

16º Congresso de Inovação, Ciência e Tecnologia do IFSP - 2025

O DESENVOLVIMENTO DE UMA PLATAFORMA DE SUPORTE AO ENSINO DE ENGENHARIA DE SOFTWARE

JOÃO P. VERÍSSIMO¹, RAUL E. S. CASTANHARI²

¹Graduando em Tecnologia de Análise e Desenvolvimento de Sistemas, Bolsista PIBIFSP, IFSP, Câmpus Jacareí, joaopedrov0@gmail.com

²Professor EBTB em Ciências de Computação, IFSP, Câmpus Jacareí, raul.castanhari@ifsp.edu.br

Área de conhecimento (Tabela CNPq): 1.03.03.04-9 Sistemas de Informação.

RESUMO: Este artigo apresenta o desenvolvimento do software "The Architect", um sistema web desenvolvido com o intuito de apoiar graduandos em cursos de computação durante o aprendizado das práticas da engenharia de software, por tratar-se de uma disciplina extremamente importante, mas frequentemente mal compreendida devido à sua natureza abstrata. A solução proposta tem como objetivo ser uma ferramenta de ensino que possibilita a criação de artefatos de software com apoio de materiais didáticos integrados. No estado atual do projeto já é possível gerenciar os artefatos de diferentes projetos e ter acesso a materiais didáticos, porém, avaliações com o público-alvo evidenciaram a necessidade de reforçar a integração do material didático com o desenvolvimento dos projetos, a fim de tornar o processo de criação mais intuitivo.

PALAVRAS-CHAVE: sistema web; engenharia de software; educativo.

THE DEVELOPMENT OF A SUPPORT PLATFORM FOR SOFTWARE ENGINEERING TEACHING

ABSTRACT: This article presents the development of "The Architect", a web-based system designed to support undergraduate computer science students in learning software engineering practices. This is an extremely important discipline, yet one often misunderstood due to its abstract nature. The proposed solution aims to be a teaching tool that enables the creation of software artifacts supported by integrated learning materials. The project's current state allows for managing artifacts from different projects and accessing learning materials. However, evaluations with the target audience highlighted the need to strengthen the integration of learning materials with project development to make the creation process more intuitive.

KEYWORDS: web system; software engineering; educational.

INTRODUÇÃO

A engenharia de software preocupa-se com tudo que envolve o processo de desenvolver sistemas, desde a análise e construção até a validação e manutenção do sistema (SOMMERVILLE, 2011). O

conhecimento das práticas da engenharia de software é a ferramenta que possibilitou a construção de sistemas cada vez mais complexos de forma consistente e madura, possibilitando a presença frequente de soluções computacionais sofisticadas no cotidiano. Entretanto, a importância dos estudos acerca das práticas e metodologias da engenharia de software se deve justamente ao fato de suportarem consistentemente o desenvolvimento de software complexo, que naturalmente seria de difícil compreensão não só pelos clientes, como também pelo próprio time de desenvolvimento. Por esse motivo, a disciplina acaba não sendo bem compreendida por estudantes que já não tenham uma experiência prévia com o desenvolvimento de software (PARRACHO; ZACARIAS; SANTOS, 2025), em especial a construção de sistemas que se tornaram complexos demais para serem compreendidos com precisão sem a necessidade de uma documentação robusta. Santos et al. mapearam ferramentas criadas no contexto do ensino da engenharia de software, constatando o vasto uso de jogos e simuladores para o ensino da disciplina em trabalhos como o SPARSE (SOUZA et al., 2010) que buscava trazer o ensino mais próximo da prática. Nesse contexto, o sistema "The Architect" tem como objetivo principal ser uma ferramenta que suporte o gerenciamento dos artefatos de software com uma abordagem educativa, com guias integrados aos campos de criação desses artefatos para fornecer apoio ao estudante durante o processo. Como uma ferramenta de ensino, o sistema também prevê a importação e exportação desses projetos de forma a permitir o seu uso como meio de criação das atividades da disciplina.

MATERIAIS E MÉTODOS

O sistema foi desenvolvido com as práticas do desenvolvimento ágil de software (BECK et al., 2001), buscando retornos frequentes das partes interessadas. O planejamento do projeto foi feito inicialmente através de reuniões com o docente que lecionava engenharia de software na instituição para entender as maiores dificuldades observadas, bem como discutir possíveis abordagens para solucionar esses problemas. Com os objetivos iniciais pré-estabelecidos iniciou-se a criação dos requisitos e outros artefatos de software, frequentemente apresentados para validação do docente para garantir o alinhamento dos interesses com a solução proposta pelo projeto.

A plataforma foi desenvolvida com base no mapa de navegação simplificado da figura 1, onde o usuário é recepcionado com uma breve explicação sobre o sistema e já são apresentadas suas funcionalidades. A página de materiais didáticos contém uma coleção de conteúdos para apoio ao estudo dos assuntos abordados na disciplina, enquanto a página de projetos direciona o usuário a uma interface para criação ou importação de projetos que serão visualizados ou editados no sistema.

A aplicação foi desenvolvida com HTML, CSS, JavaScript e Bootstrap com foco no desenvolvimento modular e separação clara de responsabilidades, favorecendo a manutenibilidade (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2011). Os dados gerados na aplicação são mantidos apenas no navegador, portanto, não há tráfego desses materiais via rede nem a necessidade de gerenciar um banco de dados no estado atual. O fluxo de uso da ferramenta consiste em criar ou importar um projeto, fazer a leitura ou edições de interesse e, por fim, exportar o arquivo resultante em formato *JSON* criado pelo sistema ou, futuramente, em tabelas.

Os materiais didáticos e guias presentes na plataforma foram escritos com base no material das aulas e livros sobre engenharia de software (SOMMERVILLE, 2011).

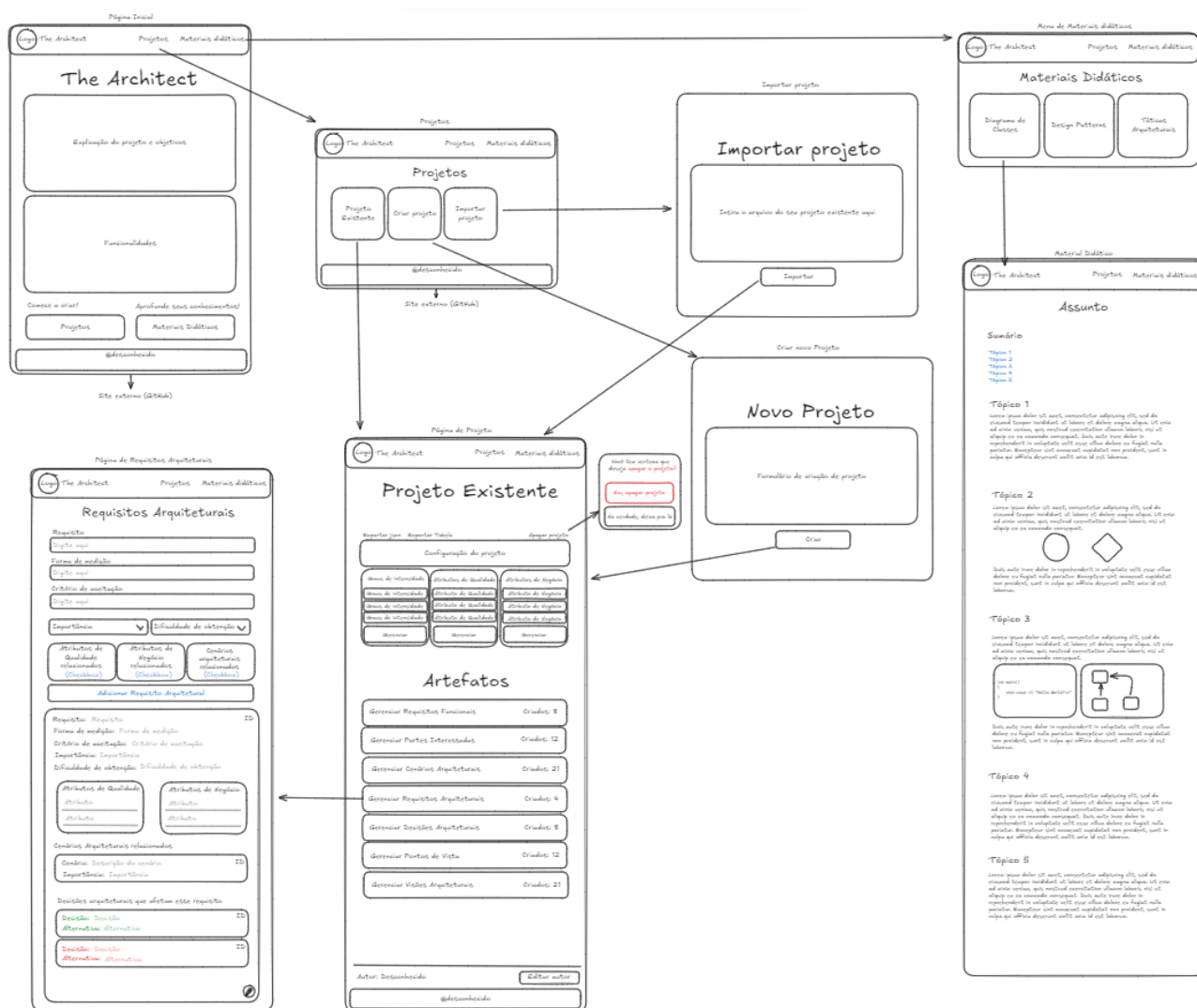


Figura 1: Mapa de navegação simplificado. Fonte: Própria autoria

RESULTADOS E DISCUSSÃO

Durante o planejamento do sistema, foram obtidos vários artefatos de software, definindo as funcionalidades e prioridades técnicas do projeto. A tabela 1 contém os principais requisitos funcionais obtidos durante o planejamento.

A interface do sistema resultante foi condizente com o mapa de navegação final, as figuras 2 e 3 ilustram a página do menu de projetos e a interface para edição de um projeto único, respectivamente. O exemplo da figura 3 foi gerado com inteligência artificial (Google, 2025) para fins ilustrativos.

Como previsto no planejamento e ilustrado pela figura 3, é possível importar projetos criados na plataforma para edição ou visualização. Ao escolher um projeto existente, o usuário é redirecionado para a interface de edição do projeto escolhido. O modelo de dados resultante contém um único tipo de objeto a ser persistido, que contém todas as configurações de um projeto, permitindo a fácil serialização do objeto para exportação.

Com a apresentação do sistema aos estudantes, notou-se que alguns deles apresentavam certa dificuldade em usar o editor de projeto. Portanto, os esforços no desenvolvimento do trabalho passarão a se concentrar em uma cobertura de guias mais robusta, com o objetivo de facilitar o entendimento e uso da plataforma.

ID	Requisito	Importância
RF1	O sistema deve permitir a criação e gerenciamento de artefatos de Engenharia de Software	Alta
RF2	O sistema deve conter explicações didáticas sobre cada tipo de artefato que ele permite a criação e gerenciamento	Alta
RF3	O sistema deve conter explicações didáticas sobre os diferentes tipos de diagramas e, quando cabível, seus equivalentes em código fonte.	Alta
RF4	O sistema deve permitir a importação e exportação de projetos em algum formato de arquivo	Média
RF5	O sistema deve fornecer recomendações de ferramentas que podem ser usadas na produção de artefatos de Engenharia de Software	Baixa
RF6	O sistema deve permitir a exportação de artefatos de Engenharia de Software para tabelas	Média

Tabela 1: Requisitos funcionais do sistema. Própria autoria

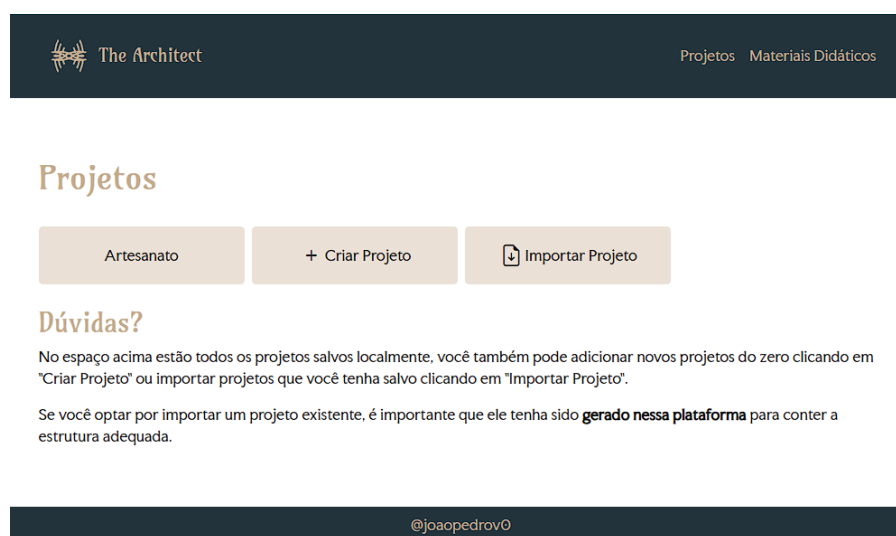


Figura 2: Menu de projetos. Fonte: Própria autoria



Figura 3: Interface para edição de projeto. Fonte: Própria autoria

CONCLUSÕES

O sistema desenvolvido tem como objetivo tornar-se uma ferramenta de ensino da engenharia de software em sala de aula, buscando resolver as dificuldades encontradas pelos estudantes em compre-

ender e escrever seus artefatos de software. Durante o desenvolvimento houve algumas dificuldades no processo de construção da interface com a alta complexidade de gerenciar a forma de exibição dos artefatos de software com naturezas tão distintas. Os próximos passos desse projeto envolvem a migração do *front-end* da aplicação para um *framework* mais robusto em sua gestão de estado e que viabilize de forma mais intuitiva a construção de componentes para representar os artefatos. Além disso, também pretende-se permitir que os estudantes salvem seus projetos de forma persistente para acesso através de uma conta na plataforma.

CONTRIBUIÇÕES DOS AUTORES

J. P. V. G planejou, desenvolveu o projeto e atuou na redação do trabalho. R. E. S. C orientou o projeto. Todos os autores contribuíram com a revisão do trabalho e aprovaram a versão submetida.

REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR ISO/IEC 25010: Engenharia de sistemas e de software — Requisitos de qualidade e avaliação de sistema e de software (SQuaRE) — Modelos de qualidade de sistema e de software*. Rio de Janeiro, 2011.

BECK, K. et al. *Manifesto para Desenvolvimento Ágil de Software*. 2001. <<https://agilemanifesto.org/iso/ptbr/manifesto.html>>. Acesso em: 19 de setembro de 2025.

BootstrapDoc BOOTSTRAP. *Introduction*. [s.d.]. Acesso em: 26 de junho de 2025. Disponível em: <<https://getbootstrap.com/docs/5.3/getting-started/introduction>>.

Google. *Gemini*. Mountain View, CA: [s.n.], 2025. <<https://gemini.google.com>>. Acesso em: 16 de agosto de 2025. Software de chat conversacional.

PARRACHO, T. de M.; ZACARIAS, R. O.; SANTOS, R. P. dos. Os efeitos da experiência de desenvolvedor no processo de ensino-aprendizagem de engenharia de software. *ETD-Educação Temática Digital*, v. 27, p. e025021–e025021, 2025.

SANTOS, R. E. et al. Ferramentas, métodos e experiências no ensino de engenharia de software: um mapeamento sistemático. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2014. v. 25, n. 1, p. 544.

SOMMERVILLE, I. Software engineering (ed.). *America: Pearson Education Inc*, 2011.

SOUZA, M. M. et al. Sparse: Um ambiente de ensino e aprendizado de engenharia de software baseado em jogos e simulação. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2010. v. 1, n. 1.