

Sobre o Ensino de Software Livre na Graduação: Estratégias e Lições Aprendidas

Title: On Open-source Software Graduate Teaching: Strategies and Lessons Learned

Título: Sobre la Enseñanza de Software Libre en la Graduación: Estrategias y Lecciones Aprendidas

Paulo Afonso Parreira Júnior
Universidade Federal de Lavras (UFLA)
ORCID: [0000-0002-8877-2931](https://orcid.org/0000-0002-8877-2931)
pauloa.junior@ufla.br

Júlio César Alves
Universidade Federal de Lavras (UFLA)
ORCID: [0000-0002-4848-9453](https://orcid.org/0000-0002-4848-9453)
juliocesar.alves@ufla.br

André Pimenta Freire
Universidade Federal de Lavras (UFLA)
ORCID: [0000-0001-7894-9740](https://orcid.org/0000-0001-7894-9740)
apfreire@ufla.br

Heitor Augustus Xavier Costa
Universidade Federal de Lavras (UFLA)
ORCID: [0000-0002-9903-7414](https://orcid.org/0000-0002-9903-7414)
heitor@ufla.br

Igor Scaliante Wiese
Universidade Tecnológica Federal do Paraná (UTFPR) -
Campus Campo Mourão
ORCID: [0000-0001-9943-5570](https://orcid.org/0000-0001-9943-5570)
igor.wiese@gmail.com

Resumo

Uma das formas de se proporcionar o contato de estudantes de computação com o universo do software livre é por meio da oferta de disciplinas com esta temática. Tal contato permite que esses estudantes tenham, dentre outras coisas, a oportunidade de lidar com problemas reais, vivenciados diariamente na indústria de software. O presente artigo descreve um relato de experiência sobre duas ofertas da disciplina “Desenvolvimento de Software Livre” para os cursos de Ciência da Computação e Sistemas de Informação. Espera-se, por meio deste trabalho, poder compartilhar boas práticas, desafios e lições aprendidas sobre o ensino de software livre na graduação, de tal forma a desvendar “o caminho das pedras” e incentivar outros(as) docentes a se debruçarem sobre o tema. Como principais resultados das ofertas desta disciplina, a partir das respostas dadas em um questionário respondido por 37 estudantes, observou-se que 36 (97%) deles(as) recomendariam fortemente a disciplina a outras pessoas e que 31 (83%) se sentiram mais confiantes e mais preparados(as) para contribuir com projetos de software livre, após terem cursado a disciplina.

Palavras-chave: Educação em Computação, Software Livre, Relato de Experiência

Abstract

One way to provide computer science students with exposure to the world of open-source software is through the offering of courses on this topic. Such exposure allows these students to have, among other things, the opportunity to deal with real-world problems experienced daily in the software industry. This paper describes a case study of two offerings of the course “Open-source Software Development” for Computer Science and Information Systems courses. It is hoped that this work will be able to share good practices, challenges, and lessons learned about

teaching free software in undergraduate courses, in such a way as to reveal “the path of the stones” and encourage other teachers to focus on the subject. As the main results of the offerings of this course, based on the answers given in a questionnaire answered by 37 students, it was observed that 36 (97%) of them would strongly recommend the course to others and that 31 (83%) felt more confident and more prepared to contribute to free software projects after having taken the course.

Keywords: *Computing Education, Open-Source Software, Experience Report.*

Resumen

Una de las formas de proporcionar el contacto de los estudiantes de computación con el universo del software libre es mediante la oferta de asignaturas con esta temática. Este contacto permite que estos estudiantes tengan, entre otras cosas, la oportunidad de enfrentarse a problemas reales experimentados diariamente en la industria del software. El presente artículo describe un informe de experiencia sobre dos ofertas de la asignatura “Desarrollo de Software Libre” para los cursos de Ciencia de la Computación y Sistemas de Información. Se espera, a través de este trabajo, poder compartir buenas prácticas, desafíos y lecciones aprendidas sobre la enseñanza de software libre en la licenciatura, de manera que se pueda desentrañar “el camino de las piedras” e incentivar a otros docentes a abordar el tema. Como principales resultados de las ofertas de esta asignatura, a partir de las respuestas dadas en un cuestionario respondido por 37 estudiantes, se observó que 36 (97%) de ellos recomendarían fuertemente la asignatura a otras personas y que 31 (83%) se sintieron más seguros y preparados para contribuir con proyectos de software libre después de haber cursado la asignatura.

Palabras clave: *Educación Informática, Software Libre, Informe de Experiencia.*

1. Introdução

O conceito de *software livre* propõe, dentre outras coisas, um mecanismo de licenciamento por meio do qual o software pode ser utilizado, redistribuído e modificado (o que implica em acesso ao seu código-fonte) praticamente sem restrições (Kon et al., 2011; Pinto, 2021). Segundo Steinmacher et al. (2020), a ideia de que projetos de software livre pudessem ser bem-sucedidos nem sempre foi bem aceita ou compreendida, afinal, como projetos desenvolvidos de forma distribuída, com trabalho voluntário e sem uma estrutura formal de comando e controle poderiam progredir? No entanto, casos de sucesso, tais como o *kernel Linux*¹ e os diversos produtos da organização *Apache*², mostraram que esse modelo não só é viável, mas que, muitas vezes, pode gerar resultados até melhores do que os de outros projetos de software não-livres (Steinmacher et al., 2020).

Além de ser uma excelente oportunidade de aprendizado, a participação de estudantes de computação em projetos de software livre pode ser um diferencial em suas carreiras, uma vez que diversas empresas têm valorizado e buscado esse tipo de experiência nos currículos e portfólios de futuros(as) colaboradores(as) (Steinmacher et al., 2020). Por meio desse contato com o universo do software livre, esses(as) estudantes podem conhecer suas potencialidades e desafios, bem como aprender a lidar com problemas reais, vivenciados diariamente na indústria de software (Braught et al., 2018; Hollister & Leon, 2022; Nascimento et al., 2019; Zorzo et al., 2017). Um exemplo de iniciativa nessa linha é o programa *Google Summer of Code*³, que já financiou a participação de

¹<https://www.kernel.org/>

²<https://www.apache.org/>

³<https://summerofcode.withgoogle.com/>

estudantes e pesquisadores em mais de 680 projetos de software livre ao redor do mundo.

Outra forma de se proporcionar essa aproximação entre estudantes e projetos de software, no contexto universitário, é por meio da oferta de disciplinas sobre desenvolvimento de software livre. Apesar de ser relativamente fácil encontrar ementas, planos de ensino e até mesmo notas de aula sobre o assunto, publicamente disponíveis na Internet (Castor, 2012; Kon, 2017), nota-se a escassez de trabalhos na literatura com relatos de experiência sobre a oferta de disciplinas com essa temática. Esse é um ponto que merece a atenção da comunidade acadêmica, pois quantas boas práticas, desafios e experiências, em geral, poderiam ser aprendidas, debatidas e aproveitadas por outros(as) docentes ao redor do país, caso tais relatos fossem divulgados?

Diante dessa necessidade, Parreira Júnior et al. (2023) publicaram um relato de experiência sobre a primeira oferta da disciplina “Desenvolvimento de Software Livre” para os cursos de Ciência da Computação e Sistemas de Informação da Universidade Federal de Lavras. Segundo os autores, os resultados foram bastante promissores, dentre eles, não ter havido desistências durante a disciplina e o índice de aprovação ter sido de 100%, com todas as notas acima de 85 pontos. Apesar disso, alguns pontos para aprimoramento da disciplina foram levantados, tais como: (i) aprofundar o ensino dos conceitos de *Git* e *GitHub*; (ii) adicionar um novo tópico ao conteúdo programático da disciplina, a respeito do processo de construção de projetos de software livre; e (iii) aceitar, no trabalho prático da disciplina, contribuições de estudantes que pretendam criar seus próprios projetos de software livre.

Neste contexto, o presente trabalho procura complementar o relato anteriormente citado, apresentando como os pontos fracos observados foram atacados em uma segunda oferta da disciplina e quais resultados foram alcançados. Dos resultados obtidos, pode-se destacar: (i) a disciplina contou com 60 estudantes matriculados(as), em suas duas ofertas, ocupando 100% das vagas disponíveis, e todos(as) chegaram ao final do semestre, ou seja, não houve desistências; (ii) o percentual de aprovação foi de 100%, com quase todas as notas finais acima de 80 (oitenta) pontos; (iii) a partir das respostas dadas em um questionário respondido por 37 estudantes da disciplina, percebeu-se que 36 (97%) deles(as) recomendariam fortemente a disciplina a outras pessoas e 31 (83%) se sentiram mais confiantes e preparados(as) para contribuir com projetos de software livre, após terem cursado a disciplina; (iv) a contribuição de um dos estudantes da disciplina foi para uma biblioteca *JavaScript* com mais de 2.1 milhões de downloads semanais; (iv) houve relatos de estudantes que, após terem cursado a disciplina, contribuíram com bibliotecas de software livre para resolverem problemas em seus locais de trabalho; entre outros.

O restante deste artigo encontra-se organizado como segue: a Seção 2 destaca e discute os principais trabalhos relacionados a este relato de experiência; a Seção 3, por sua vez, descreve o planejamento e a execução das duas ofertas da disciplina “Desenvolvimento de Software Livre”, apontando os resultados obtidos e as lições aprendidas; a Seção 4 apresenta e discute os resultados das avaliações feitas com os(as) estudantes da disciplina; e, por fim, na Seção 5 estão as considerações finais e as propostas de trabalhos futuros.

2. Trabalhos Relacionados

Com o intuito de encontrar trabalhos relacionados ao relato aqui apresentado, inicialmente, realizou-se uma busca no portal SOL (*SBCOpenLib*)⁴, que concentra publicações de vários eventos e periódicos nacionais (alguns deles relacionados à Informática na Educação e Educação em Informática), utilizando a *string* de busca “*software livre*” AND (“*ensino*” OR “*aprendizagem*” OR “*disciplina*”). A busca foi realizada com os filtros “Anais de eventos” e “Periódicos” ativados e considerando “Qualquer lugar” do artigo. Também foi realizada uma busca no *Google Scholar*⁵, utilizando a *string* de busca “*open-source*” AND “*teaching*”. Diversos estudos foram retornados, os quais podem ser classificados, basicamente, em 2 (duas) categorias: (i) aqueles que desenvolvem ou usam software livre para o contexto de ensino-aprendizagem nas mais diversas áreas, inclusive a computação (Andrade et al., 2021; Camargo et al., 2019; Canez & Severo, 2016; de Oliveira et al., 2019; Ribeiro et al., 2017; Rossi & Koike, 2022); e (ii) aqueles que fazem uso do código-fonte de projetos de software livre no processo de ensino-aprendizagem de conceitos de computação, tais como “Engenharia de Software” (Tonhão et al., 2022; Xing, 2010), “*Design*” (Carrington & Kim, 2003), “*Teste*” (Deng et al., 2020) e “Evolução de Software” (Petrenko et al., 2007). Contudo, não foram encontrados estudos que tratavam especificamente do ensino de software livre.

A partir disso, realizou-se uma nova busca no portal SOL, desta vez, com enfoque em relatos de experiência sobre o ensino de disciplinas de graduação. Para isso, a seguinte *string* de busca foi utilizada: “*relato de experiência*” AND (“*graduação*” OR “*ensino superior*”). Dos resultados obtidos, grande parte deles eram relatos de experiência sobre a aplicação de diferentes tipos de metodologias de ensino, tais como sala de aula invertida e gamificação (Almeida et al., 2019; Diniz & Ferreira, 2018; Silva, 2018) no Ensino Superior, e sobre a utilização de ferramentas educacionais de apoio ao ensino-aprendizado de algumas disciplinas (Perin et al., 2021). Mais uma vez, não foram encontrados estudos que discorressem sobre a experiência de se ofertar disciplinas com a temática de desenvolvimento de software livre.

É importante ressaltar que um Mapeamento Sistemático seria necessário para tratar de forma exaustiva a literatura de relatos de experiência sobre o ensino de disciplinas de graduação. Contudo, a partir da revisão realizada neste trabalho, foi possível notar que há uma lacuna na literatura de ensino de computação, a respeito de relatos de experiência sobre o ensino de software livre, o que levanta um ponto de atenção para comunidade acadêmica. Apesar de ser fácil encontrar ementas, planos de curso e até mesmo notas de aulas publicamente disponibilizadas na Internet, apenas isto não é suficiente. Em um relato de experiência, boas práticas, desafios e lições aprendidas são compartilhadas, a fim de serem absorvidas e aproveitadas por outros(as) docentes. Tais relatos podem ser úteis ao mostrar “o caminho das pedras” e permitir que o(a) docente prepare melhor a disciplina, dedicando seus esforços para a ministração e discussão dos conteúdos com os(as) estudantes.

⁴<https://sol.sbc.org.br>

⁵<https://scholar.google.com.br>

3. Relato de Experiência

A disciplina, cujas ofertas são relatadas neste artigo, foi incluída na grade curricular dos cursos de Ciência da Computação e Sistemas de Informação da Universidade Federal de Lavras com o nome “Desenvolvimento de Software Livre”. Trata-se de uma disciplina eletiva, com carga horária de 68 horas-aula e um total de 30 vagas⁶. O restante desta seção apresenta o “Planejamento da Disciplina” (Subseção 3.1) e sua “Execução e Resultados” (Subseção 3.2), considerando as ofertas do primeiro e do segundo semestres letivo do ano 2022. Daqui adiante, por questão de simplicidade, essas ofertas serão denominadas “oferta 2022/1” e “oferta 2022/2”, respectivamente.

3.1. Planejamento da Disciplina

Conforme comentado na introdução, há boas razões para se ofertar uma disciplina que contemple a temática de desenvolvimento de software livre para os cursos de computação, dentre elas, os possíveis impactos positivos para o posicionamento do(a) profissional frente à indústria de desenvolvimento de software. Contudo, diante do propósito de ofertar uma disciplina nestes moldes, algumas dúvidas ocorreram, a saber: qual bibliografia utilizar? quais assuntos abordar? como avaliar o conhecimento obtido pelos(as) estudantes?

O primeiro passo tomado para responder a essas questões foi recorrer a informações disponibilizadas publicamente por outros(as) docentes que já ministraram disciplinas similares. Esse passo foi importante para a geração de uma primeira versão do plano de curso da disciplina, o qual foi bastante inspirado nas disciplinas oferecidas pelo Instituto de Matemática e Estatística da Universidade de São Paulo (IME/USP) (Kon, 2017) e pelo Centro de Informática da Universidade Federal de Pernambuco (CIn/UFPE) (Castor, 2012). É importante ressaltar que o IME/USP teve grande contribuição na história do movimento do software livre no Brasil, tendo sido, possivelmente, a primeira instituição de ensino brasileira a ter uma versão do Linux instalada em seus laboratórios de informática (Kon et al., 2011). As informações disponibilizadas publicamente sobre estas disciplinas permitiu, dentre outras coisas, obter uma referência bibliográfica básica e complementar, algumas ideias de tópicos a serem abordados, bem como alguns tipos de avaliações e projetos práticos passíveis de serem realizados.

A versão preliminar do plano de curso da disciplina foi disponibilizada para apreciação de dois docentes com ampla experiência no ensino de computação, sendo um deles docente de uma disciplina sobre desenvolvimento de software livre. A partir do *feedback* recebido, o plano de curso foi atualizado, tornando-se a versão utilizada durante a oferta 2022/1 da disciplina. Após a primeira oferta da disciplina, alguns tópicos sofreram modificações, a fim de atender a algumas reivindicações dos alunos da disciplina, conforme comentado na Seção 4. Os tópicos do conteúdo programático da disciplina, considerando cada oferta, são apresentados na Tabela 1 e o seu detalhamento é feito ao longo do texto desta subseção.

Antes de adentrar nas especificidades de cada tópico, é importante ressaltar que a disciplina em questão possuía 4 aulas de 50 minutos por semana, divididas em 2 dias, isto é, era uma disciplina de 4 créditos. Os tópicos anteriores aos projetos práticos ocupavam uma semana de aula, cada. Na primeira aula da semana era apresentado o conteúdo programático correspondente ao

⁶Os(as) estudantes podem se matricular na disciplina a partir do 4º Módulo (segundo ano) do curso.

Tabela 1: Conteúdo programático da disciplina “Desenvolvimento de Software Livre”.

#	Tópico	2022/1	2022/2
01	Leitura e discussão sobre o artigo “A Catedral e o Bazar”, de Eric Raymond	X	X
02	Apresentação e discussão sobre o documentário “Revolution OS”	X	-
03	Conceitos, história e modelos de negócio em Software Livre	X	X
04	Licenciamento de software	X	X
05	Palestra convidada	X	X
06	Versionamento de software com Git e Github - Parte 1 (conceitos de <i>clone</i> , <i>commit</i> , <i>push</i> , etc)	X	X
07	Versionamento de software com Git e Github - Parte 2 (conceitos de <i>fork</i> , <i>branch</i> , <i>pull request</i> , etc)	X	X
08	Diretrizes para mantenedores de projetos de Software Livre	-	X
09	Projeto 1: repositório sobre personalidades envolvidas com Software Livre	X	X
10	Projeto 2: contribuições para projetos de software livre reais	X	X

tópico e na segunda era realizada a atividade avaliativa sobre o conteúdo apresentado. A atividade avaliativa era realizada de forma remota, utilizando o Ambiente Virtual de Aprendizagem (AVA) da Instituição. Os tópicos correspondentes aos projetos práticos da disciplina ocupavam mais de uma semana de aula, conforme será explicado posteriormente neste texto. Outro ponto importante a ser destacado é com relação à distribuição de notas da disciplina. A disciplina, tanto na primeira como na segunda oferta, foi organizada em torno de três tipos de avaliações com pesos diferentes, a saber: *Atividades avaliativas (35 pontos)*, *Projeto 1 (25 pontos)* e *Projeto 2 (40 pontos)*. Mais detalhes sobre a distribuição desses pontos são apresentados a seguir.

01 - Leitura e discussão sobre o artigo “A Catedral e o Bazar”, de Eric Raymond. O artigo “A Catedral e o Bazar”, de Eric S. Raymond (Raymond, s.f.), discute as vantagens técnicas do software livre e aborda os mecanismos de funcionamento do desenvolvimento descentralizado (Kon et al., 2011). Este artigo é considerado um dos fatos mais significativos para o crescimento e o amadurecimento do movimento do software livre na década de 1990 (Kon et al., 2011; Steinmacher et al., 2020). As ideias aventadas por Raymond, por exemplo, foram importantes para a tomada de decisão dos executivos na *Netscape* quanto à liberação do código-fonte do navegador *Mozilla*, em 1998 (Kon et al., 2011; Moore, s.f.). Desta forma, é importante que aqueles(as) que se propõem a estudar e adentrar no universo do software livre conheçam tais ideias.

Para este tópico da disciplina, o planejamento adotado foi disponibilizar o artigo aos(as) estudantes na primeira aula da semana e realizar uma discussão sobre ele no fórum do AVA da Instituição. Para guiar tal discussão, um tópico foi aberto no fórum pelo docente da disciplina, com a seguinte apresentação:

No artigo “A Catedral e o Bazar”, Eric S. Raymond apresenta suas observações sobre métodos de Engenharia de Software, com base no processo de desenvolvimento do Linux e em suas experiências administrando o projeto fetchmail, sendo algumas delas: (1) Todo bom trabalho de software começa colocando o dedo na ferida de um programador; (2) Os programadores bons sabem o que escrever. Os grandes sabem o que reescrever (e reusar); (3) Planeje jogar algo fora; você irá, de qualquer maneira; (4) Tratar seus usuários como codesenvolvedores é seu caminho mais fácil para uma melhora do código e depuração eficaz; (5) Libere cedo. Libere frequentemente. E ouça seus frequentes; e (6) Dada uma base grande o suficiente de beta-testers e codesenvolvedores, praticamente todo problema será caracterizado rapidamente e a solução será óbvia para alguém.

Após essa apresentação do assunto, solicitou-se que os(as) estudantes realizassem as seguin-

tes atividades:

Dadas as observações acima, escolha 1 (uma) para discutir, apresentando a visão do autor sobre ela e a sua opinião (se você concorda, discorda e por que). Se possível, apresente exemplos para enriquecer a discussão. Após isso, escolha a resposta de um colega seu e faça um comentário sobre ela, complementando-a ou corrigindo algo (construtivamente), caso necessário.

O AVA utilizado na Instituição é o Moodle⁷, o qual possui um tipo de fórum conhecido como “P e R (Perguntas e Respostas)”. Neste tipo de fórum, os(as) estudantes devem postar primeiro, antes de ver as publicações de cada participante, o que pode estimular a criatividade e originalidade das respostas. Esta atividade foi realizada nas duas ofertas da disciplina e teve um caráter avaliativo, representando 5 pontos da nota final da disciplina.

02 - Apresentação e discussão sobre o documentário “Revolution OS”. O documentário “Revolution OS” (Moore, s.f.), de 85 minutos, apresenta várias personalidades e fatos históricos importantes relacionados ao movimento de software livre. Alguns exemplos são Richard M. Stallman e seu projeto GNU, Linux Torvalds e o sistema operacional Linux, o surgimento do conceito *open-source* e da organização OSI (*Open-Source Initiative*), entre outros. Analogamente ao que ocorreu na atividade 01, esta atividade consistiu em apresentar o documentário aos(as) estudantes na primeira aula da semana e, posteriormente, realizar uma discussão a seu respeito, via fórum.

Neste caso, a atividade proposta foi a seguinte:

Após ter assistido ao “Revolution OS”, escolha um aspecto retratado neste documentário que lhe chamou a atenção e discorra sobre ele. Por exemplo, o documentário contou a história de duas importantes iniciativas do mundo do Software Livre: a Free Software Foundation (FSF) e a Open-Source Initiative (OSI). Uma boa ideia seria discorrer sobre as principais similaridades e pontos de divergência entre a FSF e a OSI, do ponto de vista da definição de Software Livre.

Esta atividade também foi avaliativa e representou 5 pontos da nota final da disciplina. As atividades 01 e 02 foram realizadas de modo assíncrono, via fórum, com o intuito de ambientar os(as) estudantes ao processo de contribuição para projetos de software livre que, na maior parte das vezes, é realizado por meio de interações assíncronas, via lista de emails ou por meio de sistemas de controle de versões de código na nuvem, tal como o *GitHub*⁸.

É importante ressaltar que esta atividade foi realizada, em caráter obrigatório, apenas na oferta 2022/1 (Tabela 1). Isso ocorreu, pois percebeu-se que grande parte do conteúdo apresentado no documentário já era coberto pelas aulas expositivas da disciplina. Além disso, após a primeira oferta, houve a necessidade de inclusão de um novo tópico (#8) ao conteúdo programático da disciplina, o que exigiu uma semana de aula para atendê-lo. Por esses motivos, na oferta 2022/2, o documentário “Revolution OS” foi disponibilizado como um conteúdo complementar e não-avaliativo da disciplina e a carga horária dedicada a essa atividade foi realocada outro tópico.

03 - Conceitos, história e modelo de negócios em Software Livre e 04 - Licenciamento de software. Esses dois tópicos do conteúdo programático foram abordados por meio aulas expositivas, com exercícios de fixação a serem realizados via AVA. Para o tópico “Conceitos, história e modelos de negócio em Software Livre” foi utilizado, principalmente, o capítulo da Jornada de Atualização em Informática (JAI) de 2011, intitulado “Software Livre e Propriedade Intelec-

⁷<https://moodle.org>

⁸<https://github.com>

tual: Aspectos Jurídicos, Licenças e Modelos de Negócio” (Kon et al., 2011). Quanto ao tópico “Licenciamento de software”, o livro “101 - Open Source Licensing: From Concepts to Practical Implications” (Pinto, 2021) foi a principal referência utilizada.

Ambos os tópicos foram apresentados nas duas ofertas da disciplina e acompanhados por exercícios avaliativos, totalizando *10 pontos* da nota final da disciplina. Para a realização dos exercícios, a ferramenta “Questionário”, do Moodle, foi utilizada. Por meio desta ferramenta, questões podem ser aleatoriamente escolhidas a partir de um banco de questões pré-cadastrado pelo(a) docente. No contexto desta disciplina, foram criadas questões discursivas sobre os tópicos abordados em sala de aula, como por exemplo: *Cite duas ou mais razões para a escolha de Software Livre ao invés de Software Restrito.*

05 - Palestra convidada. O objetivo deste tópico foi oferecer flexibilidade ao conteúdo programático da disciplina. Ou seja, por meio dele, o(a) docente pode convidar pesquisadores/profissionais da indústria que tenham experiência sobre a temática de desenvolvimento de software livre para palestrarem a respeito de assuntos que não são abordados na disciplina ou que são abordados de forma superficial. Alguns exemplos de assuntos interessantes a serem abordados são: aspectos jurídicos relacionados ao desenvolvimento de software, desafios encontrados por pessoas novatas durante o processo de contribuição para projetos software livre, dinâmica das comunidades de software livre, ética e profissionalismo no desenvolvimento de software livre, entre outros. Outra oportunidade oferecida por esse tópico do conteúdo programático é permitir que estudantes que já cursaram a disciplina e que continuam contribuindo com projetos de software livre possam ser convidados(as) para comentarem sobre suas experiências, desafios e lições aprendidas.

06 - Versionamento de software com Git e Github - Parte 1 e 07 - Versionamento de software com Git e Github - Parte 2. O conhecimento sobre versionamento de software é fundamental para se obter bons resultados no processo de contribuição com projetos de software livre. Hoje em dia, *Git* se tornou a ferramenta do estado-da-arte sobre este assunto. Assim sendo, estes dois tópicos têm como intuito prover uma base de conhecimento requerida para que os(as) estudantes consigam realizar uma contribuição com algum projeto de software livre. Na primeira aula, comandos mais elementares, tais como *git clone*, *git commit* e *git push*, foram apresentados, enquanto que na segunda aula, o enfoque foi apresentar conceitos um pouco mais complexos e que são bastante utilizados por aqueles(as) que se propõem a contribuir para projetos de software livre, tais como *fork*, *branch* e *pull requests*. Ambos os tópicos foram apresetados nas duas ofertas da disciplina e acompanhados de exercícios práticos de fixação, totalizando *10 pontos* da nota da disciplina. Para estes exercícios, também foi utilizada a ferramenta “Questionário”, do Moodle. Um exemplo de questão proposta foi a seguinte⁹:

Realize as atividades abaixo e envie o link do seu repositório como resposta para esta questão. (1) Crie uma conta no Github, caso ainda não tenha uma. (2) Usando sua conta, crie um repositório denominado minhabio-gcc259, contendo um arquivo README. (3) No arquivo README do seu repositório, elabore um breve currículo seu, contendo dados pessoais e profissionais, cursos realizados, entre outras informações (faça uso de elementos de cabeçalho, itálico e negrito, listas, links, etc). (4) Clone o repositório minhabio-gcc259 em sua máquina local. (4) Adicione um avatar (ou foto) ao arquivo README do seu repositório. Crie um commit e depois faça um push para o repositório remoto (assim, seu repositório deverá conter, pelo menos, dois commits).

⁹*minhabio-gcc159* é o nome do repositório que deveria ser criado pelo(a) estudante em sua conta pessoal no GitHub, contendo informações básicas sobre seu perfil acadêmico e profissional, como se fosse um breve currículo.

Como pode ser visto, como resposta para a atividade em questão, o(a) estudante deveria enviar o *link* do seu repositório do *GitHub*, que deveria estar de acordo com as instruções apresentadas na atividade.

08 - Diretrizes para mantenedores de projetos de Software Livre. Esse tópico do conteúdo programático também foi abordado por meio aula expositiva, com exercícios de fixação a serem realizados via AVA. A principal fonte de consulta para elaboração deste conteúdo foi o artigo intitulado “Let me in: Guidelines for the Successful Onboarding of Newcomers to Open Source Projects” (Steinmacher et al., 2019). Este tópico foi apresentado apenas na oferta 2022/2, uma vez que ele surgiu a partir da análise da avaliação realizada com os(as) estudantes da disciplina da oferta 2022/1, conforme explicado na Seção 4. Exercícios avaliativos acompanharam esse tópico, totalizando 5 pontos da nota final da disciplina.

09 - Projeto 1: repositório sobre personalidades envolvidas com Software Livre. De acordo com Steinmacher et al. (Steinmacher et al., 2020), existem diferentes formas de contribuir para projetos de software livre, tais como, enviar relatos de *bugs* encontrados, escrever documentação, traduzir a interface gráfica, promover a comunidade, responder questões de usuários, entre outros. Além desses, uma das principais formas de contribuição que vêm à mente quando se fala em projetos de software livre é a contribuição com código fonte (Steinmacher et al., 2020). O objetivo dos dois últimos tópicos do conteúdo programático da disciplina (#09 e #10), ambos apresentados nas duas ofertas, foi colocar os(as) estudantes em contato com esse tipo de atividade.

O “Projeto 1” foi proposto com o intuito de preparar os(as) estudantes para que sua experiência ao contribuir para um projeto de software livre real fosse menos “dolorosa”; é uma espécie de treino nas águas calmas de uma piscina, antes de se aventurar no mar aberto dos projetos de software livre reais. Assim sendo, os(as) estudantes deveriam formar equipes de até 3 (três) participantes e enviar suas contribuições para o projeto denominado “Personalidades do Software Livre”¹⁰. Este projeto, desenvolvido pelo docente da disciplina, tem como objetivo apresentar detalhes sobre a vida e as contribuições de algumas personalidades do mundo do Software Livre. A ideia é que cada turma da disciplina seja responsável pela pesquisa, redação e manutenção da história de uma personalidade. Para a oferta 2022/1 da disciplina, *Richard Matthew Stallman* foi escolhido como personalidade a ser apresentada. Para a oferta 2022/2, o escolhido foi *Linus Benedict Torvalds*.

O projeto, hospedado publicamente no *GitHub*, possui uma lista de *issues*, as quais as equipes podem escolher para contribuir. Além disso, foi permitido às equipes criar novas *issues* e cabia ao docente da disciplina aceitá-las ou não. Inicialmente, foram cadastradas 10 *issues*, que versavam sobre vida pessoal, carreira profissional, polêmicas, entre outros. A título de exemplo, a *issue* “Carreira profissional” foi descrita da seguinte forma:

Detalhes da vida profissional, tais como: onde e quando trabalhou profissionalmente? quais funções desempenhava? quais foram suas principais contribuições profissionais? qualquer outra informação que considere relevante sobre sua vida profissional

Além da descrição das *issues*, havia também uma seção na página principal do repositório, denominada “Como contribuir com o projeto?”, a qual especificava as regras para contribuição a serem seguidas pelas equipes (ver Figura 1).

¹⁰<https://github.com/ufla-gcc259/personalidades-sl>

Como contribuir com o projeto?

Com sua equipe de trabalho prático da disciplina, escolha uma *issue* do repositório para ser resolvida e **comunique aos outros** sobre isso.

É possível que a equipe proponha uma *issue* a ser trabalhada, desde que se tenha a anuência do professor da disciplina.

Faça um *fork* do repositório, crie uma nova *branch* e desenvolva o trabalho que se pede na *issue*, levando em conta as diretrizes abaixo:

- Utilize o arquivo correto para realizar as modificações, conforme especificado na *issue* escolhida. Por exemplo: `vida-pessoal.md`, `carreira.md`
- O texto elaborado pode conter um teor mais informal, contudo, a ortografia e gramática da língua portuguesa devem ser respeitadas.
- Ao usar siglas, não esqueça de descrevê-las no texto. Por exemplo: FSF (*Free Software Foundation*).
- Cite as fontes de sua pesquisa, **sempre**. Para isso, utilize o recurso de notas de rodapé, o qual é explicado com mais detalhes [aqui](#).

Finalizado o trabalho, submeta um PR (*Pull Request*), cujo corpo da mensagem deve estar de acordo com o seguinte *template*:

```

Descrição: apresente uma descrição sucinta (aprox. 2 ou 3 linhas) do trabalho realizado pela equipe.
<LINHA EM BRANCO>
Fluxo de trabalho e equipe: descreva como foi conduzido o trabalho em equipe. Quem fez o que? Quem c
<LINHA EM BRANCO>
Principais dificuldades: descreva as principais dificuldades encontradas durante a realização do tra
<LINHA EM BRANCO>
Fixes #<NÚMERO DA ISSUE QUE FOI RESOLVIDA>

```

Acompanhe o processo de avaliação do PR, pois modificações podem ser solicitadas pelo mantenedor do repositório.

Figura 1: Seção “Como contribuir com o projeto?”, do README do repositório “Personalidades do Software Livre”.

Foi estabelecido um período de 2 (duas) semanas para realização do “Projeto 1”. Ao final da primeira semana, um(a) participante de cada equipe deveria realizar um *Pull Request* (PR)¹¹ com as contribuições da equipe e enviar o *link* para o PR, via AVA. O docente da disciplina, então, revisava cada PR e enviava um *feedback* para a próxima entrega. Ao final da segunda semana, cada equipe deveria ter atualizado o PR com as possíveis modificações requisitadas pelo docente. Uma vez encerrado o período especificado para o projeto, o docente da disciplina revisava novamente cada PR e atribuía uma nota, levando em consideração os seguintes critérios de avaliação: (i) nível de interação da equipe via *Github* (8,75 pontos); (ii) adequação do PR às diretrizes para contribuição do projeto (8,75 pontos); e (iii) qualidade do material produzido pela equipe (7,5 pontos). O “Projeto 1” representava 25 *pontos* da nota final da disciplina.

10 - Projeto 2: contribuições para projetos de software livre reais. Uma vez tendo experimentado o processo de contribuição em um ambiente controlado, i.e., entre as equipes e o docente da disciplina apenas, seria a hora de contribuir para projetos de software livre reais, não vinculados à disciplina em si. Cada equipe deveria realizar 2 (dois) tipos de contribuição para projetos de software livre: (i) uma contribuição relacionada à documentação ou tradução; e (ii)

¹¹ *Pull Request* (PR) é um recurso da plataforma *Github* que permite aos(as) contribuidores(as) informar outras pessoas sobre as alterações feitas em uma *branch* de um repositório. Após a abertura de um PR, as alterações podem ser discutidas e revisadas com colaboradores(as), antes que as alterações sejam mescladas na *branch* base (*Github.com*, s.f.-b)

uma contribuição de código, que poderia ser, por exemplo, a correção de algum *bug*, a refatoração de alguma parte do código ou então a implementação de uma nova *feature*. As contribuições não precisavam ocorrer, necessariamente, em um mesmo projeto de software livre. Na oferta 2022/2, especificamente, havia também a possibilidade de a equipe criar o seu próprio projeto de software livre. Caso o projeto proposto envolvesse documentação e código, ele seria considerado como um dos tipos de contribuição possíveis apenas (documentação ou código). Essa foi mais uma mudança realizada no conteúdo programático da disciplina, a fim de contemplar as demandas levantadas a partir da avaliação da oferta 2022/1.

O primeiro passo para contribuir com software livre é escolher o projeto e uma tarefa apropriada para contribuição, porém, isso nem sempre é algo trivial e pode requerer algum tempo (Steinmacher et al., 2020). Pensando nisso, foi elaborada uma “Chamada de projetos para contribuição”, a qual foi encaminhada para a lista de estudantes de graduação e pós-graduação e de docentes do departamento de Ciência da Computação da Universidade Federal de Lavras. O objetivo dessa chamada foi conhecer projetos (por exemplo, apoios computacionais desenvolvidos por estudantes de iniciação científica ou mestrado, notas de aula, exemplos de código usados em aula, entre outros) que pudessem ser disponibilizados para receber contribuições das equipes da disciplina. A justificativa apresentada aos membros das listas foi que esta seria uma interação do tipo “ganha-ganha”, na qual os(as) estudantes aprenderiam o conteúdo, ao cumprir um requisito da disciplina, e, ao mesmo tempo, o projeto em questão poderia ter uma ou mais *issues* resolvidas. Ao todo, foram recebidos 16 (dezesesseis) projetos aptos para contribuição (8 na primeira oferta e 8 na segunda), os quais foram disponibilizados às equipes, contendo as seguintes informações: nome do projeto, breve descrição, *link* do repositório, tipo(s) de contribuição esperado(s) (implementação, tradução, etc) e contato da pessoa mantenedora. Cada projeto deveria manter, também, uma ou mais *issues* descrevendo cada contribuição individualmente.

Um exemplo de projeto disponibilizado, juntamente com uma de suas *issues*, é apresentado a seguir:

Projeto: Jogo Nave

Breve descrição: um jogo simples de nave, usando componentes básicos da API Swing em Java. Ele é usado nas aulas iniciais de disciplinas de Orientação a Objetos.

Link do repositório: <https://github.com/ufla-ppoo/Nave2>

Tipo(s) de contribuição: desenvolvimento

Contato da pessoa mantenedora: Prof. Júlio César (juliocesar.alves@ufla.br)

Issue #1 - Encapsular objetos do Cenário: a classe “JogoNave” cria três objetos (nave, inimigo e placar) para passar para o construtor do cenário. Isso faz sentido no projeto original Nave 1.0 (que utiliza o BlueJ). Neste projeto, faria mais sentido que a própria classe “Cenario” criasse esses objetos.

Além da lista de projetos advindos da chamada realizada, os recursos abaixo são disponibilizados aos(as) estudantes para auxiliar na escolha de projetos para contribuição:

- **CodeTriage**¹²: uma plataforma para auxiliar na tarefa de escolher um projeto para contribuição. A plataforma conta também com alguns tutoriais sobre essa temática.
- **Awesome First PR Opportunities**¹³: um repositório com sugestões de projetos para con-

¹²<https://www.codetriage.com>

¹³<https://github.com/MunGell/awesome-for-beginners>

tribuição, classificados por linguagem de programação.

- **Encontrando maneiras de contribuir com o código aberto no GitHub**¹⁴. Tutorial do *GitHub* sobre como contribuir para projetos de software livre relevantes.

Uma vez escolhido o projeto, as equipes deveriam acessar o *issue tracker* do projeto e buscar por tarefas abertas e etiquetadas como *good-for-beginner*, *first-timers-only*, *easy*, *newcomer*, *good-first-bug*, seguindo recomendação de Steinmacher et al. (Steinmacher et al., 2020).

Foi estabelecido um período de 4 (quatro) semanas para realização do “Projeto 2”. Ao final da terceira semana, um(a) participante da equipe deveria enviar, via AVA, o nome dos(as) participantes da equipe e o *link* dos 2 (dois) PR com as contribuições realizadas. Caso a equipe tenha desenvolvido seu próprio projeto de software livre, ela deve enviar o link para o repositório do projeto. É importante ressaltar que a criação de um projeto de software livre entraria como uma das contribuições exigidas no trabalho, sendo assim, a equipe deveria ainda enviar o PR referente ao outro tipo de contribuição.

Por fim, na quarta semana, cada equipe deveria realizar uma apresentação (com slides) sobre as contribuições realizadas. As apresentações teriam entre 8 e 15 minutos¹⁵ e deveriam contemplar, no mínimo, os seguintes assuntos: quais foram os projetos e as *issues* escolhidas? como foi o fluxo de trabalho entre os(as) participantes da equipe? houve algum tipo de comunicação com a pessoa mantenedora do projeto, caso sim, como foi? o que foi feito para resolver as *issues* e quais foram os resultados obtidos? os PRs foram aprovados? quais foram as principais dificuldades encontradas durante o processo de contribuição?

O docente da disciplina revisou os PRs e a apresentação da equipe e atribuiu uma nota, levando em consideração os seguintes critérios de avaliação: (i) nível de interação da equipe via *GitHub* (7,5 pontos); (ii) adequação dos PRs às diretrizes para contribuição do(s) projeto(s) (8,75 pontos); (iii) qualidade e profundidade da contribuição realizada pela equipe (8,75 pontos); e (iv) qualidade da apresentação realizada (15 pontos). Como pode ser visto na Tabela 1, o “Projeto” representou 40 pontos da nota final da disciplina.

3.2. Execução e Resultados

Pode-se dizer que a oferta 2022/1 da disciplina “Desenvolvimento de Software Livre” ocorreu com êxito. A disciplina contou com 30 estudantes matriculados(as), preenchendo todas as vagas disponíveis. Todos os(as) 30 estudantes chegaram até o final da disciplina, realizando no mínimo dois tipos de contribuição para projetos de software livre. Quanto ao percentual de aprovação, este foi de 100%, com todas as notas finais acima de 85 (oitenta e cinco) pontos. Quanto à oferta 2022/2, novamente, todas as vagas foram preenchidas, não houve desistências e o índice de aprovação foi de 100%, com apenas duas notas abaixo de 80 pontos.

Entende-se que o fato de a disciplina ser eletiva (estudantes podem escolher, ou não, fazê-la) pode ter influenciado positivamente o bom resultado. Contudo, levando em consideração outras

¹⁴Disponível em <https://docs.github.com/pt/get-started/exploring-projects-on-github/finding-ways-to-contribute-to-open-source-on-github>

¹⁵Foi estabelecido um limite mínimo de tempo para a apresentação, a fim de que os(as) estudantes soubessem que apresentações muito curtas e pouco explicativas poderiam ser penalizadas.

disciplinas eletivas já ministradas pelo mesmo docente, o resultado desta esteve, realmente, acima da média. A título de comparação, em geral, o índice médio de desistências em disciplinas eletivas do Departamento de Ciência da Computação desta Instituição gira em torno de 10%. Além do fato de ser uma disciplina eletiva, outros fatores podem ter influenciado positivamente o resultado, tais como: (i) a maioria dos(as) estudantes estava em um estágio avançado do curso (da metade para frente), ou seja, já haviam passado pelo início de curso, muitas vezes conturbado por causa da transição do ensino médio para a universidade, mudanças de residência, entre outros; (ii) a relevância do assunto para o momento atual do mercado de desenvolvimento de software, no qual várias empresas, tais como Facebook, Google, entre outras, têm promovido o desenvolvimento e uso de software livre¹⁶; (iii) o fato de o processo avaliativo da disciplina ter sido distribuído ao longo do semestre, permitindo que os(as) estudantes recebessem *feedbacks* parciais sobre seu progresso e pudessem se recuperar, se necessário; entre outros.

Quanto às atividades que envolveram discussões por meio de fóruns (**atividades #01 e #02**) no AVA, considera-se que o resultado foi satisfatório, tendo a participação de todos(as) estudantes. Além disso, a utilização do fórum “P e R (Perguntas e Respostas)” contribuiu bastante para a discussão, pois garantiu a troca de experiências mais diversificadas. A grande maioria dos(as) estudantes fizeram apenas o que se pediu na tarefa, por exemplo, na discussão sobre o artigo de Eric Raymond, eles(as) tinham que postar sua opinião sobre um dos pontos do artigo e comentar a postagem de um(a) colega. Contudo, houve alguns poucos casos de estudantes que demonstraram maior engajamento, ao ler e comentar mais de uma postagem dos(as) colegas. Houve uma predominância na escolha do tópico 5 (Libere cedo. Libere frequentemente. E ouça seus fregueses) do artigo de Eric Raymond. Isso se deve, possivelmente, pelo fato de muitos(as) estudantes estarem envolvidos em um contexto de desenvolvimento ágil de software, nas empresas em que estagiam, e esse ser um tópico se alinha bastante com os princípios de agilidade.

Como ideia para as próximas ofertas, pretende-se procurar outros artigos, filmes e documentários relacionados ao universo de software livre, a fim de serem explorados nestas duas primeiras semanas de aulas. As Figuras 2 e 3 apresentam, respectivamente: (i) um exemplo de resposta de um(a) estudante no fórum da atividade #01; e (ii) o comentário de um(a) estudante sobre a resposta de seu(sua) colega.

Quanto às aulas expositivas a respeito dos conceitos e da história do software livre, sobre licenciamento de software e sobre diretrizes para mantenedores de projetos de software livre (**atividades #03, #04 e #08**, respectivamente), não houve grandes novidades, a não ser a boa interação dos(as) estudantes com perguntas na sala de aula, principalmente, a respeito de licenciamento de software. O tempo de aula foi suficiente para apresentar os conteúdos, considerando também o tempo para perguntas dos(as) estudantes.

A respeito da palestra convidada (**atividade #05**), na oferta 2022/1, a mesma foi realizada por um docente universitário com experiência em pesquisa sobre software livre. O tema da palestra foi “Open-source, como começar? motivações e desafios!”¹⁷. Já na oferta 2022/2, o palestrante também era docente universitário e tinha larga experiência com software livre e segurança da informação, por isso, sua palestra foi intitulada “Software livre e desenvolvimento seguro”¹⁸.

¹⁶Por exemplo, Google e Facebook são as empresas criadoras e mantenedoras de dois dos principais *frameworks* *JavaScript* utilizados pelo mercado, respectivamente, *Angular* e *React*; ambos os *frameworks* são software livre

¹⁷<https://www.youtube.com/watch?v=z1cmMSek8OQ>

¹⁸<https://www.youtube.com/watch?v=Ee1HhsDLpjQ>

Re: Discussão sobre o artigo "A Catedral e o Bazar", de Eric S. Raymond

por [redacted] - Wednesday, 8 Jun 2022, 22:26

7. Libere cedo. Libere frequentemente. E ouça seus fregueses. Segundo o autor, a ideia de liberar frequentemente e ouvir aos feedbacks, vem do fato de manter os seus contribuidores e usuários sempre estimulados e recompensados, fazendo isso de forma que este sintam-se valorizado e tenha seu trabalho constantemente melhorado e alterado. O exemplo utilizado pelo autor para justificar isso, foi o de Linus Torvalds que, no início do projeto do Linux, liberava um novo kernel mais de uma vez por dia, mantinha seus co-desenvolvedores e sempre procurava novos colaboradores para o projeto. Concordo com a lição passada pois, acredito que a release rápida de novas features, garante aos usuários a possibilidade de conhecer novas ferramentas, aprender a utilizá-las e encontrar erros no início de uma nova ferramenta que facilitarão o desenvolvimento futuro (corroborando isso, encontrei um artigo explicando dez motivos pelos quais um bug deve ser resolvido assim que encontrado: <https://www.ministryoftesting.com/dojo/lessons/ten-reasons-why-you-fix-bugs-as-soon-as-you-find-them>). Além disso, a liberação frequente é uma abordagem bastante utilizada na engenharia de software, que tem como objetivo reduzir o custo, tempo e risco das novas alterações e features permitindo mudanças incrementais (mais sobre: https://en.wikipedia.org/wiki/Continuous_delivery).

Figura 2: Postagem no fórum da atividade 01.

Re: Discussão sobre o artigo "A Catedral e o Bazar", de Eric S. Raymond

por [redacted] - Friday, 10 Jun 2022, 16:05

Incrementando a questão da entrega frequente em engenharia de software, acho válido mencionar o Scrum, que tem como uma de suas bases a "entrega contínua". Acho que dá pra fazer um paralelo entre a lição e o que você disse com o Scrum, uma vez que se assemelham com a ideia do Scrum de reduzir os erros do produto final a partir da opinião do cliente em diversos pontos da produção.

Figura 3: Resposta à uma postagem no fórum da atividade 01.

Ambas as palestras foram realizadas de forma *online*. Uma das vantagens de a palestra ser *online* é que, além de poder contar com docentes de outras instituições, sem o custo de transporte e acomodação, a palestra pode ser facilmente gravada e disponibilizada para acesso posterior às pessoas interessadas. Além disso, foi possível abrir espaço para que outras pessoas participassem da palestra, não apenas os(as) estudantes matriculados(as) na disciplina.

A respeito das atividades sobre *Git* e *Github* (atividades #06 e #07), as mesmas foram realizadas em laboratório de informática. Havia certa heterogeneidade com relação ao conhecimento dos(as) estudantes sobre este assunto. Porém, na oferta 2022/1, isso foi notado apenas durante a realização dos trabalhos práticos. Desta forma, na oferta 2022/2, houve uma investigação melhor a respeito do conhecimento prévio dos(as) estudantes já nas primeiras semanas de aulas. Isso foi feito pedindo aos(as) estudantes que respondam algumas das perguntas do questionário avaliativo discutido na Seção 4 logo no início do semestre.

Quanto ao “Projeto 1”, como foi dito anteriormente, notou-se, em ambas as ofertas, dificuldade por parte de alguns(mas) estudantes, com relação ao uso do *Git* e do *GitHub*, principalmente, no que se refere ao tratamento de conflitos e ao fluxo de trabalho envolvendo mais de uma pessoa por equipe. No entanto, o fato de o projeto ter uma entrega parcial e a possibilidade de receber o *feedback* do docente permitiu que os(as) estudantes corrigissem seus erros até a data de entrega final.

Ainda sobre o “Projeto 1”, foi permitido que as equipes propusessem novas *issues*, com a anuência do docente. Isso é interessante, pois fomenta a criatividade dos(as) estudantes e já configura-se como um tipo de contribuição. Por exemplo, na oferta 2022/1, uma equipe propôs uma *issue* para incluir detalhes sobre os prêmios e reconhecimentos que a personalidade do mun-

[Voltar](#)

Richard Matthew Stallman

Esta foi a personalidade do mundo do Software Livre escolhida para ser estudada pela turma 2022/1 da disciplina *Desenvolvimento de Software Livre*.

Abaixo, estão os principais assuntos sobre a vida e as contribuições desta importante figura para o movimento do Software Livre mundial.

- [Vida pessoal](#)
- [Formação acadêmica](#)
- [Carreira profissional](#)
- Relação com o movimento de Software Livre
 - [O projeto GNU](#)
 - [A FSF \(Free Software Foundation\)](#)
 - [A licença GPL \(GNU Public License\)](#)
- [Polêmicas](#)
- [Curiosidades](#)
- [Produção \(livros, artigos, programas de computador, etc\)](#)
- [Prêmios e reconhecimentos](#)
- [Álbum de fotos](#)

Figura 4: Página sobre Richard M. Stallman.

do do software livre havia recebido, *issue* esta, que não havia sido pensada originalmente pelo docente da disciplina. Contudo, um problema que poderia ocorrer é que, como havia um número de *issues* correspondente ao número de equipes, ao propor uma nova, alguma *issue* importante seria deixada sem resolução. A fim de resolver esse problema, na oferta 2022/2 da disciplina, as *issues* foram etiquetadas como “rígidas” ou “flexíveis”, sendo que o primeiro tipo de *issue* precisa obrigatoriamente ser atendido, enquanto que o segundo tipo pode ser substituído por outra, a critério da equipe. Assim, caso todas as *issues* flexíveis já tenham sido escolhidas, a única possibilidade será atender a uma *issue* rígida.

Outro ponto interessante a se destacar sobre o “Projeto 1” é que, além de enviar a contribuição, cada equipe deveria informar, na descrição do seu PR, como foi o fluxo de trabalho e quais foram as principais dificuldades encontradas. A maioria das equipes, em ambas as ofertas, optou por escolher um(a) participante para fazer um *fork*¹⁹ do projeto para sua conta e adicionar os(as) demais como colaboradores(as) do repositório. Assim, todas as modificações eram feitas em um único repositório que, posteriormente, daria origem ao PR.

As Figuras 4 e 5 apresentam, respectivamente, uma parte da página de Richard M. Stallman no repositório “Personalidades do Software Livre” e de sua seção “Vida Pessoal”.

Para corrigir o “Projeto 1”, o docente da disciplina fazia o *checkout* de cada PR para uma nova *branch* da sua cópia local do repositório, usando o comando listado no Código 1.

Assim, ele poderia visualizar o conteúdo do PR de forma mais legível, facilitando sua avaliação. Uma vez feitas as anotações sobre as contribuições entregues, o docente acessava o *link* do PR e iniciava uma discussão com o(a) autor(a) do PR, a fim de solicitar alterações, caso necessário. A Figura 6 mostra um exemplo de solicitação de mudança realizada pelo docente. Finalizadas

¹⁹Um *fork* é uma cópia de um repositório; fazer o *fork* de um repositório permite ao(à) desenvolvedor(a) realizar experiências sem comprometer o projeto original (Github.com, s.f.-a)

[Voltar](#)

Vida pessoal

Onde e quando nasceu?

Richard M. Stallman nasceu no dia 16 de março de 1953, em *Manhattan*, na cidade de Nova Iorque, nos Estados Unidos ^[1]. Filho único.

Quem eram os pais? Com o que eles trabalhavam?

Seus pais possuíam herança judia ^[1]. Seu pai, Daniel Stallman, era tipógrafo, isto é, trabalhava com prensas tipográficas. Sua mãe, Alice Lippman, trabalhava como professora ^[2]. Eles se separaram quando Stallman tinha 9 anos. Ele tinha um relacionamento problemático com os seus pais e sentia como se não possuísse um lar ^[3].

Figura 5: Seção sobre a vida pessoal de Richard M. Stallman.

```
git fetch origin pull/ID/head:BRANCHNAME
```

Código 1: Comando para realizar *checkout* de uma *branch* específica de código..

as correções por parte da equipe, o docente fazia o *merge* para a *branch* base (*main*) do repositório original.

A realização do “Projeto 1” foi bastante positiva, pois permitiu aos(as) estudantes ganharem confiança sobre o processo de contribuição com projetos de software livre. Porém, é no “**Projeto 2**” que os resultados mais interessantes apareceram. Na oferta 2022/1, dos 8 (oito) projetos advindos da chamada para contribuição feita pelo docente às listas de *email* de docentes e estudantes, 4 (quatro) pertenciam a docentes, 2 (dois) pertenciam a estudantes da própria disciplina e 2 (dois) pertenciam a uma aluna de iniciação científica. Sete projetos (87%) foram contemplados com contribuições realizadas pelas equipes da disciplina. Houve também contribuições para outros projetos, fora do âmbito da chamada para contribuição. Na oferta 2022/2, dos 8 (oito) projetos advindos da chamada para contribuição, 7 (sete) pertenciam a docentes e 1 (um) pertencia a um estudante de graduação. Novamente, sete projetos (87%) foram contemplados com contribuições realizadas pelas equipes da disciplina. Também houve contribuições para outros projetos, fora do âmbito da chamada para contribuição.

Em ambas as ofertas, de acordo com as equipes, o maior desafio para realização do trabalho foi encontrar uma *issue* adequada para contribuir. Essa foi uma reclamação recorrente, o que evidencia a necessidade de se buscar outros meios para ajudar os(as) estudantes a encontrarem um projeto/*issue* para contribuição. Segundo os(as) estudantes, as plataformas sugeridas pelo docente (*CodeTriage* e *Awesome First PR Opportunities*) eram úteis até o ponto de se encontrar um projeto, mas, depois disso, ficava difícil achar uma *issue* adequada e que estivesse bem explicada. Por causa disso, a maior parte das equipes procuraram por projetos em que se conhecia de perto o(a) mantenedor(a), a fim de facilitar a comunicação durante o processo de contribuição. Para a próxima oferta da disciplina, pretende-se utilizar o recurso de “diário”, ao longo da execução do “Projeto 2”, assim os(as) estudantes poderão registrar o processo deles(as) de escolha dos projetos, das *issues* e sobre como foi todo o processo de contribuição, incluindo a parte “social/emocional/mental”. A utilização deste recurso terá como intuito buscar descobrir o que “deu errado” e o que

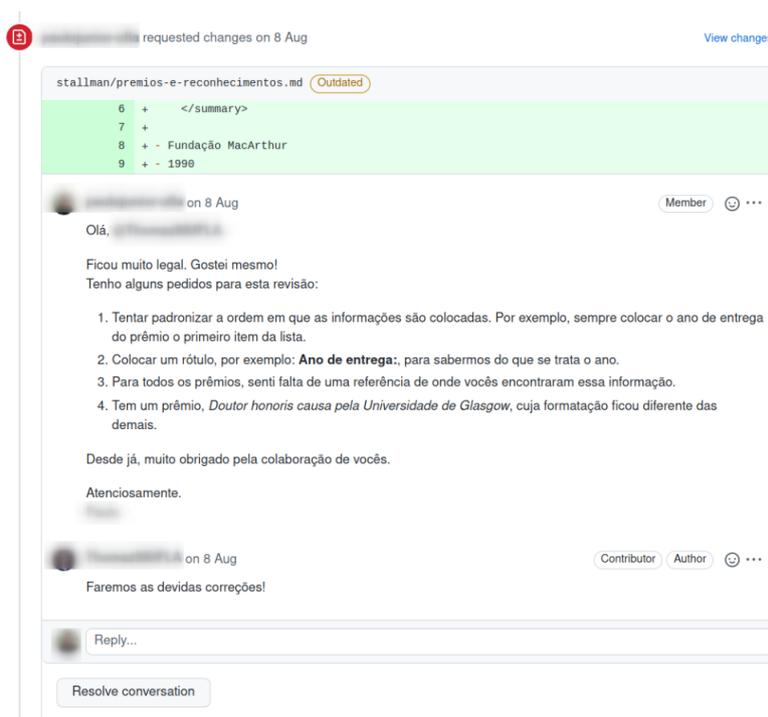


Figura 6: Solicitação de mudança feita pelo docente.

“funcionou bem”, podendo ser uma boa contribuição para guiar docentes quanto ao que adaptar de conteúdo da disciplina.

Como pontos positivos, pode-se destacar a satisfação demonstrada pelos estudantes da disciplina que tiveram seus projetos/*issues* escolhidos para receberem contribuições. Todas as *issues* cadastradas para esses projetos, que incluíam desde a confecção de um logotipo para o projeto até tarefas de implementação, foram concluídas. Um fato interessante a ser destacado é que, em um dos projetos, o estudante cadastrou as *issues* tanto em português quanto em inglês e as etiquetou com “help wanted”, o que fez com que ele recebesse uma contribuição “inesperada” de uma desenvolvedora da Coreia do Sul (ver Figura 7). Outro destaque foi a contribuição de um estudante da disciplina para o projeto *pino-pretty*, uma biblioteca *JavaScript* que mais de 2.1 milhões de downloads semanais²⁰. A contribuição correspondeu à correção de um *bug* no código da biblioteca, a qual foi aceita pelo mantenedor do projeto.

Uma dificuldade percebida, que ocorreu nas duas ofertas da disciplina, foi de uma equipe se esquecer de declarar interesse na realização da *issue*. Isso acabou ocasionando com que outra equipe começasse a trabalhar na mesma *issue* e ao mesmo tempo. Quando a primeira equipe percebeu o equívoco, escreveu para o mantenedor, mas já era tarde. O mantenedor do projeto propôs, então, que a equipe continuasse o trabalho de resolver a *issue*, mas que o *merge* seria feito do PR da equipe que demonstrou publicamente interesse pela *issue*. Em ambas as ofertas da disciplina, as equipes decidiram abandonar o trabalho com a *issue* em questão e partir para outra. Esse ocorrido foi interessante, pois demonstrou que as equipes não estavam interessadas apenas

²⁰<https://www.npmjs.com/package/pino-pretty>

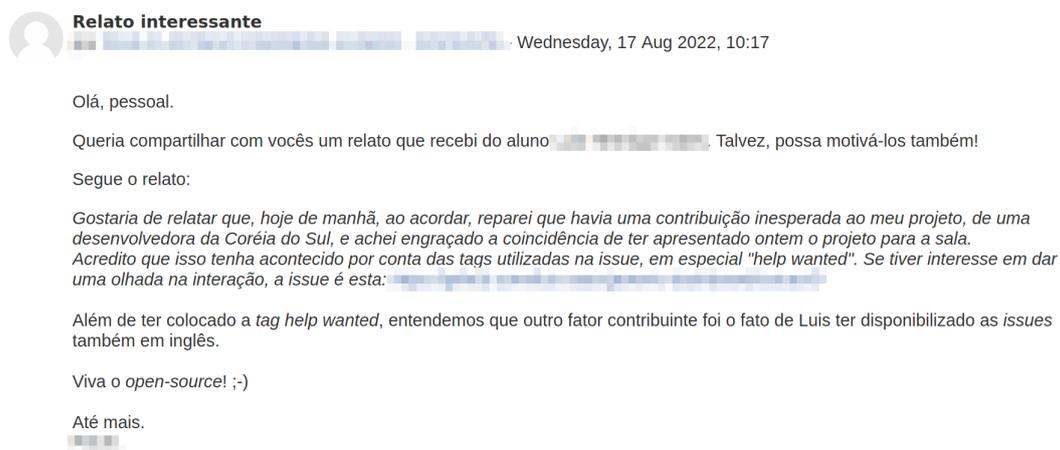


Figura 7: Contribuição internacional no repositório de um estudante da disciplina.

na nota da disciplina, mas sim em poder ter seu trabalho reconhecido perante a comunidade do projeto.

Ao final, todas as equipes entregaram os dois tipos de contribuição exigidos no “Projeto 2” e apresentaram seus resultados perante a turma. No caso da oferta 2022/1, duas equipes de alunos apresentaram seus próprios projetos de software livre como uma das contribuições exigidas pelo trabalho.

4. Avaliação da Disciplina

Esta seção relata os resultados de uma avaliação feita com os(as) estudantes da disciplina “Desenvolvimento de Software Livre”, em suas duas ofertas. A avaliação foi realizada por meio de um questionário eletrônico, de forma anônima, e não teve qualquer impacto na nota dos(as) estudantes.

O questionário ficou disponível por 2 semanas (as últimas duas semanas de aula) e teve a participação de 20 estudantes da disciplina ($\approx 67\%$) na oferta 2022/1 e 17 ($\approx 57\%$), na oferta 2022/2. Acredita-se que a participação não foi unânime por dois motivos: (i) ao final de período a sobrecarga de trabalhos e provas é alto, e isso que pode ter feito com que alguns(mas) estudantes desistissem de dedicar tempo para responder ao questionário; e (ii) a instituição já possui seu próprio processo de avaliação de disciplinas, por meio de questionários, e isso pode ter gerado uma certa aversão a questionários adicionais por parte dos(as) estudantes. O questionário da instituição não foi utilizado como base para esta avaliação, pois tratava-se de um questionário genérico, aplicado a todas as disciplinas e, portanto, não contemplaria as características específicas desta disciplina.

Inicialmente, foi perguntado aos(às) estudantes: *Quanto tempo, fora da sala de aula, você julga ter dedicado semanalmente para as atividades da disciplina?*, tendo como opções *até 30 minutos, até 1 hora, entre 1 e 2 horas ou mais do que 2 horas*. Conforme pode ser visto no gráfico

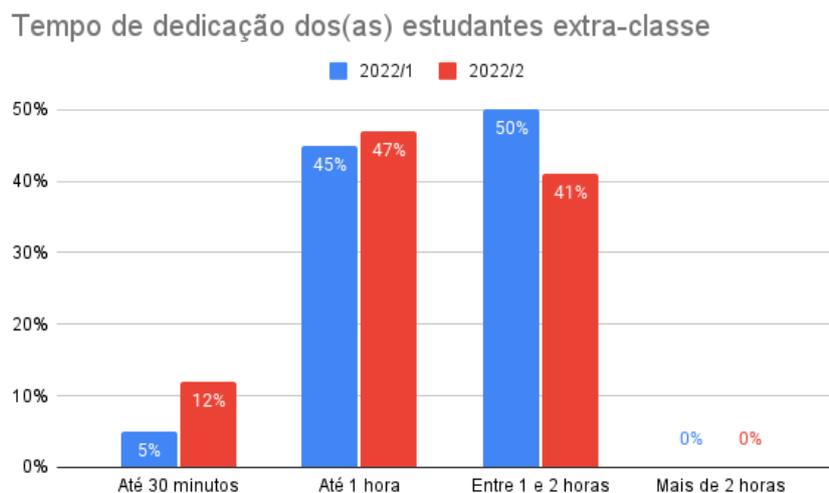


Figura 8: Comparativo do tempo de dedicação extra-classe dos(as) estudantes nas duas ofertas da disciplina.

da Figura 8, na oferta 2022/1, notou-se que 50 % dos(as) estudantes gastava entre 1 e 2 horas extra-classe, com as atividades da disciplina. Os outros 50% foram divididos da seguinte forma: 45 % gastavam até 1 hora e 5 % até trinta minutos. Na oferta 2022/2, os resultados foram bem similares: 41 % dos(as) estudantes apontaram que gastavam entre 1 e 2 horas extra-classe, enquanto que o restante ficou dividido da seguinte forma: 47 % gastavam até 1 hora e 12 % até trinta minutos. Ou seja 100 % dos(as) respondentes admitiram dedicar-se até 2 horas com atividades extra-classe para esta disciplina. Esse resultado é importante, pois a carga horária extra-classe pode influenciar no índice de desistência por parte dos(as) estudantes. Ao que parece, até 2 horas por semana é uma carga horária aceitável para uma disciplina deste tipo.

A segunda pergunta feita foi: *Como você julgaria seu nível de conhecimento sobre os tópicos abaixo, ANTES de cursar a disciplina?*, tendo como opções *Muito Baixo (MB)*, *Baixo (B)*, *Razoável (R)*, *Alto (A)* e *Muito Alto (MA)*. Os tópicos aos quais se refere a pergunta anterior, juntamente com o resumo das respostas são apresentados na Tabela 2. As respostas contemplam as duas ofertas da disciplina, portanto, o número máximo de respondentes é 37 (20 da oferta 2022/1 e 17 da oferta 2022/2), com exceção do tópico “Desenvolvimento de projetos de Software Livre”, o qual foi ministrado apenas da oferta 2022/2.

Tabela 2: Conhecimento sobre alguns tópicos, ANTES da disciplina.

Tópico	MB	B	R	A	MA
Conceitos e história do software livre	15	16	3	2	1
Licenciamento de software	19	11	5	1	1
Modelos de negócio envolvendo software livre	21	12	2	1	1
Processo de contribuição para projetos de software livre	13	10	10	3	1
Desenvolvimento de projetos de Software Livre (oferta 2022/2)	6	3	6	1	1
Conceitos de <i>Git</i> e <i>GitHub</i> (<i>commit</i> , <i>pull request</i> , <i>branches</i>)	0	2	7	12	16

Como pode ser visto, os(as) estudantes declararam menor conhecimento a respeito dos tópicos “Licenciamento de software” e “Modelos de negócio envolvendo software livre”, sendo o

primeiro de grande importância, não apenas para aqueles(as) que pretendem contribuir com projetos de software livre, mas também para quem faz uso de algum serviço, biblioteca ou software livre em geral. A falta de conhecimento sobre esse assunto pode levar a problemas que colocam em risco à autenticidade do software, do ponto de vista legal. Por outro lado, um dos tipos de conhecimento mais bem dominados pelos(as) respondentes foi sobre “Git” e “GitHub”. Isso se justifica, pois grande parte dos(as) estudantes da disciplina já estavam realizando estágio e, conseqüentemente, usavam esses recursos nas empresas em que eles(as) estavam estagiando.

Analogamente, foi perguntado aos(às) estudantes *Como você julgaria seu nível de conhecimento sobre os tópicos abaixo, DEPOIS de cursar a disciplina?*, utilizando-se a mesma escala da pergunta anterior. Os resultados são apresentados na Tabela 3.

Tabela 3: Conhecimento sobre alguns tópicos, DEPOIS da disciplina.

Tópico	MB	B	R	A	MA
Conceitos e história do software livre	0	0	2	13	22
Licenciamento de software	0	0	8	14	15
Modelos de negócio envolvendo software livre	0	2	4	22	9
Processo de contribuição para projetos de software livre	0	0	1	12	24
Desenvolvimento de projetos de Software Livre (oferta 2022/2)	0	0	1	4	12
Conceitos de <i>Git</i> e <i>GitHub</i> (<i>commit, pull request, branches</i>)	0	0	1	6	30

Como pode ser visto, para todos os tópicos, não houve respostas do tipo “Muito Baixo” ou “Baixo”. Os tópicos “Conceitos e história do software livre”, “Processo de contribuição para projetos de software livre” e “Conceitos de *Git* e *GitHub* (*commit, pull request, branches*)” foram os que tiveram melhores resultados, com 19 estudantes tendo marcado a opção “Alto” ou “Muito Alto”, seguidos pelo tópico “Licenciamento de software”. O resultado menos satisfatório foi o do tópico “Modelos de negócio envolvendo software livre”, o que indica a necessidade de melhor explorar esse assunto em ofertas futuras da disciplina. Apesar de ter havido um resultado positivo para o tópico “Conceitos de *Git* e *GitHub* (*commit, pull request, branches*)”, com base nas respostas para as questões dissertativas deste questionário (discutidas mais abaixo), bem como das dificuldades percebidas no “Projeto 1”, salienta-se a necessidade de aprimorar o ensino deste tópico em ofertas futuras.

Por fim, a Tabela 4 apresenta o progresso do conhecimento dos(as) estudantes da oferta 2022/1 sobre os mesmos tópicos, após terem cursado a disciplina. Como era de se esperar, nenhum(a) estudante indicou que seu conhecimento regrediu. No entanto, houve casos em que o conhecimento se manteve o mesmo, principalmente, a respeito do tópico “Conceitos de *Git* e *GitHub* (*commit, pull request, branches*)”. Conforme já comentado, isso se deve ao fato que muitos estudantes já estão em período de estágio e, provavelmente, haviam tido contado com essas tecnologias em seus locais de trabalho.

Tabela 4: Progresso do conhecimento sobre alguns tópicos, DEPOIS da disciplina - Oferta 2022/1.

Tópico	Regrediu	Manteve	Melhorou
Conceitos e história do software livre	0%	5%	95%
Licenciamento de software	0%	0%	100%
Modelos de negócio envolvendo software livre	0%	5%	95%
Processo de contribuição para projetos de software livre	0%	5%	95%
Conceitos de <i>Git</i> e <i>GitHub</i> (<i>commit, pull request, branches</i>)	0%	55%	45%

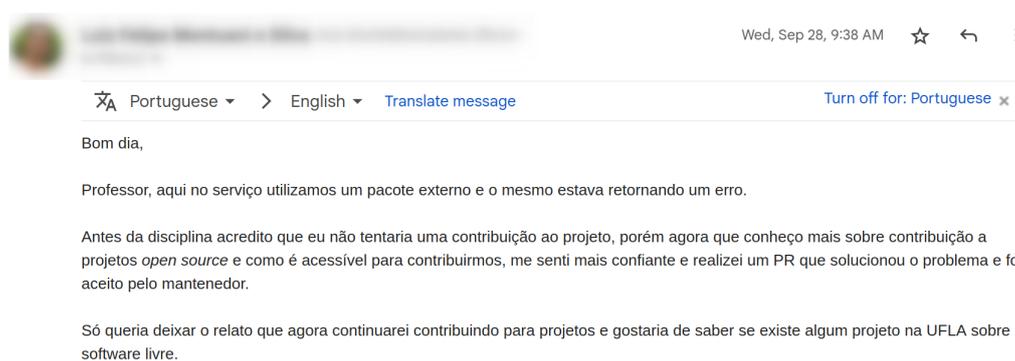


Figura 9: Relato de um estudante sobre sua contribuição para um projeto de software livre real.

A próxima questão pedia que cada estudante julgasse certas afirmações colocadas no questionário, utilizando a seguinte escala: *Discordo Totalmente (DT)*, *Discordo Parcialmente (DP)*, *Neutro (N)*, *Concordo Parcialmente (CP)* e *Concordo Totalmente (CT)*. A Tabela 5 apresenta as afirmações feitas, juntamente com as respostas dadas pelos(as) estudantes.

Tabela 5: Julgamento de algumas afirmações sobre a disciplina.

Afirmação	DT	DP	N	CP	CT
Os assuntos abordados atenderam bem ao propósito da disciplina	0	0	0	3	34
O nível de profundidade com que os assuntos foram abordados na disciplina foi adequado	0	0	1	10	26
A metodologia de avaliação proposta para a disciplina foi adequada	0	0	0	3	34
Eu recomendaria esta disciplina para outras pessoas	0	0	0	1	36
Após ter cursado a disciplina, eu me sinto mais confiante e preparado(a) para contribuir com projetos de software livre	0	0	1	5	31

Mais uma vez, nota-se que não houve respondentes que escolheram as opções “Discordo Totalmente” ou “Discordo Parcialmente”, o que demonstra que os tópicos, o nível de profundidade e a metodologia de avaliação empregados na disciplina foram bem aceitos pelos(as) estudantes. A afirmação com melhor resultado foi “Eu recomendaria esta disciplina para outras pessoas”, o que corrobora com a afirmação anterior. Já a afirmação com resultado menos positivo foi “O nível de profundidade com que os assuntos foram abordados na disciplina foi adequado”, o que corrobora com os resultados da questão anterior, ou seja, o nível de profundidade de alguns tópicos, tais como “Modelos de negócio envolvendo Software Livre”, precisa ser aprimorado.

Um ponto que merece destaque aqui é o relato recebido de um estudante, a respeito da afirmação “Após ter cursado a disciplina, eu me sinto mais confiante e preparado(a) para contribuir com projetos de Software Livre”. Algumas semanas após o encerramento da disciplina, o estudante escreveu ao docente informando que o fato de ter feito a disciplina contribuiu para que ele resolvesse o problema com uma biblioteca de software livre que era utilizada na empresa em que ele estagiava. O relato completo pode ser visto na Figura 9. Um relato como o deste estudante provê indícios de que a disciplina cumpriu seu objetivo e que pode trazer impactos significativos na vida profissional dos(as) estudantes.

Por fim, foi questionado o seguinte: *Há algum assunto que você considera que poderia ser*

abordado na disciplina ou alguma sugestão para as próximas ofertas da disciplina?. Como esta era uma questão não obrigatória, apenas 11 (onze) estudantes a responderam, na oferta 2022/1. Após uma análise das respostas, notou-se que elas podem ser agrupadas em duas principais sugestões: (i) dar um enfoque maior na parte de *Git* e *Github* (4 respostas); e (ii) apresentar conteúdo sobre a criação de um projeto de software livre (5 respostas). Abaixo, encontra-se o comentário feito por um(a) estudante, a respeito do sugestão (ii):

“A única parte que não gostei da disciplina foi o último trabalho, em específico a parte de contribuir com uma nova feature / correção de bugs. Em minha opinião, o primeiro trabalho já cobre bem como deve ser feita a contribuição, sem necessitar conhecimento em nenhuma tecnologia específica, logo, se posso dar uma opinião, para o segundo trabalho seria mais interessante colocar grupos para fazer um repositório com algum projeto, criar issues com tags corretas, criar um padrão para as contribuições e tal. É um outro lado do software aberto que não foi tão tratado.”

Conforme pode ser notado, apresentar o processo de contribuição para software livre do ponto de vista do(a) mantenedor(a) é um conteúdo que foi bastante requerido pelos(as) estudantes na oferta 2022/1. Um desses estudantes comentou, inclusive, que foi necessário um esforço considerável para criar o repositório, escrever as *issues*, criar um guia para contribuição, entre outros. Assim, é mais do que justo recompensar esse tipo de esforço, que está bastante alinhado com a proposta da disciplina.

Desta forma, para na oferta 2022/2, a descrição do “Projeto 2” sofreu modificações, a fim de que seja aceita, como uma opção de contribuição, a criação de projetos de software livre. Além disso, foi incluído um tópico a mais na disciplina, relacionado à “Construção de projetos de software livre”, por meio do qual foram abordados as principais atividades para confecção de um projeto de software livre, tais como redigir um guia para contribuição, descrever as *issues* do projeto, escrever um guia para configuração do ambiente do projeto, entre outros.

Quanto à oferta 2022/2, apenas 6 (seis) estudantes responderam a esta questão. Após uma análise das respostas, notou-se que elas concentram-se em dar um enfoque maior na parte de *Git* e *Github* (2 respostas), modelos de negócio em software livre (1 resposta) e licenciamento de software (2 respostas). Neste caso, pretende-se continuar aprimorando o conteúdo da disciplina a respeito desses tópicos, trazendo mais exemplos concretos sobre os tópicos de modelos de negócio e licenciamento. Além disso, pretende-se acrescentar mais uma semana de atividades prática sobre *Git* e *Github*, destacando alguns pontos mais avançados que não foram explorados nos tópicos sobre essas tecnologias, por exemplo como resolver conflitos de *merge* em um repositório com muitos colaboradores.

5. Considerações Finais

Este trabalho apresentou um relato de experiência sobre a oferta da disciplina “Desenvolvimento de Software Livre” para os cursos de Ciência da Computação e Sistemas de Informação da Universidade Federal de Lavras. Como foi dito na introdução deste artigo, faltam estudos desse tipo na literatura e os *insights* aqui apresentados demonstram que a simples disponibilização de ementas, planos de curso e material didático na Internet, apesar de relevantes, são insuficientes.

Conforme Steinmacher et al. (2020), contribuir para um projeto de software livre é mui-

to mais do que apenas lidar com os desafios técnicos. A contribuição envolve também aspectos não-técnicos, como habilidades sociais e aprendizado de técnicas e ferramentas relacionadas ao processo de contribuição. Notou-se, por meio desta disciplina, que os aspectos técnicos, em termos de conhecer as tecnologias de determinado projeto de software livre, não foi o maior desafio. A maior barreira foi aquela denominada por Steinmacher et al. (2015) e Balali et al. (2018) como “Barreira de Processo”, isto é, barreiras impostas pela organização, por procedimentos ou práticas internas, podendo incluir dificuldades para identificação de tarefas apropriadas para novas pessoas desenvolvedoras, tempo de revisão de código prolongado, entre outros. Neste sentido, entende-se que a disciplina atingiu seu objetivo, ao tentar reduzir o impacto desse tipo de barreira para os(as) estudantes.

Artigo Premiado Estendido

Esta publicação é uma versão estendida de artigo premiado no II Simpósio Brasileiro de Educação em Computação (EduComp 2023), intitulado “A Liga do Pensamento Computacional: uma narrativa distópica para gamificar uma disciplina introdutória de computação”, DOI: 10.5753/educomp.2023.228207

Referencias

- Almeida, A., Melo, C. I., Harvey, M., Lima, M. V., & Chaves, P. J. (2019). Metodologias Ativas à Luz de Comenius: uma Experiência na Pós-Graduação [GS Search]. *Anais do IV Congresso sobre Tecnologias na Educação*, 60-68. <https://doi.org/10.5753/ctrle.2019.8876>
- Andrade, R., Gonçalves, L. L., Rezende, I., & Schiavoni, F. L. (2021). A inserção do músico no universo das práticas musicais com mediações tecnológicas baseadas em software livre [GS Search]. *Anais do VI Congresso sobre Tecnologias na Educação*, 424-429. <https://sol.sbc.org.br/index.php/ctrle/article/view/17589>
- Balali, S., Steinmacher, I., Annamalai, U., Sarma, A., & Gerosa, M. A. (2018). Newcomers’ Barriers... Is That All? An Analysis of Mentors’ and Newcomers’ Barriers in OSS Projects [GS Search]. *Computer Supported Cooperative Work (CSCW)*, 27(3), 679-714. <https://doi.org/10.1007/s10606-018-9310-8>
- Braught, G., McCormick, J., Bowring, J., Burke, Q., Cutler, B., Goldschmidt, D., Krishnamoorthy, M., Turner, W., Huss-Lederman, S., Mackellar, B., et al. (2018). A multi-institutional perspective on H/FOSS projects in the computing curriculum [GS Search]. *ACM Transactions on Computing Education (TOCE)*, 18(2), 1-31.
- Camargo, R., Ribeiro, C., Junior, F. S., Anastácio, P., & Merlin, J. (2019). Utilização de Pygame para Ensino e Aprendizado de Orientação a Objetos [GS Search]. *Anais do XVI Congresso Latino-Americano de Software Livre e Tecnologias Abertas*, 164-167. <https://doi.org/10.5753/latinoware.2019.10355>
- Canez, A., & Severo, C. (2016). Implicações do software livre no ensino e aprendizagem: um estudo de caso no ensino médio politécnico [GS Search]. *Anais do XXII Workshop de Informática na Escola*, 476-484. <https://doi.org/10.5753/cbie.wie.2016.476>

- Carrington, D., & Kim, S.-K. (2003). Teaching software design with open source software [GS Search]. *33rd Annual Frontiers in Education, 2003. FIE 2003.*, 3, S1C-9. <https://doi.org/10.1109/FIE.2003.1265910>
- Castor, F. (2012). *Desenvolvimento de Software de Código Aberto/Livre*. <https://sites.google.com/a/cin.ufpe.br/floss/>
- Deng, L., Dehlinger, J., & Chakraborty, S. (2020). Teaching Software Testing with Free and Open Source Software [GS Search]. *2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, 412-418. <https://doi.org/10.1109/ICSTW50294.2020.00074>
- de Oliveira, A. M., de Azevedo, A. V., Mota, G., Lourenço, R., Dantas, F. G., & Maia, D. (2019). Software Educativo Encantando Cordel [GS Search]. *Anais do IV Congresso sobre Tecnologias na Educação*, 610-616. <https://doi.org/10.5753/ctrl.2019.8939>
- Diniz, L. M., & Ferreira, F. (2018). Kahoot!: um relato de experiência no contexto acadêmico [GS Search]. *Anais do XXVI Workshop sobre Educação em Computação*. <https://doi.org/10.5753/wei.2018.3530>
- Github.com. (s.f.-a). *Bifurcar um repositório*. <https://docs.github.com/pt/get-started/quickstart/fork-a-repo>
- Github.com. (s.f.-b). *Sobre solicitação de pull*. <https://docs.github.com/pt/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-pull-requests>
- Hollister, B., & Leon, J. (2022). Open-Source Development in Undergraduate Software Engineering Education [GS Search]. *International Journal on E-Learning*, 21(2), 139-161.
- Kon, F. (2017). *MAC0470 - Desenvolvimento de Software Livre*. <https://bcc.ime.usp.br/principal/catalogo2017/disciplinas/MAC0470.html>
- Kon, F., Lago, N., Meirelles, P., & Sabino, V. (2011). Software Livre e Propriedade Intelectual: Aspectos Jurídicos, Licenças e Modelos de Negócio [GS Search]. En *Jornada de Atualização em Informática (JAI) do Congresso da Sociedade Brasileira de Computação* (pp. 59-107). Editora PUC-Rio.
- Moore, J. (s.f.). *Revolution OS*. <https://www.youtube.com/watch?v=Z3f-M43DiD4>
- Nascimento, D. M. C., Flach Garcia Chavez, C. v., & Bittencourt, R. A. (2019). Does FLOSS in Software Engineering Education Narrow the Theory-Practice Gap? A Study Grounded on Students' Perception [GS Search]. *IFIP International Conference on Open Source Systems*, 153-164.
- Parreira Júnior, P., Alves, J., Freire, A., Costa, H., & Wiese, I. (2023). Um Relato de Experiência sobre a Oferta da Disciplina Desenvolvimento de Software Livre para os Cursos de Ciência da Computação e Sistemas de Informação [GS Search]. *Anais do III Simpósio Brasileiro de Educação em Computação*, 194-204. <https://doi.org/10.5753/educomp.2023.228199>
- Perin, A., Paim, P., & Valentim, N. (2021). Experiência sobre o uso de ferramentas de apoio à pesquisa experimental em uma disciplina de IHC [GS Search]. *Anais do XXXII Simpósio Brasileiro de Informática na Educação*, 1297-1307. <https://doi.org/10.5753/sbie.2021.218615>
- Petrenko, M., Poshyvanyk, D., Rajlich, V., & Buchta, J. (2007). Teaching Software Evolution in Open Source [GS Search]. *Computer*, 40(11), 25-31. <https://doi.org/10.1109/MC.2007.402>

- Pinto, G. (2021). *Open Source Licensing 101 - From Concepts to Practical Implications* (1ª). Self-published.
- Raymond, E. S. (s.f.). *The Cathedral and the Bazaar*. <http://catb.org/esr/writings/cathedral-bazaar>
- Ribeiro, G., Reis, J., Franco, F., Felix, M., Guedes, A., & Nogueira, U. (2017). Software livre como ferramenta no processo do ensino aprendizado: uma experiência com turmas do EJA [GS Search]. *Anais do XXIII Workshop de Informática na Escola*, 974-982. <https://doi.org/10.5753/cbie.wie.2017.974>
- Rossi, J., & Koike, C. (2022). LPBOT: Um assistente virtual de apoio ao ensino de programação [GS Search]. *Anais do XIX Congresso Latino-Americano de Software Livre e Tecnologias Abertas*, 133-136. <https://doi.org/10.5753/latinoware.2022.228030>
- Silva, E. H. (2018). A Gamificação na Educação Superior a Distância: o Programa Desafio Nota Máxima [GS Search]. *Anais do II Seminário de Educação a Distância da Região Centro-Oeste*. <https://doi.org/10.5753/seadco.2018.14839>
- Steinmacher, I., Conte, T., Gerosa, M. A., & Redmiles, D. (2015). Social Barriers Faced by Newcomers Placing Their First Contribution in Open Source Software Projects [GS Search]. *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work and Social Computing*, 1379-1392. <https://doi.org/10.1145/2675133.2675215>
- Steinmacher, I., Pinto, G., Igor, W., & Gerosa, M. A. (2020). Software livre. En *Computação e Sociedade - Volume 3: A tecnologia* (pp. 47-65). EdUFMT Digital. <https://sol.sbc.org.br/livros/index.php/sbc/catalog/book/75>
- Steinmacher, I., Treude, C., & Gerosa, M. A. (2019). Let Me In: Guidelines for the Successful Onboarding of Newcomers to Open Source Projects [GS Search]. *IEEE Software*, 36(4), 41-49. <https://doi.org/10.1109/MS.2018.110162131>
- Tonhão, S., Silva, W., Colanzi, T., & Steinmacher, I. (2022). Uma plataforma gamificada de desafios baseados em worked examples extraídos de projetos de Software Livre para o ensino de Engenharia de Software [GS Search]. *Anais Estendidos do XVII Simpósio Brasileiro de Sistemas Colaborativos*, 33-38. https://doi.org/10.5753/sbsc_estendido.2022.19331
- Xing, G. (2010). Teaching Software Engineering Using Open Source Software [GS Search]. *Proceedings of the 48th Annual Southeast Regional Conference*. <https://doi.org/10.1145/1900008.1900085>
- Zorzo, A. F., Nunes, D., Matos, E. S., Steinmacher, I., Leite, J. C., Araujo, R., Correia, R. C. M., & Martins, S. (2017). *Referenciais de Formação para os Cursos de Graduação em Computação*. SBC. <https://www.sbc.org.br/documentos-da-sbc/send/127-educacao/1155-referenciais-de-formacao-para-cursos-de-graduacao-em-computacao-outubro-2017>