

Perspectivas e Desafios na condução de uma Fábrica de Software: Um Relato de Experiência

Francisco Carlos M. Souza¹, Alinne C. Correa Souza¹, Marlon Marcon¹,
André Ortoncelli¹, Rafael Alves P. de Oliveira¹

¹Coordenação de Engenharia de Software
Universidade Tecnológica Federal do Paraná (UTFPR) – Dois Vizinhos, PR – Brasil

{franciscosouza, alinnesouza, marlonmarcon}@utfpr.edu.br

{ortoncelli, raoliveira}@utfpr.edu.br

Abstract. This paper presents a report on the experience of implementing and conducting an academic Software Factory within an undergraduate Software Engineering program. Based on the Project-Based Learning methodology, the discipline engages students in developing socially impactful MVPs organized in agile teams guided by the Scrum framework. The work discusses the main methodological steps, highlights recurring challenges, such as team composition, scope definition, and communication, and outlines the solutions adopted. Reflections from two years of experience emphasize the importance of integrating theory and practice and offer insights for replicating similar initiatives in other educational contexts.

Resumo. Este artigo apresenta um relato sobre a experiência de implementação e condução de uma Fábrica de Software acadêmica dentro de um programa de graduação em Engenharia de Software. Baseada na metodologia de Aprendizagem Baseada em Projetos, a disciplina envolve os alunos no desenvolvimento de MVPs de impacto social organizados em equipes ágeis guiadas pelo framework Scrum. O trabalho discute as principais etapas metodológicas, destaca desafios recorrentes, como composição da equipe, definição de escopo e comunicação, e descreve as soluções adotadas. Reflexões de dois anos de experiência enfatizam a importância da integração entre teoria e prática e oferecem insights para replicar iniciativas semelhantes em outros contextos educacionais.

1. Introdução

A graduação em Engenharia de Software tem como objetivo formar profissionais capazes de atuar em todo o ciclo de vida de desenvolvimento de sistemas. O curso contempla uma diversidade de disciplinas que abordam aspectos teóricos e práticos, tais como análise de requisitos, arquitetura de software, validação e verificação, entre outras. No entanto, apesar da base conceitual oferecida, muitos alunos têm dificuldades em conectar esses conhecimentos de forma integrada. Tradicionalmente, o ensino do desenvolvimento de software é fragmentado, com cada disciplina focando em um aspecto específico do processo. Essa abordagem, embora eficiente para aprofundamento em temas pontuais, tende a dificultar a compreensão do fluxo contínuo e interdependente das atividades reais em projetos de software. O aluno muitas vezes domina bem os conceitos individualmente,

mas sente dificuldade em aplicá-los em conjunto, como seria exigido em um ambiente profissional [Ouhbi and Pombo 2020].

Nesse contexto, torna-se essencial a oferta de experiências que proporcionem uma visão holística do processo de desenvolvimento de software. A disciplina de Fábrica de Software surge como uma resposta a essa lacuna, promovendo o desenvolvimento de MVPs (Minimum Viable Products) por times, simulando um ambiente de trabalho colaborativo e realista. Essa proposta permite que os estudantes passem por todas as fases de um projeto, desde a concepção da ideia até a entrega funcional de um sistema. Um dos principais problemas do ensino tradicional é a dissociação entre teoria e prática. Muitos alunos concluem o curso com domínio de técnicas e ferramentas, mas sem compreender como articular esse conhecimento na resolução de problemas reais. A ausência de atividades integradoras contribui para a sensação de insegurança diante de desafios complexos ou de tomada de decisões em contextos reais.

Nesse sentido, metodologias ativas como a Aprendizagem Baseada em Projetos (*Project-Based Learning - PBL*) têm sido bastante utilizadas, por proporcionarem um contexto prático, colaborativo e significativo para a aprendizagem. O PBL permite que os estudantes desenvolvam competências técnicas e socioemocionais por meio da resolução de problemas, favorecendo a integração de conteúdos e a compreensão da aplicação dos conceitos teóricos em projetos reais [Pucher and Lehner 2011].

A literatura sobre ensino em Engenharia de Software aponta para a importância de metodologias baseadas em projetos, aprendizado ativo e ambientes simulados como estratégias eficazes para o desenvolvimento de competências técnicas e comportamentais. Estudos destacam que a integração de disciplinas em experiências práticas favorece a retenção do conhecimento e a formação de profissionais mais preparados para o mercado [Fioravanti et al. 2018, Souza et al. 2019, Pérez and Rubio 2020]. Nesse sentido, este artigo alinha-se a essas abordagens, ao propor e relatar a experiência de uma disciplina voltada à construção de Produtos Mínimos Viáveis em times, promovendo a aplicação dos conteúdos aprendidos ao longo do curso.

O objetivo deste artigo é compartilhar, por meio de um relato de experiência, os aprendizados e desafios enfrentados por docentes na condução da disciplina de Fábrica de Software em um curso de Bacharelado em Engenharia de Software (BES) ao longo de dois anos. Busca-se apresentar a metodologia adotada, os resultados observados, as dificuldades enfrentadas e as adaptações realizadas, com o intuito de contribuir com outros docentes e instituições interessadas em implementar iniciativas semelhantes. Como contribuição principal, este trabalho apresenta uma descrição prática da condução da disciplina, com base na experiência real de aplicação, destacando elementos processuais, ferramentas utilizadas, estratégias de avaliação, e reflexões críticas sobre o papel docente no processo de aprendizagem.

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta a fundamentação teórica sobre metodologias de ensino com base em projetos e ambientes simulados. A Seção 3 discute os principais trabalhos relacionados ao tema. A Seção 4 descreve a estrutura e a metodologia de condução da disciplina. A Seção 5 apresenta o relato de experiência dos docentes com base nas etapas metodológicas da disciplina. A Seção 5, intitulada *Perspectivas e Reflexões*, sintetiza os aprendizados, os principais

problemas enfrentados e as ações adotadas. Por fim, a Seção 7 apresenta as conclusões do trabalho.

2. Fundamentação Teórica

2.1. Desenvolvimento de Software e Fábrica de Software

O desenvolvimento de software é uma atividade complexa que envolve diversas fases interdependentes, tais como, levantamento de requisitos, análise, projeto, implementação, testes, implantação e manutenção. A engenharia de software busca sistematizar esse processo, promovendo qualidade, produtividade e previsibilidade na entrega de sistemas. No ensino superior, esses conceitos são tradicionalmente apresentados de forma segmentada, o que pode dificultar a compreensão do ciclo de vida completo de um projeto [Sommerville 2010].

Quando os conhecimentos sobre desenvolvimento são oferecidos de forma isolada em disciplinas, os alunos tendem a visualizar cada etapa como um processo estanque, sem perceberem as relações entre elas. Isso compromete a capacidade de entendimento da dinâmica real do desenvolvimento de software, em que decisões técnicas, prazos e demandas mudam constantemente. Nesse contexto, surge o conceito de Fábrica de Software como uma solução para integrar teoria e prática. O termo é utilizado tanto na indústria quanto na academia para descrever estruturas organizadas que visam sistematizar e padronizar o processo de desenvolvimento. Em ambientes de Fábrica de Software, processos bem definidos são aplicados com o objetivo de otimizar recursos humanos e tecnológicos, garantindo eficiência e qualidade.

No ambiente acadêmico, a Fábrica de Software funciona como um espaço de aprendizagem prática, onde os estudantes desenvolvem projetos reais ou simulados, assumindo papéis profissionais em equipes multidisciplinares. Essa vivência promove uma compreensão mais realista e integrada do processo de desenvolvimento e possibilita o desenvolvimento de competências técnicas e comportamentais fundamentais para a atuação profissional.

2.2. Metodologias de Ensino Baseado em Projetos

A Aprendizagem Baseada em Projetos, é uma metodologia ativa que propõe a construção do conhecimento por meio da realização de projetos complexos, contextualizados e com significado para os estudantes. Diferente do modelo tradicional centrado na transmissão de conteúdo, o PBL coloca os alunos como protagonistas do processo de aprendizagem, estimulando a autonomia, o pensamento crítico e a resolução de problemas [Gary 2015].

Sua aplicação no ensino de engenharia tem se mostrado eficaz, especialmente em contextos onde se busca integrar teoria e prática. Ao enfrentar desafios reais ou simulados em equipe, os estudantes têm a oportunidade de aplicar conhecimentos adquiridos em diferentes disciplinas, promovendo uma visão sistêmica dos problemas. Isso contribui para o desenvolvimento de competências técnicas e socioemocionais, como colaboração, liderança e comunicação.

Resultados reportados na literatura indicam que o PBL aumenta o engajamento dos alunos, melhora a retenção do conhecimento e favorece a formação de profissionais

mais preparados para o mercado de trabalho. O envolvimento direto em projetos permite que os estudantes compreendam a relevância do que estão aprendendo, tornando o processo educativo mais significativo [Cintra and Bittencourt 2023].

No contexto do ensino de Engenharia de Software, o PBL tem sido comumente utilizado em disciplinas que envolvem o desenvolvimento de sistemas, especialmente aquelas que simulam ambientes reais de trabalho. A implementação dessa metodologia, aliada à estrutura de uma Fábrica de Software, potencializa o aprendizado por meio da prática e prepara os estudantes para os desafios do exercício profissional.

3. Trabalhos Relacionados

A formação em Engenharia de Software enfrenta o desafio contínuo de conectar os fundamentos teóricos com as demandas práticas do desenvolvimento de sistemas complexos. Tradicionalmente, o ensino fragmentado por disciplinas dificulta a visão integrada do ciclo de vida do software, resultando em profissionais que, embora dominem conceitos isolados, podem ter dificuldades em aplicá-los de forma coesa em cenários reais [Ouhbi and Pombo 2020]. Essa separação entre teoria e prática motiva a busca por abordagens pedagógicas que promovam uma experiência de aprendizado mais holística e contextualizada.

Nesse contexto, metodologias ativas de ensino têm ganhado espaço, com destaque para a PBL. Revisões sistemáticas apontam benefícios do PBL na educação em Engenharia de Software, como maior engajamento e desenvolvimento de habilidades práticas, mas também alertam para desafios relacionados à avaliação do aprendizado individual, ao gerenciamento do escopo dos projetos e à carga de trabalho docente [Ouhbi and Pombo 2020, Hakak et al. 2023].

Uma manifestação comum de abordagens ativas e baseadas em projetos na Engenharia de Software é a implementação de “Fábricas de Software” em ambientes acadêmicos. Essas iniciativas buscam simular ambientes profissionais, proporcionando aos estudantes vivências práticas no desenvolvimento de software. Diversos relatos de experiência exploram modelos de Fábricas de Software acadêmicas, seja como projetos de extensão [Borges et al. 2012], projetos de ensino vinculados a cursos específicos [dos Santos et al. 2021], ou como estruturas mais permanentes para atender demandas internas e externas, envolvendo alunos de diferentes níveis [Neto et al. 2017, Brito 2013].

Os objetivos dessas iniciativas frequentemente proporcionam experiência prática relevante [Borges et al. 2012, dos Santos et al. 2021], capacitam os alunos com tecnologias de mercado [Borges et al. 2012], e, em alguns casos, suprem demandas de desenvolvimento das próprias instituições [Brito 2013]. A percepção discente sobre essas experiências tende a ser positiva, com relatos de melhor absorção de conteúdo em comparação com aulas expositivas tradicionais [Neto et al. 2017] e valorização do contato com problemas reais [dos Santos et al. 2021]. Uma pesquisa realizada por Oliveira *et al.*, com estudantes, professores e profissionais de TI também apontou a percepção da importância de Fábricas de Software interdisciplinares e de abordagens pedagógicas orientadas à solução de problemas [Oliveira et al. 2017].

Contudo, a implementação e condução dessas Fábricas de Software acadêmicas apresentam desafios recorrentes. A necessidade de uma metodologia de desenvolvimento bem definida e adaptada ao contexto acadêmico é um ponto crucial para garantir

a qualidade e a continuidade dos projetos, especialmente considerando a rotatividade de alunos [Brito 2013]. A gestão das equipes, a definição de papéis, o acompanhamento dos projetos e a avaliação do desempenho dos estudantes são aspectos complexos. Adicionalmente, a sobrecarga de atividades, tanto para alunos quanto para professores, é um fator de atenção, podendo ser intensificada por contextos específicos, como o ensino remoto [dos Santos et al. 2021]. A integração de alunos de diferentes períodos e níveis de conhecimento também requer estratégias pedagógicas e de gestão adequadas [dos Santos et al. 2021, Neto et al. 2017].

Em síntese, a literatura demonstra que as Fábricas de Software acadêmicas, frequentemente associadas a princípios de PBL, representam uma abordagem promissora para reduzir a lacuna entre teoria e prática na Engenharia de Software. Experiências relatadas indicam benefícios na aprendizagem e na preparação para o mercado. No entanto, desafios relacionados à estruturação metodológica, gestão de equipes e projetos, avaliação e carga de trabalho persistem. Este trabalho se insere nesse contexto, buscando contribuir ao relatar e analisar a experiência de condução de uma disciplina de Fábrica de Software focada na construção de MVPs ao longo de múltiplas ofertas, detalhando a metodologia empregada, os resultados observados, as dificuldades enfrentadas e as adaptações realizadas, com ênfase nas percepções e desafios sob a ótica docente.

4. Fábrica de Software no curso BES

A Fábrica de Software é uma disciplina curricular extensionista oferecida em conjunto para os 4º e 6º semestres do curso BES. Com carga horária semanal de quatro aulas de 45 minutos, totalizando 180 minutos por semana, seu foco não está na transmissão de conteúdo técnico, mas na vivência prática do desenvolvimento de software. Embora sejam apresentados conceitos pontuais sobre metodologias ágeis, gestão de projetos e validação com usuários, o principal objetivo da disciplina é permitir que os estudantes atuem diretamente no desenvolvimento de projetos reais.

A Figura 1 apresenta a metodologia adotada na disciplina, evidenciando as etapas que orientam desde a seleção inicial dos projetos até a avaliação final. A representação enfatiza a integração entre as atividades envolvendo os docentes, como Recebimento de Propostas, Seleção de Projetos, *Survey de Perfil*, Composição dos Times, Atribuição dos Projetos com as atividades dos estudantes, que são Execução com o *Scrum*, Validação com Usuários, Entrega final e *Survey de Avaliação*.

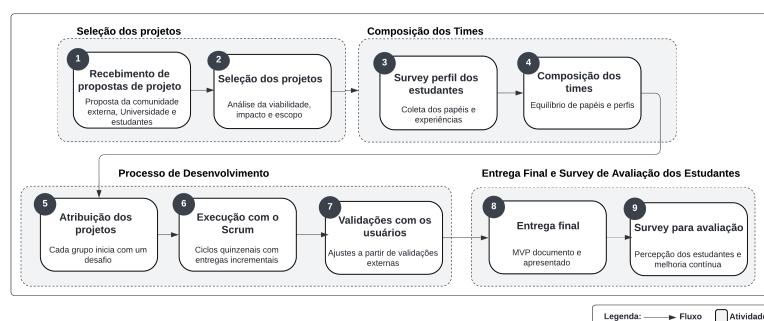


Figure 1. Visão geral da metodologia de ensino

A disciplina é ministrada por dois docentes da área de Engenharia de Software e ocorre em um espaço físico exclusivo dentro da universidade. A sala da Fábrica de Software é organizada com mesas retangulares dispostas em formato de ilhas, favorecendo a dinâmica de trabalho em equipe, a colaboração entre os membros dos times e o acompanhamento próximo por parte dos professores.



Figure 2. Layout do espaço físico da Fábrica

No início de cada semestre, os docentes realizam um processo de seleção de projetos, recebendo propostas oriundas da comunidade externa (instituições públicas e ONGs), da comunidade acadêmica e também de estudantes que desejam propor soluções com potencial de impacto social. Essas propostas são analisadas e filtradas pelos professores, considerando critérios como viabilidade técnica, escopo adequado para o semestre letivo e relevância social.

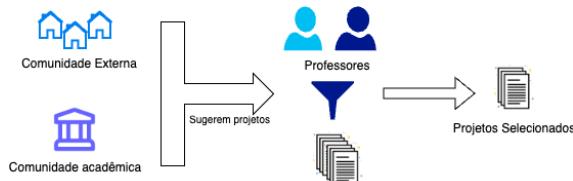


Figure 3. Fluxo de Seleção dos Projetos

Com os projetos definidos, os docentes conduzem uma dinâmica inicial para conhecer o perfil dos alunos matriculados. Por meio de um *survey*, são coletadas informações sobre as habilidades, tempo de experiência, tecnologias dominadas, senioridade (júnior, pleno ou sênior) e preferências de papel (frontend, backend, UI/UX, testador, engenheiro de requisitos, DBA, entre outros). Com base nesses dados, são compostos os times e cada equipe é alocada em um dos projetos selecionados.

A execução dos projetos segue a metodologia *Scrum*, adaptada ao contexto educacional. O trabalho é organizado em sprints quinzenais e reuniões semanais de acompanhamento (*status points*) com os docentes. O ciclo de desenvolvimento contempla desde o levantamento de requisitos e prototipação até o desenvolvimento, testes e validações. A gestão das atividades é realizada com ferramentas como *Trello* ou *Jira*, enquanto o controle de versão é feito com *GitHub*.

As tecnologias utilizadas nos projetos são escolhidas pelos próprios estudantes, de acordo com suas competências e preferências. Frameworks e linguagens como *React*, *Angular*, *Node.js*, *Django*, *Spring*, entre outros, são frequentemente adotados, além de recursos em nuvem como *AWS* e *Google Cloud*. Sempre que possível, os projetos são

submetidos a ciclos de validação com usuários reais e parceiros externos ao longo do semestre. Esse contato constante permite ajustes iterativos nas soluções, favorecendo uma aprendizagem centrada no usuário e alinhada com as necessidades do público-alvo.

Ao término da disciplina, os alunos respondem a um *survey* avaliando o processo de desenvolvimento, o engajamento individual e coletivo, a dinâmica de trabalho em equipe, as ferramentas utilizadas e a percepção sobre a qualidade das entregas. Os resultados são analisados pelos docentes e utilizados para ajustes e melhorias nos ciclos subsequentes.

5. Análise e Discussão dos Resultados

Esta Seção apresenta o relato de experiência dos docentes responsáveis pela condução da disciplina Fábrica de Software ao longo de dois anos de execução. A experiência é organizada em torno das principais etapas metodológicas adotadas: seleção de projetos, composição dos times, processo de desenvolvimento, entrega final e avaliação via *survey*. As reflexões destacam os aprendizados, ajustes realizados e os desafios enfrentados em cada etapa.

5.1. Seleção dos Projetos

A etapa de seleção de projetos tem se consolidado como um dos pilares da disciplina, tanto pelo seu impacto na motivação dos estudantes quanto pela relevância social das soluções desenvolvidas. Ao longo dos semestres, os docentes adotaram uma abordagem aberta à recepção de propostas oriundas de três frentes principais: comunidade externa, comunidade acadêmica e iniciativas espontâneas de alunos que propõem soluções tecnológicas com potencial de contribuição social.

A experiência tem mostrado que um dos maiores desafios dessa etapa está em equilibrar a relevância e a viabilidade técnica dos projetos com o tempo e os recursos disponíveis no semestre letivo. A seleção é realizada com base em critérios como aderência aos objetivos extensionistas, potencial de aprendizado para os estudantes, clareza dos requisitos iniciais e possibilidade de envolvimento com usuários reais. Além disso, busca-se distribuir projetos com diferentes níveis de complexidade entre os times, considerando também as composições de equipe planejadas.

Apesar dos cuidados no processo de curadoria, um dos principais desafios recorrentes enfrentados pelos docentes é a desmotivação de alguns estudantes em relação aos projetos designados. Em determinados casos, a falta de afinidade com a temática proposta ou a percepção de baixa complexidade técnica do desafio comprometem o engajamento individual e o desempenho coletivo do time. Essa situação evidencia a importância de alinhar, sempre que possível, os interesses dos estudantes com as demandas disponíveis, sem comprometer os critérios pedagógicos e extensionistas que orientam a disciplina.

5.2. Composição dos Times

Ao longo dos semestres, diferentes abordagens foram testadas pelos docentes no processo de formação dos times, com o objetivo de identificar estratégias mais eficazes para a criação de equipes equilibradas e produtivas. Inicialmente, foi adotado um modelo em que os próprios alunos eram responsáveis por montar suas equipes. A escolha era feita, em geral, com base em afinidade, colegas de turma, amigos próximos ou pessoas já

conhecidas. Embora esse formato promovesse um ambiente de conforto e familiaridade, observou-se que ele frequentemente resultava em grupos desequilibrados, compostos majoritariamente por estudantes de nível júnior ou com sobreposição de papéis (por exemplo, equipes com vários estudantes atuando como *backend* e nenhum responsável por testes ou UX). Essa formação comprometia a dinâmica do trabalho e gerava frustração interna entre os membros.

Em um segundo modelo, os docentes passaram a montar as equipes manualmente, com foco na distribuição dos papéis necessários para os projetos. Apesar de garantir que todas as funções estivessem cobertas, essa estratégia também apresentava limitações, especialmente pela desconsideração dos diferentes níveis de senioridade. Como consequência, era comum a formação de grupos compostos quase exclusivamente por alunos com pouca experiência, o que prejudicava a execução dos projetos e a qualidade dos entregáveis.

A melhor configuração observada ocorreu quando passamos a considerar tanto os papéis quanto os níveis de experiência dos estudantes. Para isso, foi aplicado no início do semestre um *survey* detalhado com os alunos, no qual coletamos informações sobre habilidades técnicas, tempo de experiência, senioridade (júnior, pleno ou sênior), tecnologias dominadas e preferências de atuação (*frontend*, *backend*, UI/UX, testes, engenharia de requisitos, etc.). Com esses dados, realizamos a composição dos times buscando equilibrar as competências e promover a colaboração entre perfis diversos. Apesar dos resultados positivos com esse modelo mais criterioso, o processo de formação manual dos times mostrou-se excessivamente trabalhoso, especialmente em turmas grandes com mais de 70 estudantes. Esse esforço evidenciou a necessidade de automatização da formação de equipes, levando à busca por soluções que possibilitem escalar a abordagem adotada, mantendo a qualidade e o equilíbrio dos times.

5.3. Processo de Desenvolvimento

O processo de desenvolvimento dos projetos é estruturado com base no framework *Scrum*, adaptado ao contexto educacional da disciplina. Na primeira edição da Fábrica de Software, o trabalho foi organizado em sprints semanais, com o objetivo de manter ciclos curtos e contínuos de entrega. No entanto, essa abordagem se mostrou ineficaz, pois, o intervalo de uma semana não era suficiente para que os estudantes realizassem entregas com substância, resultando em sprints dedicadas quase exclusivamente a reuniões de alinhamento, sem tempo hábil para desenvolvimento técnico mais sólido. Diante disso, nos semestres seguintes, foi realizada uma reformulação da dinâmica. Foi estabelecido encontros semanais denominados *status points*, com foco no acompanhamento do andamento dos projetos e orientação por parte dos docentes. As entregas, por sua vez, passaram a ser organizadas em sprints quinzenais, com *reviews* realizadas em sala de aula. Essa mudança permitiu maior maturidade nas entregas e melhor aproveitamento do tempo de desenvolvimento.

Outro ponto recorrente observado foi a dificuldade de comunicação entre os integrantes das equipes. Embora os alunos tivessem à disposição canais como *e-mail*, grupos de *WhatsApp* e servidores do *Discord*, a interação fora do horário das aulas era muitas vezes ineficiente. Os alinhamentos mais efetivos acabavam ocorrendo apenas durante os encontros presenciais da disciplina, o que comprometia o fluxo contínuo do projeto.

Para mitigar esse problema e promover uma comunicação mais estruturada, passamos a utilizar o Discord como ferramenta oficial da disciplina. Criamos canais exclusivos para cada time, nos quais toda comunicação deveria ocorrer, com o objetivo de manter o histórico de decisões e facilitar o acompanhamento por parte dos professores. Em algumas semanas, os próprios *status points* passaram a ser realizados dentro desses canais, otimizando o tempo de aula para que os estudantes pudessem dedicar-se às atividades técnicas com mais autonomia.

Essas adaptações têm contribuído significativamente para tornar o processo de desenvolvimento mais coerente com a realidade de equipes ágeis, ao mesmo tempo em que respeitam os limites do ambiente acadêmico. A experiência mostra que, além da metodologia, o suporte à comunicação e à gestão do tempo é um fator crítico para o sucesso dos projetos.

5.4. Entrega Final e Survey de Avaliação dos Estudantes

A etapa de encerramento da disciplina é marcada pela entrega final dos MVPs desenvolvidos ao longo do semestre. Cada equipe realiza uma apresentação formal do seu projeto, demonstrando as principais funcionalidades implementadas, as tecnologias utilizadas e as decisões de projeto tomadas durante o processo. Foi observado que o nível de maturidade dos projetos ao final do semestre varia conforme a composição dos times, a complexidade do desafio e o grau de engajamento dos estudantes. Em geral, os grupos conseguem entregar sistemas funcionais, navegáveis e com funcionalidades alinhadas aos requisitos iniciais. No entanto, nem sempre o escopo previsto é totalmente atendido, sendo comum a priorização de funcionalidades centrais em detrimento de aspectos secundários. Essa priorização, inclusive, é vista como um aprendizado valioso dentro da abordagem orientada a MVP.

Após a entrega final, é aplicado um *survey* com o objetivo de obter a percepção dos estudantes sobre a disciplina. As questões abordam aspectos como a dinâmica de trabalho em equipe, clareza dos papéis, qualidade das interações, ferramentas utilizadas, participação dos docentes, dificuldades enfrentadas e percepção sobre o próprio desempenho e o do grupo. A análise dos dados tem se mostrado fundamental para a evolução contínua da disciplina.

De forma geral, os estudantes reconhecem a Fábrica de Software como uma das experiências mais marcantes do curso, destacando o caráter prático, o contato com demandas reais e a oportunidade de aplicar conhecimentos adquiridos em outras disciplinas. Por outro lado, os principais pontos de atenção referem-se à distribuição de tarefas dentro dos times, desafios de comunicação e discrepâncias no nível de dedicação entre os membros. Esses *feedbacks* têm sido utilizados para ajustes metodológicos e intervenções mais pontuais ao longo dos semestres, fortalecendo o processo de ensino-aprendizagem e a própria gestão da disciplina.

6. Perspectivas e Desafios

A experiência acumulada ao longo de dois anos na condução da disciplina Fábrica de Software tem reforçado a importância de metodologias ativas, como o PBL, na formação de engenheiros de software. A proposta de desenvolver projetos reais em um ambiente colaborativo permitiu aos estudantes aplicar conhecimentos teóricos em situações práticas. Durante esse percurso, diversos desafios foram identificados, exigindo adaptações contínuas

na metodologia. A Tabela 1 apresenta um resumo dos principais problemas enfrentados e as ações implementadas para mitigá-los.

Table 1. Principais problemas enfrentados e ações adotadas

Problemas	Ações adotadas
Desmotivação com projetos não alinhados ao interesse dos alunos	Curadoria docente com critérios de escopo e impacto, além de acolhimento de propostas dos próprios alunos
Desequilíbrio nas equipes formadas livremente	Aplicação de survey para coleta de perfil técnico e composição equilibrada dos times
Sobrecarga na formação manual dos times	Proposição de automatização do processo com base nos dados coletados via survey
Sprint semanal com entregas frágeis	Adoção de sprints quinzenais e encontros semanais de <i>status point</i>
Baixa comunicação entre os membros dos times	Utilização do Discord como canal oficial, com histórico e canais por equipe

A condução da disciplina evidencia que, embora o PBL traga benefícios significativos, sua implementação exige um esforço constante de planejamento, acompanhamento e adaptação. A atuação docente se estende para além da mediação de conteúdo, assumindo também papéis de mentoria, curadoria e gestão de times. Como perspectiva futura, pretende-se avançar na automação do processo de formação de equipes, sistematizar os critérios de avaliação por meio de rubricas mais consistentes e ampliar a parceria com instituições externas para diversificar ainda mais os projetos. Além disso, há o interesse em promover estudos de acompanhamento ao longo do tempo sobre o impacto dessa experiência na formação profissional dos egressos.

7. Conclusões

A experiência de condução da disciplina Fábrica de Software ao longo de dois anos evidenciou o potencial da PBL no ensino de Engenharia de Software. A proposta de integrar estudantes em times multidisciplinares, atuando em projetos com impacto social e orientados por práticas ágeis, possibilitou o desenvolvimento de competências técnicas de forma colaborativa. A vivência prática proporcionada pela disciplina favoreceu o engajamento discente, a aplicação integrada de conhecimentos adquiridos ao longo do curso e o contato direto com demandas reais.

Ao mesmo tempo, a trajetória revelou desafios importantes, como a complexidade na composição equilibrada dos times, a necessidade de estratégias eficazes de comunicação entre os membros e a adequação de escopo dos projetos à realidade do semestre letivo. As ações implementadas ao longo dos ciclos, como a formalização dos canais de comunicação, o uso de surveys para formação das equipes e a reestruturação da cadência de entregas, demonstraram-se eficazes na mitigação desses problemas. A disciplina segue em constante aprimoramento, com perspectivas de automatização de processos e aprofundamento na análise dos impactos formativos na visão dos egressos, podendo servir como referência para outras iniciativas de ensino que buscam aproximar teoria e prática.

References

- Borges, K. S., Carvalho, T. P. d., and Moraes, M. A. C. d. (2012). Programa de extensão “fábrica de software acadêmica”: contribuindo para a formação profissional na área da informática. In *Anais do XX Workshop sobre Educação em Computação (WEI)*, Curitiba, PR. Parte do CSBC 2012. Páginas podem variar.
- Brito, M. C. A. (2013). Elaboração de uma metodologia de desenvolvimento de software em uma instituição de ensino. *Revista Brasileira de Informática na Educação*, 21(02):52.
- Cintra, C. and Bittencourt, R. (2023). As experiências de estudantes em um curso de engenharia de computação baseado em pbl. In *Anais do XXXI Workshop sobre Educação em Computação*, pages 327–338, Porto Alegre, RS, Brasil. SBC.
- dos Santos, A. S., Neves, M., Rodrigues, Y., de Oliveira, N. H., Kuhn, D., Santos, G., and Silva, R. A. (2021). Experiência do projeto “fábrica de software” em um curso de engenharia de software. In *Escola Regional de Engenharia de Software (ERES)*, pages 89–98. SBC.
- Fioravanti, M. L., Sena, B., Paschoal, L. N., Silva, L. R., Allian, A. P., Nakagawa, E. Y., Souza, S. R., Isotani, S., and Barbosa, E. F. (2018). Integrating project based learning and project management for software engineering teaching: An experience report. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education, SIGCSE ’18*, page 806–811, New York, NY, USA. Association for Computing Machinery.
- Gary, K. (2015). Project-based learning. *Computer*, 48(9):98–100.
- Hakak, S., Jamlos, M. F. B., Nasir, M. T., and Affendi, Z. H. (2023). Benefits and challenges of project-based learning for software engineering education: A systematic literature review. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 14(1):970–978.
- Neto, E. B. C., Lopes, A. S. B., and Nascimento, D. S. C. (2017). Um relato de experiência da implantação de um modelo de fábrica de software escola (fases). In *Workshop sobre Educação em Computação (WEI)*, pages 2247–2256. SBC.
- Oliveira, M. d. C. O. d., Oliveira, S. R. B., and Meira, S. L. (2017). Condução de uma fábrica de software e o processo de aprendizagem em cursos de graduação em ti: Uma aplicação de um survey sobre a percepção da importância. In *Anais do XXVIII Simpósio Brasileiro de Informática na Educação (SBIE 2017)*, pages 92–101, Recife, PE. Sociedade Brasileira de Computação (SBC).
- Ouhbi, S. and Pombo, N. (2020). Software engineering education: Challenges and perspectives. In *2020 IEEE Global Engineering Education Conference (EDUCON)*, pages 202–209.
- Pérez, B. and Rubio, A. L. (2020). A project-based learning approach for enhancing learning skills and motivation in software engineering. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education, SIGCSE ’20*, page 309–315, New York, NY, USA. Association for Computing Machinery.

- Pucher, R. and Lehner, M. (2011). Project based learning in computer science – a review of more than 500 projects. *Procedia - Social and Behavioral Sciences*, 29:1561–1566. The 2nd International Conference on Education and Educational Psychology 2011.
- Sommerville, I. (2010). *Software Engineering*. Addison-Wesley Publishing Company, 9th edition.
- Souza, M., Moreira, R., and Figueiredo, E. (2019). Students perception on the use of project-based learning in software engineering education. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering*, SBES '19, page 537–546, New York, NY, USA. Association for Computing Machinery.