

Práticas pedagógicas com tecnologias digitais em cursos superiores de Computação: construção de software em uma ação de extensão universitária

Paulo H. S. Oliveira¹²³, Carine R. Costa¹, Olinta M. F. Oliveira⁴, Eugênio J. M. C. Carvalho², Rogéria L. W. Gonçalves², Rosemara P. Lopes¹

¹ Faculdade de Educação (FE) – Programa de Pós-graduação em Educação (PPGE) – R. 235, 307 – Setor Leste Universitário, Goiânia - GO, 74605-050.

² Instituto Acadêmico de Ciência e Tecnologia (IACT) – Universidade Estadual de Goiás (UEG) – Rod. BR-153, Quadra Área, Km 99, S/n - Distrito Agroindustrial de Anápolis (DAIA), Anápolis – GO – Brasil.

³ Centro de Recursos Computacionais (CERCOMP) – Campus Goiás da Universidade Federal de Goiás (UFG) – Av. Bom Pastor, S/n - Setor Aretião, Goiás - GO, 76600-000.

⁴ Colégio Tecnológico do Estado de Goiás (COTEC) – COTEC Goiandira Ayres do Couto – Rua Aeroporto, 19-141 – CEP: 76600-000 – Goiás-GO.

paulo.oliveira@ueg.br, carine.rodrigues@ifmt.edu.br,
olintamartins88@gmail.com, eugenio.carvalho@ueg.br,
rogeria.goncalves@ueg.br, rosemaralopes@ufg.br

Abstract. This paper reports a set of pedagogical activities applied in the Software Construction and Interdisciplinary Software Development Practice courses of the bachelor's degree in Information Systems, with the aim of discussing and improving teaching practice in contemporary society. For this purpose, a teaching methodology based on software development trails is described, which covers topics ranging from requirements elicitation to prototyping, data modeling, coding and documentation. In addition, the results of online surveys applied to students are presented, concluding with questions about teacher training and educational infrastructure.

Resumo. Este trabalho relata atividades pedagógicas aplicadas nas aulas das disciplinas Construção de Software e Prática Interdisciplinar de Desenvolvimento de Software, do curso de bacharelado em Sistemas de Informação, com o objetivo de debater e aprimorar a prática docente na sociedade contemporânea. Para tanto, descreve-se uma metodologia de ensino baseada em trilhas de desenvolvimento de software, que abrange desde a elicitação de requisitos até a prototipação, modelagem de dados, codificação e documentação. Adicionalmente, são apresentados resultados de questionários on-line aplicados aos alunos, concluindo com indagações sobre formação de professores e infraestrutura educacional.

1. Introdução

Na sociedade contemporânea, as pessoas estão cada vez mais conectadas a dispositivos eletrônicos e digitais, os quais permitem realizar atividades pessoais, profissionais e outras, por meio da Internet. Desse modo, parte considerável da

vida das pessoas está atrelada aos usos da Internet e aos milhares de sistemas de informação e comunicação (SI). As tecnologias digitais de informação e comunicação (TDIC) avançam a passos largos e, nesse sentido, está consolidada uma transição para uma sociedade digitalizada, onde a informação se torna o fator central para a organização social e econômica (Masuda, 1980).

Para Castells (1999), o impacto das TDIC e das redes globais na transformação da sociedade, especialmente no que se refere à economia, política e comunicação, é determinante. Na perspectiva de Ferreira (2000, p. 92), “[...] a análise sociológica das novas tecnologias e da organização do trabalho revela-se cada vez mais pertinente”. Importa ressaltar que, segundo Bell (1973), a informação e os serviços tomariam o lugar das indústrias tradicionais, e, de fato, em alguma medida, é o que observamos nos dias atuais. Alguns exemplos são empresas operando por meio de diversos SI que gerenciam estoques, compra e venda, transações financeiras, *marketing* e vendas, setor de serviços, de forma presencial ou virtual, algoritmos que utilizam Inteligência Artificial (IA) e quase tudo aquilo que envolve conjuntos de dados. A esse respeito, Valente (2018) acrescenta que as profissões estão evoluindo rapidamente com a chegada da Indústria 4.0.

O contexto supramencionado se correlaciona com a educação em Computação, em especial a educação superior em Sistemas de Informação e áreas afins, que trata da criação de sistemas de informação, isto é, a partir da elicitação de regras de negócio (como as organizações empresariais funcionam), para, assim, desenvolver sistemas computacionais que possam otimizar os processos organizacionais. Em tempos digitais, ensinar como desenvolver esses sistemas, e ao mesmo tempo, inovar em sala de aula, torna-se uma necessidade, quando se busca o engajamento dos alunos, aliado à compreensão do conteúdo de ensino. Com base em Gatti *et al.* (2019), comprehende-se o conceito de inovação no âmbito das práticas pedagógicas, como transformações que acontecem em um contexto histórico específico, não necessariamente sendo novas ou inéditas, mas visando aprimorar estruturas ou processos previamente estabelecidos.

Tendo em vista esses pressupostos, busca-se elencar fatores e aspectos considerados essenciais no processo de ensino-aprendizagem das disciplinas “Construção de *Software*” e “Prática Interdisciplinar de Aplicações em Sistemas de Informação”, trabalhadas de forma conjunta, as quais são componentes curriculares obrigatórios, ofertados no sexto semestre da matriz curricular do curso superior em Sistemas de Informação, de modo presencial, sendo um professor regente em cada disciplina. O curso é ofertado no Campus Cora Coralina, Unidade Universitária Itaberaí, no município de Itaberaí, Goiás. Nessas disciplinas foram utilizadas abordagens pedagógicas com TDIC, bem como estratégias que estimularam a interação com a comunidade local. Ainda, retomando Gatti *et al.* (2019), entende-se o professor como um ser social inserido em um processo contínuo de desenvolvimento profissional. Portanto, compartilhar vivências como as relatadas neste trabalho permite momentos de aprendizado, trocas de experiências e reflexões voltadas ao aperfeiçoamento profissional docente.

Levando em conta o objetivo acima, na seção seguinte, são mencionados os artefatos tecnológicos e as práticas para construção de *software*. Dando continuidade, na terceira seção, discorre-se sobre os procedimentos pedagógicos realizados nas referidas disciplinas. Na quarta seção, descreve-se a trilha de desenvolvimento criada durante a execução das mesmas, passando, em seguida, à quinta seção, que aborda as percepções dos discentes sobre a metodologia aplicada para o desenvolvimento de *software*. Por fim, a sexta e última seção contém considerações finais sobre o exposto.

2. Tecnologias em aulas teóricas e práticas de construção de *software*

As disciplinas universitárias abordadas neste relato abrangeram práticas de desenvolvimento de *software* e extensão universitária, uma vez que houve o envolvimento da comunidade local, não acadêmica. Merece destaque o ‘fazer extensão’, que “[...] como processo educativo, cultural e científico, traz o compromisso de organização de ações nas quais a comunidade externa à universidade é protagonista” (Melo; Werz, 2018, p. 66).

Diante de trabalhos relacionados (Abich; Cavalheiro; Parizi, 2024; Franco; Franco, 2023; Tives; Marini; Canedo, 2024), infere-se que o diálogo com a sociedade pode trazer novos saberes, a partir da união e do debate entre os conhecimentos acadêmico e popular (Melo; Werz, 2018), que ocorre a partir da interação entre desenvolvedores de *software* e público-alvo, neste caso, a comunidade local.

O propósito das disciplinas ensejou possibilitar aos alunos: a) interagir com a comunidade na busca por possíveis empresas/instituições e ao escolher um comércio, b) cumprir o que estabelecem os dispositivos legais da universidade, que indicam a obrigatoriedade de curricularização de atividades de extensão¹ e c) colocar em prática os conhecimentos e saberes adquiridos ao longo de todo o curso, bacharelado em Sistemas de Informação, para o desenvolvimento de um sistema informatizado que atendesse aos anseios do público-alvo selecionado. Adiante, é descrita a metodologia de trabalho nas disciplinas.

Cabe enfatizar a relevância e o desafio de ministrar disciplinas na área de Computação de forma que os alunos permaneçam nos cursos. Segundo Pinto e Nascimento (2018, p. 304), “[...] uma das principais razões do alto índice de reprovação e de desistência dos cursos de computação e de tecnólogos nas áreas de TI se deve à falta de compreensão dos problemas e dificuldade de raciocínio lógico”. No contexto das disciplinas, práticas pedagógicas, abrangendo aulas teórico-práticas e orientações, foram essenciais para a união dos três grupos de alunos formados por três ou quatro integrantes cada, para as práticas aqui elencadas, o entendimento da proposta das disciplinas e também para o sucesso quanto ao desenvolvimento dos *softwares*.

As aulas teóricas foram ministradas com o propósito de revisar conteúdos e habilidades necessárias para a concepção e desenvolvimento de *software*. Os *softwares* gerados nestas disciplinas, detalhados *a posteriori* (Quadro 1), foram delineados a partir de duas diretrizes principais, determinadas pelos professores das disciplinas. A primeira

¹ Conforme Resolução do Conselho Nacional de Educação/Câmara de Educação Superior (CNE/CES) nº 7, de 18 de dezembro de 2018, que delibera acerca da regulamentação sobre ações de extensão na educação superior brasileira (Brasil, 2018).

diretiva definiu que o *software* a ser desenvolvido pelos discentes pudesse ser acessado em diferentes sistemas operacionais (multiplataforma), desde que houvesse conexão com a Internet. A segunda diretiva determinou que o sistema deveria ser responsivo, ou seja, que pudesse ser acessado e se ajustasse aos tamanhos das telas de dispositivos diversos, como *smartphones*, *tablets* e outros.

A partir das diretivas iniciais, os professores ministrantes das disciplinas sugeriram as tecnologias para atender ao propósito planejado, quais sejam: linguagem de programação *Hypertext Preprocessor* (PHP versão 8 ou superior), linguagem de marcação *Hypertext Markup Language* (HTML), linguagem de estilos *Cascading Style Sheets* (CSS), linguagem de programação JavaScript (JS), linguagem *Structured Query Language* (SQL), *framework* para *front-end* (*Bootstrap* versão 5), BrModelo versão 3.2 ou superior, planilhas eletrônicas e editores de texto, SQLYog ou similar e banco de dados MySQL versão 5.3 ou superior, sendo esta última a tecnologia responsável pela criação de um banco de dados relacional para armazenamento e manipulação de dados.

As tecnologias supramencionadas contextualizam um extenso e complexo conjunto de possibilidades para a concepção e desenvolvimento de um *software*. Cada uma delas contribuiu para que os sistemas de informação pudessem ser iniciados e concluídos.

3. Abordagem inicial e procedimentos pedagógicos

A proposta de execução das disciplinas teve viés interdisciplinar e extensionista. Nesse sentido, a turma composta por 11 alunos foi dividida em três grupos, balanceando a quantidade de integrantes por grupo, na medida do possível. Uma vez determinados os grupos e o líder de cada grupo, foi realizada a apresentação das ementas e objetivos das disciplinas, de modo pormenorizado, explicou-se a cada grupo sobre a escolha de uma empresa da comunidade que pudesse ser estudada e informatizada.

A partir da escolha de uma empresa local, os alunos foram orientados a se reunir com os solicitantes do *software* e seguir as etapas imprescindíveis ao desenvolvimento de *software*, previamente ensinadas em disciplinas ministradas desde o início do curso superior. As etapas são: a) reunião inicial entre os integrantes do grupo, b) reunião inicial de cada grupo com o requerente da aplicação, com vistas à coleta de requisitos, c) análise de requisitos, d) criação do documento de requisitos, e) criação dos modelos conceitual, lógico e físico, f) criação do modelo entidade-relacionamento (MER), f) prototipação visual das telas em baixo, médio e alto nível, g) início do desenvolvimento do código *front-end* e *back-end*, h) testes de *software* ao longo do desenvolvimento, i) reunião semanal com os professores das disciplinas para alinhamento de ideias e j) entrega de módulos de cada sistema.

Os professores das disciplinas resgataram os saberes técnicos junto a cada grupo, e ressaltaram, principalmente, o fator tempo/prazo para a execução de *software*, considerando que as disciplinas são ofertadas semestralmente no curso. Ao ser apontado o fator “prazo”, foi delimitado um cronograma, de modo que os alunos pudessem se concentrar e organizar o tempo entre os membros de seus respectivos grupos para produzir o *software*, cada um com uma tarefa ou todos os membros envolvidos com

todas as tarefas. Para cada grupo, foi designado um líder, responsável por determinar a divisão de tarefas ou desenvolvê-las conjuntamente.

Nomear um líder é uma ação intencional e remete à disciplina Gestão de Projetos, um dos componentes curriculares do curso, cujo objetivo é estabelecer planos de gerenciamento no escopo de projetos, inclusive, projetos de desenvolvimento de *software*. Nesse sentido, a disciplina prevê diversas abordagens, como gerenciamento de tempo, gerenciamento de custos (não incluso no desenvolvimento destes *softwares*), gerenciamento de riscos, de aquisições etc. O líder do grupo e os demais alunos foram orientados e esclarecidos quanto às responsabilidades envolvidas em cada parte do projeto de desenvolvimento. Reuniões semanais de orientação foram conduzidas pelos professores de cada disciplina.

4. Trilha de desenvolvimento de *software* e ensino

A partir da retomada de conhecimentos adquiridos em disciplinas anteriores do curso, foi proposto um instrumento, com o objetivo de guiar o desenvolvimento de *software*, similar à uma trilha, composta por seis etapas e respectivos passos bem definidos. Cada grupo, de acordo com sua escolha, teve autonomia para visitar comércios locais da cidade e propor o desenvolvimento de um *software* que pudesse informatizar a empresa escolhida. O Quadro 1 apresenta a trilha de desenvolvimento de *software*, fundamentada em Pressman (2021) e Sommerville (2011), utilizada no decorrer do semestre letivo.

		ETAPAS E TAREFAS		Incluído?	
				SIM	NÃO
		PLANEJAMENTO INICIAL			
ETAPA 01	1. Definição do escopo: identificar os objetivos do <i>software</i> e as principais funcionalidades.		(X)	()	
	2. Estudo de viabilidade: avaliar aspectos técnicos, econômicos e legais.		()	(X) ²	
	3. Stakeholders: identificar e envolver as partes interessadas.		(X)	()	
ENGENHARIA DE REQUISITOS					
ETAPA 02	1. Concepção: identificar os <i>stakeholders</i> e seus diferentes pontos de vista sobre o negócio e o <i>software</i> proposto. Desenhar a visão geral do <i>software</i> .		(X)	()	
	2. Elicitação: levantar requisitos de usuário do sistema sob duas perspectivas, sendo elas “requisitos funcionais” e “requisitos não funcionais”.		(X)	()	
	3. Elaboração: detalhar cada requisito descrito em linguagem natural em modelos conceituais, como SysML, UML ou FAD. Técnica que visa eliminar ambiguidades, inconsistências, omissões e erros dos requisitos.		(X)	()	
	4. Negociação: identificar os conflitos entre os requisitos para negociar as soluções com os <i>stakeholders</i> , priorizando, eliminando, combinando ou modificando os requisitos.		(X)	()	
	5. Especificação: especificar, em termos técnicos, ou seja, desenvolver os requisitos de sistema que devem atender aos requisitos de usuário. Na especificação passa-se da perspectiva do problema (requisitos de usuário) para a perspectiva da solução (requisitos de sistema).		(X)	()	
	6. Validar: validar a cobertura do sistema, ou seja, o atendimento de todos os requisitos de usuário pelo sistema proposto. É homologado o aceite dos <i>stakeholders</i> sobre os requisitos desenvolvidos.		(X)	()	
	7.1 Gerenciamento: garantir o escopo do produto, certificando o processo de		(X)	()	

² Em ambientes profissionais de desenvolvimento de *software* (financiado), essa tarefa seria incluída no projeto.

	<p>desenvolvimento e o atendimento de seus requisitos.</p> <p>7.2 Gestão de mudanças: garantir a rastreabilidade das mudanças durante o processo de desenvolvimento e realizar análises de impacto das mudanças propostas para evidenciar sua viabilidade técnico-financeira.</p> <p>8. Documento de requisitos de software (DRS): detalhar, claramente, as necessidades do <i>software</i>, incluindo objetivo, escopo e público-alvo. Listar requisitos funcionais e não funcionais, além de restrições, premissas e critérios de aceitação para validação. Diagramas, <i>mockups</i> e fluxogramas complementam o documento, tornando-o uma referência essencial para garantir o alinhamento entre equipe e <i>stakeholders</i> ao longo do projeto.</p>		
ETAPA 03	<p style="text-align: center;">PROTOTIPAÇÃO VISUAL</p> <p>1. Wireframes (esboço inicial): esboçar telas iniciais, em busca da definição da estrutura e o <i>layout</i> do sistema, de forma simples, sem foco em detalhes visuais. Trabalha-se nessa fase o posicionamento de elementos, como botões, menus e caixas de texto. Aplicativos como <i>Balsamiq</i>, <i>Figma</i> e <i>Wireframe.cc</i> podem ser usados, pois permitem uma visão clara da disposição funcional antes da aplicação do <i>design</i> final.</p> <p>2. Mockups (alta fidelidade): versões detalhadas das telas, apresentando cores, fontes e imagens para simular a aparência final do sistema. O objetivo é definir o estilo visual, incluindo tipografia e paleta de cores, garantindo alinhamento com a identidade da marca.</p> <p>3. Prototipação interativa: criação de uma experiência navegável, simulando o fluxo do usuário no sistema. Em tecnologias como <i>InVision</i> ou <i>Figma</i>, é possível testar a interface e obter <i>feedbacks</i> de usuários reais. Objetiva identificar melhorias e realizar ajustes antes da implementação final, garantindo uma experiência mais intuitiva e eficiente.</p> <p>4. Prototipação visual em HTML: analogamente ao item anterior, essa prototipação visa criar telas em HTML e estilizadas com CSS para que o solicitante do <i>software</i> veja como será a navegação pelo sistema.</p>	(X)	()
ETAPA 04	<p style="text-align: center;">MODELAGEM DE DADOS</p> <p>1. Modelo Entidade-Relacionamento (MER): é uma representação conceitual (modelo conceitual) do banco de dados, usado para descrever as entidades (objetos principais do sistema) e seus relacionamentos (tipos de relacionamento (1:1, 1:N, N:N)), sem considerar aspectos técnicos como tipos de dados ou restrições específicas de um Sistema Gerenciador de Banco de Dados (SGBD).</p> <p>2. Modelo lógico: representar, detalhadamente, o banco de dados, a partir do MER. Organiza as entidades em tabelas e define atributos como colunas, com tipos de dados genéricos. Introduz as chaves primárias (PK) e estrangeiras (FK) para garantir a integridade dos dados e descreve regras de negócios, como restrições de unicidade ou obrigatoriedade. Ocorre ainda o processo de normalização para eliminar redundâncias.</p> <p>3. Modelo físico: implementar, de fato, o banco de dados em um SGBD específico, traduzindo o modelo lógico em <i>scripts SQL</i>. Nessa fase são definidos os tipos de dados específicos do sistema escolhido (como <i>VARCHAR</i> ou <i>INTEGER</i>) e índices para melhorar o desempenho, incluindo elementos como <i>triggers</i>, <i>views</i> e <i>stored procedures</i>. Além disso, são consideradas particularidades técnicas, como particionamento de tabelas, ajuste de armazenamento e configurações de segurança, garantindo que o banco de dados funcione de maneira eficiente e confiável no ambiente de produção.</p>	(X)	()
ETAPA 05	<p style="text-align: center;">DESENVOLVIMENTO E CODIFICAÇÃO</p> <p>1. A implementação define a arquitetura do <i>software</i>, como monolítica, MVC ou microserviços, conforme a complexidade do projeto. O <i>back-end</i> trata da lógica de negócios, APIs e segurança, enquanto o <i>front-end</i> cria interfaces responsivas</p>	(X)	()

	e centradas na experiência do usuário.		
	2. Para garantir a qualidade, são realizados testes unitários, de integração, funcionais, de performance e de usabilidade, avaliando desde partes do código até a interação entre módulos, conformidade com requisitos e experiência do usuário.	(X)	()
	3. A implantação envolve a configuração do sistema na linguagem PHP em um <i>host</i> específico para hospedagens <i>web</i> , permitindo o acesso em tempo real via URL. Ao longo do desenvolvimento e após o lançamento, a manutenção foca na correção de <i>bugs</i> , atualizações e suporte técnico, garantindo a continuidade e adaptação do <i>software</i> às necessidades dos usuários.	(X)	()
DOCUMENTAÇÃO E TREINAMENTO			
ETAPA 06	1. Documentação: reúne todas as informações essenciais do sistema, incluindo as modelagens de dados (conceitual, lógico e físico), fluxos de trabalho e especificações técnicas. Além disso, um manual do sistema é elaborado para orientar os usuários sobre suas funcionalidades, desde o acesso até o uso avançado de recursos, garantindo clareza e suporte contínuo.	(X)	()
	2. Treinamento: é oferecido aos usuários finais e equipes técnicas para garantir o domínio das funcionalidades e a utilização eficiente do sistema. Essa etapa pode incluir <i>workshops</i> , tutoriais interativos e materiais de apoio, promovendo uma transição suave para o uso do sistema e maximizando seu valor.	()	(X) ³

Quadro 1. Trilha para desenvolvimento de software. Fonte: Autores (2024).

A partir da trilha de desenvolvimento de *software*, considerando as etapas apontadas no Quadro 1, as disciplinas foram ministradas focando no contato direto com os alunos, de modo a esclarecer as inúmeras dúvidas que tiveram ao longo de todo o processo. Algumas aulas foram teóricas e objetivaram relembrar conceitos importantes abordados ao longo da formação acadêmica do estudante. Outras aulas foram demonstrativas, momento em que os professores fizeram codificações similares ao que seria necessário aos sistemas. Os docentes ainda se reuniram com os alunos semanalmente, desde o início das disciplinas, para discutir todas as fases do desenvolvimento, orientando-os sobre como realizar as entrevistas e reuniões com os solicitantes do *software* até a modelagem dos dados, codificação e documentação do sistema.

Cada tarefa das etapas dispostas no Quadro 1 foi conduzida pelos professores das disciplinas a partir do diálogo com os líderes e demais membros de cada grupo. A orientação quanto à aplicação de técnicas de desenvolvimento de *software* foi amparada pelo instrumento criado, chamado de “trilha de aprendizagem de *software*”, um caminho planejado a ser percorrido na condução das disciplinas e nos processos de ensino-aprendizagem.

5. Percepção dos alunos quanto à metodologia das aulas

Este estudo de cunho descritivo e propositivo teve abordagem colaborativa, com traços da pesquisa-ação, na medida em que pesquisadores (professores) e participantes (discentes) colaboraram na identificação de problemas, planejamento e execução de ações, e avaliação dos resultados. Para Gil (2008), a pesquisa-ação combina a produção de conhecimento com a resolução de problemas concretos, destacando sua aplicação em

³ O sistema não está totalmente concluído, portanto, até o momento em que este texto foi redigido, não houve treinamento.

ambientes educativos e comunitários. Ademais, Minayo (2014) destaca que a pesquisação é uma prática essencialmente qualitativa, enfatizando a participação ativa dos sujeitos no processo e sua contribuição para mudanças sociais. Apesar disso, este trabalho não se caracteriza como relato de pesquisa, haja vista não arrolar o Comitê de Ética em Pesquisa, mas um relato de experiência de cunho investigativo.

Ao final das disciplinas, foi realizada a aplicação de um questionário eletrônico⁴, via *Google Forms*, a fim de identificar a compreensão dos alunos em relação a aspectos abordados em ambas as disciplinas e, de modo geral, as percepções ao planejar e desenvolver uma solução tecnológica para atender um requisitante de *software* em potencial, ou seja, uma pessoa da comunidade.

De um total de 11 alunos, dez responderam ao questionário, sendo o grupo composto por jovens com idade entre 18 e 35 anos. Ao todo, foram desenvolvidos três sistemas, voltados para serviços de oficina mecânica, restaurante e *petshop*.

Em relação às práticas pedagógicas dos professores, os dados obtidos indicaram que a maioria dos participantes as avaliaram satisfatoriamente, durante o desenvolvimento do projeto de *software*. As reuniões semanais com os professores, a utilização de exemplos práticos de codificação relacionados ao projeto foram consideradas úteis, por conferirem sentido ao trabalho realizado, juntamente com a orientação sobre questões estruturais, com nove menções indicando a relevância desses momentos, sugerindo que o direcionamento foi adequado para a criação do *software*.

Ainda sobre as práticas, a proposta de desenvolver um sistema voltado para uma empresa local foi bem recebida (oito menções), corroborando e evidenciando a integração entre teoria e prática. Consequentemente, as disciplinas conseguiram atender à necessidade de conectar o conhecimento acadêmico com problemas reais da comunidade não acadêmica.

No entanto, a avaliação das metodologias de sala de aula e o desempenho do líder de grupo, com sete menções, indicam que houve espaço para melhorias nesses aspectos. Destarte, a dificuldade enfrentada na modelagem de dados (sete menções) revelou a complexidade do processo, embora a orientação recebida tenha contribuído para superá-la. Em síntese, os dados indicam que, embora a experiência tenha sido em grande parte satisfatória, há pontos que podem ser aprimorados para otimizar os resultados do processo de ensino-aprendizagem e desenvolvimento do projeto.

Perguntados sobre a preparação para atuar no mercado de trabalho após a experiência, três respondentes informaram que se sentiam aptos, seis marcaram “neutro”, e somente um discordou. Nesse caso, o fato de o mercado de trabalho tecnológico exigir cada vez mais o domínio de diversas tecnologias e outras qualidades profissionais pode ter influenciado na opção pela alternativa “neutro”. Por fim, nove discentes informaram que recomendariam as disciplinas cursadas para outros alunos, e somente um respondeu “talvez”.

Questionados sobre o nível de exigência nas disciplinas, sete estudantes consideraram que foi exigido “na medida correta”, dois assinalaram “um pouco mais” e

⁴ Disponível em: <https://forms.gle/PsG69PFZGb75QXF1A>. Acesso em: 17 fev. 2025.

um afirmou que poderia ter sido cobrado “muito mais”, evidenciando, assim, que todo o processo foi considerado possível de execução.

Partindo da premissa de que toda prática pedagógica é também social (Freire, 2008), os aprendizados e desafios vivenciados ao longo das disciplinas ministradas dividiram-se em aspectos humanos e técnicos, conforme apresentado na Tabela 1. No tocante aos aspectos humanos, oito discentes de um total de dez consideraram que “trabalhar em equipe/interagir”, assim como proatividade, visão de projetos e desenvolver a criatividade, reforçam a importância de práticas pedagógicas pautadas em momentos colaborativos voltados à formação humana. Sobre os conhecimentos técnicos, as etapas de instalação e configuração das aplicações de desenvolvimento, engenharia de requisitos, codificação e documentação foram os momentos de maiores saberes adquiridos.

Aspectos considerados no processo	Aprendizados adquiridos	Dificuldades encontradas
Aspectos humanos		
Organização	03	02
Iniciativa/proatividade	04	04
Pontualidade (capacidade de cumprir prazos)	01	04
Compromisso/Comprometimento	03	03
Gestão de tempo (família, trabalho, universidade etc)	02	07
Visão de projetos (etapas e prazos)	04	04
Trabalhar em equipe/interagir	08	04
Desenvolver a criatividade (encontrar soluções)	04	05
Resiliência (adaptação a mudanças)	02	03
Aspectos técnicos		
Instalação e configuração das aplicações de desenvolvimento	03	05
Etapa de engenharia de requisitos	03	06
Etapa de prototipação	02	06
Etapa de modelagem de dados	01	06
Etapa de codificação	03	06
Etapa de hospedagem da aplicação	01	06
Etapa de documentação	03	06

Tabela 1. Aprendizados e dificuldades vivenciados pelos estudantes, nos aspectos humano e técnico. Fonte: Autores (2024).

Dentre as maiores dificuldades elencadas pelos estudantes no desenvolvimento do projeto relatado, em relação aos aspectos humanos, destacam-se a gestão do tempo e o desenvolvimento da criatividade. Quanto às questões técnicas, os estudantes apontaram dificuldades em todas as etapas, as quais, de certa forma, foram superadas, quando consideradas as aprendizagens obtidas durante todo o processo, além dos produtos gerados pelos estudantes. A infraestrutura e conectividade da universidade foram avaliadas como satisfatórias para a execução das tarefas.

O conjunto de dados obtidos demonstra que, no geral, os participantes tiveram uma experiência satisfatória com o projeto, destacando a importância das orientações dos professores, a relevância dos exemplos práticos e a aplicabilidade do *software* desenvolvido para uma empresa local. No entanto, existem pontos a serem aprimorados, como o refinamento das orientações sobre a liderança dos grupos. A dificuldade na

modelagem de dados e o envolvimento com a comunidade local também surgiram como pontos que exigem uma abordagem mais detalhada.

6. Considerações finais

Este trabalho contém o relato de práticas pedagógicas realizadas nas disciplinas “Construção de Software” e “Prática Interdisciplinar de Aplicações em Sistemas de Informação”, com destaque para um instrumento de elaboração própria, identificado como trilha de desenvolvimento de *software*, para aplicação em cursos de Computação. Os dados apurados sobre as percepções dos alunos matriculados nas disciplinas em relação às práticas pedagógicas utilizadas pelos professores revelam um alinhamento com o conceito de inovação apresentado por Gatti *et al.* (2019), uma vez que, embora as práticas pedagógicas aplicadas não fossem inéditas, elas promoveram mudanças que aprimoraram a formação dos estudantes.

Algumas reflexões emergem do conteúdo relatado: a) os professores de cursos de Computação estão preparados para desenvolver práticas pedagógicas com TDIC que promovam o desenvolvimento de *software* a partir de uma abordagem extensionista? b) Quanto à formação de professores de Computação, inicial e continuada, considerando os diversos níveis de ensino (técnico, superior e pós-graduação) e áreas diversas de formação em computação, ela contempla a importância do controle de evasão e técnicas pedagógicas diversificadas para o exercício do magistério em sala de aula e fora dela? Diversas outras questões importantes e relacionadas ao trabalho desse professor poderiam ser trazidas para discussão neste estudo.

Por fim, ressalvadas as suas limitações, este relato ateve-se ao objetivo de compartilhar práticas pedagógicas de docentes em Computação, voltadas ao desenvolvimento de *software* a partir de diversas TDIC e aplicação de uma trilha parametrizada em etapas e atividades técnicas situadas na interface entre ensino e extensão. Os resultados permitem concluir que a articulação entre teoria (conhecimento científico) e prática, com foco nas particularidades do campo de atuação profissional, contribui para a promoção de uma aprendizagem que favorece o engajamento e diminui o risco de evasão.

Referências

- ABICH, D.; CAVALHEIRO, B.; PARIZI, R.. PIDS: Projetos Integradores em Desenvolvimento de Software como Prática Colaborativa na Curricularização da Extensão. In: Simpósio Brasileiro de Sistemas Colaborativos (SBSC), Salvador/BA. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2024. p. 189-192.
- BELL, D. *O advento da sociedade pós-industrial*: uma tentativa de previsão social. São Paulo: Cultrix, 1973.
- CASTELLS, M. *A sociedade em rede*. MAJER, R. V. (trad.). São Paulo: Paz e Terra, 1999.
- BRASIL. Resolução CNE/CES nº 7, de 18 de dezembro de 2018. Dispõe sobre a extensão na educação superior brasileira e regulamenta a carga horária mínima para

- as atividades de extensão nos cursos de graduação. Diário Oficial da União: seção 1, Brasília, DF, 19 dez. 2018.
- FERREIRA, J. M. C.. *Novas tecnologias e organização do trabalho*. Organizações & Sociedade, v. 7, n. 19, p. 91–108, set. 2000.
- FRANCO, M. E.; FRANCO, P. F.. Curricularização da Extensão: Relato de Experiência no Curso de Sistemas de Informação do IFSULDEMINAS. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 2023, João Pessoa. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2023. p. 1-8.
- FREIRE, Paulo. *Educação como prática da liberdade*. 31. ed. Rio de Janeiro: Paz e Terra, 2008.
- GATTI, B. A. et al.. *Professores do Brasil*: novos cenários de formação. Brasília: Unesco, 2019.
- GIL, A. C.. *Métodos e Técnicas de Pesquisa Social*. 6 ed. São Paulo: Atlas, 2008.
- MASUDA, Y. *The information society as post-industrial society*. Tokyo: Institute for the Information Society, 1980.
- MELO, A. M.; WERZ, M. C. G. Informática na educação e práticas extensionistas: interação universidade-escola em perspectiva. In: VALENTE, J. A.; FREIRE, F. M. P.; ARANTES, F. L. *Tecnologia e educação*: passado presente e o que está por vir. Campinas: Unicamp/NIED, 2018. p. 65-98.
- MINAYO, M. C. de S.. *O Desafio do Conhecimento*: Pesquisa Qualitativa em Saúde. 14 ed. São Paulo: Hucitec, 2014.
- PINTO, S. C. C. da S.; NASCIMENTO, G. S. R. do. O pensamento computacional e a nova sociedade. In: VALENTE, J. A.; FREIRE, F. M. P.; ARANTES, F. L.. *Tecnologia e educação*: passado presente e o que está por vir. Campinas: Unicamp/NIED, 2018. p. 302-322.
- PRESSMAN, R. S.. *Engenharia de software*: uma abordagem profissional. MAXIM, B. R. (trad.). 9 ed. Porto Alegre: AMGH, 2021.
- SOMMERVILLE, I.. *Engenharia de software*. BOSNIC, I.; GONÇALVES, K. G. O. (trad.). 9 ed. São Paulo: Pearson, 2011.
- TIVES, H. A.; MARINI, A.; CANEDO, E. D.. Curricularização da Extensão na Prática: Relato de uma Experiência em Computação. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE), Rio de Janeiro/RJ. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2024. p. 2192-2205.
- VALENTE, J. A. Inovação nos processos de ensino e de aprendizagem: o papel das tecnologias digitais. In: VALENTE, J. A., FREIRE, F. M. P., ARANTES, F. L. *Tecnologia e educação*: passado presente e o que está por vir. Campinas: Unicamp/NIED, 2018. p. 17-41.