

Ensino de Engenharia de Software em um Curso de Sistemas de Informação: Uma Análise dos Problemas e Soluções na Perspectiva de Professores e Alunos

Wellington L. Lemos¹, José Adson O. G. da Cunha¹, Juliana de A. G. Saraiva¹

¹Curso de Bacharelado em Sistemas de Informação – Universidade Federal da Paraíba

(UFPB) – Campus IV

58280-990 – Rio Tinto – PB – Brasil

wellington.lourenco@dcx.ufpb.br, adson@dcx.ufpb.br,
julianajags@dcx.ufpb.br

Abstract. *The area of Software Engineering (ES) consists of several sub-areas and is usually taught in disciplines with a dense theoretical-conceptual content distributed in the curricular matrix of undergraduate courses. In this context, several researchers have sought to create a motivating environment by proposing ES teaching techniques, tools and experiences. Based on semi-structured interviews with five Professors and eighteen students from disciplines related to the ES of an undergraduate course in Information Systems, this work presents an analysis of problems and solutions from the perspective of faculty and students in a Federal Institution. Despite being based on a single case study, the results generate reflections on the teaching of Software Engineering in courses in the area of computing.*

Resumo. *A área de Engenharia de Software (ES) é constituída por várias subáreas e é normalmente lecionada em disciplinas com um denso conteúdo teórico-conceitual distribuídas na matriz curricular de cursos de graduação. Nesse contexto, vários pesquisadores têm buscado criar um ambiente motivador através da proposição de técnicas, ferramentas e experiências de ensino de ES. Baseado em entrevistas semiestruturadas com cinco professores e dezoito alunos de disciplinas relacionadas à ES de um curso de graduação em Sistemas de Informação, este trabalho apresenta uma análise de problemas e soluções na perspectiva do corpo docente e discente em uma Instituição Federal de Ensino Superior. Apesar de ser baseado em um único estudo de caso, os resultados geram reflexões sobre o ensino de Engenharia de Software em cursos na área de computação.*

1. Introdução

Dentre as disciplinas de cursos na área de Computação, a Engenharia de Software (ES) apresenta-se como uma daquelas de maior relevância [ACM/IEEE 2013]. Apesar disso, a indústria de software se queixa de que os cursos de graduação não ensinam as competências necessárias para que os alunos desempenhem suas atividades com eficiência [Wangenheim e Silva 2009].

Diante da influência de aspectos didáticos e pedagógicos no aprendizado de ES e suas subáreas, vários pesquisadores têm proposto diferentes formas de estimular o pensamento crítico, a habilidade de solução e a aprendizagem de conceitos fundamentais. A partir de um mapeamento sistemático da literatura sobre o ensino de ES no Brasil, Cunha et al. (2018) identificaram 113 artigos categorizados em jogos, metodologias de ensino, ferramentas e integração entre disciplinas. Os resultados demonstram a predominância de trabalhos relacionados a jogos e metodologias de ensino. Apesar da importância do primeiro, como objeto de aprendizagem alternativa às aulas expositivas tradicionais, os trabalhos não têm disponibilizado links para acesso e uso dos jogos por outros professores. Poucos trabalhos foram publicados referentes à integração entre disciplinas das matrizes curriculares.

Apesar da importância de tais propostas para a melhoria do ensino de ES, é importante analisar o contexto dentro do qual o curso é ministrado. Tal necessidade é reforçada nos Referenciais de Formação para os Cursos de Computação da SBC para elaboração dos Projetos Pedagógicos dos Cursos (PPCs) [Zorzo et al. 2017], cabendo a cada curso definir a sua própria estratégia considerando o contexto regional, objetivos do curso, competências do corpo docente, dentre outros fatores.

Este trabalho tem como objetivo mapear os problemas e soluções no ensino-aprendizagem de ES sob o ponto de vista de professores e alunos de um curso de Bacharelado em Sistemas de Informação de uma Instituição Federal de Ensino Superior. Para tanto, foram realizadas entrevistas semiestruturadas com cinco professores e dezoito alunos de disciplinas relacionadas à Engenharia de Software da matriz curricular do curso. Os resultados apresentados são importantes para gerar reflexões sobre as necessidades de quem leciona e de quem aprende tendo em vista uma melhor assimilação do conteúdo e aperfeiçoamento das competências.

As demais seções deste trabalho estão estruturadas da seguinte forma: A Seção 2 apresenta os trabalhos relacionados. A Seção 3 descreve a metodologia utilizada. A Seção 4 expõe os resultados da pesquisa. Na Seção 5 são apresentadas as ameaças à validade. Por fim, a Seção 6 apresenta as conclusões.

2. Trabalhos Relacionados

Diante do volume de conteúdo relacionado à Engenharia de Software e das exigências de um mercado cada vez mais dinâmico, alguns pesquisadores têm identificado as dificuldades no processo de ensino-aprendizagem para compreender e buscar estratégias que preparem os estudantes para as demandas da indústria de software.

Através de uma pesquisa de opinião com professores no Brasil, Ferreira et al. (2018) mapearam dificuldades no ensino de ES. Quanto à pedagogia, somada à falta de tempo, observou-se que a variedade de processos de ES, diversidade de tecnologias e a dificuldade em criar abstrações e representar projetos reais comprometem o ensino-aprendizagem de ES. Em relação às dificuldades com recursos, identificou-se a falta de recursos bibliográficos e aspectos relacionados ao uso do ferramental disponível. Os autores identificaram ainda dificuldades relacionadas aos alunos, referentes à falta de maturidade técnica, dificuldade em programar e dificuldade na engenharia de requisitos.

A partir de um *survey* sobre a aprendizagem de tópicos de Engenharia de Software recomendados pelos currículos de referência da ACM/IEEE e SBC, Portela et

al. (2015) concluíram que houve convergência em relação à preferência de professores e alunos por projetos, aulas de laboratório e discussão de casos práticos.

Prikladnicki et al. (2009) descrevem diversas experiências de ensino evitando as abordagens tradicionais. Dentre as lições sobre experiências de abordagens metodológicas com o foco no aluno, destacam que o uso de projetos de software para mesclar teoria e prática em disciplinas onde a experiência desempenha um papel fundamental; o uso de jogos e dinâmicas para facilitar a compreensão pelos participantes; e o uso de atividades lúdicas para que os alunos aprendam a lidar com o novo, o inesperado e as dificuldades comuns em projetos de software.

De forma complementar a tais estudos, o presente trabalho pretende apresentar uma análise de problemas e soluções na perspectiva de professores e alunos no ensino de Engenharia de Software em uma Instituição Federal de Ensino Superior.

3. Método de Pesquisa

A pesquisa é baseada em uma abordagem qualitativa [Merriam e Tisdell 2015] através de entrevistas semiestruturadas com professores e alunos. O conteúdo das entrevistas foi analisado e categorizado em problemas e soluções para posterior avaliação de acordo com a gravidade e prioridade.

3.1 Coleta de Dados

O estudo foi realizado em um campus de interior de uma Instituição Federal de Ensino Superior na qual é ministrado o curso de Sistemas de Informação. O curso é integral (manhã e tarde), possui uma entrada anual de 50 alunos e carga horária de 3000 horas/aula (200 créditos). O tempo mínimo para integralização curricular é de 8 e o máximo de 12 meses. A maioria dos alunos vêm de escola pública do interior de vários estados circunvizinhos.

A Figura 1 apresenta as disciplinas da matriz curricular referentes à Engenharia de Software e subáreas ofertadas por período. Cada disciplina é caracterizada pelo professor, representado pelo código (PR#), quantidade de vezes que ministrou a disciplina e o tempo da entrevista, além de três alunos com diferentes níveis de desempenho, caracterizados pelo código (AL#), período atual do aluno e tempo da entrevista. Os códigos dos professores e alunos foram embaralhados para impossibilitar a associação às disciplinas. O gênero dos termos utilizados nos trechos das entrevistas também foram alterados.

| P4 | P5 | P6 | P7 |
|--|---|---|---|
| PR01 7x 47'56" Engenharia de Software AL03 5º 12'24" AL08 7º 20'40" AL18 11º 15'53" | PR02 4x 47'56" Engenharia de Software Aplicada AL04 7º 10'48" AL16 7º 28'59" AL9 9º 19'15" | PR05 3x 29'51" Gerência de Projetos de Software AL06 9º 10'46" AL17 9º 17'51" AL11 9º 13'02" | PR03 3x 29'18" Gestão de Qualidade de Software AL02 9º 23'34" AL15 9º 13'53" AL01 10º 11'41" |
| PR04 2x 34:09 Modelagem de Dados AL05 5º 12'10" AL10 7º 10'39" AL13 7º 19'11" | | PR06 5x 19:42 Projeto Avançado de Software AL12 7º 12'24" AL14 7º 20'20" AL07 7º 11'40" | |

Figura 1. Disciplinas da matriz curricular

Os guias de entrevista, sendo um para professor e outro para aluno, foram compostos de perguntas abertas com o objetivo de explorar experiências e opiniões (ver Figura 2). Cada guia de entrevista foi refinada através de uma entrevista-piloto cujos dados foram descartados.

Para atender aos requisitos éticos deste tipo de pesquisa, cada participante foi informado sobre os objetivos da pesquisa e seus direitos a fim de garantir a confidencialidade dos dados fornecidos, o anonimato do participante e o direito de se retirar da pesquisa a qualquer momento [Merriam e Tisdell 2015]. A autorização foi concedida através de uma pergunta específica sobre seu acordo em participar desta pesquisa. As entrevistas foram realizadas nas próprias instalações da Instituição e através de teleconferência durante o período de 13 de março a 29 de abril de 2018. Todo o áudio das sessões de entrevista foi gravado com o consentimento dos participantes para posterior transcrição.

| Roteiro da Entrevista com Professores | Roteiro da Entrevista com Alunos |
|---|---|
| <p>Q1. [Background] Fale um pouco de você: sua formação e trajetória profissional.</p> <p>Q2. [Background] Há quanto tempo você trabalha nesta universidade?</p> <ul style="list-style-type: none"> Sondagem: Quais disciplinas você ministra? Há quanto tempo ministra cada uma? <p>Q3. [Experiência] Fale um pouco sobre seu método de ensino da disciplina.</p> <ul style="list-style-type: none"> Sondagem: Sua experiência prévia de mercado influencia no seu método de ensino? <p>Q4. [Opinião] Você enfrentou algum problema ao longo da disciplina para aplicar sua metodologia de ensino?</p> <ul style="list-style-type: none"> Sondagem: Quais soluções foram dadas para resolver tais problemas? <p>Q5. [Opinião] Quais são os desafios da disciplina na perspectiva de infraestrutura, método de ensino e corpo discente?</p> <ul style="list-style-type: none"> Sondagem: Esses desafios podem ser solucionados? Como? <p>Q6. [Opinião] Como você vê a integração entre as disciplinas relacionadas ao desenvolvimento de software?</p> <ul style="list-style-type: none"> Sondagem: O que tem impedido essa integração? <p>Q7. Você gostaria de adicionar alguma informação ou observação que não foi perguntada, mas que você considera importante para o ensino de disciplinas relacionadas à engenharia de software?</p> | <p>Q1. [Background] Fale um pouco sobre você: por que você escolheu esse curso?</p> <ul style="list-style-type: none"> Sondagem: Quais disciplinas você mais gostou? <p>Q2. [Opinião] Conte-me um pouco sobre sua situação atual no curso.</p> <ul style="list-style-type: none"> Sondagem: Em que período você se encontra? O que tem achado das disciplinas até o momento? <p>Q3. [Experiência] Fale um pouco sobre sua experiência na disciplina.</p> <ul style="list-style-type: none"> Sondagem: O que achou do assunto apresentado e metodologia de ensino? <p>Q4. [Opinião] Você enfrentou algum problema na disciplina?</p> <ul style="list-style-type: none"> Sondagem: Os assuntos apresentados foram bem assimilados? <p>Q5. [Experiência] Como você avalia seu desempenho na disciplina?</p> <ul style="list-style-type: none"> Sondagem: Você acredita que as disciplinas que você pagou ou deixou de pagar anteriormente a esta influenciaram no seu desempenho? <p>Q6. [Opinião] Como você avalia a integração entre disciplinas?</p> <ul style="list-style-type: none"> Sondagem: Você acredita que essa integração existe? O que está faltando? <p>Q7. Você gostaria de adicionar alguma informação ou observação que não foi perguntada, mas que você considera importante para o ensino de disciplinas relacionadas à engenharia de software?</p> |

Figura 2. Roteiros das entrevistas

3.2 Análise dos Dados

Os dados coletados foram analisados através do método de análise de conteúdo [Bardin 2013], composto por três fases: i) pré-análise; ii) exploração do material; e iii) tratamento dos resultados, inferência e interpretação.

O conteúdo das entrevistas foi analisado por meio da técnica Análise Categórica. Em tal abordagem os textos são agrupados em categorias. As seguintes etapas foram realizadas de acordo com o método de Análise de Conteúdo Temático-Categorial [Oliveira 2008]:

- **Preparação das informações:** Todas as entrevistas foram transcritas para serem posteriormente analisadas através da técnica de Análise de Conteúdo.
- **Transformação do conteúdo em unidades de registro (UR):** Nesta etapa determina-se a escolha do tipo de unidade de registro (palavra, frase ou parágrafo) que foi adotada pelo pesquisador no decorrer da análise. Neste trabalho, foi definido como UR os recortes do texto avaliados como relevantes para compreensão do objeto de estudo.
- **Classificação de unidades de registro em temas:** Cada unidade de registro é associada a um ou mais temas, com o propósito de associar URs dentro de um mesmo contexto de significação. Cada tema, portanto, é composto por uma ou mais URs. Os temas emergem durante o processo de análise, ou seja, não são pré-estabelecidos.
- **Classificação de temas em categorias:** Refere-se ao agrupamento dos temas e suas respectivas unidades de registro em categorias. Cada categoria está associada a um conjunto de temas, e cada tema à apenas uma categoria, conforme ilustrado na Figura 3.
- **Descrição:** Esta etapa corresponde ao resumo numérico das etapas anteriores com o objetivo de apresentar a soma das unidades de registro encontradas no texto de cada entrevista, por tema. Nesta pesquisa, a relevância dos problemas e soluções foi obtido a partir de um *member checking* com os participantes. Para cada problema, o respondente avaliou o nível de gravidade em baixo, médio ou alto. De modo semelhante, para cada solução, o respondente avaliou o nível de prioridade em baixo, médio ou alto.
- **Interpretação:** Para cada tema pertencente a uma categoria específica, o investigador deve fazer suas ponderações, a fim de elucidar a compreensão e a importância da existência do tema.

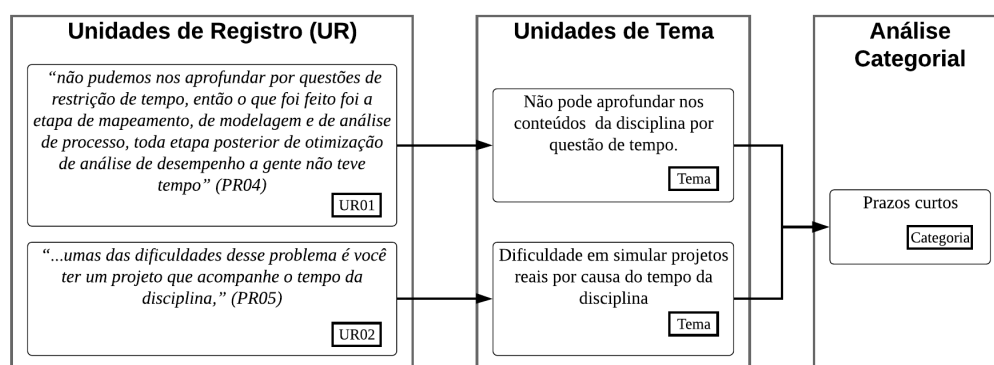


Figura 3. Exemplo Conteúdo Temático-Categorial

4. Resultados

Os problemas e soluções nas perspectivas de professores e alunos estão ilustrados nas Figuras 4 e 5, respectivamente. A numeração no canto inferior direito de cada categoria representa a relação da solução com o problema, quando explicitada.

4.1 Problemas

A Figura 4 apresenta os problemas mapeados, cada qual analisado e associado com uma Unidade de Registro (UR).

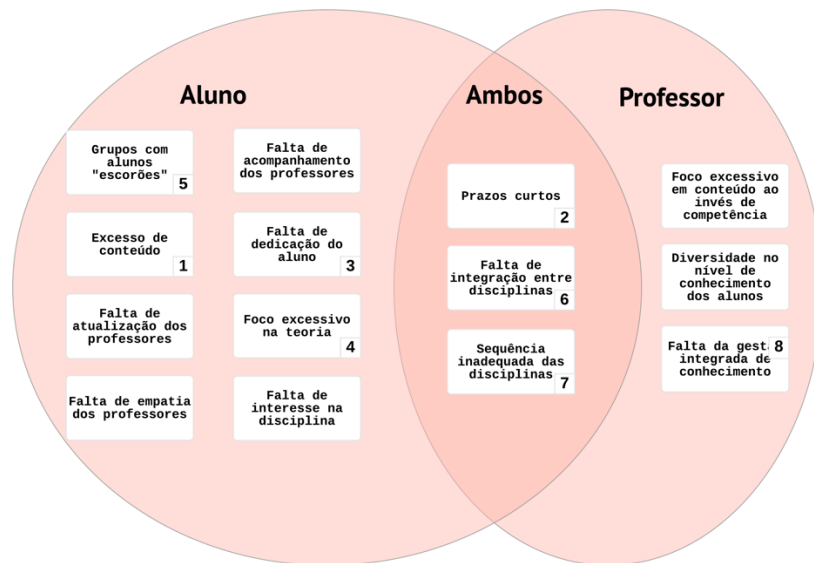


Figura 4. Modelo Representativo dos Problemas

4.1.1 Perspectiva dos Alunos

O processo de ensino-aprendizagem é composto pela colaboração mútua entre aluno e professor. Nesse sentido, independente da metodologia adotada pelo professor, caso não haja dedicação do aluno, o processo de apreensão e construção do conhecimento ficará comprometido. A “**falta de dedicação do aluno**” caracteriza tal consciência pelo aluno.

- “Não aprendi tudo por desleixo meu.” (AL13)

Dadas as várias subáreas da Engenharia de Software, é natural que o aluno apresente uma identificação por algumas específicas, em geral, divididas entre as que requerem algum tipo de programação ou não. A “**falta de interesse na disciplina**”, portanto, prejudica seu desempenho.

- “Eu não tenho tanto interesse na área de gerência, meio que essa falta de interesse dificulta quando você vai fazer as coisas.” (AL11)

O trabalho em equipe é um dos principais pilares em projetos de desenvolvimento de software [Dutra e Prikladnicki 2014]. Diante disso, é importante a existência de projetos para que os alunos desenvolvam as *soft skills* [Ahmed et al. 2015]. Em projetos de software reais, estudos indicam que o engajamento dos membros

da equipe influencia na motivação individual [França e Da Silva 2009]. Em projetos de software acadêmicos, a existência de “**grupos com alunos ‘escorões’**” pode influenciar no aprendizado individual.

- *“Tem aquelas pessoas que tá na equipe que desmotiva totalmente porque não faz nada, atrapalha e ainda passa como um bom aluno. É horrível.” (AL09)*

As várias subáreas da Engenharia de Software precisam ser bem dimensionadas e divididas entre diferentes disciplinas da matriz curricular. O “**excesso de conteúdo**” em uma mesma disciplina, sem tempo suficiente para melhor trabalhar cada tópico através de exercícios, pode comprometer o aprendizado do aluno. Além do excesso de conteúdo, o “**foco excessivo na teoria**”, com poucas atividades práticas, é outro fator que prejudica o aprendizado.

- *“Eu estudei atropelado pelo excesso de conteúdo. Bicho, seiscentos slides.” (AL08)*
- *“Eu acho que teria que existir uma maneira correta de aplicar essa teoria, tá ligado? Não ser assim, empurra e lê tantos capítulos pra próxima aula. Não funciona.” (AL17)*

Os alunos destacaram também problemas relacionados aos professores: “**falta de atualização**”, “**falta de acompanhamento**” e “**falta de empatia**”. Tais características servem como insumo para reflexão do corpo docente para se manter atualizado, incluindo os métodos e técnicas que têm sido utilizados pelo mercado; disponibilizar atendimento aos alunos em horário extraclasse; além de avaliar o contexto e situação de cada aluno para nortear sua conduta em sala de aula.

- *“Alguns professores da gente não evoluem, estão parados no tempo.” (AL08)*
- *“Eu acho que a professora deveria ter acompanhado mais de perto, pra poder focar mais, porque era mais solto.” (AL16)*
- *“Ele era arrogante. Isso desmotivava tanto, que quando eu tinha uma dívida, eu não perguntava mais.” (AL16)*

4.1.2 Perspectiva dos Professores

Os professores, por sua vez, relataram problemas relacionados aos alunos e estrutura das disciplinas. A “**diversidade no nível de conhecimento dos alunos**” é um fator complicador que impede o professor de avançar o conteúdo de forma satisfatória, fazendo-o expor menos conteúdo para melhor praticá-lo com a turma.

- *“Tem gente que sabe muito bem os requisitos para disciplina e até mais, e tem gente que sabe muito pouco. É sempre difícil dar uma nivelada na turma. (PR06)*

A competência é o comportamento que se espera observar no egresso do curso. Nesse sentido, os conteúdos específicos ministrados em unidades curriculares devem ser vistos como formas de mobilização das competências necessárias [Zorzo et al. 2017]. É importante que o professor tenha em mente a relação do conteúdo com a competência requerida. Sendo assim, o “**foco excessivo em conteúdo ao invés de competência**” foi apresentado como um fator prejudicial à formação.

- *“A gente tem excelentes professores com muito conteúdo, mas o método já não é mais adequado às demandas que um profissional lá fora exige.” (PR05)*

A “**falta de gestão integrada de conhecimento**” foi considerada um problema por dificultar uma visão, por parte de todo o corpo docente, do conteúdo ministrado nas

disciplinas, fazendo com que haja uma redundância de um determinado conteúdo em várias disciplinas. Por outro lado, conteúdos importantes deixam ser lecionados.

- “Por não ter uma ferramenta que forneça uma visão do que é ministrado em outras disciplinas, muitas vezes há uma repetição do conteúdo já ministrado em outras disciplinas.” (PR04)

4.1.3 Perspectiva de Ambos

A “**falta de integração entre disciplinas**” dificulta o aprendizado e compromete a visão sistêmica da Engenharia de Software pelo aluno. É importante que tal integração seja algo institucionalizado e comum entre várias disciplinas e não apenas como ações pontuais.

- “O que falta são disciplinas integrativas que formalizem a execução de projetos interdisciplinares.” (PR03)

Além da falta de integração, a “**sequência inadequada das disciplinas**” compromete a aquisição de conhecimentos considerados pré-requisitos. Especificamente em relação à Engenharia de Software, foi considerado que tal disciplina deva ser ministrada na primeira metade do curso.

- “A gente começa a lidar com projetinho de disciplina desde o primeiro período e só depois você vê como é que você vai aprender a fazer um projeto (considerando os aspectos da Engenharia de Software). Eu sou totalmente contra isso.” (AL17)

Por fim, diante das várias disciplinas cursadas em paralelo pelos alunos, os “**prazos curtos**” para entrega das atividades foram citados como um problema. Na perspectiva dos professores, o tempo influencia no aprofundamento de alguns conteúdos, sendo apresentados de forma parcial ou superficial.

- “Não pudemos nos aprofundar por questões de restrição de tempo.” (PR04)

4.2 Soluções Propostas

A Figura 5 apresenta os problemas mapeados, cada qual analisado e associado com uma Unidade de Registro (UR).

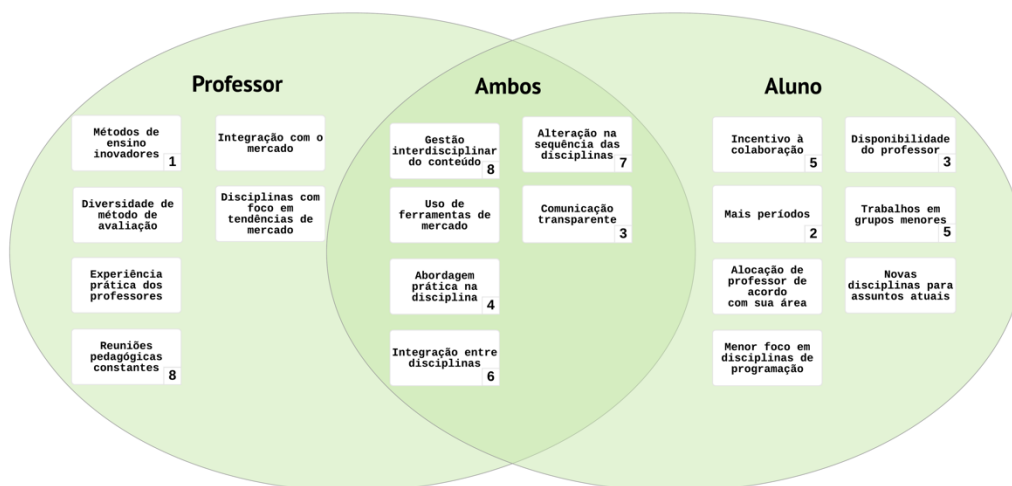


Figura 5. Modelo Representativo das Soluções

4.2.1 Perspectiva dos Alunos

Muitos alunos relataram problemas com trabalhos em grupo. Como forma de minimizar o impacto de alunos “escorões”, foram sugeridos “**trabalhos em grupos menores**”. Além disso, em tais grupos, é importante que o professor acompanhe de perto e fomente o “**incentivo à colaboração**”.

- “Os professores fazem grupos muito grandes de pessoas. Acharia melhor um grupo menor, acho que forçava mais o pessoal do grupo a desenvolver mais coisas.” (AL04)
- “Trabalhar a mentalidade das pessoas para que seja desenvolvida esse espírito colaborativo. Tanto pra quem oferece (ajuda), como para quem recebe.” (AL02)

Os alunos explicitaram também o desejo por “**novas disciplinas para assuntos atuais**”. Em Ferreira et al. (2018), os autores enfatizaram a necessidade de se focar em tópicos emergentes como Sistemas-de-Sistemas, Ecossistemas de Software, Engenharia de Software para Startups e Internet das Coisas. Especificamente na matriz curricular de um curso de Sistemas de Informação, é importante que haja uma distribuição equilibrada das disciplinas entre aquelas de cunho técnico e gerencial. Apesar da importância, os alunos indicaram a necessidade de “**menor foco em disciplinas de programação**” e um maior foco em disciplinas relacionadas à gestão, como empreendedorismo, gerenciamento de processos de negócios e afins.

- “Eu acho a grade da gente meio antiga já, tá meio defasada.” (AL03)
- “Nas próximas mudanças no curso que vierem, não focarem só em programar, porque como todo mundo sabe a área de sistemas de informação é muito ampla.” (AL18)

Cada professor tende a ter sua área específica de conhecimento, e, por isso, é sugerível que a alocação de disciplinas respeite tal premissa. Além da “**alocação de professor em disciplina de acordo com sua área**”, os alunos destacaram também a “**disponibilidade do professor**”, relacionado à empatia, para explicar pacientemente o conteúdo.

- “Muitas vezes ele (professor) não sabia nem o que a gente tava usando. Tinha que explicar o que era a ele.” (AL04)
- “Ele explicava muito bem. Quando a pessoa não entendia direito era só ir a ele e ele explicava.” (AL15)

Dada a quantidade de conteúdo, foi sugerida ainda reestruturação da matriz curricular em “**mais períodos**”.

- “Eu acho que o nosso curso deveria ter pelo menos 9 semestres.” (AL14)

4.2.2 Perspectiva dos Professores

Os professores destacaram a importância de “**disciplinas com foco em tendências de mercado**”. Além do conteúdo ministrado, há a necessidade da “**integração com o mercado**” através de visitas técnicas e palestras com profissionais da área, e utilização de “**métodos de ensino inovadores**” que fujam de aulas meramente expositivas.

- “Eu tô sempre ligado no que tá acontecendo, pra poder trazer algo que realmente eu sei que o aluno vai poder usar.” (PR06)
- “A gente fez visitas técnicas em todas as turmas. A gente foi pra Natal, pra Recife e João Pessoa.” (PR03)

- *“Hoje em dia tem vários métodos educacionais sendo discutidos no mundo, como sala invertida, inquiry-based learning. A gente tem que repensar esse formato que está aqui.” (PR05)*

A **“experiência prática dos professores”** é importante para dar maior peso ao que é exposto em sala de aula considerando o contexto de como tal conteúdo é aplicado na prática. A **“diversidade no método de avaliação”**, não concentrando-se apenas em provas escritas, oferece oportunidade de aprendizado para alunos com diferentes estilos de aprendizagem [França et al 2016].

- *“Eu tive que dar um pouco da minha cara à disciplina. Tem uma influência muito grande da experiência, da minha bagagem.” (PR04)*
- *“As duas primeiras unidades têm uma prova individual. A avaliação da terceira unidade é um projeto que perpassa todas as três unidades.” (PR04)*

As **“Reuniões pedagógicas constantes”** são um importante elemento para gerar uma maior colaboração, organização e novas ideias entre os docentes.

- *“Eu acho que as reuniões pedagógicas ajudam e deve ser mais rotineiro.” (PR06)*

4.2.3 Perspectiva de Ambos

A **“alteração na sequência das disciplinas”** foi indicada por professores e alunos para proporcionar um melhor rendimento dos alunos e menor desequilíbrio no nível de conhecimento nas turmas.

- *“Eu sugeriria deixar modelagem (processos de negócios, engenharia de requisitos e análise e projeto) para ver um semestre depois de engenharia de software para ver aquilo de uma forma mais aprofundada.” (PR04)*

A **“integração entre disciplinas”**, a **“abordagem prática na disciplina”**, a **“gestão interdisciplinar do conteúdo”**, e **“uso de ferramentas de mercado”** foram destacados por professores e alunos como essenciais no processo de ensino-aprendizagem.

- *“Foi boa (a integração) porque a gente pôde aprender mais avaliando os trabalhos da outra turma. A gente pôde aprender com os erros e acertos deles.” (AL15)*
- *“Eu vejo algumas disciplinas que não são tão práticas. É muita teoria. Você precisa saber a teoria, mas quando coloca em prática tudo aquilo que está aprendendo você fixa.” (AL07)*
- *“É preciso sistematizar a gestão do conhecimento do que é ministrado nas disciplinas.” (PR04)*
- *“A gente conheceu algumas ferramentas que são usadas no mercado.” (AL02)*

Por fim, a **“Comunicação transparente”** na forma de feedback constante sobre o desempenho e pontos de melhoria entre aluno e professor sobre todo e qualquer tipo de assunto.

- *“Sempre tive o costume de mandar um formulário para a galera me avaliar, porque eu acho que assim, o professor tem que tá sempre melhorando, e nada melhor do que os alunos para analisar se o que ele fez foi bom ou foi ruim.” (PR04)*

A Tabela 1 apresenta o mapeamento direto de parte das soluções com os problemas que emergiram das entrevistas. Dos problemas ilustrados na Figura 4, alguns foram elencados sem uma solução direta. Da mesma forma, algumas soluções ilustradas na Figura 5 não possuem um problema direto. Tais casos devem ser aprofundados em uma nova rodada de entrevistas.

Tabela 1. Problemas e soluções

| Problema | Solução |
|---|---|
| Excesso de conteúdo | Métodos de ensino inovadores |
| Prazos curtos | Mais períodos |
| Falta de dedicação do aluno | Comunicação transparente e Disponibilidade do professor |
| Foco excessivo na teoria | Abordagem prática na disciplina |
| Grupos com alunos “escorões” | Trabalhos em grupos menores |
| Falta de integração entre disciplinas | Integração entre disciplinas (Projetos interdisciplinares) |
| Sequência inadequada das disciplinas | Alteração na sequência das disciplinas |
| Falta de gestão integrada do conhecimento | Reuniões pedagógicas constantes e Gestão interdisciplinar do conhecimento |

4.3 Gravidade dos problemas e prioridade das soluções

Tendo em vista a validação do conteúdo mapeado, foi realizado um *member checking* com todos os participantes através da ferramenta *Google Forms*. Para cada problema ilustrado na Figura 4, o respondente avaliou o nível de gravidade em baixo, médio ou alto. De modo semelhante, para cada solução da Figura 5, o respondente avaliou o nível de prioridade em baixo, médio ou alto. Uma vez que professores e alunos são sujeitos ativos e de igual importância no processo de ensino-aprendizagem, não houve distinção entre eles na avaliação da gravidade dos problemas e prioridade das soluções.

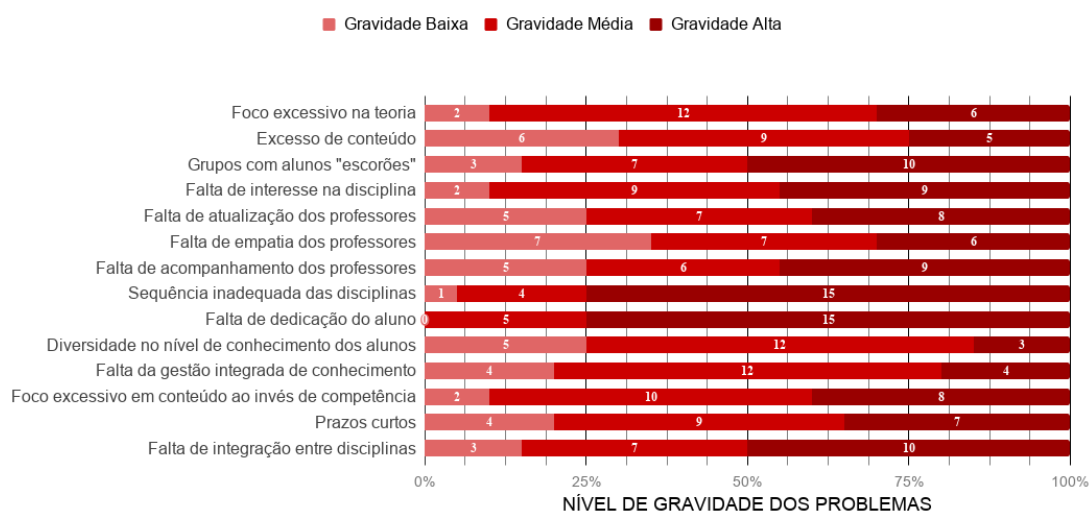


Figura 6. Avaliação da gravidade dos problemas

De acordo com a avaliação dos participantes, a “sequência inadequada das disciplinas”, “falta de dedicação do aluno”, “falta de integração entre disciplinas”, e

“grupos com alunos ‘escorões’” foram os problemas de maior gravidade. A menos grave foi a “diversidade no nível de conhecimento dos alunos”. Apesar destes resultados, quase todos tiveram níveis médios e altos de gravidade, demonstrando uma aceitação pelos problemas encontrados.

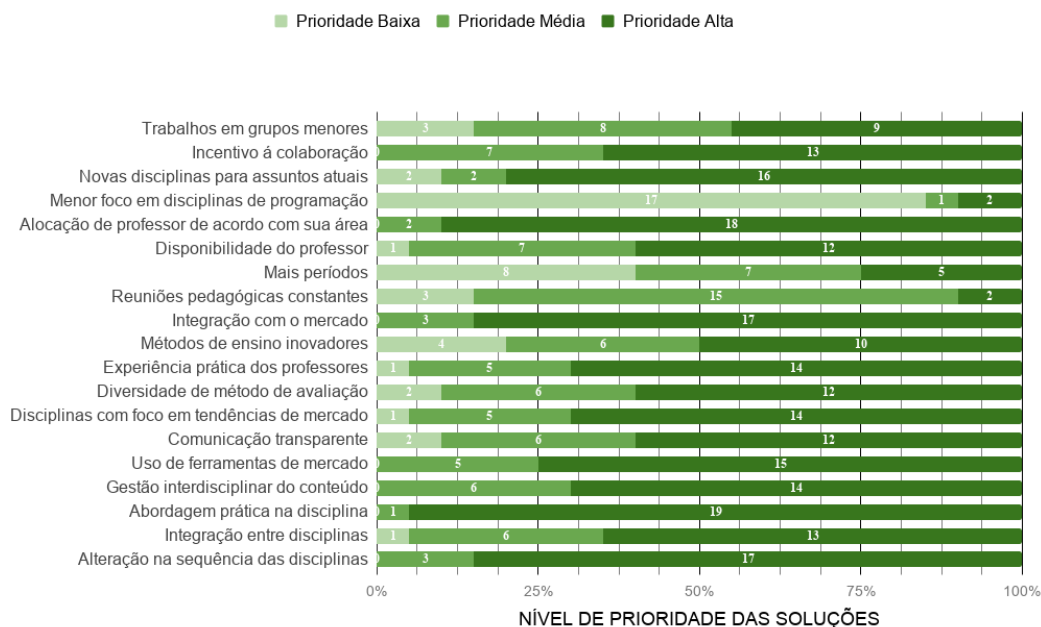


Figura 7. Avaliação da prioridade das soluções

As soluções com alto nível de prioridade foram: “abordagem prática na disciplina”, “alocação de professor de acordo com sua área”, “alteração na sequência das disciplinas”, e “novas disciplinas para assuntos atuais”. A solução de menor prioridade foi “menor foco em disciplinas de programação”. Da mesma forma que os problemas, a maioria das soluções tiveram níveis médios e altos de prioridade, demonstrando uma aprovação por parte dos entrevistados.

5. Ameaças à Validade

Uma vez transcritas, as entrevistas foram analisadas em várias rodadas para refinamento das unidades de tema e categorias. Considerando que cada participante foi entrevistado uma única vez, uma nova rodada de entrevistas para aprofundar os problemas e soluções levantados daria uma maior riqueza de detalhes ao fenômeno estudado.

Apesar de terem sido entrevistados três alunos com diferentes desempenhos em cada disciplina, uma amostra maior poderia trazer novos elementos. Além disso, entrevistas com professores e alunos de outras instituições contribuiria com novas perspectivas.

6. Considerações Finais

A abordagem qualitativa utilizada teve o propósito de mapear problemas e soluções no ensino de ES na perspectiva de professores e alunos. O estudo foi baseado em entrevistas semiestruturadas com cinco professores e dezoito alunos, sendo três por

disciplina. Uma vez transcritas, as entrevistas foram analisadas para definição das categorias temáticas.

Os problemas relacionados à forma como a matriz curricular é estruturada e as disciplinas são ministradas, o que inclui a “sequência inadequada das disciplinas”, “foco excessivo em conteúdo ao invés de competência”, “foco excessivo na teoria”, “excesso de conteúdo”, “falta da gestão integrada de conhecimento” e “prazos curtos” foram destacados como influenciadores negativos no ensino-aprendizagem de ES. Tais fatores fortalecem a importância do Núcleo Docente Estruturante na definição de um Projeto Pedagógico do Curso (PPC) aderente às necessidades de professores e alunos em relação ao ensino de Engenharia de Software. Além da estrutura curricular, os alunos e professores possuem papel importante no processo. Em relação ao corpo discente, foram mapeadas a “falta de dedicação do aluno”, “grupos com alunos ‘escorões’”, “falta de interesse na disciplina” e “diversidade no nível de conhecimento dos alunos”. Em relação aos professores, por sua vez, destacou-se a “falta de atualização”, falta de acompanhamento” e “falta de empatia”.

Como forma de maximizar o processo de ensino-aprendizagem de ES, a maioria dos fatores estão relacionadas à estrutura curricular e a forma como as disciplinas são ministradas: “abordagem prática na disciplina”, “alteração na sequência das disciplinas”, “integração com o mercado”, “novas disciplinas para assuntos atuais”, “uso de ferramentas de mercado”, “gestão interdisciplinar do conteúdo”, “disciplinas com foco em tendências de mercado”, “incentivo à colaboração”, “integração entre disciplinas”, “diversidade de método de avaliação”, “métodos de ensinados inovadores”, “trabalhos em grupos menores”, “alocação de professor de acordo com sua área”, “reuniões pedagógicas”, “mais períodos” e “menor foco em disciplinas de programação”. Além desses fatores, os professores têm papel importante através da sua “experiência prática”, “disponibilidade” e “comunicação transparente”.

Apesar de ser baseado em um único estudo de caso, os resultados apresentados geram reflexões sobre o ensino de Engenharia de Software em cursos na área de computação. Como trabalho futuro, pretende-se analisar a relação do problema/solução com cada disciplina a fim de verificar se o problema é pontual em uma disciplina ou transversal às demais disciplinas. Além disso, pretende-se mapear quais problemas os egressos encontram ao entrar no mercado de trabalho e como poderiam ser supridos ao longo da graduação.

Referências

- ACM/IEEE (2013) “Computer science curricula 2013 – Curriculum guidelines for undergraduate degree programs in Computer Science”, <https://www.acm.org>, April.
- Ahmed, F., Capretz, L. F., Bouktif, S., & Campbell, P. (2015). Soft skills and software development: A reflection from the software industry. arXiv preprint arXiv:1507.06873.
- Bardin, L. “Content analysis” Editions Lisbon, 3th edition, 2013.
- França, C. Cunha, J. A., Adjard, D., e Alan, F. (2016) Uma Investigação sobre Estilos de Aprendizagem e Hábitos de Estudo de Engenheiros de Software. IX Fórum de Educação em Engenharia de Software (FEES 2016) do XXX Simpósio Brasileiro de Engenharia de Software (SBES 2016).

- Cunha, J. A. O., Marques, G. A., Lemos, W. L., Câmara Jr, U. D., & Vasconcellos, F. J. (2018). Software engineering education in Brazil: a mapping study. In Proceedings of the XXXII Brazilian Symposium on Software Engineering (pp. 348-356). ACM.
- Dutra, A. C. S., & Prikladnicki, R. (2014). Formação de Equipes de Alto Desempenho Para Desenvolvimento de Software. FEES 2014, 70.
- Ferreira, T., Viana, D., Fernandes, J., & Santos, R. (2018). Identifying emerging topics and difficulties in software engineering education in Brazil. In Proceedings of the XXXII Brazilian Symposium on Software Engineering (pp. 230-239). ACM.
- Franca, A. C. C., & da Silva, F. Q. (2009). An empirical study on software engineers motivational factors. In 2009 3rd International Symposium on Empirical Software Engineering and Measurement (pp. 405-409). IEEE.
- Merriam, S. B., & Tisdell, E. J. (2015). Qualitative research: A guide to design and implementation. John Wiley & Sons.
- Oliveira, Denize Cristina de. (2008) "Análise de conteúdo temático-categorial: uma proposta de sistematização." *Rev. enferm. UERJ* 16.4.
- Portela, C. S., Vasconcelos, A. M., & Oliveira, S. R. (2015). Análise da Relevância dos Tópicos e da Efetividade das Abordagens para o Ensino de Engenharia de Software. In Fórum de Educação em Engenharia de Software (FEES). In VI Congresso Brasileiro de Software: Teoria e Prática (CBSOFT).
- Prikladnicki, R., Albuquerque, A. B., von Wangenheim, C. G., & Cabral, R. (2009). Ensino de engenharia de software: desafios, estratégias de ensino e lições aprendidas. FEES-Fórum de Educação em Engenharia de Software, 1-8.
- Wangenheim, C. e Silva, D. (2009). Qual Conhecimento de Engenharia de Software é Importante para um Profissional de Software? Em Anais do II Fórum de Educação em Engenharia de Software. Fortaleza, Brasil.
- Zorzo, A., Nunes, D., Matos, E., Steinmacher, I., Leite, J., Araujo, R., ... & Martins, S. (2017). Referenciais de formação para os cursos de graduação em computação. sociedade brasileira de computação (SBC). 153p. Technical report.