo a realização de modificações e sua execução.

Em contrapartida, alguns usuários de Android tinham uma percepção deste como um produto aberto, como demonstrado no comentário do(a) usuário(a) "employeen05" (2011, tradução nossa):

O Android é um (potencialmente, dependendo de como é empacotado) sistema operacional completamente *open source*, utilizando o *kernel* Linux e nesse sentido acho que é ótimo. Quando a maioria dos fabricantes de hardware e operadoras terminam de modificá-lo, ele não é tão livre e aberto quanto antes, mas qualquer pessoa pode baixar, compilar e personalizar o código-fonte a seu gosto, e supondo que você tenha um dispositivo rooteado (todos eles são rooteados mais cedo ou mais tarde), você pode utilizá-lo como quiser. Eu acho que é uma ótima opção e estou realmente satisfeito com o sucesso. Só gostaria que mais pessoas que realmente gostam do Android entendessem a importância e a vantagem de algo mais aberto (do que, digamos, o iOS), pois isso poderia ajudar mais pessoas a ter mais interesse por outros *softwares* e sistemas operacionais de código aberto.

Este comentário, realizado em 2011, demonstra como o(a) usuário(a) percebia o Android como um sistema operacional mais aberto do que seu principal concorrente. Até o ano de 2012, o Android ainda não era a plataforma dominante no

mercado, que era ocupada pelo iOS, cuja plataforma é proprietária e apresenta restrições quanto a modificações de *software* e *hardware*, além de maior dependência do usuário com a empresa desenvolvedora. Dessa forma, ao entrar no mercado como uma plataforma de código aberto com um nível de personalização e flexibilidade superiores aos do iOS, o Android ganhou reconhecimento por oferecer maior liberdade ao usuário.

Além disso, comparação com o sistema proprietário iOS é a principal característica dos comentários sobre o sistema Android, evidenciada ao se analisar os comentários de maior sentimento positivo para o Android, em que se observa uma frequente comparação deste com seu principal concorrente no mercado. Estes comentários tendem a exaltar características positivas tanto do *software* quanto dos equipamentos de *hardware* comercializados com o sistema, como pode ser observado no comentário do(a) usuário(a) “The_Gecko” (2010, tradução nossa):

Eu tenho um HTC Hero. Eu o amo. Quando eu estava procurando por um novo telefone, eu estava tendendo a comprar um iPhone, mas 5 minutos brincando com o Hero do meu amigo me convenceram. Eu prefiro muito mais a ele do que um iPhone por muitas razões, principalmente porque você tem muito mais liberdade com um telefone Android. Meu conselho é, se possível, brincar com um iPhone e um telefone Android, senti-los. Leia avaliações. GSM arena é um bom lugar para começar. No que diz respeito aos telefones HTC, eu os recomendo. Não o Hero agora, existem melhores; o Desire ou o Legend. O player de música é bom, a memória é expansível. Eles têm o Android 2.1 que = (sic) bluetooth. Ou você poderia pegar o Droid, que é, como eu entendi, um excelente telefone.

O comentário ainda demonstra como os usuários elogiavam as características do *hardware* oferecido, o que não foi identificado nos comentários positivos sobre o Firefox OS.

Já os comentários negativos sobre o Android, também tenderam a compará-lo com o sistema iOS, mas com uma opinião favorável a este último, como é exemplificado pelo comentário do(a) usuário(a) “OverlordXenu” (2011, tradução nossa):

E, por essa questão eu odiava o Android, (sic) era tão deficiente em comparação com o iOS, nada era intuitivo e depois de uma semana eu ainda não tinha descoberto como marcar um e-mail como não lido. Eu já disse antes e vou dizer de novo: O Android é aberto apenas para as operadoras e fabricantes de dispositivos. Ignorando o Nexus, você tem que fazer algo semelhante ao jailbreak para fazer modificações extensas, carregar uma nova ROM, etc. Isso anula a garantia. O Google corrompeu (o termo) "aberto" para fins de marketing. O Android, em seu estado em um telefone médio, viola o núcleo da filosofia FOSS. Você não "possui" seu dispositivo. A única maneira pela qual um telefone médio é mais aberto do que um iPhone é a capacidade de instalar aplicativos de fontes externas (o que deveria, admitidamente, ser uma opção no iOS).

O comentário não apenas demonstra um sentimento negativo em relação ao sistema Android, mas demonstra uma insatisfação com o nível de abertura da plataforma, questionando o fato do *software* ser desenvolvido na forma de código aberto, mas distribuído em uma plataforma fechada, que impede modificações e contradiz os princípios do *software* livre. Este sentimento está em acordo com a percepção da Mozilla que levou à proposta do Firefox OS de criar um sistema operacional para *smartphones* completamente livre. A existência deste tipo de sentimento dentre usuários do Android pode explicar por que o Firefox OS atraiu tanta atenção durante sua conceptualização e produziu um sentimento geral positivo. Nesse sentido, a apresentação, pela Mozilla, de um produto que, no sentido prático para o usuário, apresentava o mesmo nível de liberdade e personalização que um dispositivo Android, como evidenciado pelo(a) usuário(a) “sQtWLgK”, corrobora a ideia de que o projeto falhou em criar um produto que transmitisse os objetivos destacados pela fundação, de criar um sistema operacional completamente livre, pois o produto final criado não apresentou diferenciação, quanto a este aspecto, em relação aos dispositivos Android já comercializados.

De forma geral, o sentimento que pode ser observado nas discussões na rede social Reddit em relação ao projeto Firefox OS é de que a Mozilla possuía uma missão nobre e demandada, mas falhava em gerar utilidade para o seu produto. Este sentimento pode ser exemplificado com o comentário acalorado do(a) usuário(a) “Brian4LLP” (2014, tradução nossa), feito em resposta a um *post* mencionando o Firefox OS: “Não, novamente, eu ainda acho que você está bêbado. Firefox OS é uma solução procurando um problema. Isso não vai mudar.”. Este comentário foi realizado em 2014, ano em que o Firefox OS teve suas primeiras

unidades voltadas ao consumidor final comercializadas.

Este ponto de vista demonstra a percepção do(a) usuário(a) do Firefox OS como um produto bom e inovador, quando se refere ao *software* como uma “solução”, mas demonstra sua percepção de falta de utilidade gerada pelo produto apresentado no mercado, quando afirma que esta “solução” está “esperando por um problema”.

Os comentários realizados na rede social Reddit correspondem com as demais informações obtidas sobre a recepção do projeto no mercado. A correspondência se deu principalmente em relação ao interesse de busca na *internet* e à quantidade de comentários no Reddit, que se elevavam substancialmente no início do projeto, provavelmente motivados pelo sentimento positivo dos usuários em relação aos objetivos da Mozilla com o Firefox OS, mas que começaram a decair a partir do primeiro trimestre de 2013, quando houve a primeira apresentação pública de um dispositivo executando o sistema operacional, possivelmente motivados pela insatisfação dos usuários com o produto apresentado. Mesmo com a proporção de sentimentos bons para ruins se mantendo, houve uma redução na intensidade dos bons sentimentos a partir desse mesmo período.

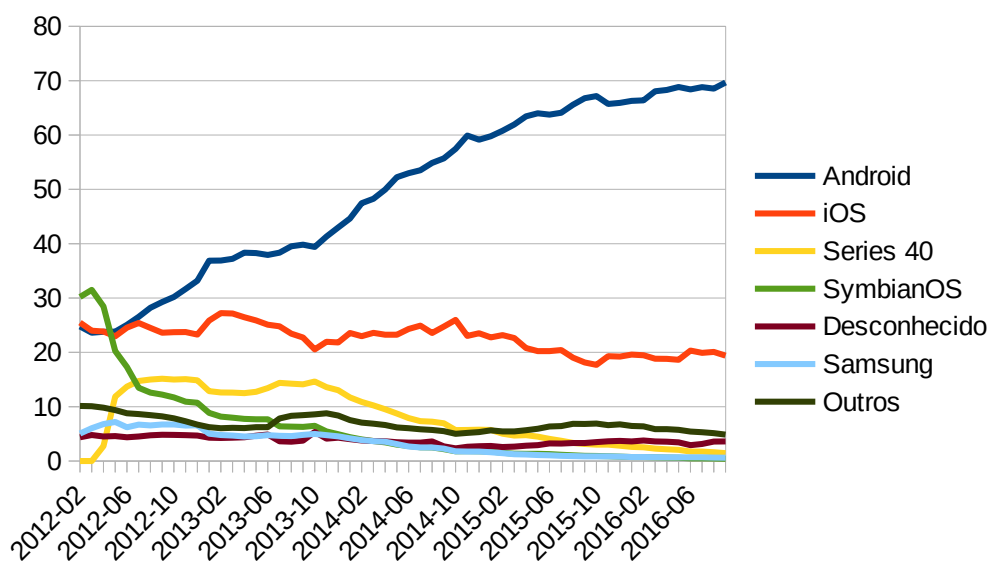
A análise da recepção mercadológica do Firefox OS muito se contrastou com os resultados da análise da produção técnica, sugerindo uma dificuldade dos desenvolvedores do projeto e da Mozilla em perceber estas tendências, já que apenas a partir de 2015, após perda significativa de interesse dos usuários, se observou tentativas de alterar os rumos do projeto. Contudo, as medidas tomadas não focaram nas principais frustrações apresentadas pelos usuários.

6.2.6 O desafio do Firefox OS diante do crescimento do Android

O período em que foi desenvolvido o projeto Firefox OS coincidiu com o período de maior crescimento, em base de usuários, do sistema operacional Android, e sua consolidação como plataforma dominante no mercado global. Na Figura 22, são apresentados os dados de cota de mercado medidos pelo

StatCounter (2022a) a partir de fevereiro de 2012, até setembro de 2016, período de desenvolvimento oficial do Firefox OS, e pode-se observar que houve uma rápida expansão do Android, partindo de menos de 30% do mercado para aproximadamente 70%. Observa-se ainda que, com exceção da plataforma iOS, a segunda dominante no mercado, todos os competidores da época perderam espaço e se reduziram, tendendo ao desuso. É importante destacar que a base de usuários do Firefox OS foi pequena suficiente para sequer ser detectada como uma categoria própria, sendo agrupada na categoria “outros”.

Figura 22 - Evolução da participação no mercado de diferentes sistemas operacionais móveis



Fonte: StatCounter (2022b)

O Android cresceu fortemente a partir de 2012, se apresentando para os usuários como uma plataforma mais acessível e aberta em relação ao iOS, ao mesmo tempo que competitiva, de forma que fabricantes de *hardware* construíram suas cadeias produtivas de *smartphone* sobre esta plataforma, assim como desenvolvedores de aplicativos, construíram sua base de conhecimento sobre o Android. Estas características constituíram uma espécie de inércia no crescimento do sistema, que seria dificilmente contida naquele momento histórico em que se expandia para diferentes mercados, alcançando públicos de países emergentes e não usuários de *smartphones* dos países desenvolvidos. É importante destacar que o Android se portava como uma alternativa mais barata em relação ao seu principal

concorrente, o iOS, o que contribuiu para a conquista do mercado nos países emergentes.

Ao se analisar as características do desenvolvimento do Android em relação aos fatores estudados neste trabalho, é possível desenvolver uma maior compreensão da solidez do seu crescimento, já que o projeto demonstra características identificadas como motivadoras do sucesso.

Algumas dessas características foram apresentadas e discutidas na seção 6.2.4. Dentre elas, estão a execução de tarefas e a coordenação dos desenvolvedores e colaboradores, inferidas a partir de quantidade e distribuição de *commits* em sua base de dados. O Android apresenta uma atividade estável de desenvolvimento, com elevação de atividade quando há novidades em desenvolvimento e estabilização quando há versões menores a serem trabalhadas. A distribuição de *commits* por desenvolvedor apresentou uma maior concentração de tarefas no Android, em comparação com o Firefox OS, mais ainda contou com aproximadamente 20% de contribuições entre os desenvolvedores não inseridos no grupo dos 500 mais frequentes, além de contar com mais de 5000 desenvolvedores registrados em sua base de código, um número que veio crescendo estavelmente desde seu primeiro lançamento.

O projeto apresenta ainda outros fatores motivadores do sucesso, como o uso de licença permissiva (Apache 2), relacionada a uma maior atratividade de desenvolvedores durante os estágios iniciais; uma alta portabilidade do *software*, uma vez que é compatível com celulares, *tablets*, televisores, entre outros dispositivos móveis, além de compatível com computadores e emuladores; tradução para diferentes idiomas, o que está relacionado à popularidade; e uso de linguagens de programação populares.

Contudo, é possível que outros fatores tenham contribuído para o alcance de uma posição hegemônica no mercado, incluindo práticas que podem ser consideradas anticompetitivas. Nesse sentido, observa-se que, apesar de ser um projeto de *software* livre, o Android é distribuído juntamente com código proprietário e sobre plataformas de *hardware* fechadas, com restrições de modificação de *software*, impedindo os usuários de realizarem a substituição do sistema distribuído nos *smartphones* por padrão, diferentemente da forma conforme possível em

equipamentos do tipo *desktop*.

Além deste impedimento, o ambiente de execução de aplicativos do Android não é compatível com outras plataformas, diferenciando-se novamente da plataforma *desktop*, em que o usuário pode instalar em sua máquina o sistema operacional de sua escolha. Desse modo, com o seu crescimento no mercado, acompanhado da criação de uma grande base de aplicativos, o Android causa um efeito de dependência dos usuários sobre sua plataforma. Isto dificultando a concorrência, uma vez que o uso de um novo sistema impede, na maioria das vezes, o reaproveitamento de *hardware*, além de ser incompatível com os aplicativos já criados e mantidos pelos desenvolvedores. Esta incompatibilidade se torna mais expressiva quando se trata de aplicativos para funções essenciais, como acesso a contas bancárias ou serviços governamentais, criando um fator de afastamento do uso de outras plataformas, já que experimentar um novo sistema se traduz em perder acesso imediato a esses serviços.

Tais práticas recebem críticas e já foram motivadoras de ações judiciais contra a Google, como a multa por condutas anticompetitivas, imposta pela Comissão de Competição da Índia (COMPETITION COMMISSION OF INDIA, 2022). A comissão acusou a empresa de abusar da sua posição dominante no mercado, incluindo o monopólio sobre a distribuição de aplicativos e a distribuição de uma suíte de aplicativos proprietária pré-instalada, enquanto cria restrições para que aplicativos de terceiros realizem as mesmas funcionalidades, além da realização de acordos para manter seu serviço de busca como o padrão. Estas práticas foram consideradas pela comissão como dificultadoras da concorrência e criadoras de limitações aos fabricantes no desenvolvimento de suas próprias versões do Android. A Google vem, desde 2015, sofrendo uma série de processos baseados em leis antitruste e por práticas anticoncorrência em diferentes locais no mundo, incluindo a União Europeia, os Estados Unidos da América, Coreia do Sul, Rússia e Canadá (PEROTTI, 2017).

Contudo, essas práticas haviam sido antecipadas pela Mozilla e integravam parte da missão com o projeto Firefox OS, que buscava criar uma plataforma não apenas de código livre, mas que também estimulasse a livre concorrência. Entretanto, as restrições impostas pela Google sobre a plataforma Android, junto do

seu forte e acelerado crescimento no mercado, contribuíram para dificultar a entrada do Firefox OS como um competidor. A Mozilla não foi a única a buscar a participação no mercado de *smartphones*, e todas as outras iniciativas observadas na Figura 22, criadas por diferentes empresas, como a Microsoft, Nokia e Samsung, não foram bem sucedidas, consolidando-se apenas a Google e a Apple no mercado.

Dessa forma, o insucesso do Firefox OS em se estabelecer no mercado de *smartphones* não se tratou de um caso isolado, mas de um padrão observado em outros projetos, o que pode ser um indicativo de uma barreira de entrada neste segmento, representando um desafio adicional para qualquer competidor.

6.2.7 Sumarização

O projeto Firefox OS foi desenvolvido pela Mozilla com o objetivo de criar um sistema operacional livre para *smartphones*, baseado nos padrões abertos da *web*. A Mozilla considerava que a falta de abertura das plataformas de *smartphone* era um problema, o que se alinhava com o pensamento de muitos usuários, se apresentado, desse modo, como uma boa escolha, o que é demonstrado na forma positiva como esta missão é discutida pelos usuários.

Do ponto de vista do planejamento técnico, o uso de tecnologias e linguagem de programação dos padrões *web* também foi um fator positivo para o projeto, além, do uso de uma licença semi-restritiva, fatores que tendem a atrair mais desenvolvedores para projetos de *software* livre. Ademais, o *software* desenvolvido possuía ainda algum grau de modularidade, sendo composto por diferentes componentes que se comunicavam, formando o sistema completo. A modularidade foi igualmente identificada como um fator positivamente relacionado ao sucesso técnico dos projetos de *software* livre.

A relação dos fatores pesquisados na literatura com as características identificadas no Firefox OS é apresentadas na tabela 2, na qual se pode observar como os mesmos foram encontrados no projeto, exercendo influência positiva, exceto pela fraca portabilidade do *software*.

Tabela 2 - Relação de fatores influentes do sucesso no projeto Firefox OS

Fator influenciador do sucesso	
Execução de tarefas	Foi identificada elevada atividade de desenvolvimento, o que indica uma igualmente elevada execução de tarefas.
Tipo de licença	Uso de licença semi-resritiva, o que eleva a atratividade do projeto durante suas fases iniciais.
Coordenação de equipe e colaboradores	Foi identificada uma maior distribuição de tarefas entre os desenvolvedores do projeto, em comparação com o Android.
Portabilidade	O sistema não permitia sua execução na maior parte das plataformas existentes, contando apenas com um simulador e a plataforma comercializada.
Linguagem de programação utilizada	Foram utilizadas principalmente linguagens <i>web</i> para o desenvolvimento, o que, devido à sua popularidade, torna o projeto mais atrativo para novos desenvolvedores.
Nicho de mercado	O projeto foi desenvolvido objetivando o público de pessoas excluídas digitalmente em países em desenvolvimento, o que representava um mercado numeroso e em expansão, além de se tratar de uma área popular da tecnologia.

Fonte: Elaborado pelo autor, 2022

Dessa forma, a elaboração inicial do Firefox OS se demonstrou eficiente e alinhada com os principais fatores identificados na literatura, mas foi a partir da construção do produto e da execução do projeto que este perdeu tração. O produto desenvolvido não correspondeu com as expectativas dos usuários, o que gerou frustrações e afastamento progressivo de pessoas interessadas no sistema. Este afastamento pode ser identificado nas reduções de interesse de pesquisa e na quantidade de comentários sobre o projeto, realizados no Reddit.

Muitos usuários esperavam um *hardware* de elevada qualidade e igualmente aberto, mas receberam um equipamento bastante limitado nas suas especificações e com restrições de acesso, de tal forma que não se podia instalar sua própria versão modificada do Firefox OS. Isto fez com que alguns usuários questionassem a ideologia do projeto, considerando esta prática desalinhada com a missão apresentada pela Mozilla.

Outro motivo de frustração foi a impossibilidade de instalar o Firefox OS em

seus próprios dispositivos pré-existentes, sem a necessidade de aquisição de novos equipamentos. Esta frustração envolve diferentes questões, possuindo relação com restrições de portabilidade do *software* desenvolvido, o que é apresentado na literatura como um fator de influência negativa para sucesso dos projetos, e relação com fatores externos, uma vez que os *smartphones* presentes no mercado possuem restrições que impedem, na maioria das vezes, a instalação de um sistema operacional diferente do padrão de fábrica. Esta prática prejudicou não apenas o Firefox OS, mas a todos os sistemas operacionais para *smartphones* concorrentes da época, que não conseguiram se adentrar no mercado, que se estabilizou no duopólio da Google e da Apple.

As frustrações dos usuários e a redução do seu interesse no projeto foram identificadas ainda a partir do primeiro trimestre de 2013, mas não se observou mudanças na estratégia da Mozilla nesse período. Apenas a partir de 2015, após o Firefox OS já ter perdido grande parte do interesse do público, que foram observadas providências e, ainda assim, não agindo sobre as frustrações observadas. Este atraso e aparente desalinhamento com os interesses dos usuários podem representar uma ineficácia na detecção do *feedback* do público, o que demonstra ter contribuído para o projeto ser descontinuado.

7 CONSIDERAÇÕES FINAIS

Desenvolver projetos de *software* livre bem-sucedidos é uma tarefa complexa, mas que pode ser facilitada por meio da identificação de fatores motivadores do sucesso, a partir de estudos científicos. O tema foi explorado na literatura de tal forma que foi possível construir uma lista de fatores relevantes para que gestores de *software* livre possam se atentar a essas características no desenvolvimento dos seus projetos. Contudo, ainda há espaço para novos estudos, além da necessidade de se rever a contribuição de alguns dos fatores, pois a maioria dos trabalhos encontrados utilizaram dados de projetos desenvolvidos antes de 2010.

Os artigos pesquisados utilizaram, em sua maior parte, dados de uma plataforma que já não é mais amplamente utilizada pelos projetos de *software* livre. Por isso, métricas utilizadas nos estudos não mais podem ser facilmente obtidas, como número de *downloads*, coparticipação de autores em projetos, número de acessos na página, categorias de *software*, entre outras, dificultando a realização de novas pesquisas na área. Para contornar essa limitação, o presente trabalho utilizou de metodologias alternativas para a obtenção dos dados, incluindo a mineração de dados em repositórios Git e comentários de redes sociais, que se demonstraram eficazes e suficientes para a realização da análise, permitindo sua utilização em novas pesquisas.

Nesse sentido, observa-se que a velocidade de transformação nas áreas tecnológicas é alta o suficiente para impactar as pesquisas científicas relacionadas, sendo necessário o contorno de obstáculos advindo das mudanças frequentes, o que se torna um fator de dificuldade no estudo científico de fenômenos relacionados à tecnologia. Desse modo, tais fenômenos tendem a se desenvolver na sociedade, enquanto o conhecimento sobre os mesmos tende a se dar com algum atraso.

Este trabalho analisou 26 artigos sobre fatores motivadores do sucesso de projetos de *software* livre, publicados entre 2006 e 2021. Dentre as principais descobertas, identificou-se uma queda no número de publicações após um pico em 2012 – 2013, possivelmente devido a mudanças na plataforma preferida para hospedar projetos. Nesse sentido, o trabalho discutiu o arquivo de dados de

pesquisa do SourceForge, um projeto que forneceu aos pesquisadores um banco de dados temporal de dados do SourceForge, que já foi um dos repositórios mais importantes de *software* de código aberto do mundo. Porém, a plataforma perdeu sua relevância como fonte de dados após ser ultrapassado em popularidade pelo GitHub. Com a análise dos dados, foi possível constatar que a falta de uma fonte alternativa de dados limita o escopo de pesquisas sobre o sucesso de projetos de *software* de código aberto.

Foram identificados como principais fatores motivadores do sucesso em projetos de *software* livre: uma rede eficiente de desenvolvedores, usuários e projetos, uma eficiente execução de tarefas de desenvolvimento, a escolha de um tipo de licença atrativa, uma ampla portabilidade do *software* para diferentes plataformas, o engajamento da comunidade, a realização de traduções para diferentes idiomas, a utilização de linguagem de programação popular, a diversificação dos papéis de desenvolvimento, e a escolha de um nicho de mercado apropriado.

Outra etapa do trabalho foi estudar a descontinuidade do projeto Firefox OS, sob a perspectiva dos fatores identificados como motivadores do sucesso, e estabelecendo um comparativo com o projeto Android. Para isso, foram utilizados recursos de pesquisa bibliográfica e documental, além de dados dos repositórios Git e comentários da rede social Reddit. O projeto estudado apresentou uma boa construção de um conceito, desenvolvendo uma base para o *software*, que acompanhou as melhores práticas identificadas a partir da literatura. No prosseguimento do projeto estudado, houve uma elevada atividade de desenvolvimento e captação de desenvolvedores, o que é conhecido como um dos fatores importantes para o sucesso de projetos de *software* livre, mas houve uma baixa recepção do produto pelos usuários, que se afastaram progressivamente do projeto.

Este afastamento foi identificado como reação às insatisfações dos usuários com o produto apresentado, sob as principais alegações de que o *hardware* oferecido não era de boa qualidade, assim como este não era uma plataforma aberta, contrastando com o *software* desenvolvido. Outra reclamação recorrente se dava pelo fato de que não era possível instalar o Firefox OS em dispositivos já

existentes no mercado.

Com base nas insatisfações dos usuários observa-se que o *software* desenvolvido não possuía alta portabilidade, um fator abordado em alguns dos estudos e de relevância no caso estudado. Sem a possibilidade de portabilidade, os usuários dependiam da aquisição de um novo dispositivo, cujo *hardware* não foi bem recebido pelos mesmos.

Essa característica de dependência de uma nova plataforma de *hardware* para execução do *software* se demonstrou um fator relevante na baixa recepção do Firefox OS pelo público, trazendo a hipótese de que este fator pode ser também relevante para projetos de *software* livre, de forma geral, como uma sugestão para trabalhos futuros.

Contudo, o fator decisivo para a baixa adoção do Firefox OS, quando comparado com o Android, se apresentou como uma falha em entregar um produto que atendesse às demandas dos usuários e que criasse utilidade para os mesmos. Não foram identificadas alterações na estratégia da Mozilla para o sistema nos meses seguintes à apresentação do protótipo, ocorrida em fevereiro de 2012, nem após o lançamento do dispositivo no mercado, ocorrida em julho de 2013, sugerindo um possível desatentamento dos gestores para as demandas dos seus usuários.

Devido ao uso de comentários na rede social Reddit, este trabalho apresenta algumas limitações decorrentes da proporção demográfica dessa rede. O *site* não divulga dados oficiais de demografia, mas estimativas realizadas por terceiros mostram uma predominância de usuários residentes nos Estados Unidos da América (STATISTA, 2021), de tal forma que as opiniões e sentimentos expressados na rede social podem não representar adequadamente o público-alvo almejado pelo projeto. Entretanto, este fato não elimina a relevância dos dados, uma vez que há, entre os países mais participativos na rede, alguns onde o Firefox OS foi lançado, além de, apesar de não representar o principal público-alvo, os Estados Unidos também foram um dos países onde o produto foi lançado e divulgado. É relevante ainda destacar que o Reddit é composto por comunidades distintas, cuja demografia pode variar para cada uma, e não há dados demográficos sobre as comunidades específicas nas quais o produto foi comentado.

Por fim, o estudo de caso do projeto Firefox OS se demonstrou útil para obter

novas informações sobre sua descontinuidade, ampliando a base de conhecimento sobre o sucesso de projetos de *software* livre, além de permitir a validação dos fatores já estudados na literatura. A execução deste trabalho permitiu ponderar que, até mesmo projetos livres, de natureza colaborativa, com comunidades de voluntários e participação aberta a todos, podem falhar em observar as experiências, expectativas e retornos do público, levando a divergências entre desenvolvedores e usuários. Essas divergências mostram que, apesar da transparência e abertura geralmente presente no desenvolvimento de *software* livre, há possíveis dificultadores na relação e comunicação entre as diferentes partes interessadas.

8 ENTREGÁVEIS DE ACORDO COM OS PRODUTOS DO TCC

Junto deste projeto, foram entregues, a matriz FOFA (SWOT), apresentada no Anexo A, um artigo intitulado “indicadores bibliométricos sobre o tema “inovação e software livre” na base de dados de artigos científicos web of science” submetido a uma revista com Qualis B3 na área do PROFNIT e um *software*¹⁴ entregue como produto.

¹⁴Disponível em: <https://github.com/vitorsgoncalves/tcc-profnit>

REFERÊNCIAS

AGARWAL, Nitin; RATHOD, Urvashi. Defining 'success' for software projects: An exploratory revelation. **International Journal of Project Management**, [S. l.], v. 24, n. 4, p. 358–370, 2006. DOI: 10.1016/j.ijproman.2005.11.009.

ANDERSSON, Martin; KUSETOGULLARI, Anna; WERNBERG, Joakim. Software development and innovation: Exploring the software shift in innovation in Swedish firms. **Technological Forecasting and Social Change**, [S. l.], v. 167, p. 120695, 2021. DOI: 10.1016/j.techfore.2021.120695.

ANDROID OPEN SOURCE PROJECT. **Como fazer o download da origem**. 2022. Disponível em: <https://source.android.com/docs/setup/download/downloading?hl=pt-br>. Acesso em: 20 nov. 2022.

ANTWERP, Matthew Van; MADEY, Greg. Advances in the SourceForge Research Data Archive. *Em*: 2008, Milão. **Anais [...]**. *Em*: FOURTH INTERNATIONAL CONFERENCE ON OPEN SOURCE SYSTEMS. Milão Disponível em: <https://flosshub.org/sites/flosshub.org/files/srda2008.pdf>.

ATKINSON, Roger. Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria. **International Journal of Project Management**, [S. l.], v. 17, n. 6, p. 337–342, 1999. DOI: 10.1016/S0263-7863(98)00069-6.

AYYOUBZADEH, Seyed Mohammad; AYYOUBZADEH, Seyed Mehdi; ZAHEDI, Hoda; AHMADI, Mahnaz; KALHORI, Sharareh R. Niakan. Predicting COVID-19 Incidence Through Analysis of Google Trends Data in Iran: Data Mining and Deep Learning Pilot Study. **JMIR Public Health and Surveillance**, [S. l.], v. 6, n. 2, p. e18828, 2020. DOI: 10.2196/18828.

AZEVEDO, Lenno. **Campus Party Refice 2014 – 2º dia**. 2014. Disponível em: <https://mozillabr.org/2014/07/25/campus-party-refice-2014-2a-dia>. Acesso em: 16 fev. 2023.

BALTER, Ben. **Open source license usage on GitHub.com**. **The GitHub Blog**, 2015. Disponível em: <https://github.blog/2015-03-09-open-source-license-usage-on-github-com/>. Acesso em: 25 out. 2022.

BARROS, Thiago. **Novas imagens mostram como ficará o Firefox OS**. 2012. Disponível em: <https://www.techtudo.com.br/noticias/2012/07/novas-imagens-mostram-como-ficara-o-firefox-os.ghtml>. Acesso em: 24 nov. 2022.

BERKHOLZ, Donnie. **GitHub language trends and the fragmenting landscape**. 2014. Disponível em: <https://redmonk.com/dberkholz/2014/05/02/github-language-trends-and-the-fragmenting-landscape/>. Acesso em: 13 dez. 2022.

BHATT, Punita; AHMAD, Ali J.; ROOMI, Muhammad Azam. Social innovation with open source software: User engagement and development challenges in India. **Technovation**, [S. l.], v. 52–53, p. 28–39, 2016. DOI: 10.1016/j.technovation.2016.01.004.

BLIND, Knut; BÖHM, Mirko; GRZEGORZEWSKA, Paula; KATZ, Andrew; MUTO, Sachiko; PÄTSCH, Sivan; SCHUBERT, Torben. The impact of Open Source Software and Hardware

on technological independence, competitiveness and innovation in the EU economy. **Final Study Report. European Commission, Brussels, doi, [S. l.], v. 10, p. 430161, 2021.**

BOLLIER, David; HELFRICH, Silke (ORG.). **The wealth of the commons: a world beyond market and state.** Amherst, Mass: Levellers Press, 2012.

BONACCORSI, Andrea; ROSSI, Cristina. Why Open Source software can succeed. **Research Policy**, Open Source Software Development. [S. l.], v. 32, n. 7, Open Source Software Development, p. 1243–1258, 2003. DOI: 10.1016/S0048-7333(03)00051-9.

BRIAN4LLP. **Reddit Comments.** (Pushshift, Org.), 2014. Disponível em: <https://files.pushshift.io/reddit/comments/>. Acesso em: 1 nov. 2022.

BRIANPEIRIS. **Reddit Comments.** (Pushshift, Org.), 2015. Disponível em: <https://files.pushshift.io/reddit/comments/>. Acesso em: 1 nov. 2022.

CAPILUPPI, A.; LAGO, P.; MORISIO, M. Characteristics of open source projects. *Em: PROCEEDINGS OF THE SEVENTH EUROPEAN CONFERENCE ON SOFTWARE MAINTENANCE AND REENGINEERING 2003*, Benevento, Italy. **Anais [...]**. Benevento, Italy: IEEE Comput. Soc, 2003. p. 317–327. DOI: 10.1109/CSMR.2003.1192440. Disponível em: <http://ieeexplore.ieee.org/document/1192440/>. Acesso em: 13 dez. 2022.

CARNEIRO, Herman Anthony; MYLONAKIS, Eleftherios. Google Trends: A Web-Based Tool for Real-Time Surveillance of Disease Outbreaks. **Clinical Infectious Diseases**, [S. l.], v. 49, n. 10, p. 1557–1564, 2009. DOI: 10.1086/630200.

CHENGALUR-SMITH, Indushobha; SIDOROVA, Anna; DANIEL, Sherae. Sustainability of Free/Libre Open Source Projects: A Longitudinal Study. **Journal of the Association for Information Systems**, [S. l.], v. 11, n. 11, 2010. DOI: 10.17705/1jais.00244. Disponível em: <https://aisel.aisnet.org/jais/vol11/iss11/5>.

CLARK, Libby. **Enterprise Adoption of Open Source Practices is On the Rise.** **Linux.com**, 2014. Disponível em: <https://www.linux.com/news/enterprise-adoption-open-source-practices-rise/>. Acesso em: 1 jun. 2022.

COELHO, Jailton; VALENTE, Marco Tulio. Why modern open source projects fail. *Em: PROCEEDINGS OF THE 2017 11TH JOINT MEETING ON FOUNDATIONS OF SOFTWARE ENGINEERING 2017*, New York, USA. **Anais [...]**. New York, USA: Association for Computing Machinery, 2017. p. 186–196. DOI: 10.1145/3106237.3106246. Disponível em: <https://doi.org/10.1145/3106237.3106246>. Acesso em: 27 mar. 2022.

COMPETITION COMMISSION OF INDIA. **CCI imposes a monetary penalty of Rs. 1337.76 crore on Google for anti-competitive practices in relation to Android mobile devices.** , 2022. Disponível em: <https://www.cci.gov.in/antitrust/press-release/details/261/0>. Acesso em: 30 nov. 2022.

CROWSTON, Kevin; ANNABI, Hala; HOWISON, James. DEFINING OPEN SOURCE SOFTWARE PROJECT SUCCESS: International Conference on Information Systems, ICIS 2003. *Em: 2003, Anais [...]*. [s.l.: s.n.] p. 327–340. Disponível em: <http://www.scopus.com/inward/record.url?scp=85134942598&partnerID=8YFLogxK>. Acesso em: 22 out. 2022.

CUNNINGHAM, Andrew. **Firefox OS spreads from phones to TVs**. 2015. Disponível em: <https://arstechnica.com/gadgets/2015/01/firefox-os-spreads-from-phones-to-tvs/>. Acesso em: 30 nov. 2022.

DANIEL, Sherae; AGARWAL, Ritu; STEWART, Katherine J. The Effects of Diversity in Global, Distributed Collectives: A Study of Open Source Project Success. **Information Systems Research**, [S. l.], v. 24, n. 2, p. 312–333, 2013. DOI: 10.1287/isre.1120.0435.

DANIEL, Sherae; MIDHA, Vishal; BHATTACHERJEE, Anol; SINGH, Shivendu Pratap. Sourcing knowledge in open source software projects: The impacts of internal and external social capital on project success. **The Journal of Strategic Information Systems**, [S. l.], v. 27, n. 3, p. 237–256, 2018. DOI: 10.1016/j.jsis.2018.04.002.

DANIEL, Sherae; STEWART, Katherine. Open source project success: Resource access, flow, and integration. **The Journal of Strategic Information Systems**, [S. l.], v. 25, n. 3, p. 159–176, 2016. DOI: 10.1016/j.jsis.2016.02.006.

DEFREEZ, Daniel; SHASTRY, Bhargava; CHEN, Hao; SEIFERT, Jean-Pierre. **A First Look at Firefox OS Security**. arXiv, , 2014. DOI: 10.48550/arXiv.1410.7754. Disponível em: <http://arxiv.org/abs/1410.7754>. Acesso em: 30 nov. 2022.

DELONE, William; MCLEAN, Ephraim. The DeLone and McLean Model of Information Systems Success: A Ten-Year Update. **J. of Management Information Systems**, [S. l.], v. 19, p. 9–30, 2003. DOI: 10.1080/07421222.2003.11045748.

DÍAZ, Fernando; HENRÍQUEZ, Pablo A. Social sentiment segregation: Evidence from Twitter and Google Trends in Chile during the COVID-19 dynamic quarantine strategy. **PLOS ONE**, [S. l.], v. 16, n. 7, p. e0254638, 2021. DOI: 10.1371/journal.pone.0254638.

DONNELLY, Caroline. **Firefox OS for smartphones launched by Mozilla**. 2013. Disponível em: <https://www.itpro.com/operating-systems/19288/firefox-os-smartphones-launched-mozilla>. Acesso em: 25 nov. 2022.

DREILING, A.; KLAUS, H.; ROSEMAN, M.; WYSSUSEK, B. Open Source Enterprise Systems: Towards a Viable Alternative. *Em: PROCEEDINGS OF THE 38TH ANNUAL HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES 2005*, **Anais [...]**. *Em: PROCEEDINGS OF THE 38TH ANNUAL HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES*. [s.l.: s.n.] p. 227b–227b. DOI: 10.1109/HICSS.2005.473.

DULONG DE ROSNAY, Mélanie; STALDER, Felix. Digital commons. **Internet Policy Review**, [S. l.], v. 9, n. 4, p. 1–22, 2020. DOI: 10.14763/2020.4.1530.

EBERT, Christof. Open Source Drives Innovation. **IEEE Software**, [S. l.], v. 24, n. 3, p. 105–109, 2007. DOI: 10.1109/MS.2007.83.

EMPLOYEEENO5. **Reddit Comments**. (Pushshift, Org.), 2011. Disponível em: <https://files.pushshift.io/reddit/comments/>. Acesso em: 1 nov. 2022.

FERDINANDS, Gerbrich; SCHRAM, Raoul D.; BRUIN, Jonathan De; BAGHERI, Ayoub; OBERSKI, Daniel Leonard; TUMMERS, Lars; SCHOOT, Rens Van de. **Active learning for screening prioritization in systematic reviews - A simulation study**. OSF Preprints, , 2020. DOI: 10.31219/osf.io/w6qbg. Disponível em: <https://osf.io/w6qbg/>. Acesso em: 1 maio. 2023.

FERREIRA, Maria Carolina Zanini. Diretrizes de utilização de licenças de software livre e creative commons. *[S. l.]*, p. 185, 2020.

FINLEY, Klint. **GitHub Has Surpassed Sourceforge and Google Code in Popularity**. *ReadWrite*, 2011. Disponível em: <https://readwrite.com/github-has-passed-sourceforge/>. Acesso em: 24 out. 2022.

FOGEL, Karl. **Producing Open Source Software: How to Run a Successful Free Software Project**. [s.l.] : O'Reilly Media, Inc., 2005.

FRANCIS, Ben. **The Story of Firefox OS**. *Medium*, 2017. Disponível em: <https://medium.com/@bfrancis/the-story-of-firefox-os-cb5bf796e8fb>. Acesso em: 11 maio. 2022.

GADYATSKAYA, Olga; MASSACCI, Fabio; ZHAUNIAROVICH, Yury. Security in the Firefox OS and Tizen Mobile Platforms. *Computer*, *[S. l.]*, v. 47, n. 6, p. 57–63, 2014. DOI: 10.1109/MC.2014.165.

GHAPANCHI, Amir Hossein. Rallying competencies in virtual communities: A study of core processes and user interest in open source software projects. *Information and Organization*, *[S. l.]*, v. 23, n. 2, p. 129–148, 2013. DOI: 10.1016/j.infoandorg.2013.03.001.

GHAPANCHI, Amir Hossein. Predicting software future sustainability: A longitudinal perspective. *Information Systems*, *[S. l.]*, v. 49, p. 40–51, 2015. DOI: 10.1016/j.is.2014.10.005.

GHAPANCHI, Amir Hossein; AURUM, Aybuke. The impact of project capabilities on project performance: Case of open source software projects. *International Journal of Project Management*, *[S. l.]*, v. 30, n. 4, p. 407–417, 2012. a. DOI: 10.1016/j.ijproman.2011.10.002.

GHAPANCHI, Amir Hossein; AURUM, Aybuke. Competency rallying in electronic markets: implications for open source project success. *Electronic Markets*, *[S. l.]*, v. 22, n. 2, p. 117–127, 2012. b. DOI: 10.1007/s12525-012-0088-0.

GHAPANCHI, Amir Hossein; AURUM, Aybuke; LOW, Graham. A taxonomy for measuring the success of open source software projects. *First Monday*, *[S. l.]*, 2011. DOI: 10.5210/fm.v16i8.3558. Disponível em: <https://journals.uic.edu/ojs/index.php/fm/article/view/3558>. Acesso em: 16 out. 2022.

GHAPANCHI, Amir Hossein; TAVANA, Madjid. A Longitudinal Study of the Impact of Open Source Software Project Characteristics on Positive Outcomes. *Information Systems Management*, *[S. l.]*, v. 32, n. 4, p. 285–298, 2015. DOI: 10.1080/10580530.2015.1079999.

GHAPANCHI, Amir Hossein; WOHLIN, Claes; AURUM, Aybüke. Resources contributing to gaining competitive advantage for open source software projects: An application of resource-based theory. *International Journal of Project Management*, *[S. l.]*, v. 32, n. 1, p. 139–152, 2014. DOI: 10.1016/j.ijproman.2013.03.002.

GITHUB. **The largest open source community in the world**. 2022a. Disponível em: <https://github.com>. Acesso em: 24 out. 2022.

GITHUB. **Build software better, together**. 2022b. Disponível em: <https://github.com>. Acesso em: 24 out. 2022.

GOLDMAN, Ron; GABRIEL, Richard P. **Innovation Happens Elsewhere: Open Source as Business Strategy**. [s.l.] : Morgan Kaufmann, 2005.

GONZALEZ-BARAHONA, Jesus M.; ROBLES, Gregorio; HERRAIZ, Israel; ORTEGA, Felipe. Studying the laws of software evolution in a long-lived FLOSS project. **Journal of Software: Evolution and Process**, [S. l.], v. 26, n. 7, p. 589–612, 2014. DOI: 10.1002/smr.1615.

GOODWIN, Richard. **Mozilla's Boot 2 Gecko and why it could change the world - Features - Know Your Mobile**. 2012. Disponível em: https://web.archive.org/web/20120408135126/http://www.knowyourmobile.com/features/1267951/mozillas_boot_2_gecko_and_why_it_could_change_the_world.html. Acesso em: 24 nov. 2022.

GOOGLE. **Alterações de comportamento do Android 7.0 | Desenvolvedores Android**. 2020. Disponível em: <https://developer.android.com/about/versions/nougat/android-7.0-changes?hl=pt-br>. Acesso em: 8 mar. 2023.

GOOGLE. **FAQ about Google Trends data - Trends Help**. 2022. Disponível em: <https://support.google.com/trends/answer/4365533?hl=en>. Acesso em: 14 dez. 2022.

GREWAL, Rajdeep; LILIEN, Gary L.; MALLAPRAGADA, Girish. Location, Location, Location: How Network Embeddedness Affects Project Success in Open Source Systems. **Management Science**, [S. l.], v. 52, n. 7, p. 1043–1056, 2006. DOI: 10.1287/mnsc.1060.0550.

GRØNLI, Tor-Morten; HANSEN, Jarle; GHINEA, Gheorghita; YOUNAS, Muhammad. Mobile Application Platform Heterogeneity: Android vs Windows Phone vs iOS vs Firefox OS. *Em: 2014 IEEE 28TH INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION NETWORKING AND APPLICATIONS 2014*, **Anais [...]**. . *Em: 2014 IEEE 28TH INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION NETWORKING AND APPLICATIONS*. [s.l.: s.n.] p. 635–641. DOI: 10.1109/AINA.2014.78.

GRUSH, Andrew. **Tizen vs Firefox OS vs Ubuntu: Battle for Fourth Place | Mobile Magazine**. , 2013. Disponível em: <https://mobilemag.com/2013/02/27/tizen-firefox-os-ubuntu-competing/>. Acesso em: 25 nov. 2022.

HALL, Christine. **SourceForge Not Making A Graceful Exit**. 2015. Disponível em: <https://fossforce.com/2015/06/sourceforge-not-making-a-graceful-exit/>. Acesso em: 24 out. 2022.

HARS, Alexander; OU, Shaosong. Why is Open Source Software Viable? A Study of Intrinsic Motivation, Personal Needs and Future Returns. **AMCIS 2000 Proceedings**, [S. l.], 2000. Disponível em: <https://aisel.aisnet.org/amcis2000/379>.

HEATER, Brian. **Firefox OS heading to Brazil, Mexico, Poland, Spain and more; Huawei device coming soon**. 2013. Disponível em: <https://www.engadget.com/2013-02-24-firefox-os-global.html>. Acesso em: 25 nov. 2022.

HIGA, Paulo. **Firefox OS: não adianta ser barato se não for bom**. 2015. Disponível em: <https://tecnoblog.net/arquivo/178658/firefox-os-mudanca-estrategia/>. Acesso em: 30 nov. 2022.

HUTTO, C.; GILBERT, Eric. VADER: A Parsimonious Rule-Based Model for Sentiment

Analysis of Social Media Text. **Proceedings of the International AAI Conference on Web and Social Media**, [S. l.], v. 8, n. 1, p. 216–225, 2014. DOI: 10.1609/icwsm.v8i1.14550.

INTERNET ARCHIVE. **Internet Archive: Wayback Machine**. 2022. Disponível em: <https://archive.org/web/>. Acesso em: 14 dez. 2022.

JANCZUKOWICZ, Ewa. **Firefox OS Overview**. [s.l.] : Télécom Bretagne, 2013. Disponível em: <https://hal.archives-ouvertes.fr/hal-00961321>. Acesso em: 11 maio. 2022.

JENKINS, Jason. **Ubuntu Touch beats Firefox OS to win best of MWC from CNET**. 2013. Disponível em: <https://www.cnet.com/tech/mobile/ubuntu-touch-beats-firefox-os-to-win-best-of-mwc-from-cnet/>. Acesso em: 25 nov. 2022.

JIAU, Hewijin Christine; KAO, Chia Hung. Assessing the efficacy of user and developer activities in facilitating the development of OSS projects. **Journal of Software Maintenance and Evolution: Research and Practice**, [S. l.], v. 21, n. 5, p. 287–314, 2009. DOI: 10.1002/smr.409.

JUN, Seung-Pyo; YOO, Hyoung Sun; CHOI, San. Ten years of research change using Google Trends: From the perspective of big data utilizations and applications. **Technological Forecasting and Social Change**, [S. l.], v. 130, p. 69–87, 2018. DOI: 10.1016/j.techfore.2017.11.009.

KECHAOU, Zied; BEN AMMAR, Mohamed; ALIM, Adel. M. Improving e-learning with sentiment analysis of users' opinions. *Em: 2011 IEEE GLOBAL ENGINEERING EDUCATION CONFERENCE (EDUCON) 2011, Anais [...]. . Em: 2011 IEEE GLOBAL ENGINEERING EDUCATION CONFERENCE (EDUCON)*. [s.l: s.n.] p. 1032–1038. DOI: 10.1109/EDUCON.2011.5773275.

LECLAIR, Haidee. **65% of companies are contributing to open source projects | Opensource.com**. 2016. Disponível em: <https://opensource.com/business/16/5/2016-future-open-source-survey>. Acesso em: 19 jun. 2022.

LIN, Fred. **Developing FirefoxOS**. [s.l: s.n.]

LOVERIDGE, Sam. **Panasonic 2015 Smart TV line to run on Firefox OS**. 2015. Disponível em: <https://www.trustedreviews.com/news/panasonic-2015-smart-tv-line-to-run-on-firefox-os-2917557>. Acesso em: 30 nov. 2022.

MACHADO, André. **MWC 2014: Firefox OS terá smartphone de US\$ 25**. 2014. Disponível em: <https://oglobo.globo.com/economia/mwc-2014-firefox-os-tera-smartphone-de-us-25-11691347>. Acesso em: 30 nov. 2022.

MAHMOOD, Nasir; HAFEEZ, Yaser; IQBAL, Khalid; HUSSAIN, Shariq; AQIB, Muhammad; JAMAL, Muhammad; SONG, Oh-Young. Mining software repository for cleaning bugs using data mining technique. **Computers, Materials & Continua**, [S. l.], v. 69, n. 1, p. 873–893, 2021.

MAJUMDER, Suvodeep; CHAKRABORTY, Joymallya; AGRAWAL, Amritanshu; MENZIES, Tim. **Communication and Code Dependency Effects on Software Code Quality: An Empirical Analysis of Herbsleb Hypothesis**. arXiv, , 2022. DOI: 10.48550/arXiv.1904.09954. Disponível em: <http://arxiv.org/abs/1904.09954>. Acesso em: 8 mar. 2023.

MCCANN, John. **Mozilla Firefox OS: 10 things you need to know**. 2012. Disponível em: <https://www.techradar.com/news/phone-and-communications/mobile-phones/mozilla-firefox-os-10-things-you-need-to-know-1088177>. Acesso em: 25 nov. 2022.

MCCASKILL, Steve. **MWC 2014: New Firefox OS Smartphones As Mozilla Makes Push**. 2014. Disponível em: <https://www.silicon.co.uk/workspace/firefox-os-flame-smartphones-139788>. Acesso em: 30 nov. 2022.

MCCRACKEN, Julie. **B2G OS and Gecko Announcement from Ari Jaaksi & David Bryant**. 2016. Disponível em: <https://groups.google.com/g/mozilla.dev.fxos/c/FoAwifahNPY/m/Lppm0VHVBAAJ>. Acesso em: 11 maio. 2022.

MÉNDEZ-DURÓN, Rebeca; GARCÍA, Clara E. Returns from social capital in open source software networks. **Journal of Evolutionary Economics**, [S. l.], v. 19, n. 2, p. 277–295, 2009. DOI: 10.1007/s00191-008-0125-5.

MERGESTAT. **SQL for the Software Development Lifecycle**. 2022. Disponível em: <https://www.mergestat.com/>. Acesso em: 14 dez. 2022.

MIDHA, Vishal; PALVIA, Prashant. Factors affecting the success of Open Source Software. **Journal of Systems and Software**, [S. l.], v. 85, n. 4, p. 895–905, 2012. DOI: 10.1016/j.jss.2011.11.010.

MIES, Ginny. **First Look at Mozilla's Web Platform for Phones: "Boot to Gecko" | PCWorld**. 2012. Disponível em: https://web.archive.org/web/20120402210600/http://www.pcworld.com/article/250879/first_look_at_mozillas_web_platform_for_phones_boot_to_gecko.html. Acesso em: 24 nov. 2022.

MILKSTEAKSONTHEHOUSE. **Reddit Comments**. (Pushshift, Org.), 2014. Disponível em: <https://files.pushshift.io/reddit/comments/>. Acesso em: 1 nov. 2022.

MOZILLA. **Mozilla Gains Global Support For a Firefox Mobile OS | The Mozilla Blog**. 2012. Disponível em: <https://blog.mozilla.org/en/mozilla/firefox-mobile-os/>. Acesso em: 24 nov. 2022.

MOZILLA. **Mozilla Announces Global Expansion for Firefox OS**. 2013. Disponível em: <https://blog.mozilla.org/press/2013/02/firefox-os-expansion>. Acesso em: 25 nov. 2022.

MOZILLA. **Firefox OS: Looking Ahead**. 2014a. Disponível em: <https://blog.mozilla.org/blogarchive/blog/2014/01/06/firefox-os-looking-ahead>. Acesso em: 30 nov. 2022.

MOZILLA. **Mozilla Showcases First Year of Success with Firefox OS at Mobile World Congress 2014**. 2014b. Disponível em: <https://blog.mozilla.org/blogarchive/blog/2014/02/18/mozilla-showcases-first-year-of-success-with-firefox-os-at-mobile-world-congress-2014>. Acesso em: 30 nov. 2022.

MOZILLA. **Firefox OS Expands to Higher-Performance Devices and Pushes the Boundaries of Entry-Level Smartphones**. 2014c. Disponível em: <https://blog.mozilla.org/press/2014/02/firefox-os-news-2>. Acesso em: 30 nov. 2022.

MOZILLA. **Firefox OS architecture - Mozilla | MDN**. 2014d. Disponível em: https://web.archive.org/web/20140304165533/https://developer.mozilla.org/en-US/Firefox_OS/Platform/Architecture. Acesso em: 11 maio. 2022.

MOZILLA. **Firefox OS Expands to Nearly 30 Countries | The Mozilla Blog**. 2014e. Disponível em: <https://blog.mozilla.org/en/mozilla/firefox-os-expands-to-nearly-30-countries/>. Acesso em: 30 nov. 2022.

MOZILLA. **MWC 2015: Mozilla anunciará novidades para o Firefox OS**. 2015a. Disponível em: <https://blog.mozilla.org/press-br/2015/02/24/mwc-2015-mozilla-anunciara-novidades-para-o-firefox-os>. Acesso em: 30 nov. 2022.

MOZILLA. **Firefox OS Pivot to Connected Devices**. 2015b. Disponível em: <https://blog.mozilla.org/press/2015/12/firefox-os-pivot-to-connected-devices>. Acesso em: 30 nov. 2022.

MOZILLA. **Mozilla-B2G**. [s.d.]. Disponível em: <https://github.com/mozilla-b2g>. Acesso em: 11 maio. 2022a.

MOZILLA. **Quem somos**. [s.d.]. Disponível em: <https://foundation.mozilla.org/pt-BR/who-we-are/>. Acesso em: 11 maio. 2022b.

MOZILLA DEVELOPER NETWORK. **Introdução ao Firefox OS - Mozilla | MDN**. 2015. Disponível em: https://web.archive.org/web/20150906065704/https://developer.mozilla.org/pt-BR/Firefox_OS/Introdu%C3%A7%C3%A3o. Acesso em: 24 nov. 2022.

NAST, Condé. **Hands On With Boot2Gecko, the Mobile OS Built Entirely on Web Standards**. 2012. Disponível em: <https://stag2.wired.com/2012/05/hands-on-with-boot2gecko-the-mobile-os-built-entirely-on-web-standards/>. Acesso em: 25 nov. 2022.

NEATH, Kyle. **Those are some big numbers. The GitHub Blog**, 2011. Disponível em: <https://github.blog/2011-04-20-those-are-some-big-numbers/>. Acesso em: 20 jun. 2022.

NĚMEC, Radek. Business intelligence system success assessment using DeLone & McLean model. [S. l.], 2011.

NERI, Federico; ALIPRANDI, Carlo; CAPECI, Federico; CUADROS, Montserrat; BY, Tomas. Sentiment Analysis on Social Media. *Em*: 2012 IEEE/ACM INTERNATIONAL CONFERENCE ON ADVANCES IN SOCIAL NETWORKS ANALYSIS AND MINING 2012, **Anais** [...]. . *Em*: 2012 IEEE/ACM INTERNATIONAL CONFERENCE ON ADVANCES IN SOCIAL NETWORKS ANALYSIS AND MINING. [s.l.: s.n.] p. 919–926. DOI: 10.1109/ASONAM.2012.164.

NLTK PROJECT. **NLTK :: Natural Language Toolkit**. 2022. Disponível em: <https://www.nltk.org/>. Acesso em: 14 dez. 2022.

NYMAN, Robert. **Announcing Boot to Gecko (B2G) – Booting to the Web – Mozilla Hacks - the Web developer blog**. 2011. Disponível em: <https://hacks.mozilla.org/2011/07/announcing-boot-to-gecko-b2g-booting-to-the-web>. Acesso em: 24 nov. 2022.

O'GRADY, Stephen. **What Black Duck Can Tell Us About GitHub, Language Fragmentation and More**. 2011. Disponível em:

<https://redmonk.com/sogrady/2011/06/02/blackduck-webinar/>. Acesso em: 24 out. 2022.

OPEN SOURCE INITIATIVE. **Frequently Answered Questions | Open Source Initiative**. [s.d.]. Disponível em: <https://opensource.org/faq#free-software>. Acesso em: 13 dez. 2022.

OVERLORDXENU. **Reddit Comments**. (Pushshift, Org.), 2011. Disponível em: <https://files.pushshift.io/reddit/comments/>. Acesso em: 1 nov. 2022.

PAGE, Matthew J. et al. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. **BMJ**, [S. l.], p. n71, 2021. DOI: 10.1136/bmj.n71.

PANDYA, Sharnil; MEHTA, Pooja. **A Review On Sentiment Analysis Methodologies, Practices And Applications**. [s.l.: s.n.].

PELAT, Camille; TURBELIN, Clément; BAR-HEN, Avner; FLAHAULT, Antoine; VALLERON, Alain-Jacques. More Diseases Tracked by Using Google Trends. **Emerging Infectious Diseases**, [S. l.], v. 15, n. 8, p. 1327–1328, 2009. DOI: 10.3201/eid1508.090299.

PENG, Gang; WAN, Yun; WOODLOCK, Peter. Network ties and the success of open source software development. **The Journal of Strategic Information Systems**, [S. l.], v. 22, n. 4, p. 269–281, 2013. DOI: 10.1016/j.jsis.2013.05.001.

PEROTTI, Elena. **Google's Antitrust Woes Around the World**. Rochester, NY, 2017. DOI: 10.2139/ssrn.3060298. Disponível em: <https://papers.ssrn.com/abstract=3060298>. Acesso em: 30 nov. 2022.

PUSHSHIFT (ORG.). **Reddit Comments**. , 2012. Disponível em: <https://files.pushshift.io/reddit/comments/>. Acesso em: 1 nov. 2022.

PUSHSHIFT (ORG.). **Reddit Comments**. , 2016. Disponível em: <https://files.pushshift.io/reddit/comments/>. Acesso em: 1 nov. 2022.

PUSHSHIFT. **Reddit Statistics**. 2022. Disponível em: <https://pushshift.io/>. Acesso em: 14 dez. 2022.

QUINN, James Brian; BARUCH, Jordan J.; ZIEN, Karen Anne. Software-Based Innovation. **The Journal of Product Innovation Management**, [S. l.], v. 3, n. 14, p. 229–230, 1997.

RAKOWSKI, Brian. **Introducing Pixel, our new phone made by Google**. 2016. Disponível em: <https://blog.google/products/pixel/introducing-pixel-our-new-phone-made-google/>. Acesso em: 8 mar. 2023.

RAYMOND, Eric S. A brief history of hackerdom. **DiBona, Ockman y Stone, Open Sources, y www.tuxedo.org/~esr/writings/cathedral-bazaar/hacker-history/(primera versión 1992)**, [S. l.], 1999. a.

RAYMOND, Eric S. **Open Sources: Voices from the Open Source Revolution**. Text.Article. 1999b. Disponível em: <https://www.oreilly.com/openbook/opensources/book/raymond.html>. Acesso em: 13 dez. 2022.

RAYMOND, Eric S. **The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary**. 1st edition ed. Beijing; Cambridge, Mass: O'Reilly Media, 2001.

RED HAT. **The State of Enterprise Open Source**. , 2021. Disponível em: <https://www.redhat.com/rhdc/managed-files/rh-enterprise-open-source-report-f27565-202101-en.pdf>.

RIVERA, Nicolás. **Probamos el Firefox OS de \$25: la democratización del smartphone**. 2014. Disponível em: <http://hipertextual.com/2014/02/firefox-os-25-dolares>. Acesso em: 30 nov. 2022.

ROBINSON, L. et al. Digital inequalities 2.0: legacy inequalities in the information age. **First Monday**, [S. l.], v. 25, n. 7, 2020. Disponível em: <https://ora.ox.ac.uk/objects/uuid:ac36d8d5-cd59-4871-ab30-7925b6714243>. Acesso em: 21 jun. 2022.

ROMERO-OLMO, Juan-Antonio. ANÁLISIS DE LA API DE FIREFOX OS.DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN PARA TERMINALES CON FIREFOX OS. [S. l.], 2015. Disponível em: <http://tauja.ujaen.es/jspui/handle/10953.1/4229>. Acesso em: 29 nov. 2022.

ROTER, George. **Firefox OS/Connected Devices Announcement**. 2016. Disponível em: <https://discourse.mozilla.org/t/firefox-os-connected-devices-announcement/6864>. Acesso em: 30 nov. 2022.

ROWINSKI, Dan. **Mozilla Unveils Firefox OS Smartphones At Mobile World Congress**. **ReadWrite**, 2013. Disponível em: <https://readwrite.com/mozillas-firefox-os-smartphones-unveiled-at-mobile-world-congress/>. Acesso em: 25 nov. 2022.

SANTOS, Carlos; KUK, George; KON, Fabio; PEARSON, John. The attraction of contributors in free and open source software projects. **The Journal of Strategic Information Systems**, [S. l.], v. 22, n. 1, p. 26–45, 2013. DOI: 10.1016/j.jsis.2012.07.004.

SAVOV, Vlad. **The \$25 smartphone is as limited as its price suggests**. 2014. Disponível em: <https://www.theverge.com/2014/2/26/5448892/the-25-dollar-smartphone-is-as-limited-as-it-sounds>. Acesso em: 22 fev. 2023.

SAWH, Michael. **Android and iOS rivals: Firefox OS, Ubuntu Touch, Tizen and Sailfish**. 2013. Disponível em: <https://www.t3.com/features/android-and-ios-alternatives>. Acesso em: 25 nov. 2022.

SELIANA, Neni; SUROSO, Arif Imam; YULIATI, Lilik Noor. EVALUATION OF E-LEARNING IMPLEMENTATION IN THE UNIVERSITY USING DELONE AND MCLEAN SUCCESS MODEL. **Jurnal Aplikasi Manajemen**, [S. l.], v. 18, n. 2, p. 345–352, 2020. DOI: 10.21776/ub.jam.2020.018.02.15.

SEN, Ravi; SINGH, Siddhartha S.; BORLE, Sharad. Open source software success: Measures and analysis. **Decision Support Systems**, [S. l.], v. 52, n. 2, p. 364–372, 2012. DOI: 10.1016/j.dss.2011.09.003.

SHANKLAND, Stephen. **Firefox OS review: Unspectacular Firefox OS is OK for low-end phones (hands-on)**. 2013. Disponível em: <https://www.cnet.com/reviews/firefox-os-preview/>. Acesso em: 25 nov. 2022.

SHANKLAND, Stephen. **With Firefox OS, Mozilla begins the \$25 smartphone push**. 2014. Disponível em: <https://www.cnet.com/tech/mobile/with-firefox-os-mozilla-begins-the-25-smartphone-push/>. Acesso em: 30 nov. 2022.

SHANKLAND, Stephen. **Mozilla overhauls Firefox smartphone plan to focus on quality, not cost.** 2015. Disponível em: <https://www.cnet.com/tech/mobile/mozilla-overhauls-firefox-smartphone-plan-to-focus-on-quality-not-cost/>. Acesso em: 30 nov. 2022.

SHANKLAND, Stephen. **Firefox fail: Layoffs kill Mozilla's push beyond the browser.** [s.d.]. Disponível em: <https://www.cnet.com/tech/tech-industry/mozilla-layoff-firefox-device-relevance/>. Acesso em: 11 maio. 2022.

SHARMA, Dipti; SABHARWAL, Munish; GOYAL, Vinay; VIJ, Mohit. Sentiment Analysis Techniques for Social Media Data: A Review. *Em: (Ashish Kumar Luhach, Janos Arpad Kosa, Ramesh Chandra Poonia, Xiao-Zhi Gao, Dharm Singh, Org.)FIRST INTERNATIONAL CONFERENCE ON SUSTAINABLE TECHNOLOGIES FOR COMPUTATIONAL INTELLIGENCE 2020, Singapore. Anais [...].* Singapore: Springer, 2020. p. 75–90. DOI: 10.1007/978-981-15-0029-9_7.

SILVA, Rafael. **Boot to Gecko vira Firefox OS e recebe apoio de fabricantes.** 2012. Disponível em: <https://tecnoblog.net/arquivo/106307/mozilla-firefox-os/>. Acesso em: 24 nov. 2022.

SINGH, Param Vir. The small-world effect: The influence of macro-level properties of developer collaboration networks on open-source project success. **ACM Transactions on Software Engineering and Methodology**, [S. l.], v. 20, n. 2, p. 1–27, 2010. DOI: 10.1145/1824760.1824763.

SLAYERBOY. **Reddit Comments.** (Pushshift, Org.), 2014. Disponível em: <https://files.pushshift.io/reddit/comments/>. Acesso em: 1 nov. 2022.

SONG, Jaeyoon; KIM, Changhee. What Is Needed for the Sustainable Success of OSS Projects: Efficiency Analysis of Commit Production Process via Git. **Sustainability**, [S. l.], v. 10, n. 9, p. 3001, 2018. DOI: 10.3390/su10093001.

SOURCEFORGE RESEARCH DATA ARCHIVE. **Open Source Software Research.** 2016. Disponível em: http://zerlot.cse.nd.edu/mediawiki/index.php/Main_Page. Acesso em: 24 out. 2022.

SOUSA, Luciano Cunha De; DIAS, Cleidson Nogueira; ALFINITO, Solange. Software livre como fator de inovação para o setor de TIC. **Revista Eletrônica de Sistemas de Informação**, [S. l.], v. 13, n. 3, 2014. DOI: 10.21529/RESI.2014.1303004. Disponível em: <http://www.periodicosibepes.org.br/index.php/reinfo/article/view/1631>. Acesso em: 2 dez. 2021.

SQTLWLGK. **Reddit Comments.** (Pushshift, Org.), 2014. Disponível em: <https://files.pushshift.io/reddit/comments/>. Acesso em: 1 nov. 2022.

SQUIRE, Megan. The Lives and Deaths of Open Source Code Forges. *Em: PROCEEDINGS OF THE 13TH INTERNATIONAL SYMPOSIUM ON OPEN COLLABORATION 2017, New York, NY, USA. Anais [...].* New York, NY, USA: Association for Computing Machinery, 2017. p. 1–8. DOI: 10.1145/3125433.3125468. Disponível em: <https://doi.org/10.1145/3125433.3125468>. Acesso em: 4 dez. 2022.

STALLMAN, Richard. Viewpoint Why “open source” misses the point of free software. **Communications of the ACM**, [S. l.], v. 52, n. 6, p. 31–33, 2009. DOI: 10.1145/1516046.1516058.

STALLMAN, Richard Matthew; GAY, Joshua. **Free Software, Free Society: Selected Essays of Richard M. Stallman**. Boston, MA: CreateSpace Independent Publishing Platform, 2009.

STATCOUNTER. **Mobile Operating System Market Share Worldwide**. 2022a. Disponível em: <https://gs.statcounter.com/os-market-share/mobile/worldwide>. Acesso em: 4 maio. 2022.

STATCOUNTER. **FAQ**. 2022b. Disponível em: <https://gs.statcounter.com/faq>. Acesso em: 14 dez. 2022.

STATISTA. **Reddit user worldwide 2020, by country**. 2021. Disponível em: <https://www.statista.com/forecasts/1174696/reddit-user-by-country>. Acesso em: 8 mar. 2023.

STATISTA. **Google Play Store: number of apps 2022**. 2022. Disponível em: <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>. Acesso em: 14 dez. 2022.

STEINMACHER, Igor; GRACIOTTO SILVA, Marco Aurélio; GEROSA, Marco Aurelio. Barriers Faced by Newcomers to Open Source Projects: A Systematic Review. *Em: 2014, Anais [...]*. *Em: IFIP ADVANCES IN INFORMATION AND COMMUNICATION TECHNOLOGY*. [s.l.: s.n.] DOI: 10.1007/978-3-642-55128-4_21.

STINE, Robert A. Sentiment Analysis. **Annual Review of Statistics and Its Application**, [S. l.], v. 6, n. 1, p. 287–308, 2019. DOI: 10.1146/annurev-statistics-030718-105242.

SUTANTO, Juliana; JIANG, Qiqi; TAN, Chuan-Hoo. The contingent role of interproject connectedness in cultivating open source software projects. **The Journal of Strategic Information Systems**, [S. l.], v. 30, n. 1, p. 101598, 2021. DOI: 10.1016/j.jsis.2020.101598.

SUTANTO, Juliana; KANKANHALLI, Atreyi; TAN, Bernard C. Y. Uncovering the relationship between OSS user support networks and OSS popularity. **Decision Support Systems**, [S. l.], v. 64, p. 142–151, 2014. DOI: 10.1016/j.dss.2014.05.014.

THE_GECKO. **Reddit Comments**. (Pushshift, Org.), 2010. Disponível em: <https://files.pushshift.io/reddit/comments/>. Acesso em: 1 nov. 2022.

TIDELIFT. **The 2019 Tidelift managed open source survey results**. 2019. Disponível em: <https://www.tidelift.com/subscription/managed-open-source-survey>. Acesso em: 26 out. 2022.

TRENHOLM, Richard. **Alcatel One Touch Fire review: Alcatel One Touch Fire: Hands-on with the first Firefox phone**. 2013. Disponível em: <https://www.cnet.com/reviews/alcatel-one-touch-fire-preview/>. Acesso em: 25 nov. 2022.

VENTURA, Magda Maria. O estudo de caso como modalidade de pesquisa. **Revista SoCERJ**, [S. l.], v. 20, n. 5, p. 383–386, 2007.

VEREZ, Anthony; HUGUES, Guillaume. **Security Model of Firefox OS**. [S. l.], , 2013. Disponível em: http://verez.net/docs/ffos_paper.pdf.

VOSEN, Simeon; SCHMIDT, Torsten. Forecasting private consumption: survey-based indicators vs. Google trends. **Journal of Forecasting**, [S. l.], v. 30, n. 6, p. 565–578, 2011. DOI: 10.1002/for.1213.

WANG, Jing. Survival factors for Free Open Source Software projects: A multi-stage perspective. **European Management Journal**, [S. l.], v. 30, n. 4, p. 352–371, 2012. DOI: 10.1016/j.emj.2012.03.001.

WANG, Jing; HU, Michael Y.; SHANKER, Murali. Human agency, social networks, and FOSS project success. **Journal of Business Research**, [S. l.], v. 65, n. 7, p. 977–984, 2012. DOI: 10.1016/j.jbusres.2011.04.014.

WANG, Lu; HUANG, Minxue; LIU, Maohong. How founders' social capital affects the success of open-source projects: A resource-based view of project teams. **Electronic Commerce Research and Applications**, [S. l.], v. 30, p. 51–61, 2018. DOI: 10.1016/j.elerap.2018.05.003.

WATTS, Duncan J.; STROGATZ, Steven H. Collective dynamics of 'small-world' networks. **Nature**, [S. l.], v. 393, n. 6684, p. 440–442, 1998. DOI: 10.1038/30918.

WEI, Kwang Su; LOONG, Alain Chong Yee; LEONG, Ying-Mei; OOI, Keng-Boon. Measuring ERP system success: a respecification of the Delone and McLean's IS success model. *Em: 2009, Anais [...]. Em: SYMPOSIUM ON PROGRESS IN INFORMATION & COMMUNICATION TECHNOLOGY. : Citeseer, 2009. p. 7–12.*

WEISSGERBER, Peter; POHL, Mathias; BURCH, Michael. Visual Data Mining in Software Archives to Detect How Developers Work Together. *Em: FOURTH INTERNATIONAL WORKSHOP ON MINING SOFTWARE REPOSITORIES 2007, Anais [...]. Em: FOURTH INTERNATIONAL WORKSHOP ON MINING SOFTWARE REPOSITORIES. [s.l: s.n.] p. 9–9.* DOI: 10.1109/MSR.2007.34.

WEN, Wen; FORMAN, Chris; GRAHAM, Stuart. Research Note —The Impact of Intellectual Property Rights Enforcement on Open Source Software Project Success. **Information Systems Research**, [S. l.], v. 24, p. 1131–1146, 2013. DOI: 10.1287/isre.2013.0479.

WESTHUIZEN, Danie; FITZGERALD, Edmond. Defining and measuring project success. [S. l.], 2005.

WIJNHOFEN, Fons; PLANT, Olivia. Sentiment analysis and Google trends data for predicting car sales. [S. l.], 2017.

WILLIAMS, Andrew. **Firefox OS**. 2013. Disponível em: <https://www.trustedreviews.com/reviews/firefox-os>. Acesso em: 25 nov. 2022.

WILLIAMS, Sam. **Free as in freedom: Richard Stallman's crusade for free software**. [s.l.] : Sebastopol, Calif. : Farnham : O'Reilly, 2002. Disponível em: <http://archive.org/details/freeasinfreedomr00will>. Acesso em: 6 nov. 2022.

WIRED STAFF. Mobile World Congress Trends: HP's First Android Slate and a Firefox OS. **Wired**, [S. l.], 2013. Disponível em: <https://www.wired.com/2013/02/mwc-hp-asus-lenovo-firefox-os/>. Acesso em: 25 nov. 2022.

WRAY, Barry A.; MATHIEU, Richard G.; TEETS, Jay M. Identifying how determinants impact security-based open source software project success using rule induction. **International Journal of Electronic Marketing and Retailing**, [S. l.], v. 2, n. 4, p. 352, 2009. DOI: 10.1504/IJEMR.2009.025249.

YAKUBU, Hanan; KWONG, C. K. Forecasting the importance of product attributes using online customer reviews and Google Trends. **Technological Forecasting and Social Change**, [S. l.], v. 171, p. 120983, 2021. DOI: 10.1016/j.techfore.2021.120983.

YANG, Li; LI, Ying; WANG, Jin; SHERRATT, R. Simon. Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning. **IEEE Access**, [S. l.], v. 8, p. 23522–23530, 2020. DOI: 10.1109/ACCESS.2020.2969854.

YANG, Xin; KULA, Raula; YOSHIDA, Norihiro; IIDA, Hajimu. Mining the Modern Code Review Repositories: A Dataset of People, Process and Product. *Em*: 2016, **Anais [...]**. [s.l: s.n.] DOI: 10.1145/2901739.2903504.

ZAPPONI, Carlo. **Github Language Stats**. 2022. Disponível em: https://madnight.github.io/github/#/pull_requests/2022/3. Acesso em: 6 dez. 2022.

ZIEGLER, Chris. **With Firefox OS, Mozilla gets a little dirty to clean the mobile web**. 2013. Disponível em: <https://www.theverge.com/2013/2/27/4031362/firefox-os-mozilla-gets-a-little-dirty-to-clean-the-mobile-web>. Acesso em: 25 nov. 2022.

ZTE CORPORATION. **ZTE Open – the World’s First Firefox OS Smartphone**. 2013. Disponível em: <http://www.zte.com.cn/content/zte-site/www-zte-com-cn/global/about/news/387841>. Acesso em: 25 nov. 2022.

APÊNDICE A – Matrix FOFA (SWOT)

	AJUDA	ATRAPALHA
INTERNA	<p>Forças:</p> <ul style="list-style-type: none"> • Produto desenvolvido apresenta utilidade para novas pesquisas; • Contribuição para o desenvolvimento científico da área estudada; • Os resultados da pesquisa podem auxiliar gestores de projetos de software livre a obterem maiores chances de sucesso em suas iniciativas. 	<p>Fraquezas:</p> <ul style="list-style-type: none"> • Devido a restrições de disponibilidade de dados, os dados utilizados podem não representar em totalidade o objeto de estudo; • Devido ao uso de dados disponíveis publicamente, é possível que parte das ações e intenções da organização responsável pelo objeto de estudo não tenham sido capturado.
EXTERNA	<p>Oportunidades:</p> <ul style="list-style-type: none"> • O estudo de caso realizado pode ser de interesse à organização responsável pelo desenvolvimento do objeto de estudo; • O produto desenvolvido pode ser do interesse de instituições de ensino pesquisa, assim como de organizações atuantes no mercado. 	<p>Ameaças:</p> <ul style="list-style-type: none"> • Considerando que as principais fontes de dados utilizados na pesquisa são controladas por organizações com interesses distintos, é possível que a reprodutibilidade do trabalho possa ser comprometida no futuro.

APÊNDICE B – *Scripts* utilizados na metodologia do trabalho

Analizador de repositório Git:

Este script analisa os repositórios Git contidos no diretório em que é executado

em seguida, retorna arquivos CSV contendo a quantidade de commits por mês, autores/quantidade de commits por autor e novos autores por mês

os resultados são gravados no próprio diretório de execução

OBS: É necessário obter um binário do software mergestat para a execução deste script

O mergestat pode ser obtido por meio do link:
<https://github.com/mergestat/mergestat-lite>

```
import os, subprocess
```

```
import csv
```

```
import sqlite3
```

```
from os import path
```

```
pasta = os.path.abspath(os.getcwd())
```

```
total_commits = 0
```

```
autores = []
```

```
commits_mes = []
```

```
autores_mes = []
```

```
meses_repetidos = {}
```

```
qt_repetida = []
```

```
def minerar(dirs):
```

```
global total_commits
global autores
global commits_mes
global autores_mes
global meses_repetidos
global qt_repetida
```

```
#obtem o total de commits
query = subprocess.check_output(['mergestat', 'select count(*) from commits',
"--repo="+ dirs, "-f", "single"])
total_commits += int(query)
```

```
#obtem uma lista de autores e commits por autor
query = subprocess.check_output(['mergestat', 'select author_name,
strftime(\'%Y-%m\',min(author_when)) as data ,count(author_name) as
author_commits from commits group by author_name ', "--repo="+ dirs, "-f", "json"])
```

```
tempdict = {}
exec( "temp = " + query.decode('utf-8').replace("null", ""null""), tempdict)
temp = tempdict['temp']
for dicio in temp:
    achou = False
    for dicio2 in autores:
        if dicio2["author_name"] == dicio["author_name"]:
            dicio2["author_commits"] += dicio["author_commits"]
            achou = True
            if dicio["data"] in meses_repetidos:
                meses_repetidos[dicio["data"]] += 1
            else:
                meses_repetidos[dicio["data"]] = 1
            break
    if not achou:
        autores.append(dicio)
```

```
# obtém os commits por mês
query = subprocess.check_output(['mergestat', 'select strftime(\'%Y-%m\',
author_when) as data, count(author_when) as commits from commits group by data
', "--repo="+ dirs, "-f", "json"])
```

```
exec( "temp = " + query.decode('utf-8').replace("null", ""null""), tempdict )
temp = tempdict['temp']
```

```
for dicio in temp:
    achou = False
    for dicio2 in commits_mes:
        if dicio2["data"] == dicio["data"]:
            dicio2["commits"] += dicio["commits"]
            achou = True
            break
    if not achou:
        commits_mes.append(dicio)
```

```
#obtem os autores por mes
query = subprocess.check_output(['mergestat', 'select count(author_name) as
autores, strftime(\'%Y-%m\', data2) as data from (select distinct author_name,
min(author_when) as data2 from commits group by author_name ) group by data',
"--repo="+ dirs, "-f", "json"])
```

```
exec( "temp = " + query.decode('utf-8').replace("null", ""null""), tempdict)
temp = tempdict['temp']
```

```
for dicio in temp:
    achou = False
    for dicio2 in autores_mes:
        if dicio2["data"] == dicio["data"]:
            dicio2["autores"] += dicio["autores"]
```

```

        achou = True
        break
    if not achou:
        autores_mes.append(dicio)

for dirs in next(os.walk(pasta))[1]:
    print(dirs)

if path.exists(dirs + "/.git") and dirs != "external":
    minerar(dirs)
else:

    for subdirs in next(os.walk(dirs))[1]:
        if path.exists(dirs + "/.git"):
            minerar(dirs)

# remove os autores duplicados
for mes in autores_mes:
    if mes["data"] in meses_repetidos:
        mes["autores"] -= meses_repetidos[mes["data"]]

with open('totais.csv', 'w', encoding='UTF8', newline='') as f:
    writer = csv.writer(f)
    writer.writerow(['autores', 'commits'])
    writer.writerow([str(len(autores)), str(total_commits)])

with open('commits_mes.csv', 'w', encoding='UTF8', newline='') as f:
    writer = csv.DictWriter(f, fieldnames=['data', 'commits'])
    writer.writeheader()
    writer.writerows(commits_mes)

```

```
with open('autores_mes.csv', 'w', encoding='UTF8', newline='') as f:
    writer = csv.DictWriter(f, fieldnames=['data','autores'])
    writer.writeheader()
    writer.writerows(autores_mes)
```

```
with open('autores.csv', 'w', encoding='UTF8', newline='') as f:
    writer = csv.DictWriter(f, fieldnames=['author_name','author_commits',
'data'])
    writer.writeheader()
    writer.writerows(autores)
```

Pré-processamento dos dados obtidos a partir do Pushshift:

```
#!/bin/bash
```

#Este script preprocessa os dados obtidos a partir do PushShift, extraindo apenas as linhas que contenham as palavras-chave desejadas e gerando arquivos no formato CSV

#Para isso, devem ser passados, como parâmetro a palavra-chave desejada, o caminho onde se encontram os arquivos compactados, o caminho de saída para os CSV gerados, e, opcionalmente, um caminho temporário para realizar o processamento.

Utilização: ./preprocessa_dados.sh <palavra-chave> <diretório de origem dos dados> <diretório de saída para os arquivos CSV <diretório temporário opcional>>

#Se forem utilizadas mais de uma palavra-chave como entrada, utilize o formato "palavra1.*palavra2|palavra2.*palavra1", enviando o argumento entre aspas.

```

# Verifica se foram fornecidos os argumentos necessários
if [ $# -lt 3 ] || [ $# -gt 4 ]; then
    echo -e "Erro: argumentos inválidos.\n"
    echo -e "Utilização: ./preprocessa_dados.sh <palavra-chave> <diretório de
origem dos dados> <diretório de saída para os arquivos CSV> <diretório temporário
opcional> \n\nPara maiores informações, consultar a ajuda fornecida no cabeçalho
deste script \n"
    exit 1
fi

palavra_chave="$1"
caminho_entrada="$2"
caminho_saida="$3"
caminho_processamento="{4:-$caminho_entrada}"

for caminho_arquivo in "$caminho_entrada"/*.zst; do
    arquivo="$(basename "$caminho_arquivo" .zst)"

    if [ "$caminho_processamento" != "$caminho_entrada" ]; then
        cp "$caminho_arquivo" "$caminho_processamento/$arquivo.zst"
        caminho_arquivo="$caminho_processamento/$arquivo.zst"
    fi

    echo "'texto','sub','data','autor','id'" > "$caminho_saida/$arquivo.csv"

    zstd -dcf --long=31 "$caminho_arquivo" | grep -i -E $palavra_chave | \
        jq -r ' [.body, .subreddit, .created_utc, .author, .id] | @csv' >>
"$caminho_saida/$arquivo.csv"

    if [ "$caminho_processamento" != "$caminho_entrada" ]; then
        rm "$caminho_processamento/$arquivo.zst"
    fi
done

```

Análise de sentimento:

Este script realiza a análise de sentimento dos dados previamente processados e organizados em arquivos CSV, um arquivo para cada mês

A análise de sentimento é realizada utilizando a metodologia VADER, por meio da biblioteca NLTK

Como entrada, são passados os argumentos do caminho para os arquivos CSV e das palavras-chave desejadas

Como saída, é passado o diretório para gravar os resultados no formato CSV

Os arquivos de saída são constituídos de 1 arquivo CSV para cada arquivo de entrada e de um arquivo adicional, com a média dos resultados.

```
import csv
```

```
import datetime
```

```
import os
```

```
import re
```

```
import sys
```

```
import nltk
```

```
nltk.download('vader_lexicon')
```

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
```

```
sia = SentimentIntensityAnalyzer()
```

```
caminho_entrada = sys.argv[1]
```

```
caminho_saida = sys.argv[2]
```

```
palavras = sys.argv[3:]
```

```
historico = []
```

```

for files in os.listdir(caminho_entrada):
    if files.endswith(".csv"):
        ano = files[3:10]

        caminho = open(caminho_entrada + "/" + files, "r")
        arquivo = csv.DictReader(caminho)

        nbom = 0
        nruim = 0
        nneutro = 0
        soma_sentimento = 0

        with open(caminho_saida + "/" + files[:-4] + '_sentimento.csv', 'w',
encoding='UTF8', newline=") as f:
            writer = csv.DictWriter(f, arquivo.fieldnames + ["vader"])
            writer.writeheader()

            for row in arquivo:

                tex_original = row["texto"]

                a = tex_original.split("\n")
                b = []

                for trecho in a:
                    if any(palavra in trecho.lower() for palavra in palavras):
                        b += [trecho]

                if len("".join(a).strip("\n").strip(' ')) == 0:
                    continue
                else:
                    tex = '\n'.join(b)

                data = datetime.datetime.fromtimestamp(int(row["data"]))

```



```

ss = sia.polarity_scores(tex)
valor_vader = ss["compound"]

row.update({"vader": valor_vader})

#print(valor_vader)
if valor_vader > 0.05:
    nbom += 1

elif valor_vader < -0.05:
    nruim += 1

else:
    nneutro += 1

soma_sentimento += valor_vader

writer.writerow(row)

        media = soma_sentimento / (nneutro+nruim+nbom) if
(nneutro+nruim+nbom) else 0
        historico.append({'ano': ano, 'nbom': nbom, 'nneutro': nneutro, 'nruim':
nruim , 'media': media})
        print(historico)

with open(caminho_saida + '/sentimento_mes.csv', 'w', encoding='UTF8',
newline=") as f:
    writer = csv.DictWriter(f, fieldnames=['ano','nbom', 'nneutro', 'nruim',
'media' ])
    writer.writeheader()
    writer.writerows(historico)

```

APÊNDICE C – Código-fonte do produto desenvolvido

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import csv
import datetime
import os
import re
import sys

from argparse import ArgumentParser
from gooey import Gooey, GooeyParser

import nltk
nltk.download('vader_lexicon')

from nltk.sentiment.vader import SentimentIntensityAnalyzer
sia = SentimentIntensityAnalyzer()

@Gooey(
    program_name="Análise de sentimento",
    language="portuguese",
    program_description="Análise de sentimento utilizando a metodologia
Vader",
    image_dir=".",
    default_size=(610, 660),
)
def main():

    parser = GooeyParser()
```

```
required = parser.add_argument_group(
    'Parâmetros',
    goocy_options={
        'show_border': False,
        'columns': 2,
    }
)
```

```
required2 = parser.add_argument_group(
    "",
    goocy_options={
        'show_border': False,
        'columns': 1,
    }
)
```

```
required.add_argument(
    "-i",
    "--input",
    required=True,
    metavar="Entrada",
    help="Caminho onde se encontram os CSVs de entrada",
    widget="DirChooser"
)
```

```
required.add_argument(
    "-o",
    "--output",
    required=True,
    metavar="Saída",
    help="Caminho a serem gravados os CSVs resultantes",
    widget="DirChooser"
)
```

```
required2.add_argument(
```

```
    "-c",
    "--column",
    required=True,
    metavar="Nome da coluna",
    help="Nome da coluna com o texto a ser analisado",
)
required2.add_argument(
    "-k",
    "--keywords",
    required=False,
    metavar="Palavras-chave",
    help="Palavras-chave a serem consideradas na seleção dos trechos
(separadas por espaço)",
)
```

```
args = parser.parse_args()
```

```
caminho_entrada = args.input
caminho_saida = args.output
if args.keywords is not None:
    palavras = args.keywords.split(' ')
else: palavras = None
coluna = args.column
```

```
historico = []
```

```
for files in os.listdir(caminho_entrada):
    if files.endswith(".csv"):
        ano = files.replace(".csv", "")

        caminho = open(caminho_entrada + "/" + files, "r")
        arquivo = csv.DictReader(caminho)
```

```
nbom = 0
nruim = 0
nneutro = 0
soma_sentimento = 0
```

```
with open(caminho_saida + "/" + files[:-4] + '_sentimento.csv', 'w',
encoding='UTF8', newline=") as f:
```

```
writer = csv.DictWriter(f, arquivo.fieldnames + ["vader"])
writer.writeheader()
```

```
for row in arquivo:
```

```
    tex_original = row[coluna]
```

```
    a = tex_original.split('\n')
```

```
    if palavras is not None:
```

```
        b = []
```

```
        for trecho in a:
```

```
            if any(palavra in trecho.lower() for palavra in palavras):
```

```
                b += [trecho]
```

```
        else:
```

```
            b = a
```

```
    if len("".join(b).strip('\n').strip(' ')) == 0:
```

```
        continue
```

```
    else:
```

```
        tex = '\n'.join(b)
```

```
        ss = sia.polarity_scores(tex)
```

```
        valor_vader = ss["compound"]
```

```

row.update({"vader": valor_vader})

#print(valor_vader)
if valor_vader > 0.05:
    nbom += 1

elif valor_vader < -0.05:
    nruim += 1

else:
    nneutro += 1

soma_sentimento += valor_vader

writer.writerow(row)

        media = soma_sentimento / (nneutro+nruim+nbom) if
(nneutro+nruim+nbom) else 0
        historico.append({'ano': ano, 'nbom': nbom, 'nneutro': nneutro, 'nruim':
nruim , 'media': media})

        with open(caminho_saida + '/sentimento_mes.csv', 'w', encoding='UTF8',
newline=") as f:
            writer = csv.DictWriter(f, fieldnames=['ano','nbom', 'nneutro', 'nruim',
'media' ])
            writer.writeheader()
            writer.writerows(historico)

if __name__ == "__main__":
    main()

```

APÊNDICE D – Documentação do produto desenvolvido

Analizador de sentimento

Este software foi criado como produto da dissertação de mestrado de título "Fatores influenciadores do sucesso em projetos de software livre: um estudo de caso da descontinuação do Firefox OS" e entregue como produto tecnológico para o programa de mestrado PROFNIT. Sua idealização se deu a partir dos procedimentos metodológicos realizados no estudo, como uma forma de facilitar a realização de análises de sentimento por novos pesquisadores, principalmente de áreas de estudo não tecnológicas, de forma a reduzir as barreiras de facilitar a viabilidade de novos estudos.

Trata-se de uma ferramenta simplificada para realização de análise de sentimento a partir da metodologia VADER (<https://github.com/cjhutto/vaderSentiment>) para analisar a intensidade de sentimento de grandes quantidades de texto. Esta ferramenta foi projetada para uso em aplicações científicas, quando a análise de sentimento pode ser útil para avaliar a opinião dos participantes em um estudo ou para analisar comentários de usuários em fóruns ou redes sociais.

Funcionalidades

- Importação de dados no formato CSV: O software permite a leitura de dados no formato CSV, devido à sua praticidade e popularidade em aplicações científicas. O usuário pode optar entre importar um único arquivo CSV ou vários arquivos CSV para análise.
- Exportação de dados no formato CSV: O software exporta os dados analisados em um novo arquivo CSV, sem modificar o arquivo de origem, contendo a estrutura do arquivo original, adicionada de uma nova coluna, com o valor agregado de intensidade de sentimento calculada. É também

exportado um arquivo CSV auxiliar, contendo uma linha para cada CSV de entrada e a intensidade média dos textos de cada um deles, para facilitar a obtenção de médias nos casos em que os dados são organizados em múltiplos CSVs separados por períodos de tempo

- Seleção da coluna a ser analisada: O usuário pode escolher qual coluna do CSV será analisada, de forma que o software analisará apenas o texto contido na coluna selecionada, ignorando as demais.
- Filtragem por palavras-chave: O usuário pode escolher uma ou mais palavras-chave para filtrar os dados. Apenas as linhas que contêm pelo menos uma das palavras-chave selecionadas serão analisadas. Esta funcionalidade permite auxiliar a análise de dados com pouco ou nenhum pré-processamento.

Instalação

Se você utiliza sistemas operacionais Windows ou Linux em arquitetura X64, foram criados executáveis que podem ser baixados e executados diretamente, a partir do link abaixo, sem a necessidade de conhecimentos técnicos específicos:

<https://github.com/vitorsgoncalves/tcc-profnit/releases/tag/v0.1-alpha>

Na página acima, você deve baixar o arquivo identificado no nome com sua plataforma. Junto da do software, é fornecido um arquivo zip contendo alguns arquivos de exemplo, para facilitar a realização de testes.

Caso utilize um sistema diferente, ou o procedimento acima não funcione, siga as instruções abaixo:

1. Instale o Python em sua máquina, se não estiver disponível, a partir das instruções do site oficial:

<https://www.python.org/about/gettingstarted/>

2. Abra um terminal de linha de comando e baixe ou clone o repositório do software a partir do comando:

```
git clone https://github.com/vitorsgoncalves/tcc-profnit
```

3. Entre no diretório do software com o comando:

```
cd tcc-profnit/produto
```

4. Instale os pacotes Python necessários utilizando o comando:

```
pip install -r requirements.txt
```

5. Execute o software a partir do comando abaixo:

```
python analise_sentimento.py
```

Execução

Ao iniciar o software, será exibida uma tela como essa:

Figura 23 - Tela principal do *software*, exibindo as opções disponíveis

Análise de sentimento

Análise de sentimento
Análise de sentimento utilizando a metodologia Vader

Parâmetros

Entrada
Caminho onde se encontram os CSVs de entrada

Saída
Caminho a serem gravados os CSVs resultantes

Nome da coluna
Nome da coluna com o texto a ser analisado

Palavras-chave
Palavras-chave a serem consideradas na seleção dos trechos (separadas por espaço)

Fonte: Elaborado pelo autor, 2023

A tela principal consiste em dois campos de seleção de diretório, dois campos para inserção de parâmetros textuais, e os botões para iniciar e cancelar o processo.

No campo de “Entrada”, deve ser definido o diretório no qual se encontram os arquivos CSV a serem analisados. É importante destacar que todos os CSVs dentro desse diretório serão lidos, e que estes devem conter campos padronizados. Recomenda-se que este diretório contenha apenas os arquivos desejados para o

estudo.

No campo de “Saída”, deve ser definido o caminho para gravação dos arquivos CSV gerados pelo software. Recomenda-se que não seja o mesmo diretório de entrada, para evitar que estes sejam lidos como entrada ao se executar novamente o procedimento.

O campo “Nome da coluna” deve ser preenchido com o nome da coluna do arquivo CSV que contenha os dados textuais a serem analisados.

Já o campo “Palavras-chave” é opcional, e deve ser preenchido com as palavras-chave desejadas nos casos em que nem todas as linhas do arquivo CSV deverão ser analisadas. Ao se utilizar esse recurso, serão consideradas apenas as linhas que contenham pelo menos uma das palavras-chave inseridas e o arquivo de saída conterá apenas as linhas classificadas.

Um exemplo de preenchimento dos campos pode ser visto na figura abaixo:

Figura 24 - Imagem da tela principal com os dados preenchidos

Análise de sentimento

Configurações
Análise de sentimento utilizando a metodologia Vader

Parâmetros

Entrada
Caminho onde se encontram os CSVs de entrada
C:\Users\Usuario\Desktop\dados_

Saída
Caminho a serem gravados os CSVs resultantes
C:\Users\Usuario\Desktop

Nome da coluna
Nome da coluna com o texto a ser analisado
comment

Palavras-chave
Palavras-chave a serem consideradas na seleção dos trechos (separadas por espaço)

Fonte: Elaborado pelo autor, 2023

Após preencher os dados, pressione o botão "Iniciar e aguarde a execução em uma nova tela. Dependendo do tamanho dos arquivos de entrada, o processo pode levar muito tempo. Após o término, será exibido um diálogo indicando o fim da operação. Os resultados podem ser encontrados na pasta definida como saída. O arquivo gerado é uma cópia do arquivo original, com uma nova coluna referente ao valor calculado como intensidade de sentimento. Um exemplo de arquivos gerados

pode ser visto nas imagens abaixo:

Figura 25 - Imagem demonstrando um exemplo de saída de dados

	A	B	C
1	comment	author	vader
2	I really liked the software, it is light and practical and it helped me in my research work.	author1	0.4754
3	The software is interesting, but I think it still has a lot to be improved.	author2	0.7184
4	At first, it seemed very useful, but I couldn't insert it into my methodology, so It didn't help me.	author3	-0.2928
5	I just LOVED this software! So many hours of work saved in my research, I wish I had discovered this much sooner.	author4	0.8866
6	The functionality of the software itself is good, but the graphical interface leaves a lot to be desired. I didn't like it at all.	author5	0.2347
7			

Fonte: Elaborado pelo autor, 2023

Exemplo do segundo arquivo de saída, com os números de comentários de cada tipo e as médias calculadas para cada arquivo de entrada:

Figura 26 - Exemplo do segundo arquivo de saída, com os números de comentários de cada tipo e as médias calculadas para cada arquivo de entrada

	A	B	C	D	E
1	arquivo	nbom	nneutro	nruim	media
2	jan	1	2	2	0.044579999999999974
3	fev	4	0	1	0.40446
4	mar	4	1	0	0.353280000000000004

Fonte: Elaborado pelo autor, 2023

Limitações e observações

- O léxico utilizado pelo método VADER é baseado no idioma inglês e pode não gerar resultados aceitáveis para outros idiomas.
- A metodologia não é livre de falhas, e o valor calculado pode não corresponder adequadamente com o texto analisado em alguns casos.

Possibilidades futuras

- Adaptar a metodologia para outros idiomas.
- Permitir a inserção de palavras-chave com operadores lógicos.
- Permitir escolher a metodologia utilizada dentre outras opções populares.