

A Contribution to the Integration of Software Engineering Tools for Agile Methodologies in Education

Fernanda Mota¹, Stephanie L. Brião¹, Felipe M. Goulart²,
Guilherme Baccarin², Miriam B. Born²

¹Universidade Federal do Rio Grande (FURG) - Rio Grande – RS – Brazil
Universidade Católica de Pelotas (UCPel) - Pelotas – RS – Brazil

{fernanda.mota, stephanie.loi}@furg.br, mbborn@inf.ufpel.edu.br

Abstract. *This paper proposes an educational environment, named MouraOF, designed to support the understanding of the software development lifecycle and agile methodologies. The tool integrates essential software engineering stages—such as requirements gathering and delivery—into an interactive environment that allows for the simulation of projects from start to finish. The methodology combines principles of instructional design and empirical evaluation. Consequently, the web-based prototype includes modules for requirements management, a task board, simplified version control, and a dashboard tracking velocity, lead time, and test coverage. Initial validation involves a heuristic evaluation of the interface and the application of the Technology Acceptance Model (TAM) to preliminarily assess technology acceptance. Thus, from a practical standpoint, this research aids the educational process by facilitating informed choices regarding agile methodologies and fostering competencies such as planning, communication, and continuous quality assessment.*

Resumo. *O presente trabalho propõe um ambiente educacional, denominado MouraOF, para apoiar a compreensão do ciclo de desenvolvimento de software e das metodologias ágeis. Essa ferramenta integra etapas essenciais da engenharia de software, como o levantamento de requisitos da fase de entrega, em um ambiente interativo que permite simular projetos do início ao fim. Também a metodologia combina princípios de design instrucional e de avaliação empírica. Assim, o protótipo web terá módulos para gestão de requisitos, quadro de tarefas, controle simplificado de versões e painel de velocidade, tempo de espera e cobertura de testes. A validação inicial contempla avaliação heurística da interface e aplicação do modelo TAM para análise preliminar da aceitação da tecnologia. Logo, esta pesquisa, do ponto de vista prático, auxilia no processo de formação educacional, facilitando a escolha informada de metodologias ágeis e desenvolvendo competências como planejamento, comunicação e avaliação contínua da qualidade.*

1. Introdução

A Engenharia de Software (ES) é um campo interdisciplinar que aplica princípios sistemáticos e quantitativos ao desenvolvimento, à operação e à manutenção de sistemas [De Vasconcelos et al. 2006, Committee et al. 1990]. Sendo possível reduzir a complexidade, assegurar a qualidade, indicar a produtividade e garantir que os produtos atendam aos requisitos dentro dos prazos e dos custos

definidos [MAFFEO 1992, Maxim and PRESSMAN 2021]. A ES permite ajustes sem reiniciar o projeto, evitando atrasos e custos adicionais [Turine et al. 1996]. Há diversas ferramentas para apoiar etapas específicas, como o levantamento de requisitos, a codificação e os testes [Mnkandla 2008], mas muitas não contemplam todo o ciclo [de Souza and Santander 2011].

A crescente demanda por produtos de maior qualidade e de entrega rápida evidencia a necessidade de soluções integradas também na área educacional. Então, as metodologias ágeis [Beck et al. 2001] surgem como alternativas às abordagens tradicionais, caracterizadas por processos rígidos e documentação extensa. As metodologias ágeis priorizam indivíduos, colaboração e adaptação a mudanças. Então, no processo educacional, essas metodologias promovem eficiência, reduzem desperdícios e mantêm alinhamento às necessidades dos alunos [dos Santos Soares 2004, Bastos and Bastos 2021].

Além disso, as *Application Programming Interfaces* (APIs) são importantes para a integração entre sistemas, abstraindo a complexidade da implementação e disponibilizando apenas os recursos relevantes [Stamboroski et al. 2016, Schmitt 2013]. Entre as APIs, destaca-se a REST, que opera por meio de métodos HTTP como GET, POST, PUT e DELETE, sendo stateless [Jorge 2020]. O padrão arquitetural *Model-View-Controller* (MVC) organiza sistemas em três camadas independentes — *Model*, *View* e *Controller* — para elevar a escalabilidade e facilitar a manutenção [dos Santos Barbosa et al. 2016].

Logo, este trabalho tem como objetivo desenvolver uma aplicação educativa para apoiar a aprendizagem sobre o ciclo de desenvolvimento de software e as metodologias ágeis. Abrangendo desde a obtenção dos requisitos junto ao aluno até a validação e a disponibilização do material final. Embora existam diversas ferramentas voltadas ao gerenciamento de projetos e ao ensino de metodologias ágeis, a maioria delas concentra-se na apresentação de conteúdos ou no suporte a etapas específicas do desenvolvimento de software. Nesse contexto, o MouraFO diferencia-se por incorporar um mecanismo de recomendação de metodologias ágeis capaz de auxiliar estudantes e docentes na escolha da abordagem mais adequada para cada projeto. A recomendação é realizada a partir da análise de características relacionadas à equipe, ao contexto organizacional e aos requisitos do projeto, permitindo que os usuários compreendam não apenas as particularidades de diferentes metodologias, mas também os fatores que influenciam sua seleção em cenários reais de desenvolvimento de software.

2. Trabalhos relacionados

Ferramentas de ES podem ser integradas e direcionadas a aplicações educacionais. Em [Oliveira and de Souza Castro 2015] apresenta-se um *framework* de práticas ágeis para apoiar modelos. Por meio das metodologias ágeis abaixo, o processo de solução técnica apoia o modelo CMMI, e o projeto e a construção do produto (PCP) apoiam o MPS.BR. Entre as metodologias, o *Scrum* destaca-se pela objetividade na definição de papéis e pela facilidade de adaptação [Cruz 2013]. O *Extreme Programming* (XP) foi criado para superar os longos ciclos de desenvolvimento dos modelos tradicionais e baseia-se na comunicação, na simplicidade e no *feedback* [Franco 2007]. Em contrapartida, o Kanban utiliza controle visual para elevar o fluxo de trabalho [Boeg 2010], e o *Lean Software Development* (LSD) adapta a filosofia *Lean* ao desenvolvimento de software, com foco na eliminação de desperdícios e na

maximização da eficiência [Poppendieck and Cusumano 2012]. O Crystal é uma família de metodologias que se adapta ao tamanho da equipe e ao grau de criticidade do projeto [Highsmith and Cockburn 2001]. O *Feature Driven Development* (FDD) tem um ciclo de vida que abrange concepção, planejamento e construção, com práticas de modelagem de objetos de domínio, inspeções de qualidade e *builds* frequentes, promovendo clareza de responsabilidades e visibilidade do progresso [Palmer and Felsing 2001, Silva et al. 2009].

Em contrapartida, [Schoepping et al. 2012], o objetivo é avaliar o *framework* pelas práticas ágeis [Vilain et al. 2007], verificando quais apresentam melhores indicadores para implementação em um processo de desenvolvimento. O resultado obtido com o trabalho permite que a definição de um novo processo a partir do *framework* seja feita de maneira mais simples e assertiva, pois haverá indícios da agilidade de cada prática. O trabalho de [Mnkandla 2008] desenvolve uma estrutura que reúne o melhor conjunto de práticas ágeis, em vez de uma única metodologia específica, proporcionando maior agilidade e fluidez no desenvolvimento de software pela equipe.

[Brandãoi and de Oliveiraii] analisaram como os princípios clássicos da gestão podem ser aplicados ao aprendizado pessoal, considerando as crescentes demandas cognitivas e profissionais do contexto moderno. A metodologia consistiu em uma análise teórico-conceitual, complementada por referências bibliográficas e por exemplos documentais extraídos de notícias e reportagens que foram utilizadas apenas como ilustrações, sem caráter empírico, com foco em carreira e desenvolvimento pessoal.

Além dos estudos apresentados anteriormente, existem pesquisas voltadas especificamente para o ensino de metodologias ágeis em contextos educacionais. [de Oliveira Simões et al.] demonstrou o potencial do uso de métodos ativos, em especial o uso da simulação com blocos de montar, como estratégia para o ensino do *framework* Scrum em cursos de graduação em Engenharia Elétrica (ênfase em Computação). A abordagem prática proporcionou aos alunos não apenas o contato direto com os papéis, eventos e artefatos do Scrum, mas também a vivência de situações reais enfrentadas em projetos de desenvolvimento ágil, como falhas de comunicação, ambiguidade de requisitos e desafios no planejamento.

O AgileMaster [McGee 2014] destaca-se por fornecer um ambiente voltado ao apoio do ensino de metodologias ágeis, permitindo que estudantes tenham contato com conceitos relacionados ao gerenciamento de projetos e ao desenvolvimento iterativo. De forma semelhante, o AgileLearning [Lang 2017] foi concebido como uma plataforma educacional destinada ao aprendizado de práticas ágeis, oferecendo recursos para auxiliar a compreensão dos conceitos e processos empregados durante o desenvolvimento de software.

No contexto nacional, [Meireles and Bonifácio 2015] apresentam um relato de experiência que combina Aprendizagem Baseada em Problemas (PBL) e métodos ágeis no ensino de Engenharia de Software. Os autores observaram que a utilização dessas abordagens contribuiu para sugerir o engajamento dos estudantes e favorecer a aplicação prática dos conteúdos trabalhados em sala de aula. Por outro lado, [Andrade et al. 2016] investigam a aplicação da metodologia Scrum em ambiente de sala de aula, destacando sua contribuição para o desenvolvimento de competências relacionadas ao trabalho colabora-

tivo e à organização de atividades. Em ambos os casos, as metodologias são previamente definidas pelos docentes e aplicadas diretamente aos estudantes.

Diferentemente dessas abordagens, o MouraFO busca apoiar uma etapa anterior ao processo de execução dos projetos: a seleção da metodologia ágil mais adequada para cada contexto. Enquanto os trabalhos relacionados concentram-se na utilização de uma metodologia específica durante atividades educacionais, a proposta apresentada incorpora um mecanismo de recomendação capaz de analisar características do projeto e da equipe, sugerindo metodologias compatíveis com o cenário informado pelo usuário. Dessa forma, o ambiente não apenas apoia o aprendizado sobre metodologias ágeis, mas também estimula a reflexão crítica sobre os critérios envolvidos na tomada de decisão durante o planejamento de projetos de software.

3. Sistema MouraFO

Nesta seção, é apresentada a concepção do sistema MouraFO, caracterizando suas interfaces e respectivas descrições. Por fim, serão identificados os requisitos funcionais e não funcionais para a concepção do projeto e de suas funcionalidades, com base nos requisitos levantados. O desenvolvimento do MouraFO foi feito para ser acessado por qualquer *browser* comumente encontrado no mercado atual, sendo possível o acesso via *mobile*.

3.1. Arquitetura do MouraFO

A arquitetura proposta do MouraFO pode ser observada na Figura 1. Nela, é apresentada uma visão geral do projeto, incluindo suas interfaces e funcionalidades. Pode-se observar que existem algumas interfaces de CRUD, que significam *Create*, *Read*, *Update* e *Delete* para operações com registros. Na interface de projetos, há funcionalidades que a distinguem das demais presentes no sistema, como a escolha da metodologia a ser aplicada e o acompanhamento do andamento da etapa do projeto. Existe também uma interface para o cadastro e a manipulação dos eventos cadastrados.

Aliado a isso, ainda existe a interface de autenticação no sistema para controlar quem tem ou não acesso a determinadas áreas. Na Figura 1, é possível observar conexões entre os círculos, que representam uma interface ou um grupo de interfaces semelhantes; essas conexões indicam o fluxo entre as interfaces. Cada grupo de círculos possui uma cor diferente, sendo cada cor uma das telas estáticas de informações técnicas e conhecimento; as telas de CRUD, de cadastro, leitura, atualização e exclusão de registros; e as ações, as atividades particulares de cada tela.

3.2. Requisitos do MouraFO

Os requisitos funcionais a serem atendidos são: i) Visualizar a lista de cada tipo de registro do banco de dados; ii) Disponibilizar a autenticação de usuários; iii) Possibilitar que o usuário altere sua senha, caso seja de seu interesse; iv) Possibilitar o cadastro de novos registros; v) Realizar a alteração de registros; vi) Realizar a exclusão de registros; vii) Proibir a edição de dados sensíveis e a exclusão de projetos em andamento; viii) Restringir o avanço de etapas, permitindo o avanço apenas para etapas de desenvolvimento ou para a interrupção de projetos em andamento; ix) Filtragem e paginação dos dados listados, proporcionando boa legibilidade dos dados encontrados; x) Interface de Ajuda que contemple todas as funcionalidades; xi) Interface pública de acesso com informações sobre metodologias ágeis; xii) Interface pública de acesso com informações sobre as APIs.

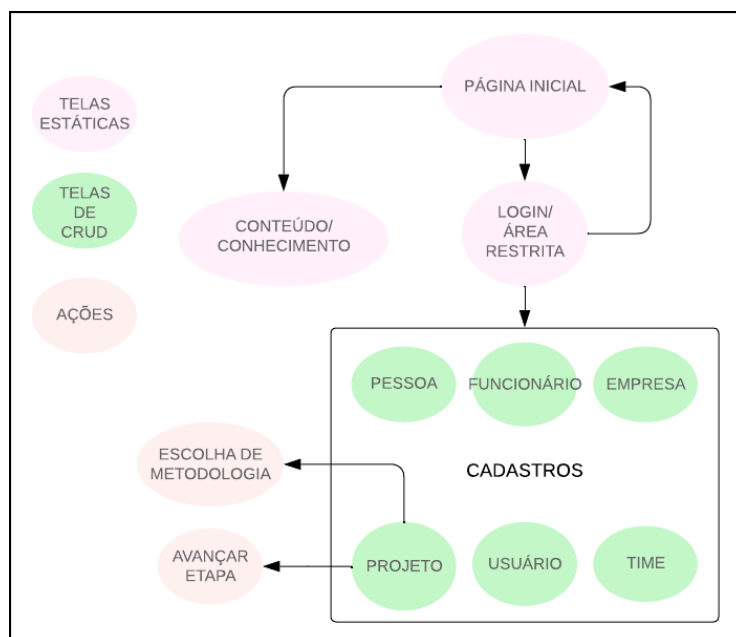


Figure 1. Fluxo de usabilidade do sistema.

Com base na arquitetura de software do sistema MouraFO, foram definidos os requisitos não funcionais a serem atendidos, dentre os quais se destacaram: i) Limitar o número de requisições por minuto na navegação de páginas a 60; ii) Impedir o *login* caso o número de tentativas inválidas ultrapasse 3 em 10 minutos; iii) Operar com recursos computacionais inferiores a 512 MB de memória RAM, tornando-o, assim, acessível a um público mais amplo; iv) Ser responsivo a uma variedade de dispositivos; v) Ser compatível com a maioria das versões de navegadores atualmente utilizadas; vi) Usar *tokens* de segurança em requisições sensíveis; vii) Tempo de resposta inferior a 3 segundos entre requisições; viii) Contemplar sua execução em sistemas operacionais Windows, Linux ou macOS. Após a identificação e análise dos requisitos, foram desenvolvidas as funcionalidades que serão descritas na próxima seção.

3.3. Funcionalidades do MouraFO

As interfaces são interligadas por setas, as quais representam o modo como o usuário deve navegar internamente no sistema, permitindo o livre fluxo entre elas. Porém, há dependências entre registros; para manter a coesão dos dados, é necessário relacioná-los entre si, evitando repetições desnecessárias. As Figuras 2, 3, 4 e 5 apresentam interfaces geradas a partir da simulação de um cenário de aprendizagem em que os alunos utilizam o sistema MouraFO para adquirir conhecimentos sobre Engenharia de Software e metodologias ágeis. Nessa simulação, os estudantes desempenham atividades relacionadas ao ciclo de desenvolvimento de software, incluindo o estudo de diferentes metodologias, o gerenciamento de projetos, a organização de equipes e o acompanhamento das etapas de execução. As telas ilustradas têm como objetivo fornecer evidências preliminares como o ambiente apoia o processo de ensino-aprendizagem por meio da experimentação prática e da interação com ferramentas típicas do contexto profissional de desenvolvimento de software.

Caso o usuário utilize o sistema apenas como ferramenta de gerenciamento de

projetos, o MouraFO permite cadastrar um time responsável pelo projeto a ser gerenciado. A interface inicial do sistema MouraFO apresenta uma breve introdução ao surgimento do sistema, à motivação de seu desenvolvimento, entre outras informações, conforme pode ser observado na Figura 2. Nesta interface, não é necessária autenticação por parte do usuário e, por meio dela, é possível acessar informações sobre conteúdo para usuários que desejam aprender sobre metodologias ágeis e APIs. É possível observar na Figura 3, uma lista de interfaces que o aluno poderá acessar, com ênfase nas interfaces de Metodologias e de APIs, às quais todos os usuários do MouraFO têm acesso. Onde se observa uma pequena seta para baixo, indica-se que há mais de uma interface relacionada a tal tema, como em Metodologias, nas quais existem interfaces sobre XP, Scrum, Kanban, FDD e Crystal.

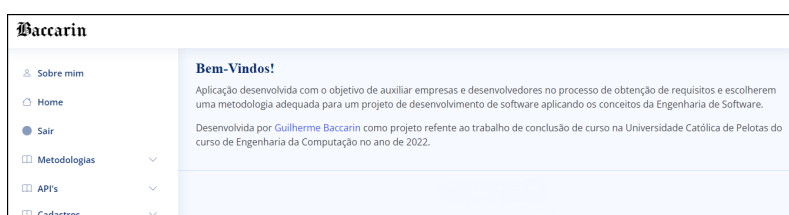


Figure 2. Interface Inicial do MouraFO

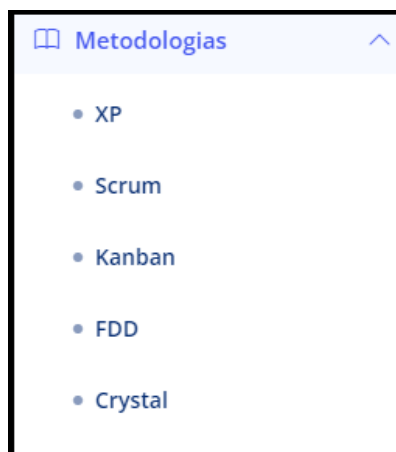


Figure 3. Interfaces das Metodologias do MouraFO

O MouraFO dispõe de diversas informações sobre metodologias ágeis, como interfaces relacionadas ao XP e ao REST, apresentadas na Figura 4C. Tais interfaces não requerem autenticação para serem acessadas, permitindo o acesso a quaisquer alunos que desejem conhecer metodologias ágeis e APIs. Uma das motivações para o desenvolvimento do MouraFO é proporcionar ao aluno a oportunidade de aprender sobre os assuntos abordados no sistema. Na Figura 4D, há uma série de afirmações sobre as características da metodologia, destacando algumas de suas práticas mais comuns, possibilitando que o aluno avalie se a metodologia se aplica ao seu projeto.

Para ter acesso aos registros no MouraFO, é necessário autenticar-se, inserindo as informações de login previamente cadastradas, para garantir a segurança e a integridade do sistema como um todo. Na Figura 5E, pode-se observar que, para salvar um novo registro de 'Empresa', é necessário apenas informar o nome e o respectivo CNPJ, visto

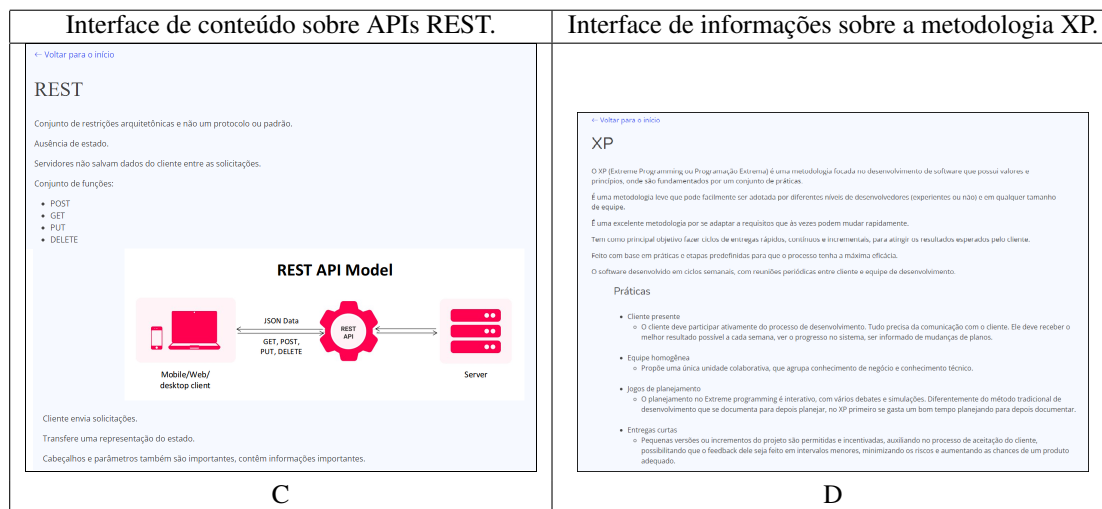


Figure 4. Interfaces do MouraFO

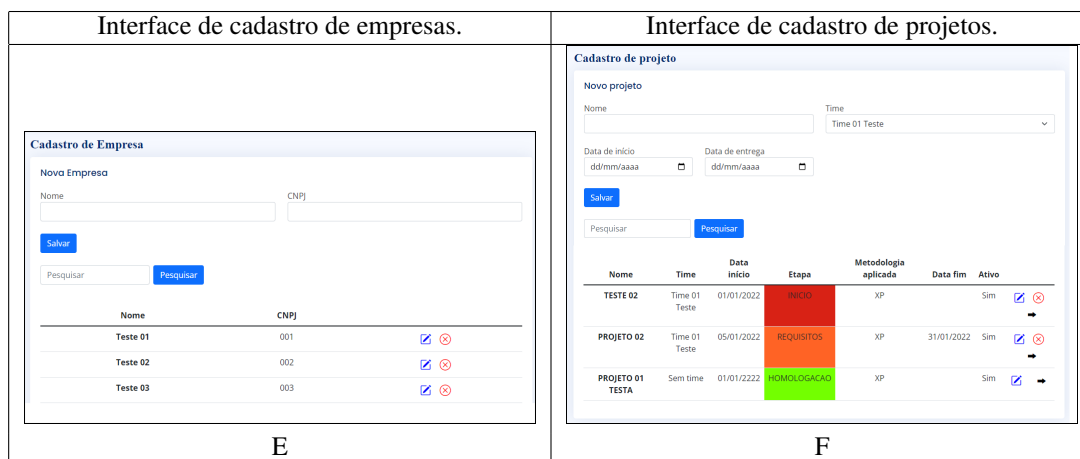


Figure 5. Interfaces do sistema MouraFO

que este não precisa necessariamente ser o número oficial, pois o MouraFO não realiza o cálculo necessário para a validação do CNPJ.

No MouraFO há interfaces para cadastro de pessoas, funcionários, usuários, times, empresas e projetos. Nas Figuras 5E e 5F, é possível observar semelhanças e diferenças entre as interfaces, mantendo sempre um padrão comum entre elas. Na Figura 5F, encontram-se os campos necessários para o cadastro de registros do tipo 'Projeto'. Sendo eles o nome do projeto e o time responsável, cadastrados anteriormente, e as datas de início e de entrega. Vale ressaltar que, como todo projeto é suscetível a atrasos, a data de entrega pode ser alterada posteriormente, se necessário. É possível observar, nesta interface em especial, a coluna 'Etapas', presente na lista de exibição de registros, que serve como uma ferramenta visual e de fácil compreensão para diferenciar as etapas do projeto, sendo que cada cor é atribuída a uma etapa diferente, escolhida aleatoriamente. Após a inserção dos dados sensíveis do projeto a ser cadastrado, o aluno é direcionado à interface para escolher a metodologia ágil a ser implementada no projeto.

Devido a isso, serão exibidas ao aluno diversas perguntas. O estudante irá sele-

cionar quais questões estão relacionadas ao universo da equipe e do projeto a ser desenvolvido. Para facilitar a visualização do usuário, é possível usar uma separação por cor de fundo, em que cada parte contém questões referentes a uma metodologia ágil diferente. Por fim, o MouraFOirá exibir para o usuário qual foi a metodologia mais adequada e irá retornar para a interface de cadastro de 'Projeto'.

3.3.1. Mecanismo de Recomendação de Metodologias Ágeis

O mecanismo de recomendação constitui o principal diferencial do MouraFO. Seu objetivo é auxiliar estudantes e docentes na seleção de metodologias ágeis compatíveis com as características do projeto que está sendo desenvolvido. Para isso, o sistema apresenta um conjunto estruturado de perguntas relacionadas ao contexto organizacional, ao tamanho da equipe, ao grau de complexidade do projeto, à frequência esperada de mudanças nos requisitos, ao nível de documentação desejado e à participação dos *stakeholders* durante o desenvolvimento.

O Algoritmo 1 apresenta o fluxo geral utilizado pelo mecanismo de recomendação. Inicialmente, todas as metodologias recebem pontuação zero. Em seguida, cada resposta fornecida pelo usuário é analisada e associada às metodologias compatíveis com as características informadas. Ao final do processo, a metodologia com maior pontuação é recomendada ao usuário. Em situações de empate, todas as metodologias com a maior pontuação são apresentadas para análise.

Algorithm 1 Processo de Recomendação de Metodologias Ágeis

Require: Respostas fornecidas pelo usuário

Ensure: Metodologia(s) recomendada(s)

```

1: Inicializar pontuação de todas as metodologias com zero
2: for all resposta  $r$  do questionário do
3:   Obter metodologias associadas à resposta  $r$ 
4:   for all metodologia  $m$  associada a  $r$  do
5:      $pontuacao[m] \leftarrow pontuacao[m] + 1$ 
6:   end for
7: end for
8:  $maior \leftarrow \max(pontuacao)$ 
9:  $recomendadas \leftarrow \{m \mid pontuacao[m] = maior\}$ 
10: if  $|recomendadas| = 1$  then
11:   Retornar metodologia recomendada
12: else
13:   Retornar conjunto de metodologias empatadas
14: end if
```

Cada pergunta está associada a uma ou mais metodologias ágeis contempladas pelo sistema. Quando uma resposta é selecionada pelo usuário, pontos são atribuídos às metodologias que apresentam características compatíveis com a situação descrita. Ao final do questionário, o sistema calcula a pontuação total obtida por cada metodologia e apresenta aquela que melhor se adapta ao cenário informado. Por exemplo, projetos ca-

racterizados por equipes pequenas, comunicação intensa e mudanças frequentes tendem a favorecer metodologias como XP e Crystal, conforme pode ser observado na Tabela 1. Os cenários que demandam gerenciamento visual contínuo do fluxo de trabalho tendem a favorecer o uso do Kanban. Projetos que exigem maior formalização e acompanhamento estruturado de funcionalidades podem apresentar maior aderência ao FDD. De forma semelhante, contextos que valorizam ciclos iterativos curtos e participação frequente do cliente tendem a favorecer o Scrum.

Pergunta	Metodologias favorecidas
Equipe pequena?	XP, Crystal
Mudanças frequentes?	Scrum, Kanban
Alta documentação?	FDD
Fluxo contínuo?	Kanban

Table 1. Exemplo de projeto caracterizado por equipes no MouraFO

O mecanismo foi implementado utilizando uma estratégia baseada em regras explícitas de associação entre respostas e metodologias. Dessa forma, além de fornecer uma recomendação automática, o sistema permite que os usuários compreendam os fatores considerados durante o processo de decisão, contribuindo para o aprendizado dos conceitos envolvidos na seleção de metodologias ágeis. Em situações de empate entre metodologias, todas as opções com a maior pontuação são apresentadas ao usuário, permitindo que a decisão final seja realizada de forma consciente e fundamentada.

Do ponto de vista pedagógico, o MouraFO foi concebido como um ambiente de aprendizagem baseado na experimentação prática e na resolução de problemas. A proposta busca aproximar os estudantes de situações semelhantes às encontradas em projetos reais de desenvolvimento de software, permitindo que conceitos relacionados à Engenharia de Software e às metodologias ágeis sejam explorados em um contexto aplicado. Entre os principais objetivos de aprendizagem destacam-se a compreensão das características das diferentes metodologias ágeis, o desenvolvimento de habilidades de planejamento e organização de projetos, o fortalecimento da tomada de decisão baseada em critérios técnicos e a reflexão sobre a adequação de processos de desenvolvimento em diferentes cenários organizacionais.

4. Resultados

Nesta seção, serão apresentadas as avaliações realizadas para a validação do MouraFO, evidenciando os principais aspectos e os resultados obtidos.

4.1. Análise da Interface do MouraFO

Para a avaliação do MouraFO desenvolvido, utilizou-se um instrumento de [Yáñez Gómez et al. 2014], que aborda as funcionalidades e características de um sistema de forma simples, para que o discente e o docente as compreendam, abrangendo diversas categorias da ferramenta. Diversas categorias do instrumento original não se aplicam ao contexto do MouraFO, especialmente aquelas relacionadas ao comércio eletrônico, transações financeiras e compartilhamento externo de informações. Por esse motivo, optou-se por destacar apenas os critérios efetivamente pertinentes ao contexto educacional da ferramenta, conforme pode ser observado na Tabela 2.

Categoria	Aplicável	Resultado
Visibilidade	Sim	Atendida
Consistência	Sim	Atendida
Reconhecimento	Sim	Atendida
Flexibilidade	Parcial	Parcialmente atendida
Estética	Parcial	Parcialmente atendida
Privacidade	Sim	Atendida
Compras/Banco	Não	Não aplicável

Table 2. Relação de categorias relacionadas ao MouraFO

A análise heurística permitiu identificar aspectos positivos relacionados à consistência da interface, à visibilidade das informações, à facilidade de navegação e ao reconhecimento de funcionalidades por parte dos usuários. Por outro lado, observou-se que diversas categorias presentes no instrumento de avaliação não se aplicam diretamente ao contexto do MouraFO, especialmente aquelas relacionadas a transações financeiras, comércio eletrônico, compartilhamento externo de informações e funcionalidades bancárias. Dessa forma, os resultados foram analisados considerando apenas os critérios efetivamente compatíveis com as características do ambiente educacional proposto. Os achados obtidos indicam que a interface apresenta condições adequadas para apoiar a utilização do sistema em atividades educacionais, embora melhorias relacionadas à documentação, acessibilidade e suporte ao usuário ainda possam ser implementadas em versões futuras.

4.2. Modelo TAM

O *Technology Acceptance Model* (TAM), proposto por [Davis et al. 1989] a partir de um contrato IBM–MIT nos anos 1980, é uma teoria de sistemas de informação específica para tecnologia que visa explicar e medir os determinantes da utilização de computadores, mapeando como fatores externos afetam crenças, atitudes e intenções de uso. O TAM tem como propósito avaliar o impacto de variáveis fundamentais identificadas em pesquisas anteriores, destacando-se a Utilidade Percebida (*Perceived Usefulness* — PU), definida como o grau em que um indivíduo acredita que o uso de um sistema ampliaria seu desempenho no trabalho, e a Facilidade de Uso Percebida (*Perceived Ease of Use* — PEOU), entendida como o grau em que o uso do sistema seria livre de esforço; no modelo, a intenção de uso é determinada pela atitude em relação ao uso e pela PU, enquanto a relação entre atitude e intenção reflete a tendência das pessoas a formar intenções favoráveis a ações sobre as quais têm avaliação positiva.

Para a avaliação do sistema, foi disponibilizado para uso a um grupo de 5 pessoas, entre 20 e 60 anos, todas com graduação completa e residentes em Pelotas, RS. As afirmativas foram aplicadas ao grupo de usuários, sendo divididas nos seguintes construtos: i) Facilidade de uso percebida: 1- As funcionalidades de Visualização e Gerência do MouraFO são de entendimento intuitivo; 2- Os procedimentos para definição dos parâmetros necessários às funcionalidades do MouraFO são claros; 3- Com pouco esforço, consigo definir o fluxo ideal de uso dentro do MouraFO; ii) Utilidade percebida: 4- As funcionalidades disponibilizadas pelo MouraFO são relevantes para um gestor de projeto; 5- O MouraFO pode auxiliar na gerência de um projeto; 6- O MouraFO pode

reduzir os custos de tempo e ferramentas para a gerência de um projeto. Os valores das respostas variam da seguinte forma: 0 - Discordo Totalmente; 0,25 - Discordo Parcialmente; 0,5 - Indiferente; 0,75 - Concordo Parcialmente; 1 - Concordo Totalmente.

O cálculo do valor médio das respostas às afirmativas referentes ao construto 'Facilidade de uso percebida', obteve um valor médio, entre as três afirmativas, de 71% do valor máximo total. No cálculo do valor médio a partir das respostas às afirmativas referentes ao construto 'Utilidade percebida', obteve-se um valor médio, entre as três afirmativas, de 93% do valor máximo. Com esse resultado, pode-se concluir que o grupo ao qual foram aplicadas tais afirmativas reconheceu a utilidade real do sistema no universo da gerência de projetos. Pode-se observar, nos resultados, que, apesar do grupo amostral ser pequeno, houve indícios de que os usuários aceitaram o uso do MouraFO. Os resultados indicam uma percepção positiva de utilidade e facilidade de uso por parte dos participantes. Entretanto, devido ao reduzido tamanho da amostra e ao caráter preliminar da avaliação, os resultados devem ser interpretados como evidências iniciais do potencial da ferramenta para apoiar o ensino de metodologias ágeis. Dessa forma, os resultados sugerem potencial de utilização do sistema em contextos educacionais, mas não permitem generalizações sobre seus impactos pedagógicos ou sobre ganhos efetivos de aprendizagem. Destaca-se como limitação desta etapa de validação o número reduzido de participantes e o fato de os avaliadores não representarem integralmente o público-alvo pretendido, composto principalmente por estudantes e docentes em processos formais de aprendizagem.

4.3. Vantagens e Limitações

Durante este trabalho, observou-se que o mercado oferece diversas ferramentas de gestão de projetos e de apoio à implementação de metodologias ágeis. No entanto, a maioria atende apenas a etapas específicas do desenvolvimento, o que exige o uso simultâneo de várias soluções. Com o crescimento dos projetos e das equipes, essa fragmentação torna o processo mais complexo e menos produtivo.

O MouraFO integra as diferentes fases do ciclo de vida de um projeto de software e oferece suporte técnico na escolha da metodologia ágil mais adequada. Para isso, disponibiliza questionários que auxiliam na tomada de decisão e permitem o gerenciamento de múltiplos projetos, atendendo tanto a pequenas quanto a grandes empresas. Atualmente, o MouraFO contempla seis metodologias ágeis: Scrum, Crystal, FDD, Kanban, XP e Lean, com possibilidade de expansão futura. Além disso, prevê o aprimoramento dos questionários para tornar a recomendação mais precisa. Por estar em fase de desenvolvimento, ainda apresenta limitações, como a ausência de redefinição de senha, a impossibilidade de envio de e-mails pelo gestor e o número reduzido de perguntas para avaliação metodológica. Os resultados obtidos indicam potencial para utilização do ambiente em contextos educacionais relacionados ao ensino de metodologias ágeis.

Como limitações desta pesquisa, destaca-se que a avaliação realizada concentrou-se em aspectos relacionados à usabilidade e à aceitação da tecnologia, não contemplando ainda a mensuração direta de aprendizagem ou o acompanhamento do uso do sistema em disciplinas reais de Engenharia de Software. Além disso, a amostra utilizada na avaliação foi reduzida e composta por participantes com perfil diferente daquele que representa o público-alvo principal da proposta. Consequentemente, os resultados obtidos devem ser

interpretados como evidências preliminares, sendo necessárias investigações futuras para avaliar o impacto do MouraFO sobre a aprendizagem, o engajamento e o desenvolvimento de competências relacionadas às metodologias ágeis.

5. Conclusão

O sistema MouraFO é uma ferramenta aplicável ao ciclo completo do desenvolvimento de software, integrando diversas metodologias ágeis externas em um único local. Assim, os resultados obtidos indicam potencial do MouraFO para apoiar processos de ensino e aprendizagem relacionados às metodologias ágeis, embora estudos adicionais sejam necessários para avaliar efetivamente seus impactos educacionais. Ao integrar diversas ferramentas em um único ambiente, o MouraFO permite que o discente/docente utilize apenas a ambientação do MouraFO, reduzindo conflitos e simplificando a rotina de trabalho. Aliado a esse ganho de eficiência no fluxo de aprendizagem, o professor pode se abster de pesquisar no mercado diferentes ferramentas para cada etapa do desenvolvimento, evitando consumir o tempo dos colaboradores e direcionando os recursos financeiros para outras áreas, pois o sistema é *open-source*.

A revisão da literatura em engenharia de software, metodologias ágeis e APIs evidencia a necessidade crescente de desenvolver ferramentas que agilizem e facilitem o desenvolvimento de projetos educacionais. Considerando esse cenário, este trabalho teve como objetivo geral desenvolver o MouraFO, que integra ferramentas de engenharia de software relacionadas às metodologias ágeis para o setor educacional.

Como trabalhos futuros, pretende-se ampliar a validação pedagógica do MouraFO por meio de estudos envolvendo estudantes e docentes em contextos reais de ensino. Pretende-se também realizar experimentos educacionais utilizando avaliações pré e pós-teste para investigar possíveis ganhos de aprendizagem relacionados à Engenharia de Software e às metodologias ágeis. Além disso, serão conduzidos estudos comparativos com outras ferramentas educacionais existentes na literatura, buscando analisar aspectos como engajamento, apoio à tomada de decisão, desenvolvimento de competências e efetividade do processo de ensino-aprendizagem. Em paralelo, novas funcionalidades poderão ser incorporadas ao sistema, incluindo a ampliação do conjunto de metodologias suportadas, melhorias nos mecanismos de recomendação e recursos adicionais de acompanhamento do progresso dos usuários.

6. Declaração de Uso de Inteligência Artificial

Os autores declaram que ferramentas baseadas em Inteligência Artificial foram utilizadas apenas como apoio à revisão textual e organização do manuscrito, permanecendo sob responsabilidade dos autores todas as decisões científicas, análises, resultados e conclusões apresentados neste trabalho.

References

- Andrade, B., Brito, M., and Lima, C. (2016). Metodologia ágil scrum em um ambiente de sala de aula. In *Anais do Workshop de Informática na Escola*, volume 22, pages 251–260.
- Bastos, A. M. and Bastos, A. M. (2021). Uso do scrum como método para otimização na elaboração de projetos. *Research, Society and Development*, 10(9):e40610918329–e40610918329.

- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., et al. (2001). Manifesto for agile software development. 2001.
- Boeg, J. (2010). Kanban em 10 passos. *Tradução de Leonardo Campos, Marcelo Costa, Lúcio Camilo, Rafael Buzon, Paulo Rebelo, Eric Fer, Ivo La Puma, Leonardo Galvão, Thiago Vespa, Manoel Pimentel e Daniel Wildt. C4Media*, page 27.
- Brandãoi, I. C. and de Oliveiraii, G. B. Princípios de gestão aplicados ao aprendizado pessoal: uma análise integrativa entre teorias clássicas e metodologias ágeis e.
- Committee, S. . S. C. et al. (1990). Ieee standard glossary of software engineering terminology (ieee std 610.12-1990). *New York, NY: The Institute of Electrical and Electronics Engineers*.
- Cruz, F. (2013). *Scrum e PMBOK unidos no Gerenciamento de Projetos*. Brasport.
- Davis, F. D., Bagozzi, R. P., and Warshaw, P. R. (1989). User acceptance of computer technology: a comparison of two theoretical models. *Management science*, 35(8):982–1003.
- de Oliveira Simões, P. B., Alves-Souza, S. N., and da Silva, D. A. Métodos ativos de aprendizagem no ensino de scrum em engenharia de computação: A perspectiva discente.
- de Souza, C. F. and Santander, V. F. (2011). Uma proposta de elicitação e análise de requisitos no contexto de médias e pequenas empresas de desenvolvimento de software. In *CIBSE*, pages 285–296.
- De Vasconcelos, A. M. L., Rouiller, A. C., Machado, C. Â. F., and de Medeiros, T. M. M. (2006). Introdução à engenharia de software e à qualidade de software. *Minas Gerais: Universidade Federal de Lavras*.
- dos Santos Barbosa, A. M., Vitor, D. C., Silva, H. A. F., Júnior, N. G. Q., Ribeiro, R. A., de Carvalho, J. C. S., and Luz, F. D. V. (2016). Website empresarial com arquitetura mvc utilizando ddd e angularjs. *RE3C-Revista Eletrônica Científica de Ciência da Computação*, 11(1).
- dos Santos Soares, M. (2004). Comparação entre metodologias ágeis e tradicionais para o desenvolvimento de software. *INFOCOMP Journal of Computer Science*, 3(2):8–13.
- Franco, E. F. (2007). *Um modelo de gerenciamento de projetos baseado nas metodologias ágeis de desenvolvimento de software e nos princípios da produção enxuta*. PhD thesis, Universidade de São Paulo.
- Highsmith, J. and Cockburn, A. (2001). Agile software development: The business of innovation. *Computer*, 34(9):120–127.
- Jorge, L. C. (2020). Projeto e arquitetura de api rest para sistema de monitoramento de redes ópticas. *Repositório Universidade de Brasília*.
- Lang, G. (2017). Agile learning: Sprinting through the semester. *Information Systems Education Journal*, 15(3):14–21.
- MAFFEO, B. (1992). *Engenharia de Software e Especificação de Sistemas*. Campus.

- Maxim, B. and PRESSMAN, R. S. (2021). Engenharia de software: uma abordagem profissional.
- McGee, T. (2014). Going agile?: How to reduce your risk of failure. Project Management Institute.
- Meireles, M. C. and Bonifácio, B. (2015). Uso de métodos ágeis e aprendizagem baseada em problema no ensino de engenharia de software: Um relato de experiência. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 26, page 180.
- Mnkandla, E. (2008). *A selection framework for agile methodology practices: A family of methodologies approach*. PhD thesis, University of the Witwatersrand.
- Oliveira, S. R. B. and de Souza Castro, V. (2015). Um framework de práticas ágeis para apoio à implementação do processo de projeto e construção do produto. *iSys-Brazilian Journal of Information Systems*, 8(2):78–97.
- Palmer, S. R. and Felsing, M. (2001). *A practical guide to feature-driven development*. Pearson Education.
- Poppendieck, M. and Cusumano, M. A. (2012). Lean software development: A tutorial. *IEEE software*, 29(5):26–32.
- Schmitt, P. R. M. (2013). Aplicação web utilizando api google maps. B.S. thesis, Universidade Tecnológica Federal do Paraná.
- Schoepping, G. et al. (2012). Um estudo exploratório a partir de um framework para seleção de práticas ágeis. *Repositório da UFSC*.
- Silva, F., Hoentsch, S. C., and Silva, L. (2009). Uma análise das metodologias ágeis fdd e scrum sob a perspectiva do modelo de qualidade mps. br. *Scientia Plena*, 5(12).
- Stamboroski, M. et al. (2016). Desenvolvimento de uma api para integração de softwares sob um padrão aberto. *Repositório da UFSC*.
- Turine, M. A. S., Masiero, P. C., et al. (1996). Especificação de requisitos: uma introdução. *Repositório da USP*.
- Vilain, P., Fagundes, P. B., and Machado, T. L. (2007). A framework for selecting agile practices and defining agile software processes. In *SEKE*, pages 25–28. Citeseer.
- Yáñez Gómez, R., Cascado Caballero, D., and Sevillano, J.-L. (2014). Heuristic evaluation on mobile interfaces: A new checklist. *The Scientific World Journal*, 2014.