

Um Relato de Experiência sobre a Oferta da Disciplina Desenvolvimento de Software Livre para os Cursos de Ciência da Computação e Sistemas de Informação

Paulo Afonso Parreira Júnior
pauloa.junior@ufla.br
Universidade Federal de Lavras
(UFLA)

Júlio César Alves
juliocesar.alves@ufla.br
Universidade Federal de Lavras
(UFLA)

André Pimenta Freire
apfreire@ufla.br
Universidade Federal de Lavras
(UFLA)

Heitor Augustus Xavier Costa
heitor@ufla.br
Universidade Federal de Lavras
(UFLA)

Igor Scaliante Wiese
igor.wiese@gmail.com
Universidade Tecnológica Federal do
Paraná (UTFPR) - Campus Campo
Mourão

RESUMO

Além de ser uma excelente oportunidade de aprendizado, a participação em projetos de software livre pode ser um diferencial na carreira de um(a) profissional, pois diversas empresas têm valorizado e buscado esse tipo de experiência nos currículos de futuros(as) colaboradores(as). Assim, torna-se importante que estudantes de Computação tenham contato com o universo do software livre, a fim de conhecerem suas potencialidades e desafios, bem como terem a oportunidade de lidar com problemas reais, vivenciados diariamente na indústria de software. Uma das formas de proporcionar esse contato é por meio da oferta de disciplinas com esta temática. Este trabalho tem como objetivo apresentar o relato de experiência sobre a oferta da disciplina “Desenvolvimento de Software Livre” para os cursos de Ciência da Computação e Sistemas de Informação. Como principais destaques deste relato, têm-se que: (i) não houve desistências ao longo do semestre; (ii) o percentual de aprovação foi de 100%; (iii) a partir das respostas dadas em um questionário respondido por 20 estudantes, percebeu-se que 19 (95%) deles(as) recomendariam fortemente a disciplina a outras pessoas e 16 (80%) se sentiram mais confiantes e mais preparados(as) para contribuir com projetos de software livre, após terem cursado a disciplina; (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.

CCS CONCEPTS

• **Social and professional topics** → **Information technology education**; Computing education.

PALAVRAS-CHAVE

Educação em Computação, Software Livre, Relato de Experiência

Fica permitido ao(s) autor(es) ou a terceiros a reprodução ou distribuição, em parte ou no todo, do material extraído dessa obra, de forma verbatim, adaptada ou remixada, bem como a criação ou produção a partir do conteúdo dessa obra, para fins não comerciais, desde que sejam atribuídos os devidos créditos à criação original, sob os termos da licença CC BY-NC 4.0.

EduComp'23, Abril 24-29, 2023, Recife, Pernambuco, Brasil (On-line)

© 2023 Copyright mantido pelo(s) autor(es). Direitos de publicação licenciados à Sociedade Brasileira de Computação (SBC).

1 INTRODUÇÃO

É sabido que o software, um dos principais componentes de qualquer sistema computacional, está sob a proteção da Lei dos Direitos Autorais [8], o que garante ao seu autor o direito exclusivo de exploração de sua obra. Produtos de software disponibilizados sob licenças que impõem restrições quanto ao seu uso, distribuição ou acesso ao código-fonte são conhecidos como *software restrito* [24, 27]. Contudo, essa não é a única forma de se licenciar software. O conceito de *software livre*, por exemplo, propõe 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 [24, 27].

Segundo Steinmacher et al. [13], 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, projetos como o do sistema operacional Linux mostraram que esse modelo é viável e que, muitas vezes, pode gerar resultados melhores do que os de outros projetos de software proprietário [13]. Diante disso, várias empresas, universidades e organizações públicas passaram a financiar projetos de software livre [13]. A empresa Google, por exemplo, por meio do seu programa *Google Summer of Code*, já financiou o desenvolvimento de mais de 680 projetos de software livre [16].

Além de ser uma excelente oportunidade de aprendizado, a participação em projetos de software livre pode ser um diferencial na carreira de um(a) profissional, pois diversas empresas têm valorizado e buscado esse tipo de experiência nos currículos e portfólios de futuros(as) colaboradores(as) [13]. Neste contexto, torna-se importante que estudantes de Computação tenham contato com o universo do software livre, a fim de conhecerem suas potencialidades e desafios, bem como terem a oportunidade de lidar com problemas reais, vivenciados diariamente na indústria de software [3, 17, 19, 21].

Uma das formas de se proporcionar esse contato é por meio da oferta de disciplinas sobre desenvolvimento de software livre. Apesar de ser relativamente fácil encontrar ementas, planos de curso e até mesmo notas de aula publicamente disponíveis na Internet

[7, 18], 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?

Neste sentido, este trabalho tem como objetivo cobrir uma lacuna da literatura de Educação em Computação no Brasil, ao apresentar 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 de Lavras. Como principais destaques deste relato, têm-se que: (i) a disciplina contou com 30 estudantes matriculados(as), 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 todas as notas finais acima de 85 (oitenta e cinco) pontos; (iii) a partir das respostas dadas em um questionário respondido por 20 estudantes da disciplina, percebeu-se que 19 (95%) deles(as) recomendariam fortemente a disciplina a outras pessoas e 16 (80%) se sentiram mais confiantes e preparados(as) para contribuir com projetos de software livre, após terem cursado a disciplina; (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 da oferta da disciplina “Desenvolvimento de Software Livre”, apontando os resultados obtidos e as lições aprendidas; a Seção 4 apresenta e discute os resultados de uma avaliação feita 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 (*SB-COpenLib*)¹, 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*”). Também foi realizada uma busca no *Google Scholar*², 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 [4, 5, 9, 26]; 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” [30, 31], “*Design*” [6], “*Teste*” [10] e “*Evolução de Software*” [23]. 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 [1, 11, 28] no Ensino Superior, e sobre a utilização de ferramentas educacionais de apoio ao ensino-aprendizado de algumas disciplinas [22]. 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.

3 RELATO DE EXPERIÊNCIA

A disciplina, cuja oferta é relatada 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 vagas³. O restante desta seção apresenta o “Planejamento da Disciplina” (Subseção 3.1) e sua “Execução e Resultados” (Subseção 3.2), considerando a oferta do primeiro semestre letivo do ano 2022.

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) [18] e pelo Centro de Informática da Universidade Federal de Pernambuco (CIn/UFPE) [7]. É importante ressaltar que o IME/USP teve grande contribuição na história do movimento do software livre no Brasil, tendo sido, possivelmente, a

¹<https://sol.sbc.org.br>

²<https://scholar.google.com.br/>

³Os(as) estudantes podem se matricular na disciplina a partir do 4º Módulo (segundo ano) do curso.

primeira instituição de ensino brasileira a ter uma versão do Linux instalada em seus laboratórios de informática [27]. 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 na oferta da disciplina⁴ relatada neste trabalho. Os principais tópicos do conteúdo programático da disciplina são apresentados na Tabela 1 e o seu detalhamento é feito ao longo do texto desta subseção.

Tabela 1: Conteúdo programático da disciplina “Desenvolvimento de Software Livre” - 2022/1

Id	Tópico	Nota
01	Leitura e discussão sobre o artigo “A Catedral e o Bazar”, de Eric Raymond	5 pontos
02	Apresentação e discussão sobre o documentário “Revolution OS”	5 pontos
03	Conceitos e história do Software Livre	5 pontos
04	Licenciamento de software	5 pontos
05	Palestra convidada	
	Versionamento de software com Git e Github - Parte 1 (conceitos de <i>clone</i> , <i>commit</i> , <i>push</i> , etc)	5 pontos
06	Versionamento de software com Git e Github - Parte 2 (conceitos de <i>fork</i> , <i>branch</i> , <i>pull request</i> , etc)	5 pontos
07	Projeto 1: repositório sobre personalidades envolvidas com Software Livre	25 pontos
08	Projeto 2: contribuições para projetos de software livre reais	45 pontos
09		

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 [25], discute as vantagens técnicas do software livre e aborda os mecanismos de funcionamento do desenvolvimento descentralizado [27]. Este artigo é considerado um dos fatos mais significativos para o crescimento e o amadurecimento do movimento do software livre na década de 1990 [13, 27]. 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 [20, 27]. 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 Ambiente Virtual de Aprendizagem (AVA) da Instituição. Para guiar tal discussão,

⁴O plano de curso completo pode ser obtido em: <https://encurtador.com.br/drvTY>

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 fregueses; 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 seguintes 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 *Moodle*⁵, 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 teve um caráter avaliativo e representou 5% da nota final da disciplina, conforme pode ser visto na terceira coluna da Tabela 1.

02 - Apresentação e discussão sobre o documentário “Revolution OS”. O documentário “Revolution OS” [20], 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% 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*⁶.

03 - Conceitos e história do Software Livre e 04 - Licenciamento de software. Esses dois tópicos do conteúdo programático

⁵<https://moodle.org/>

⁶<https://github.com/>

foram abordados por meio aulas expositivas, com exercícios de fixação a serem realizados via AVA. Para o tópico “Conceitos e história do Software Livre”⁷ foi utilizado, principalmente, o capítulo da Jornada de Atualização em Informática (JAI) de 2011, intitulado “Software Livre e Propriedade Intelectual: Aspectos Jurídicos, Licenças e Modelos de Negócio” [27]. Quanto ao tópico “Licenciamento de software”⁸, o livro “101 - Open Source Licensing: From Concepts to Practical Implications” [24] foi a principal referência utilizada.

Ambos os tópicos foram acompanhados por exercícios avaliativos, totalizando 10% 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 discursivas⁹ 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. Para este conteúdo da disciplina, não houve atividade avaliativa.

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 aula¹⁰, comandos mais elementares, tais como *git clone*, *git commit* e *git push*, foram apresentados, enquanto que na segunda aula¹¹, 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 acompanhados de exercícios práticos de fixação, totalizando 10% da nota da disciplina. Para estes exercícios, também foi utilizada

a ferramenta “Questionário”, do Moodle. Um exemplo de questão proposta foi a seguinte:

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).

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 - Projeto 1: repositório sobre personalidades envolvidas com Software Livre. De acordo com Steinmacher et al. [13], 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 [13]. O objetivo dos dois últimos tópicos do conteúdo programático da disciplina (tópicos 08 e 09) 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”¹². 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 primeira oferta da disciplina, *Richard Matthew Stallman* foi escolhido como personalidade a ser apresentada.

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

⁷Slides disponíveis em: <https://encurtador.com.br/oyBKO>

⁸Slides disponíveis em: <https://encurtador.com.br/nzJLT>

⁹O banco de questões, no formato Moodle XML, encontra-se disponível em: <https://encurtador.com.br/fDQSY>

¹⁰Slides disponíveis em: <https://encurtador.com.br/cms29>

¹¹Slides disponíveis em: <https://encurtador.com.br/wTWY6>

¹²Disponível em: <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”

projeto?”, a qual especificava as regras para contribuição a serem seguidas pelas equipes (ver Figura 1).

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 uma *Pull Request* (PR)¹³ 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). Como pode ser visto na Tabela 1, o “Projeto 1” representava 25% da nota final da disciplina.

09 - 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) uma contribuição de código, que poderia ser, por exemplo, a

¹³ *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 [15]

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.

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 [13]. 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 8 (oito) projetos aptos para contribuição, 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 (julioce-sar.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**¹⁴: 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**¹⁵: um repositório com sugestões de projetos para contribuição, classificados por linguagem de programação.
- **Encontrando maneiras de contribuir com o código aberto no GitHub**¹⁶. Tutorial do *GitHub* sobre como contribuir para projetos de software livre relevantes.

¹⁴ <https://www.codetriage.com/>

¹⁵ <https://github.com/MunGell/awesome-for-beginners>

¹⁶ <https://docs.github.com/pt/get-started/exploring-projects-on-github/finding-ways-to-contribute-to-open-source-on-github>

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. [13].

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. Por fim, na quarta semana, cada equipe deveria realizar uma apresentação (com slides) sobre as contribuições realizadas. As apresentações teriam entre 8 e 15 minutos 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 (20 pontos). Como pode ser visto na Tabela 1, o “Projeto” representou 45% da nota final da disciplina.

3.2 Execução e Resultados

Pode-se dizer que a primeira oferta 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, ou seja, não houve desistências ao longo do semestre. Quanto ao percentual de aprovação, este foi de 100%, com todas as notas finais acima de 85 (oitenta e cinco) 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 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, Twitter, entre outras, têm promovido o desenvolvimento e uso de software livre; (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. Cabe ressaltar ainda que, além da nota da disciplina ter sido distribuída

Re: Discussão sobre o artigo "A Catedral e o Bazar", de Eric S. Raymond
por [nome] - 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. Concorro 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 [nome] - 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

ao longo de 9 (nove) atividades, houve ainda 2 (dois) encontros para recuperação de atividades avaliativas, um antes da “Palestra convidada” e outro antes do “Projeto 1”. Nestas ocasiões, estudantes que tivessem deixado de fazer alguma das atividades avaliativas de 01 à 07 (ver Tabela 1) poderiam realizá-la, a fim de recuperar sua nota.

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 casos de estudantes que demonstraram maior engajamento, ao ler e comentar mais de uma postagem dos(as) colegas. 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 e sobre licenciamento de software (atividades 03 e 04), 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), 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!” e a mesma foi realizada de forma *online*, pois o pesquisador era de outra instituição. 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 palestra foi muito bem recebida pelo público, prova disto é que a mesma iniciou-se às 21h e às 22:40h ainda havia mais de 40 participantes *online*.

A respeito das atividades sobre *Git* e *Github* (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, isso foi notado apenas durante a realização dos trabalhos práticos. Desta forma, pretende-se investigar melhor o conhecimento prévio dos(as) estudantes já nas primeiras semanas de aulas, a fim de preparar um material mais adequado para o nível de conhecimento médio da turma. Uma forma de fazer isso é pedir 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 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. 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. Mesmo assim, para a próxima oferta, pretende-se aprimorar o material sobre *Git* e *Github*, a fim de elucidar melhor os conceitos com os quais os(as) estudantes tiveram dificuldade.

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, uma equipe propôs uma *issue* para incluir detalhes sobre os prêmios e reconhecimentos que a personalidade do mundo 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 próxima oferta da disciplina, pretende-se etiquetar as *issues* 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 optou por escolher um(a) participante para fazer um *fork*¹⁷ do projeto para sua conta e adicionar os(as) demais como colaboradores(as) do

¹⁷Um *fork* é uma cópia de um repositório; fazer o *fork* de um repositório permite ao(a) desenvolvedor(a) realizar experiências sem comprometer o projeto original [14]

[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](#)
- [Album de fotos](#)

Figura 4: Página sobre Richard M. Stallman

[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

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

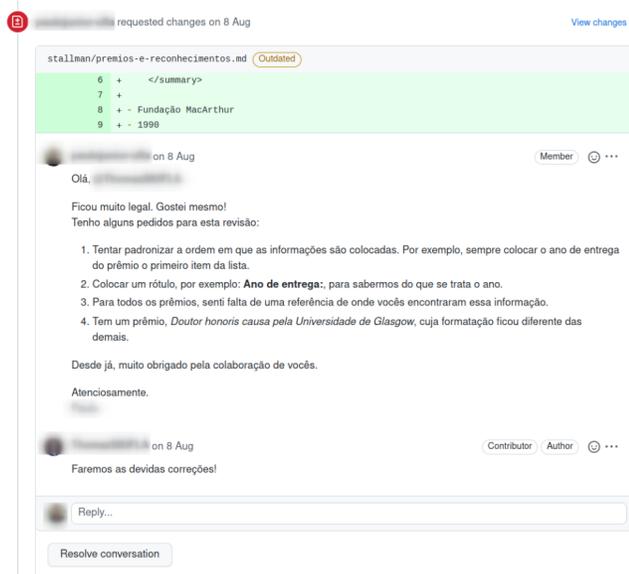


Figura 6: Solicitação de mudança feita pelo docente

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 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 “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).

Uma dificuldade percebida ocorreu com uma equipe, a qual estava trabalhando na *issue* de um projeto, cujo mantenedor era o docente da disciplina. A equipe esqueceu de declarar interesse na realização da *issue*, conforme explicado pelo docente e treinado no

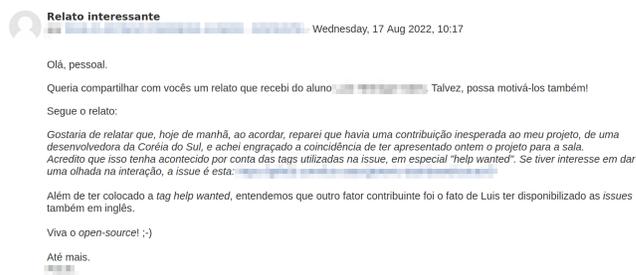


Figura 7: Contribuição internacional no repositório de um estudante da disciplina

“Projeto 1”, e começou a trabalhar na mesma. Outro estudante, de uma equipe diferente, demonstrou interesse na *issue* e o mantenedor permitiu que ele trabalhasse, sem saber que havia outra equipe trabalhando na mesma *issue*. Quando a primeira equipe percebeu o deslize e foi ao repositório do projeto, percebeu-se a mensagem do colega. A equipe 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 do colega que demonstrou publicamente interesse pela *issue*. A equipe, então, decidiu abandonar o trabalho com a *issue* em questão e partir para outra. Esse ocorrido foi interessante, pois demonstrou que a equipe não estava interessada apenas 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.

4 AVALIAÇÃO DA DISCIPLINA

Esta seção relata os resultados de uma avaliação feita com os(as) estudantes da disciplina “Desenvolvimento de Software Livre”, em sua primeira oferta. A avaliação foi realizada por meio de um questionário eletrônico¹⁸, 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 (≈ 67%). 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*

¹⁸Disponível em: <https://encurtador.com.br/EH01>

hora, entre 1 e 2 horas ou mais do que 2 horas. Como resultado, 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% gastava até 1 hora e 5% até trinta minutos. Ou seja 95% dos(as) respondentes admitiu 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.

Tabela 2: Conhecimento sobre alguns tópicos, ANTES da disciplina

Tópico	MB	B	R	A	MA
Conceitos e história do software livre	8	8	2	1	1
Licenciamento de software	13	3	4	0	0
Modelos de negócio envolvendo software livre	13	5	2	0	0
Processo de contribuição para projetos de software livre	7	4	6	2	1
Conceitos de <i>Git</i> e <i>GitHub</i> (<i>commit</i> , <i>pull request</i> , <i>branches</i>)	0	1	5	5	9

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(as) 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.

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

Tabela 3: Conhecimento sobre alguns tópicos, DEPOIS da disciplina

Tópico	MB	B	R	A	MA
Conceitos e história do software livre	0	0	1	11	8
Licenciamento de software	0	0	3	11	6
Modelos de negócio envolvendo software livre	0	0	4	13	3
Processo de contribuição para projetos de software livre	0	0	1	10	9
Conceitos de <i>Git</i> e <i>GitHub</i> (<i>commit</i> , <i>pull request</i> , <i>branches</i>)	0	0	1	3	16

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.

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 4 apresenta as afirmações feitas, juntamente com as respostas dadas pelos(as) estudantes.

Tabela 4: 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	17
O nível de profundidade com que os assuntos foram abordados na disciplina foi adequado	0	0	1	4	15
A metodologia de avaliação proposta para a disciplina foi adequada	0	0	0	3	17
Eu recomendaria esta disciplina para outras pessoas	0	0	0	1	19
Após ter cursado a disciplina, eu me sinto mais confiante e preparado(a) para contribuir com projetos de software livre	0	0	1	3	16

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.

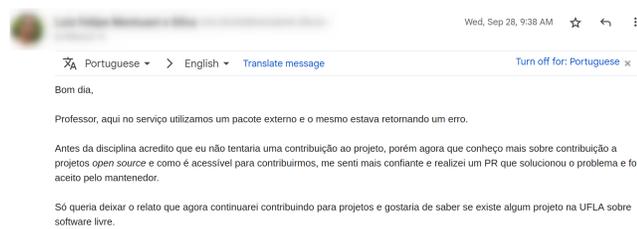


Figura 8: Relato de um estudante sobre sua contribuição para um projeto de software livre real

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 8. 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. 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. Apesar de ter havido 2 (dois) estudantes que criaram seus próprios repositórios para disponibilizar aos(as) colegas, no “Projeto 2”, isso foi feito por conta própria. 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 a próxima oferta da disciplina, pretende-se alterar a descrição do “Projeto 1”, a fim de que seja aceita, como uma opção de contribuição, a criação de projetos de software livre. A razão para se alterar a descrição do “Projeto 1” e não a do “Projeto 2” é que, ao criar o seu próprio projeto de software livre, a equipe estará trabalhando em um “ambiente controlado”, i.e., que interage

apenas com o docente da disciplina. Assim, a alteração foi realizada no “Projeto 1”, para que não haja uma equipe que tenha trabalhado apenas neste “ambiente controlado”, deixando de perceber os desafios e as potencialidades existentes ao contribuir com projetos de fora do âmbito da disciplina. Além disso, pretende-se incluir um tópico a mais na disciplina, relacionado à “Construção de projetos de software livre”, por meio do qual pretende-se abordar 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. Um bom material didático para abordar esse tópico é o livro “Producing Open Source Software How to Run a Successful Free Software Project” [12].

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. [13], contribuir para um projeto de software livre é muito 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. [29] e Balili et al. [2] 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. Contudo, ao longo da execução da disciplina, foi possível perceber alguns pontos que precisam ser aprimorados. A partir deles, decidiu-se, como trabalho futuro, aperfeiçoar o plano de curso da disciplina para (i) aprofundar o ensino dos conceitos de *Git* e *Github* e de modelos de negócio com software livre; (ii) adicionar um novo tópico ao conteúdo programático da disciplina, com o qual pretende-se abordar o processo de construção de um projeto de software livre; e (iii) aceitar, no “Projeto 1”, contribuições de estudantes que pretendam criar seus próprios projetos de software livre.

A disciplina “Desenvolvimento de Software Livre” foi ofertada novamente no semestre 2022/2 e, antes mesmo de o período de matrícula se encerrar, já contava com 25 estudantes matriculados(as). Ao final desta nova oferta da disciplina, pretende-se atualizar este relato, apresentando os impactos obtidos a partir das modificações realizadas.

REFERÊNCIAS

- [1] Andrea Almeida, Carlos Ian Melo, Myrcea Harvey, Marcos Vinicius Lima, and Pedro Jonatas Chaves. 2019. Metodologias Ativas à Luz de Comenius: uma Experiência na Pós-Graduação. In *Anais do IV Congresso sobre Tecnologias na Educação* (Recife). SBC, Porto Alegre, RS, Brasil, 60–68.
- [2] Sogol Balali, Igor Steinmacher, Umayal Annamalai, Anita Sarma, and Marco Aurelio Gerosa. 2018. Newcomers' Barriers. . . Is That All? An Analysis of Mentors' and Newcomers' Barriers in OSS Projects. *Computer Supported Cooperative Work (CSCW)* 27, 3, 679–714.
- [3] Grant Braught, John McCormick, James Bowring, Quinn Burke, Barbara Cutler, David Goldschmidt, Mukkai Krishnamoorthy, Wesley Turner, Steven Huss-Lederman, Bonnie Mackellar, et al. 2018. A multi-institutional perspective on H/FOSS projects in the computing curriculum. *ACM Transactions on Computing Education (TOCE)* 18, 2, 1–31.
- [4] Ramon Camargo, Carlos Ribeiro, Fabio Sordi Junior, Paulo Anastácio, and José Merlin. 2019. Utilização de Pygame para Ensino e Aprendizado de Orientação a Objetos. In *Anais do XVI Congresso Latino-Americano de Software Livre e Tecnologias Abertas* (Foz do Iguaçu). SBC, Porto Alegre, RS, Brasil, 164–167.
- [5] Alexander Canez and Carlos Severo. 2016. Implicações do software livre no ensino e aprendizagem: um estudo de caso no ensino médio politécnico. In *Anais do XXII Workshop de Informática na Escola* (Uberlândia). SBC, Porto Alegre, RS, Brasil, 476–484.
- [6] D. Carrington and S.-K. Kim. 2003. Teaching software design with open source software. In *33rd Annual Frontiers in Education, 2003. FIE 2003.*, Vol. 3. IEEE, Westminster, CO, USA, 51C–9.
- [7] Fernando Castor. [n.d.]. *Desenvolvimento de Software de Código Aberto/Livre*. Retrieved 14 out 2022 from <https://sites.google.com/a/cin.ufpe.br/floss/>
- [8] Presidência da República. 1998. *Lei Nº 9.610, de 19 de fevereiro de 1998*. Retrieved 14 out 2022 from http://www.planalto.gov.br/ccivil_03/leis/19610.htm
- [9] Amanda Maria de Oliveira, Adriana Virginia de Azevedo, Gelly Mota, Roberia Lourenço, Felipe Gilberto Dantas, and Dennys Maia. 2019. Software Educativo Encantando Cordel. In *Anais do IV Congresso sobre Tecnologias na Educação* (Recife). SBC, Porto Alegre, RS, Brasil, 610–616.
- [10] Lin Deng, Josh Dehlinger, and Suranjan Chakraborty. 2020. Teaching Software Testing with Free and Open Source Software. In *2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. IEEE, Porto, Portugal, 412–418.
- [11] Luciana Mara Diniz and Fischer Ferreira. 2018. Kahoot!: um relato de experiência no contexto acadêmico. In *Anais do XXVI Workshop sobre Educação em Computação* (Natal). SBC, Porto Alegre, RS, Brasil, 10.
- [12] Karl Fogel. 2018. *Producing Open Source Software How to Run a Successful Free Software Project*. <http://producingoss.com>
- [13] Igor Steinmacher, Gustavo Pinto, Igor Wiese, Marco Aurelio Gerosa. 2020. Software livre. In *Computação e Sociedade - Volume 3: A tecnologia*. EdUFMT Digital, Cuiabá-MT, 47–65.
- [14] Github.com. [n.d.]. *Bifurcar um repositório*. Retrieved 17 out 2022 from <https://docs.github.com/pt/get-started/quickstart/fork-a-repo>
- [15] Github.com. [n.d.]. *Sobre solicitação de pull*. Retrieved 17 out 2022 from <https://docs.github.com/pt/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-pull-requests>
- [16] Google. [n.d.]. *About Google Open Source*. Retrieved 14 out 2022 from <https://opensource.google/about>
- [17] Brad Hollister and Juan Leon. 2022. Open-Source Development in Undergraduate Software Engineering Education. *International Journal on E-Learning* 21, 2, 139–161.
- [18] Fabio Kon. [n.d.]. *MAC0470 - Desenvolvimento de Software Livre*. Retrieved 14 out 2022 from <https://bcc.ime.usp.br/principal/catalogo2017/disciplinas/MAC0470.html>
- [19] Avelino F. Zorzo; Daltro Nunes; Eivaldo S. Matos; Igor Steinmacher; Jair C. Leite; Renata Araujo; Ronaldo C. M. Correia; Simone Martins. 2017. *Referenciais de Formação para os Cursos de Graduação em Computação*. SBC. Retrieved 14 out 2022 from <https://www.sbc.org.br/documentos-da-sbc/send/127-educacao/1155-referenciais-de-formacao-para-cursos-de-graduacao-em-computacao-outubro-2017>
- [20] J.T.S. Moore. [n.d.]. *Revolution OS*. Retrieved 17 out 2022 from <https://www.youtube.com/watch?v=Z3f-M43DiD4>
- [21] Debora Maria Coelho Nascimento, Christina von Flach Garcia Chavez, and Roberto Almeida Bittencourt. 2019. Does FLOSS in Software Engineering Education Narrow the Theory-Practice Gap? A Study Grounded on Students' Perception. In *IFIP International Conference on Open Source Systems*. Springer, 153–164.
- [22] Ana Perin, Polianna Paim, and Natasha Valentim. 2021. Experiência sobre o uso de ferramentas de apoio à pesquisa experimental em uma disciplina de IHC. In *Anais do XXXII Simpósio Brasileiro de Informática na Educação* (Online). SBC, Porto Alegre, RS, Brasil, 1297–1307.
- [23] Maksym Petrenko, Denys Poshyvanyk, Vaclav Rajlich, and Joseph Buchta. 2007. Teaching Software Evolution in Open Source. *Computer* 40, 11, 25–31.
- [24] Gustavo Pinto. 2021. *Open Source Licensing 101 - From Concepts to Practical Implications* (1ª ed.). Self-published.
- [25] Eric S. Raymond. [n.d.]. *The Cathedral and the Bazaar*. Retrieved 17 out 2022 from <http://catb.org/esr/writings/cathedral-bazaar>
- [26] Gabriela Ribeiro, Julilma Reis, Rodrigo Franco, Maria Felix, Albertina Guedes, and Ubirajara Nogueira. 2017. Software livre como ferramenta no processo do ensino aprendido: uma experiência com turmas do EJA. In *Anais do XXIII Workshop de Informática na Escola* (Recife). SBC, Porto Alegre, RS, Brasil, 974–982.
- [27] Fabio Kon; Nelson Lago; Paulo Meirelles; Vanessa Sabino. 2011. Software Livre e Propriedade Intelectual: Aspectos Jurídicos, Licenças e Modelos de Negócio. In *Jornada de Atualização em Informática (JAI) do Congresso da Sociedade Brasileira de Computação*. Editora PUC-Rio, 59–107.
- [28] Eduardo Henrique Silva. 2018. A Gamificação na Educação Superior a Distância: o Programa Desafio Nota Máxima. In *Anais do II Seminário de Educação a Distância da Região Centro-Oeste* (Campo Grande/MS). SBC, Porto Alegre, RS, Brasil.
- [29] Igor Steinmacher, Tayana Conte, Marco Aurélio Gerosa, and David Redmiles. 2015. Social Barriers Faced by Newcomers Placing Their First Contribution in Open Source Software Projects. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work and Social Computing* (Vancouver, BC, Canada) (CSCW '15). Association for Computing Machinery, New York, NY, USA, 1379–1392.
- [30] Simone Tonhão, Williamson Silva, Thelma Colanzi, and Igor Steinmacher. 2022. Uma plataforma gamificada de desafios baseados em worked examples extraídos de projetos de Software Livre para o ensino de Engenharia de Software. In *Anais Estendidos do XVII Simpósio Brasileiro de Sistemas Colaborativos* (Online). SBC, Porto Alegre, RS, Brasil, 33–38.
- [31] Guangming Xing. 2010. Teaching Software Engineering Using Open Source Software. In *Proceedings of the 48th Annual Southeast Regional Conference* (Oxford, Mississippi) (ACM SE '10). Association for Computing Machinery, New York, NY, USA, Article 57, 3 pages.