

Utilização da técnica de análise de similares no refinamento e ampliação de requisitos de um software educacional: um relato de experiência

Otacílio J. Pereira¹, Thiago S. Oliveira¹, Raíssa S. A. Moreira¹,
Guilherme G. Souza¹, João Vitor da Souza Silva¹

¹Depto de Engenharias e Computação – Universidade Estadual de Santa Cruz (UESC)
Rod. Jorge Amado, Km 16, Bairro Salobrinho – CEP 45662-900 – Ilhéus – BA – Brazil

Abstract. *This work presents a similar analysis to refine requirements engineering. The software to be developed aims to teach some simple artificial intelligence techniques related to problem solving through state space search and the tool is inspired by the movement of a robot in an industry. In a preliminary view, the functionalities would be limited to moving the robot and then requesting an automatic solution through the tool. Then, a systematic similar analysis was envisaged with two tools related to teaching programming and data structures, Mimo and Visualgo.Net. The analysis showed new possibilities, for example, of planning exercises, of distinguishing teacher and student profiles, of exploring degrees of difficulty, of adopting reward schemes, of incorporating explanations, among others. Therefore, the technique in question proved to be interesting for refining a requirements engineering process.*

Resumo. *Este trabalho apresenta uma análise de similares para refinar uma engenharia de requisitos. O software a ser desenvolvido visa ensinar algumas técnicas simples de inteligência artificial relacionadas com a resolução de problemas por busca em espaço de estados e a ferramenta é inspirada na movimentação de um robô em uma indústria. Em uma visão preliminar, as funcionalidades se limitariam a movimentar o robô e depois solicitar a solução automática pela ferramenta. Daí uma análise de similares de forma sistemática foi prevista com duas ferramentas relacionadas com o ensino de programação e estruturas de dados, o Mimo e o Visualgo.Net. A análise mostrou novas possibilidades, por exemplo, de planejar exercícios, de distinguir perfis de professor e de aluno, de explorar graus de dificuldades, de adotar esquemas de recompensas, de incorporar explicações, além de outras. Desta forma, a técnica em questão se mostrou interessante para refinar um processo de engenharia de requisitos.*

1. Introdução

O desenvolvimento de software tem tido grande importância, e para compreender isso basta perceber o papel que a tecnologia de informação desempenha em diversos aspectos do cotidiano de pessoas e organizações. Estabelecimentos comerciais usam sistemas para automação de seus negócios, empresas controlam seus processos como recursos humanos e produção, e as pessoas usam softwares para sua comunicação e

acesso à conta bancária por exemplo. Enfim, soluções baseadas em software tem apresentado caráter pervasivo e integrado às tarefas de diversos setores e pessoas.

Na Educação, soluções digitais facilitam o estudo e o aprendizado ao fornecer plataformas interativas e ambientes de prática como: sistemas de gestão de conteúdos, simuladores e outros softwares educacionais. Além disso, estas soluções podem promover a personalização do ensino com recursos adaptáveis às necessidades do usuário, tornando o processo de aprendizagem mais eficiente e eficaz. E existem diversas soluções disponíveis, seja para o uso direto de professores e aprendizes, seja para servirem de base para fomentar e inspirar novas ideias e soluções.

O objetivo deste artigo é apresentar como a realização de uma análise de similares contribuiu para ampliar e refinar um conjunto de requisitos para um software educacional. O software educacional é fruto de um projeto de pesquisa e versa sobre o ensino de técnicas de inteligência artificial aplicadas em um cenário inspirado na Indústria 4.0. O projeto está segmentado em algumas partes, por exemplo, a que envolve computação gráfica, os módulos das técnicas de IA, além de outras. A estruturação da análise de similares ocorreu na parte referente aos estudos sobre experiência do usuário (UX, do inglês, *User Experience*) cujo propósito é aplicar técnicas dessa área visando enriquecer a visão do software a implementar.

No tópico 2 é realizada a revisão bibliográfica de tópicos relacionados ao desenvolvimento de software. O tópico 3 descreve o cenário do projeto de pesquisa e o tópico 4 apresenta uma análise de requisitos preliminar do software, antes da ampliação dos requisitos. O tópico 5 aborda a análise de similares em si com a apresentação das ferramentas selecionadas e os aspectos interessantes identificados durante seus usos. No capítulo 6, o resultado da análise é então discutido com incorporação de novas funcionalidades, ampliando e refinando a visão preliminar dos requisitos. Por fim, as considerações finais são feitas no tópico 7.

2. Revisão Bibliográfica

O trabalho trata de uma das técnicas empregadas no desenvolvimento de software, especificamente ligada à análise de requisitos. Portanto este trabalho será posicionado junto a temas como Engenharia de Software e de Requisitos, Engenharia de Usabilidade, Experiência de Usuário e a Análise de Similares.

2.1. Engenharia de Software e Engenharia de Requisitos

Conforme Pressman (2021), a Engenharia de Software abrange um processo, um conjunto de métodos (práticas) e um leque de ferramentas que possibilitam aos profissionais desenvolverem software de qualidade. Observando esta afirmação e a associação com o conceito de processo, naturalmente surgem elementos e questões como: atividades (o que fazer?); artefatos (o que produzir?) e responsáveis (quem?). Ao resgatar da definição a ideia de métodos (práticas), é possível elencar mais um elemento, as técnicas (“como fazer”).

Um tópico tradicional no estudo desta área são os modelos de processo, isto é, de um desenho geral de como as etapas se desenrolam na construção de um software. Alguns exemplos são o modelo em cascata, o iterativo e incremental e o baseado em componentes. Apesar de mais tradicional, o modelo em cascata é interessante pois oferece base para atividades e questões que, mesmo com abordagens diferentes,

precisam ser endereçadas. As etapas deste modelo envolvem a análise de requisitos, planejamento, projeto, desenvolvimento, implantação e manutenção.

A análise de requisitos possui alguns termos correlatos, por exemplo, de engenharia de requisitos. Segundo Valente (2022), ela envolve um conjunto de atividades relacionadas com a descoberta, análise, especificação e manutenção dos requisitos de um sistema. As duas primeiras atividades, descoberta e análise, possuem maior relação com a temática deste trabalho.

2.2. Engenharia de Usabilidade e Experiência do Usuário (UX)

Um requisito de software é uma funcionalidade ou propriedade que o software deve satisfazer. Segundo a ISO25010 (2024), norma que trata um conjunto de características para qualidade do produto de software, algumas destas características são: adequação funcional, eficiência de desempenho, confiabilidade, segurança e outras. A usabilidade, que também é abordada na norma ISO 9241-11 (2024), é definida como a medida pela qual um produto pode ser usado por usuários para alcançar objetivos específicos com efetividade, eficiência e satisfação em um contexto de uso.

Inicialmente este conceito tinha correlação com ergonomia de software e amigabilidade. Talvez o conceito de ergonomia advém da época em que a tecnologia de informação era bastante empregada em empresas e não tanto por usuários gerais, isso por volta da década de 80 e início dos anos 90. Com a popularização dos computadores, da Internet e depois dos smartphones, a tecnologia de informação ampliou bastante seus horizontes. Daí, se antes software era endereçado para ambientes mais controlados no mundo das empresas, agora ela tem um caráter pervasivo, diversas pessoas das mais variadas naturezas se utilizam destes recursos tecnológicos. Neste contexto, o termo usabilidade foi evoluindo, trazendo outras questões derivadas, e hoje a área de Experiência do Usuário se destaca como uma vertente que aborda esta temática.

Em síntese, o foco é gerar produtos que atendam bem as necessidades do usuário gerando uma boa experiência de uso. Para atingir este objetivo, pode-se recorrer a algumas atividades inspiradas no processo de Engenharia de Usabilidade proposto por disciplina de professor do Departamento de Ciência da Computação da UFMG (Pádua, 2024). Ele é composto basicamente das atividades de: i) análise de contexto de uso; ii) estabelecimento dos requisitos de usabilidade; iii) design da interação e iv) a avaliação da usabilidade. Ao decompor um pouco mais, para Pádua (2024), a análise de contexto de uso envolve: i) a análise dos usuários, ii) a análise das tarefas, iii) a análise de similares ou concorrentes e iv) a análise do ambiente.

2.3. Análise de similares

Ao observar a sequência anterior de atividades propostas, a análise de similares está dentro de uma análise do contexto de uso junto, por exemplo, com análise de usuários e de tarefas. Na análise de similares, a ideia é levantar soluções correlacionadas com a essência do problema que o usuário precisa resolver. Depois, faz-se a avaliação das soluções selecionadas para que requisitos sejam elicitados e explorados. Esta técnica, assim como outras de experiência do usuário, tem forte relação com aspectos de inovação. Em essência a busca é por conceber e resolver o problema de alguém, preferencialmente de forma diferenciada e melhor. E para encontrar a diferenciação, é necessário justamente estabelecer paralelos com soluções existentes.

A análise de similares em geral envolve tarefas como o estabelecimento de critérios para identificar similares ou concorrentes, a pesquisa por estas soluções, a organização das sessões de uso, a coleta e registro de boas características e depois a análise dessas características para incorporação no conjunto de requisitos do software. E tudo isso obedecendo preceitos éticos e regramentos de propriedade intelectual.

3. Cenário

As atividades deste relato aconteceram no em um projeto de pesquisa em uma universidade da Bahia no ano de 2024. O projeto é vinculado ao curso de Ciência da Computação em um Departamento de Engenharias e Computação. O objetivo é investigar, conceber e experimentar um processo de desenvolvimento de software e para isso usando software educacional como estudo de caso. O foco do software é abordar técnicas de inteligência artificial, no caso a busca em espaço de estados e os algoritmos genéticos, inspirados em cenários industriais.

O software deve explorar a movimentação de um robô em uma fábrica. A Figura 1 ilustra uma ideia preliminar e rudimentar do software. Na esquerda (Figura 1.a) está apresentada uma imagem alusiva a um chão de fábrica e na direita (Figura 1.b) está a ilustração de uma árvore de busca por um caminho na indústria. Basicamente, pela proposta original do software, o usuário estabelece uma origem e um destino da movimentação e daí explora as movimentações para descoberta do caminho. Em paralelo, ele pode observar a árvore de navegação construída. Depois de uma exploração inicial, em algum momento, o usuário pode pedir para o sistema calcular a rota e veria o robô percorrendo o caminho entre origem e destino.

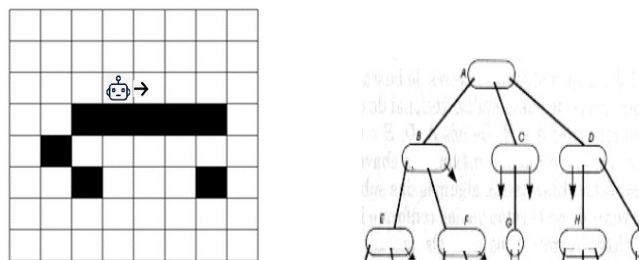


Figura 1 – Figuras para ilustrar funcionalidades preliminares do software

Percebe-se que o conteúdo explorado envolve habilidades sobre programação e estruturas de dados, relevantes para cursos de computação. Portanto, um dos critérios a se considerar na seleção de ferramentas similares foi a aderência com estes assuntos.

4. Análise de requisitos preliminar

Durante a proposição do projeto na universidade, uma análise de requisitos preliminar foi feita, inspirada em uma das aulas do professor sobre as técnicas de IA em questão. O diagrama de casos de uso (Figura 2) e um detalhamento subsequente (Tabela 1) permitem compreender as funcionalidades identificadas nesses momentos iniciais.

Basicamente a utilização envolveria três estágios. No primeiro, referente ao caso de uso “Explorar caminhos”, o usuário movimentava livremente o robô buscando um caminho entre uma origem e destino. A visão da Figura 1, discutida anteriormente, pode ajudar nesta compreensão. Para movimentação, o usuário usa operações como mover

para cima, baixo, esquerda e direita. Em um segundo momento, ao “Visualizar solução”, o usuário requisita a execução do algoritmo para estabelecer o caminho e compara a solução automática com a que ele explorou. No terceiro estágio, de “Acompanhar o algoritmo de busca”, o foco não é só obter a solução do problema mas compreender a forma como o algoritmo foi construindo a árvore de busca até a solução.

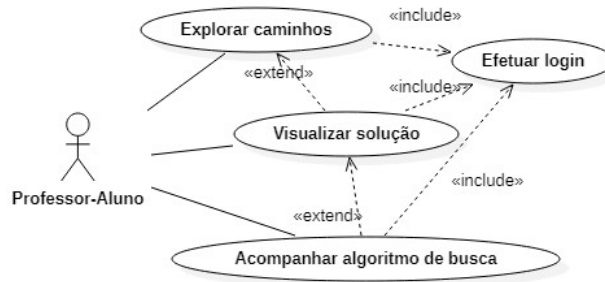


Figura 2. Diagrama de casos de uso de uma especificação preliminar

Como além de produzir o software, o projeto de pesquisa visa refletir sobre o processo de engenharia de software, foi feito também um detalhamento dos casos de uso. A Tabela 1 apresenta um exemplo para “Explorar caminhos”.

Tabela 1. Especificação do Caso de Uso de exemplo

Nome	Explorar caminhos	Ator	Professor-Aluno	ID	CSU_01
Breve descrição	Permite ao usuário explorar livremente movimentações do robô para encontrar um caminho entre uma origem e um destino.				
Pré-condição	Usuário cadastrado no sistema.				
Pós-condição	Um caminho explorado é guardado e pode ser comparado com um caminho calculado pelo algoritmo de busca.				
Fluxo Normal					
Ator			Sistema		
1. O professor-aluno ativa o caso de uso acessando a sistema.			2. O sistema disponibiliza uma visão da indústria com quadriculados.		
3. O professor-aluno indica as coordenadas de origem e destino do caminho, bem como a posição dos obstáculos.			4. O sistema configura internamente os dados do cenário (origem, destino e obstáculos) e habilita as movimentações do robô (para cima, para baixo, esquerda e direita).		
5. O professor-aluno realiza as movimentações explorando as possibilidades e estratégias para construir o caminho.			6. O Sistema reposiciona o robô conforme as movimentações.		
7. Ao chegar no destino o professor-aluno recebe a mensagem de sucesso.			8. O sistema guarda o caminho encontrado pois este pode ser usado para a comparação com a solução algorítmica feita em <Extends Visualizar Solução>		
8. O professor-aluno encerra a exploração.					
Fluxos Alternativos					
Em 3: O professor-aluno pode solicitar uma configuração aleatória sugerida pelo sistema. O sistema então gera a origem, destino e algumas posições de obstáculos.					

Fluxos excepcionais
Em 6 : Ao movimentar, quando há obstáculos, o robô fica parado e uma mensagem de orientação é disparada para o usuário.

5. A análise de similares

A primeira ação na análise de similares foi selecionar os softwares a serem explorados e como base foram usados dois deles: o aplicativo "Mimo: Lógica de Programação" (Mimo GmbH, 2024) e o site "VisuAlgo.net" (NUS, 2024). Eles foram considerados como similares pois em ambos o foco é o ensino de programação, algoritmos e estruturas de dados, semelhante ao que se pretende com o software a ser construído.

5.1. Softwares selecionados: Mimo e VisuAlgo.Net

O Mimo trata o ensino de lógica de programação focada em iniciantes e com uma abordagem gamificada, o aprendizado é progressivo com lições interativas e exercícios práticos. Os conteúdos variam desde os fundamentos de HTML e CSS até conceitos básicos de programação em Python e JavaScript. Cada módulo é composto por pequenos desafios, quizzes e exemplos de código, o que permite ao aluno testar seus conhecimentos imediatamente após a apresentação dos conceitos. O sistema de recompensas inclui certificados, progresso em porcentagem e incentiva o aluno. Adicionalmente, o conteúdo pode se adaptar às preferências iniciais do usuário com trajetórias personalizadas. A abordagem gradual e cumulativa, combinada com feedback constante e recursos visuais, torna a ferramenta eficaz para quem busca iniciar sua jornada na programação. Entretanto, o foco em atividades básicas pode limitar o aprofundamento em tópicos mais complexos, tornando-o mais adequado para iniciantes.

O VisuAlgo (NUS, 2024) é uma ferramenta online dedicada à visualização de algoritmos e estruturas de dados. Seu principal diferencial é a capacidade de animação e visualização passo a passo dos algoritmos, permitindo observar o funcionamento interno seja pelas linhas de código, seja pela animação das estruturas. Ele oferece uma vasta gama de algoritmos, incluindo árvores binárias de busca, grafos, ordenação e pesquisa e outros. O usuário pode interagir diretamente inserindo dados e, além disso, há possibilidades de exibir explicações do conteúdo e fazer quizzes.

5.1. Estratégias para a realização da análise de similares

Inicialmente pensou-se em fazer uma análise de similares conjunta, professor e aluno. Entretanto isso poderia enviesar a análise pois o professor já possui algumas percepções, seja por conta da sua experiência com as disciplinas, seja porque já conhecia uma das ferramentas. Portanto, uma melhor abordagem foi o aluno envolvido fazer a análise para obter a perspectiva do estudante.

Para a realização em si uma divisão em dois momentos foi estabelecida. Primeiro o aluno realizou um estudo mais “tradicional” apenas pesquisando e estudando diretamente o conteúdo que se visava aprender. Para isso ele fez revisões, exercícios, codificações e explorações complementares dos conteúdos estudados nas suas disciplinas durante a graduação. No segundo momento, ele estudou o mesmo conteúdo agora com o suporte da ferramenta. Isso ajudou para que a análise tivesse como referência o problema que o usuário precisa tratar, no caso, aprender programação ou estrutura de dados. Além disso, isso permitiu estabelecer comparativos entre o estudo

tradicional e por meio da ferramenta. Vale lembrar, esta estratégia de dois momentos de estudos foi aplicada aos experimentos com as duas ferramentas.

As características de cada software e impacto nos requisitos preliminares do sistema serão detalhados mais adiante, nesta seção vale algum resumo. Para o software Mimo percebeu-se que: i) ele se ajusta ao nível de conhecimento do usuário por meio de uma pequena entrevista inicial, direcionando o foco de aprendizado; ii) há forte uso de elementos visuais; iii) há um feedback constante; iv) ele explora a progressão em um mapa de aprendizado; e v) ele inclui exercícios cumulativos e mini-projetos para consolidar o aprendizado. Para o VisuAlgo.net, vale comentar: i) o site permite uma maneira eficiente de visualização do funcionamento interno de algoritmos complexos; ii) a visualização e simulação ocorre de forma interativa e descomplicada; iii) parte da interatividade ocorre com a execução passo a passo dos algoritmos visualizada em um pseudocódigo iv) ele oferece a possibilidade de ver a teoria por trás do algoritmo selecionado; e v) ele usa quizzes para reforçar o aprendizado.

Para além da identificação das características, em uma visão geral, percebeu-se que os métodos tradicionais e os sistemas interativos não se excluem, mas podem se complementar. O ensino tradicional proporciona uma compreensão teórica essencial, enquanto os sistemas interativos oferecem uma experiência prática mais dinâmica e engajante. As ferramentas digitais adicionam elementos visuais e interativos que tornam o aprendizado mais acessível e divertido, especialmente para iniciantes. Em relação ao Visualgo.Net, ele serve de apoio para compreensão, mas a codificação em si das estruturas é melhor tratada de maneira mais tradicional. Em um resumo geral, as palavras-chave sobre o uso de ferramentas seriam: “complementar”, “prático”, “engajante” e “dinâmico”.

6. Impacto no refinamento e ampliação dos requisitos

A partir da análise anterior, um conjunto de novos requisitos foram elicitados e serão comentados neste tópico.

6.1. Exercícios pré-configurados e diferenciação de perfis

A partir de algumas das ideias observadas, o sistema a ser implementado deverá permitir a configuração de exercícios com base na definição da origem, destino e posicionamento de obstáculos na indústria. Eles poderão ter níveis de dificuldades e organização em listas com sugestão de ordem para evolução dos estudos. Por exemplo, o professor pode configurar um exercício inicial para mover o robô de uma origem para um destino muito próximo e com a indústria sem obstáculos. Na sequência, em grau de dificuldade crescente, as origens e destinos podem ficar cada vez mais distantes e com mais obstáculos. Alguns exercícios podem exigir o entendimento de casos particulares em que o algoritmo pode ter um comportamento específico. Antes, na versão preliminar, a ferramenta apenas previa explorar caminhos livremente sem este melhor planejamento através do conceito de exercícios organizados em listas.

Estes requisitos refletem, por exemplo, a ideia de progressão presente no Mimo feita por meio de exercícios práticos e cumulativos, com uma noção de encadeamento guiado pelo aumento de dificuldade. Pensar na ideia de exercícios também foi inspirado pelos quizzes no Visualgo.Net, que pede algumas respostas do usuário sobre os conteúdos explorados.

Estas novas características, naturalmente provocaram também a separação dos atores em dois perfis. Se antes ambos professor e aluno usariam livremente a ferramenta configurando seus próprios casos, agora, será necessário existir um ator mais experiente (professor) para fazer toda a configuração, organização e encadeamento dos exercícios.

6.2. Recompensas e Gamificação

As funcionalidades do tópico anterior sobre exercícios, juntamente com outras características, permitem prever a ideia de recompensas e eventuais estratégias de gamificação. As recompensas podem ser das mais simples, individuais, de medalhas, selos ou indicadores de progressão. Elas são conquistadas conforme o aluno evolui nas listas de exercícios. Tempo de dedicação, quizzes respondidos ou outros avanços também podem gerar pontuação. A ideia central é gerar engajamento por meio de recompensas conquistadas com evolução no sistema. Estas características foram naturalmente originadas do estilo de recompensas encontradas no Mimo e que são comuns em outros ambientes gamificados. Outra forma de gamificação diz respeito a competição saudável entre usuários da ferramenta, com cada um tendo sua pontuação classificada e publicada com outros colegas integrantes de um mesmo grupo.

6.3. Explicações e Certificados

O Visualgo.Net apresenta uma característica interessante de apresentar o conteúdo de uma estrutura de dados antes que ela seja simulada com suas operações. O Mimo também usa explicações conciliadas com os exercícios e quizzes. Desta forma, outra funcionalidade inspirada destas ideias é apresentar explicações que podem ser intercaladas com ações no uso da ferramenta.

Por exemplo, o professor pode indicar que em algum momento uma explicação deve ser apresentada para explicar um conceito ou detalhe da técnica de busca em espaço de estados. Em outro momento, durante uma decisão sobre qual direção seguir, a ferramenta pode apresentar a explicação do que são sucessores, isto é, as novas posições originadas a partir de uma certa posição. Além disso, pode ser dada uma explicação sobre como o algoritmo determina estas posições. Enfim, além de exercícios, a ferramenta passa a apresentar explicações que reforçam os conteúdos.

Incorporar explicações, exercícios práticos e quizzes, e organizados em sessões de aprendizagem torna possível prever eventuais emissões de certificados uma vez que itens sejam cumpridos. Com isso, professores de disciplinas correlacionadas podem aceitá-los para covalidar pontuações ou capítulos em suas disciplinas convencionais.

6.4. Outras características

Além destas novas funcionalidades com mudança estrutural na concepção da ferramenta, outras características mais pontuais foram identificadas, por exemplo: i) o uso de recursos visuais dinâmicos para simulações; ii) o acompanhar a simulação pelo pseudocódigo do algoritmo; iii) o feedback imediato próximo de situações a aprender; iv) o apresentar uma interface intuitiva e amigável; e v) ter design leve.

Vale ainda comentar eventuais funcionalidades que não foram aproveitadas. Por exemplo, o Mimo permite a elaboração de mini-projetos, em que exercícios pontuais vão sendo resolvidos até um estágio em que um mini-projeto de uma certa página pode ser concluído. Apesar de achar bem interessante esta característica, não foi possível

criar uma correlação neste momento, mas certamente isso está no consciente da equipe para em algum momento o estalo criativo (“*insight*”) ocorrer.

6.5. Diagrama de casos de uso reformulado e paralelo com versão original

Com as novas ideias estabelecidas, é possível e interessante observar um novo diagrama de casos de uso para criar um paralelo com a visão inicial. A Figura 3 apresenta esta nova versão incorporando apenas algumas funcionalidades discutidas nos itens 6.1 de exercícios e 6.3 de explicações. O software terá outros requisitos funcionais, mas esta versão já facilita a discussão da questão deste trabalho que é a utilidade da análise de similares para ampliar uma perspectiva preliminar. Os casos de uso que apresentam alguma equivalência com casos de uso da versão preliminar (Figura 2) estão em cinza e com asterisco no nome.

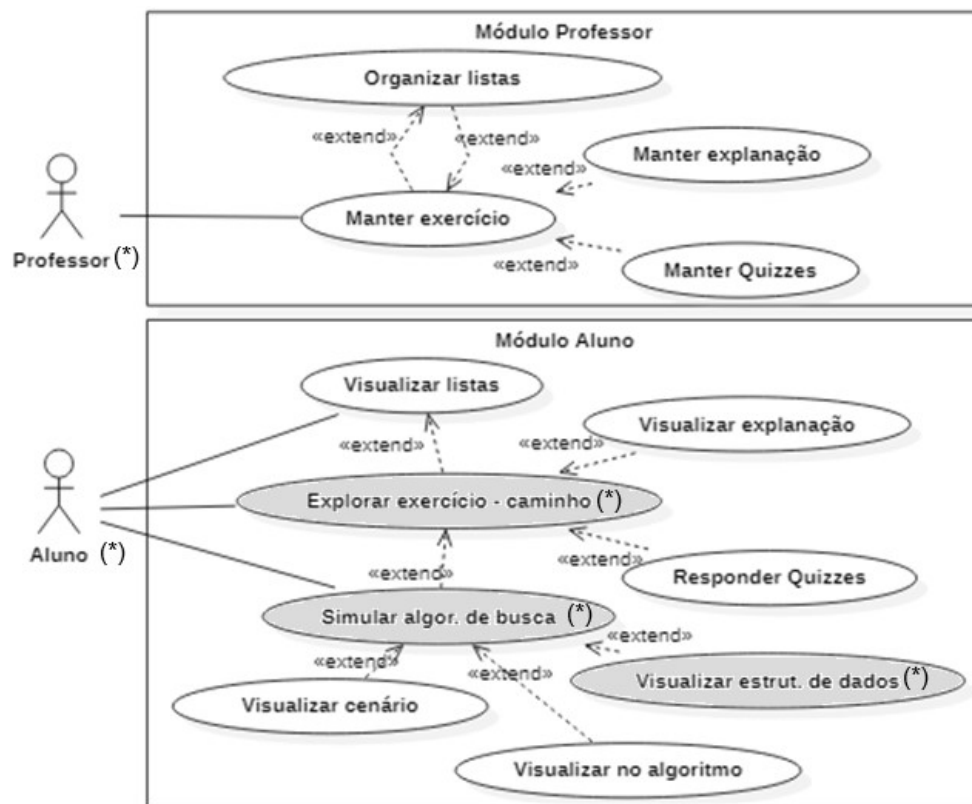


Figura 3 – Diagrama de Casos de Uso após análise de similares

Ao fazer uma comparação com o diagrama de caso de uso inicial (Figura 2) já é perceptível algumas mudanças. A primeira é o destaque para os dois módulos, do professor e do aluno. O que antes era o conceito de explorar caminhos passa agora a ser tratado como resolver exercícios, isto é, o professor mantém um exercício em que ele configura origem, destino, obstáculos, níveis de dificuldades e outros. Daí, no módulo do aluno, o exercício pode ser resolvido. Os exercícios podem ser organizados em listas para encadear e sugerir uma sequência de estudo. Juntamente com os exercícios, o sistema contempla também as ideias de explicações e quizzes.

O caso de uso “Simular algor. de busca”, no módulo do aluno, equivale a solicitar uma solução automática ao sistema, prevista anteriormente. Ainda que antes

houvesse esta perspectiva de avaliar a execução do algoritmo passo a passo, mostrando a estrutura de dados (árvore de busca) da solução, estes recursos passaram a ficar mais sofisticados com mais visões integradas: a do cenário ilustrativo, a das linhas do algoritmo e a das estruturas de dados vinculadas, muito baseado nas ideias do Visualgo.Net.

Como foi dito, o diagrama não apresenta todas as funcionalidades que estão adicionadas após revisão. Por exemplo, a configuração de recompensas, os certificados, cadastros básicos e segurança não estão apresentados. O foco foi criar um primeiro paralelo entre a especificação preliminar e a versão posterior, proveniente da análise de similares.

6. Conclusões

A desenvolvimento de software articula elementos como atividades, artefatos, responsáveis e técnicas para a construção de sistemas. Este trabalho apresentou o uso da análise de similares, uma das técnicas usadas na engenharia de requisitos, sobretudo em tempo de elicitação e comum no contexto de experiência do usuário (UX). A técnica se mostrou bem interessante em dois sentidos, primeiro pela quantidade de novos requisitos e casos de uso que foram agregados. Isso permitiu uma nova visão e uma nova perspectiva para a ferramenta, muito mais interessante. O segundo fator interessante foi que, o aluno envolvido ao usar a técnica pode explorar o propósito da ferramenta, se ambientar e enriquecer as contribuições para o projeto de pesquisa.

Referências Bibliográficas

- ISO 25000. ISO 25000 Standards – Software and Data Quality, ISO/IEC 25010. Disponível em: <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>. Acesso em: 20 out 2024.
- ISO 9241-11. Requisitos Ergonômicos para Trabalho de Escritórios com Computadores Parte 11 – Orientações sobre Usabilidade. Disponível em: <https://www.iso.org/standard/63500.html>. Acesso em: 20 out 2024.
- Pádua, C. Engenharia de Usabilidade – materiais de aula. Disponível em: <https://homepages.dcc.ufmg.br/~clarindo/disciplinas/eu/material/index.htm>. Acesso em: 30 set 2024.
- Mimo GmbH. Learn to code with Mimo: Python, Javascript, HTML, CSS and more. Disponível em: <http://mimo.org>. Acesso em: 22 mar.2024.
- National University of Singapore (NUS). VisuAlgo.net – visualising data structures and algorithms through animation. Disponível em: <https://visualgo.net/en>. Acesso em: 22 mar. 2024.
- Pressman, R., Maxim, B. R. (2021) Engenharia de Software. 9.ed. AMGH: Porto Alegre.
- Valente, M.T. (2022) Engenharia de Software Moderna, Editora Independente.
- Sommerville, I. (2019) Engenharia de Software. 10.ed. Pearson Universidades: São Paulo.