

PAULO ANDRÉ PIMENTA ARAGÃO

**Scrum XPerience: um protótipo de jogo voltado para o ensino
de metodologias ágeis**

São José do Rio Preto

2021

Paulo André Pimenta Aragão

Scrum XPerience: um protótipo de jogo voltado para o ensino de metodologias ágeis

Trabalho de Conclusão de Curso apresentado ao Departamento de Ciências de Computação e Estatística do Instituto de Biociências Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientadora: Profa. Dra. Rogéria Cristiane Gratão de Souza

São José do Rio Preto

2021

Paulo André Pimenta Aragão

Scrum XPerience: um protótipo de jogo voltado para o ensino de metodologias ágeis

Trabalho de Conclusão de Curso apresentado ao Departamento de Ciências de Computação e Estatística do Instituto de Biociências Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Banca Examinadora:

Profa. Dra. Rogéria Cristiane Gratão de Souza
UNESP – São José do Rio Preto
Orientadora

Profa. Dra. Adriana Barbosa Santos
UNESP – São José do Rio Preto

Prof. Dr. Adriano Mauro Cansian
UNESP – São José do Rio Preto

São José do Rio Preto

2021

A659s Aragão, Paulo André Pimenta
Scrum XPerience: um protótipo de jogo voltado para o ensino de metodologias ágeis / Paulo André Pimenta Aragão. -- São José do Rio Preto, 2022
87 p. : il., tabs.

Trabalho de conclusão de curso (Bacharelado - Ciência da Computação) - Universidade Estadual Paulista (Unesp), Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto
Orientadora: Rogéria Cristiane Gratão Souza

1. Engenharia de software. 2. Desenvolvimento ágil de software. 3. Jogos educativos. 4. Arquitetura de software. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

Para Keyla, Antonio, Edson e Daniela.

Agradecimentos

Primeiramente, gostaria de agradecer a Deus, que me forneceu as forças necessárias para a realização de todo o trabalho. Graças a Ele, consegui superar as dificuldades encontradas no ano de 2020, dada a pandemia decorrida pelo COVID-19. Além disso, graças a Ele, pude me juntar ao curso de Bacharelado em Ciência da Computação no ano de 2018 e superar os desafios de ter ingressado com um mês de atraso.

Também gostaria de agradecer a todos os meus familiares por todo o apoio e motivação fornecidos por eles. Em especial, agradeço minha mãe, Keyla, que toda minha vida me motivou a me dedicar e a me tornar cada dia melhor. Agradeço meu pai, Antonio, o qual tive mais contato apenas ao ingressar na universidade, mas que desde então me apoiou em cada decisão e esteve ao meu lado nos momentos bons e ruins. Agradeço imensamente ao parente caminhoneiro Edson Pimenta, o qual tenho o maior orgulho de chamar de avô, que sempre acompanhou minhas atividades, que me motivou e deu forças para seguir em frente, além de realizar grandes chamadas telefônicas nos momentos mais cruciais da faculdade. Ao restante dos familiares, agradeço novamente todo o apoio.

Agradeço a todos os meus amigos por sempre estarem comigo, dando motivação e forças para concluir meus trabalhos. Em especial, gostaria de agradecer meu amigo Nagi, o qual considero um irmão e me apoia a mais de dez anos. Agradeço aos amigos Rodolfo e João, pelos momentos de alegria e descontração quando eu mais necessitava. Agradeço ao amigo Adônis pelo apoio, motivação e ajuda em tudo. Agradeço aos amigos Victor e Leandro, os quais conheci na universidade e sempre me apoiaram e auxiliaram na maior parte do meu conhecimento atual. Agradeço especialmente a Daniela, que me motivou e apoiou em todas as minhas ações, acadêmicas ou não, além de ter me auxiliado na escolha de tema da minha iniciação científica.

Gostaria de agradecer toda a orientação e ensinamento concedido pela professora doutora Rogéria, a qual me orienta desde o início de 2019. Passei por fases boas e ruins, mas graças a sua orientação e motivação, consegui chegar até onde cheguei como aluno e como pessoa e, por isso, sou eternamente grato. Além disso, gostaria de agradecer a todos os membros do Grupo de Estudos e Pesquisas em Engenharia de Software (GEPES), o qual fiz parte por três anos. Em especial, agradeço ao professor Allan, que me auxiliou, principalmente, na etapa inicial de elaboração do projeto de iniciação

científica. Também agradeço a todos os meus professores, os quais participaram da minha formação na UNESP.

Por fim, agradeço à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) – processo nº 2019/20663-0 pelos recursos concedidos.

– Tudo o que temos de decidir é o que fazer com o tempo que nos é dado.

(Gandalf, O Senhor dos Anéis: A Sociedade do Anel – J. R. R. Tolkien)

Resumo

A Engenharia de Software é uma área de relevância para o desenvolvimento de sistemas computacionais, dado que ela define técnicas e metodologias que contribuem para a garantia da qualidade do produto final, bem como para a organização da produção do sistema. Dentre as duas vertentes de métodos de desenvolvimento existentes, tradicional e ágil, destaca-se o uso das metodologias ágeis, em especial as metodologias Scrum e XP, com o intuito de aumentar a velocidade de produção do software e promover a aceitação às mudanças nos requisitos. Neste contexto, é importante contribuir para a formação sólida dos futuros profissionais desenvolvedores de software, de forma que tenham o conhecimento necessário para aplicar os princípios ágeis adequadamente. No entanto, o uso de métodos tradicionais para o ensino do vasto referencial teórico envolvendo a Engenharia de Software promove pouca dinâmica e prática. Isso resulta, em sua maior parte, na desmotivação dos alunos, prejudicando suas formações. Como solução para esse problema, softwares educativos foram desenvolvidos, em especial jogos educativos, promovendo um ensino dinâmico e divertido aos estudantes. Dada a importância de tais jogos para o processo de ensino-aprendizagem da Engenharia de Software, é importante que seu desenvolvimento possua uma base conceitual consolidada. Para isso, este trabalho apresenta um modelo arquitetural para jogos educativos voltados ao ensino das metodologias ágeis Scrum e XP. De maneira complementar, como forma de avaliar a aplicabilidade da arquitetura estabelecida para o desenvolvimento de futuros jogos educativos relacionados a essa área, foi desenvolvido um protótipo de jogo educativo. Tal protótipo foi avaliado por 30 alunos de graduação e pós-graduação em Computação e seis docentes atuantes na área de Engenharia de Software, os quais contribuíram para a identificação de problemas e sugestões de melhorias, bem como para o apontamento dos aspectos positivos do trabalho. Com isso, foi possível concluir que o protótipo elaborado contribui para a aprendizagem de metodologias ágeis, bem como para o avanço da pesquisa e desenvolvimento de jogos educativos voltados para essa área.

Palavras-chave: ensino de metodologias ágeis, arquitetura de software, protótipo de jogo educativo.

Abstract

Software Engineering is a relevant area for software development, defining techniques and methodologies that contribute to the quality assurance of the final product and the organization of the system's production. Among the two strands of existing development methods, traditional and agile, the use of agile methodologies stands out, especially the Scrum and XP methodologies, to increase the speed of software production and promote acceptance of changes in requirements. In this context, it is relevant to contribute to the solid formation of future software developers, given the necessary knowledge for them to apply agile principles properly. However, the use of traditional methods for the teaching of the vast theoretical framework involving Software Engineering promotes small dynamics and practice. It results, for the most part, in demotivating students, harming their education. As a solution for this problem, educational software emerged, especially educational games, promoting dynamic and fun teaching for students. Given the importance of such games for the teaching-learning process of Software Engineering, it is relevant that their development has a consolidated conceptual basis. So, this work presents an architectural model for educational games aimed at teaching agile Scrum and XP methodologies. In addition, an educational game prototype was developed to assess the applicability of the established architecture for the development of future educational games. This prototype was evaluated by 30 undergraduate and graduate students in Computing and six professors working in Software Engineering area, who contributed to identifying problems and suggestions of improvements and pointed out the positive aspects of the work. Thus, it was possible to conclude that the prototype developed contributes to the learning of agile methodologies, the advancement of research, and the development of educational games aimed at this area.

Keywords: agile software development, software architecture, educational game prototype.

Lista de Ilustrações

Figura 1.	Fluxograma do Trabalho	5
Figura 2.	Diagrama Arquitetural.....	28
Figura 3.	Decomposição Modular – Acesso.....	29
Figura 4.	Decomposição Modular – Configurações da Partida.....	30
Figura 5.	Decomposição Modular – Ensino	31
Figura 6.	Decomposição Modular – Conteúdo Didático	32
Figura 7.	Decomposição Modular – Gerenciador.....	33
Figura 8.	Decomposição Modular – Gerenciador de Erros	34
Figura 9.	Decomposição Modular – Gerenciador de <i>Feedback</i>	35
Figura 10.	Página Inicial	38
Figura 11.	Página de Cadastro	38
Figura 12.	Página de Editar Dados.....	39
Figura 13.	Telas de Autenticação e Configuração da Sala	40
Figura 14.	Tela de Configurações da Partida.....	40
Figura 15.	Modelo da Tela de Escolha de Ação	41
Figura 16.	Estatísticas e <i>Feedbacks</i>	45
Figura 17.	<i>Ranking</i>	46
Figura 18.	Preenchimento das Escolhas do Jogo	46
Figura 19.	Modos de Jogo.....	47
Figura 20.	Ajuda	47
Figura 21.	Mapa do Jogo	48
Figura 22.	Protótipo em Execução	48
Figura 23.	Exemplo de <i>Feedbacks</i>	49
Figura 24.	Histograma da Avaliação dos Alunos	53

Lista de Tabelas

Tabela 1.	Princípios Ágeis.....	8
Tabela 2.	Princípios e Etapas da XP.....	9
Tabela 3.	Entidades do Time Scrum	10
Tabela 4.	Etapas da Sprint.....	11
Tabela 5.	Tabela de Dimensões.....	12
Tabela 6.	Métodos Motivacionais	13
Tabela 7.	Questões de Pesquisa – Jogos Educativos.....	14
Tabela 8.	<i>Strings</i> de Busca – Jogos Educativos	14
Tabela 9.	CrITÉrios de Inclusão e Exclusão – Jogos Educativos	15
Tabela 10.	Questões de Pesquisa – Protótipos de Jogos Educativos.....	17
Tabela 11.	<i>Strings</i> de Busca – Protótipos de Jogos Educativos	17
Tabela 12.	CrITÉrios de Inclusão e Exclusão – Protótipos de Jogos Educativos	18
Tabela 13.	Aplicação dos CrITÉrios Avaliativos	20
Tabela 14.	Requisitos Funcionais.....	25
Tabela 15.	Cenários do Protótipo	36
Tabela 16.	Papéis do Scrum XPerience – XP	42
Tabela 17.	Papéis do Scrum XPerience – Scrum	43
Tabela 18.	Relação entre Papéis e Cenários do Scrum XPerience.....	44

Lista de Abreviaturas e Siglas

XP – *eXtreme Programming*

IEEE – *Institute of Electrical and Electronic Engineers*

ACM – *Association for Computing Machinery*

IBILCE – Instituto de Biociências, Letras e Ciências Exatas

FAPESP – Fundação de Amparo à Pesquisa do Estado de São Paulo

QP – Questão de Pesquisa

TCC – Trabalho de Conclusão de Curso

Sumário

Capítulo 1 – Introdução	1
1.1. Contextualização do Trabalho	1
1.2. Motivação	2
1.3. Justificativa	3
1.4. Objetivos	4
1.5. Metodologia	4
1.6. Organização da Monografia.....	5
Capítulo 2 – Revisão Bibliográfica.....	7
2.1. Metodologias Ágeis	7
2.2. Características Almejadas para Softwares Educativos	11
2.3. Jogos Educativos para o Ensino de Engenharia de Software	13
2.4. Protótipos de Jogos Educativos	17
2.5. Análise Empírica de Jogos Educativos para Engenharia de Software.....	19
2.6. Considerações Finais	21
Capítulo 3 – Scrum XPerience	23
3.1. Descrição Geral.....	23
3.2. Requisitos Funcionais	24
3.3. Arquitetura de Software	28
3.3.1. Acesso	29
3.3.2. Configurações da Partida	29
3.3.3. Ensino.....	31
3.3.4. Conteúdo Didático.....	32
3.3.5. Gerenciador	32
3.3.5.1. Gerenciador de Erros	34
3.3.5.2. Gerenciador de <i>Feedback</i>	34

3.4. Protótipo do Jogo Educativo	35
3.4.1. Controle de Acesso.....	37
3.4.2. Partida.....	39
3.4.3. Área do Docente	49
3.5. Considerações Finais	50
Capítulo 4 – Avaliação do Protótipo	51
4.1. Processo de Avaliação	51
4.2. Formulário Avaliativo para Alunos	52
4.3. Formulário Avaliativo para Docentes	53
4.4. Reestruturação do Protótipo.....	54
4.5. Considerações Finais	55
Capítulo 5 – Conclusão	57
5.1. Considerações Finais	57
5.2. Contribuições	58
5.3. Trabalhos Futuros	59
Referências	61
Apêndice A – Formulário Avaliativo para Alunos	66
Apêndice B – Formulário Avaliativo para Docentes	72

Capítulo 1 – Introdução

1.1. Contextualização do Trabalho

O uso das metodologias e práticas definidas pela Engenharia de Software é fundamental para o gerenciamento e desenvolvimento de um software (PRESSMAN, 2011; SOMMERVILLE, 2011). Portanto, os desenvolvedores de soluções computacionais devem possuir um embasamento sólido a respeito das técnicas da Engenharia de Software, de modo a garantir a qualidade do produto a ser desenvolvido. Porém, dado que essa é uma área com vasto referencial teórico, seu ensino é dificultado, causando, diversas vezes, desmotivação nos alunos.

A falta de conhecimento sobre a Engenharia de Software pode comprometer a qualidade dos softwares desenvolvidos, o que incentivou pesquisadores a conduzirem estudos e elaborarem soluções computacionais que auxiliassem no processo de ensino-aprendizagem dessa disciplina. Com isso, os softwares educativos, especificamente jogos educativos, se destacaram, uma vez que o seu uso demonstrou resultados positivos no ensino de tópicos da Engenharia de Software (GARCÍA *et al.*, 2019; SANTOS *et al.*, 2019).

Os softwares educativos abordam, dentre diversos outros, os tópicos de gerenciamento de projetos e metodologias de processo, as quais se dividem em tradicionais e ágeis (CALDERÓN *et al.*, 2017; WANGENHEIM *et al.*, 2013; LEE, 2016; JÁCOME, 2013). As metodologias tradicionais se caracterizam por realizar um

planejamento inicial das etapas a serem contempladas pelo desenvolvimento do produto, não considerando possíveis mudanças (PRESSMAN, 2011). Essa abordagem pode resultar em retrabalho e problemas na implementação de alterações solicitadas pelo cliente, o que gera custo e possíveis alterações de prazo de entrega do software. As metodologias ágeis, por sua vez, buscam acelerar o processo de desenvolvimento do sistema, aceitando as mudanças como inevitáveis, o que diminui significativamente seu impacto. Assim, os métodos ágeis promovem maior comunicação, tanto entre a equipe, quanto entre cliente e desenvolvedores, reduzindo riscos e priorizando o menor tempo de entrega do produto. Dentre as metodologias ágeis existentes, as que ganharam mais enfoque foram o Scrum e a XP, sendo o Scrum a mais utilizada atualmente.

Dado o aumento da elaboração e uso de softwares educativos voltados para o ensino da Engenharia de Software, surgiram trabalhos que buscam analisar a qualidade de tais soluções, como os estudos de Petri *et al.* (2018) e Kosa *et al.* (2016). Também foram conduzidos estudos que propõem questionários avaliativos voltados para jogos educativos sobre Engenharia de Software (SAVI *et al.*, 2011; PETRI *et al.*, 2020), os quais devem ser respondidos pelo público que fez uso do software. Por fim, também foi possível observar a definição de características almejadas para tais softwares educativos (ALMEIDA; ALMEIDA, 2015).

1.2. Motivação

Dada a importância dos jogos educativos para auxiliar o ensino da Engenharia de Software, tem-se a necessidade de estabelecer uma base sólida para construção de soluções de qualidade que sejam capazes de cumprir as expectativas relacionadas ao ensino dessa disciplina. Para tanto, destaca-se a elaboração de um modelo arquitetural para facilitar o entendimento do software almejado, bem como sua implementação. Uma arquitetura de software é elaborada com base em um conjunto de requisitos, os quais detalham o funcionamento e comportamento do sistema alvo. Segundo Pressman (2011), a arquitetura de software representa o sistema a ser implementado, a qual permite a análise e atendimento dos requisitos definidos, reduzindo também os riscos do desenvolvimento.

De forma complementar, a prototipagem também é uma técnica definida pela Engenharia de Software para auxiliar o desenvolvimento de soluções computacionais de qualidade, incluindo jogos educativos, sendo voltada para testes e experimentos. Assim,

é válido ressaltar a importância da construção de um protótipo de software, visando garantir melhor qualidade a um sistema (SOMMERVILLE, 2011; PRESSMAN, 2011).

Diante deste cenário, a definição de um modelo arquitetural para jogos educativos voltados ao ensino de metodologias ágeis, especialmente para Scrum e XP, auxiliará o desenvolvimento de novos jogos educativos sobre esse tema. Ademais, a construção de um protótipo de software, com base na arquitetura proposta, pode comprovar a aplicabilidade do modelo arquitetural, além de apontar eventuais necessidades de aprimoramentos, facilitando e contribuindo para a qualidade dos futuros jogos educativos voltados para o ensino de metodologias ágeis.

1.3. Justificativa

Os jogos educativos voltados para o ensino dos diversos tópicos da Engenharia de Software ganharam destaque por apresentarem bons resultados com o seu uso, potencializando o processo de ensino-aprendizagem de tal disciplina (GARCÍA *et al.*, 2019; SANTOS *et al.*, 2019).

Dentre os principais tópicos abordados pelos jogos educativos voltados para a Engenharia de Software, salienta-se as metodologias ágeis (CALDERÓN *et al.*, 2017; WANGENHEIM *et al.*, 2013; LEE, 2016; JÁCOME, 2013). No entanto, observa-se que os trabalhos disponíveis na literatura abordam, em sua maior parte, a metodologia Scrum (AMMONS; BANSAL, 2017; SOUZA *et al.*, 2017; LEE, 2016; BRITO, 2016; CASTRO *et al.*, 2017; WANGENHEIM *et al.*, 2013), inclusive com a definição de protótipos (BRITO; VIEIRA, 2017; GKITSIS, 2011). Assim, novos projetos voltados para o ensino de outras metodologias ágeis, como a XP, assim como soluções que promovam comparativos entre metodologias ágeis, representam uma importante contribuição para o aprimoramento do processo de ensino-aprendizagem deste tópico.

Além disso, por meio de análise observacional, foi possível detectar carências nas soluções atuais, como: falta de interação social em jogos digitais; não uso de instruções simples; não gerenciamento de erros e fornecimento de *feedbacks*; progressão indevida do aprofundamento do conteúdo; e não avaliação dos conceitos ensinados. Assim, tem-se a necessidade de propor soluções alternativas para sanar tais deficiências.

Por fim, foi possível notar, por meio de uma revisão sistemática da literatura, que existem poucos trabalhos referentes à arquitetura de software para jogos educativos (STAVREV *et al.*, 2018; VEGT *et al.*, 2016; RODRÍGUEZ-SILVA *et al.*, 2015;

SKALKA; DRLÍK, 2018). Esses estudos focam nas características almejadas para jogos educativos, de modo a facilitar o seu desenvolvimento e garantir a sua qualidade. A quantidade restrita de pesquisa referente a esse tópico indica outra carência a ser sanada, em conjunto com as demais previamente identificadas, justificando o desenvolvimento do presente trabalho.

1.4. Objetivos

Este trabalho estabelece, como objetivo geral, contribuir para o processo de ensino-aprendizagem de metodologias ágeis, usando uma abordagem capaz de motivar o engajamento dos alunos aos estudos.

Para tanto, dois objetivos específicos foram estabelecidos:

- Desenvolver um diagrama arquitetural voltado para jogos educativos sobre ensino ágil, enfatizando as metodologias Scrum e XP, com o intuito de sanar as principais carências identificadas nas soluções existentes;
- Desenvolver um protótipo de jogo educativo, a partir do diagrama arquitetural, de modo a evidenciar sua aplicabilidade para nortear a construção de novos jogos educativos de qualidade.

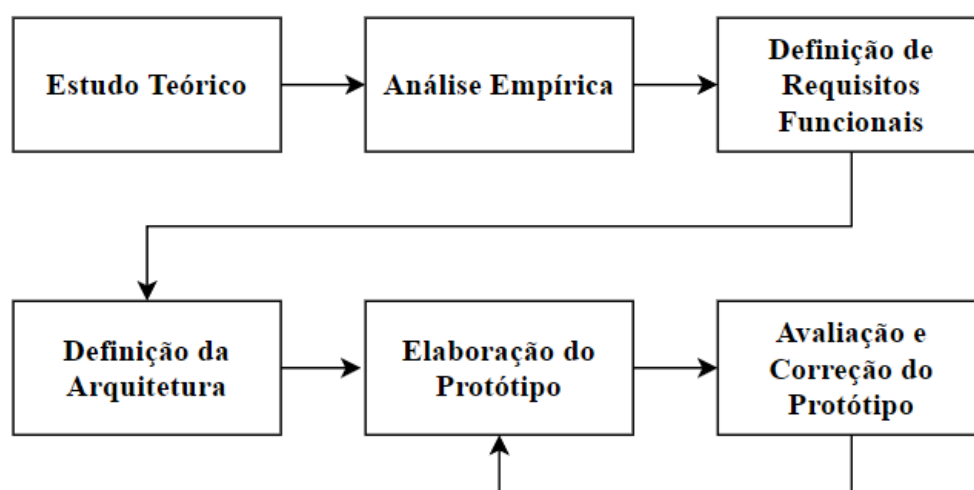
1.5. Metodologia

Para o desenvolvimento deste trabalho, foram estabelecidas seis etapas, conforme esquematizadas na Figura 1 e descritas a seguir:

- Estudo teórico sobre metodologias ágeis, especificamente sobre o Scrum e XP, e sobre prototipagem. Além disso, realização de duas revisões sistemáticas da literatura, sendo uma a respeito do uso de jogos educativos para o ensino de Engenharia de Software e uso de arquitetura de software para jogos educativos, e uma acerca do uso de protótipos para o desenvolvimento de jogos educativos;
- Análise empírica dos jogos educativos voltados ao ensino ágil encontrados na literatura, tendo como base o estudo teórico previamente realizado, com o intuito de identificar suas principais deficiências e aspectos positivos;
- Definição de requisitos funcionais, a partir das características almejadas para jogos educativos e das carências observadas nas etapas anteriores, com o intuito de embasar a definição da arquitetura de software definida neste trabalho;

- Definição de um diagrama arquitetural para jogos educativos voltados ao ensino das metodologias ágeis Scrum e XP, contemplando sua decomposição modular;
- Prototipação de um jogo educativo, embasado na arquitetura definida, de modo a comprovar sua aplicabilidade e permitir a avaliação da qualidade dos resultados obtidos;
- Avaliação e correção do protótipo, se necessário, com base na obtenção de *feedbacks* de alunos de graduação e pós-graduação em Computação, bem como de docentes da área, após manipulação do protótipo por tempo determinado e devolução de questionário de avaliação devidamente preenchido.

Figura 1. Fluxograma do Trabalho



Fonte: Elaborado pelo autor

1.6. Organização da Monografia

Nesta seção é apresentada uma visão geral sobre os demais capítulos contidos no presente trabalho, visando cumprir os requisitos estabelecidos para a disciplina Trabalho de Conclusão de Curso.

No Capítulo 2 – Revisão Bibliográfica está contemplada a fundamentação teórica necessária para a compreensão deste trabalho. Desse modo, neste capítulo são apresentados os conceitos de metodologias ágeis, com foco no Scrum e XP, e as revisões sistemáticas da literatura conduzidas para identificar o estado da arte acerca do tema deste trabalho.

No Capítulo 3 – Scrum XPerience os desenvolvimentos da arquitetura de software e do protótipo de jogo educativo são detalhados. Para o diagrama arquitetural, sua decomposição modular é apresentada, descrevendo o funcionamento de cada módulo definido. Em relação ao protótipo, o desenvolvimento é apresentado seguindo a sequência de implementação, de modo a contemplar todos os módulos da arquitetura, além de imagens que ilustram as telas do Scrum XPerience em execução.

No Capítulo 4 – Avaliação do Protótipo as avaliações feitas por alunos de graduação, pós-graduação e docentes da área de Engenharia de Software são apresentadas, destacando os resultados e as alterações efetivadas com base nos apontamentos dos avaliadores.

No Capítulo 5 – Conclusão as considerações finais do trabalho são descritas, destacando suas principais contribuições e os trabalhos futuros identificados.

Capítulo 2 – Revisão Bibliográfica

2.1. Metodologias Ágeis

A ideia de que um software de qualidade era produzido somente a partir de um planejamento detalhado, de forma a segui-lo rigorosamente, era comum entre as décadas de 1980 e 1990 (SOMMERVILLE, 2011). Com o passar dos anos, novas técnicas e ideologias surgiram, dando origem às metodologias de desenvolvimento denominadas ágeis, as quais foram propostas com o intuito de solucionar os problemas encontrados nas metodologias tradicionais, como a má comunicação entre cliente e desenvolvedores, por exemplo (PRESSMAN, 2011).

Assim, os métodos ágeis ganharam maior destaque e, conseqüentemente, tornaram-se objeto de estudos para seu aprimoramento. Uma das contribuições mais importantes foi o Manifesto Ágil, o qual define um conjunto de princípios que devem ser englobados pelas metodologias ágeis (BECK *et al.*, 2001). Sommerville (2011) destaca um conjunto de cinco princípios ágeis, os quais podem ser observados na Tabela 1.

Tabela 1. Princípios Ágeis

Princípio	Descrição
Envolvimento do cliente	Deve haver envolvimento do cliente no desenvolvimento do software, promovendo novas funções e <i>feedbacks</i> sobre as iterações.
Entrega incremental	O software deve ser desenvolvido em incrementos, contando com a participação do cliente.
Pessoas, não processos	Os integrantes da equipe devem possuir a liberdade para utilizar seus próprios modos de trabalhar, sem seguir modelos prescritivos.
Aceitação de mudanças	O planejamento do software deve ser feito de modo a esperar que ocorram mudanças durante seu desenvolvimento.
Simplicidade	Tornar o software simples sempre que possível.

Fonte: Elaborado pelo autor

Diversas metodologias ágeis foram propostas como forma de desenvolver soluções computacionais com qualidade, no menor tempo possível. As metodologias que se destacaram foram a XP e o Scrum.

A respeito da XP, tem-se a definição de um conjunto de valores, os quais conduzem as atividades, ações e tarefas dessa metodologia (PRESSMAN, 2011). Tais valores são: comunicação, simplicidade, *feedback*, coragem e respeito. Ademais, a XP faz uso do paradigma de orientação a objetos e define um conjunto de princípios e etapas (PRESSMAN, 2011; SOMMERVILLE, 2011; BECK; GAMMA, 2000; WILDT *et al.*, 2015), como pode ser observado na Tabela 2.

O Scrum, por sua vez, é outra metodologia ágil que utiliza princípios do manifesto ágil para realizar suas atividades (PRESSMAN, 2011). Os criadores do Scrum apontam em seu guia oficial (SUTHERLAND; SCHWABER, 2017) cinco valores fundamentais para o Scrum, sendo eles: coragem para conduzir o trabalho de forma correta e enfrentar problemas; ser aberto ao trabalho; ter comprometimento para atingir as metas; ter foco, tanto no trabalho quanto nos objetivos; e respeitar os demais integrantes da equipe.

Tabela 2. Princípios e Etapas da XP

Princípio	Descrição
Planejamento	Essa etapa se inicia com a definição dos requisitos do sistema. Para tanto, é necessário um conjunto de histórias de usuários, as quais descrevem os objetivos e comportamentos esperados para o software. É atribuído um valor para cada história, o qual identifica sua importância. A equipe, por sua vez, atribui um custo para cada história. Além disso, é feita uma estimativa de prazo para o desenvolvimento de cada história e caso esse tempo ultrapasse três semanas, a história é dividida. Ademais, a velocidade do desenvolvimento do projeto é calculada após a entrega do primeiro <i>release</i> .
Projeto	O desenvolvimento do software segue o princípio <i>Keep It Simple</i> , o que indica que é desejado um sistema simplificado, atendendo as necessidades imediatas.
Refatoração	A equipe deve manter o código do sistema simples e manutenível, realizando melhorias sempre que possível.
Programação em Pares	Para o desenvolvimento do sistema, os desenvolvedores são divididos em pares, de modo a auxiliar e analisar as implementações uns dos outros.
Testes	A XP identifica a importância de testes de unidades, os quais buscam validar a implementação de funções, e de testes de aceitação, os quais são propostos pelo cliente e se referem ao software inteiro.
Etapa	Descrição
Pequenos Releases	O software é construído em incrementos. Desse modo, sua primeira versão consiste em um sistema com os requisitos essenciais. Em seguida, são realizados <i>releases</i> , de forma a evoluir o software a ser produzido.
Desenvolvimento <i>Test-First</i>	É importante definir e elaborar os testes de uma nova funcionalidade antes de realizar sua implementação.
Propriedade Coletiva	Não há ilhas de expertise na XP. Desse modo, toda a equipe se responsabiliza pelo software inteiro. Além disso, todos os desenvolvedores podem alterar qualquer parte do sistema.
Integração Contínua	Ao finalizar um trabalho, ele é integrado ao software, passando pelos testes necessários.
Ritmo Sustentável	A equipe deve buscar manter um ritmo de constante qualidade de produção. Portanto, horas-extras não são consideradas eficazes, sendo desencorajadas na XP.
Cliente no Local	Ao menos um representante do usuário final do software a ser desenvolvido deve participar da equipe XP. Com isso, ele será responsável por identificar novos requisitos, bem como alterações e remoções de funções.

Fonte: Elaborado pelo autor

A equipe, também conhecida como Time Scrum, deve se auto-organizar e é composta por três entidades (SUTHERLAND; SCHWABER, 2017), conforme apresentado na Tabela 3.

Tabela 3. Entidades do Time Scrum

Entidade	Descrição
Product Owner	Responsável por atribuir e garantir o máximo de valor possível para o produto a ser desenvolvido. Desse modo, o Product Owner gerencia os requisitos do sistema, de forma a classificá-los por sua importância, resultando no Backlog do Produto. Apenas uma pessoa é prevista para este cargo.
Time de Desenvolvimento	Consiste na equipe que irá desenvolver os incrementos e, consequentemente, o software. O Time de Desenvolvimento deve ser capaz de gerenciar e organizar seu trabalho, além de ser multifuncional, não possuir títulos, não ser subdividido e ser responsável por todo o software. As equipes devem ser compostas por três a nove integrantes, porém há a possibilidade de relacionamento com outras equipes durante o desenvolvimento de um sistema.
Scrum Master	Responsável por garantir o entendimento dos conceitos, valores e práticas do Scrum por toda a equipe. Ademais, o Scrum Master organiza as reuniões diárias, gerencia o que deve ser realizado em um incremento, registra as decisões, realiza a medição do progresso total do desenvolvimento e realiza a comunicação com a gerência e clientes (SOMMERVILLE, 2011). Apenas uma pessoa é prevista para esse cargo.

Fonte: Elaborado pelo autor

A duração das etapas do Scrum é predeterminada, não ultrapassando quatro semanas e não sendo inferior à duas. O conjunto de etapas é chamado de Sprint, a qual é composta, segundo Sutherland; Schwaber (2017), por cinco etapas, como pode ser observado na Tabela 4.

É importante observar que durante uma Sprint não ocorrem alterações de alto risco, possibilitando que os desenvolvedores mantenham seu foco no que está sendo implementado.

A partir desses estudos sobre as metodologias XP e Scrum, foram observadas semelhanças entre elas, como o planejamento inicial, entrega frequente de incrementos e a responsabilização do software todo pela equipe. Além disso, existem alguns pontos parecidos entre essas metodologias, mas que possuem uma execução divergente, como o incentivo da participação do cliente no desenvolvimento do software, a realização de testes e as reuniões da equipe.

Tabela 4. Etapas da Sprint

Etapas	Descrição
Planejamento	Define quais requisitos do Backlog do Produto serão realizados na Sprint, o que resulta no Backlog da Sprint. Além disso, é definido o prazo de duração da Sprint.
Reuniões Diárias	Também conhecidas como <i>daily scrum</i> , são reuniões de aproximadamente 15 minutos, feita de forma oral e informal, as quais servem para planejar o trabalho daquele respectivo dia. Além disso, todos os membros devem responder a três questões (PRESSMAN, 2011): “O que você realizou desde a última reunião?”, “O que você planeja realizar até a próxima reunião?”, e “Quais obstáculos você está encontrando para atingir a meta?”.
Desenvolvimento	Consiste na produção dos incrementos do sistema, de forma a construir o software.
Revisão e Retrospectiva da Sprint	Avalia quais metas foram cumpridas e o modo como isso foi feito. Assim, identificam-se possíveis melhorias para as próximas Sprints, tanto para sua condução quanto para o software a ser desenvolvido.

Fonte: Elaborado pelo autor

2.2. Características Almejadas para Softwares Educativos

A falta de dinâmica nos métodos tradicionais de ensino da Engenharia de Software pode resultar na ineficiência de aprendizagem dessa área (ALHAMMAD; MORENO, 2018; KOSA *et al.*, 2016). Como solução para esse problema, surgiram propostas de softwares educativos, com ênfase na elaboração de jogos (CALDERÓN *et al.*, 2017; MOLLÉRI *et al.*, 2018; BEPPE *et al.*, 2018).

Assim, o estudo da qualidade de tais softwares se mostrou necessário para definir funções e características almejadas para softwares educativos, maximizando o processo de ensino-aprendizagem. Desse modo, foram encontrados quatro trabalhos, descritos a seguir, que estudam e definem características para softwares educativos, as quais serviram como base para a análise de alguns jogos educativos realizada neste trabalho.

Almeida; Almeida (2015) realizam um detalhamento sobre o desenvolvimento e avaliação de softwares educativos de diversos tipos, incluindo jogos. Dessa forma, os autores definem os aspectos pedagógicos a serem considerados por jogos educativos, bem como descrevem pontos a serem considerados em seu desenvolvimento. Ademais, com os métodos de avaliação contidos neste livro, foi possível definir um conjunto de critérios avaliativos para conduzir, no presente trabalho, uma análise de jogos existentes. Dentre as características almejadas para um jogo educativos, os autores destacam: usar recursos motivacionais; possuir uma sequência de aprendizagem; apresentar erros de forma clara; apresentar os resultados do aluno, bem como seu desempenho; ser adaptável ao nível do jogador; e fornecer *feedback* ao aluno.

O estudo de Petri *et al.* (2020), por sua vez, trata de um aprimoramento do modelo de avaliação de jogos educativos para ensino de Engenharia de Software denominado MEEGA e proposto por Savi *et al.* (2011). O MEEGA busca avaliar diversos aspectos do jogo por meio de um questionário avaliativo, o qual é respondido pelos alunos-jogadores. Desse modo, Petri *et al.* (2020) propõe o MEEGA+, que evolui o MEEGA realizando uma junção com outros questionários avaliativos encontrados na literatura. Na Tabela 5, são identificadas as dimensões contempladas pelo MEEGA+, as quais também foram utilizadas para a definição dos critérios avaliativos no presente trabalho.

Tabela 5. Tabela de Dimensões

Dimensão	Definição
Diversão	O jogo produz sensação de prazer e felicidade, sendo uma distração.
Desafio	O jogo propõe obstáculos e situações que devem desafiar o aluno de acordo com sua competência, de modo a manter a atenção do jogador.
Satisfação	O aluno sente que uma determinada tarefa foi realizada devido ao seu esforço.
Operabilidade	O jogo é de fácil manuseio, possuindo comandos claros e intuitivos.

Fonte: Adaptado de Petri *et al.*, 2020

Henrique (2016) propõe requisitos de software voltados para softwares educativos, classificando-os em três tipos: pedagógicos, gerais e de usabilidade. Tais

requisitos foram utilizados tanto para a definição dos critérios avaliativos quanto para a elaboração do modelo arquitetural no presente trabalho.

Por fim, Yuxuan (2020) estabelece diretrizes voltadas para o desenvolvimento de softwares educativos, enfatizando a motivação do usuário. Para isso, a autora fez uso de atributos e métodos, os quais devem ser considerados na avaliação da motivação do usuário. O presente trabalho utiliza três métodos motivacionais apontados por Yuxuan (2020), para a definição dos critérios avaliativos, dada a relação entre tais métodos e as demais características citadas anteriormente (ALMEIDA; ALMEIDA, 2015). Tais métodos são apresentados na Tabela 6.

Tabela 6. Métodos Motivacionais

Método	Descrição
Envolvimento-do-eu	O aluno sente-se envolvido com o software ao receber um novo desafio. Além disso, em seu íntimo, o fracasso na execução de tal desafio causará perda de autoestima.
Competição	Há motivação nos alunos ao superar seus adversários ou a si mesmos.
Interação social	Deve haver comunicação e interação entre os alunos, o que os torna capazes de executar determinadas atividades.

Fonte: Elaborado pelo autor

2.3. Jogos Educativos para o Ensino de Engenharia de Software

Como forma de evidenciar o estado da arte acerca do uso de softwares educativos para o ensino de Engenharia de Software, foi realizada uma revisão sistemática da literatura. Desse modo, a revisão sistemática se iniciou com a definição dos objetivos: obter informações a respeito de softwares educativos, incluindo jogos, voltados ao ensino de Engenharia de Software, bem como os tópicos abordados em tais soluções. Além disso, tal revisão sistemática buscou dados a respeito de arquitetura de software voltada para jogos educativos.

A partir da definição dos objetivos, foram elaboradas cinco questões de pesquisa (QP), as quais são apresentadas na Tabela 7.

Tabela 7. Questões de Pesquisa – Jogos Educativos

Identificador	Descrição
QP1	Quais são os principais tópicos da Engenharia de Software que os softwares educativos abrangem?
QP2	Os jogos educativos são usados para o ensino de tópicos da Engenharia de Software?
QP3	A abordagem de desenvolvimento ágil é considerada pelos softwares educativos?
QP4	Como são aplicados os conceitos de arquitetura de software para o desenvolvimento de jogo educativo?
QP5	Os jogos educativos sobre Engenharia de Software trazem melhorias para o processo de ensino-aprendizagem?

Fonte: Elaborado pelo autor

Os estudos foram identificados por meio de *strings* de busca, estabelecidas tanto na língua portuguesa quanto inglesa, listadas na Tabela 8.

Na sequência, foram aplicados critérios de inclusão e exclusão nos estudos encontrados a partir das *strings* de busca, os quais são apresentados na Tabela 9. Tal aplicação resultou na identificação de 53 estudos finais que foram analisados por completo. A seguir, encontram-se os resultados obtidos para cada QP definida.

Tabela 8. Strings de Busca – Jogos Educativos

Português	Inglês
(jogo OR simulação OR sistema OR software) AND (educação OR ensino OR aprendizagem) AND (“engenharia de software” OR “metodologia ágil” OR “abordagem ágil” OR “desenvolvimento ágil”)	(game OR simulation OR system OR software) AND (educational OR teaching OR learning) AND (“software engineering” OR “agile methodology” OR “agile approach” OR “agile development”)
(uso OR aplicação OR conceitos OR definição) AND (arquitetura de software) AND (“desenvolvimento ágil” OR “jogo educativo”)	(use OR application OR concepts OR definition) AND (software architecture) AND (“agile development” OR “educational game”)

Fonte: Elaborado pelo autor

Tabela 9. Critérios de Inclusão e Exclusão – Jogos Educativos

Critérios de Inclusão	Critérios de Exclusão
• O estudo propõe algum software educativo para ensino de tópico(s) da Engenharia de Software; • O estudo utiliza conceitos de arquitetura de software para jogo(s) educativo(s) • O estudo é uma proposta de software educativo para ensino de metodologia(s) ágil(eis); • O estudo detalha os processos realizados no desenvolvimento do software educativo, contemplando principalmente o modo como o tema a ser ensinado é abordado; • O estudo é um estudo de caso, análise, revisão ou mapeamento sistemático sobre jogos educativos na Engenharia de Software; • O estudo possui data de publicação superior ou igual a 2016. Caso não possua, tem pelo menos 15 citações na sua fonte de busca.	• O estudo diz respeito a um software educativo não relacionado com a Engenharia de Software; • O estudo mostra conceitos de arquitetura de software que não dizem respeito a softwares educativos; • O estudo não está em português ou em inglês; • O estudo não retrata como foi feita a abordagem do tema escolhido; • O estudo trata de gamificação em softwares ou atividades que não possuem a finalidade de ensinar algum tópico da Engenharia de Software; • O estudo possui data de publicação anterior a 2016 e não possui, pelo menos, 15 citações.

Fonte: Elaborado pelo autor

Para a QP1, foi notado que os principais tópicos abordados pelos softwares educativos voltados ao ensino de Engenharia de Software são: testes de software; gerenciamento de projetos; engenharia de requisitos; e metodologias ágeis, os quais dão maior ênfase ao Scrum (CASTRO *et al.*, 2017; AMMONS; BANSAL, 2017; BRITO, 2016; SOUZA *et al.*, 2017; WANGENHEIM *et al.*, 2013; LEE, 2016). Foi possível observar que os trabalhos encontrados realizam o ensino de forma genérica para os tópicos de metodologias ágeis e gerenciamento de projetos, resultando em um ensino incompleto.

Em relação à QP2, é possível observar que os jogos educativos são utilizados para o ensino de Engenharia de Software, contemplando os mesmos tópicos destacados na QP1. Além disso, nota-se que os jogos educativos são divididos entre digitais, os quais utilizam computadores para sua execução, e não digitais, os quais utilizam cartas e tabuleiros, por exemplo, sendo ambas as vertentes utilizadas para o ensino de Engenharia

de Software (JÁCOME, 2013; BEPPE *et al.*, 2018; FARIAS *et al.*, 2012; NAVARRO; HOEK, 2004). Foi possível observar que os jogos digitais, ao contrário dos não digitais, não fornecem um aprendizado em grupo, sendo todos feitos para apenas um jogador por máquina.

Sobre a QP3, observa-se que o desenvolvimento ágil é contemplado pelos softwares educativos encontrados, incluindo os jogos educativos. Além disso, nota-se que dentre os oito jogos educativos voltados ao ensino ágil encontrados, apenas dois eram voltados a metodologia XP (JÁCOME, 2013; NAVARRO; HOEK, 2004), enquanto os seis restantes eram voltados ao Scrum (BRITO, 2016; CASTRO *et al.*, 2017; AMMONS; BANSAL, 2017; SOUZA *et al.*, 2017; WANGENHEIM *et al.*, 2013; LEE, 2016). Não foram encontrados estudos relacionados a proposta de softwares educativos para ensino de outras metodologias ágeis.

A respeito da QP4, os conceitos de arquitetura de software para a produção de jogos educativos são poucos utilizados. Dos 53 trabalhos encontrados, apenas quatro se referiam a esse tópico, sendo eles: Rodríguez-Silva *et al.* (2015), Vegt *et al.* (2016), Stavrev *et al.* (2018) e Skalka; Drlík (2018). Apesar da carência de estudos a respeito deste tópico, todos os trabalhos encontrados atendem as características almejadas para jogos educativos, além de facilitar o desenvolvimento do jogo.

Por fim, em relação a QP5, foi possível observar que os jogos educativos voltados ao ensino de Engenharia de Software se mostraram eficazes para o ensino dos tópicos dessa disciplina (FERNANDES; WERNER, 2009; KOSA *et al.*, 2016). Para medir tal eficiência foram utilizados questionários avaliativos existentes, como o MEEGA (SAVI *et al.*, 2011), ou de produção autoral embasado nos existentes (SANTOS *et al.*, 2019; HODA, 2019). Estudos como os de Brito (2016) e García *et al.* (2020) realizaram também a avaliação do conteúdo que foi ensinado, auxiliando na comprovação da qualidade dos resultados.

Desse modo, foi possível observar que os jogos educativos voltados para o ensino de Engenharia de Software contemplam as metodologias ágeis, com maior foco no Scrum. Assim, existe a necessidade de elaboração de novos trabalhos relacionados às outras metodologias, promovendo maior conhecimento aos alunos. Além disso, essa revisão sistemática colaborou para a definição dos critérios avaliativos, utilizados para a análise dos jogos encontrados, bem como para a definição dos requisitos funcionais que embasaram a estruturação do diagrama arquitetural proposto no presente trabalho.

2.4. Protótipos de Jogos Educativos

Seguindo o estudo teórico que embasou o desenvolvimento deste trabalho, foi conduzida uma segunda revisão sistemática da literatura, a qual teve como objetivo analisar o estado da arte acerca do uso de protótipos para o desenvolvimento de jogos educativos, não necessariamente voltados para o ensino de Engenharia de Software. Ademais, esta revisão sistemática buscou informações sobre o modo como a prototipagem de jogos educativos é realizada. Assim, foram definidas três QP, as quais são apresentadas na Tabela 10.

Tabela 10. Questões de Pesquisa – Protótipos de Jogos Educativos

Identificador	Descrição
QP1	Os protótipos de software são utilizados no desenvolvimento de jogos educativos voltados ao ensino de tópicos da Engenharia de Software?
QP2	Existem protótipos de jogos educativos voltados ao ensino de metodologias ágeis?
QP3	Como são aplicados os conceitos de prototipagem no desenvolvimento de jogos educativos?

Fonte: Elaborado pelo autor

Com a definição dos objetivos e das questões de pesquisa, foi possível determinar um conjunto de *strings* de busca, tanto para o idioma inglês quanto para o português, as quais foram utilizadas para a identificação dos estudos. Na Tabela 11 são apresentadas as *strings* de busca usadas no presente trabalho.

Tabela 11. Strings de Busca – Protótipos de Jogos Educativos

Português	Inglês
(jogo) AND (educativo OR aprendizagem OR ensino) AND (“engenharia de software” OR “metodologias ágeis” OR scrum OR “extreme programming”)	(game) AND (educational OR learning OR teaching) AND (“software engineering” OR “agile methodology” OR scrum OR “extreme programming”)
(utilização OR aplicação OR proposta) AND (“protótipo de software educativo” OR “protótipo de jogo educativo”)	(use OR application OR proposal) AND (“educational software prototype” OR “educational game prototype”)

Fonte: Elaborado pelo autor

A partir dos estudos encontrados e após a aplicação dos critérios de inclusão e exclusão apresentados na Tabela 12, foi iniciada a etapa de leitura e seleção dos 50 trabalhos selecionados, obtendo as informações necessárias para responder cada QP definida.

Tabela 12. Critérios de Inclusão e Exclusão – Protótipos de Jogos Educativos

Critérios de Inclusão	Critérios de Exclusão
• O estudo apresenta um protótipo de jogo educativo voltado para o ensino de algum tópico da Engenharia de Software; • O estudo detalha o processo de prototipagem e/ou desenvolvimento de jogo educativo; • O estudo é um estudo de caso, mapeamento ou revisão sistemática da literatura sobre prototipagem de jogos educativos; • O estudo atende, pelo menos, um dos critérios anteriores e possui data de publicação igual ou superior a 2016 e/ou possui, pelo menos, 20 citações.	• O estudo não apresenta um protótipo de software para jogo educativo; • O estudo não detalha o processo de prototipagem e/ou desenvolvimento de um jogo educativo; • O estudo não promove informações sobre prototipagem de jogos educativos; • O estudo não está na língua inglesa ou portuguesa; • O estudo possui data de publicação anterior a 2016 e não possui, pelo menos, 20 citações.

Fonte: Elaborado pelo autor

Em relação à QP1, foi possível observar que existe o uso de protótipos no desenvolvimento de jogos educativos, como pode ser notado nos trabalhos de Gkritsi (2011), Maxim *et al.* (2016), Brito; Vieira (2017), Molléri *et al.* (2018) e Ünlü *et al.* (2020). Além disso, foi observado que os protótipos são apresentados em conjunto com a implementação do jogo, o que indica que são criados protótipos evolutivos de alta fidelidade.

Sobre a QP2, foi notado o uso de protótipos para jogos educativos voltados ao ensino de metodologias ágeis, com foco na metodologia Scrum (BRITO; VIEIRA, 2017; GKRTSI, 2011). No entanto, não foram encontrados trabalhos referentes ao uso de protótipos para a elaboração de jogos educativos voltados para outras metodologias ágeis.

Por fim, a respeito da QP3, observou-se que o uso de protótipos de software no desenvolvimento de jogos é considerado eficiente, auxiliando nas etapas de produção do jogo, principalmente nos testes e avaliações. Segundo Filho *et al.* (2013), os protótipos

são imprescindíveis para o desenvolvimento de jogos, dado que eles facilitam o apontamento de erros. Além disso, foi encontrado um modelo para a elaboração de protótipos de jogos, não somente educativos, denominado XS-Game (CONFESSOR, 2019), proposto a partir de adaptações nos modelos existentes.

Com isso, observou-se a relevância da prototipagem para a produção de jogos no geral, incluindo os jogos educativos voltados ao ensino de Engenharia de Software e seus tópicos. Também foi notado que os protótipos utilizados nos jogos educativos sobre ensino ágil contemplam apenas a metodologia Scrum. Desse modo, essa revisão sistemática contribuiu para ressaltar a importância do desenvolvimento de um protótipo de jogo educativo voltado para o ensino das metodologias ágeis Scrum e XP, utilizando um diagrama arquitetural como base.

2.5. Análise Empírica de Jogos Educativos para Engenharia de Software

Com a condução da primeira revisão sistemática, referente ao uso de jogos educativos para Engenharia de Software, foram identificados oito jogos educativos voltados para o ensino ágil. Dentre eles, seis são referentes ao Scrum e dois voltados à XP, conforme segue:

- **Scrumify** (AMMONS; BANSAL, 2017) – Jogo digital, feito para apenas um jogador. O objetivo do jogo é, por meio do personagem controlável Scrum Knight, chegar ao final de todas as fases, obtendo informações a respeito do Scrum. O jogo requisita do jogador o conhecimento obtido para avançar nas fases. O jogo termina após as quatro fases serem finalizadas;
- **Scrumi** (SOUZA *et al.*, 2017) – Jogo de tabuleiro digital, no qual o jogador deve completar todo o percurso para vencer. Para avançar no tabuleiro, o jogador deve responder corretamente aos questionários a respeito do Scrum. O jogador vence quando chegar ao fim do trajeto;
- **Scrum-X** (LEE, 2016) – Jogo digital, feito para apenas um jogador. Esse jogo simula um ambiente de gerenciamento de projeto, fazendo o uso da metodologia Scrum. Assim, em cada etapa, o jogador deve realizar suas escolhas da melhor forma possível, podendo verificar seus impactos no desenvolvimento do software fictício. O jogo é encerrado quando o software fictício for entregue ou quando o tempo se esgotar;

- **2TScrum** (BRITO, 2016) – Jogo de tabuleiro, feito para um ou mais jogadores. Assim, o objetivo é percorrer todo o trajeto, passando por todas as etapas do Scrum. O jogo finaliza quando um jogador, ou equipe, cumprir todos os itens do Backlog do Produto definido no início do jogo;
- **Agility Scrum** (CASTRO *et al.*, 2017) – Jogo para grupos de quatro a seis alunos. O objetivo do jogo é elaborar um circuito digital no menor tempo possível, utilizando a metodologia Scrum. Vence o grupo que finalizar o circuito no menor tempo, passando por todas as etapas do Scrum;
- **Scrumia** (WANGENHEIM *et al.*, 2013) – Jogo multijogador, em que a equipe deve elaborar origamis utilizando a metodologia Scrum. Vence a equipe que atingir o número requisitado de origamis em menos tempo;
- **SimSE** (NAVARRO; HOCK, 2004) – Jogo digital para um jogador. O objetivo do jogo é simular um ambiente de desenvolvimento XP, em que o jogador possua o papel de gerente de projetos. O jogo termina quando o produto fictício for entregue, de acordo com as etapas e princípios da XP;
- **Softbook** (JÁCOME, 2013) – Livro-jogo, feito para um jogador. O jogador deve trilhar os caminhos no livro de modo a seguir os princípios e etapas da metodologia XP. O jogo termina quando o aluno chegar ao final do desenvolvimento do software fictício.

Com isso, foi possível analisar os jogos encontrados, de modo a identificar carências e aspectos positivos das propostas existentes. Na Tabela 13 é ilustrado o resultado dessa avaliação.

Tabela 13. Aplicação dos Critérios Avaliativos

Metodologia	Formato	Jogo	Critérios Avaliativos							
			Princípios Ágeis	Principais Etapas	Uso sem Treinamento	Motivação	Gerenciamento de Erros e Feedback	Divertido	Progressão	Avaliação
Scrum	Digital	Scrumify								
		Scrumi								
		Scrum-X								
	Não digital	2TScrum								
		Agility Scrum								
		Scrumia								
XP	Digital	SimSE								
	Não digital	Softbook								

Legenda: ■ Atende ■ Não atende

Fonte: Elaborado pelo autor

Com a aplicação dos critérios avaliativos, foi observado que nenhum dos jogos atendeu todos os requisitos. No entanto, nota-se que todos eles atendem ao menos quatro dos oito critérios utilizados para análise. Assim, todos os jogos englobam os princípios ágeis, além das principais etapas da metodologia abordada. Todos eles também motivam os alunos a aprenderem o conteúdo ensinado, além de serem considerados divertidos.

Entretanto, apenas três jogos educativos fazem uso de instruções simples, podendo ser utilizados sem treinamento prévio. Além disso, apenas três jogos fazem gerenciamento de erros e fornecem *feedbacks* aos alunos e quatro realizam a progressão da dificuldade e do conteúdo ensinado. Ademais, apenas dois dos oito jogos educativos fazem uma avaliação do conteúdo que está sendo ensinado. Também foi observado que os jogos educativos melhor avaliados são digitais, sendo eles o Scrumify e Scrumi. Com isso, é possível concluir que, apesar dos jogos educativos digitais serem eficientes para o processo de ensino-aprendizagem, existem algumas carências a serem sanadas.

2.6. Considerações Finais

Com o estudo teórico realizado neste capítulo, foi possível observar a importância da Engenharia de Software, especialmente as metodologias ágeis, para o desenvolvimento de software, bem como a necessidade de zelar pela qualidade da aprendizagem de seus conceitos. No entanto, dado que essa disciplina possui um vasto referencial teórico, seu ensino é dificultado pela desmotivação dos alunos, impactando no processo de ensino-aprendizagem. Assim, foram propostos diversos softwares educativos como solução para esse problema, principalmente jogos educativos. Além disso, dado o destaque que tais jogos educativos ganharam, diversos trabalhos surgiram para estudar e definir características e comportamentos almejados para tais softwares.

A partir da primeira revisão sistemática conduzida, foi observado que os jogos educativos contemplam, dentre outros tópicos, as metodologias ágeis, porém com maior ênfase ao Scrum. Ademais, essa revisão da literatura também indicou a carência de estudos relacionados à arquitetura de software voltada para jogos educativos, o que indica a relevância do diagrama arquitetural definido neste trabalho.

Após isso, outra revisão sistemática da literatura foi conduzida, com o objetivo de identificar o uso de protótipos para o desenvolvimento de jogos educativos, não apenas voltados para o ensino ágil. Assim, foi possível identificar que os protótipos utilizados são, em sua maioria, desenvolvidos com as mesmas ferramentas a serem utilizadas na

produção do jogo educativo, caracterizando um protótipo de alta fidelidade. Além disso, foi observado o uso de protótipos para a construção de jogos educativos sobre métodos ágeis.

Com base em tais estudos, foi possível avaliar os jogos educativos encontrados, voltados para o ensino de metodologias ágeis. Essa análise possibilitou a indicação de carências contidas nas soluções atuais, assim como seus aspectos em comum. A partir dessas carências, no Capítulo 3 é apresentado o conjunto de requisitos funcionais definido, bem como o diagrama arquitetural estabelecido e o protótipo desenvolvido, capaz de sanar tais carências.

Capítulo 3 – Scrum XPerience

3.1. Descrição Geral

A partir do levantamento teórico realizado anteriormente, o qual forneceu uma base de conhecimento sólida a respeito da importância e uso de jogos educativos para o ensino de Engenharia de Software, bem como as carências das soluções atuais, foi definido um conjunto de requisitos funcionais de modo a sanar os problemas encontrados.

Com base nos requisitos definidos, foi possível definir um diagrama arquitetural voltado para jogos educativos sobre ensino das metodologias ágeis Scrum e XP. Tal arquitetura tem o intuito de simplificar o desenvolvimento de novos softwares educativos do tipo jogo, além de promover uma visão de alto nível acerca do sistema a ser implementado.

A proposta da arquitetura de software para jogos educativos pode ser avaliada, confirmando sua qualidade e aplicabilidade, por meio de um protótipo de software. Assim, a prototipação de um jogo educativo voltado ao ensino de Scrum e XP foi desenvolvida, de modo a contemplar os módulos previstos pela arquitetura e, consequentemente, validar o modelo arquitetural por completo.

Com isso, espera-se que este trabalho contribua para o desenvolvimento de futuros jogos educativos, voltados ao ensino ágil, sanando as deficiências encontradas nas soluções atuais. Ademais, este trabalho busca avançar as pesquisas sobre o ensino de

metodologias ágeis, de modo a motivar novos estudos relacionados à outras metodologias além do Scrum e da XP.

3.2. Requisitos Funcionais

A partir do estudo teórico, foi obtido o conhecimento necessário para realizar o desenvolvimento deste trabalho. Para a elaboração da arquitetura de software, foi definido um conjunto de requisitos funcionais, divididos em sete categorias, em que cada categoria representa uma carência observada durante a análise literária. Na Tabela 14 são apresentados os requisitos definidos.

A seguir, encontram-se as especificações dos 21 requisitos identificados:

- RF1) Princípios ágeis: O jogo deve contemplar os princípios ágeis, visando ensiná-los ao aluno durante sua interação com o software. Além disso, o jogo deve contemplar as etapas previstas pelas metodologias XP e Scrum;
- RF2) Funções previstas pela metodologia: O jogo deve englobar as funções previstas pelas metodologias abordadas, permitindo ao aluno escolher qual deseja desempenhar;
- RF3) Comparativo entre metodologias ágeis: O jogo deve realizar um comparativo entre as diferentes metodologias englobadas, indicando as ações permitidas e os impactos de cada uma;
- RF4) Simular ambiente de desenvolvimento em grupo: O jogo deve possibilitar o trabalho em equipe para o desenvolvimento de um projeto fictício, com cada aluno do grupo desempenhando seu papel na metodologia escolhida. A partir disso, as ações tomadas pelos jogadores serão avaliadas, viabilizando a avaliação individual dos alunos;
- RF5) Escolha de metodologia: O jogo deve possibilitar a escolha da metodologia a ser abordada pelo grupo. Assim, antes do início da partida, o software deve apresentar aos jogadores as opções disponíveis, bem como suas principais características;

Tabela 14. Requisitos Funcionais

Abrangência no tratamento de metodologias ágeis	Mecanismo de gerenciamento de erros e fornecimento de <i>feedback</i>
RF1) Englobar os princípios ágeis, bem como as etapas das metodologias Scrum e XP; RF2) Contemplar os papéis previstos para os membros das equipes segundo cada metodologia ágil (e.g. Scrum Master, Product Owner).	RF11) Gerenciar os erros cometidos pelo aluno, apontando-os de forma clara e apresentando suas devidas soluções; RF12) Apresentar avaliação do progresso do aluno, de modo a revelar pontos a serem melhorados.
Comparativo entre duas ou mais metodologias ágeis	Avaliação sobre a efetividade do ensino do conteúdo
RF3) Permitir a realização de comparações entre as metodologias ágeis XP e Scrum para embasar a escolha pelos grupos da metodologia que adotarão em determinado projeto.	RF13) Realizar avaliação da aplicação dos conceitos ágeis, afim de verificar se eles estão sendo corretamente ensinados.
Interação social	Jogo multijogador
RF4) Simular o desenvolvimento de projetos em grupo, com os alunos desempenhando papéis previstos para as equipes ágeis; RF5) Permitir a escolha, pelos grupos de alunos, da metodologia de desenvolvimento a ser adotada para determinado projeto proposto pelo professor.	RF14) Exigir a criação de uma conta para acesso ao software, requerendo e validando dados pessoais, tanto do aluno quanto do professor, para o seu cadastramento no sistema; RF15) Ter um sistema de autenticação, facilitando a identificação dos usuários; RF16) Ter dois tipos de usuários, sendo o professor e o aluno-jogador; RF17) Possibilitar a edição dos dados do jogador;
Características de qualidade para jogos educativos	RF18) Permitir que o aluno decida, dentre um conjunto predeterminado de opções, qual ação deseja realizar no jogo; RF19) Limitar o número de jogadores por grupo, não contabilizando o professor, de forma que seja possível englobar os papéis previstos pelas metodologias; RF20) Estabelecer um conjunto de regras, claras e bem definidas, de modo a minimizar falhas no fluxo do jogo; RF21) Possibilitar ao administrador do sistema gerar e manter as bases de dados necessárias para o jogo.
RF6) Estimular o aluno-jogador por meio de desafios e rankings, com o intuito de promover competitividade; RF7) Fornecer um tutorial para ensinar a mecânica do jogo; RF8) Disponibilizar ajuda/diretrizes que possam ser acessadas a qualquer momento; RF9) Indicar a sequência de ensino que será seguida pelo jogo; RF10) Progredir no aprofundamento do conteúdo, sem desestimular o aluno-jogador.	

Fonte: Elaborado pelo autor

- RF6) Competitividade: O jogo deve fazer uso da competitividade entre os alunos, de modo a estimulá-los. Para tanto, o jogo deve designar pontuações para as equipes com base no desempenho da partida. A pontuação é atribuída quando o grupo realiza uma ação corretamente, de acordo com a metodologia aplicada. Nenhum ponto é aplicado caso a equipe cometa algum erro. A equipe que finalizar a partida com mais pontos terá a melhor classificação. O jogo deve manter um *ranking* com a classificação das equipes, o qual pode ser acessado a qualquer momento;
- RF7) Tutorial: O jogo deve ensinar sua mecânica ao aluno-jogador por meio de um tutorial, apresentando as regras e ações permitidas pelo software;
- RF8) Ajuda: O jogo deve disponibilizar uma opção de ajuda sempre que requisitado. Essa opção não deve apresentar dicas, explicações ou conceitos da partida, mas sim os comandos utilizados pelo jogo;
- RF9) Sequência de ensino: O jogo deve indicar a sequência de ensino adotada de forma clara ao aluno-jogador, o qual poderá perceber o avanço do ensino, facilitando o processo de ensino-aprendizagem;
- RF10) Aprofundamento do conteúdo: O jogo deve progredir o aprofundamento do conteúdo ensinado. É importante observar que essa progressão não deve ser feita abruptamente, de forma a não causar frustração e desmotivação aos alunos;
- RF11) Gerenciamento de erros: O jogo deve realizar o gerenciamento dos erros cometidos pelo aluno-jogador. Desse modo, os erros devem ser apontados de forma compreensível, apresentando suas respectivas soluções corretas. Os erros podem ser apresentados tanto em forma textual quanto em acontecimentos durante a partida;
- RF12) Avaliação do progresso do aluno: O jogo deve avaliar o aluno de modo a identificar seus pontos positivos e negativos, possibilitando identificar os tópicos que requerem maior atenção;
- RF13) Avaliação do ensino do conteúdo: O jogo deve avaliar a aplicação dos conceitos ágeis pelo grupo, de modo a verificar se o conteúdo foi ensinado de forma correta;

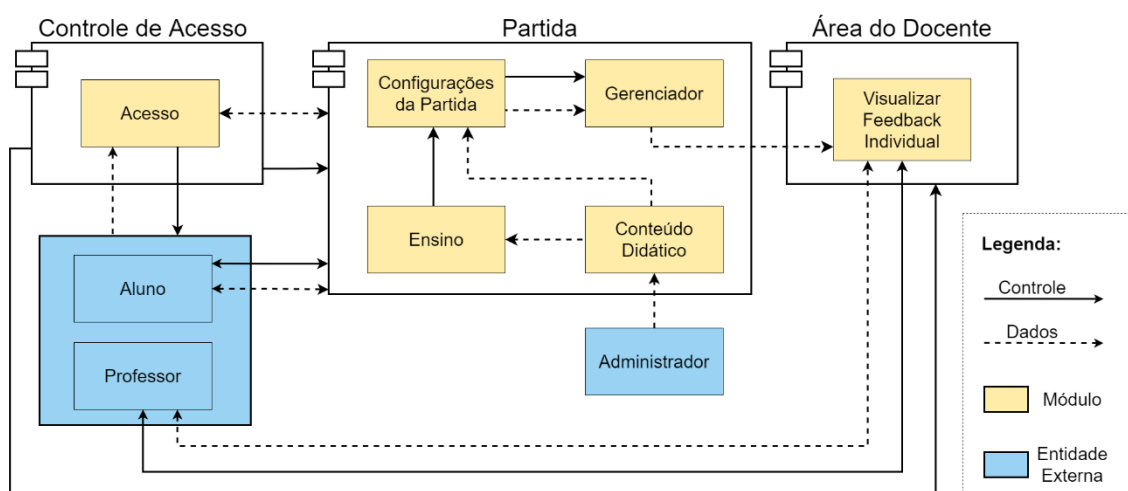
- RF14) Cadastro: O jogo deve requerer o cadastramento dos usuários, professor ou aluno, para que eles possam acessar o software. Para isso, é necessário o fornecimento de dados como nome de usuário, tipo de conta (aluno ou professor), e-mail e senha. É importante destacar que tais dados podem ser alterados, conforme necessário;
- RF15) Autenticação: O jogo deve realizar a autenticação dos usuários, de modo a identificá-los. Para isso, o usuário deve fornecer seu nome de usuário e senha ao sistema. O software, por sua vez, deve reconhecer se tal usuário é um aluno ou professor;
- RF16) Tipos de usuário: O jogo deve englobar dois tipos de contas, sendo uma para o aluno e outra para o professor. Os alunos podem decidir a metodologia a ser abordada pelo jogo, bem como suas funções. Já o professor pode acessar os erros e acertos individuais dos alunos;
- RF17) Edição de dados: O jogo deve permitir aos usuários editarem seus dados cadastrados. Para tanto, o usuário deve estar autenticado;
- RF18) Escolha de ações: O jogo deve permitir que os alunos decidam suas ações, dentre um conjunto predeterminado de opções. Cada ação deve ter um impacto na partida, o qual deve ser apresentado ao jogador;
- RF19) Limite de jogadores: O jogo deve limitar o número de jogadores por grupo. A quantidade deve ser suficiente para que todos os papéis previstos pelas metodologias sejam contemplados por ao menos um jogador. Esse limitante depende das diferentes implementações, sendo três o número definido para o protótipo desenvolvido no presente trabalho;
- RF20) Regras: O jogo deve possuir um conjunto de regras, as quais definem seu escopo. Essas regras devem ser bem definidas e claras, sendo entendida pelos jogadores;
- RF21) Administração: O jogo deve permitir ao administrador do sistema realizar o gerenciamento das bases de dados que serão apenas consultadas durante a partida. Assim, o administrador deve inserir informações como regras, comandos, ações possíveis e metodologias em suas respectivas bases.

3.3. Arquitetura de Software

A partir dos requisitos estabelecidos, foi possível elaborar a arquitetura de software, direcionada a jogos educativos sobre ensino de metodologias ágeis, especialmente o Scrum e a XP.

A partir da Figura 2, pode-se observar a definição de três componentes, sendo eles: Controle de Acesso, o qual controla o acesso dos usuários ao sistema, garantindo as devidas permissões para cada tipo de usuário; Partida, o qual é responsável por realizar o ensino das metodologias ágeis aos alunos por meio de uma simulação de produção de um software fictício, de acordo com o método escolhido; e Área do Docente, o qual se responsabiliza por disponibilizar os *feedbacks* individuais ao professor. Destaca-se a comunicação do componente Controle de Acesso com os demais componentes, dado que é necessária a autenticação do usuário para o uso do software.

Figura 2. Diagrama Arquitetural



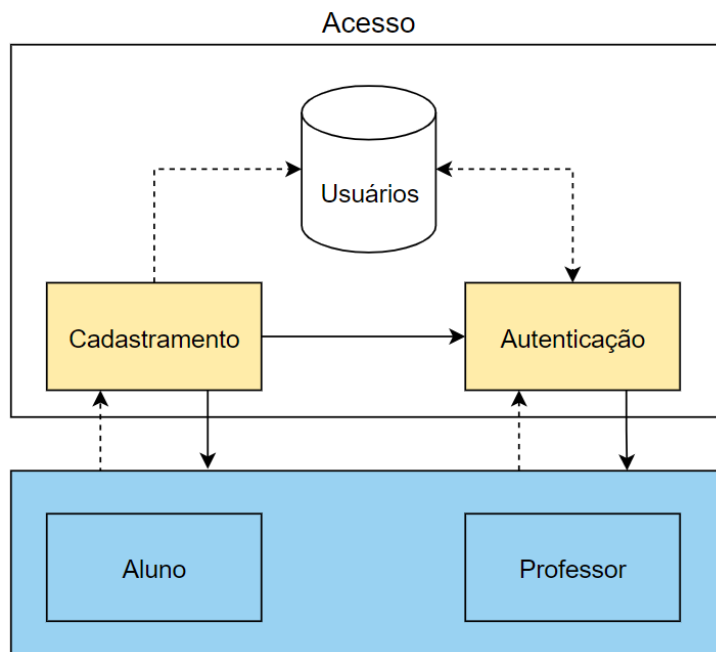
Fonte: Elaborado pelo autor

A decomposição modular de cada módulo apresentado no diagrama arquitetural é descrita nas subseções a seguir, com exceção do módulo Visualizar *Feedback Individual*, uma vez que ele se refere apenas a uma consulta de dados previamente armazenados, permitindo que o professor visualize o *feedback* individual de um determinado aluno, gerado durante uma partida. Portanto, este módulo atende aos requisitos RF11, RF12 e RF13, não sendo necessária sua decomposição.

3.3.1. Acesso

Este módulo garante acesso ao sistema apenas para usuários cadastrados. Portanto, atende aos requisitos RF14, RF15, RF16 e RF17. Na Figura 3 é apresentada a decomposição modular proposta para o módulo Acesso, a qual é detalhada na sequência.

Figura 3. Decomposição Modular – Acesso



Fonte: Elaborado pelo autor

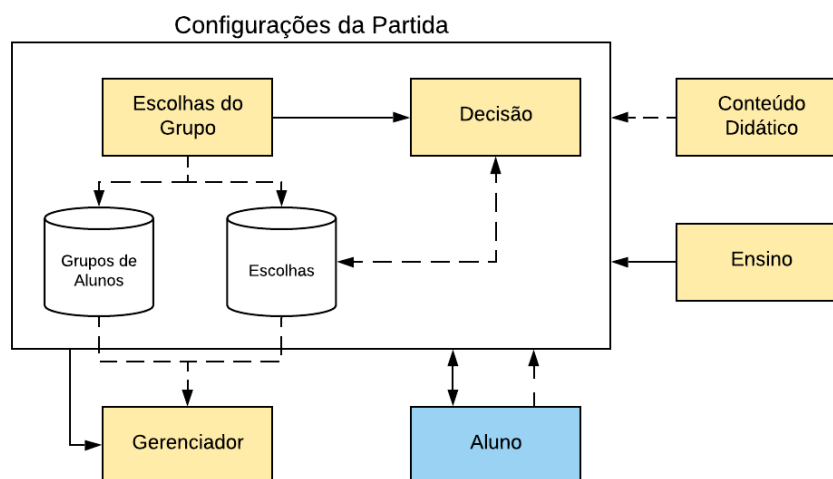
- **Cadastramento:** Este módulo realiza o cadastro de um novo usuário no sistema. Para isso, são utilizados os dados fornecidos pelo usuário, como nome de usuário, tipo de conta, e-mail e senha. Após o cadastramento, o usuário é autenticado ao sistema. Assim, este módulo se relaciona com a Autenticação, efetivando o acesso do usuário ao sistema;
- **Autenticação:** Este módulo realiza a autenticação do usuário no sistema, permitindo seu acesso ao jogo. Logo, com o nome de usuário e senha fornecidos pelo usuário, o software deve verificar se existe algum usuário com esses dados cadastrados para ser autenticado. Caso não exista, apresenta-se uma mensagem de erro. Este módulo também possibilita a edição dos dados do usuário.

3.3.2. Configurações da Partida

Este módulo faz o gerenciamento das escolhas dos jogadores. Assim, é permitido aos alunos-jogadores realizarem escolhas a respeito da metodologia a ser empregada,

assim como seus papéis previstos, que serão desempenhados pelos alunos. Ademais, este módulo limita o número de jogadores por grupo e apresenta um conjunto de decisões referentes ao papel escolhido pelo aluno. Portanto, este módulo atende aos requisitos RF1, RF2, RF4, RF5, RF10, RF18 e RF19. Na Figura 4 é apresentada a decomposição modular relacionada ao módulo Configurações da Partida, a qual é detalhada na sequência.

Figura 4. Decomposição Modular – Configurações da Partida



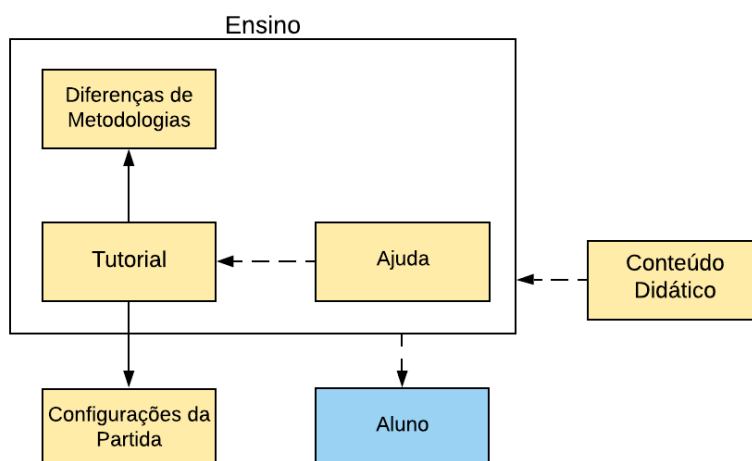
Fonte: Elaborado pelo autor

- **Escolhas do Grupo:** Este módulo é responsável por apresentar as opções de metodologias, Scrum e XP, apontando uma breve descrição do funcionamento de cada uma delas, tendo como base as informações disponibilizadas pelo módulo Conteúdo Didático. Em seguida, deve-se permitir a definição dos papéis relacionados à metodologia ágil escolhida. Assim, o grupo poderá escolher qual dos métodos e funções desejam adotar para a realização da partida. Ademais, este módulo limita o número de alunos por grupo, além de inserir os dados referentes às escolhas na base Escolhas e às informações do grupo em Grupos de Alunos;
- **Decisão:** Este módulo é responsável por apresentar as opções de decisões disponíveis para cada jogador e registrar a escolhida na base de dados Escolhas. Assim, essas opções terão como base a metodologia e papel escolhidos pelo aluno. Desse modo, o módulo Decisão recebe e envia dados para a base de dados Escolhas. Além disso, o módulo Decisão deve se relacionar com o módulo Escolhas do Grupo, uma vez que as opções disponíveis dependem do papel escolhido.

3.3.3. Ensino

Este módulo deve ensinar o aluno-jogador sobre os funcionamentos, regras e conceitos necessários para iniciar a partida. Portanto, este módulo atende aos requisitos RF1, RF2, RF3, RF7, RF8, RF9 e RF20. Na Figura 5 é ilustrada a decomposição modular referente ao módulo Ensino, a qual é detalhada na sequência.

Figura 5. Decomposição Modular – Ensino



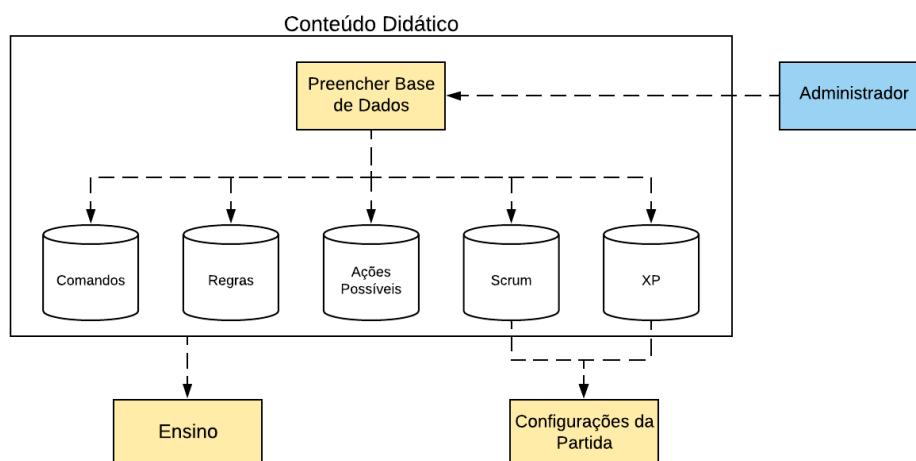
Fonte: Elaborado pelo autor

- **Tutorial:** Este módulo ensina o funcionamento do jogo para o aluno-jogador, destacando suas regras e as ações que o aluno pode realizar, com base nas informações disponibilizadas pelo módulo Conteúdo Didático. Além disso, há o relacionamento entre o módulo Tutorial e os módulos Configurações da Partida e Diferenças de Metodologias, uma vez que a partida deve contemplar as metodologias Scrum e XP, bem como suas diferenças e papéis previstos;
- **Diferenças de Metodologias:** Este módulo estabelece as diferenças entre as metodologias Scrum e XP, apresentando-as aos aluno-jogadores no decorrer da partida, a partir dos dados fornecidos pelo módulo Conteúdo Didático. Assim, o jogo promove a comparação entre ambas metodologias, de modo a auxiliar o aluno nas ações a serem tomadas;
- **Ajuda:** Este módulo disponibiliza ajuda ao jogador, sempre que solicitado, com base nos dados disponibilizados pelo módulo Conteúdo Didático. Assim, são apresentados, resumida e simplificada, os controles e opções do jogo. Além disso, a Ajuda deve fornecer dados para o módulo Tutorial, uma vez lá ocorre o ensino completo do funcionamento do jogo, contemplando seus comandos.

3.3.4. Conteúdo Didático

Este módulo é responsável por gerar as bases de dados consultadas durante a partida. Apenas a entidade externa Administrador pode gerar tais bases. Portanto, este módulo atende ao requisito RF21. Na Figura 6 é apresentada a decomposição modular proposta para o módulo Conteúdo Didático, a qual é detalhada na sequência.

Figura 6. Decomposição Modular – Conteúdo Didático



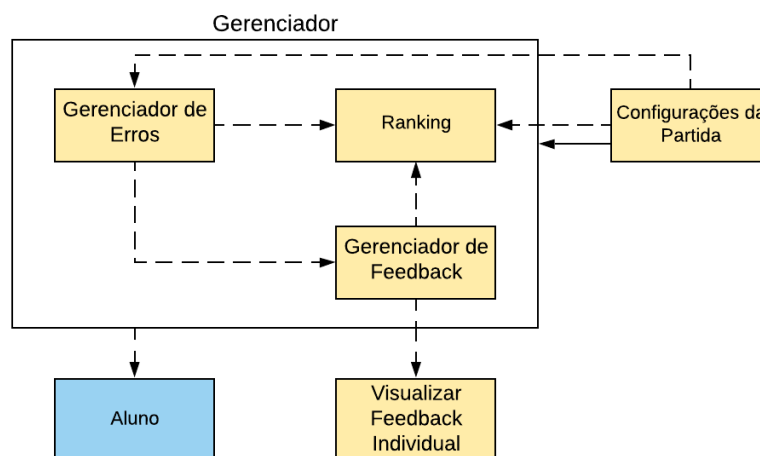
Fonte: Elaborado pelo autor

- **Preencher Base de Dados:** Este módulo é responsável apenas pelo armazenamento de dados nas bases de dados que são necessárias para o funcionamento da partida. Dessa forma, apenas a entidade externa Administrador tem acesso a este módulo. Destaca-se que o módulo Ensino utiliza informações de todas as bases de dados, enquanto o módulo Configurações da Partida utiliza apenas os dados das bases Scrum e XP.

3.3.5. Gerenciador

Este módulo é responsável por gerenciar erros, *feedbacks* e o ranking dos alunos. Desse modo, o Gerenciador deve identificar os erros e acertos dos jogadores, além de apresentar os aspectos positivos e negativos da partida. Este módulo também se responsabiliza por definir um *ranking* dos grupos, a partir das informações coletadas. Portanto, este módulo atende aos requisitos RF6, RF11, RF12, RF13 e RF16. Na Figura 7 é ilustrada a decomposição modular referente ao módulo Gerenciador, a qual é detalhada na sequência.

Figura 7. Decomposição Modular – Gerenciador



Fonte: Elaborado pelo autor

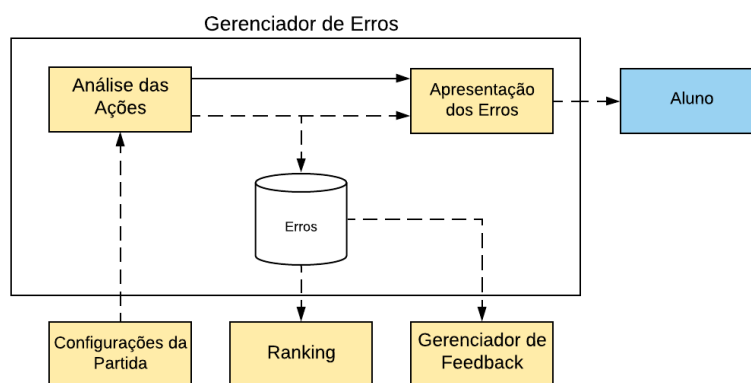
- **Gerenciador de Erros:** Este módulo faz o gerenciamento dos erros cometidos pelo jogador durante a partida. Assim, o Gerenciador de Erros, com base nas escolhas feitas pelo aluno-jogador, indica suas falhas cometidas. Tais escolhas são disponibilizadas pelo módulo Configurações da Partida. Ademais, o Gerenciador de Erros determina e envia as pontuações dos grupos para o *Ranking* e envia dados para o Gerenciador de *Feedback*;
- **Gerenciador de *Feedback*:** Este módulo gerencia os *feedbacks*, individual e geral, com base nos dados recebidos do Gerenciador de Erros. O *feedback* individual aponta os erros e acertos de um aluno, de acordo com sua função prevista na metodologia escolhida, sendo visível apenas para ele e ao professor. Para que o professor possa visualizar tal *feedback*, este módulo possui um relacionamento com o módulo Visualizar *Feedback* Individual. O *feedback* geral, por sua vez, apresenta os aspectos positivos e negativos gerais da equipe, referentes à metodologia abordada. Além disso, este módulo envia estatísticas, envolvendo o número de erros e acertos e a pontuação do grupo, por exemplo, para o módulo *Ranking*;
- ***Ranking*:** Este módulo apresenta o *ranking* dos grupos, de modo a estabelecer competitividade entre os alunos. Para isso, o *Ranking* recebe as pontuações do Gerenciador de Erros, as estatísticas do Gerenciador de *Feedback* e os dados do grupo das Configurações da Partida, produzindo uma classificação das equipes.

Dada a abrangência dos módulos Gerenciador de Erros e Gerenciador de *Feedback*, a seguir é descrita a decomposição modular realizada para ambos.

3.3.5.1. Gerenciador de Erros

Na Figura 8 é ilustrada a decomposição modular do módulo Gerenciador de Erros, a qual é detalhada na sequência.

Figura 8. Decomposição Modular – Gerenciador de Erros



Fonte: Elaborado pelo autor

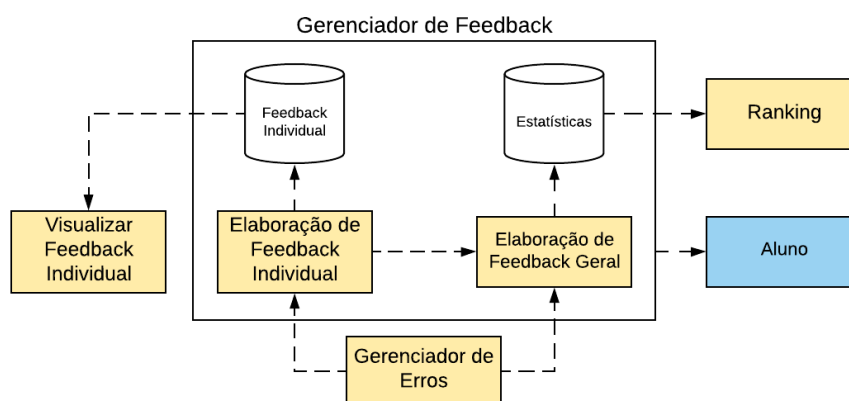
- **Análise das Ações:** Este módulo é responsável pela análise do desempenho de cada aluno, considerando seu papel escolhido. Para que isso seja possível, este módulo se relaciona com as Configurações da Partida;
- **Apresentação dos Erros:** Este módulo é responsável por apresentar ao aluno-jogador seus erros cometidos durante a partida. Para isso, este módulo se relaciona com o módulo Análise das Ações, dado que é necessário a identificação dos erros para, em seguida, apresentá-los.

3.3.5.2. Gerenciador de *Feedback*

Na Figura 9 é ilustrada a decomposição modular do módulo Gerenciador de *Feedback*, a qual é detalhada na sequência.

- **Elaboração de *Feedback* Individual:** Este módulo deve elaborar um *feedback*, a partir dos dados recebidos do módulo Gerenciador de Erros. Tal *feedback* deve apontar os aspectos positivos e negativos do aluno sobre a partida desempenhada. Além disso, este módulo armazena dados na base *Feedback* Individual, a qual fornece dados visíveis apenas ao aluno e ao professor;

Figura 9. Decomposição Modular – Gerenciador de *Feedback*



Fonte: Elaborado pelo autor

- **Elaboração de *Feedback* Geral:** Este módulo é responsável por elaborar e apresentar um *feedback* referente ao desempenho geral do grupo, a partir do cálculo e armazenamento de estatísticas, o que viabiliza a definição de um *ranking*. Dessa forma, os erros e acertos se referem ao conhecimento geral do grupo a respeito da metodologia aplicada. Para isso, existem comunicações com os módulos *Elaboração de Feedback Individual* e *Gerenciador de Erros*.

3.4. Protótipo do Jogo Educativo

Dada a proposta do modelo arquitetural para jogos educativos voltados para o ensino ágil, foi iniciada a elaboração do protótipo e, para tanto, primeiramente foi feita a modelagem tridimensional dos objetos e personagens contidos nos cenários do jogo. Para isso, foi utilizado o software Blender em conjunto com modelos gratuitos encontrados nos sites: Free3D¹ - para os modelos das cadeiras, teclados, mouses, projetor e televisor; e Textures² - para o piso e paredes.

Ademais, para a produção do protótipo do jogo educativo, foi necessário desenvolver um site, o qual utiliza as linguagens HTML e CSS, juntamente com a ferramenta Bootstrap, para o *frontend* e Python, em conjunto com o *framework* Django, para o *backend*, e um executável do protótipo, o qual foi desenvolvido utilizando o software Unity, em conjunto com a linguagem de programação C#. É válido ressaltar que foi utilizada a biblioteca Photon Engine, escrita na linguagem C#, para realizar a conexão entre os jogadores, possibilitando o jogo ser multijogador.

¹ <https://www.free3d.com>

² <https://www.textures.com>

Dado que o protótipo promove o ensino das metodologias Scrum e XP realizando uma simulação de desenvolvimento de um software fictício, quatro cenários foram propostos, conforme apresentado na Tabela 15.

Tabela 15. Cenários do Protótipo

Cenário	Descrição
Sala de Desenvolvimento	Este cenário contém elementos como mesas, cadeiras e computadores, de forma a caracterizar um ambiente de desenvolvimento de software. Ao se aproximar deste local, o jogo disponibiliza uma tela de escolhas referentes ao desenvolvimento de um software fictício para o jogador.
Sala de Reunião de Equipe	Este cenário contém elementos de modo a caracterizar um local de reuniões rápidas e informais. Desse modo, apenas uma mesa e uma lousa são necessárias para este ambiente. Ao se aproximar deste cenário, uma tela contendo as escolhas referentes à reunião de equipe deve ser apresentada ao jogador.
Sala de Reunião com a Diretoria	Este cenário contém apenas cadeiras, uma mesa, um computador, um caderno de anotações e uma lixeira. A aproximação de um jogador a este ambiente faz com que uma tela com as informações referentes ao progresso do desenvolvimento do projeto fictício seja apresentada.
Sala de Reunião com o Cliente	Este cenário contém elementos como mesa, cadeiras e um televisor. Ao se aproximar deste ambiente, é disponibilizado uma tela com as escolhas referentes à interação entre equipe de desenvolvimento e cliente para o jogador.

Fonte: Elaborado pelo autor

Ademais, foi preciso definir as principais características do jogo, indicando o gênero a ser adotado. Assim, foi definido que o jogo será em primeira pessoa e multijogador, o qual pertence ao gênero *Massively Multiplayer Online* (MMO) em primeira pessoa.

Finalizada a modelagem do piso e do personagem principal do jogo, com o auxílio do software Blender foram realizadas as animações do personagem, sendo elas:

movimentação para frente; movimentação para trás; e ocioso, popularmente conhecido como *idle*.

Após isso, os três componentes definidos no diagrama arquitetural (Figura 2) foram implementados, conforme descrito nas subseções a seguir.

3.4.1. Controle de Acesso

O módulo Acesso foi o primeiro a ser desenvolvido, possibilitando o cadastramento e autenticação dos usuários ao sistema.

Para efetuar o cadastro de um novo usuário, foi elaborado o site. A produção de um sistema a parte do jogo é justificada pela facilidade de compartilhar informações, dado que o site pode ser acessado de qualquer dispositivo ao redor do mundo. Além disso, o site é importante para a alteração dos dados do usuário. Feito o cadastramento, é possível realizar o *download* do executável do jogo e iniciar uma partida. Para o site elaborado, foram definidas três telas:

- Página Inicial – Disponibiliza informações sobre como acessar o jogo educativo, conforme apresentado na Figura 10;
- Cadastro – Usada pelos alunos-jogadores para a criação de uma nova conta. Observa-se que a conta para professor possui privilégios para acesso dos *feedbacks* de todos os alunos e, portanto, não deve ser criada por qualquer usuário. Desse modo, uma conta para o professor deve ser criada pelo administrador do sistema. Na Figura 11 é ilustrada a página de Cadastro do site;
- Editar Dados – Permite que um usuário cadastrado visualize e altere suas informações pessoais, conforme ilustrado na Figura 12.

Figura 10. Página Inicial

Scrum XPerience Cadastro Ranking Editar Dados Buscar Aluno Sair

Scrum XPerience

Essa é a página principal do site referente ao jogo educativo para ensino de metodologias ágeis.

O protótipo desse jogo está sendo desenvolvido para apoiar o ensino das metodologias ágeis Scrum e eXtreme Programming, as quais são as mais utilizadas atualmente. Para isso, está sendo utilizado o software Unity, em conjunto com a linguagem de programação C# para a produção do jogo. Este jogo busca corrigir carências encontradas em soluções anteriores, sendo o único, dentre todos os jogos digitais encontrados pelos autores, do estilo multijogador.

Como Jogar?

Para que você possa jogar o jogo educativo para ensino de metodologias ágeis, é necessário realizar duas etapas:

- 1) Criar uma conta.** Caso você ainda não tenha uma conta, basta acessar a [página de cadastramento](#).
- 2) Baixar o Jogo.** Para fazer o download do jogo, [clique aqui](#).

Procure por um feedback individual [aqui](#).

Fonte: Elaborado pelo autor

Figura 11. Página de Cadastro

Scrum XPerience Cadastro Ranking Login

Cadastro

Nome: Insira seu nome

Sobrenome: Insira seu sobrenome

Idade: Insira sua idade

E-mail: Insira um e-mail

Nome de usuário: Insira um nome de usuário
150 caracteres ou menos. Letras, números e @/./+/-/_ apenas.

Senha: Insira uma senha

Confirmar senha: Confirme a senha

Cadastrar

Aviso Legal

Ao se cadastrar, seus dados pessoais serão utilizados apenas para as seguintes funcionalidades:

- Acesso e uso da aplicação e seus serviços disponíveis;
- Acesso aos feedbacks das partidas, gerados automaticamente pelo jogo, pelo(s) professor(es) responsável(is).

Observação: Todos os dados serão excluídos após o período de testes.

Fonte: Elaborado pelo autor

Figura 12. Página de Editar Dados

Scrum XPerience Cadastro Ranking Editar Dados Buscar Aluno Sair

Alterar Dados de Cadastro

Nome: Paulo André Sobrenome: Pimenta Aragão

E-mail: andre.aragao@unesp.br Nome de usuário: andre.aragao
150 caracteres ou menos. Letras, números e @/./+/_ apenas.

☐ Alterar senha Salvar

Fonte: Elaborado pelo autor

Com o *download* do jogo, a partir da Página Inicial, o usuário pode se autenticar ao sistema e iniciar uma partida no jogo. Assim, quando o executável do jogo for iniciado, será requisitado o nome de usuário e senha para autenticar o jogador, dando-o permissão para acesso ao jogo.

Após o desenvolvimento do módulo Acesso, foi definida a comunicação de dados entre o módulo Acesso e o componente Partida, utilizando duas URLs. Essa comunicação é importante para que seja armazenado o histórico de um jogador ao término de uma partida. Desse modo, as URLs são: uma para o retorno de dados do jogador, como o número total de acertos e erros, por meio de um arquivo do tipo JSON; e uma para a recepção de dados de uma partida referente a um determinado jogador.

3.4.2. Partida

Com a finalização do desenvolvimento do componente Controle de Acesso, que engloba o módulo Acesso, foi realizada a implementação do componente Partida. Considerando que o jogo deve ser multijogador, a elaboração do executável foi iniciada com a definição de duas telas, sendo elas: tela de autenticação, a qual requisita os dados para realizar a autenticação do usuário ao sistema; e a tela de configuração da sala, a qual possibilita ao usuário criar uma sala ou ingressar em uma já existente. Na Figura 13 são ilustradas ambas as telas.

Figura 13. Telas de Autenticação e Configuração da Sala

Nome de usuário:
Nome de usuário...

Senha:
Senha...

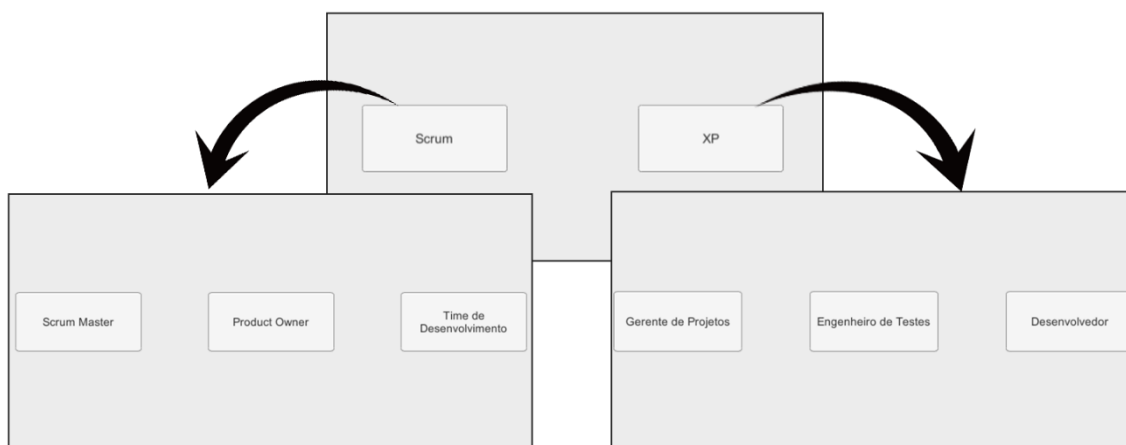
Nome da Sala:
Escolha um nome para a sala...

Criar Sala

Entrar na Sala

Fonte: Elaborado pelo autor

Em sequência, o módulo Configurações da Partida foi desenvolvido, o qual contempla a escolha da metodologia a ser abordada, bem como os papéis a serem desempenhados pelos alunos, como apresentado na Figura 14. Para realizar a escolha da metodologia abordada, foi desenvolvido um mecanismo de votação entre os jogadores. Assim, considerando o limite de três jogadores por sala, a metodologia deve possuir pelo menos dois votos para ser escolhida. Após feita a votação, o jogador decidirá seu papel no jogo e, com isso, tal papel se torna indisponível para os demais jogadores, ou seja, isso torna impossível a definição de papéis duplicados e ausentes.

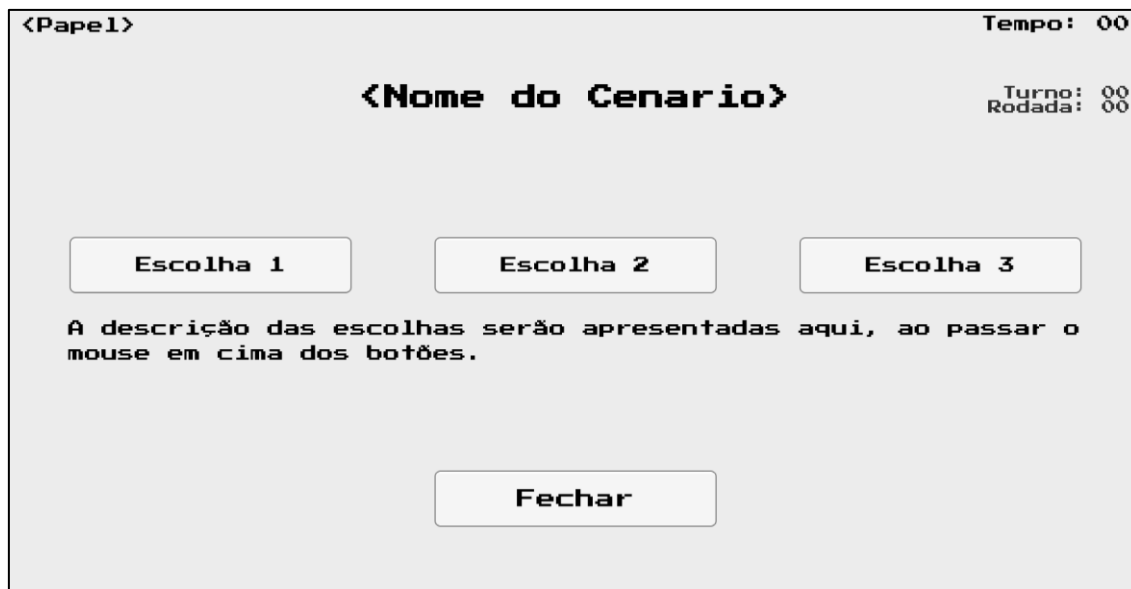
Figura 14. Tela de Configurações da Partida

Fonte: Elaborado pelo autor

Além das escolhas relacionadas à definição da metodologia e seus papéis, foram implementadas as escolhas que o jogador deve realizar durante a partida, as quais são disponibilizadas de acordo com o cenário que o jogador se encontrar e seu papel desempenhado. Na Figura 15 é ilustrado o modelo utilizado para a tela de escolhas, o qual é modificado dinamicamente dependendo do cenário que o jogador estiver. Os campos

que possuem os símbolos “<” e “>” são alterados para o papel escolhido pelo jogador e o cenário que ele se encontra. Já os botões “Escolha 1”, “Escolha 2” e “Escolha 3” serão alterados para as escolhas disponíveis para aquele local e papel escolhido.

Figura 15. Modelo da Tela de Escolha de Ação



Fonte: Elaborado pelo autor

Após tais implementações, foi definida a duração da partida. Desse modo, cada partida será dividida em cinco turnos, cada um contendo quatro rodadas, de modo a simular o desenvolvimento incremental e iterativo de um software fictício. Assim, cada turno representa um incremento, enquanto cada rodada remete a uma etapa do incremento. Além disso, para cada rodada há um tempo limite para realizar as ações, o qual foi definido como dois minutos, levando um total de 40 minutos para finalizar o jogo.

Em sequência, os papéis para cada metodologia abordada pelo jogo educativo foram definidos e descritos, respeitando o limite de três jogadores por grupo. Assim, na Tabela 16 são apresentados os papéis da metodologia XP, e na Tabela 17 são apresentados os papéis da metodologia Scrum, disponíveis no Scrum XPerience.

Tabela 16. Papéis do Scrum XPerience – XP

Papel	Descrição
Gerente de Projetos	O gerente de projetos é o responsável por gerenciar o desenvolvimento do software, garantindo que o sistema seja desenvolvido de acordo com o desejo do cliente. O aluno que escolher este papel deve se responsabilizar por gerenciar o desenvolvimento do projeto fictício, conduzindo escolhas de forma semelhante a uma junção de Product Owner com Scrum Master
Engenheiro de Testes	Sem dúvida um papel crucial para a XP, o Engenheiro de Testes é o responsável pelos seguintes aspectos: definir quais testes serão aplicados no software a ser desenvolvido; auxiliar na elaboração desses testes, produzindo testes automatizados com as devidas ferramentas; e analisar os resultados dos testes aplicados na iteração desenvolvida. Assim, o aluno que escolher esse papel fica responsável pela parte mais importante do desenvolvimento segundo a metodologia XP: os testes. As escolhas dele certamente terão grande impacto na condução correta ou não dos valores da XP, o que impacta no desenvolvimento do software
Desenvolvedor	O desenvolvedor é o responsável pela elaboração dos incrementos e, consequentemente, do software final a ser entregue. Desse modo, ele deve trabalhar em uma equipe auto organizável, de modo a realizarem uma seleção dos requisitos a serem implementados por incremento. O principal ponto é a entrega frequente de releases, construindo o software de “pouco em pouco”. Assim, o jogador que escolher esse papel terá basicamente as mesmas escolhas do membro do time de desenvolvimento Scrum, apenas adequando os nomes e algumas ações para a metodologia XP

Fonte: Elaborado pelo autor

Tabela 17. Papéis do Scrum XPerience – Scrum

Papel	Descrição
Product Owner	As principais funções do Product Owner são: definir os itens do Backlog de Produto; realizar a priorização desses itens, possibilitando a realização do Sprint Backlog; e garantir que não haja interferência nos requisitos que estão sendo implementados em uma determinada Sprint. Desse modo, o Product Owner tem suas escolhas relacionadas à elicitação dos requisitos que o software fictício deve conter, organizando-os de acordo com a prioridade de desenvolvimento. Além disso, ele irá auxiliar na definição do escopo do desenvolvimento de uma Sprint. Por fim, o Product Owner pode verificar as estatísticas do jogo, junto com o Scrum Master
Scrum Master	O papel do Scrum Master é assegurar que o Scrum seja compreendido e executado por toda a equipe de desenvolvimento. Ele também é o responsável por determinar os limites da Sprint, em conjunto com o Product Owner. Além disso, Scrum Master é o responsável pela condução e organização da daily scrum. Desse modo, o Scrum Master tem suas escolhas voltadas para a organização dos requisitos do software (por prioridade de desenvolvimento), juntamente com o Product Owner. Além disso, ele deverá atuar na reunião com a equipe, além de poder ter acesso às estatísticas do grupo, as quais são mostradas na sala de reunião com a diretoria
Time de Desenvolvimento	O Time de Desenvolvimento Scrum é a parte da equipe que irá desenvolver o software de fato. Assim, eles possuem conhecimento técnico necessário para realizar todas as etapas da fase de produção do sistema, desde a prototipagem até a entrega do produto final, passando por revisões de código e elaboração de documentações. Desse modo, no jogo, o jogador que escolher esse papel vai representar apenas um membro do time de desenvolvimento. Assim, ele deve fazer escolhas dentro do escopo do time de desenvolvimento, as quais se referem, em sua maior parte, ao desenvolvimento do software fictício. Além disso, o membro do time de desenvolvimento deve participar da reunião da equipe, de forma a participar das reuniões diárias, planejar, revisar e fazer a retrospectiva da Sprint

Fonte: Elaborado pelo autor

Em cada rodada, o jogador deve realizar duas escolhas, sendo cada uma em um cenário diferente do jogo, o qual dependerá do papel que o aluno está desempenhando. Caso o jogador não realize todas as escolhas a tempo, as restantes serão consideradas como erro. A partida é finalizada de duas formas:

- Derrota – O time sofrerá uma derrota quando totalizar uma quantidade de erros igual ou superior a 60, representando metade do desenvolvimento conduzido de forma incorreta. Com isso, os jogadores serão informados da derrota, possibilitando voltar ao menu principal do jogo e começar uma nova partida;
- Vitória – O grupo será vitorioso caso consiga finalizar todos os turnos, sem atingir a marca de 60 erros. Em seguida, com base nos acertos e erros, os *feedbacks* são elaborados e disponibilizados, acrescentando a pontuação ao grupo, e a partida é finalizada.

Com a definição dos papéis e modos de finalização da partida, foi necessário definir as relações entre os papéis e cenários do jogo, como apresentado na Tabela 18.

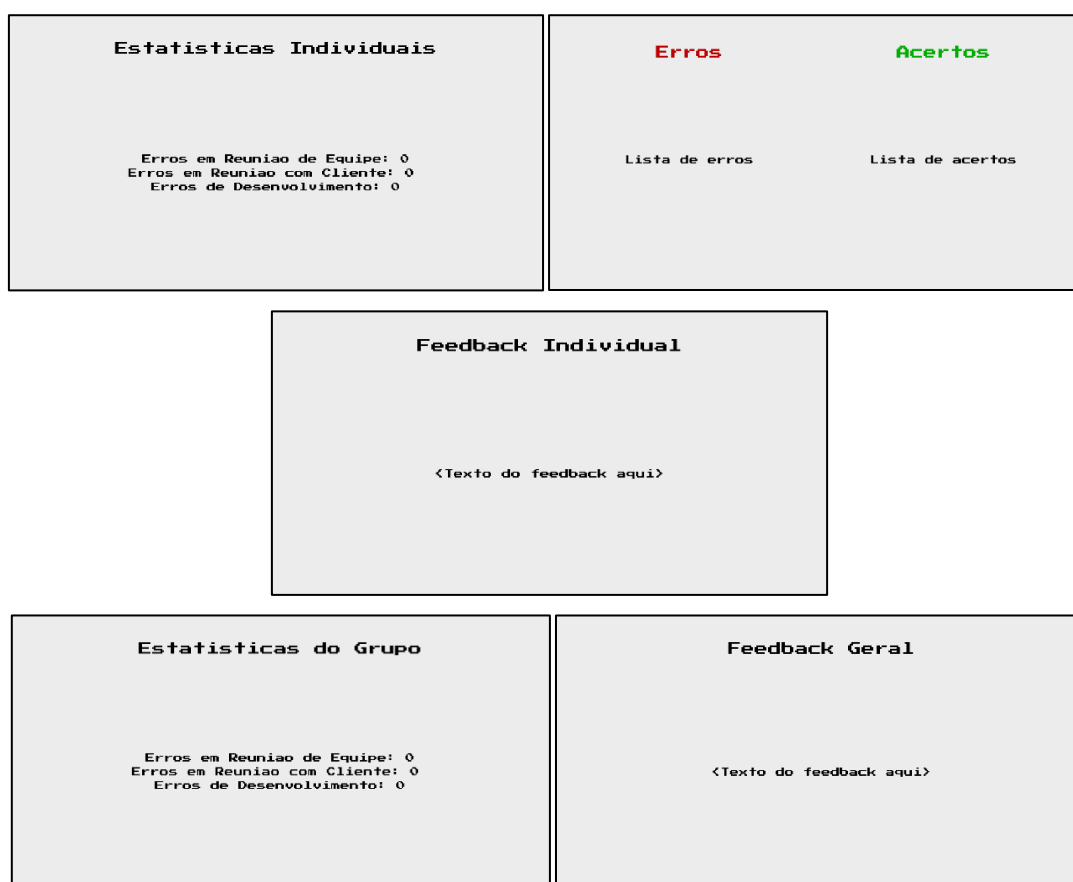
Tabela 18. Relação entre Papéis e Cenários do Scrum XPerience

	Product Owner	Scrum Master	Time de Desenvolvimento	Gerente de Projetos	Engenheiro de Testes	Desenvolvedor
Sala de Desenvolvimento						
Sala de Reunião de Equipe						
Sala de Reunião com a Diretoria						
Sala de Reunião com o Cliente						

■ Possui escolha ■ Não possui escolha

Fonte: Elaborado pelo autor

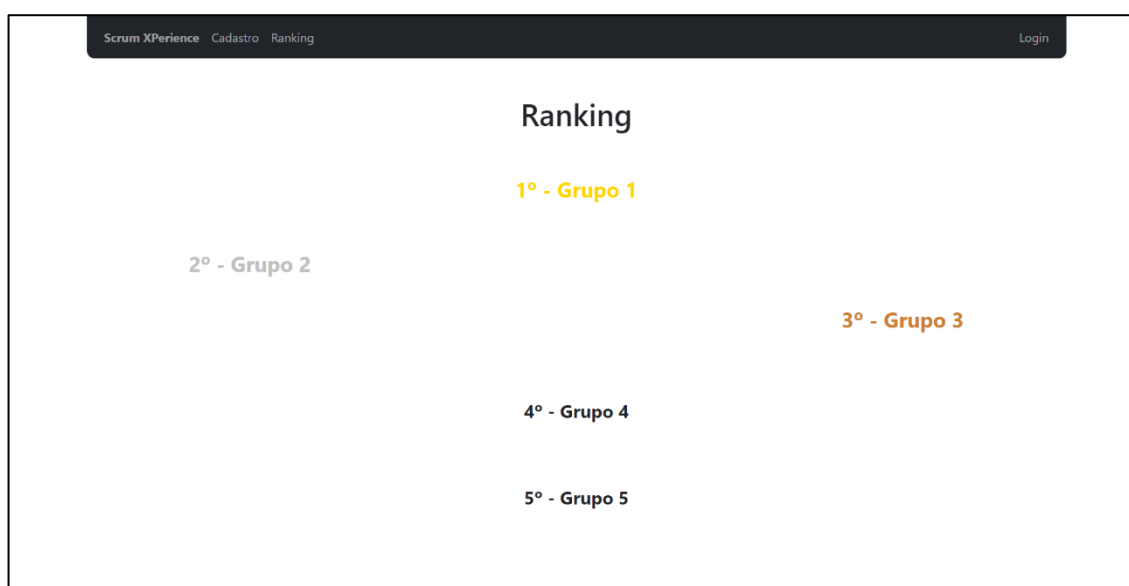
Com a finalização da prototipação do módulo Configurações da Partida, o módulo Gerenciador foi desenvolvido. Durante a execução da partida, o módulo Gerenciador se responsabiliza por analisar as decisões tomadas pelo jogador, definindo se ela é um erro ou acerto. Assim, ao finalizar a partida, o jogo consultará os erros e acertos armazenados, de forma a produzir as estatísticas individuais do jogador, bem como seu *feedback*. Em seguida, são apresentadas as estatísticas e *feedback* gerais, referente ao grupo como um todo. As telas de estatísticas e *feedbacks* são ilustradas na Figura 16.

Figura 16. Estatísticas e *Feedbacks*

Fonte: Elaborado pelo autor

Com isso, a implementação do *Ranking* foi viabilizada, a qual utiliza os erros e estatísticas do grupo para definir uma pontuação e, conseqüentemente, uma classificação. A pontuação é atribuída da seguinte forma: cada decisão correta adiciona um ponto para o grupo. Na Figura 17 é ilustrado um exemplo do *ranking*, o qual será disponibilizado no site.

Após isso, o módulo Conteúdo Didático, o qual é responsável por permitir ao administrador do sistema realizar o preenchimento das bases de dados que serão apenas consultadas durante a partida, foi elaborado. Assim, cabe apenas à entidade externa Administrador inserir, excluir ou alterar as informações contidas nessas bases de dados. Na Figura 18, uma ilustração das telas de preenchimento das escolhas do jogo, em que o Administrador pode alterar ou criar uma escolha.

Figura 17. Ranking

Fonte: Elaborado pelo autor

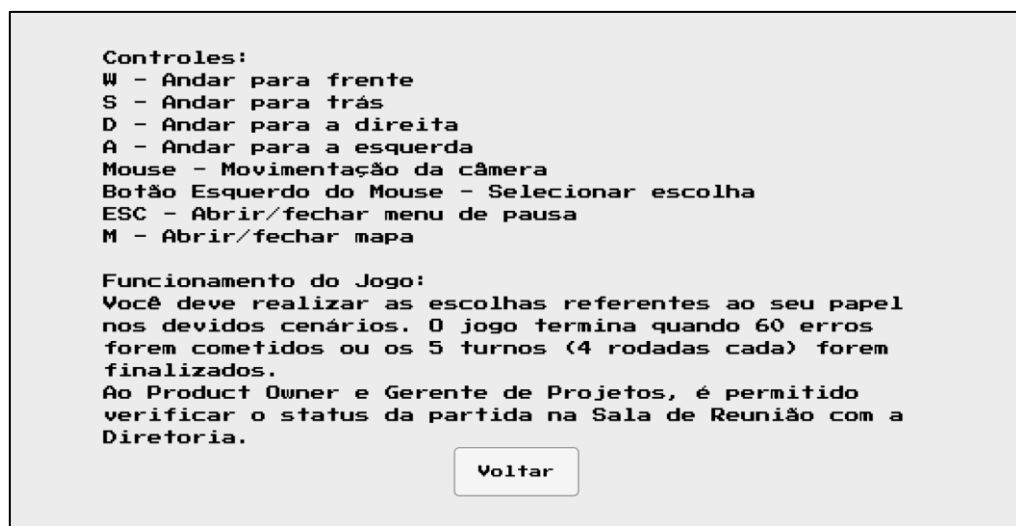
Figura 18. Preenchimento das Escolhas do Jogo

Fonte: Elaborado pelo autor

Em seguida, o módulo Ensino foi desenvolvido, o qual é voltado para o ensino dos conceitos e dinâmica do jogo para os alunos-jogadores. Para isso, foi necessário definir o conjunto de regras da partida, além da mecânica básica e a diferença entre Scrum e XP. Assim, um tutorial jogável foi desenvolvido, o qual ensina ao jogador o funcionamento da partida, contemplando as regras, comandos e ações possíveis. É válido destacar que, para jogar uma partida em grupo, é obrigatório que todos os jogadores joguem o tutorial ao menos uma vez antes de participar de uma partida em grupo. O tutorial está disponível no menu principal sempre que o jogador desejar jogá-lo, conforme ilustrado na Figura 19. Ademais, durante a partida, ao acessar o menu de pausa, é possível acessar a Ajuda, contemplando os comandos do jogo e seu objetivo, como ilustrado na Figura 20.

Figura 19. Modos de Jogo

Fonte: Elaborado pelo autor

Figura 20. Ajuda

Fonte: Elaborado pelo autor

Com isso, o componente Partida foi completamente finalizado, resultando no executável do protótipo. Assim, na Figura 21 é ilustrado o mapa do jogo, de forma a apresentar a localização dos cenários do jogo utilizando uma visão de topo da cena principal. Na Figura 22 é ilustrado o protótipo em execução.

Figura 21. Mapa do Jogo



Fonte: Elaborado pelo autor

Figura 22. Protótipo em Execução



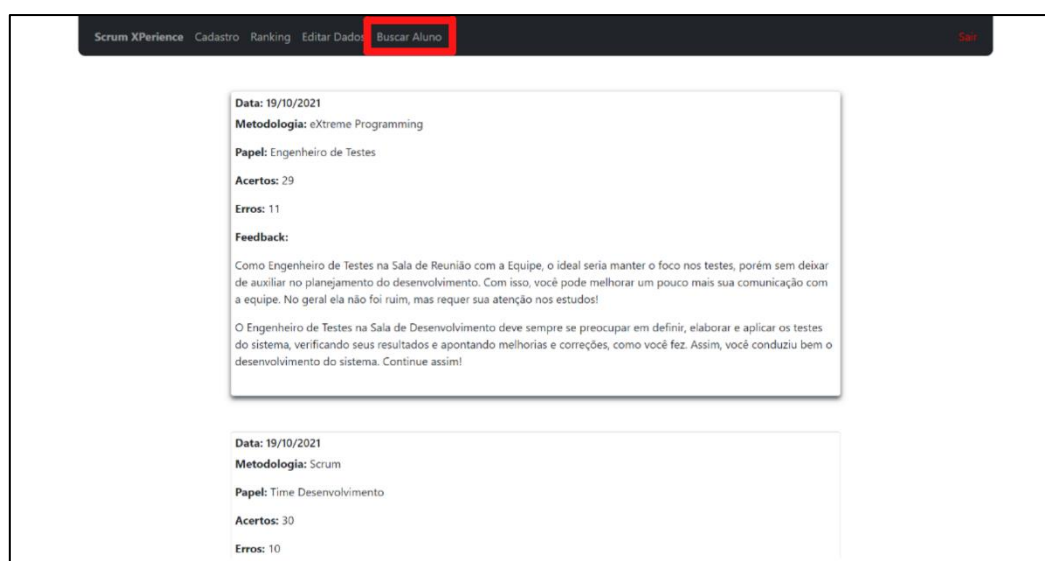
Fonte: Elaborado pelo autor

3.4.3. Área do Docente

O último módulo desenvolvido foi o Visualizar *Feedback* Individual, o qual permite ao professor buscar e visualizar os *feedbacks* individuais de um determinado aluno. Para isso, é necessária uma conta com permissão de Professor, a qual é garantida apenas pelo Administrador. Com isso, o professor pode se autenticar no site e utilizar a opção “Buscar Aluno” para procurar um determinado aluno. Ao realizar a busca, o site irá retornar uma lista contendo todos os alunos com nomes parecidos com o parâmetro de pesquisa, juntamente com um link para acesso aos *feedbacks* individuais.

Ao acessar o link do aluno desejado, o professor é redirecionado à uma página contendo todos os *feedbacks* individuais do aluno, os quais são separados utilizando uma estilização em *cards*. Dessa forma, são disponibilizadas as seguintes informações da partida: data; metodologia; papel; quantidade de acertos; quantidade de erros; e o *feedback* individual. Na Figura 23 é ilustrado um exemplo da tela de *feedbacks*.

Figura 23. Exemplo de *Feedbacks*



Fonte: Elaborado pelo autor

Com isso, todos os módulos da arquitetura foram completamente desenvolvidos, resultando em um protótipo de jogo educativo de alta fidelidade. Com a implementação do protótipo foi possível observar que o diagrama arquitetural proposto é aplicável, uma vez que não foram necessárias alterações para o desenvolvimento do protótipo de jogo educativo. Com o protótipo finalizado, viabilizou-se a sua avaliação pelo público alvo, ou seja, alunos e docentes, com o intuito de identificar possíveis aprimoramentos e

correções almejadas para garantir melhor qualidade ao protótipo e, com isso, auxiliar no desenvolvimento de futuros jogos educativos voltados ao ensino ágil.

3.5. Considerações Finais

Neste capítulo, os resultados obtidos com o desenvolvimento da pesquisa foram apresentados, destacando a elaboração do diagrama arquitetural para jogos educativos voltado ao ensino das metodologias ágeis Scrum e XP e a construção do protótipo de jogo educativo, demonstrando a viabilidade do uso da arquitetura definida.

No Capítulo 4, a avaliação do protótipo e os resultados obtidos são descritos, bem como as alterações realizadas.

Capítulo 4 – Avaliação do Protótipo

4.1. Processo de Avaliação

Com a finalização do desenvolvimento do protótipo do jogo educativo, foi iniciado o processo de avaliação envolvendo o público alvo: alunos de graduação e pós-graduação e docentes vinculados a cursos de Computação. Logo, foi necessário elaborar dois formulários avaliativos, sendo um para os alunos-jogadores, que efetivamente jogaram o jogo, e outro para os docentes, que avaliaram a área exclusiva para docentes disponível no jogo. Para estruturação do formulário, foram utilizados alguns estudos que auxiliaram na definição dos aspectos a serem avaliados, como os de Cupers Schmid e Hildebrand (2013), Lima *et al.* (2017), Petri *et al.* (2020) e Souza e Souto (2015). Nos Apêndices A e B são apresentados os formulários avaliativos disponibilizados aos alunos e aos docentes, respectivamente. O período de avaliação foi de 18 de outubro de 2021 a 03 de novembro de 2021.

Observa-se que as respostas dos formulários utilizavam, em sua maior parte, uma escala linear que variava de 1 a 5, sendo 1 correspondente à “discordo plenamente” e 5 à “concordo plenamente”. As demais respostas eram de múltipla escolha, permitindo ao avaliador apenas concordar ou discordar das afirmações propostas. Uma resposta de campo aberto também foi utilizada em cada formulário, possibilitando aos avaliadores colocarem suas sugestões e comentários livremente.

O convite aos alunos foi realizado a partir de uma lista eletrônica composta por alunos atualmente matriculados no curso de Bacharelado em Ciência da Computação do

IBILCE/UNESP, além de amigos do autor, externos à referida Instituição, que atuam na área de Computação. Os docentes, por sua vez, foram contatados por e-mail, enviado com base em uma seleção prévia de 23 docentes vinculados à renomadas Instituições de Ensino Superior brasileiras, públicas e privadas, atuantes na área de Engenharia de Software. Desses, apenas seis retornaram contato, demonstrando interesse e disponibilidade para realizar a avaliação. É válido destacar que todas as respostas, tanto dos alunos quanto dos professores, eram anônimas. Os resultados do processo de avaliação são discutidos nas subseções a seguir.

4.2. Formulário Avaliativo para Alunos

Este formulário foi dividido em quatro seções, sendo elas: dados do jogador, que busca analisar qual a área e nível de formação do avaliador, bem como se ele possui conhecimento sobre Engenharia de Software e métodos ágeis; avaliação da interface, que busca analisar a adequação da interface definida para o jogo; avaliação do conteúdo e jogabilidade; e opinião geral do jogador sobre sua experiência com o jogo (campo aberto). Assim, o jogo recebeu um total de 30 avaliações de alunos-jogadores.

Em relação aos dados dos jogadores, foi possível observar que todos os avaliadores são da área de Computação, sendo 29 de Ciência da Computação e 1 de Análise e Desenvolvimento de Sistemas, os quais fazem parte do público alvo. Além disso, apenas um dos avaliadores possui nível de formação de mestrado, sendo os 29 restantes enquadrados em nível de graduação. Por fim, apenas cinco avaliadores não cursaram a disciplina de Engenharia de Software e não possuem conhecimento prévio sobre metodologias ágeis.

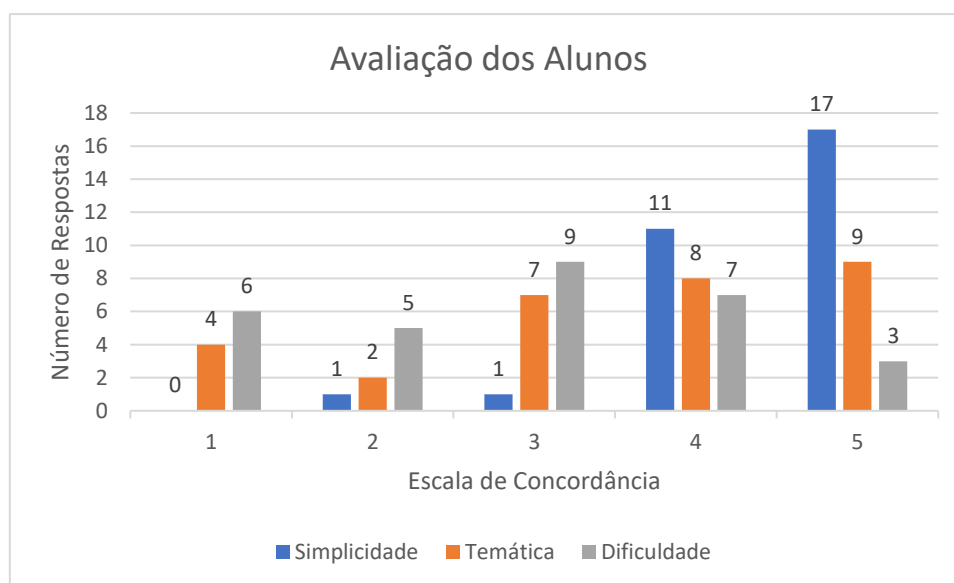
Da análise das respostas obtidas com relação à interface foi possível observar que o Scrum XPerience obteve a maior parte de suas avaliações concentradas nas escalas de 3 a 5, correspondendo de médio a ótimo. Como é possível observar no histograma ilustrado na Figura 24, tanto a simplicidade quanto a temática do protótipo Scrum XPerience receberam predominantemente avaliações positivas. No entanto, as avaliações da dificuldade se concentraram nas escalas de 1 a 3, revelando uma deficiência do jogo que demanda atenção futura.

Já com relação ao conteúdo e jogabilidade, destaca-se o fato de que 26 avaliadores afirmaram que o jogo contribuiu para o aprendizado sobre os conceitos de metodologias ágeis, enquanto apenas três discordaram e um se manteve neutro. Além disso, nota-se

que, segundo os avaliadores, as escolhas elaboradas são de fácil compreensão, o jogo é divertido e imersivo. Logo, depreende-se que, em âmbito geral, o conteúdo e a jogabilidade do jogo educativo agradaram a maioria dos jogadores.

Em relação às opiniões, foi possível observar que houveram poucos comentários ou sugestões sobre o conteúdo do jogo, os quais eram relacionados à adição de novas metodologias e possibilitar escolhas em grupo. Além disso, diversos jogadores reportaram problemas encontrados durante a partida, sendo eles: desconexão dos jogadores; alta sensibilidade do mouse; alta duração da partida. Sugestões como alteração de cor do texto para acertos e erros, reconexão dos jogadores, zerar o cronômetro após todos finalizarem suas decisões, não precisando aguardar obrigatoriamente o tempo limite fixado para uma rodada, e utilizar efeitos sonoros, também foram apresentados pelos jogadores em suas avaliações. Ademais, o primeiro grupo de avaliadores reportou que a movimentação da câmera causava náusea, sugerindo que isso fosse alterado.

Figura 24. Histograma da Avaliação dos Alunos



Fonte: Elaborado pelo autor

4.3. Formulário Avaliativo para Docentes

Este formulário, por sua vez, foi elaborado para analisar apenas a Área do Docente, não demandando que os professores jogassem. Desse modo, seis docentes participaram da avaliação, respondendo às questões do formulário.

Com base nas respostas obtidas, observa-se que cinco avaliadores concordam que a interface do sistema é intuitiva e fácil de entender e um se manteve neutro. Cinco avaliadores concordam que as informações fornecidas pelos *feedbacks* são suficientes para identificar erros conceituais, enquanto um se manteve neutro em relação a esse aspecto. Todos os professores concordam que as informações dos *feedbacks* auxiliam no planejamento das aulas. Ademais, todos concordam que o jogo auxilia no ensino de Scrum e XP. Por fim, cinco avaliadores afirmaram que adotariam o jogo como apoio para o ensino de Scrum e XP e um se manteve neutro. Com isso, é possível depreender que a Área do Docente agrada aos professores, fornecendo informações relevantes sobre as dificuldades encontradas pelos alunos, de modo a auxiliar no planejamento das aulas.

Além disso, referente às opiniões dos professores, foi possível observar que a iniciativa de utilizar jogos educativos colaborativos para o ensino de metodologias ágeis foi considerada interessante. Todos os comentários foram positivos e sugestões como a produção de um vídeo explicativo do jogo e apresentação das escolhas individuais de cada partida foram contempladas na seção de opiniões.

4.4. Reestruturação do Protótipo

A partir da avaliação, tanto pelos alunos quanto pelos professores, foi possível identificar correções e aprimoramentos do protótipo por meio dos comentários e sugestões dos avaliadores, de modo a garantir melhor qualidade aos resultados do presente trabalho. Dessa forma, as alterações efetivadas são apontadas a seguir.

A primeira alteração foi a movimentação da câmera. A primeira versão do protótipo contava com uma leve movimentação vertical da câmera junto à movimentação do personagem, com o intuito de assemelhar a movimentação real da cabeça, promovendo maior imersão. No entanto, alguns alunos se sentiram desconfortáveis com isso e relataram uma sensação de mal estar. Desse modo, tal movimentação foi retirada.

A segunda alteração efetivada, que foi apontada pelos avaliadores, foi o uso de cores para a identificação de escolha correta ou errada. Assim, quando um jogador realiza uma escolha certa, o texto que lhe é apresentado possui a cor verde. Caso o jogador faça uma decisão errada, o texto é apresentado na cor vermelha. Além disso, o menu de pausa do jogo foi alterado, de modo a contemplar um botão para sair do jogo, conforme apontado nas sugestões dos avaliadores.

Um dos aspectos mais comentados nas sugestões do formulário avaliativo foi a alta sensibilidade da câmera. Além disso, um dos avaliadores do jogo identificou em seu comentário que a sensibilidade da movimentação da câmera poderia estar relacionada à qualidade gráfica do jogo. Assim, qualidades mais baixas resultavam em sensibilidade menores, enquanto qualidades mais altas representavam sensibilidades maiores. Como forma de solucionar esse problema, uma opção para alterar a sensibilidade do mouse e, conseqüentemente, da câmera, foi adicionada ao menu de pausa do jogo. Com isso, os jogadores podem definir a sensibilidade que mais lhe agrada, sem que a experiência do jogo seja comprometida.

Outro ponto comentado por diversos alunos no formulário avaliativo foi a duração da partida. Como apontado anteriormente, cada rodada tinha duração de dois minutos, totalizando 40 minutos para uma partida toda. Assim, os jogadores sugeriram que a rodada acabasse logo que as escolhas fossem feitas ou que fosse diminuído o tempo de cada rodada. Porém, considerando o fato de que, para a Sala de Reunião com a Diretoria, por exemplo, apenas alguns papéis específicos podem acessá-la, é importante que o tempo restante da rodada seja aproveitado para verificar o status da partida ou, ainda, para realizar discussões entre os alunos de quais aspectos merecem atenção, visando alcançar um melhor desempenho do grupo na partida. Desse modo, optou-se por reduzir o tempo de cada rodada para um minuto e 15 segundos.

Por fim, um erro ao se aproximar da TV da Sala de Reunião com o Cliente fazia com que a interface do jogo desaparecesse. A provável causa disso é o colisor do objeto com o jogador. Assim, como correção, a TV foi posicionada no lado oposto da sala, o qual não é acessível fisicamente ao jogador.

4.5. Considerações Finais

Neste capítulo foi apresentada a avaliação do Scrum XPerience, contemplando a visão dos alunos-jogadores e dos professores atuantes na área de Engenharia de Software. Tal análise buscou verificar seus aspectos positivos e deficiências existentes relacionados à interface, jogabilidade e conteúdo, além de identificar a opinião do público sobre o protótipo.

Assim, os alunos puderam contribuir com a identificação de problemas relacionados com a jogabilidade, realizando sugestões de aprimoramento para o protótipo. Os professores, por sua vez, realizaram suas avaliações a respeito da Área do

Docente, verificando a qualidade dos *feedbacks* individuais, de modo a identificar se o jogo auxilia ou não o professor no processo de ensino de metodologias ágeis, especialmente as metodologias Scrum e XP. Além disso, com base nos problemas encontrados e nas sugestões, foi possível realizar correções no protótipo, de modo a melhorar sua qualidade final.

É válido observar que o Scrum XPerience recebeu, em sua maioria, tanto pelos jogadores quanto pelos docentes, avaliações entre 3 e 5, considerando a escala de 1 a 5, indicando que o protótipo teve suas avaliações entre média e ótima. No entanto, a dificuldade do jogo e a frustração dos jogadores foram os aspectos que receberam as piores avaliações, demandando atenção no desenvolvimento futuro do projeto.

No Capítulo 5 as contribuições deste trabalho são apontadas, bem como propostas para trabalhos futuros.

Capítulo 5 – Conclusão

5.1. Considerações Finais

A partir dos problemas existentes com o ensino tradicional da Engenharia de Software, jogos educativos foram propostos como método de apoio e ganharam notoriedade pela dinâmica e diversão fornecidas para o processo de ensino-aprendizagem. A partir da análise da literatura, foi possível identificar que, dentre os tópicos da Engenharia de Software abordados pelos jogos educativos, destaca-se as metodologias ágeis como uma das principais. Apesar da relevância do ensino ágil em jogos educativos, foi possível observar que eles abordam, em sua maior parte, apenas a metodologia Scrum. Além disso, outras carências foram observadas por meio de uma análise dos jogos educativos existentes, como a falta de interação social e gerenciamento de erros e fornecimento de *feedbacks*.

Assim, uma arquitetura de software foi definida com o intuito de abordar duas metodologias ágeis. Logo, além do Scrum, a XP também foi considerada na solução desenvolvida. Além disso, a interação social foi contemplada, projetando uma solução multijogador, e o acompanhamento do desempenho dos jogadores foi priorizado, de forma que alunos e professores, a partir da solução apresentada, pudessem ter conhecimento de aspectos conceituais que demandam mais atenção no processo de ensino-aprendizagem adotado.

A partir de tal arquitetura foi possível desenvolver o Scrum XPerience e, assim, obter uma solução capaz de sanar as carências observadas nas soluções existentes, bem como de características almejadas identificadas na literatura para softwares educativos.

Com a finalização do protótipo, foi realizada a avaliação pelo público alvo: alunos de graduação e pós-graduação de cursos de Computação e docentes atuantes na área de Engenharia de Software. A partir da análise das respostas dos formulários avaliativos, foi possível identificar alguns problemas e melhorias para o protótipo que foram parcialmente incorporados. Porém, é importante destacar o fato de que a maior parte dos alunos avaliadores consideram que o jogo contribui para a aprendizagem de conceitos ágeis e consideram ele divertido. Em relação aos professores, foi possível identificar que o *feedback* produzido pelo jogo educativo auxilia no planejamento das aulas, facilitando a identificação das dificuldades enfrentadas pelos alunos.

Assim, é possível concluir que o protótipo desenvolvido demonstra que a arquitetura de software estabelecida é viável para conduzir a implementação futura de jogos educacionais de qualidade no contexto de metodologia ágeis. Além disso, o Scrum XPerience se mostrou uma proposta em potencial para contribuir para o avanço das pesquisas nessa área.

5.2. Contribuições

As principais contribuições resultantes deste trabalho são:

- Elaboração de uma arquitetura de software capaz de nortear o desenvolvimento de jogos educativos variados, voltados ao ensino das metodologias ágeis Scrum e XP. Destaca-se o fato de tal arquitetura ter sido estruturada de forma a generalizar os aspectos de qualidade almejados para jogos educativos, fornecendo um arcabouço abrangente capaz de respeitar a criatividade dos desenvolvedores. Com isso, diferentes jogos podem ser criados sob a temática de metodologias ágeis, contribuindo para a garantia de qualidade e diversidade dos jogos educativos voltados para o ensino dessa área;
- Desenvolvimento de um protótipo de jogo educativo voltado para o ensino de Scrum e XP, utilizando a arquitetura definida neste trabalho. O Scrum XPerience contribui para o processo de ensino aprendizagem de maneira diferenciada, uma vez que sua concepção e desenvolvimento tiveram como foco sanar as

deficiências identificadas nas soluções atuais. Além disso, em decorrência da avaliação do protótipo por alunos e professores da área de Computação, foi possível incorporar melhorias ao Scrum XPerience, tornando-o ainda mais abrangente e capaz de atender às expectativas do público alvo.

5.3. Trabalhos Futuros

A partir da análise das respostas obtidas com os formulários avaliativos, tanto dos docentes quanto dos alunos, foi possível identificar sugestões envolvendo implementação de melhorias para este projeto. Assim, tais apontamentos indicam a evolução do presente trabalho, de modo a aprimorar o protótipo desenvolvido e, conseqüentemente, garantir melhor qualidade aos futuros jogos educativos voltados para esta área.

Um aprimoramento apontado pelos alunos-jogadores envolve a desconexão repentina dos jogadores da partida. Analisando o protótipo elaborado e a biblioteca utilizada, foi possível identificar que o problema está relacionado nos métodos que realizam a conexão entre o cliente e servidor, fornecido pelo *framework* Photon Engine. Apesar de tal problema ter sido apontado por apenas seis avaliadores, isso pode afetar a experiência dos jogadores, causando frustração aos alunos. É válido destacar que o Photon Engine não possui servidores no Brasil, além de ter sido utilizado a versão gratuita do serviço, voltada apenas para testes. Assim, pode-se supor que o problema esteja relacionado a tais aspectos, podendo ser resolvido de duas formas: utilização da versão paga do Photon Engine, a qual fornece mais recursos para a aplicação; ou utilização de um sistema de *host* e cliente, em que um dos jogadores do grupo é o *host* e os demais se conectam a ele.

Outra implementação futura identificada a partir de sugestões dos jogadores é a adição de níveis de dificuldade para a partida. Assim, as diferentes dificuldades podem variar o tempo limite e a quantidade de erros máximos permitidos. Em dificuldades mais fáceis, o tempo deve ser maior, bem como a quantidade de erros máximos permitidos para o grupo, possibilitando aos alunos tomarem suas decisões calmamente. Em dificuldades maiores, o tempo e o limite de erros devem ser menores, desafiando os alunos a tomarem decisões rápidas e corretas. Ademais, é importante que a reconexão dos jogadores que tiveram sua conexão com a Internet interrompida seja feita sem demandar o encerramento da partida atual.

Além disso, outro trabalho futuro é permitir que as escolhas dos jogadores possam ser realizadas em grupo, por meio de votação. Desse modo, os alunos devem discutir entre si para tomar a decisão correta, promovendo maior interação social. Para isso, poderá ser adicionado um chat para os alunos apresentarem suas ideias dentro do jogo.

Ademais, a adição de cenários, metodologias e papéis é considerada uma implementação futura para o protótipo. Assim, novos cenários e papéis poderão englobar outras etapas do desenvolvimento ágil, aumentando a qualidade do ensino promovido pelo jogo. Novas metodologias expandirão o conhecimento dos alunos sobre os diferentes métodos ágeis existentes, promovendo uma visão crítica sobre o desenvolvimento ágil e aprimorando sua formação.

Futuramente, recomenda-se que o jogo educativo utilize recursos sonoros, além da música já disponibilizada no menu principal. Tais efeitos podem incluir o barulho de passos ao andar pelos cenários, música durante a partida e sons para evidenciar acertos e erros cometidos, por exemplo. Assim, espera-se promover maior imersão ao jogo, fazendo com que o usuário volte sua atenção para ele, e, conseqüentemente, sinta-se mais motivado a aprender.

Conforme recomendado por dois docentes, o jogo educativo, em especial a Área do Docente, deve disponibilizar aos professores uma visão dos erros e acertos individuais dos alunos em cada partida. Assim, o professor poderá analisar os conceitos que os alunos possuem maior dificuldade, alterando o planejamento da aula e garantindo o ensino eficaz de metodologias ágeis. Por fim, futuramente, o jogo educativo deve englobar outras línguas além do português brasileiro, ampliando o número de potenciais beneficiados com esta solução.

Referências

- ALHAMMAD, M. M.; MORENO, A. M. Gamification in software engineering education: A systematic mapping. In: **Journal of Systems and Software**, v. 141, p. 131-150, 2018.
- ALMEIDA, R. L. F.; ALMEIDA, C. A. S. **Fundamentos e análise de software educativo**. Brasil: EdUECE, 2015.
- AMMONS, B.; BANSAL, S. K. Scrumify: A Software Game to Introduce Agile Software Development Methods. In: **Journal of Engineering Education Transformations**, v. 0, p. 1-7, 2017.
- BECK, K.; *et al.*, “**Manifesto for Agile Software Development**”. 2001. Available in: <http://agilemanifesto.org/>. Acesso em 04 de julho de 2021.
- BECK, K.; GAMMA, E. **Extreme programming explained: embrace change**. Estados Unidos da América: Addison-Wesley Professional, 2000.
- BEPPE, T. A.; ARAÚJO, Í. L.; ARAGÃO, B. S.; ISMAYLE, S. S.; XIMENES, D.; ANDRADE, R. M. C. GreaTest: a card game to motivate the software testing learning. In: **Proceedings of the XXXII Brazilian Symposium on Software Engineering**. 2018. p. 298-307. São Carlos, Brasil.

- BRITO, A. S. **'2TSCRUM': um jogo de tabuleiro para o ensino do Scrum.** Monografia (Bacharelado/Tecnólogo em Sistemas de Informação). Universidade Federal do Ceará, Quixadá, 2016.
- BRITO, A.; VIEIRA, J. '2TScrum' A Board Game to Teach Scrum. In: **Proceedings of the 31st Brazilian Symposium on Software Engineering.** 2017. p. 279-288. Fortaleza, Brasil.
- CALDERÓN, A.; RUIZ, M.; ORTA, E. Integrating serious games as learning resources in a software project management course: the case of ProDec. In: **I International Workshop on Software Engineering Curricula for Millennials.** 2017. p. 21-27.
- CASTRO, R. M.; SIQUEIRA, S. W. M.; ALMEIDA, D. N.; NASCIMENTO, F. C. AGILITY SCRUM – Um Jogo para Ensino da Metodologia SCRUM. In: **XXV Workshop sobre Educação em Computação.** 2017. p. 2217- 2226.
- CONFESSOR, B. R. **XS-GAME: ENGENHARIA DE JOGOS VOLTADA PARA DESENVOLVEDORES INDIVIDUAIS.** Monografia (Graduação em Engenharia de Computação e Informação). Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2019.
- CUPERSCHMID, A. R. M.; HILDEBRAND, H. R. Avaliação Heurística de Jogabilidade. In: **Proceedings of SBGames.** 2013. p. 371-378. São Paulo, Brasil.
- FARIAS, V.; MOREIRA, C.; COUTINHO, E.; SANTOS, I. S. iTest Learning: Um jogo para o ensino do planejamento de testes de software. In: **V Fórum de Educação em Engenharia de Software.** Natal, 2012.
- FERNANDES, L.; WERNER, C. Sobre o uso de Jogos Digitais para o Ensino de Engenharia de Software. In: **II Fórum de Educação em Engenharia de Software.** Fortaleza, 2009.
- FILHO, M. M.; BENICIO, I. V.; CAMPOS, F.; NEVES, A. M. M. A importância da prototipação no design de games. In: **Proceedings of SBGames.** 2013. p. 312-318. São Paulo, Brasil.
- GARCÍA, I.; PACHECO, C.; LEÓN, A.; CALVO-MANZANO, J. A. Experiences of using a game for improving learning in software requirements elicitation. In: **Computer Applications in Engineering Education**, v. 27, p. 249-265, 2019.

- GKRITSI, A. **Scrum Game: an agile software management game**. Dissertation (MSc Software Engineering). University of Southampton, Southampton, 2011.
- HENRIQUE, M. S. **EDUCATALOG4RE: um catálogo de requisitos para auxiliar o desenvolvimento softwares educacionais**. Dissertação (Mestrado em Ciência da Computação). Universidade Federal de Pernambuco, Recife, 2016.
- HODA, R. Using Agile Games to Invigorate Agile and Lean Software Development Learning in Classrooms. In: **Agile and Lean Concepts for Teaching and Learning**. p. 391-414. Singapura: Springer, 2019.
- JÁCOME, W. C. **Softbook: Desenvolvimento de Software Como uma Aventura**. Monografia (Bacharelado em Ciência da Computação). Universidade Federal do Rio Grande do Norte, Natal, 2013.
- KOSA, M.; YILMAZ, M.; O'CONNOR, R. V.; CLARKE, P. M. Software engineering education and games: a systematic literature review. In: **Journal of Universal Computer Science**, v. 22, p. 1558-1574, 2016.
- LEE, W. L. SCRUM-X: An interactive and experiential learning platform for teaching scrum. In: **VII International Multi-Conference on Complexity, Informatics and Cybernetics – VII International Conference on Society and Information Technologies**. 2016. p. 192-197.
- LIMA, I.; BUENO, D.; PERRY, G. T. Avaliação de usabilidade e de experiência de jogo digital educacional: uma experiência com o SUSCity. In: **Revista Novas Tecnologias na Educação**. p. 246-255, v. 15, n. 1, 2017.
- MAXIM, B. R.; KAUR, R.; APZYNSKI, C.; EDWARDS, D.; EVANS, E. An agile software engineering process improvement game. In: **2016 IEEE Frontiers in Education Conference**. IEEE, 2016. p. 1-4.
- MOLLÉRI, J. S.; GONZALEZ-HUERTA, J.; HENNINGSSON, K. A legacy game for project management in software engineering courses. In: **III European Conference of Software Engineering Education**. 2018. p. 72-76.
- NAVARRO, E. O.; HOEK, A. SIMSE: An Interactive Simulation Game for Software Engineering Education. In: **VII International Conference on Computers and Advanced Technology in Education**. 2004. p. 12-17.

- PETRI, G.; WANGENHEIM, C. G.; BORGATTO, A. F. MEEGA+: Um Modelo para a Avaliação de Jogos Educacionais para o ensino de Computação. In: **Revista Brasileira de Informática na Educação**, v. 27, p. 52-81, n. 3, 2020.
- PETRI, G.; WANGENHEIM, C. G.; BORGATTO, A. F. Qualidade de jogos digitais e não digitais utilizados para o ensino de engenharia de software no Brasil. **Revista de Gestão e Avaliação Educacional**, v. 7, n. 14, p. 9-29, 2018.
- PRESSMAN, R. S. **Engenharia de Software: Uma Abordagem Profissional**. Brasil: AMGH Editora, 2011.
- RODRÍGUEZ-SILVA, D. A.; RODRÍGUEZ-CRUZ, I.; GONZÁLEZ-CASTAÑO, F. J. A software architecture for virtualized educational multimedia games in IPTV environments. In: **Journal Software: Practice and Experience**, v. 45, p. 143-160, 2015.
- SANTOS, H. M.; DURELLI, V. H. S.; SOUZA, M.; FIGUEIREDO, E.; SILVA, L. T.; DURELLI, R. S. CleanGame: Gamifying the Identification of Code Smells. In: **XXXIII Brazilian Symposium on Software Engineering**. p. 437-446, 2019.
- SAVI, R.; WANGENHEIM, C. G.; BORGATTO, A. F. A model for the evaluation of educational games for teaching software engineering. In: **XXV Brazilian Symposium on Software Engineering**. p. 194-203, 2011.
- SKALKA, J.; DRLÍK, M. P. Proposal of System Architecture for Programming Learning and Teaching Environment. In: **XXII International Conference on Application of Information and Communication Technologies**. 2018. p. 1-6.
- SOMMERVILLE, I. **Engenharia de Software**. Brasil: Pearson, 2011.
- SOUZA, A.; SEABRA, R.; RIBEIRO, J.; RODRIGUES, L. SCRUMI: Um Jogo Virtual Sério de Tabuleiro para o Ensino do Framework SCRUM. In: **XIII Simpósio Brasileiro de Sistemas de Informação**. p. 520-526, 2017.
- SOUZA, É. R.; SOUTO, E. Utilização de heurísticas de jogos para avaliação de um aplicativo gamificado. In: **Proceedings of SBGames**. 2015. p. 666-673. Teresina, Brasil.
- STAVREV, S.; TERZIEVA, T.; GOLEV, A. Concepts for distributed input independent architecture for serious games. In: **Proceedings of CBU International Conference**. 2018. p. 1166-1172. Praga, República Checa.

- SUTHERLAND, J.; SCHWABER, K. **Guia do Scrum. Um guia definitivo para o Scrum: As regras do Jogo.** Scrum Org., Disponível em: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Portuguese-Brazilian.pdf>, 2017.
- ÜNLÜ, K.; ARDIÇ, B.; TÜZÜN, E. CRSg: a serious game for teaching code review. In: **Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering.** 2020. p. 1561-1565. Nova Iorque, Estados Unidos da América.
- VEGT, W.; WESTER, W.; NYAMSUREN, E.; GEORGIEV, A.; ORTIZ, I. M. RAGE architecture for reusable serious gaming technology components. In: **International Journal of Computer Games Technology**, v. 2016, p. 3-13, 2016.
- WANGENHEIM, C. G.; SAVI, R.; BORGATTO, A. F. SCRUMIA – An educational game for teaching SCRUM in computing courses. In: **Journal of Systems and Software**, v. 86, p. 2675-2687, 2013.
- WILDT, D.; MOURA, D.; LACERDA, G.; HELM, R. **eXtreme Programming: Práticas para o dia a dia no desenvolvimento ágil de software.** Brasil: Editora Casa do Código, 2015.
- YUXUAN, C. **Diretrizes para desenvolvimento de software educativo com ênfase na qualidade motivacional.** Dissertação (Mestrado em Ciência da Computação). Universidade Estadual Paulista “Júlio de Mesquita Filho”, São José do Rio Preto, 2020.

Apêndice A – Formulário Avaliativo para Alunos

Avaliação Scrum XPerience

Olá! Este é o formulário avaliativo do jogo educativo Scrum XPerience, resultante de um projeto de pesquisa desenvolvido pelo aluno Paulo André Pimenta Aragão, sob orientação da Profa. Dra. Rogéria Cristiane Gratão de Souza, com apoio financeiro da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP - 2019/20663-0).

Este formulário servirá para a análise da opinião do público-alvo, de modo a identificar possíveis melhorias e corrigir eventuais erros. Assim, almeja-se contribuir para que o ensino de metodologias ágeis, especialmente o Scrum e a XP, seja melhorado, enriquecendo a formação de futuros profissionais.

Com isso em mente, pedimos a sua colaboração, por meio de feedback a respeito do jogo. Esclarecemos que todas as informações fornecidas e os resultados obtidos durante o jogo serão mantidos em sigilo, e estes últimos só serão utilizados para fins de pesquisa acadêmica, incluindo publicações científicas, sem qualquer identificação pessoal dos respondentes. Você não terá quaisquer benefícios ou direitos financeiros sobre os eventuais resultados decorrentes da pesquisa. Este estudo é importante porque seus resultados fornecerão informações relevantes sobre a contribuição da tecnologia para a educação, bem como para o processo de ensino-aprendizagem.

*Obrigatório

Dados do Jogador

Esta seção contempla perguntas sobre sua formação acadêmica para definição do perfil dos respondentes

1. Em qual dos cursos de graduação listados se enquadra sua formação (em andamento ou concluída)? Em caso de você não cursar computação, por favor informar em "Outros". *

Marcar apenas uma oval.

- ☐ Ciência da Computação
- ☐ Engenharia da Computação
- ☐ Análise e Desenvolvimento de Sistemas
- ☐ Sistemas de Informação
- ☐ Outro: _____

2. Qual o seu nível de formação atualmente? (Exemplo: se você está cursando graduação, assinale "Graduação"). *

Marcar apenas uma oval.

- ☐ Graduação
☐ Mestrado
☐ Doutorado
☐ Tecnólogo

3. Você já cursou, ou cursa atualmente, a disciplina Engenharia de Software? *

Marcar apenas uma oval.

- ☐ Sim
☐ Não

4. Você já estudou, ou estuda atualmente, conteúdo envolvendo conceitos de metodologias ágeis? *

Marcar apenas uma oval.

- ☐ Sim
☐ Não

Avaliação da Interface

Esta seção tem como objetivo avaliar a qualidade da interface definida para o jogo.

5. O jogo é intuitivo, explicativo e fácil de entender. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Plenamente	☐	☐	☐	☐	☐	Concordo Plenamente

6. O estilo do jogo é padronizado e consistente. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Plenamente	☐	☐	☐	☐	☐	Concordo Plenamente

7. O tutorial do jogo atende as expectativas.

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Plenamente	☐	☐	☐	☐	☐	Concordo Plenamente

8. O tutorial do jogo é fácil acesso. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Plenamente	☐	☐	☐	☐	☐	Concordo Plenamente

9. A temática do jogo é atrativa. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Plenamente	☐	☐	☐	☐	☐	Concordo Plenamente

10. As informações de ajuda e regras do jogo são de fácil acesso. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Plenamente	☐	☐	☐	☐	☐	Concordo Plenamente

Avaliação do
Conteúdo e
Jogabilidade

Estamos quase terminando! Nesta seção encontram-se perguntas sobre o conteúdo do jogo, bem como sua jogabilidade.

11. Você considera as escolhas do jogo de fácil compreensão? *

Marcar apenas uma oval.

☐ Sim

☐ Não

12. Você se sentiu frustrado durante algum momento do jogo? *

Marcar apenas uma oval.

☐ Sim

☐ Não

13. Há sensação de imersão com o jogo, fazendo você se sentir parte dele? *

Marcar apenas uma oval.

☐ Sim

☐ Não

14. O jogo é difícil de jogar. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Plenamente	☐	☐	☐	☐	☐	Concordo Plenamente

15. Em algum momento do jogo, você sentiu felicidade e/ou vontade de rir? *

Marcar apenas uma oval.

☐ Sim

☐ Não

16. O jogo é divertido. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo plenamente	☐	☐	☐	☐	☐	Concordo plenamente

17. O jogo contribuiu para o meu aprendizado de conceitos sobre metodologias ágeis. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Plenamente	☐	☐	☐	☐	☐	Concordo Plenamente

Sua Opinião

Por fim, queremos saber sua opinião sobre o jogo.

18. Dê sua opinião, faça sugestões de melhorias, identifique problemas e/ou faça qualquer comentário extra que você deseja a respeito do jogo.

Com isso em mente, pedimos a sua colaboração para avaliar a área destinada ao docente localizada no site do jogo. Esclarecemos que todas as informações fornecidas serão mantidas em sigilo, podendo ser utilizadas apenas para fins de pesquisa acadêmica, incluindo publicações científicas, sem qualquer identificação pessoal dos respondentes. Você não terá quaisquer benefícios e/ou direitos sobre os eventuais resultados decorrentes da pesquisa. Este estudo é importante porque seus resultados fornecerão informações relevantes sobre a contribuição da tecnologia para a educação, bem como para o processo de ensino-aprendizagem.

***Obrigatório**

1. A interface do sistema é intuitiva e fácil de entender. *

Marcar apenas uma oval.

1 2 3 4 5

Discordo plenamente ☐ ☐ ☐ ☐ ☐ Concordo plenamente

2. As informações fornecidas pelos feedbacks são suficientes para identificar erros conceituais cometidos pelos alunos. *

Marcar apenas uma oval.

1 2 3 4 5

Discordo plenamente ☐ ☐ ☐ ☐ ☐ Concordo plenamente

3. As informações fornecidas pelos feedbacks ajudam no planejamento das aulas, uma vez que contribuem para identificar aspectos que precisam ser melhor trabalhados com os alunos. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo plenamente	☐	☐	☐	☐	☐	Concordo plenamente

4. O jogo auxilia no ensino das metodologias ágeis Scrum e XP. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Plenamente	☐	☐	☐	☐	☐	Concordo Plenamente

5. Eu adotaria o jogo Scrum XPerience como apoio ao ensino de conceitos sobre as metodologias ágeis Scrum e XP. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo plenamente	☐	☐	☐	☐	☐	Concordo plenamente

6. Faça suas sugestões, comentários e/ou críticas a respeito do Scrum XPerience. Gostaríamos muito de ouvir o que você tem a dizer sobre este jogo educativo.
