

Desenvolvimento de software no curso de análise e desenvolvimento de sistemas: uma experiência interdisciplinar com extensão

Software development in the systems analysis and development program: an interdisciplinary extension experience

Viviane de Fatima Bartholo

Faculdade de Tecnologia de Ourinhos (Fatec Ourinhos),
Ourinhos, São Paulo viviane.bartholo@cps.sp.gov.br

Carolinne Roque e Faria

Faculdade de Tecnologia de Ourinhos (Fatec Ourinhos),
Ourinhos, São Paulo carolinne.faria@cps.sp.gov.br

Apresentação do estudo

A formação de profissionais na área de Tecnologia da Informação (TI) demanda metodologias que aproximem o estudante de situações reais do mundo do trabalho. Nesse contexto, é fundamental desenvolver competências voltadas à análise de problemas e à construção de soluções de software.

Entre as competências desenvolvidas destacam-se análise de requisitos, modelagem de sistemas, desenvolvimento e manutenção de software, trabalho em equipe, comunicação técnica, resolução de problemas complexos, gerenciamento de projetos e tomada de decisão. Essas competências são essenciais na formação do tecnólogo em Análise e Desenvolvimento de Sistemas (ADS), uma vez que esse profissional atua na identificação de demandas organizacionais, na análise de processos e na concepção de soluções computacionais capazes de atender às necessidades de diferentes contextos. Além dos conhecimentos técnicos, essas habilidades contribuem para a atuação colaborativa em equipes multidisciplinares, favorecendo a entrega de soluções inovadoras, eficientes e alinhadas aos objetivos das organizações.

No contexto de ADS, metodologias ativas tornam-se especialmente relevantes devido à natureza prática e dinâmica da área de TI, na qual os profissionais precisam atuar em equipes multidisciplinares, resolver problemas reais e adaptar-se continuamente às mudanças tecnológicas. A aprendizagem baseada em projetos favorece esse processo ao organizar a formação em torno de problemas autênticos, investigação, produção colabo-

rativa e socialização de resultados (THOMAS, 2000; BENDER, 2014). De modo complementar, o uso de metodologias ativas contribui para deslocar o estudante de uma postura predominantemente receptiva para uma atuação mais participativa e autoral no processo de aprendizagem (BACICH; MORAN, 2018).

Conforme Ian Sommerville (2019), a formação em Engenharia de Software deve incorporar práticas que simulem ambientes reais, como levantamento de requisitos com clientes, reuniões de *sprint*, versionamento em GitHub, divisão de papéis na equipe, prototipação, validação com usuários, documentação técnica e apresentação de *pitch*, permitindo aos estudantes vivenciar as etapas do desenvolvimento e aprimorar habilidades como trabalho em equipe, comunicação e resolução de problemas, tais como automação de processos, sistemas web, aplicativos, marketplaces, sistemas gerenciais e plataformas de serviços.

No curso de Análise e Desenvolvimento de Sistemas da Fatec Ourinhos, essa abordagem é implementada por meio de um projeto interdisciplinar que integra diferentes disciplinas ao longo da formação acadêmica. Desenvolvida desde 2022, a iniciativa já envolveu aproximadamente 240 alunos por semestre, promovendo a articulação entre conhecimentos teóricos e práticos no desenvolvimento de soluções de software.

Este trabalho tem como objetivo analisar a contribuição dessa prática pedagógica na formação dos estudantes, considerando tanto o desenvolvimento técnico quanto a percepção dos alunos sobre o processo de aprendizagem. Sua relevância está relacionada à crescente necessidade de estratégias educacionais que promovam a integração entre ensino, prática profissional e extensão, aproximando os estudantes de desafios concretos encontrados na sociedade e nas organizações. Nesse contexto, o projeto interdisciplinar configura-se como um espaço de experimentação e construção do conhecimento, permitindo aos alunos desenvolver uma visão sistêmica dos processos de desenvolvimento de software e compreender a aplicação dos conhecimentos adquiridos ao longo do curso em situações reais.

Descrição do estudo

O estudo caracteriza-se como uma pesquisa aplicada, de abordagem qualitativa e natureza descritiva, estruturada como relato de experiência. A metodologia adotada fundamenta-se na PBL, que, conforme Thomas (2000), promove a aprendizagem por meio da investigação e da resolução de problemas reais. Nesse contexto, os estudantes desenvolvem, em grupos, sistemas de software voltados para demandas identificadas em organizações, empreendedores ou na comunidade, integrando conhecimentos adquiridos em diferentes disciplinas ao longo do curso.

Durante o desenvolvimento dos projetos, os estudantes realizam o levantamento de requisitos por meio de entrevistas, questionários e reuniões com os clientes ou usuários, identificando necessidades e funcionalidades do sistema. Na etapa de análise e modelagem, elaboram diagramas UML, casos de uso, diagramas de classes e fluxos de processos para representar a estrutura e o comportamento da solução proposta. Em seguida, desenvolvem

protótipos de interfaces utilizando ferramentas de design, permitindo a validação prévia da experiência do usuário e dos requisitos funcionais.

Na fase de projeto e implementação, os alunos realizam a modelagem do banco de dados, definem a arquitetura da aplicação e desenvolvem sistemas web, aplicativos móveis, plataformas de serviços, marketplaces e sistemas gerenciais utilizando linguagens e frameworks estudados no curso. Posteriormente, executam testes funcionais e de usabilidade, identificando falhas e propondo melhorias. Ao final de cada ciclo, os grupos apresentam os resultados por meio de relatórios técnicos, demonstrações do software e apresentações para docentes, colegas e parceiros externos, possibilitando a avaliação tanto dos aspectos técnicos quanto das competências de comunicação e trabalho em equipe.

Essa abordagem está alinhada ao que propõe Pressman (2016), ao destacar que o desenvolvimento de software envolve um conjunto estruturado de atividades que exigem competências técnicas e de gestão, para além da programação. Essas atividades abrangem a comunicação com clientes para levantamento de requisitos, a modelagem de processos e sistemas, a elaboração de protótipos, a implementação e os testes de software, bem como o planejamento das atividades, a divisão de tarefas e o acompanhamento das entregas. No projeto interdisciplinar, os estudantes desenvolvem essas competências ao longo da construção de sistemas voltados para problemas reais, vivenciando diferentes etapas do ciclo de desenvolvimento de software.

A coleta de dados foi realizada por meio da observação das atividades desenvolvidas durante o projeto interdisciplinar, da análise dos artefatos produzidos pelos estudantes, tais como documentações, diagramas e protótipos, e da aplicação de um questionário eletrônico elaborado no *Google Forms*. O questionário buscou avaliar a percepção dos participantes quanto ao processo de aprendizagem, ao engajamento nas atividades propostas, ao desenvolvimento da autonomia e à contribuição da experiência para sua formação profissional. O instrumento aplicado é apresentado integralmente no Apêndice A.

O desenvolvimento do projeto ocorre de forma interdisciplinar, envolvendo disciplinas como Engenharia de Software I, II e III, Interação Humano-Computador (IHC), Banco de Dados e Laboratório de Engenharia de Software (Tabela 1).

Tabela 1. Detalhamento das disciplinas

Disciplina	Semestre	Carga horária
Engenharia de Software I	2º	80h
Engenharia de Software II	3º	80h
Engenharia de Software III	4º	80h
Interação Humano-Computador	3º	40h
Banco de Dados	3º	80h
Laboratório de Engenharia de Software	4º	80h

Fonte: elaborado pelos autores (2026)

Os estudantes são organizados em grupos de até cinco integrantes no 3º semestre, totalizando, em média, 15 novos grupos por semestre. Essa organização possibilita a diversidade de propostas e a simulação de dinâmicas colaborativas próximas às vivenciadas no ambiente profissional. Cada etapa do projeto está associada a competências específicas, permitindo a construção progressiva do conhecimento. A Tabela 2 apresenta as etapas de desenvolvimento do projeto e sua relação com as disciplinas envolvidas.

Tabela 2. Etapas de desenvolvimento dos projetos

Etapas	Descrição	Disciplinas	Competências desenvolvidas
Ideação	Definição do problema	Eng. de Software I	Visão de solução
Viabilidade/ Levantamento de Requisitos	Análise técnica e de mercado	Eng. de Software II	Análise de sistemas
Modelagem	Diagramas UML	Eng. de Software III	Projeto de software
Interface	Prototipação	IHC	Usabilidade
Banco de dados	Modelagem de dados	Banco de Dados	Estruturação
Implementação	Desenvolvimento	Lab. Eng. Software	Integração

Fonte: elaborado pelos autores (2026)

Como resultados parciais, observou-se uma evolução na qualidade das soluções desenvolvidas pelos estudantes, bem como avanços na capacidade de resolução de problemas, no trabalho colaborativo e na articulação entre conhecimentos teóricos e práticos. Também foram identificados indícios de maior engajamento, autonomia e protagonismo dos alunos ao longo do processo de desenvolvimento dos sistemas. Como forma de ilustrar a diversidade das soluções produzidas, destacam-se iniciativas voltadas a diferentes setores da sociedade, incluindo plataformas para agendamento de serviços, sistemas de apoio ao agronegócio e aplicações relacionadas à causa animal.

Os projetos apresentados na Tabela 3, abaixo, apresentam exemplos de projetos, atualmente em desenvolvimento na disciplina de Engenharia de Software II e demonstram o potencial da abordagem adotada para aproximar a formação acadêmica de desafios reais, permitindo que os estudantes desenvolvam soluções tecnológicas aplicáveis a diferentes contextos sociais, econômicos e organizacionais. Dessa forma, a experiência contribui para a formação de profissionais mais preparados para atuar na análise e no desenvolvimento de sistemas voltados às necessidades da comunidade e do mercado de trabalho.

Tabela 3. Exemplos de projetos desenvolvidos na disciplina de Engenharia de Software II

Nome do projeto	Objetivo	Etapa
DentalFlow	Desenvolver um ecossistema digital voltado à gestão de consultórios odontológicos, capaz de otimizar processos administrativos, facilitar o gerenciamento de atendimentos e aprimorar a comunicação entre profissionais e pacientes, oferecendo uma solução prática, organizada e acessível.	Viabilidade/ Levantamento de Requisitos
Sistema de gerenciamento para clínica veterinária – API <i>restful</i>	Desenvolver uma plataforma digital para gerenciamento de clínicas veterinárias independentes, capaz de centralizar informações, otimizar processos administrativos e operacionais e proporcionar maior organização no atendimento clínico.	Viabilidade/ Levantamento de Requisitos
Desenvolvimento de Um Sistema Web - PetDoa	Fornecer uma plataforma web capaz de registrar, documentar e validar as transações relacionadas à compra, venda e adoção de animais, garantindo rastreabilidade das informações, como identificação dos envolvidos e dados dos pets associados.	Viabilidade/ Levantamento de Requisitos
Festify	Desenvolver uma plataforma digital que facilite o planejamento e a organização de festas e eventos, centralizando serviços, produtos e fornecedores em um ambiente intuitivo, seguro e funcional.	Viabilidade/ Levantamento de Requisitos
MOVVA	Disponibilizar um ambiente digital capaz de centralizar e organizar o acompanhamento contínuo entre profissionais da saúde e pacientes.	Viabilidade/ Levantamento de Requisitos
Não Desperdice Ourinhos	A modelagem, especificação e desenvolvimento de um aplicativo capaz de conectar consumidores a estabelecimentos comerciais que possuam alimentos excedentes ou próximos do vencimento, permitindo sua comercialização com preços reduzidos.	Viabilidade/ Levantamento de Requisitos
PertMania	Disponibilizar uma plataforma digital que permita a consulta e visualização de pet shops, centralizando informações relevantes sobre os serviços oferecidos.	Viabilidade/ Levantamento de Requisitos
Autolink	Desenvolver uma plataforma integrada de gestão para oficinas mecânicas, associada a um módulo de assistência automotiva emergencial, utilizando geolocalização em tempo real para otimizar a comunicação entre oficinas e motoristas, reduzir o tempo de resposta e melhorar a eficiência operacional.	Viabilidade/ Levantamento de Requisitos
Vintage Hub	Desenvolver um sistema web para gestão de brechós, permitindo o controle centralizado de estoque, vendas e consignação, com geração de relatórios gerenciais para apoio à tomada de decisão.	Viabilidade/ Levantamento de Requisitos
ArtiFy	Oferecer uma plataforma digital que conecte artistas independentes a clientes em potencial, facilitando a divulgação de trabalhos artísticos e a busca por serviços de forma prática e eficiente.	Viabilidade/ Levantamento de Requisitos
Glow App	Oferecer uma plataforma digital que conecte profissionais da área da beleza a clientes interessados em seus serviços, facilitando o agendamento, a comunicação e a contratação.	Viabilidade/ Levantamento de Requisitos

Nome do projeto	Objetivo	Etapa
Moto Util	Fornecer uma plataforma digital eficiente para locação de motocicletas, permitindo o gerenciamento completo das operações de cadastro, reserva, pagamento, acompanhamento e devolução.	Viabilidade/ Levantamento de Requisitos
PetHub	Disponibilizar uma plataforma digital capaz de integrar tutores de animais de estimação e estabelecimentos do segmento pet, centralizando informações, serviços e funcionalidades relacionadas ao cuidado, bem-estar e acompanhamento dos pets.	Viabilidade/ Levantamento de Requisitos
SOS Patinha	Disponibilizar uma plataforma digital para apoiar ações relacionadas à proteção animal, promovendo maior organização, comunicação e eficiência no gerenciamento das atividades da ONG.	Viabilidade/ Levantamento de Requisitos

Fonte: elaborado pelos autores (2026)

Impactos locais obtidos

A análise dos impactos foi realizada por meio de questionário aplicado a 198 estudantes, utilizando escala Likert de 1 a 5 (1 = discordo totalmente e 5 = concordo totalmente), instrumento amplamente utilizado em pesquisas para mensurar atitudes e percepções dos respondentes (LIKERT, 1932; MALHOTRA, 2012). Os resultados evidenciam uma percepção predominantemente positiva em relação à metodologia adotada.

Em relação à compreensão dos conteúdos de Engenharia de Software, 80% dos estudantes concordaram ou concordaram totalmente que o projeto contribuiu para o aprendizado (42,5% no nível 4 e 37,5% no nível 5), enquanto 20% mantiveram-se neutros. Esses resultados sugerem que os participantes percebem o projeto como uma estratégia relevante para a consolidação dos conceitos trabalhados nas disciplinas. Ao mesmo tempo, a parcela de respostas neutras indica que a percepção não foi uniforme entre todos os estudantes, o que pode estar relacionado a diferenças de repertório técnico prévio, participação nas equipes, maturidade acadêmica ou familiaridade com práticas de desenvolvimento de software.

Quanto à aplicação prática do conhecimento, 80% dos estudantes afirmaram que conseguiram aplicar o que foi aprendido nas disciplinas (47,5% no nível 4 e 32,5% no nível 5). Esse resultado reforça a adequação da aprendizagem baseada em projetos ao ensino de Engenharia de Software, pois a metodologia exige que conceitos abstratos — como requisitos, modelagem, arquitetura, banco de dados, interface e testes — sejam mobilizados em produtos concretos. Essa articulação entre teoria e prática é coerente com a literatura sobre metodologias ativas, segundo a qual a aprendizagem tende a se tornar mais significativa quando o estudante participa da resolução de problemas contextualizados e reconhece a utilidade social e profissional do conhecimento construído (THOMAS, 2000; BACICH; MORAN, 2018).

No que se refere ao engajamento, 65% dos estudantes concordaram ou concordaram totalmente que a liberdade de escolha das tecnologias aumentou seu interesse (30% no nível 4 e 35% no nível 5), enquanto 20% apresentaram posição neutra e 12,5% indicaram discordância. Embora a maioria tenha avaliado positivamente a autonomia proporcionada pela metodologia, os resultados também sugerem que a liberdade de escolha pode representar desafios para parte dos estudantes.

Entre as possíveis explicações estão a insegurança diante da necessidade de tomar decisões técnicas, dificuldades na seleção de tecnologias adequadas, assimetria de conhecimentos entre integrantes do grupo e diferentes níveis de experiência prévia. Esse achado evidencia a importância de equilibrar autonomia discente e acompanhamento docente, especialmente por meio de orientações periódicas, critérios de avaliação explícitos, momentos de feedback e acompanhamento do planejamento das entregas.

A análise dos resultados permite compreender que os efeitos positivos da experiência não decorrem apenas da realização de um projeto final, mas do percurso formativo que articula sucessivas etapas de investigação, modelagem, implementação, validação e apresentação. Assim, o projeto interdisciplinar contribui para o desenvolvimento de competências técnicas e socioemocionais ao criar situações em que os estudantes precisam negociar soluções, justificar escolhas, documentar decisões, lidar com limitações de tempo e responder a demandas de usuários ou parceiros. Essa dinâmica aproxima a experiência educacional das práticas profissionais descritas na literatura de Engenharia de Software (PRESSMAN, 2016; SOMMERVILLE, 2019).

A dimensão extensionista da experiência ocorreu, no período analisado, principalmente pela identificação de problemas reais em diálogo com usuários, empreendedores, organizações locais e iniciativas comunitárias, bem como pela proposição de soluções orientadas a necessidades externas ao ambiente estritamente acadêmico. Projetos como Não Desperdice Ourinhos, SOS Patinha, sistemas para clínicas veterinárias, plataformas de serviços e soluções para pequenos negócios demonstram aproximação com demandas sociais e produtivas do município e da região. Ainda que nem todas as soluções tenham sido implantadas em ambiente real, a participação de públicos externos no levantamento de necessidades, na validação de propostas e na avaliação de protótipos constituiu uma ação extensionista inicial, permitindo circulação de conhecimento entre a Fatec Ourinhos e a comunidade.

Os impactos observados junto aos públicos externos foram de natureza predominantemente formativa e prospectiva: sistematização de demandas, elaboração de diagnósticos, construção de protótipos, apresentação de alternativas tecnológicas e ampliação do diálogo entre curso, comunidade e setor produtivo. Dessa forma, a extensão aparece como componente em consolidação, já presente nas etapas de identificação de problemas e validação inicial, mas ainda com necessidade de maior institucionalização, acompanhamento pós-entrega e mensuração dos efeitos sociais das soluções desenvolvidas.

Considerações finais e recomendações

A experiência interdisciplinar desenvolvida no curso de Análise e Desenvolvimento de Sistemas da Fatec Ourinhos demonstra potencial para integrar conteúdos curriculares, práticas profissionais e demandas reais, favorecendo uma formação mais contextualizada para a área de TI. Os dados obtidos indicam percepção positiva dos estudantes quanto à aprendizagem, à aplicação prática dos conhecimentos e ao desenvolvimento da autonomia, embora também revelem a necessidade de acompanhamento docente sistemático para apoiar alunos que apresentam maior dificuldade diante da liberdade tecnológica e da gestão colaborativa dos projetos.

Como contribuição pedagógica, a prática fortalece a articulação entre disciplinas e permite que competências de Engenharia de Software sejam trabalhadas de forma progressiva, desde a ideação e o levantamento de requisitos até a modelagem, prototipação, implementação e apresentação das soluções. Essa organização favorece a aprendizagem significativa, pois os estudantes aplicam conceitos em situações próximas às que encontrarão no mundo do trabalho.

No que se refere à extensão, a experiência já contempla ações de aproximação com públicos externos, especialmente por meio da identificação de demandas reais, da interação com potenciais usuários e da elaboração de soluções voltadas à comunidade, a empreendedores e a organizações locais. Recomenda-se, entretanto, ampliar a sistematização dessa dimensão nas próximas edições, registrando quais parceiros participaram de cada projeto, quais etapas envolveram interação externa, quais protótipos foram validados pelos usuários e quais impactos foram percebidos após a apresentação ou eventual implantação das soluções.

Como recomendações para continuidade, sugere-se: fortalecer parcerias com empresas, órgãos públicos, organizações sociais e empreendedores da região; realizar mostras tecnológicas e apresentações públicas dos projetos; estabelecer instrumentos de acompanhamento dos impactos externos; comparar resultados entre diferentes turmas e semestres; e aprofundar a análise qualitativa das percepções discentes por meio de questões abertas, entrevistas ou grupos focais. Espera-se que esse acompanhamento contribua para uma avaliação mais abrangente da metodologia adotada e para o aperfeiçoamento contínuo da experiência pedagógica no curso de ADS.

Referências

BACICH, Lilian; MORAN, José. **Metodologias ativas para uma educação inovadora**: uma abordagem teórico-prática. Porto Alegre: Penso, 2018.

BENDER, William N. **Aprendizagem baseada em projetos**: educação diferenciada para o século XXI. Porto Alegre: Penso, 2014.

LIKERT, Rensis. **A technique for the measurement of attitudes**. *New York: Archives of Psychology*, 1932. Disponível em: <https://psycnet.apa.org/record/1933-01885-001>. Acesso em: 30 mar. 2026.

MALHOTRA, Naresh K. **Pesquisa de marketing: uma orientação aplicada**. 6. ed. Porto Alegre: Bookman, 2012.

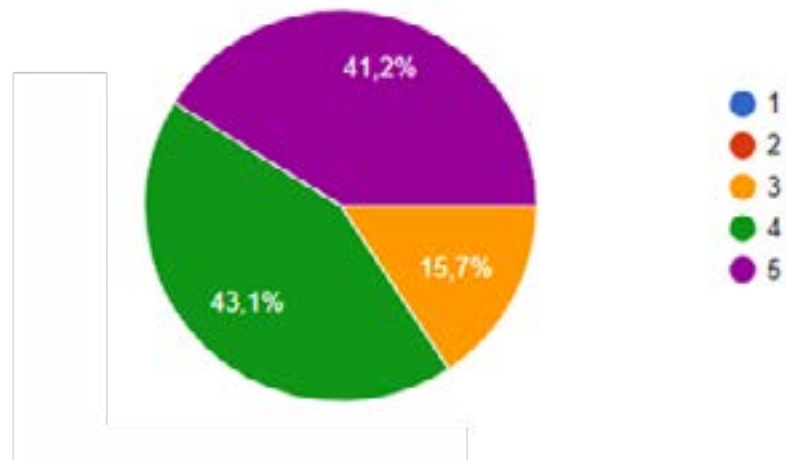
PRESSMAN, Roger S. **Engenharia de software: uma abordagem profissional**. 8. ed. Porto Alegre: McGraw-Hill, 2016

SOMMERVILLE, Ian. **Engenharia de software**. 10. ed. São Paulo: Pearson, 2019.

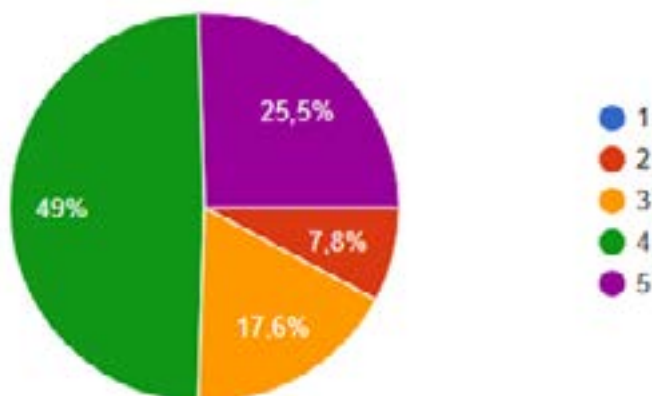
THOMAS, John W. **A review of research on project-based learning**. San Rafael: Autodesk Foundation, 2000.

Apêndice A

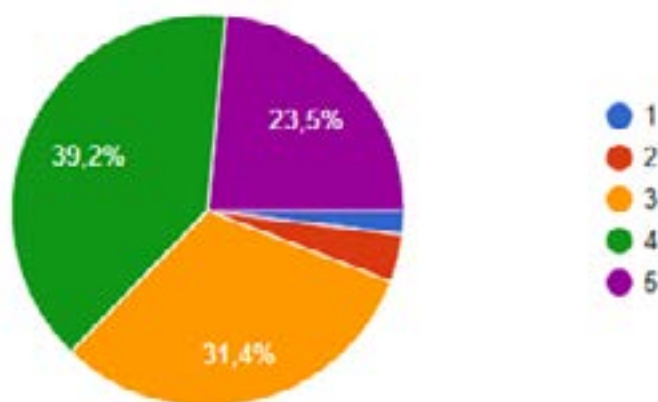
O projeto ajudou a entender melhor os conceitos de Engenharia de Software



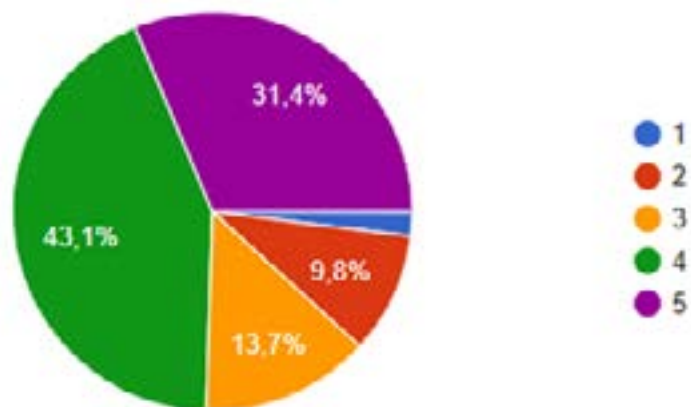
Consegui aplicar na prática o que aprendi nas disciplinas



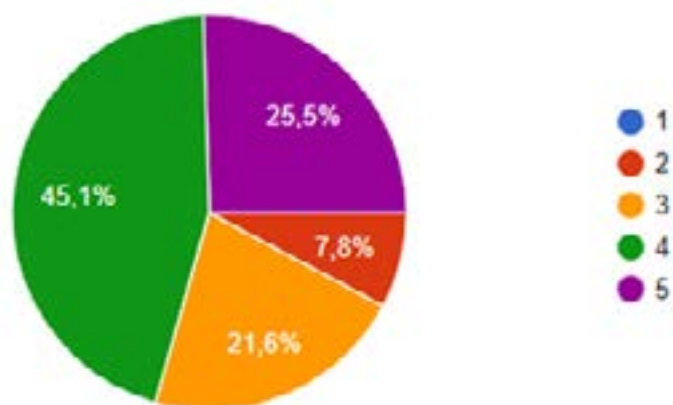
Desenvolvi melhor minha capacidade de resolver problemas



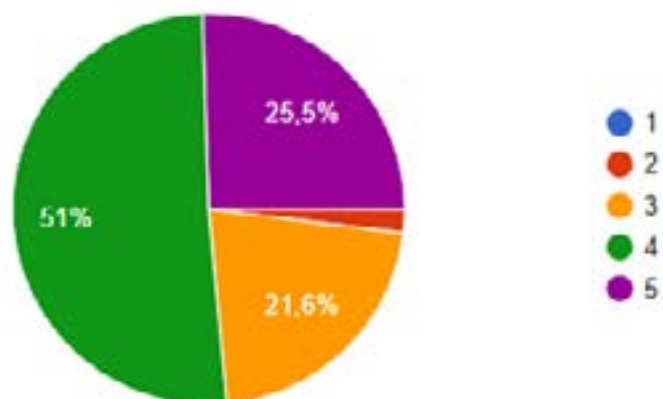
Apreendi a estruturar melhor um sistema de software



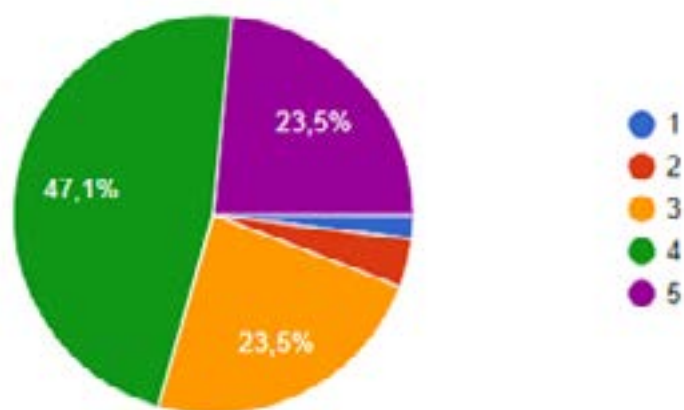
O projeto melhorou minha capacidade de trabalhar em equipe



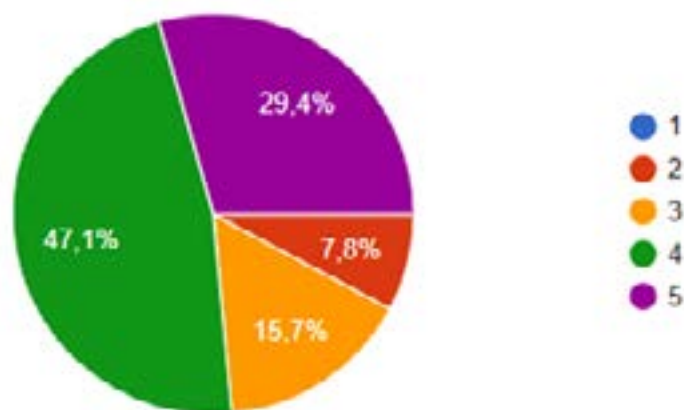
Consegui me comunicar melhor com meu grupo



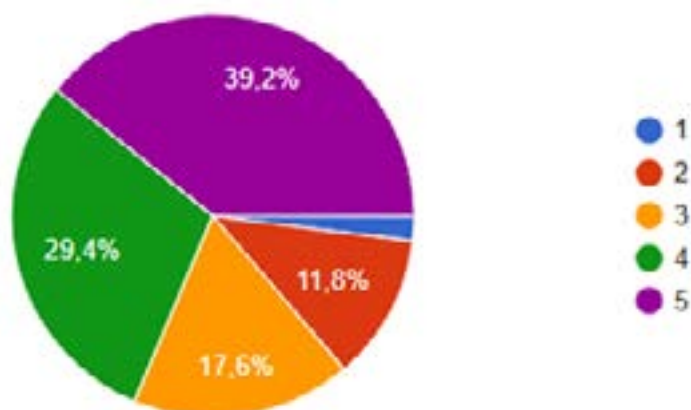
Apreendi a dividir tarefas e organizar atividades



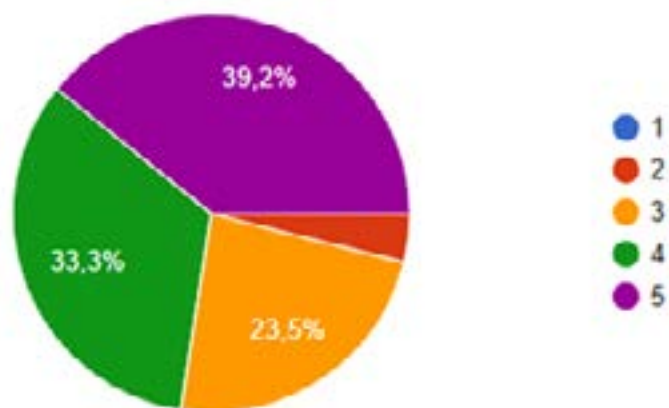
Desenvolvi mais autonomia durante o projeto



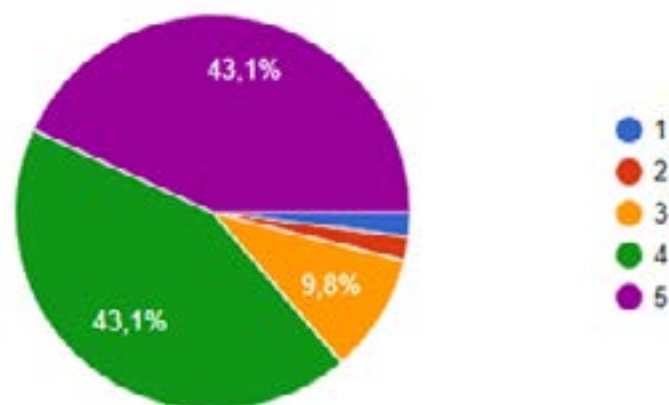
A liberdade de escolher tecnologias aumentou meu interesse



Busquei aprender novas ferramentas por conta própria



O projeto me fez perceber como a tecnologia pode resolver problemas reais



Acredito que o sistema desenvolvido poderia ser usado na prática

