

Universidade de Brasília – UnB
Faculdade UnB Gama – FGA
Engenharia de Software

**Mentoria como ferramenta pedagógica
auxiliando professores e alunos de graduação
em curso de Engenharia de Software**

Autor: Omar Faria dos Santos Júnior
Orientador: Doutora Carla Rocha

Brasília, DF
2019



Omar Faria dos Santos Júnior

**Mentoria como ferramenta pedagógica auxiliando
professores e alunos de graduação em curso de
Engenharia de Software**

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Doutora Carla Rocha

Brasília, DF

2019

Omar Faria dos Santos Júnior

Mentoria como ferramenta pedagógica auxiliando professores e alunos de graduação em curso de Engenharia de Software/ Omar Faria dos Santos Júnior. – Brasília, DF, 2019-

56 p. : il. (algumas color.) ; 30 cm.

Orientador: Doutora Carla Rocha

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB
Faculdade UnB Gama – FGA , 2019.

1. Mentoria. 2. Ensino. I. Doutora Carla Rocha. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Mentoria como ferramenta pedagógica auxiliando professores e alunos de graduação em curso de Engenharia de Software

CDU 02:141:005.6

Omar Faria dos Santos Júnior

Mentoria como ferramenta pedagógica auxiliando professores e alunos de graduação em curso de Engenharia de Software

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 27 de março de 2019 – Data da aprovação do trabalho:

Doutora Carla Rocha
Orientadora

Doutor Fábio Macedo Mendes
Convidado 1

Professora Bruna Nayara
Convidado 2

Brasília, DF
2019

Este trabalho é dedicado àqueles que sempre me apoiaram e me permitiram ter as condições necessárias para tudo que já trilhei em minha vida.

Agradecimentos

Agradeço, primeiramente a Deus, pois se não fosse por Ele nada seria possível. Em segundo lugar, agradeço a Ele novamente por ter me presenteado com o privilégio de ser parte da família maravilhosa e abençoada à qual pertença. Pessoas que em todos os momentos, com amor, carinho e conselhos me proporcionaram a tranquilidade necessária para a realização do meu trabalho. Cito aqui, meu pai, amigo, exemplo e herói, Omar Faria dos Santos, o qual tive o privilégio até mesmo de herdar o nome. Nome esse que não poderia trazer maior carga de responsabilidade em aprender, praticar e exercer, da melhor maneira possível, os ofícios aos quais me foram concedidos em todas as áreas de minha vida. À minha mãe, amiga, conselheira, intercessora, exemplo e rainha, Adriana Martins Faria dos Santos, que sempre com muito amor me cobriu de afeto, confiança e ensinamentos mais que especiais os quais carregarei por toda a minha vida. Pelo direcionamento desde sempre em tomar as atitudes mais responsáveis e de gratidão àqueles que fazem parte de minha vida e pelos puxões de orelha que me fizeram permanecer firme em busca da conclusão dessa etapa. À minha irmã, amiga, conselheira, princesa, Victória Martins Faria dos Santos, que sempre incentivou-me e me deu a responsabilidade de ser um espelho para sua vida e, com isso, fez-me cuidar cada dia mais de meus passos e pensar minhas atitudes.

Não posso deixar de agradecer à uma pessoa que, desde que conheci tem sido parte essencial no desempenho de minhas funções. À minha noiva, presente de Deus em minha vida, amiga, conselheira, fã, auxiliadora, exemplo, princesa, Suyã Helora da Silva Oliveira, que com todo amor me apoiou, incentivou, auxiliou no decorrer de todo o curso. Por me ter feito companhia em horas de realização do presente trabalho e demandado esforços para a conclusão do mesmo. Pelos sermões que levaram-me a levantar a cabeça e continuar. à toda a sua família, pais, irmãs, cunhado e sobrinhos que também me acompanharam por todo esse percurso.

À minha orientadora, Doutora Carla Rocha, que com toda maestria direcionou-me para atingir os resultados aguardados. Pela paciência e conselhos que fizeram-me continuar e não desistir. Um exemplo de orientadora, professora, líder e amiga ao qual tive o privilégio de ter presente em minha história. E ao Doutor Paulo Roberto Miranda Meirelles que me auxiliou com tamanha sabedoria durante boa parte do meu curso e que, principalmente com seu exemplo, proporcionou-me experiências e oportunidades que agregaram extremo valor em minha graduação e vida. E a todos os doutores e mestres que estiveram presente durante a minha graduação contribuindo sempre com imensa sabedoria e apreço pelo ensino dos conteúdos mais relevantes para a formação de um profissional de Engenharia de Software.

Aos meus amigos que, direta ou indiretamente, fazem parte da minha história e me impulsionam a alcançar aquilo que Deus tem para mim. São estes de extrema importância pois facilitam e podem ser comparados a degraus que me possibilitam superar as dificuldades e adversidades e alcançar o topo. Cito aqui um grande amigo/irmão, Ebenezer Andrade da Silva que, com seu conhecimento auxiliou-me na resolução desse trabalho e com companheirismo ímpar em dias difíceis.

Enfim, deixo aqui essas palavras que, do fundo do coração, exprimem apenas um pouco da gratidão que toma conta de mim.

Sou eternamente grato!

*“Não vos amoldeis às estruturas deste mundo,
mas transformai-vos pela renovação da mente,
a fim de distinguir qual é a vontade de Deus:
o que é bom, o que Lhe é agradável, o que é perfeito.
(Bíblia Sagrada, Romanos 12, 2)*

Resumo

Este artigo apresenta uma forma de ensino em Engenharia de Software tendo a mentoria como método de disseminação do conhecimento. Além disso, descreve o design do curso da Engenharia de Software da UnB que se utiliza desse método de ensino e de metodologias de desenvolvimento de software de amplitude complexa a partir de um autêntico formato de sala de aula. Envolve projetos do tipo FLOSS, do zero, onde os alunos participam de equipes e executam metodologias e práticas ágeis.

Apresentamos uma análise quantitativa e qualitativa de como a metodologia e as práticas ágeis são monitoradas por meio de projetos de código aberto. Também mostraremos evidências de que o uso do FLOSS é essencial para a evolução contínua do curso, uma vez que as lições aprendidas de um projeto são normalmente aplicadas na oferta de curso a seguir. Apresentamos os resultados de uma pesquisa aplicada onde 90 alunos contribuíram apresentando suas experiências a respeito do curso.

Palavras-chaves: Mentoria. Ensino. Engenharia de Software.

Abstract

This article presents a form of teaching in Software Engineering with mentoring as method of knowledge dissemination. In addition, it describes the design of the course of Software Engineering of UnB that uses this method of teaching and methodologies of software development of complex amplitude from an authentic classroom format. It involves FLOSS projects from scratch, where students participate in teams and execute agile methodologies and practices.

We present a quantitative and qualitative analysis of how methodology and agile practices are monitored through open source projects. We will also show evidence that the use of FLOSS is essential for the ongoing evolution of the course, since the lessons learned from a project are usually applied in the course offer to follow. We present the results of an applied research where 90 students contributed by presenting their experiences regarding the course.

Key-words: Mentoring. Teaching. Software Engineering.

Lista de ilustrações

Figura 1 – Modelo de Mentoria de jovens	27
Figura 2 – Abordagens de Ensino	29
Figura 3 – Faixa etária dos participantes	41
Figura 4 – Nível de conhecimento antes do curso	41
Figura 5 – Quantidade de coaches mobilizados	42
Figura 6 – Relevância do direcionamento	42
Figura 7 – Atividades de maior valor na <i>sprint</i>	43
Figura 8 – Nível de conhecimento agregado	43
Figura 9 – Importância de <i>stand ups</i>	44
Figura 10 – Importância de Revisão de Código	45
Figura 11 – Importância de trabalhar em grupo	45
Figura 12 – Importância do uso de repositório aberto	46
Figura 13 – Importância de práticas de software livre	46
Figura 14 – Importância do monitoramento da qualidade de código	47
Figura 15 – Importância de arquitetura bem definida	47
Figura 16 – Importância de ferramentas de auxílio no desenvolvimento	48
Figura 17 – Importância do documento de contribuição	49
Figura 18 – Importância de roadmap através de issues no desenvolvimento	49
Figura 19 – Notas para o aprendizado em contribuição a projetos de software livre .	50
Figura 20 – Projetos de software livre como oportunidades	50
Figura 21 – Confiança em contribuir com projetos de software livre	51
Figura 22 – Projetos de software livre como oportunidade de negócio	51
Figura 23 – Interesse em colaborar em projetos de software livre	52

Lista de tabelas

Tabela 1 – Práticas ágeis	34
-------------------------------------	----

Lista de abreviaturas e siglas

ESW	Engenharia de Software
MDS	Métodos de Desenvolvimento de Software
GPP	Gestão de Portifólios e Projetos de Software
FLOSS	Free/Libre and Open Source Software
OSS	Open-source software

Sumário

1	INTRODUÇÃO	23
1.1	Contexto	23
1.2	Questão de Pesquisa	23
1.3	Justificativa	23
1.4	Objetivos	24
1.4.1	Objetivo Geral	24
1.4.2	Objetivos Específicos	24
2	REFERENCIAL TEÓRICO	27
2.1	Mentoria	27
2.2	Ensino de ESW	28
2.2.1	Capstone Project	29
2.2.2	Jogos com LEGO	30
2.2.3	Sala de Aula Invertida	30
2.3	Ferramentas	30
2.3.1	Moodle-edX	30
2.3.2	Dropbox	31
2.3.3	Google Drive	31
2.3.4	Github e Gitlab	32
3	DISCIPLINA	33
3.0.1	Descrição	33
3.0.2	Metodologia das Disciplinas	33
3.0.3	Práticas ágeis	34
3.0.4	MDS – Práticas de desenvolvimento	35
3.0.5	GPP – práticas de gestão de projeto	35
3.0.6	Papel do mentor	35
4	METODOLOGIA	37
5	RESULTADOS E DISCUSSÕES	39
5.1	Análise qualitativa	39
5.1.1	Processo de aplicação do <i>survey</i>	40
5.1.2	Método de análise dos dados	40
5.1.3	Perguntas e Respostas do <i>Survey</i>	40
5.2	Considerações sobre a disciplina	51

6 CONSIDERAÇÕES FINAIS 53

REFERÊNCIAS 55

1 Introdução

1.1 Contexto

O desenvolvimento de software, popularmente falando, apresenta-se de forma solitária e dependente de poucos recursos para sua execução, a saber, um desenvolvedor e um computador. Porém, a complexidade e a crescente quantidade de tipologias, ferramentas e linguagens de programação tem tornado o processo de desenvolvimento de software cada vez mais profundo, de tal maneira que se faz necessário contar com uma equipe para sua execução. Concomitantemente à especialização feita pelos desenvolvedores desta área em determinados processos, ferramentas e linguagens de programação, surge uma oportunidade que necessita de uma nova demanda de conhecimento em outro processo, outra ferramenta e de uma linguagem de programação desconhecida pelos mesmos. É nesse momento que são reconhecidas naquela equipe pessoas especializadas e capazes de atender a esse novo contexto, pessoas essas que suas habilidades vão ao encontro dos conhecimentos requeridos nessa oportunidade. E que também se apropriam dessa fase para disseminar seus estudos entre a equipe.

A mentoria tem um papel fundamental nessa transmissão de conhecimento. Na Engenharia de Software (ESW) ela está presente nos processos de aprendizado, revisão de códigos, *pull request*, entre outras etapas, mesmo que anonimamente. O presente trabalho propôs-se a realizar um estudo de caso com o objetivo de investigar a respeito dos benefícios da aplicação da mentoria em curso de ESW.

1.2 Questão de Pesquisa

O presente trabalho procurará responder a seguinte questão de pesquisa: Quais os benefícios da aplicação da mentoria como uma ferramenta pedagógica auxiliando em disciplina orientada a projetos no curso de Engenharia de Software?

1.3 Justificativa

A mentoria apresenta inúmeras vantagens. Dentre elas destacam-se a otimização do tempo de aula, o envolvimento aprimorado dos alunos, o suporte a pensamentos conceituais e abstratos e o aprimoramento das interações professor-aluno e aluno-aluno (TRAINER; KALYANASUNDARAM; HERBSLEB, 2017). Dessa forma, é importante a motivação para o uso da mentoria tendo em vista o compartilhamento de conhecimento entre os envolvidos. Para isso, o estudo da importância da mentoria para o ensino de ESW

será de grande valia ao contribuir com informações e reflexões relativas ao do método no ensino para tal área.

Neste trabalho o termo em inglês *develop mastery* é compreendido como sinônimo para desenvolvimento do domínio das habilidades do aluno.

1.4 Objetivos

1.4.1 Objetivo Geral

Identificar quais os principais benefícios da aplicação da mentoria como uma ferramenta pedagógica no auxílio a professores e alunos de graduação em curso de ESW.

1.4.2 Objetivos Específicos

Para a realização desse trabalho foram considerados os seguintes objetivos específicos:

- Conceituar mentoria;
- Apresentar modelos e métodos alternativos de ensino no contexto de ensino superior em cursos de tecnologia;
- Apresentar o estudo de caso da disciplina Métodos de Desenvolvimento de Software (MDS) - Gestão de Portifólios e Projetos de Software (GPP);
- Análise qualitativa da efetividade do método, por meio de *survey*;
- Sugestões de melhoria;
- Aplicar *survey* para ex-alunos das disciplinas de GPP e MDS com o intuito de identificar os conceitos/técnicas de mentoria existentes; e,
- Conferir de maneira isolada o quanto cada técnica da mentoria é utilizada para evidenciar a potencialização do *develop mastery* dos alunos.

Para o efetivo desenvolvimento dos objetivos específicos em um corpo consistente de análise e argumentação, adota-se como processo metodológico uma abordagem objetiva e qualitativa, com base em estudo de caso no curso de Engenharia de Software da Universidade de Brasília utilizando-se da aplicação de *survey* com auxílio de questionário que permita um maior aprofundamento sobre o tema da pesquisa. Sem a pretensão de estabelecer um discurso conclusivo sobre as questões pesquisadas, busca-se analisar os conceitos chave tratados nesta dissertação, contribuindo com novas reflexões e perspectivas de estudo.

Para alcançar o seu objetivo central, esta dissertação encontra-se organizada em capítulos, sendo esta introdução o capítulo de número 1. No Capítulo 2, são apresentados os conceitos relevantes para o desenvolvimento desse trabalho sobre mentoria, metodologias de ensino de ESW e ferramentas utilizadas no ensino. No Capítulo 3, é apresentada a metodologia das disciplinas GPP/MDS do curso de ESW da Universidade de Brasília e o papel do mentor no curso. A metodologia desse trabalho é exposta no capítulo 4. O conteúdo do capítulo 5 apresenta os resultados da pesquisa de campo realizada através da aplicação do *survey* e considerações a respeito da disciplina. E, por fim, no capítulo 6, são apresentadas as considerações finais.

2 Referencial Teórico

2.1 Mentoria

O conceito de mentoria se refere a um método tradicional que tem como objetivo difundir conhecimentos e habilidades de um senior (mentor) para um indivíduo iniciante na área de conhecimento em questão (aprendiz) (AKIN; HILBUN, 2007). Para MCCARTHY(2012), a prática pode ser definida como um processo de aprendizagem compartilhada que promove crescimento e benefício mútuo, além da interação e apoio para as partes envolvidas que está se tornando cada vez mais relevante para o ingresso no mercado de trabalho e futuro sucesso profissional.

No âmbito de graduação, a mentoria é uma estratégia que facilita e propõe o progresso acadêmico, bem como o ascensão na carreira. Um mentor auxilia o seu orientando a estabelecer metas, objetivos, padrões e a desenvolver as habilidades necessárias para se ter sucesso. Esse é um processo benéfico, animador e protetor, proporcionando experiências harmonizadas e organizadas para facilitar o crescimento (GIRVES; ZEPEDA; GWATHMEY, 2005).

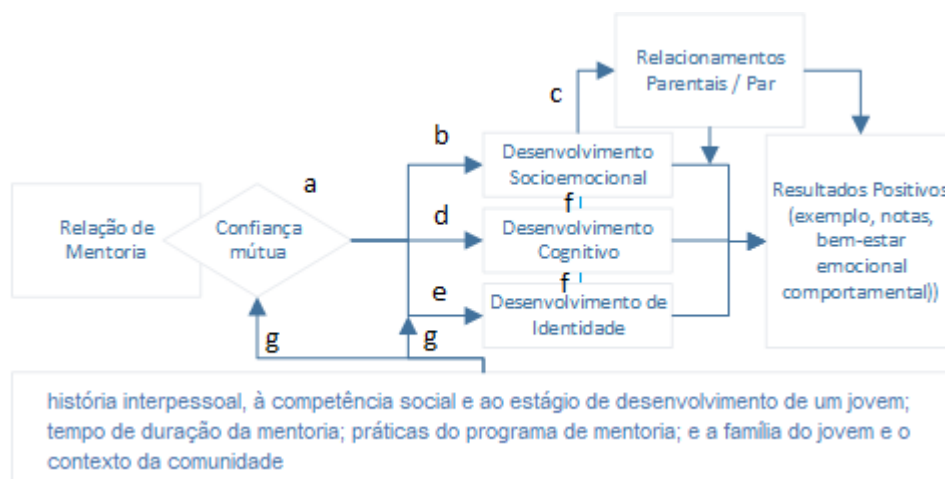


Figura 1 – Modelo de Mentoria de jovens

A figura 1, releitura com ligeiras modificações de DUBOIS et al.(2011), apresenta um modelo de mentoria de jovens. Uma conexão pessoal forte e significativa é estabelecida entre o jovem e o mentor (componente a), por exemplo, no cenário de um trabalho em conjunto para realização de tarefas orientadas para objetivos. Uma empatia interpessoal é então postulada para alimentar os processos de desenvolvimento em três áreas - socioemocional (caminho b), cognitivo (caminho d) e relacionado à identidade (caminho e) - e essas três áreas são supostamente trabalhadas em conjunto ao longo do tempo (f setas).

Experiências sociais e emocionais positivas com mentores podem tornar esse aspecto mais amplo e permitir que os jovens interajam mais efetivamente com pais e colegas (caminho c), e essas relações podem, por sua vez, mediar o efeito de ganhos no desenvolvimento socioemocional em resultados positivos. A qualidade das relações de orientação e os caminhos que a ligam, na última esfera, a resultados positivos para os jovens, podem ser condicionados por fatores relacionados à história interpessoal, à competência social e ao estágio de desenvolvimento de um jovem; tempo de duração da mentoria; práticas do programa de mentoria; e a família do jovem e o contexto da comunidade ao qual está inserido (setas g).

No modelo demonstrado na figura 1 é de vital importância que, antes de tudo, seja estabelecida uma conexão forte e significativa entre o jovem e o mentor. Esta suposição está de acordo com a pesquisa que ressalta os benefícios do equilíbrio emocional e apoio nas relações de jovens com parentes, amigos e adultos em geral (DUBOIS et al., 2011). Entretanto, essa conexão próxima, pode se dar como consequência de relações eficazes de orientação para os jovens (HAMILTON; HAMILTON, 2010). Podem haver uma admiração e o desenvolvimento de confiança para com seu mentor no contexto de trabalhar com eles em tarefas orientadas para objetivos determinados. HAMILTON; HAMILTON(1992) expressam que estabelecer a conexão emocional com um jovem como meta principal pode ser de pouco valor ou até mesmo prejudicial.

Nesse contexto existe um movimento que, como afirma SANTOS(2002), caracteriza-se ter como base o princípio do compartilhamento de conhecimento e na solidariedade exercida pela inteligência coletiva conectada na rede mundial de computadores: o software livre.

Esse movimento iniciou a partir da aversão de um integrante do MIT, Richard Stallman, pela negação ao acesso ao código fonte de um determinado projeto sendo que o mesmo havia sido desenvolvido através do conhecimento de vários programadores. Nesse momento foi criada a Free Software Foundation. Esse movimento reunia e distribuía ferramentas livres, com código-fonte aberto e acessível. Assim todos poderiam não somente usufruir das funcionalidades do programa assim como ter o acesso ao código-fonte da ferramenta em questão (SANTOS, 2002).

2.2 Ensino de ESW

No processo de ensino existem papéis principais com atribuições fundamentais para o desenvolvimento e geração de conhecimento. São eles: professor e aluno. Esses dois papéis têm funções características, o professor auxilia o aluno no seu aprendizado. Portanto, ensinar e aprender são objetivos-chave nesse processo. Porém, existem diferentes formas para que esses objetivos sejam alcançados. Duas principais metodologias de ensino são:

focado no professor e focado no aluno. A figura 2, retirada da obra de PRIKLADNICKI et al.(2009) apresenta aspectos das duas diferentes abordagens de ensino.

	Focado no professor	Focado no aluno
Papel do professor	<ul style="list-style-type: none"> • Principal fornecedor da informação • Especialista • Avaliador do rendimento 	<ul style="list-style-type: none"> • Facilitador • Fornece informação para ajudar na compreensão da informação
Clima de aprendizagem	<ul style="list-style-type: none"> • Individualista 	<ul style="list-style-type: none"> • Coletiva • Foco na coesão de grupo
Orientação	<ul style="list-style-type: none"> • Baseada na experiência e nos conhecimentos do professor 	<ul style="list-style-type: none"> • Baseada na experiência e conhecimento dos alunos
Programa de estudos	<ul style="list-style-type: none"> • Definido pelo professor 	<ul style="list-style-type: none"> • Negociado entre professor e alunos
Objetivo de ensino	<ul style="list-style-type: none"> • Definido pelo professor • Resultado padrão 	<ul style="list-style-type: none"> • Definido pelos alunos • Resultados diferentes para cada aluno
Aquisição de conhecimento	<ul style="list-style-type: none"> • Enfoque na aquisição • Foco na memorização 	<ul style="list-style-type: none"> • Enfoque na utilização e absorção de conhecimento com foco em problemas reais
Métodos de ensino	<ul style="list-style-type: none"> • Didático • Grande participação do professor 	<ul style="list-style-type: none"> • Métodos que envolvem a participação dos alunos (técnicas dinâmicas)
Foco na educação	<ul style="list-style-type: none"> • Educação individual 	<ul style="list-style-type: none"> • Educação coletiva
Avaliação	<ul style="list-style-type: none"> • Executada pelo professor • Uso tradicional de provas e notas 	<ul style="list-style-type: none"> • Os alunos também são responsáveis pela avaliação

Figura 2 – Abordagens de Ensino

O autor apresenta experiências reais de estratégias diferentes de ensino de ESW em quatro cursos e instituições diferentes usando abordagens participativas com foco no aluno. Uso de dinâmicas de grupo, *capstone project*, jogos e atividades lúdicas no aprendizado por analogia são as estratégias abordadas no estudo. No contexto desse artigo, o foco está no uso da metodologia denominada *capstone project* onde um grupo de alunos realiza o planejamento e execução de um projeto de software do início ao fim o semestre (LEBLANC et al., 2006).

2.2.1 Capstone Project

Um grupo de alunos trabalha em conjunto para desenvolver um software para atender uma necessidade real (GOOLD; HORAN, 2002). Essa abordagem permite que o aprendizado torne-se mais ativo e efetivo, além de que habilidades individuais de comunicação, trabalho em equipe e resolução de problema são aprimoradas (DAIGLE; DORAN; PARDUE, 1996). Um ponto crítico relatado por PRIKLADNICKI et al.(2009) é que o tamanho da equipe necessita de organização e do interesse de cada participante em realizar as atividades e contribuir no projeto.

2.2.2 Jogos com LEGO

Peças de LEGO são utilizadas para projetar, a partir de requisitos definidos, um produto para a retenção de conteúdos. Um dos benefícios desse método é o bom nível de familiaridade com a construção de software. Além disso, o professor acompanha de perto a atividade e percebe a dificuldade e os conteúdos que foram menos aprendidos. O desafio desse modelo foi o pouco tempo para criação do produto e a falta de seriedade de alguns alunos ao encararem essa experiência (PRIKLADNICKI et al., 2009).

2.2.3 Sala de Aula Invertida

Método pedagógico assistido por tecnologia onde visa-se exercitar a teoria e a prática. Diferente de uma aula tradicional, onde a teoria é fornecida durante as aulas presenciais e os alunos aplicam a teoria fora da sala de aula por meio de tarefas para levar para casa, em uma sala de aula invertida a teoria é fornecida antes da aula por meio de palestras gravadas *online* e materiais de apoio. Durante a sessão em sala de aula, os alunos aplicam a teoria através de atividades supervisionadas por responsáveis pela disciplina. Este conceito promove uma evolução acadêmica sobre conceitos teóricos e práticos da disciplina de maneira alternativa. Esse método tem como alguns benefícios a otimização do tempo de aula, o envolvimento aprimorado dos alunos e o desenvolvimento de maturidade dos alunos. O desafio no uso desse método é o esforço dos instrutores na preparação do material das aulas extraclasse e dificuldade em manter a motivação dos alunos (ERDOGMUS; PÉRAIRE, 2017).

2.3 Ferramentas

2.3.1 Moodle-edX

Na era da internet e da quantidade massiva de conhecimento na nuvem, a sigla MOOC define cursos online e abertos oferecidos por universidades pelo mundo. A sigla (do inglês *Massive Open Online Course*, ou em português, Curso Online Aberto e Massivo) é realizada para definir plataformas para a disseminação do conhecimento online (PEREIRA; KEPLER, 2014).

O *edX* é uma plataforma que permite a criação de MOOCs, e também oferece suporte ao gerenciamento de disciplinas presenciais. Entre os recursos disponibilizados destacam-se a funcionalidade de avaliação automática de projetos submetidos (tarefas de disciplinas), que a plataforma realiza de forma autônoma, e a parte de gerenciamento do sistema (PEREIRA; KEPLER, 2014).

Outra plataforma com o mesmo intuito é o *Moodle*. O *Moodle* é uma ferramenta que apresenta interfaces para gerenciamento e comunicação de informações que poderão

servir de conteúdo para mediar atividades, tanto em cursos do tipo presencial quanto a distância. Estas interfaces estendem o espaço para discussão dos conceitos que são aplicados nas disciplinas, possibilitando a prática colaborativa de aprendizagem (ALVES, 2009).

2.3.2 *Dropbox*

O *Dropbox* é a mais popular ferramenta de armazenamento de arquivos em nuvem (DRAGO et al., 2012). O aplicativo possibilita aos usuários sincronizar vários dispositivos através da mesma conta como também compartilhar pastas e arquivos com outros usuários.

Uma outra funcionalidade interessante é o controle de versão e atualizações que possibilita ao usuário ter o controle maior sobre o conteúdo e gerenciar o uso e atualizações de terceiros. Essa função oferece também a oportunidade de recuperação de arquivos perdidos ou excluídos daquele armazenamento (arquivos que ficam no chamado lixo eletrônico podendo ser recuperado ou excluído permanentemente) (MARTÍN; MORO,).

A plataforma fornece opções de contas gratuitas para o acesso de todos que se interessam pelo serviço. Essa opção disponibiliza 2GB de espaço para armazenamento dos arquivos. Podendo ser expandido gratuitamente para até 18GB através de indicações. Para se ter mais espaços o aplicativo tem opções de planos assinatura iniciando em 1TB e podendo ser contratado o plano de espaço indefinido até o quanto você precisar (MARTÍN; MORO,).

2.3.3 *Google Drive*

O *Google Drive* é outra plataforma de armazenamento em nuvem. Criada pela companhia *Google*, possui interface e funcionalidades muito semelhantes às do concorrente *Dropbox*. Em comparação, esse aplicativo oferece mais funcionalidades que o rival, como a integração com outros aplicativos da mesma empresa para criação, edição e compartilhamento de documentos, planilha e apresentações. São esses *Google Docs*, *Google Sheets* e *Google Slides*, respectivamente (MARTÍN; MORO,), e vários outros apps disponíveis para essa integração.

Assim como o anterior, o *Google Drive* oferece opções gratuitas e planos de assinatura. A conta gratuita oferece mais espaço de armazenamento que o outro, 15GB, e podendo chegar a 30TB no plano de maior valor (MARTÍN; MORO,).

Também contem a função de recuperação de arquivo no caso de exclusão do mesmo ou na restauração de versões anteriores de um mesmo documento (MARTÍN; MORO,).

A plataforma também pode ser acessada através dos mais populares navegadores tais como *Microsoft Internet Explorer*, *Mozilla Firefox*, *Google Chrome* e *Apple Safari*

(QUICK; CHOO, 2014).

Como plataformas de armazenamento de arquivos dos mais diversos tipos (documentos, imagens, planilhas, ...), um dos pontos-chave e mais importantes é a proteção desses dados e arquivos. Diversas camadas de criptografia realizam essa tarefa de manter a privacidade e confidencialidade dos dados (QUICK; CHOO, 2014).

2.3.4 *Github* e *Gitlab*

No desenvolvimento de software a criação, manutenção e evolução necessitam de um gerenciamento de código. Para isso, existe uma plataforma que auxilia os desenvolvedores nessa importante tarefa, o *Git*. O repositório do *Git*, líder de sua categoria conforme SPINELLIS(2012), software também apelidado de controle de versão, contém simplesmente todas as informações necessárias para manter e gerenciar a história e as versões de um projeto (LOELIGER; MCCULLOUGH, 2012).

O *Github* e o *Gitlab* se apresentam como opções para auxiliar no armazenamento em nuvem de projetos de software. Com isso, ganhou-se a reputação de importantes plataformas sociais para o gerenciamento e desenvolvimento colaborativo de projetos de software. Ambos tem as funções de análise e apresentação de dados e gráficos referentes ao projeto em questão que facilita no *feedback* da produção e de características do código produzido (KERTÉSZ, 2015).

O *Github* tem sido utilizado em cursos de engenharia de software para o desenvolvimento de projetos de software. A plataforma fornece elementos necessários para a apresentação e documentação de software, como escopo e arquitetura, assim como dados referentes ao desenvolvimento e frequência de codificação no projeto a partir de cada desenvolvedor ou outro papel responsável. É possível também realizar alterações e evoluções em repositório separado e, após concluídas e testadas, realizar um *pull request* para que o dono daquele projeto possa revisar o código gerado a ser agregado ao projeto e realizar um *merge* autorizando essa inclusão e executando assim as alterações necessárias para a perfeita incorporação (KERTÉSZ, 2015).

3 Disciplina

Este capítulo visa apresentar conceitos, metodologia e os papéis envolvendo as disciplinas de Gestão de Portfólios e Projetos de Software (GPP) e Métodos de Desenvolvimento de Software (MDS) do curso de Engenharia de Software da Universidade de Brasília.

3.0.1 Descrição

A Faculdade UnB Gama (FGA) da Universidade de Brasília (UnB) aplica o conceito de sala de aula invertida desde 2014/1 no curso de Bacharel em Engenharia de Software (ESW) nas disciplinas de GPP e MDS.

No escopo do curso, os alunos de ambas as disciplinas interagem em grupos para a realização de um projeto de software durante todo o semestre.

3.0.2 Metodologia das Disciplinas

Os estudantes se dividem em grupos, mesclando estudantes que estejam cursando ambas as disciplinas. Cada grupo seleciona um tema e deve chegar ao fim do semestre com um software. A produção desse software é acompanhada pelo professor e por uma equipe de monitores, a fim de auxiliar os grupos.

Outro ponto importante é que durante o desenvolvimento dos projetos de cada grupo, em um dado momento, o modelo de produção de software deve ser alterado. Os grupos iniciam a produção utilizando metodologias tradicionais (PMBOK/RUP) e, na metade do semestre, os modelos metodológicos de desenvolvimento a serem adotados passam a ser os ágeis (SCRUM/XP). Dessa forma, o estudante tem a possibilidade de vivenciar ambas as metodologias.

Sendo assim, o projeto visa utilizar ambas as disciplinas com o intuito de produzir um software, onde os estudantes são estimulados a utilizar não apenas o conhecimento adquirido nas disciplinas, mas em toda a sua caminhada pela faculdade até o momento.

Para avaliação existem 2 (dois) marcos principais: *Release 1* (R1) e *Release 2* (R2) onde os grupos realizam a apresentação de todo o trabalho realizado até o momento, o acompanhamento com dados e gráficos da evolução do desenvolvimento do software, o planejamento para a continuidade do trabalho e o produto de software.

3.0.3 Práticas ágeis

A tabela 1 apresenta práticas ágeis exercidas pelos grupos no decorrer o curso.

Tabela 1 – Práticas ágeis

Prática ágil	Conceito
<i>Backlog</i> do Produto	Lista de funcionalidades em ordem de relevância mantida por um <i>stakeholder</i> específico, o <i>product owner</i>
Papel de <i>Scrum master</i>	Função de facilitar o processo de desenvolvimento
Reunião de planejamento de <i>sprint</i>	Evento para criar <i>backlog</i> de <i>sprint</i>
Planning poker	Evento para estimar tarefas durante o planejamento da <i>sprint</i>
<i>Sprints</i> de tempo definido	Espaço de tempo definido para a realização das tarefas elencadas no planejamento e entrega de incremento de software
Propriedade coletiva	Posse compartilhada do produto em desenvolvimento
<i>Stand up</i>	Evento de curto período para resolução de eventuais problemas e dificuldades da equipe
Equipe de auto-organização	Equipe com membros da equipe voluntários para tarefas
Gráfico <i>Burndown</i>	Gráfico para monitorar o progresso da <i>Sprint</i>
Reunião de revisão da <i>sprint</i>	Reunião para apresentar o trabalho concluído na <i>sprint</i>
Retrospectiva da <i>Sprint</i>	Evento para aprender com as experiências e lições passadas na <i>sprint</i>
Escrever Histórias de Usuários	Documentação alto nível de de funcionalidades a serem implementadas
Definir um ritmo sustentável	Definir um ritmo de progresso visando produtividade e agilidade
Velocidade do projeto	Medição do desenvolvimento em função do tempo e <i>sprints</i>
Mova as pessoas ao redor	Disseminação de conhecimento e descentralização de responsabilidades
Código escrito para padrões acordados	Consenso para o correto e padronizado desenvolvimento por parte de toda a equipe
Codificar o teste unitário primeiro	Elaboração de testes unitários antes da realização e elaboração de código para uma funcionalidade
Produção de código em pareamento	Codificação em pares para disseminação de conhecimento e agilidade na execução de tarefas

Apenas um par integra o código de cada vez	Medida para evitar conflitos de integração
Realizar integrações frequentes	Medida de prevenção de conflitos de grande escala
Criar soluções de pico para reduzir o risco	Realização de tarefas para testar sobrecarga do sistema visando a busca de manobras para a minimização de risco
Nenhuma funcionalidade é adicionada precocemente	Revisão de código antes da integração
Refatorar quando e onde for possível	Realizar o aperfeiçoamento do código sempre que possível
Todo o código deve ter testes unitários	Necessidade de testes para minimização de riscos e <i>bugs</i>
Todo o código deve passar por todos os testes unitários antes de poder ser liberado	Realização de medida preventiva contra quebra de código
Quando um <i>bug</i> é encontrado, testes são criados	Elaboração de testes para tratar bugs encontrados no sistema
Testes de aceitação	Os testes de aceitação são executados com frequência e a pontuação é publicada

3.0.4 MDS – Práticas de desenvolvimento

MDS é uma disciplina que trabalha no estudante conhecimentos relativos a diferentes modelos de ciclo de vida para um projeto de software. Faz com que o estudante seja estimulado a compreender os diferentes aspectos de cada modelo, identificando suas características e quando usar um em detrimento de outro.

3.0.5 GPP – práticas de gestão de projeto

GPP é uma disciplina que visa dar ao estudante uma gama de conhecimentos e técnicas, com o intuito de permitir a boa gestão estratégica de projetos de software. Isso transpassa a gestão de escopo, tempo, custos, qualidade e diversos outros contextos do projeto.

3.0.6 Papel do mentor

Os mentores, também conhecidos como *coaches* no escopo da disciplina, são importantes na realização dos projetos e, no geral, atuam como suportes para os grupos. Os

mentores inicialmente realizam treinamentos para transmitir o conhecimento aos grupos visando possibilitar o desenvolvimento e elevar a qualidade do produto a ser desenvolvido. No decorrer do semestre os mentores auxiliam na revisão de artefatos elaborados pelo grupo e fornecem *feedbacks* para refinar o trabalho.

Outra tarefa importante desempenhada pelos mentores é nos marcos da disciplina onde os mesmos têm a função de analisar e avaliar o trabalho realizado pelos grupos. Esse é um momento relevante tendo em vista que essas avaliações refletem na nota do grupo e individual dos componentes desse grupo.

4 Metodologia

O presente estudo consiste em pesquisa aplicada, de caráter exploratório e descritivo, que visa não só relacionar as variáveis de análise central, bem como apresentar conjuntos de informações que possivelmente sirvam de diretrizes para ações de transformação do ensino de Engenharia de Software.

Nesse sentido, os resultados são apresentados sobre forma quantitativa e qualitativa, a partir da coleta de informações de fontes primárias e secundárias, incluindo revisão bibliográfica, além de aplicação de *survey* com o auxílio de questionário e estudo de caso.

A vantagem do estudo de caso é a sua aplicabilidade a contextos contemporâneos de vida real (DOOLEY, 2002). O autor exprime ainda que:

Investigadores de várias disciplinas usam o método de investigação do estudo de caso para desenvolver teoria, para produzir nova teoria, para contestar ou desafiar teoria, para explicar uma situação, para estabelecer uma base de aplicação de soluções para situações, para explorar, ou para descrever um objecto ou fenómeno (p. 343-344).

A planificação da pesquisa incluiu, em primeiro lugar, o levantamento dos dados secundários e a revisão da literatura e, concomitantemente, o contato com as fontes pessoais, a fim de promover a coleta de dados em campo. A aplicação dos instrumentos específicos de pesquisa foi realizada através de contato remoto, por meio digitais.

No que se refere ao procedimento adotado para esta pesquisa foi realizado um *survey* para a coleta de informações e análise da influência da mentoria aplicada nas disciplinas de GPP/MDS com alunos que já passaram pelas duas disciplinas. As questões que o compõem foram elaboradas a partir da entrevista com dois professores doutores que tiveram participação no processo de estruturação do curso envolvendo as duas disciplinas.

Foi utilizado esse tipo de pesquisa para estabelecer a metodologia utilizada no curso assim como validar processos e técnicas presentes nessa metodologia de ensino.

Para a elaboração do *survey* foi utilizada a ferramenta *Google Forms*. Com o *survey* pronto foram enviados convites para participação das pessoas por *e-mail* e *facebook*. A amostra foi feita com o auxílio dos professores entrevistados na etapa anterior onde foi passada uma lista com os nomes e *e-mails* dos alunos que já haviam cursado as duas disciplinas a partir da adequação do método em questão. As respostas foram salvas e analisadas para a conclusão deste trabalho.

O *survey* elaborado contém sete seções. A primeira seção apresenta informações a respeito do objetivo da pesquisa. A segunda seção “Informações Gerais” para levan-

tamento de dados a respeito do participante e análise da relevância e confiabilidade da participação do indivíduo na pesquisa. A terceira “Sobre as disciplinas” com perguntas a respeito das disciplinas e tópicos para que o usuário pudesse apresentar sua visão sobre expectativas e aspectos técnicos da ESW. A quarta “Sobre a metodologia e práticas ágeis” leva em consideração práticas inerentes a aspectos técnicos e a abordagem ágil no desenvolvimento de software. A quinta seção “Sobre software livre” refere-se a aspectos e políticas relacionadas ao software livre aplicadas na disciplina e sua importância. A sexta seção “Perguntas finais sobre software livre” apresenta questões para o participante expressar a sua opinião a respeito do movimento. A sétima e última seção “Sobre o código” contém questões relacionadas a aspectos e técnicas mais utilizadas no desenvolvimento do código e sobre ferramentas que auxiliam no desenvolvimento.

A apresentação dos resultados quantitativos e qualitativos foi acompanhada de análise direcionada ao contexto que configura o objeto de estudo, a fim de que se cumpra o papel científico deste projeto, no sentido de alcançar os objetivos propostos.

5 Resultados e Discussões

5.1 Análise qualitativa

A análise do *survey* foi feita conforme o direcionamento e as etapas apresentadas por (MILES et al., 1994). Nessa obra, os autores apresentam três etapas que geralmente são seguidas na análise de dados: redução, exibição e conclusão/verificação.

O tipo de pesquisa *survey* se caracteriza pela interrogação a um grupo significativo de pessoas a respeito do problema a ser estudado para, em seguida, mediante análise quantitativa e/ou qualitativa obter as conclusões a partir do estudo. Em suma, o *survey* tem a função de tornar conhecido o comportamento, a sociedade e as práticas de uma sociedade (GIL, 2015).

Como vantagens desse tipo de pesquisa (GIL, 2015) exemplifica:

- conhecimento direto da realidade já que as próprias pessoas que vivenciam a realidade em questão transmitem informações a respeito.
- economia e rapidez pela possibilidade da obtenção de grande quantidade de dados em curto período de tempo e, ainda mais, como o auxílio de ferramentas virtuais e questionário.
- quantificação através do agrupamento dos dados obtidos possibilitando assim uma análise quantitativa dos dados.

Como principais limitações desse tipo de pesquisa o autor exemplifica:

- ênfase nos aspectos perspectivas pelo fato de que cada indivíduo participante tem uma percepção, aspecto subjetivo, acerca de si mesmo e daquilo que o rodeia.
- pouca profundidade no estudo da estrutura e dos processos sociais não havendo o aprofundamento de fatores interpessoais e institucionais que determinam esses fenômenos sociais.
- limitada apreensão do processo de mudança pois oferece uma visão estática do fenômeno em questão.

Para a entrevista foi utilizado um questionário como uma técnica de investigação composta por um conjunto de questões como auxílio ao *survey*. Dessa forma, contendo questões fechadas e abertas extrairam-se informações dos entrevistados para o estudo do tema central (GIL, 2015).

5.1.1 Processo de aplicação do *survey*

O *survey* foi elaborado em ferramenta online que fornece um link o qual foi enviado para os alunos por redes sociais (e-mail, facebook e whatsapp). Uma das únicas limitações para a filtragem das respostas válidas é que a pessoa deveria ter cursado ambas as disciplinas, GPP e MDS. Àquelas que cursaram apenas MDS até a resposta do *survey* determinou-se como inválidas. Um total de 120 pessoas responderam o *survey*. Desses, 30 respostas foram consideradas inválidas pois não se encaixavam no perfil necessário para a pesquisa. Portanto, um total de 90 respostas foram consideradas válidas.

5.1.2 Método de análise dos dados

Para a análise do *survey* foram utilizados os métodos de análise quantitativa e qualitativa. A análise quantitativa foi usada principalmente para perguntas com respostas onde a realidade natural ou social é estável e quantificável, a partir de um distanciamento entre o investigador e a realidade estudada. Por outro lado, a análise qualitativa é orientada por uma perspectiva interpretativa e foi aplicada para resultados do *survey* com perguntas abertas, que possibilitavam a exposição da opinião do participante (MEIRINHOS; OSÓRIO, 2010).

5.1.3 Perguntas e Respostas do *Survey*

Nesta seção serão apresentadas algumas perguntas constantes no *survey* aplicado nessa pesquisa e uma visão geral das respostas concedidas pelos participantes que se voluntariaram.

O questionário foi realizado por pessoas de diferentes idades, 49,1% dos entrevistados têm entre 20 e 22 anos, 41,1% estão entre 23 e 25 anos e 9,9% possuem mais de 26 anos, conforme apresentado na figura 3.

Com relação ao nível de conhecimento que cada participante possuía antes de realizar as disciplinas em estudo, 3,3% declararam não ter conhecimento algum, 65,6% pouco conhecimento, 21,1% conhecimento razoável, 8,9% muito conhecimento e 1,1% afirmou possuir domínio de conteúdo.

Pode-se dizer que a maioria dos alunos entrevistados possuíam pouco conhecimento como apresenta a figura 4.

Considerando dois coaches a quantidade padrão a ser mobilizada por grupo de trabalho, 58,9% dos entrevistados afirmaram vivenciar essa realidade, 5,6% alegaram não possuir nenhum, 31,1% apenas um e 4,4% dos participantes entre três e quatro coaches como apresenta a figura 5. OS coaches têm o papel de mentor na relação e desenvolvimento do projeto.

Idade

90 respostas

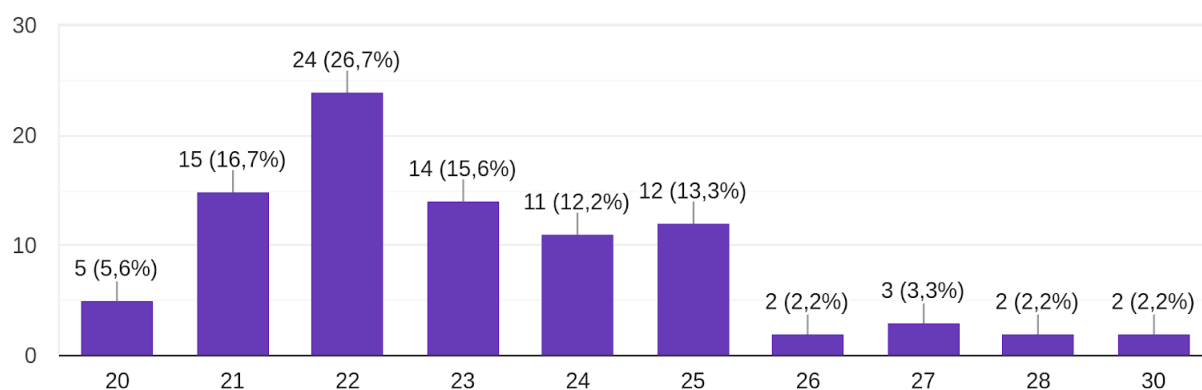


Figura 3 – Faixa etária dos participantes

Em uma escala de 0 a 10, qual o nível de conhecimento acreditava possuir antes de cursar as disciplinas?

90 respostas

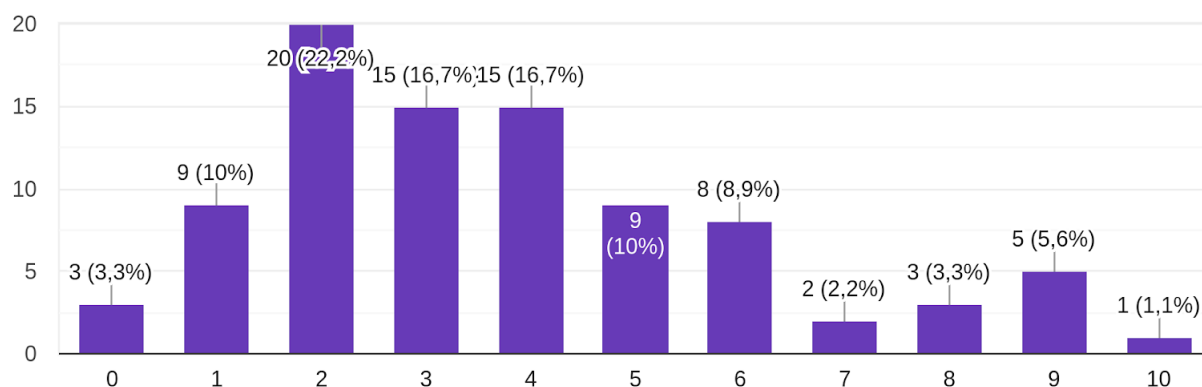


Figura 4 – Nível de conhecimento antes do curso

Nesse ponto, 31,1% do alunos afirmaram que o direcionamento dos coaches foi irrelevante ou pouco relevante para o desenvolvimento pessoal ou em grupo, 17,8% expuseram ter sido normal e 51,1% responderam que essa monitoria foi relevante ou muito relevante no desenvolvimento do projeto, como pode ser visto na figura 6.

Com relação às atividades que mais agregaram valor durante a *sprint*, pode-se

Quantos coaches foram mobilizados a tua equipe de projeto?

90 respostas

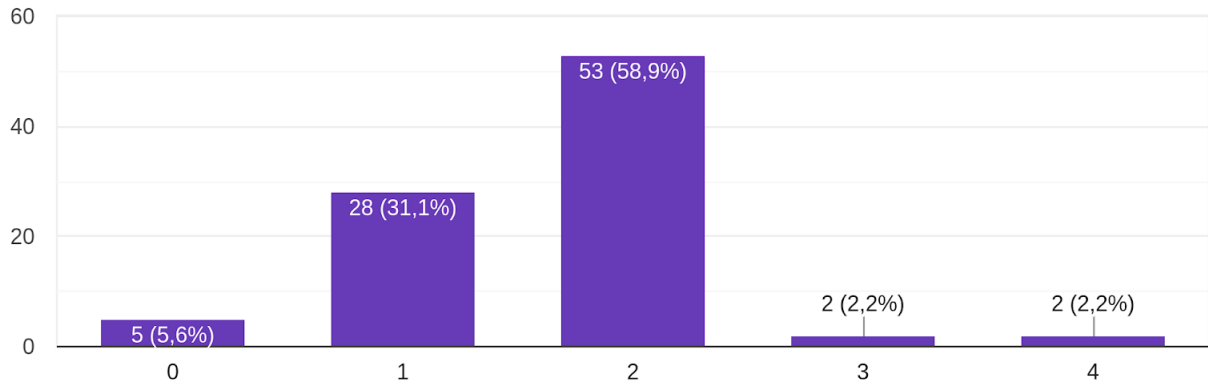


Figura 5 – Quantidade de coaches mobilizados

Considera que o direcionamento oferecido pelos coaches foi relevante para seu desenvolvimento e o da equipe?

90 respostas

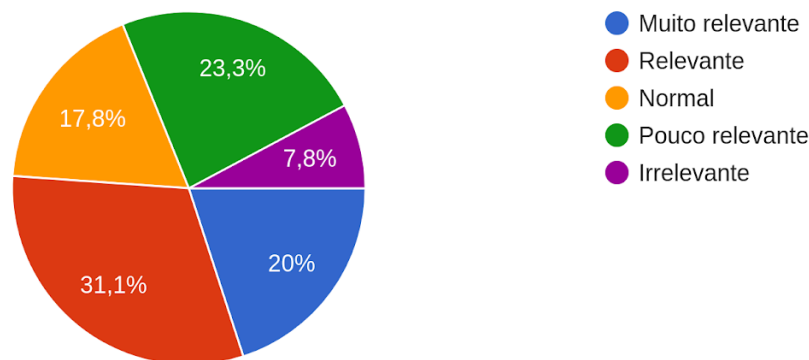


Figura 6 – Relevância do direcionamento

perceber que 70% ou mais dos participantes declararam considerar que as atividades de programar, planejar e pairar crescem mais. O pareamento é também um processo de mentoria que ocorre no desenvolvimento do projeto. 52% afirmaram que a revisão da *sprint* acrescentam valor. 40% ou menos elencaram projetar arquitetura, testar, revisar código e comunicação como atividades que agregar maior valor. Esses dados podem ser observados na figura 7.

Em sua opinião, quais das atividades agregavam maior valor durante a Sprint?

90 respostas

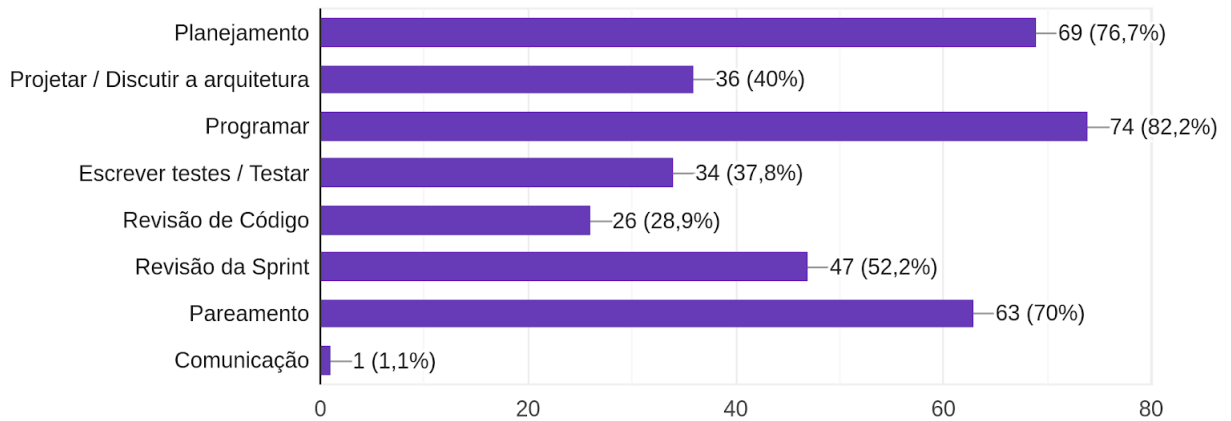


Figura 7 – Atividades de maior valor na *sprint*

Pode-se perceber através da figura 8 que após a conclusão das disciplinas, 62,2% dos participantes afirmaram ter agregado conhecimento elevado, de 8 a 10 numa escala de 0 a 10, se comparado ao inicial. Outros 33,3% conhecimento intermediário, de 5 a 7 dessa mesma escala e somente 4,4% relataram não ter agregado conhecimento relevante.

Após concluir as disciplinas, em uma escala de 0 a 10, quanto considera ter agregado ao seu nível de conhecimento inicial?

90 respostas

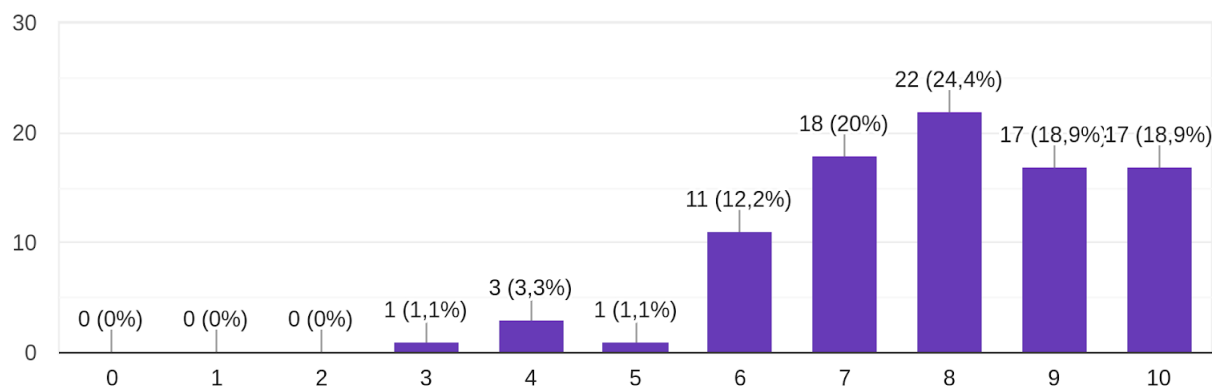


Figura 8 – Nível de conhecimento agregado

A realização de *stand ups* durante as *sprints* foi considerada muito relevante por 82,2% dos participantes. Os outros 17,8% afirmaram entender como normal ou pouco importante, conforme apresentado na figura 9.

O *stand up* é uma prática do desenvolvimento ágil onde a equipe de software se reúne para compartilhar as experiências, dificuldades e compartilhar conhecimento e informações para auxiliar os outros desenvolvedores em tarefas diferentes. Nesse processo, a mentoria também está presente pois integrantes da equipe auxiliam com dicas e conselhos para a resolução de problemas.

Qual a importância da realização de Stand-ups durante as sprints?

90 respostas

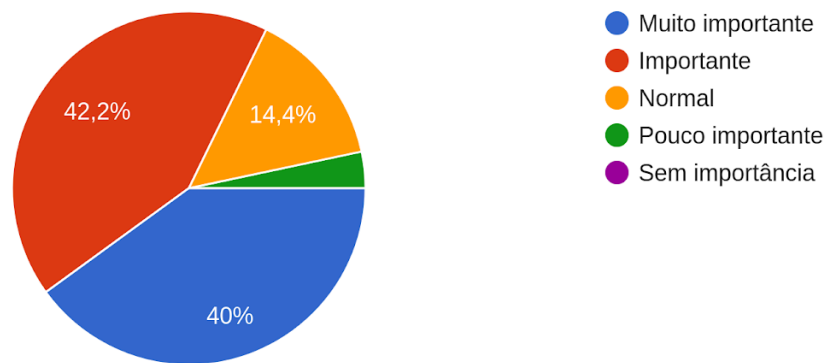


Figura 9 – Importância de *stand ups*

Dentre os participantes do *survey*, 83,3% afirmaram que a revisão de código é uma prática que tem grande importância no decorrer do projeto. Os outros 15,6% disseram que a revisão de código não apresenta melhora nem piora no projeto. O outro 1,1% considera que foi pouco relevante no desenvolvimento do projeto. Isso pode ser visto na figura 10. O processo de revisão de código, realizada por desenvolvedores mais experientes, auxilia no aprimoramento das habilidades do desenvolvedor responsável pelo código a ser revisado.

O trabalho em grupo se mostrou ser um aspecto positivo para 96,7% das pessoas que participaram da pesquisa. A minoria de 3,3% considerou ser esse um ponto normal ou pouco relevante como pode-se notar na figura 11.

O uso de ferramentas de repositório aberto foi obrigatório na realização de cada projeto. Alguns dos mais utilizados são *GitHub* e *GitLab*.

Nesse sentido, 83,3% dos alunos consideraram muito importante ou importante o uso dessas plataformas no desenvolvimento. Apenas 16,7% afirmaram julgar normal,

Qual a importância da revisão de código (pull request)?

90 respostas

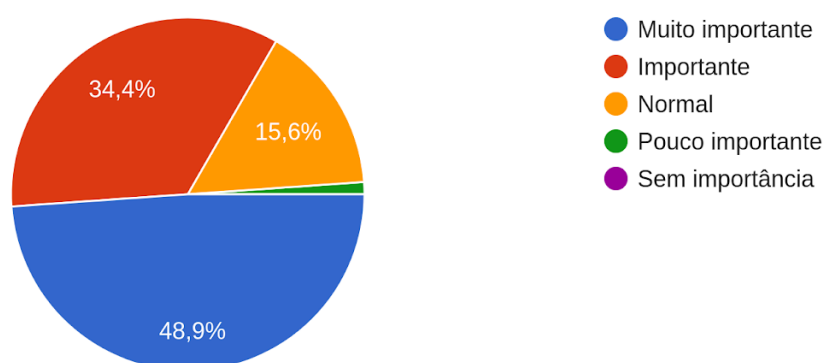


Figura 10 – Importância de Revisão de Código

Trabalhar em grupo foi importante para você durante o projeto?

90 respostas

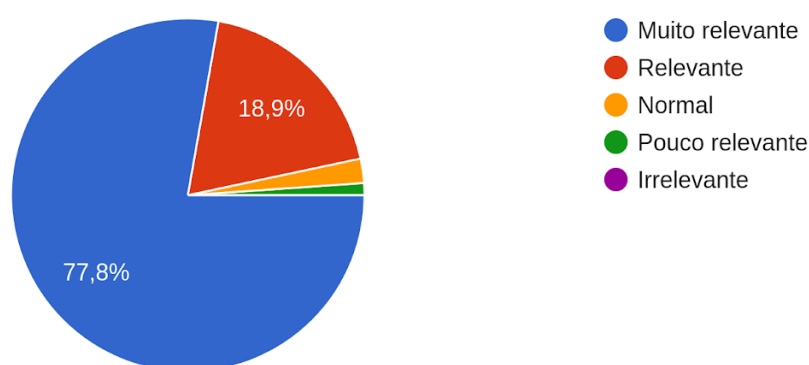


Figura 11 – Importância de trabalhar em grupo

pouco relevante ou sem importância seu uso. Esses dados podem ser contemplados na figura 12

No contexto de software livre, as práticas dessa filosofia de desenvolvimento foram avaliadas por 74,4% dos participantes como muito importante ou importante. Os outros 25,6% responderam que as consideram normal, pouco importante ou sem importância. Os dados referentes a essa pergunta podem ser observados na figura 13

Outro aspecto avaliado pelos participantes foi a importância do monitoramento da

Qual a importância do uso de repositório aberto (git)?

90 respostas

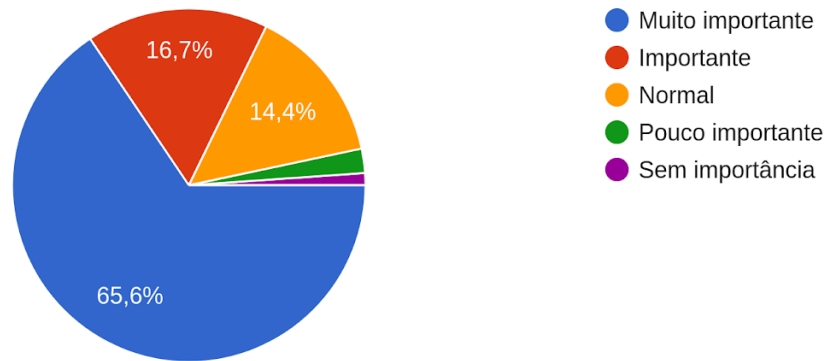


Figura 12 – Importância do uso de repositório aberto

Qual a importância das práticas de comunidades de software livre?

90 respostas

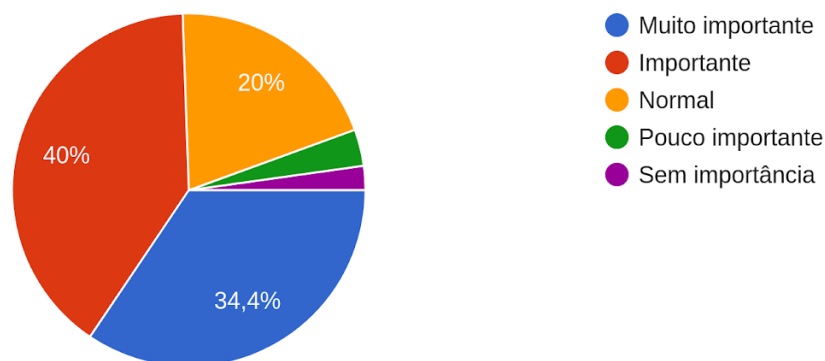


Figura 13 – Importância de práticas de software livre

qualidade de código. Nesse quesito 71,9% avaliaram como muito importante ou importante enquanto os outros 28,1% consideraram normal, pouco importante ou sem importância. A figura 14 apresenta essas informações.

A arquitetura de software, que tem a finalidade de direcionar o desenvolvimento e estrutura do código, foi reconhecida pela maioria dos voluntários. Segundo os dados exibidos na figura 15 mostra que 90% declaram muito importante ou importante a existência de uma arquitetura de software bem definida para a realização do projeto. Os outros 10%

Qual a importância do monitoramento da qualidade de código (code climate, por exemplo)?

89 respostas

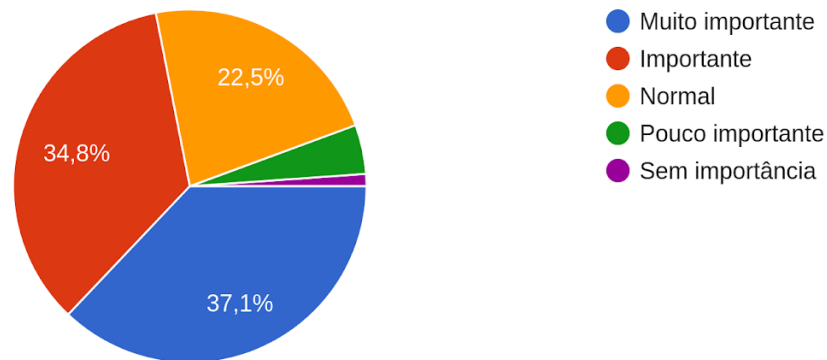


Figura 14 – Importância do monitoramento da qualidade de código

consideraram normal, não havendo opiniões negativas nessa questão.

Qual a importância de se ter uma arquitetura de software bem definida?

90 respostas

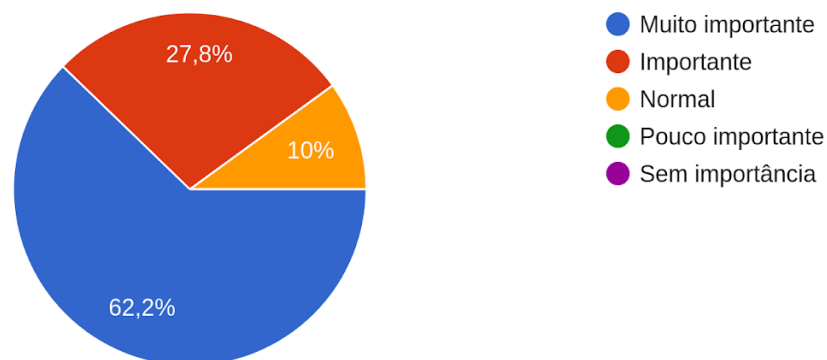


Figura 15 – Importância de arquitetura bem definida

A figura 16 mostra que 91,1% consideraram que o uso de ferramentas que auxiliem no desenvolvimento de software, para testes, integração contínua e automações, por exemplo, é muito importante ou importante. Os 8,9% restantes declararam como normal esse uso. Não havendo opiniões negativas nesse quesito.

Para a realização do projeto os grupos elaboraram um documento de instruções

Qual a importância do uso de ferramentas que auxiliem no desenvolvimento do software (testes, integração contínua, automações)?

90 respostas

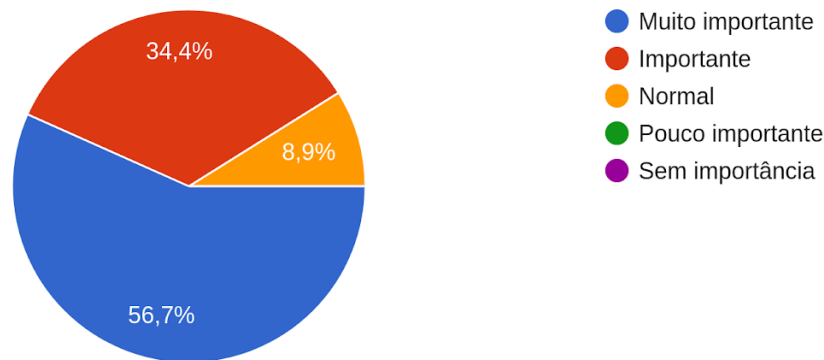


Figura 16 – Importância de ferramentas de auxílio no desenvolvimento

de contribuição para auxílio no desenvolvimento (política de *branches*, critérios para aceitação de *pull requests* - cobertura de testes subir ou manter, métricas de qualidade de código e a necessidade de se ter uma integração contínua). Dessa forma, os interessados em contribuir com o projeto deveriam tomar as devidas ações para estar de acordo com as regras do documento. Nesse sentido, os 84,4% consideraram muito importante ou importante essa medida enquanto os outros 15,6% avaliaram como normal ou pouco importante, conforme mostrado na figura 17.

Para o acompanhamento do desenvolvimento os grupos trabalhavam com a utilização de *roadmap* através de *issues* no repositório aberto de cada um. Essa prática foi considerada muito importante ou importante por 75,5% dos participantes e normal ou pouco importante pelos outros 25,5%, como está na figura 18.

A partir da contribuição em projetos de software livre, os participantes opinaram pela importância que vem essa prática como potencialização de seu aprendizado, como visto na figura 19. Nesse sentido, 80,9% declararam notas entre 7 e 10 para essa questão, 14,6% entre 5 e 6 e os outros 4,4% inferior a 4.

Os projetos de software livre e a comunidade de software têm-se tornado cada vez mais visíveis com a utilização de repositórios abertos. Empresas de todos os portes acompanham esse mundo. Com isso, a pesquisa questionou sobre as oportunidades de aprendizagem e emprego nos projetos de software livre conforme pode ser visto na figura 20. 94,4% consideraram projetos de software livre oportunidades de aprendizagem e emprego enquanto 5,6% consideraram que não.

Qual a importância de elaborar um documento com instruções de contribuição (política de branches, cri...e de se ter uma integração contínua) ?

90 respostas

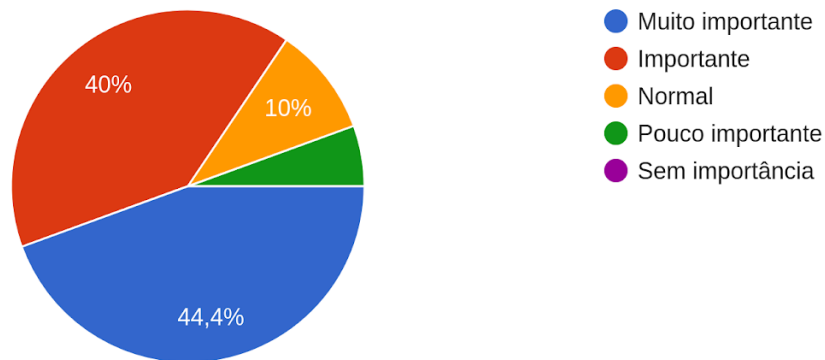


Figura 17 – Importância do documento de contribuição

Qual a importância de criar/utilizar roadmap por meio de issues?

90 respostas

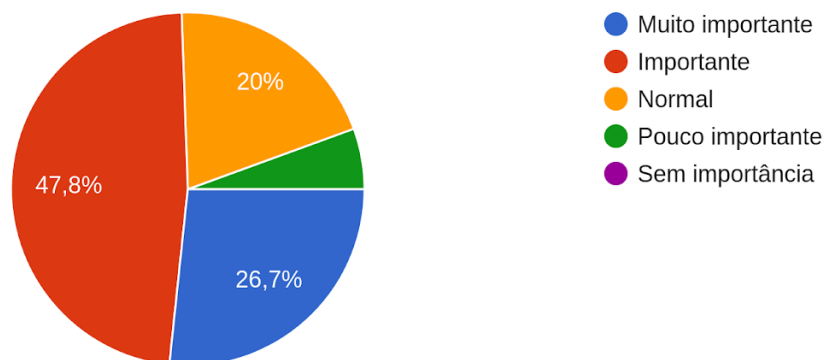


Figura 18 – Importância de roadmap através de issues no desenvolvimento

Ainda nesse universo, 65,6% disseram sentir-se confortável e confiante em contribuir com projetos de software livre, os outros 34,4% afirmaram não sentir o mesmo como está na figura 21. E os projetos de software livre foram considerados como oportunidade de negócio por 73,3% dos participantes, enquanto os outros 36,7% declararam não perceber dessa forma, como mostra a figura 22. Com isso, 81,1% demonstraram interesse em colaborar com projetos de software livre ou criá-los Já os outros 18,9%, como pode ser

De 1 a 10, qual nota você atribui ao seu aprendizado ao visitar ou contribuir com outros projetos de software livre?

89 respostas

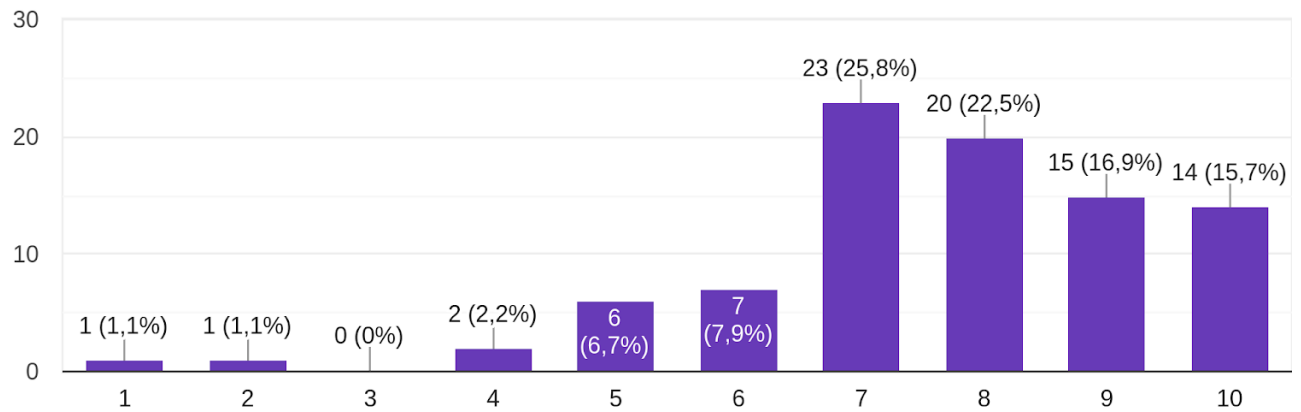


Figura 19 – Notas para o aprendizado em contribuição a projetos de software livre

Considera projetos de software livre como oportunidades de aprendizagem e/ou emprego?

90 respostas

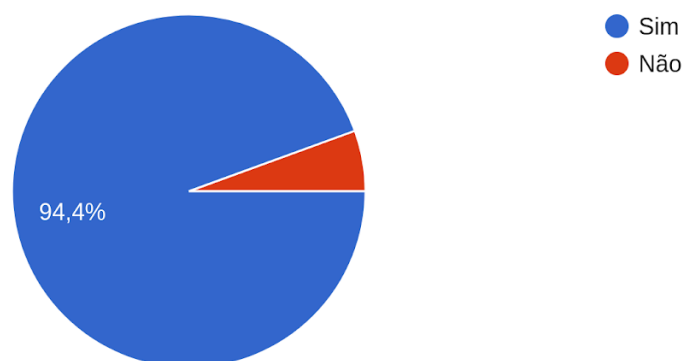


Figura 20 – Projetos de software livre como oportunidades

visto na figura 23, não têm essa mesma percepção.

Você se sente confortável/confiante em contribuir com projetos de software livre?

90 respostas

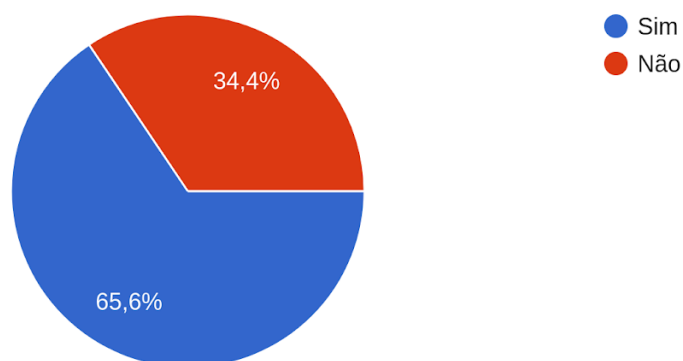


Figura 21 – Confiança em contribuir com projetos de software livre

Considera projetos de software livre como oportunidade de negócio?

90 respostas

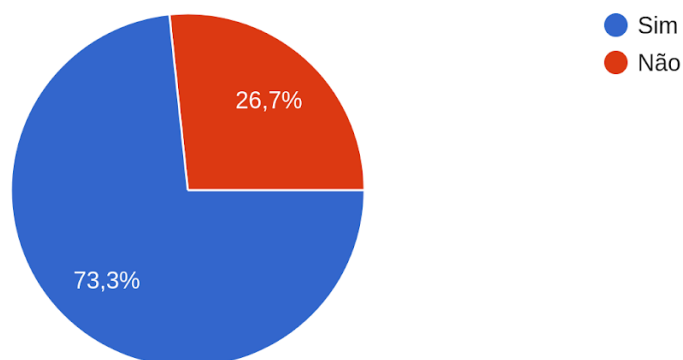


Figura 22 – Projetos de software livre como oportunidade de negócio

5.2 Considerações sobre a disciplina

Nesta seção, discutiremos algumas das percepções e lições aprendidas em quatro anos de oferta de curso. Estes são subjetivos. O objetivo desta seção é principalmente apresentar algumas questões às quais um curso autêntico usando FLOSS como ferramenta pedagógica provavelmente será confrontada.

Possui interesse em colaborar com projetos de software livre ou criá-los?

90 respostas

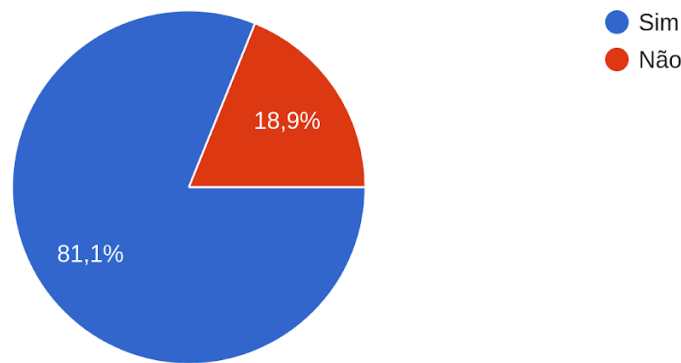


Figura 23 – Interesse em colaborar em projetos de software livre

1. Importância dos mentores - Desenvolvedores precisam ser adequadamente apoiados e receber liberdade juntamente com suas responsabilidades.
2. Incentivar os alunos a compartilhar - O compartilhamento de conhecimento tem papel importante no desenvolvimento da equipe e do produto. Percebe-se nesse processo a presença da mentoria.
3. Importância do *Roadmap* do projeto - O *Roadmap* se mostrou relevante para o direcionamento, acompanhamento e gerenciamento do projeto.
4. Gestão técnica essencial - O papel da equipe de GPP com conhecimento técnico realizando a gestão do projeto eleva o nível do desenvolvimento.
5. Cultura capacitada - Tanto a cultura ágil quanto a FLOSS devem se espalhar. Incentivar times a acompanharem grandes repositórios de OSS.
6. Não padronize o aprendizado para permitir inovação e experimentação.
7. Transparência de rastreamento em projeto ágil e OSS - Todos os projetos estão disponíveis, lições aprendidas e reuso de projetos passados fazem com que projetos evoluam de semestre a semestre.
8. A falta de motivação dos mentores é um risco do projeto - a equipe de GPP é essencial para o bom funcionamento do projeto, uma vez que é responsável pela gestão do projeto e pelo treinamento do time de MDS em boas práticas.

6 Considerações finais

Esse trabalho disponibiliza reflexões e ideias a respeito de um modelo de ensino de ESW e apresenta informações a respeito das disciplinas de GPP e MDS do curso de ESW da Universidade de Brasília colocando-as como instrumento de auxílio aos alunos que ingressam nessa etapa de formação acadêmica. Além disso, o estudo de caso pôde textificar a partir de opiniões e considerações transmitidas pelos próprios estudantes que vivenciaram a realidade desse método.

Após a aplicação do *survey* foi possível observar que houve ganho considerável de conhecimento desde o uso de metodologia de compartilhamento de estudo a partir das relações interpessoais de aprendizagem nas disciplinas em foco. Ganho esse que pode ser textificado a partir da interpretação e comparação do nível de conhecimento dos alunos antes da realização do curso ao nível de assimilação ao fim desse. Além disso, pôde-se observar que a partir do direcionamento de 1 (um) ou 2 (dois) *coaches* a maioria dos envolvidos consideraram relevante o direcionamento dispensado a eles o que potencializou o progresso das habilidades dos aprendizes. Vale ressaltar que, no que diz respeito às atividades que agregam maior valor na *sprint*, o pareamento e o trabalho em grupo tornam funcional a aplicação da mentoria na relação aluno-aluno. Ainda é importante mostrar que durante a realização de *stund-ups* e da revisão via *pull-requests* são oferecidos direcionamentos para a resolução das dificuldades explanadas pelos participantes dessa prática. Tem-se ainda uma maneira eficaz de disseminação do conhecimento conhecida como comunidade de software livre por permitir a interação entre desenvolvedores de maneira remota e virtual.

O trabalho foi importante para apresentar como a prática da mentoria ocorre na instituição e fomentou a pesquisa na área de ensino de ESW, além de possibilitar a assimilação de conceitos e reflexões. Mas ainda faz-se necessário a continuação do estudo para aprofundamento no tema e descoberta de novas metodologias de ensino e progresso do aprendizado escolar daqueles que vierem a participar de tais práticas.

Referências

- AKIN, L.; HILBUN, J. E-mentoring in three voices. *Online Journal of Distance Learning Administration*, v. 10, n. 1, 2007. Citado na página 27.
- ALVES, L. Um olhar pedagógico das interfaces do moodle. Eduneb, 2009. Citado na página 31.
- DAIGLE, R. J.; DORAN, M. V.; PARDUE, J. H. Integrating collaborative problem solving throughout the curriculum. In: ACM. *ACM SIGCSE Bulletin*. [S.l.], 1996. v. 28, n. 1, p. 237–241. Citado na página 29.
- DOOLEY, L. M. Case study research and theory building. *Advances in developing human resources*, Sage Publications Sage CA: Thousand Oaks, CA, v. 4, n. 3, p. 335–354, 2002. Citado na página 37.
- DRAGO, I. et al. Inside dropbox: understanding personal cloud storage services. In: ACM. *Proceedings of the 2012 Internet Measurement Conference*. [S.l.], 2012. p. 481–494. Citado na página 31.
- DUBOIS, D. L. et al. How effective are mentoring programs for youth? a systematic assessment of the evidence. *Psychological Science in the Public Interest*, Sage Publications Sage CA: Los Angeles, CA, v. 12, n. 2, p. 57–91, 2011. Citado 2 vezes nas páginas 27 e 28.
- ERDOGMUS, H.; PÉRAIRE, C. Flipping a graduate-level software engineering foundations course. In: IEEE PRESS. *Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track*. [S.l.], 2017. p. 23–32. Citado na página 30.
- GIL, A. C. Métodos e técnicas de pesquisa social. In: *Métodos e técnicas de pesquisa social*. [S.l.: s.n.], 2015. Citado na página 39.
- GIRVES, J. E.; ZEPEDA, Y.; GWATHMEY, J. K. Mentoring in a post-affirmative action world. *Journal of Social Issues*, Wiley Online Library, v. 61, n. 3, p. 449–479, 2005. Citado na página 27.
- GOOLD, A.; HORAN, P. Foundation software engineering practices for capstone projects and beyond. In: IEEE. *Proceedings 15th Conference on Software Engineering Education and Training (CSEE&T 2002)*. [S.l.], 2002. p. 140–146. Citado na página 29.
- HAMILTON, S. F.; HAMILTON, M. A. Mentoring programs: Promise and paradox. *Phi Delta Kappan*, Phi Delta Kappa, v. 73, n. 7, p. 546, 1992. Citado na página 28.
- HAMILTON, S. F.; HAMILTON, M. A. Building mentoring relationships. *New directions for youth development*, Wiley Online Library, v. 2010, n. 126, p. 141–144, 2010. Citado na página 28.
- KERTÉSZ, C.-Z. Using github in the classroom—a collaborative learning experience. In: *2015 IEEE 21st International Symposium for Design and Technology in Electronic Packaging (SIITME)*. [S.l.: s.n.], 2015. p. 381–386. Citado na página 32.

- LEBLANC, R. J. et al. *Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*. [S.l.]: IEEE Computer Society, 2006. Citado na página 29.
- LOELIGER, J.; MCCULLOUGH, M. *Version Control with Git: Powerful tools and techniques for collaborative software development*. [S.l.]: "O'Reilly Media, Inc.", 2012. Citado na página 32.
- MARTÍN, A. J. S.; MORO, S. P. White paper: Introduccioón a los recursos tic en la enseñanza. Decanato de la Facultad de Farmacia de la Universidad de Salamanca. Citado na página 31.
- MCCARTHY, J. International design collaboration and mentoring for tertiary students through facebook. *Australasian Journal of Educational Technology*, v. 28, n. 5, 2012. Citado na página 27.
- MEIRINHOS, M.; OSÓRIO, A. O estudo de caso como estratégia de investigação em educação. *Revista EduSer*, Instituto Politécnico de Bragança, Escola Superior de Educação, n. 2 (2), p. 49–65, 2010. Citado na página 40.
- MILES, M. B. et al. *Qualitative data analysis: An expanded sourcebook*. [S.l.]: sage, 1994. Citado na página 39.
- PEREIRA, G.; KEPLER, F. N. Edx: Uma plataforma inteligente no gerenciamento da aprendizagem. *Anais do Salão Internacional de Ensino, Pesquisa e Extensão*, v. 6, n. 1, 2014. Citado na página 30.
- PRIKLADNICKI, R. et al. Ensino de engenharia de software: desafios, estratégias de ensino e lições aprendidas. *FEES-Fórum de Educação em Engenharia de Software*, p. 1–8, 2009. Citado 2 vezes nas páginas 29 e 30.
- QUICK, D.; CHOO, K.-K. R. Google drive: forensic analysis of data remnants. *Journal of Network and Computer Applications*, Elsevier, v. 40, p. 179–193, 2014. Citado na página 32.
- SANTOS, B. de S. *A globalização e as ciências sociais*. [S.l.]: Cortez Editora, 2002. Citado na página 28.
- SPINELLIS, D. Git. *IEEE software*, IEEE, v. 29, n. 3, p. 100–101, 2012. Citado na página 32.
- TRAINER, E. H.; KALYANASUNDARAM, A.; HERBSLEB, J. D. e-mentoring for software engineering: a socio-technical perspective. In: IEEE. *Software Engineering: Software Engineering Education and Training Track (ICSE-SEET), 2017 IEEE/ACM 39th International Conference on*. [S.l.], 2017. p. 107–116. Citado na página 23.