Construindo TCCs com Sprints, Rubricas e Feedbacks: Metodologias Ágeis e Avaliação Transparente

Rafaela Otemaier[1], Cristina Souza[2], Joselaine Valaski[2], Evandro Zatti[2]

[1]Graduate Program in Computer Science (PPGIa) – Pontifícia Universidade Católica do Paraná (PUCPR)

[2]Polytechnic School – Pontifícia Universidade Católica do Paraná (PUCPR)

{kelly.rafaela, cristina.souza, joselaine.valaski

evandro.azatti@pucpr.br}

Abstract. *This article describes an experience guiding undergraduate final projects in Information Systems, structured into two semester-long courses that adopt agile methodologies for software development. To align market practices with academic rigor, assessment rubrics and continuous feedback foster self-regulation and professional skills. We detail course organization, assessment strategies, and weekly monitoring. The results indicate that this integrated approach enhances student engagement, improves learning outcomes, and bridges the gap between academia and industry expectations, contributing to a more effective educational process.*

Resumo. *Este artigo descreve uma experiência de condução de Trabalhos de Conclusão de Curso (TCC) em Sistemas de Informação, estruturada em duas disciplinas semestrais que adotam metodologias ágeis para o desenvolvimento de software. Para alinhar prática de mercado e rigor acadêmico, utilizam-se rubricas de avaliação e feedback contínuo, favorecendo autorregulação e competências profissionais. Relatamos a estrutura e condução das disciplinas e os instrumentos avaliativos. Os resultados mostram que essa abordagem integrada aumenta o engajamento discente, aprimora a aprendizagem e reduz a distância entre formação acadêmica e exigências do mercado, promovendo um processo formativo mais eficaz.*

1. Introdução

A formação de profissionais na área de Computação, sobretudo em cursos de Sistemas de Informação e Engenharia de Software, exige um equilíbrio delicado entre o rigor acadêmico e as práticas de mercado. Por um lado, as instituições de ensino superior devem garantir a qualidade e a profundidade teórica necessárias para a compreensão dos princípios fundamentais da Computação. Por outro lado, esses cursos precisam acompanhar as rápidas transformações tecnológicas e atender às demandas do mercado, que exigem desenvolvedores e analistas altamente qualificados para lidarem com projetos complexos em ambientes reais [Paasivaara et al. 2018, Hooshangi et al. 2025].

Em meio a esses desafios, metodologias ágeis de desenvolvimento de software emergiram como um padrão de mercado nas últimas duas décadas, transformando a forma

como equipes lidam com o ciclo de vida de produtos de software. Práticas como Scrum, Kanban e XP (Extreme Programming) passaram a ser adotadas não apenas em grandes corporações, mas também em médias e pequenas empresas, evidenciando a flexibilidade e a adaptabilidade trazidas por esses métodos [Villavicencio et al. 2017]. Nesse contexto, a academia viu a necessidade de repensar as estratégias de ensino-aprendizagem, aproximando os estudantes de experiências mais autênticas que reflitam cenários reais de desenvolvimento [Gruslu et al. 2018, Radermacher et al. 2014].

Diante disso, o Trabalho de Conclusão de Curso (TCC) pode atuar como um importante ponto de convergência entre a teoria e a prática. Tradicionalmente, muitos TCCs na Computação acabam concentrando-se em dissertações ou monografias, enfatizando aspectos puramente teóricos ou experimentais [Paasivaara et al. 2018, Hooshangi et al. 2025]. Embora tais abordagens sejam valiosas, há um movimento crescente em favor de propostas que utilizem metodologias ágeis para a concepção e a implementação de projetos de software, permitindo que os estudantes vivenciem problemas reais de engenharia de requisitos, design, desenvolvimento, testes e implantação [Meier et al. 2016, Gustavsson et al. 2022].

Outro desafio latente em cursos de Computação diz respeito à avaliação. Se, por um lado, o uso de metodologias ágeis pode inspirar um desenvolvimento iterativo e colaborativo [Meier et al. 2016], por outro, é preciso garantir que a avaliação seja transparente e incentive a autorregulação dos discentes. Nesse ponto, as rubricas e o feedback contínuo mostram-se aliados poderosos. As rubricas detalham critérios claros de desempenho, enquanto o feedback contínuo assegura um acompanhamento próximo das atividades, facilitando ajustes e garantindo melhor aproveitamento acadêmico [Mikalsen and Dingsøyr 2023].

Neste artigo, é apresentada uma proposta de condução do TCC em duas disciplinas semestrais, ambas apoiadas em princípios ágeis, rubricas de avaliação e feedback contínuo. O objetivo é (a) promover uma vivência prática das metodologias ágeis, alinhada às necessidades do mercado; (b) empregar ferramentas avaliativas que tornem o processo mais transparente e formativo; e (c) fortalecer a conexão entre teoria e prática, contribuindo para uma formação integral do profissional de Sistemas de Informação. Para isso, abordamos como a adoção de sprints, provas de autoria individual e acompanhamento semanal podem melhorar o desempenho dos discentes, bem como discutir as vantagens de alinhar teoria e prática.

O artigo está estruturado da seguinte forma: na Seção 2, abordamos os principais trabalhos relacionados e discutimos a relevância das metodologias ágeis, rubricas e feedback contínuo na formação em Computação. Na Seção 3, descrevemos a metodologia proposta e sua implementação em duas disciplinas de TCC. Na Seção 4, relatamos a experiência de aplicação prática. Na Seção 5, discutimos as lições aprendidas ao longo do processo. Finalmente, na Seção 6, apresentamos as conclusões e perspectivas futuras.

2. Trabalhos Relacionados

Esta seção apresenta estudos que abordam a aplicação de metodologias ágeis no contexto acadêmico e a adoção de práticas avaliativas com caráter formativo, que podem promover a formação de profissionais aptos a desenvolver soluções tecnológicas eficazes de maneira transparente..

2.1. Metodologias Ágeis em Disciplinas de TCC

As metodologias ágeis, como o Scrum, vêm sendo amplamente discutidas em cursos de Computação por aproximarem a prática acadêmica das exigências de mercado [Villavicencio et al. 2017, Meier et al. 2016]. No âmbito de TCC, seu uso tem sido reportado como benéfico para estimular autonomia, trabalho em equipe e iterações curtas [Gustavsson et al. 2022]. O foco em entregas incrementais de software contribui para alinhar teoria e prática, pois os estudantes têm a oportunidade de experimentar problemas de engenharia de requisitos, arquitetura, testes e implantação de forma integrada [Paasivaara et al. 2018, Gruslu et al. 2018]. Ao mesmo tempo, há o desenvolvimento de soft skills como comunicação e negociação com stakeholders simulados ou reais [Keogh et al. 2007, Venson et al. 2016].

Mesmo assim, há estudos mostrando que a transposição de práticas ágeis para disciplinas universitárias precisa considerar: (i) carga de trabalho docente para orientar e oferecer feedback frequente; (ii) preparo discente para trabalhar em equipes autônomas; e (iii) avaliação justa que reflita o desempenho individual dentro de um trabalho coletivo [Gruslu et al. 2018, Radermacher et al. 2014]. Esses desafios motivam a adoção de rubricas e provas de autoria individual, de modo a garantir transparência e confiabilidade no processo avaliativo [Alberti et al. 2021, Mikalsen and Dingsøyr 2023].

2.2. Uso de Rubricas para Promover a Autorregulação

Rubricas são instrumentos de avaliação que descrevem os critérios e níveis de desempenho esperados em cada critério, orientando tanto professores quanto estudantes [Lima et al. 2024]. Em Computação, estudos evidenciam que rubricas, quando bem formuladas, podem impulsionar a autorregulação, pois os estudantes compreendem melhor o que se espera em termos de entrega e podem monitorar seu progresso [Alberti et al. 2021]. Para isso, é importante que os critérios da rubrica sejam objetivos e que haja exemplos claros de cada nível de desempenho.

A aplicabilidade das rubricas em TCC se dá nas diferentes etapas de avaliação: da proposta inicial até a defesa final. Ao evidenciar o que é considerado um desempenho excelente, satisfatório ou insuficiente, a rubrica reduz subjetividades, incentiva ajustes antes da entrega final e facilita o feedback pontual por parte do docente [Lima et al. 2024]. Ademais, parte da literatura ressalta que as rubricas funcionam melhor quando associadas a um processo de orientação que inclua discussões e revisões, reforçando a ideia de que a avaliação deixa de ser meramente punitiva e torna-se formativa [Falcão et al. 2022].

2.3. Feedback Contínuo

O feedback é apontado como fator-chave no sucesso de práticas educativas que busquem aprendizagem ativa e adaptativa [Mikalsen and Dingsøyr 2023]. Em contextos de TCC ou disciplinas de projeto, costuma-se apontar que o tempo e o volume de trabalhos prejudicam a capacidade do professor em fornecer comentários individualizados e detalhados [Falcão et al. 2022]. Estratégias para contornar esse problema incluem: (a) encontros frequentes (semanais ou quinzenais) com cada equipe, (b) uso de ferramentas on-line para registro de comentários e (c) divisão de tarefas avaliativas entre vários docentes ou monitores [Mikalsen and Dingsøyr 2023].

Uma característica valiosa é que o feedback seja imediato ou oferecido em um intervalo curto, para que os estudantes incorporem as sugestões enquanto o projeto ainda

está em curso. No caso do TCC, isso se reforça pela natureza iterativa: as equipes podem corrigir rumos e aprimorar o produto a cada sprint, reduzindo retrabalhos de última hora. Além disso, adotar métodos como provas de autoria ajuda a identificar e valorizar a contribuição individual, evitando a participação passiva de alguns membros do grupo [Dupuis et al. 2010].

3. Metodologia

A proposta metodológica apresentada aplica-se ao TCC em um curso de Sistemas de Informação, no qual se dividem as atividades em duas disciplinas semestrais. Os princípios de Scrum orientam o planejamento e a execução das tarefas, enquanto rubricas e feedback contínuo definem a forma de avaliação.

3.1. Estrutura Geral das Disciplinas de TCC

O Projeto Pedagógico do curso de Sistemas de Informação em análise divide o TCC em duas disciplinas certificadoras, chamadas Desenvolvimento Ágil de Produto I (7º período) e Desenvolvimento Ágil de Produto II (8º período). Cada disciplina tem 90 horas-relógio, totalizando 180 horas-relógio (240 horas-aula). No início do 7º período, os estudantes formam equipes (preferencialmente de 3 ou 4 integrantes) e escolhem um projeto real ou factível para desenvolvimento de um sistema de informação. É incentivado que haja um cliente ou empresa como stakeholder externo, mas a orientação também aceita propostas que simulam casos reais.

A metodologia que orienta as disciplinas segue o Scrum como arcabouço de planejamento e execução. Assim, os projetos se desenvolvem em quatro sprints, duas por semestre, cada qual envolvendo as seguintes etapas: definição e priorização de requisitos (ou estórias de usuário), execução em equipe, acompanhamento semanal com professores e uma defesa a cada sprint.

3.2. Avaliação e Uso de Rubricas

A avaliação do projeto nas duas disciplinas de TCC foi concebida para balancear o aspecto formativo, sustentado por feedback contínuo, e o somativo, que reúne defesas parciais e provas de autoria individual. A fim de assegurar transparência nos critérios de correção, bem como estimular a autorregulação dos estudantes, adotaram-se rubricas que especificam diferentes níveis de desempenho em cada item analisado.

3.2.1. Avaliação Formativa

A avaliação formativa ocorre por meio de encontros semanais, nos quais professores e estudantes discutem o desenvolvimento do projeto, dificuldades técnicas, gestão de tarefas e possíveis ajustes de escopo. Essas sessões são registradas em um diário de bordo no Ambiente Virtual de Aprendizagem (AVA), permitindo que cada equipe receba orientações pontuais para aprimorar o trabalho de uma sprint para a seguinte. Por não envolver atribuição de notas, esse acompanhamento estimula a participação ativa dos estudantes, que contam com oportunidades constantes de correção e aperfeiçoamento antes das avaliações finais.

3.2.2. Avaliação Somativa e Provas de Autoria

Ao final de cada sprint, a equipe realiza uma defesa do projeto perante banca formada pelos professores orientadores. Nessa ocasião, demonstra as funcionalidades previstas, evidenciando se as estórias de usuário foram concluídas e se atendem aos critérios de aceite. Logo depois, cada integrante participa de uma *Prova de Autoria*, atividade individual que consiste em implementar ou ajustar uma funcionalidade do software (incluindo a integração front-end, back-end e banco de dados) em um período limitado de tempo. Se o estudante concluir a tarefa dentro do prazo, mantém a pontuação total obtida pela equipe; caso contrário, são aplicados descontos progressivos.

Essa estratégia possibilita mensurar a contribuição individual em um contexto de desenvolvimento coletivo, garantindo a participação efetiva de todos. Além disso, evita desequilíbrios na colaboração dentro do grupo, resultando em um processo avaliativo mais justo. [Keogh et al. 2007, Venson et al. 2016].

3.2.3. Rubricas

Todos os processos avaliativos de natureza somativa fazem uso de rubricas padronizadas, disponibilizadas com antecedência no AVA. Essas rubricas cobrem diferentes dimensões do TCC, oferecendo descrições claras de desempenho para cada critério analisado. Os principais itens costumam incluir:

- **Especificação do Tema e Escopo**: nível de clareza do problema, relevância e grau de detalhamento inicial.
- **Backlog do Produto**: completude, coerência e priorização das estórias de usuário, conforme boas práticas de desenvolvimento ágil.
- **Gerenciamento de Projeto**: uso de ferramenta on-line (por exemplo, *Kanban*), registro contínuo de tarefas e evidências de evolução consistente.
- **Versionamento**: organização de branches, frequência de *commits* e utilização apropriada de repositórios online.
- **Demonstração das Funcionalidades**: adequação aos critérios de aceite, qualidade do código e correta implementação das entregas acordadas.
- **Recursos Técnicos e Tecnológicos**: adoção de práticas de segurança, computação em nuvem, DevOps, recursos de IA ou outras tecnologias validadas pelos professores.
- **Prova de Autoria**: verificação individual da capacidade de realizar melhorias e integrações em tempo limitado.
- **Apresentação para a Comunidade Acadêmica**: exposição final (em formato *pitch*) para o público interno e externo, envolvendo demonstração do produto e participação de todos os membros da equipe.

Cada critério é associado a níveis de desempenho (por exemplo, *insuficiente, básico, adequado, excelente*), descrevendo detalhadamente os requisitos a serem atendidos para cada grau de proficiência. Ao fim de cada sprint, a soma das pontuações gera a nota parcial, e os comentários qualitativos são registrados no AVA para consulta futura.

A Figura 1 exibe um exemplo de rubrica utilizada, ressaltando como se classificam aspectos como implementação de melhorias, desenvolvimento de interfaces, modelagem

de dados e atendimento aos critérios de aceite. Cada categoria é classificada em quatro níveis (*Excelente*, *Suficiente*, *Insuficiente* ou *Não atende*), tornando a análise objetiva e consistente.

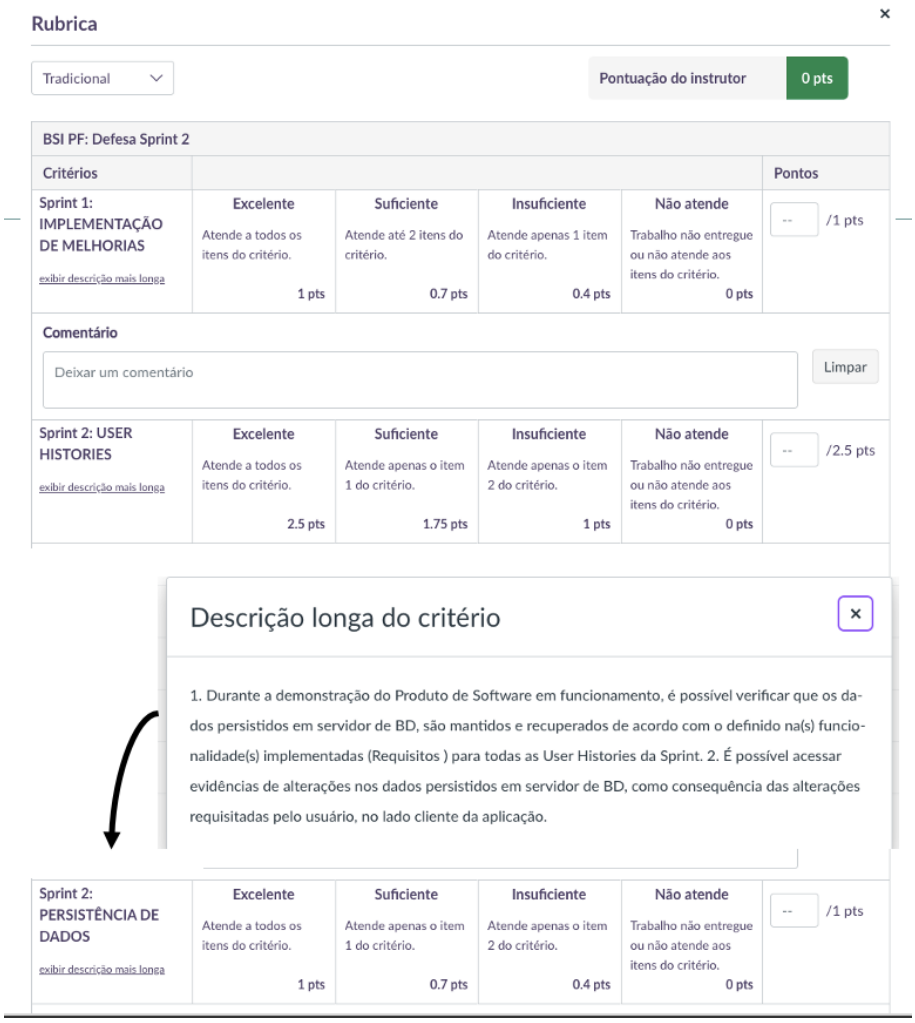

Figura 1. Exemplo de rubrica utilizada na avaliação das sprints do TCC.

Para ilustrar, no critério de **Persistência de Dados**, espera-se que o software demonstre, durante a apresentação, gravação e recuperação corretas de informações no banco de dados, em consonância com os requisitos definidos nas *User Stories* da sprint (Descrição longa do critério, apresentado na Figura 1). Além disso, deve haver evidências de que as alterações realizadas no lado cliente sejam refletidas no servidor. Atender a essas condições permite ao grupo alcançar a pontuação máxima nesse critério.

O uso de rubricas padronizadas torna o processo avaliativo mais justo e transparente, além de guiar o desenvolvimento dos estudantes ao longo do projeto. Na prática, cada equipe monitora seu progresso a partir desses critérios, direcionando esforços para melhorias pontuais ou ajustes estratégicos antes da próxima defesa. Essa abordagem fomenta a autonomia dos discentes e estimula a adoção de boas práticas de mercado, alinhadas às demandas acadêmicas e profissionais.

3.3. Condução das Disciplinas

Cada disciplina possui de 15 a 17 semanas de aula. Nos primeiros encontros de Desenvolvimento Ágil de Produto I, define-se a composição das equipes e a proposta inicial (tema, objetivos, escopo). Em seguida, planeja-se a primeira sprint (S1), com duração de cerca de metade do semestre, e a segunda sprint (S2), completando o semestre.

Em Desenvolvimento Ágil de Produto II, duas novas sprints acontecem (S3 e S4), visando entregar uma versão final funcional do software. Para ambas as disciplinas, ao término de cada sprint, há a defesa do projeto + prova de autoria, e a nota resultante contribui para a média final. Essa média considera também (em algumas turmas) itens como apresentação pública do projeto e produção de documentação final.

4. Relato da Experiência

Este relato de experiência refere-se à aplicação da metodologia descrita na Seção 3 ao longo dos últimos três anos. A Tabela 1 apresenta a quantidade de estudantes matriculados e de projetos desenvolvidos em cada ano de aplicação.

Tabela 1. Estudantes e Projetos

Ano	Qtde Estudantes	Qtde Projetos
2022	26	13
2023	35	15
2024	22	10

As experiências são relatadas de acordo com as Etapas do Projeto (Concepção e Sprints) e as avaliações (Formativas e Somativas) envolvidas.

4.1. Concepção

Esta etapa inicial do projeto tem se mostrado fundamental para o sucesso das etapas subsequentes (Sprints). A clareza e a adequação do escopo da proposta têm sido determinantes para o desenvolvimento eficiente e positivo do projeto. Observa-se que as equipes que iniciaram o projeto com uma composição já estabelecida e uma proposta bem estruturada conseguiram produzir de maneira consistente desde o primeiro dia de aula.

Um dos principais elementos dessa etapa é a utilização de um template de especificação que, ao adotar artefatos mais simples e diretos, tem auxiliado significativamente os estudantes na construção de uma visão abrangente do projeto. Esse template facilita a visualização do valor que o produto pode entregar ao negócio, considerando as necessidades do público-alvo, e define de forma clara as funcionalidades essenciais a serem implementadas para alcançar esse valor.

Quando comparada com a abordagem tradicional, aplicada em anos anteriores (até 2021), a adoção de artefatos mais simples resultou em uma redução significativa do tempo gasto pelas equipes no desenvolvimento de documentos complexos. Anteriormente, essas equipes precisavam frequentemente revisar e modificar artefatos complexos, o que acarretava retrabalho. Com a abordagem simplificada, as equipes têm se mostrado mais ágeis, conseguindo realizar alterações rápidas nos artefatos sem comprometer o progresso ou a qualidade do trabalho já desenvolvido.

Essa mudança metodológica tem se demonstrado ser eficaz, pois permite que as equipes se concentrem mais na execução e implementação das funcionalidades, ao invés de se perderem em etapas complexas de documentação. Dessa forma, a clareza e a objetividade no início do projeto favorecem uma evolução contínua e bem-sucedida nas etapas seguintes.

4.2. Sprints

A organização das etapas subsequentes à Concepção do Projeto em Sprints, com desenvolvimento iterativo e incremental de um produto funcional desde o início, tem se mostrado altamente motivadora. Esse formato tem permitido que as equipes acompanhem concretamente o progresso de seu trabalho a cada Sprint. Além disso, a flexibilidade para realizar ajustes, em conformidade com práticas ágeis do mercado, contribui para o engajamento dos estudantes. A combinação do ciclo iterativo com a visibilidade proporcionada por ferramentas de gerenciamento de tarefas fortaleceu essa motivação, enquanto a visualização do software em funcionamento desde as etapas iniciais se revelou um estímulo natural.

Adicionalmente, a coordenação conjunta das atividades de especificação das estórias de usuário e sua codificação desde o início do projeto tem exigido que as equipes tomem decisões sobre arquitetura e tecnologia de forma antecipada. Essa abordagem tem facilitado a implementação ágil de mudanças, evitando retrabalho significativo, uma vez que o projeto ainda se encontra em sua fase inicial. Em alguns casos, as equipes optaram por alterar a tecnologia utilizada na primeira sprint, e por realizarem essa modificação logo no início do projeto, os custos de retrabalho foram minimizados.

4.3. Avaliações

A avaliação é considerada um elemento central no desenvolvimento dos projetos de TCC, uma vez que o progresso dos estudantes pode ser monitorado, ajustes no processo podem ser orientados e a qualidade das entregas pode ser assegurada. A seguir, são descritas as abordagens adotadas para que a avaliação fosse conduzida ao longo das disciplinas.

4.3.1. Formativas: acompanhamento semanal

A avaliação formativa é realizada por meio de encontros semanais entre professores e estudantes, nos quais o progresso do projeto, os desafios técnicos, as estratégias de gestão de tarefas e os possíveis ajustes de escopo são discutidos. Esses encontros são registrados em um diário de bordo no Ambiente Virtual de Aprendizagem (AVA), possibilitando que a evolução do trabalho seja continuamente acompanhada. Diferentemente da avaliação somativa, essa etapa não envolve atribuição de notas, sendo priorizado o aprendizado e a melhoria incremental. Busca-se, com isso, que feedbacks detalhados sejam fornecidos para que os estudantes possam ser orientados quanto aos aspectos a serem aprimorados antes das avaliações finais.

Com essa estratégia, o aprendizado é fortalecido ao permitir que as abordagens dos estudantes sejam ajustadas com base nos retornos recebidos, preparando-os melhor para os momentos de avaliação formal. Além disso, a sistematização do acompanhamento semanal contribui para que a autonomia e a colaboração sejam favorecidas, incentivando uma postura ativa na resolução de problemas e no aprimoramento do projeto. Dessa

forma, entende-se que a avaliação formativa não apenas complementa a somativa, mas também exerce um papel essencial na construção de um ambiente de aprendizagem mais dinâmico e eficiente.

Nas primeiras aplicações da metodologia, foi relatado pelos estudantes que, durante os acompanhamentos, feedbacks e orientações divergentes eram ocasionalmente fornecidos pelos professores orientadores. Para que essa falta de uniformidade fosse evitada, foi elaborado um checklist com os itens a serem observados por todos os professores a cada semana. Essa estratégia teve como finalidade garantir maior consistência nas orientações fornecidas e minimizar os desalinhamentos entre os orientadores.

4.3.2. Somativas: rubricas e prova de autoria

A antecipação da disponibilização das especificações e rubricas das atividades somativas previstas em cada etapa foi responsável por garantir transparência no processo avaliativo e orientar as entregas de acordo com as expectativas estabelecidas pela disciplina. As rubricas, além de facilitarem o processo de correção, foram utilizadas para assegurar isonomia e tornar claros para as equipes os critérios pelos quais seriam avaliadas, estimulando, assim, a produção de entregas com maior qualidade.

Outro aspecto relevante identificado na metodologia adotada foi a utilização de bancas ao final de cada Sprint, incluindo a aplicação da avaliação de autoria individual. Essa prática tem sido considerada eficaz, pois assegura que todos os membros da equipe participem ativamente do desenvolvimento do projeto. A Prova de Autoria individual tem contribuído para que uma avaliação mais justa e equitativa do desempenho de cada estudante seja realizada, ao mesmo tempo em que a participação passiva de membros que não colaboram efetivamente é desencorajada.

Com o intuito de incentivar a implementação das melhorias sugeridas nos feedbacks fornecidos pelos professores nas atividades anteriores, um critério específico foi incluído em todas as rubricas das avaliações somativas. Essa estratégia buscou, além de facilitar a recuperação da aprendizagem, recompensar as equipes que aplicaram os feedbacks de forma eficaz, promovendo ajustes ágeis e resultando na entrega de artefatos aprimorados e mais alinhados com as expectativas estabelecidas.

5. Lições Aprendidas

A adoção de metodologias ágeis, associada ao uso de rubricas e de feedback contínuo, permitiu que importantes aspectos do processo de formação em Computação fossem revelados:

Integração Teoria–Prática. A aplicação da metodologia ágil ao TCC possibilitou que lacunas entre teoria e prática fossem reduzidas, uma vez que problemas reais de engenharia de software e gerenciamento de projetos foram enfrentados pelos discentes.

Importância do Feedback Contínuo. O acompanhamento semanal e o registro das atividades no AVA foram utilizados como ferramentas centrais na orientação dos grupos, permitindo que ajustes estratégicos fossem realizados e a evolução contínua do produto fosse promovida. Além disso, uma postura mais receptiva a críticas foi desenvolvida pelos estudantes, contribuindo para o aprimoramento de habilidades de comunicação,

resiliência e adaptabilidade — competências amplamente valorizadas no mercado de trabalho atual.

Valorização do Trabalho Individual. Embora o trabalho em grupo seja amplamente adotado em cursos de Computação, foi constatado que a combinação entre metodologias ágeis e Provas de Autoria se mostrou eficaz para o equilíbrio entre tarefas coletivas e responsabilidades individuais. Dessa forma, a ocorrência de "caronistas" foi reduzida e uma maior equidade na atribuição de notas foi assegurada.

Rubricas como Norte Avaliativo. A clareza sobre o que era esperado foi proporcionada pela disponibilização das rubricas desde o início, o que incentivou que melhorias iterativas fossem buscadas. Quando utilizadas em conjunto com feedbacks, as rubricas contribuíram para que a qualidade final das entregas fosse significativamente elevada.

Atenção ao Tamanho das Equipes. Foi observado que equipes muito grandes (com mais de quatro estudantes) podem exigir mecanismos de coordenação mais complexos, enquanto trabalhos individuais podem não favorecer plenamente a dinâmica colaborativa inerente às metodologias ágeis. Por esse motivo, recomenda-se que o número de integrantes seja ajustado e que o escopo do projeto seja delimitado, para que os benefícios do modelo possam ser plenamente concretizados.

Ajustes Futuros. Está sendo planejado, pela equipe docente, um mapeamento mais formal das competências desenvolvidas, relacionando-se os resultados de aprendizagem a indicadores específicos contidos nas rubricas. Além disso, pretende-se que parcerias externas sejam buscadas, com o intuito de ampliar o contato dos estudantes com problemas reais do mercado e, assim, reforçar sua formação prática e cidadã.

6. Conclusão

A adoção de metodologias ágeis no contexto do Trabalho de Conclusão de Curso (TCC) em Sistemas de Informação, aliada ao uso estruturado de rubricas avaliativas e feedback contínuo, demonstrou ser uma abordagem eficaz para reduzir a lacuna entre a formação acadêmica e as demandas do mercado. A experiência relatada neste estudo evidencia que a organização do TCC em ciclos iterativos e incrementais favorece a aprendizagem ativa, estimulando o desenvolvimento de competências técnicas e comportamentais essenciais para o exercício profissional.

Os resultados indicam que a transparência no processo avaliativo, proporcionada pelas rubricas, bem como a implementação de mecanismos de acompanhamento individualizado, como a prova de autoria, contribuem de forma significativa para a equidade na avaliação e para a autorregulação dos estudantes. Além disso, a estruturação do TCC em sprints, inspirada em práticas ágeis, promove um engajamento contínuo, permitindo ajustes e refinamentos ao longo do processo, o que se traduz em projetos mais alinhados a desafios reais da indústria de software.

Apesar dos benefícios observados, desafios ainda persistem, como a necessidade de um acompanhamento docente mais intensivo e a adaptação das metodologias para diferentes perfis de estudantes. Assim, futuras investigações podem explorar formas de otimizar a carga de trabalho dos orientadores, bem como avaliar o impacto da abordagem em diferentes contextos institucionais e culturais.

Como perspectiva futura, recomenda-se um mapeamento formal das competências

desenvolvidas ao longo do TCC, relacionando-as a indicadores específicos de aprendizagem e desempenho. Além disso, iniciativas que ampliem o contato dos estudantes com stakeholders externos podem fortalecer ainda mais a conexão entre academia e indústria, proporcionando experiências mais autênticas e significativas.

Com isso, conclui-se que a aplicação de metodologias ágeis, quando aliada a um modelo avaliativo estruturado e transparente, não apenas melhora os resultados acadêmicos, mas também prepara os estudantes para os desafios da prática profissional, consolidando uma formação mais alinhada às exigências contemporâneas do mercado de tecnologia.

Referências

Alberti, E., Quandt, V., Mendes, C., Bordignon, L., Przysiada, F., and Tavares, L. (2021). Apresentação de uma metodologia de avaliação baseada em rubricas para avaliação de trabalhos de conclusão de curso no curso de engenharia da computação. In *Anais do Simpósio Brasileiro de Educação em Computação*, pages 272–279. SBC.

Dupuis, R., Champagne, R., April, A., and Séguin, N. (2010). Experiments with adding to the experience that can be acquired from software courses. In *2010 Seventh International Conference on the Quality of Information and Communications Technology*, pages 1–6. IEEE.

Falcão, T., Arêdes, V., Wagner, S., Uchoa, J., Luisi, V., and Mello, R. (2022). What did i get wrong? supporting the feedback process in computer science education. In *Anais do XXX Workshop sobre Educação em Computação*, pages 239–250, Niterói, Brasil. SBC.

Gustavsson, T., Berntzen, M., and Stray, V. (2022). Changes to team autonomy in large-scale software development: a multiple case study of scaled agile framework (safe) implementations. *Int. J. Inf. Syst. Proj. Manag.*, 10(1):29–46.

Gruslu, V., Giray, G., Tüzün, E., Catal, C., and Felderer, M. (2018). Closing the gap between software engineering education and industrial needs. *arXiv preprint*.

Hooshangi, S., Shakil, A., Dasgupta, S., Davis, K. C. C., Farghally, M., Fitzpatrick, K., Gutica, M., Hardt, R., Riddle, S., and Seyam, M. (2025). Instructors' perspectives on capstone courses in computing fields: A mixed-methods study. In *2024 Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE 2024)*, pages 68–94, New York, NY, USA. Association for Computing Machinery.

Keogh, K., Sterling, L., and Venables, A. (2007). A scalable and portable structure for conducting successful year-long undergraduate software team projects. *J. Inf. Technol. Educ.: Res.*, 6(1):515–540.

Lima, M., Ferreira, D., and Dias, E. (2024). Uso de rubricas em disciplinas de programação introdutória: Uma revisão sistemática da literatura. In *Anais do XXXV Simpósio Brasileiro de Informática na Educação*, pages 1–14, Porto Alegre, RS, Brasil. SBC.

Meier, A., Kropp, M., and Perellano, G. (2016). Experience report of teaching agile collaboration and values: agile software development in large student teams. In *2016*

IEEE 29th International Conference on Software Engineering Education and Training (CSEET), pages 76–80. IEEE.

Mikalsen, M. and Dingsøyr, T. (2023). Feedback as a process in a large semi-capstone software engineering course. In *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering (EASE '23)*, pages 475–479, New York, NY, USA. Association for Computing Machinery.

Paasivaara, M., Vodă, D., Heikkilä, V. T., Vanhanen, J., and Lassenius, C. (2018). How does participating in a capstone project with industrial customers affect student attitudes? In *Proc. 40th Int. Conf. Softw. Eng. Softw. Eng. Educ. Training*, pages 49–57.

Radermacher, A., Walia, G., and Knudson, D. (2014). Investigating the skill gap between graduating students and industry expectations. In *Companion Proceedings of the 36th International Conference on Software Engineering*, pages 291–300.

Venson, E., Figueiredo, R., Silva, W., and Ribeiro, L. (2016). Academy-industry collaboration and the effects of the involvement of undergraduate students in real world activities. In *2016 IEEE Frontiers in Education Conference (FIE)*, pages 1–8. IEEE.

Villavicencio, M., Narvaez, E., Izquierdo, E., and Pincay, J. (2017). Learning scrum by doing real-life projects. In *2017 IEEE Global Engineering Education Conference (EDUCON)*, pages 1450–1456. IEEE.