

CodeSmells? Aqui não! Limpando e Refatorando Códigos na Prática: Um Relato de Experiência da Execução do Curso CodeSmells na Prática

Pedro C. Chaaban¹, Robson Medeiros¹, Jorge Luiz Sousa¹, Marina Rocha¹, George Maia¹,
Israely Lima¹, Ricardo Davi Batista¹, Jacilane de H. Rabelo¹

¹Projeto de pesquisa e extensão LearningLab – Universidade Federal do Ceará (UFC - Campus Russas)

Caixa Postal 62.900-000 – Russas – CE – Brasil

{karinpedro, robsonmedeiros, cc.jorgesousa, marinarocha, georgemaiaf, israelylim, ricardodavi}¹@alu.ufc.br, {jacilane.rabelo}²@ufc.br

Abstract. *Computer science teaching approaches are aiming for more dynamic teaching. The work developed in this article aims to describe the course of Code Refactoring offered by the LearningLab project to students of the Russas Campus of the Universidade Federal do Ceará (UFC), which uses interactive teaching with a theoretical and practical base. The CodeSmells course for beginners was conducted face-to-face with 20 students from the Software Engineering and Computer Science courses. The results show, according to the teaching methodology applied and the perception of the students, that the majority enjoyed the practical activity and that more than 80% understood and would be able to apply in practice the concepts learned.*

Resumo. *Abordagens de ensino de Computação estão visando um ensino mais dinâmico. O trabalho desenvolvido neste artigo tem como objetivo descrever o curso de Refatoração de Códigos ofertado pelo projeto LearningLab aos estudantes do Campus Russas da Universidade Federal do Ceará (UFC), o qual utilizou o ensino interativo de base teórica e prática. O curso CodeSmells para iniciantes foi realizado presencialmente com 20 estudantes dos cursos de Engenharia de Software e Ciência da Computação. Os resultados mostram, de acordo com a metodologia de ensino aplicada e a percepção dos estudantes, que a maioria gostou da atividade prática e que mais de 80% compreendem e seriam capazes de aplicar na prática os conceitos aprendidos.*

1. Introdução

O ensino de Engenharia de Software (ES) propõe o desenvolvimento de habilidades como trabalho em equipe, aplicação de técnicas para gestão de projetos de software e desenvolvimento de diferentes soluções para diferentes domínios (Souza et al., 2021). Anomalias que podem indicar problemas nos aspectos da qualidade do código do software são chamadas de *code smells* (Fowler, 2018). As ocorrências desses problemas podem dificultar a manutenção do software e levar à necessidade da refatoração de código (Fontana et al., 2012). O trabalho de Mendes et al., (2020) retrata que a Manutenção de Software foi uma das três disciplinas de maior dificuldade no aprendizado, dada a forma rápida e genérica com que os conteúdos foram transmitidos e explicados em sala de aula.

Hartwig et al., (2019) identificam as metodologias ativas mais apropriadas para o ensino de computação e informática nos cursos de graduação. Além disso, afirmam que as mesmas estão

ganhando espaço nas salas de aula com foco na inovação. Portanto, a aprendizagem ativa envolve construir conhecimento em vez de recebê-lo passivamente (Barbosa e Moura, 2013). Diante deste contexto, este artigo apresenta os resultados de um relato de experiência do ensino e aprendizagem do curso presencial intitulado: “*CodeSmells? Aqui não! Limpando e Refatorando Códigos na Prática*”.

O curso foi ofertado pelo projeto Laboratório de Ensino e Pesquisa de Tecnologias alinhadas à Gestão do Conhecimento e Inovação em Processos de Software (LearningLab) da Universidade Federal do Ceará (UFC) do Campus de Russas. O objetivo do LearningLab é ensinar na prática os conceitos teóricos de computação abordados em sala de aula. As aulas do curso foram ministradas presencialmente em laboratórios da UFC, utilizando a ferramenta de comunicação WhatsApp para a comunicação. Os dados coletados das percepções dos estudantes em relação ao ensino do curso foram analisados de forma qualitativa e quantitativa.

Além desta seção introdutória, o artigo está organizado da seguinte forma: a seção 2 apresenta os trabalhos relacionados a este trabalho; a seção 3 detalha a metodologia utilizada, durante a execução do curso; a seção 4 detalha os resultados obtidos através das análises e discussões; e, por fim, a seção 5 apresenta a conclusão e considerações finais.

2. Trabalhos Relacionados

A fim de potencializar a efetividade do ensino de programação, é fundamental resgatar na comunidade do meio os seus conhecimentos, enriquecendo o aprendizado a partir de artifícios e exemplos de diferentes âmbitos e experiências. Portanto, para complementar o trabalho desenvolvido nesse artigo, foram identificados demais estudos com propostas para a aplicação do ensino de programação, voltadas para a refatoração de códigos, visando a qualidade do software e as boas práticas de codificação.

Piccinin et al. (2020) apresentam um estudo qualitativo dos relatos de profissionais de desenvolvimento de software quanto à sua rotina de aprendizagem. Os insumos extraídos das entrevistas realizadas constata a importância da gerência da rotina desses profissionais, mapeando que suas experiências se assemelham às de aprendizes autorregulados e que suas dificuldades de planejamento se destacam como pontos fracos em seu aprendizado.

É com foco no projeto de desenvolvimento na área de tecnologia da informação que a refatoração atua como um dos principais pontos fortes de qualidade e manutenção. Dito isso, Martin *et al.*, (2008) define refatoração como um processo que melhora a estrutura interna do código sem alterar o seu comportamento interno. O trabalho de Rabelo *et al.*, (2018), ressalta a importância de aplicar a refatoração na prática para que a etapa de desenvolvimento tenha um código bem estruturado e com boa qualidade, por meio da utilização de ferramentas ou na técnica de revisão de código, chamada de *review*.

O presente artigo trata de analisar a experiência do ensino da refatoração de códigos em um curso rápido de 5 dias para os estudantes de Engenharia de Software e Ciência da Computação da Universidade Federal do Ceará, Campus de Russas. O curso adotou uma abordagem prática e de revisão de critérios de qualidade para preparar os alunos para o mercado de trabalho.

3. Metodologia

A metodologia de estudo utilizada neste trabalho é composta por 4 etapas substanciais: (a) planejamento do curso - parte teórica e prática; (b) divulgação do curso; (c) condução do curso; e, (d) execução de atividades e avaliação do curso. A seguir, serão descritas cada uma das etapas da metodologia do curso.

• Planejamento

O curso foi planejado pela categorização de suas necessidades básicas, que incluem: (1) base de estudo teórica e definição da prática; (2) escolha do público alvo; e, (3) plano de ensino. Primeiramente, definir a base teórica do curso é essencial para consulta e construção do material, o qual extraiu conteúdo do livro “Código Limpo” de Robert Martin (Martin, 2008). A partir dele, foram extraídos os pontos-chave da refatoração de códigos para o Notion⁽¹⁾ (aplicação de gerenciamento de conhecimento) e para uma apresentação com slides. Portanto, com base na revisão de códigos e no alicerce teórico, foram criadas as atividades práticas.

Com foco em auxiliar aqueles que mais convivem com a dificuldade no aprendizado de programação, a orientação do desenvolvimento das atividades, materiais e divulgação do curso foi guiada a captar a atenção e satisfazer as necessidades dos estudantes iniciantes do campus e com conhecimento escasso, os quais se encontram principalmente entre os 2º e 6º semestre dos cursos de Engenharia de Software e Ciências da Computação. O curso foi presencial com 5 aulas (duração de 2 horas cada), e foi executado no período de agosto (dia 30) a setembro (dia 15) de 2022. O plano de ensino do curso foi desenvolvido pelo time do projeto LearningLab, composto pelo instrutor, coordenadora e monitores. A visão ampla construída a partir dessa equipe, juntamente com a revisão da orientadora do projeto (que também é professora da cadeira de Manutenção de Software, desde 2020, da UFC - Campus-Russas), resultou em um cronograma presencial focado no cotidiano dos alunos do campus. O plano de ensino é descrito a seguir:

Aula 1 - Presencial | Introdução (Teórica: 2 horas): Apresentação do LearningLab; Como será o curso e atividade final; Introdução ao curso; Nomes e Funções e suas refatorações.

Aula 2 - Presencial | Comentários e Formatação (Teórica e Prática: 2 horas): Continuação do conteúdo anterior; Comentários bons e ruins; Formatação vertical/horizontal; Práticas e discussões em grupo.

Aula 3 - Presencial | Objetos/Classes (Teórica e Prática: 2 horas): Continuação do conteúdo anterior; Abstração de Dados; Práticas e discussões em grupo.

Aula 4 - Presencial | Testes e apresentação da atividade final (Teórica e Prática: 2 horas): Testes; Sites importantes; Apresentação do modelo final; Práticas e discussões em grupo.

Aula 5 - Presencial | Apresentações finais e Conclusão (Prática: 2 horas): Apresentações das atividades realizadas em grupo; Avaliação das atividades realizadas; Práticas e discussões em grupo.

3.1.1. Materiais

Os materiais didáticos práticos e intuitivos desenvolvidos pelo instrutor tem como finalidade auxiliar os participantes a pesquisar e estudar fora da sala de aula. A seguir, serão especificados os tipos de materiais disponibilizados: (a) **Slides de Apresentação**⁽²⁾: os slides da apresentação teórica eram posteriormente anexados ao grupo do *WhatsApp*⁽³⁾ do curso, facilitando a pesquisa dos estudantes; (b) **Conteúdo e Roteiro de aula no Notion**⁽¹⁾: De forma análoga aos slides, o material no Notion ajudou no estudo dos alunos e também no acompanhamento do roteiro das

aulas; (c) **Códigos de exemplo no GitHub:** códigos refatorados e formatados para o modelo didático foram anexados ao GitHub⁽⁴⁾ para apoiar as atividades de refatoração.

3.2. Divulgação do Curso

Para que o curso fosse efetivamente divulgado aos estudantes, o projeto LearningLab utilizou das suas redes sociais do projeto⁽⁵⁾ e do mecanismo de comunicação via e-mail das coordenações acadêmicas dos cursos de Engenharia de Software e Ciência da Computação. Além disso, tal divulgação foi guiada por um plano base dos processos de realização dos cursos do projeto a fim de intensificar a atração do público.

Para atingir aqueles que necessitavam, o planejamento do curso definiu seu público-alvo com base no perfil comum dos estudantes do campus. Em síntese, a organização do curso concluiu que alunos do 2º ao 6º período enfrentam dificuldades em programação devido ao conhecimento introdutório e raso, motivando a sua busca por otimização. Portanto, estudantes do 2º período em diante dos cursos de Engenharia de Software e Ciência da Computação podiam participar das inscrições. Das 40 inscrições no curso, 19 foram de alunos de Engenharia de Software e 21 de Ciência da Computação da UFC, com 20 alunos participando efetivamente das aulas.

3.3. Condução do Curso

A didática do curso foi baseada em uma metodologia dinâmica para a sala de aula, em que a teoria e a prática se complementam e desenvolvem a interação entre os alunos e o instrutor. As aulas guiaram os alunos a partir do modelo de apresentação teórica, por meio dos materiais adaptados às aulas, seguida da aplicação prática proporcionada pela participação dos alunos na refatoração de códigos em conjunto com o instrutor. É importante ressaltar a promoção ao pensamento crítico durante as aulas, que desempenhou um papel central no aprendizado da refatoração dado o seu teor revisionista. Além disso, a comunicação e atenção com os alunos foram estendidas ao *WhatsApp* com a criação de um grupo do curso.

3.4. Execução de atividades e Avaliação do Curso

A fim de proporcionar a fixação dos conceitos teóricos trabalhados no ensino de refatoração, o curso propôs como atividade prática a coleta, análise e tratamento de códigos disponíveis no *GitHub*, visando a aplicabilidade dos pilares técnicos da refatoração de códigos ao final do curso. A atividade poderia ser feita em dupla e com qualquer código, porém incentivamos os alunos a corrigirem seus próprios projetos em dupla para uma análise crítica da qualidade do código. Dessa forma, mais especificamente, o trabalho prático foi definido como uma apresentação da refatoração realizada pelos alunos, que foi dividida em 4 seções: (a) apresentação do objetivo do programa codificado, (b) amostra das modificações realizadas e suas justificativas, (c) comparação entre o antes e depois da refatoração, e (d) retrato das dificuldades na realização da atividade de refatoração.

Os critérios de avaliação foram fundamentados na aplicação prática dos conceitos teóricos de otimização da programação na refatoração de código dos alunos. Os critérios foram: (1) Aplicação dos conhecimentos vistos em sala de aula, com uma pontuação de 0 a 4, levando em consideração as boas práticas, como nomes adequados, funções bem definidas, classes apropriadas, entre outros; (2) Apresentação da refatoração, com uma pontuação de 0 a 3, avaliando a demonstração do código antes e depois da refatoração; e (3) Apresentação da atividade, com uma pontuação de 0 a 3, analisando a clareza e abrangência da apresentação dos

trabalhos. A atribuição das pontuações foi feita pela professora coordenadora do projeto em conjunto com o instrutor do curso. As apresentações dos trabalhos foram realizadas com a presença de todos os estudantes, permitindo um diálogo entre o instrutor e os alunos, onde foram destacados tanto os pontos falhos quanto os sucessos das refatorações realizadas. A Figura 1 apresenta imagens do curso realizado de forma presencial.



Figura 1. Imagens das aulas do curso

4. Resultados e Discussões

A fim de coletar o *feedback* dos alunos, as perguntas foram recolhidas em formato qualitativo e quantitativo, buscando opiniões sinceras e análises numéricas. As etapas para análise e coleta dos dados foram: (i) **Realização do Curso:** possibilitou aos estudantes vivenciarem as experiências para responderem o formulário de feedback; (ii) **Coleta de Dados:** um formulário⁽⁶⁾ elaborado pelos integrantes do projeto e validado pela coordenadora foi utilizado para coletar dados dos alunos sobre fatores relacionados à aprendizagem. Os participantes são informados que a participação para responder respeita os Termos de Consentimento Livre e Esclarecido (TCLE) e todos os dados pessoais dos participantes são omitidos; (iii) **Análise dos Dados:** a análise qualitativa utilizou procedimentos de codificação para identificar as principais categorias de resultados. A análise quantitativa utilizou de um formulário baseado no método IMI (Inventário de Motivação Intrínseca) (McAuley *et al.*, 1991) para análise dos dados coletados. A seguir será descrito os resultados desta pesquisa.

- **Análise Qualitativa dos Dados**

A análise qualitativa dos dados coletados no curso foi realizada por meio de questionários no formato do Google Forms, que continham perguntas direcionadas para a coleta de *feedback* dos alunos. Esses *feedbacks* foram utilizados para identificar pontos positivos e negativos do curso, visando melhorar cursos futuros. Para analisar os dados coletados, foi adotada procedimentos da abordagem da Teoria Fundamentada nos Dados (Grounded Theory) (GT), que consiste em uma metodologia de pesquisa qualitativa que busca desenvolver teorias a partir dos próprios dados coletados. GT foi utilizada para compreender as percepções e experiências dos participantes e

utilizar esses *insights* para aprimorar a qualidade do ensino. Essa abordagem utiliza a codificação como uma etapa fundamental, na qual os dados são transformados em códigos e categorias.

Os códigos representam as ações ou elementos de interesse para o pesquisador, enquanto as categorias são agrupamentos de códigos que compartilham características semelhantes. Essa codificação e categorização dos dados foi realizada utilizando a ferramenta ATLAS.ti, que auxilia na organização e análise qualitativa dos dados. Essa abordagem permitiu uma análise aprofundada das respostas dos alunos, identificando padrões, tendências e temas emergentes que pudessem orientar melhorias nos cursos futuros. O setor de análises do LearningLab realiza a análise qualitativa para avaliar criticamente os serviços prestados pelo projeto, mapeando êxitos e falhas para aumentar a qualidade das atividades executadas. Os participantes foram nomeados de P1 a P20 para identificação e confidencialidade. A análise resultou em boas avaliações, refletindo o bom planejamento desde a ideia inicial ao processo de montagem do curso. A seguir estão os insumos extraídos das respostas dissertativas do formulário de *feedback* do curso.

4.1.1. O que mais gostou no curso?

A rede de códigos exposta na Figura 2 mostra o que os participantes mais gostaram no curso no geral. Nesta rede, destacam-se os seguintes códigos: “*Didática do instrutor de fácil entendimento*”, “*Metodologia de ensino com práticas e exemplos*” e “*Acolhimento que o projeto dá para os alunos*”, como relatam os estudantes P5, P16 e P10:

P5 - "A didática do professor é muito boa; Todos os métodos aprendidos parecem ser muito importante para criação de um bom código".

P16 - "A metodologia de ensino foi muito boa".

P10 - "Sim, gosto muito do acolhimento que o projeto dá para os alunos".

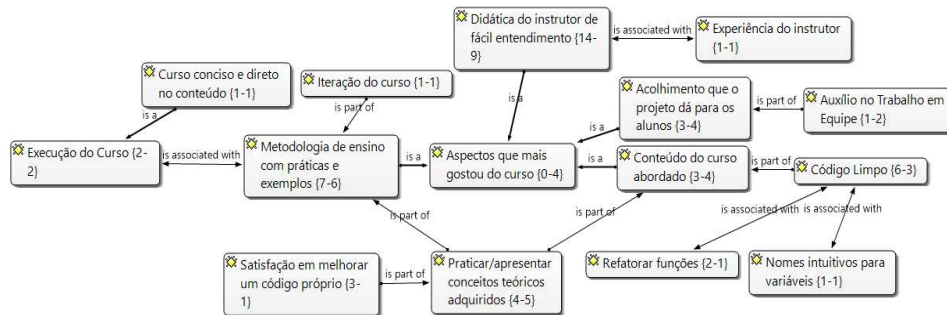


Figura 2. Rede de códigos “Aspectos que mais gostou no curso”

4.1.2. Aspectos que mais gostou - Metodologia de ensino

A Figura 3 trata mais a fundo os pontos característicos da metodologia de ensino levantados pelas respostas do formulário de *feedback*. Em suma, muitos alunos destacaram a didática do instrutor como facilitadora ao incentivar a participação da turma em um ambiente confortável. Essa percepção pode ser evidenciada pelos códigos: “*Didática do instrutor de fácil entendimento*”, “*Metodologia de ensino com práticas e exemplos*” e “*Explicação descontraída de ensino*”, como relatam os estudantes P4, P16 e P5:

P4 - "O aspecto teórico, a prática faz sim toda a diferença, mas a explicação simplificada ajudou muito".

P16 - "A metodologia de ensino foi muito boa".

P5 - "Gostei do modo explicativo e descontraído de ensino junto com a participação dos alunos".

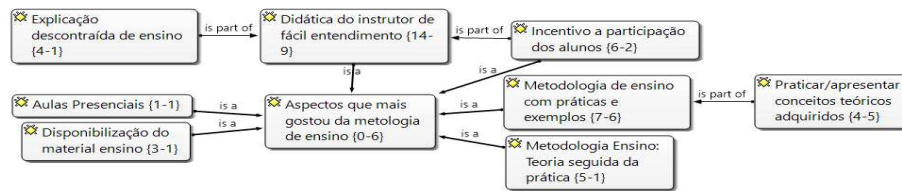


Figura 3. Rede de códigos “Aspectos positivos da metodologia de ensino”

4.1.3. Aspectos que menos gostou - Metodologia de ensino

A rede de Aspectos negativos da metodologia de ensino tem como objetivo sintetizar os pontos fracos do plano de ensino percebidos pelos alunos, os quais foram dados pelos códigos “*Ritmo Acelerado da Aula*” e “*Tempo curto de curso*”, como é dado pelos relatos de P2 e P3:

P2 - "*Poderia ter tido mais aulas para exercitar com o acompanhamento do monitor*".

P3 - "*Em relação aos exemplos, os códigos foram refatorados muito rápido*".

4.1.4. Dificuldades ao longo do curso

As dificuldades sentidas pelos alunos do curso puderam ser analisadas a partir do acúmulo de relatos pelo formulário de *feedback*. Em resumo, o código “*Dificuldade com POO*” foi o mais presente nas respostas, conforme as descrições dos participantes P11 e P16:

P11 - "*Entender como funciona a sintaxe de POO em java*".

P16 - "*O uso de classes em Python*".

4.1.5. Facilidades ao longo do curso

As facilidades percebidas pelos alunos puderam ser sintetizadas com uma análise descritiva das respostas à pergunta “*Quais foram as suas maiores facilidades ao longo do curso? Fale um pouco sobre:*”, presente no questionário de *feedback*. Em suma, os códigos “*Didática do instrutor de fácil entendimento*” e “*Aprender as boas práticas de refatoração*” foram o mais presentes nas respostas, conforme as descrições de P9 e P3:

P3 - "*Todas as aulas foram muito fáceis de acompanhar*".

P9 - "*Entender a premissa e o que deveria ser feito*".

4.1.6. Incentivo a Permanência

A rede construída pelos principais pontos motivadores à permanência dos alunos no curso tiveram como destaque os seguintes códigos: “*Didática do instrutor de fácil entendimento*”, “*Aprender as boas práticas de refatoração*” e “*Acolhimento que o projeto dá aos alunos*”, como é dado pelos relatos de P5, P3 e P10:

P5 - "*A didática do professor é muito boa*".

P3 - "*O meu interesse mesmo em aprender + sobre a linguagem, além também, de que o conteúdo me prendeu ao curso logo de início*".

P10 - "*gosto muito do acolhimento que o projeto dá para os alunos*".

4.1.7. O que menos gostou no curso?

A rede de códigos formulada a partir da pergunta “*O que você MENOS gostou no curso? Fale um pouco sobre:*” sintetiza o que os participantes menos gostaram no curso. Nesta rede destacam-se os seguintes códigos: “*Falta de Conhecimento prévio*”, “*Pouco tempo de curso*”, como relatam os estudantes P1 e P13:

P1 - "*Não entender algumas conversas por falta de conhecimento pois era um assunto pra quem já dominava alguma linguagem*".

• Resultados Fornecidos Através da Análise Quantitativa

A seguir será apresentada a análise dos gráficos que demonstram os dados coletados e analisados. O objetivo é entender o curso aplicado e melhorar os próximos.

4.2.1. Conhecimento em relação a Código Limpo/Refatoração Código

O gráfico abaixo (Figura 3) mostra o nível de conhecimento dos estudantes do curso de Code Smells por item, é notório que 85% dos participantes atribuíram nível 4 e 5 ao item “Lembrar o que é DEPOIS”, demonstrando assim que o curso foi efetivo no ensino do conteúdo proposto, vale enfatizar que itens relacionados a conhecimentos anteriores ao curso apresentam grandes divergências, provavelmente devido à diferença de semestres entre os participantes

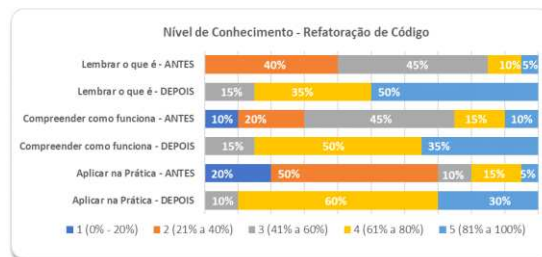


Figura 3 - Gráfico de Nível de Conhecimento

4.2.2. IMI (Inventário de Motivação Intrínseca)

Tendo como base os métodos quantitativos decidiu-se utilizar o método IMI para analisar de forma eficaz o feedback dos estudantes. O IMI tem como objetivo analisar a experiência subjetiva dos estudantes em relação à motivação intrínseca e autorregulação. Foi utilizada uma versão mais direcionada à motivação durante a realização de tarefas (Ryan et al., 1991), ou seja, os discentes foram questionados em relação às experiências com a atividade final do curso.

O questionário IMI tinha como objetivo recolher informações sobre (McAuley et al., 1991): (a) **interesse/prazer**: análise do nível de interesse que o aluno tem ao realizar a atividade final, (b) **escolha percebida**: refere-se a alguma obrigatoriedade que o aluno teria para realizar a atividade, ou se foi realmente uma escolha por si, (c) **competência percebida**: avalia a disposição e competência dos estudantes ao realizar a atividade final; e (d) **pressão/tensão**: tem como foco obter respostas se os alunos se sentiram pressionados, ansiosos ou tensos para realizar a entrega final que foi proposta. O questionário é composto por 22 questões, cada pergunta contém opções que são analisadas pelo estudante baseado na escala Likert, indo de “discordo totalmente” a “concordo totalmente”.

O gráfico de **Escolha** (Figura 4.a) tem como foco analisar de forma geral se os estudantes tiveram escolha ao realizar as entregas das atividades, foi perceptível que 70% dos estudantes concordaram com o item “*Estou satisfeito com o meu desempenho nas entregas das atividades do curso*” e 60% concordaram com o item “*Senti que era minha escolha fazer as entregas das atividades do curso*”. Fica evidente também que nos itens “*Eu realmente não tive escolha em fazer entregas das atividades do curso*” e “*Eu fiz as entregas das atividades do curso porque não tinha escolha*” a maioria dos estudantes discordou, deixando de forma explícita que a maioria dos alunos teve sua opção de escolha diante as atividades.

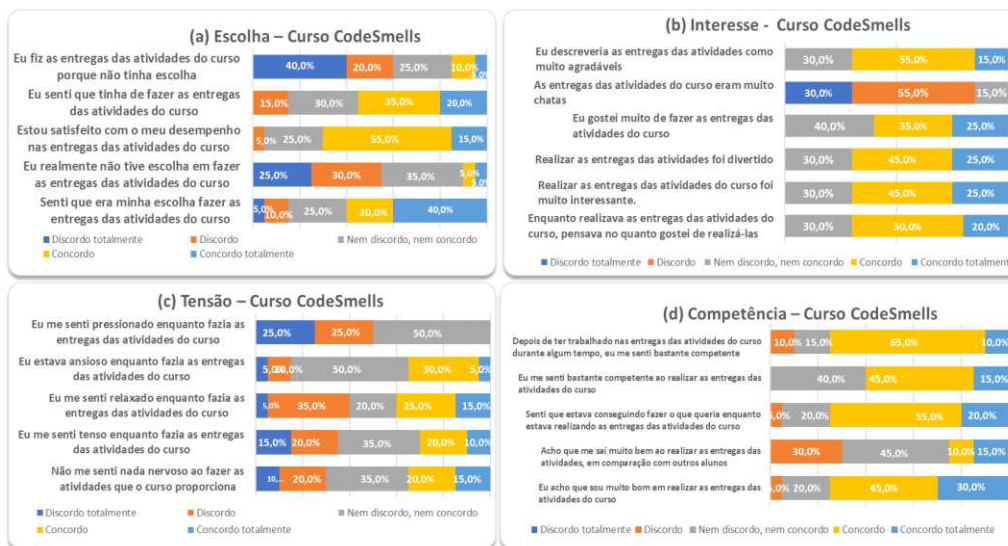


Figura 4 - Gráfico do Inventário de Motivação Intrínseca

Avaliando assim o gráfico **Interesse** da Figura 4.b, o qual tem como foco o agrado dos estudantes, percebeu-se que 60% dos estudantes concordaram com o item “*Eu gostei muito de fazer as entregas das atividades*”, tendo assim uma resposta positiva da maioria dos estudantes, onde o restante não concordou e nem discordou. Vale ressaltar também que 70% dos participantes concordaram com os itens “*Realizar as entregas das atividades foi interessante*” e “*Realizar as entregas das atividades foi divertido*”. Além disso, vale evidenciar que 85% dos alunos discordaram do item “*As entregas das atividades dos cursos eram muito chatas*”, reforçando o interesse profundo nessa entrega.

O gráfico de **Tensão** (Figura 4.c) visa avaliar a tensão/pressão que os estudantes podem ter sentido durante as atividades do curso. Analisando os dados, é visível que 30% dos estudantes concordaram com o item “*Eu me senti tenso enquanto fazia as entregas das atividades do curso*” e que 50% discordaram do item “*Eu me senti pressionado enquanto fazia as entregas da atividade do curso*”, e os outros 50% não concordaram nem discordaram, evidenciando que a maioria dos alunos não sentiu pressão/tensão ao realizar as atividades do curso.

A Figura 4.d apresenta o gráfico que aborda a **Competência** e desempenho dos estudantes ao longo do curso. É evidente que, com exceção do item “*Acho que me saí muito bem ao realizar as entregas das atividades, em comparação com os outros alunos*”, no qual apenas 25% concordaram, a maioria dos estudantes concordou com suas competências, variando entre 60% e 75% de concordância nos demais itens relacionados ao desempenho individual de cada estudante.

4.3 Ameaça à Validade dos Resultados

Segundo Wohlin *et al.*, 2012, todo estudo experimental apresenta riscos à sua validade no qual precisam ser identificados e tratados para não influenciar o resultado final. Durante o estudo, alguns riscos foram apresentados, como a análise dos dados, a baixa taxa de respostas, participantes e desistências. A interpretação incorreta dos resultados foi identificada como uma potencial ameaça à validade do estudo, no entanto, seu impacto foi minimizado por meio de revisões das conclusões. Para evitar essa consequência, uma equipe de alunos realizou uma

análise e codificação dos resultados de forma quantitativa, possibilitando a revisão dos dados. Após, duas especialistas em análise qualitativa validaram os dados analisados.

O baixo número de participantes é um fator que impacta na credibilidade dos resultados, uma vez que houve desistências dos alunos e ausência de respostas no formulário de *feedback*, o que resultou em uma redução da amostra da pesquisa. Isso limita a capacidade de generalização dos achados para a população em geral, comprometendo a validade da pesquisa. No entanto, para contornar essa limitação, foi realizada uma análise aprofundada dos dados coletados. Além do mais, o questionário para avaliação da aprendizagem, que permite a investigação sobre o nível do conhecimento pré e pós curso e perguntas de autoavaliação, foi identificado como fator de risco. Em contrapartida, Sitzmann *et al.*, 2010 e Thomas *et al.*, 2011, discorrem sobre os indícios de que a autoavaliação fornece informações relevantes e confiáveis, que são úteis para esse tipo de estudo. Essa avaliação crítica dos resultados permite fornecer lições valiosas, porém, destaca-se a importância da necessidade de pesquisas futuras com amostras mais representativas para ampliar e validar as evidências.

5. Conclusões e Trabalhos Futuros

Este artigo objetivou estudar dados sobre o desempenho do curso de refatoração de códigos, oferecido pelo projeto LearningLab, com base na análise quantitativa e qualitativa dos relatos dos alunos participantes. Como síntese, a análise quantitativa evidenciou a conformidade da maioria dos estudantes com o êxito do curso em quatro dimensões: Interesse, Competência, Tensão e Escolha. Contudo, há contraste de opiniões em questões específicas do aprendizado, retratando diferentes experiências na mesma sala de aula, especialmente em relação à tensão na entrega das atividades.

Em suma, a metodologia de ensino cativou o interesse dos alunos, o que refletiu em um aprendizado concreto dado às competências ao realizar as entregas das atividades. Ademais, na análise descritiva dos relatos do questionário de *feedback*, foi possível notar um entusiasmo dos alunos pela refatoração, tanto pelos aspectos técnicos das atividades em grupo quanto pela abordagem descontraída e simplificada da explicação teórica, baseada no diálogo. A metodologia aprovada pelo instrutor e orientadora do projeto criou um ambiente acolhedor, incentivador e eficiente para o curso.

As conclusões deste trabalho tornam viável a criação de cursos adaptados à realidade dos estudantes do Campus de Russas da UFC, público-alvo do LearningLab. Portanto, o acúmulo de recomendações e relatos permite a melhoria progressiva da qualidade das atividades e cursos propostos, como os de HTML e CSS, React Native e Flutter realizados após o curso descrito neste trabalho.

Referências

- Barbosa, E. F., Moura, D. G. (2014). Metodologias ativas de aprendizagem no ensino de engenharia. XIII International Conference on Engineering and Technology Education, v. 13, p. 16-19. ISSN 2317-4382.
- Corbin, J. and Strauss, A. (2007). Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. Sage Publications.

- Fontana, F. A., Ferme, V., Spinelli, S. (2012). Investigating the impact of code smells debt on quality code evaluation. In Proceedings of the MTD , MTD '12, p. 15–22, Piscataway, NJ, USA. IEEE Press.
- Fowler, M. (2018). Refactoring: improving the design of existing code. [S. l.]: Addison-Wesley Professional.
- Hartwig, A. K., Silveira, M., Fronza, L., Mattos, M., de Araújo Kohler, L. P. (2019). Metodologias ativas para o ensino da computação: uma revisão sistemática e um estudo prático. In Anais do Workshop de Informática na Escola, v. 25, n. 1, p. 1139-1143.
- Martin, Robert Cecil. Clean Code: A Handbook of Agile Software Craftsmanship. (2008).
- McAuley, E., Wraith, S., Duncan, T. (1991). Self-efficacy, perceptions of success, and intrinsic motivation for exercise. Journal of Applied Social Psychology, v. 21, n. 2, p. 139-155.
- Mendes, J., Costa, Y., Frazão, K., Santos, R., Santos, D., Rivero, L. (2019). Identificação das expectativas e dificuldades de alunos de graduação no ensino de engenharia de software. In Anais do XXVII Workshop sobre Educação em Computação, p. 334-347, SBC.
- Piccinin, G. F., Moro, M. M. (2020). Investigação da rotina de aprendizagem de desenvolvedores de software. In Anais do XXVIII Workshop Sobre Educação em Computação, p. 196-200, SBC.
- Rabelo, A., Maia, L. C., Parreiras, F. (2018). Performance Analysis of Computer Science Students in Programming Learning. In Anais do XXVI Workshop sobre Educação em Computação. Porto Alegre: SBC. doi:10.5753/wei.2018.3499.
- Ryan, R. M., Koestner, R., Deci, E. L. (1991). Ego-involved persistence: When free-choice behavior is not intrinsically motivated. Motivation and emotion, v. 15, n. 3, p. 185-205.
- Sitzmann, T., Ely, K., Brown, K. G., & Bauer, K. N. Self-assessment of knowledge: A cognitive learning or affective measure?. Academy of Management Learning & Education, v. 9, n. 2, p. 169-191, 2010.
- Souza, S. R., Choma Neto, J., Paschoal, L. N., Hernandez, E. (2021). Ensino Remoto Emergencial de Engenharia de Software com PBL: um relato de experiência In Anais do XXIX Workshop sobre Educação em Computação, p. 31-40, SBC.
- Thomas, G., Martin, D., Pleasants, K. Using self- and peer-assessment to enhance students' future-learning in higher education. Journal of University Teaching & Learning Practice, v. 8, n. 1, p. 1-17, 2011.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. Experimentation in software engineering. Springer Science & Business Media, 2012.

Links Citados no Artigo

- (1) <https://bit.ly/notionCodeSmells>;
- (2) <https://bit.ly/materialCodeSmells>;
- (3) https://www.whatsapp.com/?lang=pt_br;
- (4) <https://github.com/usрмаia/curso-codesmells-aqui-nao>;
- (5) <https://www.instagram.com/learninglabuf/>;
- (6) <https://bit.ly/formulariocodesmells>.