

Ensino, Pesquisa e Extensão no Ensino de Engenharia de Software: Um Relato de Experiência

Andréa Sabedra Bordin¹, Lorenzo Mendes Rodrigues¹, Tarcisio Casagrande¹

¹Universidade Federal de Santa Catarina (UFSC)
Campus Araranguá – Santa Catarina, Brasil.

andrea.bordin@ufsc.br

{lorenzo.mr.rm,tarcisio.casagrande}@gmail.com

Abstract. *This article reports a software engineering teaching experience in which a mixed methodological approach was adopted, with face-to-face classes and project development in collaboration with companies in the software industry. The projects had an extensionist and research nature and demanded increasing levels of autonomy from the students. The planning of the challenges, the form of execution and the resulting events are detailed in order to allow reproducibility. The experience was evaluated from the perspective of the students and the teacher and several suggestions to deal with the challenges are presented.*

Resumo. *Este artigo relata uma experiência de ensino de engenharia de software na qual foi adotada uma abordagem metodológica mista, com aulas presenciais e desenvolvimento de projetos em colaboração com empresas da indústria de software. Os projetos tiveram cunho extensionista e de pesquisa e demandaram níveis crescentes de autonomia dos alunos. O planejamento dos desafios, a forma de execução e os artefatos resultantes são detalhados de forma a permitir a reprodutibilidade. A experiência foi avaliada na perspectiva dos alunos e da professora e uma série de sugestões para lidar com os desafios são apresentadas.*

1. Introdução

O ensino de Engenharia de Software (ES) em cursos de graduação deve abordar um conjunto de tópicos de conhecimento determinado pelas Diretrizes Curriculares Nacionais (DCNs) da área de Computação [Ministério da Educação 2016]. Tanto as DCNs como os Referenciais de Formação da SBC [Zorzo et al. 2017] e o *Computing Curricula 2020* da ACM também destacam a importância de estabelecer estratégias entre a teoria e a prática no ensino de ES.

O ensino de ES tem um reconhecido potencial para ser abordado de forma prática, por meio de projetos que envolvam a indústria de software ([Bruegge et al. 2015], [Watson and Cutting 2022], [Raibulet and Lago 2022]). No entanto, nem todos os cursos fornecem oportunidades aos alunos de realizarem atividades práticas ou provenientes de situações do mundo real. Dentre os motivos, estão o esforço significativo em termos de tempo e recursos humanos para apoiar o processo de ensino, assim como a existência de clientes reais [Marques et al. 2014].

A extensão universitária, agora regulamentada pela Resolução Nº 7 CNE/CSE de 2018, que institui que 10% da carga horária de um curso de graduação deve ser cumprida por meio de ações de extensão, pode ser uma oportunidade para fomentar a relação dialógica entre a universidade e as demandas da sociedade relacionadas à engenharia de software. Ações de extensão na forma de projetos, cursos e oficinas, eventos e prestação de serviços estão previstas na referida resolução [Brasil 2018]. [Lopes et al. 2017] e [Lima et al. 2019] relataram experiências envolvendo o atendimento de demandas com caráter extensionista em disciplinas de ES.

Contudo, para além da indissociabilidade entre ensino e extensão, é preciso pensar a indissociabilidade também com a pesquisa. A Engenharia de Software, como qualquer outra área de conhecimento, tem problemas de pesquisa em aberto, que podem ser passíveis de uma investigação preliminar no contexto de uma disciplina. A pesquisa-ação, por exemplo, é uma maneira de abordar problemas reais dentro de contextos organizacionais, estudando e melhorando seus processos [Gray 2012]. A Política Nacional de Extensão [Brasil 2012] menciona a pesquisa-ação como uma das possibilidades para promover a indissociabilidade ensino-pesquisa e extensão.

O objetivo deste artigo é relatar uma experiência de ensino, extensão e pesquisa de Engenharia de Software por meio de uma abordagem orientada a projetos, envolvendo problemas e atores da indústria de software. São descritos o processo de planejamento, acompanhamento e avaliação das percepções dos alunos e da professora.

Além dessa seção, o artigo está organizado da seguinte forma: na Seção 2 é descrito a motivação da experiência de ensino, assim como o planejamento e a execução; os resultados da avaliação da experiência são apresentados e discutidos na Seção 3 e, por fim, as principais contribuições oriundas da experiência são expostas na Seção 4.

2. Gênese, Planejamento e Execução da Experiência

A oportunidade para planejar e executar a experiência relatada neste artigo ocorreu em 2022/02 na disciplina de Engenharia de Software II (ES II) do Bacharelado em Tecnologia da Informação de Comunicação (BTIC) da Universidade Federal de Santa Catarina. A disciplina, cuja ementa contempla tópicos relacionados à processo de software e modelos de ciclo de vida, manutenção de software e qualidade de software, possui 72 horas-aula e é ofertada no quinto semestre do curso.

No referido semestre matricularam-se apenas 3 alunos, uma situação atípica que conduziu ao entendimento de que uma abordagem de ensino predominantemente teórica não seria a mais adequada. Essa situação foi considerada uma oportunidade para introduzir uma metodologia de ensino ativa, com foco no desenvolvimento de projetos reais e no aprender-fazendo.

Foi proposta uma metodologia de ensino e aprendizagem híbrida, com exposição e diálogo de conteúdos (metodologia tradicional), conjuntamente com o desenvolvimento de projetos na área dos tópicos da disciplina, em colaboração com empresas da indústria de software da cidade e região (metodologia ativa). A metodologia ativa utilizada agregou elementos da Aprendizagem Baseada em Projetos (ABPj), como o trabalho em grupo e a solução de problemas realistas. Na ABPj, o professor tem um papel menos central, atuando como um facilitador, e os alunos assumem mais responsabilidade por sua própria

aprendizagem, o que resulta em maior envolvimento desses alunos [Martin et al. 2014]. Nas subseções seguintes são apresentados os detalhes do planejamento e da execução da disciplina.

2.1. Planejamento da Disciplina

A disciplina de ES II foi dividida em três unidades temáticas: Unidade Processo de Software (UPS), Unidade Manutenção de Software (UMS) e Unidade Qualidade de Software (UQS). Para cada unidade foram destinadas de cinco (5) a seis (6) semanas de aula, com dois (2) encontros (denominados E1 e E2) semanais de 100 minutos cada.

Para cada unidade temática foram reservadas duas (2) semanas às aulas expositivas e dialogadas e de três (3) a quatro (4) semanas para o desenvolvimento, o acompanhamento e a avaliação do projeto prático proposto a ser desenvolvido em grupo, assim como foi estabelecido o foco do projeto, se extensão ou pesquisa. Nas semanas de desenvolvimento do projeto, o primeiro encontro (E1) de cada semana foi dedicado à orientação e acompanhamento do grupo, enquanto que no segundo encontro (E2) de cada semana o grupo deveria entregar uma parte do trabalho (entrega parcial semanal).

O método de avaliação da disciplina levou em conta todas as entregas parciais de projeto de cada unidade temática, conjuntamente com a entrega final do projeto ao fim da unidade. Às entregas parciais foi atribuído peso 4 e às entregas finais de cada unidade peso 6.

A Figura 1 mostra o processo e o fluxo entre os elementos no planejamento macro. Tem-se a professora ministrando algumas aulas expositivas e os alunos organizados em grupo, com acompanhamento da professora, para desenvolver os projetos em unidades temáticas específicas, com níveis de autonomia crescente e com naturezas de resolução diferentes, envolvendo pesquisa e extensão, indissociados do ensino.

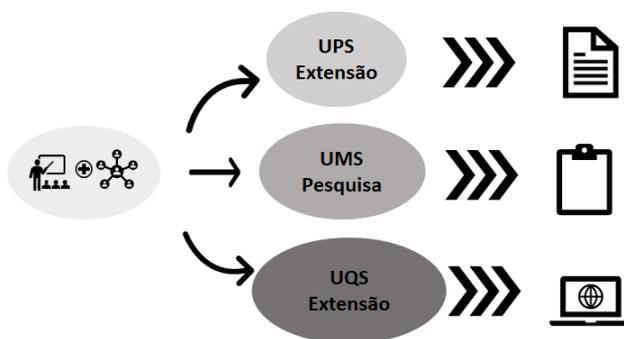


Figura 1. Elementos do Planejamento da Disciplina

2.1.1. Planejamento da Unidade Processo de Software (UPS)

As aulas expositivas desta unidade deveriam abordar conceitos fundamentais sobre processo de software e os principais modelos de processo de desenvolvimento de software. Os requisitos do projeto prático foram pensados de forma que o grupo de alunos pudesse aperfeiçoar e desenvolver novos conhecimentos e habilidades relacionadas à modelagem de processo de desenvolvimento de software, em um ambiente real de desenvolvimento

Tabela 1. Planejamento da Unidade Processo de Software

Semana	Atividade Planejada
1	E1: Apresentação do plano de ensino. E2: Aula expositiva: Fundamentos de processo de software.
2	E1: Aula expositiva: Processos de software prescritivos. E2: Aula expositiva: Processos de software ágeis: XP.
3	E1: Aula expositiva: Processos de software ágeis: Scrum. E2: Aula expositiva: Processos de software ágeis: Scrum. E2: Apresentação dos requisitos do projeto a ser desenvolvido na unidade.
4	E1: Acompanhamento do projeto 1. E2: Entrega parcial 1.
5	E1: Acompanhamento do projeto 1. E2: Entrega parcial 2.
6	E1: Acompanhamento do projeto 1. E2: Entrega e apresentação do projeto 1.

de software. O foco do projeto deveria ser na prestação de serviços, uma das possíveis ações de extensão previstas na Política Nacional de Extensão. A Tabela 1 mostra o planejamento semanal desta unidade.

Os requisitos do projeto foram estabelecidos a partir do levantamento prévio das experiências de atuação profissional dos estudantes. No grupo de estudantes, apenas um (1) tinha experiência profissional em uma empresa do segmento de software. Dessa forma, foi proposto o seguinte desafio: **Identificar o modelo de processo de desenvolvimento de uma empresa do segmento de desenvolvimento de software, realizar a modelagem do processo atual (AS-IS), identificar junto a equipe de desenvolvimento as oportunidades de melhoria do processo e modelar o processo proposto (TO-BE)**. Os requisitos foram apresentados somente no último encontro da semana 3.

Para o primeiro desafio, também foram definidos os requisitos das entregas parciais. Na **entrega parcial 1 (semana 4)** o grupo deveria entregar um documento contendo: I) Perfil da empresa (nome da empresa, segmento de desenvolvimento, número de produtos de software em desenvolvimento e em manutenção, perfil da equipe de desenvolvimento com o número de pessoas e funções); II) Características do processo de desenvolvimento atual, indicando qual o tipo de modelo de processo de desenvolvimento (tradicional/prescritivo, ágil), qual o fluxo do processo (linear, incremental, etc), quais as atividades do processo, quais as pessoas envolvidas, quais os artefatos gerados. Deveria ser desenvolvida a modelagem atual (AS-IS) do processo utilizando a notação *Business Process Modeling Notation (BPMN)* ou o diagrama de atividades da UML; III) Análise preliminar do processo atual de desenvolvimento.

Na **entrega parcial 2 (semana 5)** o grupo deveria entregar um documento estendido, contendo: I) Identificação das oportunidades de melhoria do processo de desenvolvimento; II) Modelagem do novo processo (TO-BE). Na última semana o grupo deveria apresentar o trabalho.

2.1.2. Planejamento da Unidade de Manutenção de Software (UMS)

Para esta unidade, foram planejadas aulas expositivas sobre conceitos fundamentais de manutenção de software, tipos de manutenção, processos de manutenção, engenharia reversa, reengenharia de software e gerenciamento de sistemas legados. O desafio prático

Tabela 2. Planejamento da Unidade Manutenção de Software

Semana	Atividade Planejada
7	E1: Aula expositiva: Fundamentos sobre manutenção de software. Tipos de manutenção de software. Processo de manutenção. E2: Aula expositiva: Engenharia Reversa de Software.
8	E1: Aula expositiva: Reengenharia de Software. E2: Aula expositiva: Sistemas Legados. E2: Apresentação dos requisitos do projeto a ser desenvolvido na unidade.
9	E1: Acompanhamento do projeto 2. E2: Entrega parcial 1.
10	E1: Acompanhamento do projeto 2. E2: Entrega parcial 2.
11	E1: Acompanhamento do projeto 2. E2: Entrega parcial 3.
12	E1: Acompanhamento do projeto 2. E2: Entrega e apresentação do projeto 2.

proposto para esta unidade buscou envolver várias empresas da indústria de software da região em uma atividade de pesquisa.

Dentre os tópicos estudados, optou-se pelo gerenciamento de sistemas legados, uma vez que o número de sistemas que demandam manutenção e, no futuro, serão considerados legados, cresce exponencialmente. Logo, é importante entender como esse tipo de sistema é tratado pelas empresas.

O desafio foi concebido de forma que o grupo pudesse desenvolver atividades típicas de um projeto de pesquisa com coleta de dados (*survey*). Assim, o seguinte desafio foi proposto para esta unidade: **Investigar como ocorre a tomada a decisão em relação ao que fazer com os sistemas legados mantidos pelas empresas**. Neste desafio as etapas metodológicas e respectivas entregas deveriam ser definidas pelo grupo, com a concordância da professora. As semanas 9, 10, 11 e 12 foram dedicadas à execução, acompanhamento e avaliação do projeto.

2.1.3. Planejamento da Unidade Qualidade de Software (UQS)

Nesta unidade foram planejadas aulas expositivas sobre inspeção de software, teste de software, qualidade de produto e qualidade de processo de software. Desta vez, a identificação do desafio a ser resolvido deveria partir do grupo de alunos. O único requisito que o grupo deveria cumprir era pensar em uma abordagem de solução que envolvesse o compartilhamento de conhecimentos para a comunidade acerca do tema escolhido, caracterizando dessa forma, uma ação de extensão. A Tabela 3 mostra o planejamento das aulas semanalmente.

2.2. Execução da Disciplina

A execução da disciplina buscou seguir o planejamento apresentado nas seções anteriores. No entanto, pequenos ajustes precisaram ser realizados nas semanas dedicadas aos

Tabela 3. Planejamento da Unidade Qualidade de Software

Semana	Atividade Planejada
13	E1: Aula expositiva: Qualidade Software. Inspeção de Software. E2: Aula expositiva: Teste de Software.
14	E1: Aula expositiva: Modelo de Qualidade de Produto. Medição de Qualidade.
15	E1: Aula expositiva: Modelos de maturidade: CMMI e MPS.BR E2: Entregar parcial 1.
16	E1: Acompanhamento do projeto 3. E2: Entregar parcial 2.
17	E1: Acompanhamento do projeto 3. E2: Apresentação do projeto 3.

projetos práticos. Nas subseções a seguir são apresentados os resultados gerados pelo desenvolvimento dos projetos propostos nas três unidades temáticas.

2.2.1. Unidade Processo de Software (UPS)

Para o desenvolvimento do projeto desta unidade o grupo escolheu uma empresa onde um dos integrantes já havia atuado, devido ao conhecimento pré-existente da empresa, do processo de desenvolvimento utilizado, assim como a facilidade de estabelecer o contato, quando necessário.

Para atender os requisitos da **entrega parcial 1** (subseção 2.1.1) o grupo levantou dados para caracterizar o perfil da empresa e do processo de desenvolvimento atual. A empresa atua no ramo de cursos profissionalizantes e atualmente conta com mais de 600 franquias e mais de 2 milhões de alunos. Por ser uma empresa na modalidade *franchising*, toda estrutura tecnológica relacionada à software oferecida às franquias é criada e atualizada pela equipe de TI da franqueadora. A equipe desenvolve sistemas de ERP e de EAD, sendo formada por 8 desenvolvedores, 2 gerentes de projeto, um especialista em UX/UI e 1 técnico responsável pelo suporte.

Sobre o processo de desenvolvimento atual, o grupo observou que a empresa adota um processo de desenvolvimento ágil, atendendo as demandas de manutenção e evolução dos sistemas existentes. No entanto, não existe um fluxo explicitado do processo de desenvolvimento. A modelagem do processo atual (AS-IS) ¹ identificou as atividades realizadas, quem as realiza e os artefatos gerados. A análise deste mapeamento inicial revelou que algumas práticas ágeis, como os testes unitários, não são realizadas.

A quantidade de requisitos solicitados nesta primeira entrega parcial não se mostrou factível no tempo disponível, sendo, portanto, adiada para o encontro seguinte, na semana posterior e assim sucessivamente.

Na **entrega parcial 2** o grupo identificou as oportunidades de melhoria do processo de desenvolvimento. O uso de testes unitários, um processo de testes de aceitação por parte do cliente, a integração contínua e a reunião de retrospectiva, foram práticas

¹Modelo do processo de desenvolvimento atual: shorturl.at/hkCM2

identificadas e devidamente justificadas no documento desta entrega. Dessa forma, foram adicionadas ao novo processo de desenvolvimento proposto (TO-BE) ².

2.2.2. Unidade Manutenção de Software (UMS)

Nesta unidade as etapas metodológicas do *survey* foram definidas pelo grupo, sob a orientação da professora, da seguinte forma: I) Elaboração do instrumento de coleta de dados; II) Identificação e contato com as empresas; III) Aplicação do instrumento de coleta de dados; IV) Análise dos resultados e V) Divulgação dos resultados.

Na **entrega parcial 1** (subseção 2.1.2) o grupo entregou artefatos correspondentes às etapas I e II. O instrumento de coleta de dados³ foi elaborado de forma a verificar como as empresas gerenciam os sistemas legados que dão manutenção. Foram exploradas questões demográficas e questões abertas relativas ao entendimento do que caracteriza esse tipo de sistema, quais as principais dificuldades de manutenção enfrentadas e, mais importante, quem toma e como são tomadas as decisões em relação às abordagens possíveis a esse tipo de sistema (reengenharia, substituições, aposentaria, etc). Também foi apresentada uma lista com aproximadamente 10 empresas do segmento de software da região.

Na **entrega parcial 2** o grupo não conseguiu apresentar as evidências da coleta de dados pela dificuldade em obter respostas das empresas. Das 10 empresas contactadas, apenas duas (2) aceitaram participar da coleta de dados, uma de forma presencial (entrevista) e outra através do preenchimento do formulário. Assim, somente na **entrega parcial 3** foi possível apresentar as evidências e analisar as respostas.

Os resultados, documentados por meio de um artigo longo (12 páginas)⁴ revelaram, dentre outros, que: 1) ambas as empresas caracterizam Sistema Legado (SL) como aqueles desenvolvidos com ferramentas ou linguagens antigas; 2) existem desafios na alocação de pessoas para manutenção por causa da dificuldade no entendimento das regras de negócios e do código/linguagem utilizada, que uma má decisão de alocação pode tornar o código extremamente difícil de manter, que manter pessoas alocadas somente nesse tipo de manutenção é desmotivador, que a programação em pares é uma boa solução; 3) o tempo e os recursos dispendidos são critérios que embasam decisões acerca desse tipo de sistema.

2.2.3. Unidade Qualidade de Software (UQS)

No projeto prático desta unidade o grupo teve total autonomia para escolher o foco de desenvolvimento do seu interesse. A escolha foi pesquisar sobre o processo de medição de qualidade de produto de software, com foco na série de normas sobre requisitos e avaliação da qualidade de produto de software ISO/IEC 25000 SQuaRE. Como o único requisito solicitado era o compartilhamento do conteúdo publicamente, o grupo propôs criar um curso *online* em uma plataforma EAD.

²Modelo do processo de desenvolvimento proposto: shorturl.at/gvyAN

³Formulário de coleta de dados: shorturl.at/jwyDH

⁴Link para o artigo completo: shorturl.at/tyP08

Na **entrega parcial 1** (subseção 2.1.3) o grupo entregou um documento compilado com informações sobre o conjunto de normas SQuaRE, com foco na norma 25020 - Guia e modelo de referência para medição. Na **entrega parcial 2** foi entregue o planejamento final do curso, incluindo a plataforma EAD que seria utilizada. A ferramenta escolhida foi a InTechEdu. O curso⁵ foi concebido de forma que o aluno passe por um percurso formativo composto por quatro (4) unidades de estudo, onde só é possível para passar para a próxima unidade se o aluno receber um emblema na unidade imediatamente anterior, mas para recebê-lo deve acertar todas as questões do questionário.

3. Avaliação da Experiência

A avaliação desta experiência de ensino foi realizada na perspectiva do grupo de alunos e da professora da disciplina.

3.1. Perspectiva dos Alunos

Nesta perspectiva, os alunos que concluíram a disciplina, avaliaram a metodologia da disciplina, assim como alguns aspectos dos desafios propostos. A seguir, seguem as perguntas e respostas, ora de forma *ipsis litteris*, ora sintetizadas.

1) Você considera que a metodologia utilizada na disciplina, com aulas expositivas e projetos práticos envolvendo problemas reais de empresas foi adequada? Sim, Não e Porquê.

Aluno 1: Sim, a didática ficou mais fluída, o que nos auxiliou foi os constantes feedbacks sobre o andamento dos projetos, a troca de informações em sala de aula do professor com os alunos, além dos projetos práticos que foram diferentes do habitual.

Aluno 2: Foi adequada, principalmente por ser uma experiência diferente para os alunos. O caráter mais prático e com constantes acompanhamentos para ver a evolução e com sugestão de melhoramentos em cada projeto foi algo positivamente diferente, levando a ações mais complexas a serem feitas por parte dos alunos.

2) Dos três projetos propostos avalie os aspectos positivos e negativos de cada um deles.

O projeto de análise e modelagem do processo de desenvolvimento da empresa teve como **pontos positivos**: o uso de ferramenta de modelagem de processo Bizagi, o trabalho em equipe para entendimento dos processos que levariam à modelagem, os *feedbacks* sobre a melhoria do processo de desenvolvimento de software. O Aluno 1 ainda relatou que no seu trabalho atualmente, todo fluxo do suporte foi feito utilizando a ferramenta Bizagi, o que ajudou a compreender melhor o fluxo de atendimento. Já os **pontos negativos** apontados foram: não apresentar a proposta da modelagem para a empresa. O Aluno 2, que não conhecia a empresa, destacou a falta de conhecimento dos processos da empresa.

O projeto de pesquisa sobre gerenciamento de sistemas legados nas empresas teve como **ponto positivo** as interações com as empresas e colaboradores e como **pontos negativos** o baixo interesse por parte da maioria das empresas contatadas em participar, assim como o tempo para desenvolver o projeto.

⁵<https://intecedu.rexlab.ufsc.br/course/view.php?id=496> login: es_user senha: eng_soft2

O projeto do curso online sobre atributos e avaliação da qualidade de produto de software foi considerado **positivamente** um projeto diferente, que trouxe outras habilidades e competências. Como **ponto negativo** foi indicado o prazo exíguo, diante do desafio de planejar um curso adequado, com um conteúdo complexo, a um público especializado externo.

3) Qual dos projetos propostos MAIS contribuiu para a formação de seus conhecimentos e habilidades e porquê?

Aluno 1: O projeto do sistema legados foi sem dúvidas o que mais contribuiu, pois conhecer uma empresa e dialogar com/entrevistar os colaboradores foi uma experiência que nos ajudou a entender bem o assunto. Logo após o diálogo, conversamos em sala de aula sobre algumas divergências sobre o entendimento dos colaboradores sobre o que é um sistema legado. Toda a entrevista foi transcrita e tudo foi documentado em um artigo.

Aluno 2: A modelagem do processo de desenvolvimento da empresa, pois conheci diversos novos termos, conceitos e apliquei uma ferramenta diferente de modelagem em uma área nova para mim. Foi o projeto que considero mais complexo em ações, com quantidade e importância de conteúdos abordados e utilizados.

4) Qual deles MENOS contribuiu para a formação de seus conhecimentos e habilidades e porquê?

Aqui ambos os alunos apontaram o projeto do curso de qualidade de software. Argumentaram que apesar ser "um tema incrível" e pouco conhecido e utilizado nas empresas da região, faltou tempo para elaborar um conteúdo melhor.

5) De uma forma geral, o que você melhoraria na proposta de metodologia da disciplina?

Ambos os alunos mencionaram o fator tempo como dificultador de uma metodologia com vários desafios/projetos. O Aluno 1 sugeriu que o número de projetos seja menor, de forma que o tempo dedicado a cada projeto seja maior. Já o Aluno 2 destacou que o contato com as empresas deve ser pré-arranjado visto que muitas demoram para responder ou não possuem interesse em colaborar com os alunos da universidade.

3.2. Perspectiva da Professora

Nesta perspectiva foi avaliado o que funcionou bem e o que pode ser considerado um desafio para a replicação da experiência. Primeiramente, entende-se que a abordagem de ensino mista, com algumas aulas expositivas e outras dedicadas ao desenvolvimento do projeto foi adequada ao perfil dos alunos deste curso, que é ofertado no período noturno e não dispõe da dedicação integral dos alunos.

O contato com a realidade de empresas do segmento de software da região permitiu aos alunos a aplicação dos conceitos estudados, proporcionando uma experiência de aprendizagem mais significativa, assim como uma ampliação da rede de conexões, que é muito valiosa, tanto para os estudantes como para as empresas que estão cada vez mais em busca de recursos humanos qualificados.

A forma como os desafios de projeto de cada unidade foram planejados, demandou uma autonomia paulatina e crescente dos alunos, com eles se acostumando a tomar decisões aos poucos. A natureza extensionista e de pesquisa dos desafios ofereceu aos

alunos oportunidades ricas para entender a relação ensino, pesquisa e extensão. O acompanhamento semanal dos projetos e as discussões, como mencionado pelos alunos, resultaram em uma troca de conhecimentos mais ativa e motivadora, para todas as partes.

Como desafios, destaca-se primeiramente a indisponibilidade de colaboração de algumas empresas do segmento de software contactadas. Foram realizadas diversas tentativas de contato, por meio de *e-mails* bem redigidos onde os alunos apresentavam o objetivo do trabalho e indagavam sobre o interesse e a disponibilidade da empresa em colaborar com o projeto. Algumas empresas responderam o contato inicial, mas não os contatos seguintes, e somente duas empresas realmente se envolveram nos projetos. Entende-se que um possível motivo seja a alta demanda e pressão que a indústria de software notoriamente enfrenta.

Por fim, entende-se que o tempo destinado ao desenvolvimento e orientação dos projetos pode ser considerado o maior dos desafios. Destinar 3 a 4 semanas para a execução e o acompanhamento de projetos com objetivos de aprendizagem diversos, em colaboração com atores externos e com orientação de qualidade foi desafiador, ainda que factível. Em um cenário regular, com uma turma maior, pode demandar alguns ajustes.

4. Considerações Finais

A partir dos desafios relatados, são sugeridas algumas ações para viabilizar experiências desse tipo. Em relação a indisponibilidade de empresa, sugere-se que o contato seja realizado previamente, pelo(a) professor(a) da disciplina, que pode apresentar a ideia de colaboração universidade-empresa de forma mais detalhada, obtendo com isso um número maior de empresas dispostas a colaborar. Quanto ao tempo exíguo para desenvolvimento e acompanhamento dos projetos sugere-se diminuir o número de projetos de forma a aumentar o tempo de acompanhamento de qualidade, ainda que isso gere um outro desafio, que é o planejamento de problemas que englobem vários tópicos de estudo. Convidar alunos egressos com experiência na indústria para tutorar os grupos conjuntamente com o professor, pode ser uma outra alternativa para o acompanhamento de qualidade.

O planejamento e a condução de experiências de ensino desse tipo demandam a saída de uma "zona de conforto" do professor, que precisa pensar em muitos aspectos para proporcionar uma experiência organizada e significativa para todos os envolvidos. A curricularização da extensão pode ser um estímulo para que experiências como esta possam se multiplicar.

Entende-se que esta experiência foi uma excelente oportunidade para ressignificar o ensino aprendizagem de tópicos de engenharia de software por meio de projetos práticos de caráter extensionista e de pesquisa que atendam necessidades reais de empresas do segmento de software.

A relação entre ensino, pesquisa e extensão coloca o estudante como protagonista de sua formação técnica e de sua formação cidadã, assim como permite que variadas possibilidades de articulação entre a universidade e a sociedade aconteçam. O detalhamento da experiência relatada, assim como a disponibilização dos artefatos resultantes de cada projeto, permitem a replicação da experiência em contextos semelhantes.

Referências

- Brasil (2012). Política nacional de extensão universitária.
- Brasil (2018). Resolução cne/ces nº 7, de 18 de dezembro de 2018. estabelece as diretrizes para a extensão na educação superior brasileira.
- Bruegge, B., Krusche, S., and Alperowitz, L. (2015). Software engineering project courses with industrial clients. *ACM Trans. Comput. Educ.*, 15(4).
- Gray, D. E. (2012). *Pesquisa no Mundo Real*. Editora Penso, Rio de Janeiro, RJ, 2ed. edition.
- Lima, J., Júnior, M. A., Moya, A., Almeida, R., Anjos, P., Lencastre, M., Fagundes, R., and Alencar, F. (2019). As metodologias ativas e o ensino em engenharia de software: uma revisão sistemática da literatura. In *Anais do XXV Workshop de Informática na Escola*, pages 1014–1023, Porto Alegre, RS, Brasil. SBC.
- Lopes, J., Medeiros, G., Fialho, D., and Bordin, A. (2017). Resolução de problemas no curso de engenharia de software: Uma experiência envolvendo extensão e ensino. In *Anais da I Escola Regional de Engenharia de Software*, pages 97–104, Porto Alegre, RS, Brasil. SBC.
- Marques, M. R., Quispe, A., and Ochoa, S. F. (2014). A systematic mapping study on practical approaches to teaching software engineering. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, pages 1–8.
- Martin, J. G., López, C. L., and Martínez, J. E. P. (2014). Supporting the design and development of project based learning courses. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, pages 1–6.
- Ministério da Educação (2016). Resolução nº 5, de 16 de novembro de 2016. institui as diretrizes curriculares nacionais para os cursos de graduação na área da computação.
- Raibulet, C. and Lago, P. (2022). Industrial project-based course on service oriented design: Experience sharing. *DREE '22*, page 20–24, New York, NY, USA. Association for Computing Machinery.
- Watson, E. M. and Cutting, D. (2022). Engagement contexts of software engineering education projects. In *2022 31st Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE)*, pages 1–6.
- Zorzo, A. F., Nunes, D., Matos, E., Steinmacher, I. and Leite, J., Araujo, R. M. and Correia, R., and Martins, S. (2017). Referenciais de formação para os cursos de graduação em computação.