



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Instituto de Matemática e Estatística

Leonardo Lima Marinho

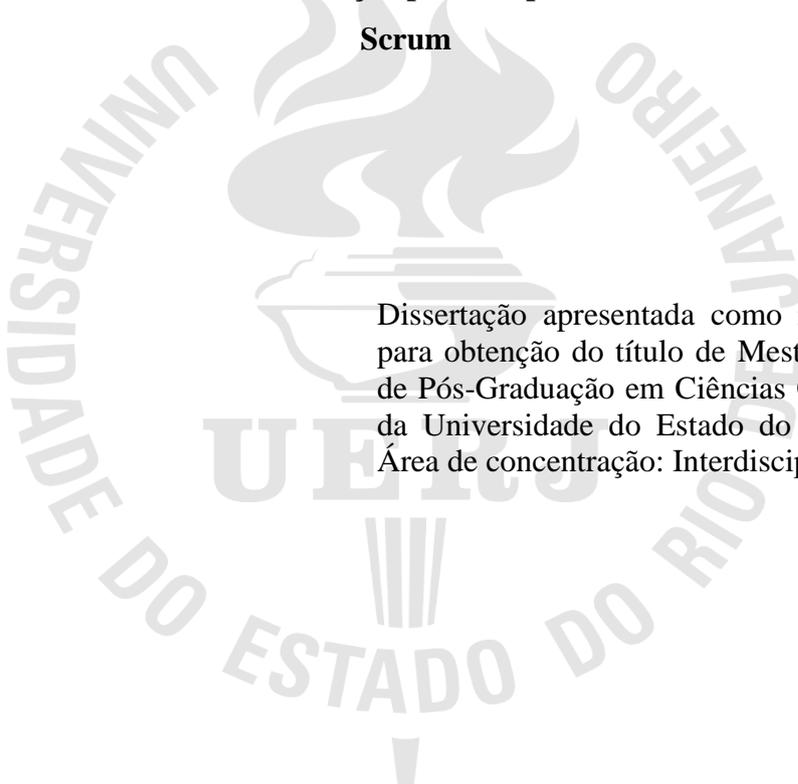
**ScrumAge: Serious Game de Simulação para o Aprendizado da
Metodologia Ágil Scrum**

Rio de Janeiro

2022

Leonardo Lima Marinho

**ScrumAge: Serious Game de Simulação para o Aprendizado da Metodologia Ágil
Scrum**



Dissertação apresentada como requisito parcial
para obtenção do título de Mestre, ao Programa
de Pós-Graduação em Ciências Computacionais,
da Universidade do Estado do Rio de Janeiro.
Área de concentração: Interdisciplinar.

Orientadoras: Prof.^a Dra. Rosa Maria Esteves Moreira da Costa
Prof.^a Dra. Vera Maria Benjamin Werneck

Rio de Janeiro

2022

CATALOGAÇÃO NA FONTE
UERJ / REDE SIRIUS / BIBLIOTECA CTC-A

M338 Marinho, Leonardo Lima
ScrumAge: Serious Game de simulação para o aprendizado da metodologia
Ágil Scrum / Leonardo Lima Marinho. – 2022.
115 f. : il.

Orientadoras: Rosa Maria Esteves Moreira da Costa, Vera Maria Benjamim
Werneck.

Dissertação (Mestrado em Ciências Computacionais) - Universidade do
Estado do Rio de Janeiro, Instituto de Matemática e Estatística.

1. Scrum (Desenvolvimento de software) - Teses. 2. Jogos -
Processamento de dados - Teses. 3. Tecnologia educacional - Teses. I. Costa,
Rosa Maria Esteves Moreira da. II. Werneck, Vera Maria Benjamim. III.
Universidade do Estado do Rio de Janeiro. Instituto de Matemática e
Estatística. IV. Título.

CDU 004.415.2

Patricia Bello Meijinhos CRB7/5217 - Bibliotecária responsável pela elaboração da ficha catalográfica

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta
dissertação, desde que citada a fonte

Assinatura

Data

Leonardo Lima Marinho

**ScrumAge: Serious Game de Simulação para o Aprendizado da Metodologia Ágil
Scrum**

Dissertação apresentada como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Ciências Computacionais, da Universidade do Estado do Rio de Janeiro. Área de concentração: Interdisciplinar.

Aprovada em 06 de setembro de 2022.
Banca Examinadora:

Prof.^a Dra. Rosa Maria Esteves Moreira da Costa (Orientadora)
Instituto de Matemática e Estatística - UERJ

Prof.^a Dra. Vera Maria Benjamim Werneck (Orientadora)
Instituto de Matemática e Estatística - UERJ

Prof.^a Dra. Fernanda Claudia Alves Campos
Universidade Federal de Juiz de Fora - UFJF

Prof.^a Dra. Flavia Maria Santoro
Instituto de Matemática e Estatística - UERJ

Rio de Janeiro

2022

RESUMO

MARINHO, Leonardo Lima. *ScrumAge: Serious Game de Simulação para o Aprendizado da Metodologia Ágil Scrum*. 2022. 118 f. Dissertação (Mestrado em Ciências Computacionais) - Instituto de Matemática e Estatística, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2022.

As metodologias ágeis estão sendo amplamente utilizadas para o gerenciamento de projetos, especialmente os de desenvolvimento de software. Para essa finalidade, o Scrum é a metodologia mais utilizada e requisitada pela indústria. Com base nisso, estudos sobre novas abordagens que busquem simplificar seu aprendizado podem ser de grande valia. Por outro lado, dentre os métodos mais promissores para auxiliar em propósitos educativos, estão os serious games, que são jogos com objetivos além do puro entretenimento. Nesse contexto, uma revisão bibliográfica foi realizada para verificar quais serious games já foram criados com a finalidade de apoiar o aprendizado do Scrum. Essa pesquisa identificou que, embora alguns jogos com a mesma finalidade já existam, ainda há espaço para novas abordagens, explorando estratégias mais motivadoras e considerando os possíveis papéis envolvidos no processo de desenvolvimento de um software. Sendo assim, este trabalho descreve a concepção, a prototipação e a avaliação de um serious game para auxiliar o aprendizado do Scrum, chamado ScrumAge. Nele, o jogador participará da simulação do desenvolvimento de um software por uma equipe utilizando o Scrum. O ScrumAge pretende se destacar entre os demais jogos de propósito similar ao ser mais acessível, ao introduzir ao jogador conceitos complementares úteis na realidade, mas usualmente desconsiderados no ensino acadêmico, e ao disponibilizar um repositório de conhecimento integrado para manter os conceitos teóricos facilmente acessíveis durante a prática simulada pelo jogo. Após o desenvolvimento do protótipo, 39 profissionais de tecnologia da informação realizaram uma avaliação que demonstrou uma percepção bastante positiva sobre o sistema.

Palavras-chave: Scrum. Metodologias ágeis. Serious Game.

ABSTRACT

MARINHO, Leonardo Lima. *ScrumAge: Simulation Serious Game for Learning the Scrum Agile Methodology*. 2022. 118 f. Dissertação (Mestrado em Ciências Computacionais) - Instituto de Matemática e Estatística, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2022.

Agile methodologies are being widely used for project management, especially software development. For this purpose, Scrum is the most used and requested methodology in the industry. Based on this, studies on new approaches that seek to simplify their learning can be of great value. On the other hand, among the most promising methods to assist in educational purposes are serious games, which are games with goals beyond pure entertainment. In this context, a bibliographic review was conducted to verify which serious games have already been created to support Scrum learning. This research identified that, although some games with the same purpose already exist, there is still room for new approaches, exploring more motivating strategies and considering the possible roles involved in the software development process. Therefore, this work describes the design, prototyping, and evaluation of a serious game to help learn Scrum, called ScrumAge. In it, the player will participate in the simulation of software development by a team using Scrum. ScrumAge intends to stand out among other similar-purpose games by being more accessible, introducing the player to complementary concepts that are useful but usually disregarded in academic teaching. Also, by providing an integrated knowledge repository to keep theoretical concepts easily accessible during the practice simulated by the game. After the prototype development, 39 information technology professionals carried out an evaluation that showed a very positive perception of the system.

Palavras-chave: Scrum. Agile Methods. Serious Game.

LISTA DE FIGURAS

Figura 1 –	O <i>framework</i> Scrum.....	18
Figura 2 –	Trabalhos correlatos – Ano de publicação.....	24
Figura 3 –	Trabalhos correlatos – Quantidade de jogadores.....	25
Figura 4 –	Trabalhos correlatos – Gênero.....	25
Figura 5 –	Trabalhos correlatos – Plataforma.....	26
Figura 6 –	Diagrama de Atividades UML: fluxo do jogo ScrumAge.....	34
Figura 7 –	Diagrama de Atividades UML: Realizar Planejamento de Sprint.....	35
Figura 8 –	Diagrama de Atividades UML: Realizar Reuniões Diárias.....	36
Figura 9 –	Diagrama de Atividades UML: Realizar Reuniões de final de Sprint.	36
Figura 10 –	Modelo Conceitual UML: ScrumAge.....	37
Figura 11 –	Resumo de conceito da Wiki no protótipo do ScrumAge.....	39
Figura 12 –	Artigo completo da Wiki no protótipo do ScrumAge.....	39
Figura 13 –	Papel atual no protótipo do ScrumAge.....	40
Figura 14 –	Tela de classificação no protótipo do ScrumAge.....	40
Figura 15 –	Acesso ao ScrumAge através de um smartphone.....	49
Figura 16 –	Tela Inicial – Introdução ao ScrumAge.....	51
Figura 17 –	Apresentação do Projeto.....	52
Figura 18 –	Escolha do Time de Desenvolvimento.....	53
Figura 19 –	Planejamento da Sprint – Parte 1: Mensagem do cliente e Objetivo da Sprint.....	54
Figura 20 –	Planejamento da Sprint – Parte 2: Priorização de tarefas.....	55
Figura 21 –	Planejamento da Sprint – Parte 3: Estimativa de tarefas.....	56
Figura 22 –	Planejamento da Sprint – Parte 4: Seleção de tarefas para o Backlog da Sprint.....	57
Figura 23 –	Reunião Diária.....	60

Figura 24 – Exemplos de cartões de tarefas.....	61
Figura 25 – Reunião Diária – Decidir sobre tarefa.....	64
Figura 26 – Reunião Diária – Problema detectado.....	65
Figura 27 – Revisão e Retrospectiva da Sprint - Quadro Kanban compactado.....	66
Figura 28 – Revisão da Sprint.....	67
Figura 29 – Retrospectiva da Sprint.....	70
Figura 30 – Fim de Jogo.....	72
Figura 31 – Grau de Escolaridade.....	77
Figura 32 – Ano de Ingresso e de Conclusão de Curso.....	78
Figura 33 – Instituições de Ensino.....	79
Figura 34 – Experiência com Desenvolvimento Ágil.....	79
Figura 35 – Experiência com Engenharia de Software.....	80
Figura 36 – Experiência com Desenvolvimento de Software.....	81
Figura 37 – Experiência com Desenvolvimento de Software com Método Ágil....	81
Figura 38 – Experiência com Desenvolvimento de Software com Scrum.....	82
Figura 39 – Gráfico das Respostas ao Questionário de Conteúdo.....	83
Figura 40 – Gráfico das Respostas ao Questionário de Interface.....	84
Figura 41 – Gráfico das Respostas ao Questionário de Usabilidade – SUS.....	86
Figura 42 – Adjetivo para a pontuação SUS de cada avaliador do ScrumAge.....	88

LISTA DE TABELAS

Tabela 1 –	Comparativo entre trabalhos correlatos encontrados na revisão da bibliografia.....	22
Tabela 2 –	Significado dos ícones dos cartões das tarefas.....	61
Tabela 3 –	Relação entre produtividade e probabilidades de adiantamento e atraso.....	63
Tabela 4 –	Relação entre a senioridade e a probabilidade de cada qualidade.....	63
Tabela 5 –	Definição da produtividade.....	70
Tabela 6 –	Textos de feedback: desempenho por tema.....	74
Tabela 7 –	Escala tipo Likert.....	76
Tabela 8 –	Escala de grau de experiência.....	80
Tabela 9 –	Dados das Respostas ao Questionário de Conteúdo.....	84
Tabela 10 –	Dados das Respostas ao Questionário de Interface.....	85
Tabela 11 –	Pontuação SUS e adjetivos correspondentes, segundo Bangor et al. (2008).....	87

LISTA DE ABREVIATURAS E SIGLAS

API –	Application Programming Interface
FDD –	Feature-Driven Development
JSON –	JavaScript Object Notation
JWT –	JSON Web Token
PO –	Product Owner
PSM-I –	Professional Scrum Master I
REST –	Representational State Transfer
RPG –	Role-Playing Game
SM –	Scrum Master
SPA –	Single Page Application
SUS –	System Usability Scale
TD –	Time de Desenvolvimento
UML –	Unified Modeling Language
XP –	Extreme Programming

SUMÁRIO

	INTRODUÇÃO	11
1.	REFERENCIAL TEÓRICO	15
1.1	Metodologias Ágeis	15
1.2	Scrum	16
1.3	Serious Games em Engenharia de Software	19
2.	REVISÃO BIBLIOGRÁFICA	20
2.1	Planejamento e Condução	20
2.2	Análise dos Resultados da Revisão	21
2.3	Considerações Finais da Revisão	27
3.	PROPOSTA: SCRUMAGE	30
3.1.	Requisitos	30
3.1.1	<u>Requisitos não funcionais</u>	30
3.1.2	<u>Requisitos funcionais</u>	31
3.2.	Planejamento e Modelagem	32
3.3.	Visão Geral	38
4	PROTÓTIPO DO SCRUMAGE	43
4.1	Etapas do Desenvolvimento	43
4.1.1	<u>Desenvolvimento do sistema</u>	43
4.1.2	<u>Escrita do conteúdo textual</u>	45
4.1.3	<u>Balanceamento</u>	46
4.2	Aspectos Técnicos	48
4.3	Cenário de Uso	50
4.3.1	<u>Reunião de Planejamento de Sprint</u>	54
4.3.2	<u>Reuniões Diárias</u>	59
4.3.3	<u>Reunião de Revisão da Sprint</u>	65
4.3.4	<u>Reunião de Retrospectiva da Sprint</u>	69
4.3.5	<u>Fim de Jogo</u>	71
5	AVALIAÇÃO DO SCRUMAGE	75

5.1	Análise dos Resultados	76
5.1.1	<u>Caracterização dos Participantes</u>	76
5.1.2	<u>Conteúdo</u>	82
5.1.3	<u>Interface</u>	84
5.1.4	<u>Usabilidade</u>	85
5.1.5	<u>Comentários dos participantes da pesquisa</u>	88
5.2	Conclusões da Avaliação	93
	CONSIDERAÇÕES FINAIS	98
	REFERÊNCIAS	101
	APÊNDICE A – Sprints e tarefas do projeto simulado no protótipo do ScrumAge	105
	APÊNDICE B – Textos de feedback ao final da partida.....	109
	APÊNDICE C – Lista de artigos da Wiki do ScrumAge	111
	APÊNDICE D – Termo de Consentimento Livre e Esclarecido	112
	APÊNDICE E – Questionário de Caracterização do Participante.....	113
	APÊNDICE F – Questionário de Avaliação do ScrumAge	114

INTRODUÇÃO

As metodologias ágeis estão dominando o mercado de desenvolvimento de software. Segundo o 15th Annual State Of Agile Report (DIGITAL.AI, 2021), 94% das empresas desse setor estão praticando métodos ágeis. Essas metodologias surgiram como alternativa ao método tradicional de desenvolvimento, que não conseguia responder rapidamente a mudanças de prioridades, levando diversos projetos ao fracasso.

Ainda de acordo com o 15th Annual State Of Agile Report (DIGITAL.AI, 2021), dentre tais metodologias, o Scrum é a mais popular para o desenvolvimento de software. Nessa pesquisa, 66% dos participantes disseram que suas equipes seguiam o Scrum mais proximamente, seguidos de 15% que disseram utilizar duas variações do Scrum, o ScrumBan (9%) e uma forma híbrida do Scrum e do XP (6%), totalizando 81% de participantes utilizando metodologias baseadas no Scrum.

Por outro lado, novos métodos de ensino vêm sendo testados para facilitar a disseminação do conhecimento de diferentes áreas. Uma dessas novas abordagens são os jogos educativos, que exploram situações lúdicas para incentivar o aprendizado. Esses jogos fazem parte de uma categoria mais abrangente: os *serious games*. Eles são jogos com um propósito além do puro entretenimento, como por exemplo para treinamento, publicidade, ou simulação, além da própria educação (SUSI et al., 2007).

Nesse sentido, há uma demanda crescente por formas de tornar o ensino tradicional mais alinhado com o conhecimento prático, com destaque para a área de gerenciamento de projetos (VON WANGENHEIM et al., 2013). Ao mesmo tempo, jogos educativos com foco no ensino de engenharia de software têm obtido bons resultados (GARCÍA et al., 2019), tornando o processo de aprendizado mais eficiente e agradável. Abordando especificamente os *serious games* para o ensino de metodologias ágeis, a revisão da literatura realizada por Rodríguez (2021) indicou que jogos com esse perfil parecem ser os instrumentos de suporte mais bem sucedidos para complementar outros métodos de ensino.

Sob esse contexto, uma revisão bibliográfica foi realizada para identificar *serious games* que foram criados com a finalidade de apoiar o aprendizado do Scrum. Nela, 10 trabalhos foram encontrados, indicando resultados majoritariamente positivos para este método de ensino complementar. Entretanto, nesses trabalhos foram identificadas algumas

lacunas, que justificam a realização de mais estudos sobre o assunto. Dentre elas, destaca-se a inexistência de propostas que seja multiplataforma, que possam ser jogados tanto em *desktops*, quanto em dispositivos móveis. Esse aspecto, associado ao fato de que a maioria dos jogos encontrados foram projetados para serem utilizados exclusivamente no meio acadêmico, em sala de aula e muitas vezes em grupo, tende a restringir a disponibilidade do sistema para pessoas que queiram aprender sobre Scrum por conta própria, ou por empresas que busquem treinar seus funcionários nessa metodologia.

Outra lacuna observada foi a falta de uma fonte de conhecimento teórico, que fosse disponibilizada internamente ao sistema, promovendo assim o fácil acesso a informações importantes sobre o Scrum, ou a qualquer outro conceito relevante para o jogo. Além disso, nenhum dos jogos estudados se destacou na apresentação de conceitos não referentes ao Scrum, mas que contribuíssem para o realismo do projeto.

Mesmo em relação à prática do Scrum simulada pelos jogos, somente um deles (ARAGÃO, 2022) permite que o jogador atue nos três papéis do time Scrum. Porém, nele o jogador deve escolher apenas em um papel por partida, sendo preciso participar de três partidas para experimentar as responsabilidades de cada papel. No mesmo sentido, nenhum dos trabalhos permite, no mesmo jogo, que o jogador priorize, estime e selecione quais tarefas devem compor o Backlog da Sprint, algumas das principais atividades do Time Scrum.

Para preencher tais lacunas, este trabalho apresenta a proposta, a prototipação e a avaliação do ScrumAge, um *serious game* que simula o desenvolvimento de um software por uma equipe utilizando a metodologia ágil Scrum. Este software visa auxiliar no aprendizado de estudantes e profissionais da área de tecnologia da informação que desejam aprender sobre o Scrum, ajudando a suprir a demanda por profissionais com esse conhecimento. No desenvolvimento deste jogo utilizou-se como inspiração as características positivas dos jogos de mesmo propósito, além de abordagens que visam propor soluções para as limitações neles identificadas.

Objetivando ser acessível em diferentes situações e por uma maior parcela de pessoas, o ScrumAge foi planejado para ser disponibilizado em um website responsivo (que se adapta a qualquer tamanho de tela), de forma que possa ser acessado por qualquer navegador, em qualquer dispositivo com acesso à internet. Nesse mesmo sentido, optou-se por criar um jogo *single player*, para que qualquer pessoa interessada em aprender sobre o Scrum possa jogar

uma partida sem depender de outras pessoas. Ainda sob a mesma ótica, é permitido que o jogador pare e retome a uma partida a qualquer momento, sem qualquer tipo de penalidade.

Já para suprir a necessidade identificada de uma base de conhecimento teórico integrada ao sistema, o ScrumAge disponibiliza uma Wiki. Ela pode ser acessada a qualquer momento pelo menu superior da aplicação, ou usando palavras-chave ao longo do jogo, onde toda palavra que possui uma referência na Wiki tem sua visualização destacada e é clicável. O ScrumAge também aborda conceitos complementares ao Scrum e comumente utilizados, como a metodologia Kanban e o Planning poker, disponibilizando artigos na base de conhecimento.

Pretendendo aumentar o realismo da simulação e tornar o jogo mais desafiador, o ScrumAge utiliza conceitos tais como: as qualidades funcional e técnica, calculadas de acordo com a senioridade do desenvolvedor que finalizou cada tarefa; a satisfação do cliente, calculada de acordo com a eficiência do gerenciamento e com a qualidade funcional das tarefas finalizadas; a produtividade do time de desenvolvimento, proporcional à qualidade técnica do projeto; entre outros. Além disso, atividades já concluídas podem necessitar de uma decisão do jogador para melhorá-las, ou podem apresentar problemas ou necessitar de melhorias, caso não estejam com uma boa qualidade.

Uma partida no ScrumAge cobre todos os eventos Scrum e os papéis do time Scrum, em que o jogador se reveza entre todos os papéis de acordo com a etapa do jogo. Dessa forma, as principais ações do time Scrum (como a priorização, a estimativa e a composição do backlog da Sprint) são realizadas pelo jogador, atuando em um papel específico para cada uma delas.

Para avaliar o protótipo do jogo, 39 profissionais de tecnologia da informação jogaram o ScrumAge e responderam a um formulário de caracterização e a questionários relacionados ao conteúdo, à interface e à usabilidade do sistema. Comentários e sugestões de melhorias também foram coletados e analisados. A análise dos resultados apontou que o jogo teve uma recepção bastante positiva pelos avaliadores.

Logo, este trabalho tem como objetivo geral propor uma abordagem prática complementar para o processo de aprendizagem da metodologia ágil Scrum. Para atingir ao objetivo geral, alguns objetivos específicos foram definidos, entre eles:

- a) Analisar os jogos já existentes com a mesma finalidade, identificando características que possam servir de inspiração e possíveis melhorias.
- b) Listar os principais requisitos que um jogo com esse objetivo deve atender.
- c) Projetar e implementar um jogo para atender a essa finalidade.
- d) Avaliar o jogo com profissionais de tecnologia da informação.

Este trabalho possui 5 capítulos. O capítulo 1 abrange o referencial teórico utilizado, enquanto o capítulo 2 exhibe os trabalhos correlatos obtidos em uma revisão bibliográfica. Já o capítulo 3 apresenta a proposta do jogo ScrumAge, ao passo que o capítulo 4 expõe o desenvolvimento de seu protótipo. Em seguida, o capítulo 5 relata a avaliação realizada com profissionais de tecnologia da informação. Por último, são descritas as considerações finais sobre o trabalho, seguidas da lista de referências e dos apêndices.

1. REFERENCIAL TEÓRICO

1.1. Metodologias Ágeis

Com a constante expansão do mercado de desenvolvimento de softwares, a partir da década de 1990 tornou-se necessária a renovação das metodologias de desenvolvimento, pois os métodos usados até então eram inspirados na engenharia tradicional e já era perceptível que a criação de um software não deveria continuar seguindo completamente essa inspiração. Enquanto obras de engenharia civil, por exemplo, precisam ser planejadas em cada pequeno detalhe antes do início da construção, pois grandes alterações posteriores costumam ser muito dispendiosas ou até mesmo impossíveis, boa parte dos softwares podem ser modificados conforme a necessidade, a um custo relativamente mais baixo.

Diante dessa necessidade de mudança de paradigma, para que os projetos de desenvolvimento de software passassem a utilizar um método mais adequado à sua natureza mutável, as metodologias ágeis foram propostas. O desenvolvimento ágil utiliza uma abordagem de planejamento incremental e muito iterativa. Dessa forma, diferentemente do método tradicional, em que a princípio se define e se documenta detalhadamente todas as fases do início ao fim do projeto, em um método ágil isso é feito em pequenas etapas, também chamadas de iterações.

Um marco para a unificação e expansão das metodologias ágeis foi a elaboração do Manifesto Ágil (BECK, 2001), que é uma declaração de valores e princípios essenciais para o desenvolvimento de software. Ele foi criado em Utah (EUA) em fevereiro de 2001, onde 17 desenvolvedores influentes no meio tecnológico se reuniram interessados em buscar uma alternativa aos processos tradicionais de desenvolvimento. Eles então decidiram documentar a reunião, que se tornou o Agile Manifesto, e se autointitularam como a Aliança Ágil. Os valores descritos no manifesto são os seguintes:

- **Indivíduos e interações** mais que processos e ferramentas
- **Software em funcionamento** mais que documentação abrangente
- **Colaboração com o cliente** mais que negociação de contratos
- **Responder a mudanças** mais que seguir um plano

Alguns exemplos de metodologias ágeis são: Extreme Programming (XP), Feature-Driven Development (FDD), Lean Development, Kanban, e Scrum, descrito a seguir.

1.2. Scrum

O Scrum é um *framework* de desenvolvimento ágil concebido para a tratativa e a resolução de problemas complexos e adaptativos, de maneira produtiva e criativa, visando entregar produtos com o mais alto valor possível (SCHWABER e SUTHERLAND, 2020). Ele é utilizado desde o início da década de 1990 e atualmente, é a metodologia ágil mais utilizada nas equipes de desenvolvimento de software (DIGITAL.AI, 2021).

A fundamentação teórica do Scrum vem do empirismo e do *lean thinking*. O empirismo afirma que o conhecimento vem da experiência e da observação, enquanto o *lean thinking* reduz o desperdício e foca no que é essencial. Dessa forma, o Scrum emprega uma abordagem iterativa e incremental para aperfeiçoar o controle e a previsibilidade de riscos. Seus pilares são: (i) a transparência, que diz que todo o trabalho deve ser visível para quem o executa e para quem o recebe; (ii) a inspeção, que se torna possível a partir da transparência, visando detectar variações ou problemas no andamento do projeto; e (iii) a adaptação, necessária quando algum desvio é encontrado na inspeção (SCHWABER e SUTHERLAND, 2020).

O Scrum tem como valores coragem, foco, comprometimento, respeito e abertura. A partir desses princípios, o time Scrum se compromete a apoiar uns aos outros e a alcançar seus objetivos. Três fases são adotadas no Scrum: Planejamento, Sprints e Encerramento. Uma Sprint é um período em que um incremento do software usável é completamente desenvolvido. O time é a unidade fundamental do Scrum, sendo constituído de uma pequena equipe de pessoas: um Scrum master, responsável por estabelecer e garantir a correta utilização do Scrum; um product owner, responsável por maximizar o valor do produto resultante do trabalho do time Scrum; e desenvolvedores, responsáveis por criarem a cada Sprint qualquer parte de um incremento utilizável (SCHWABER e SUTHERLAND, 2020).

Dentro do framework Scrum, existem cinco eventos que criam uma regularidade das tarefas e reuniões. Cada um desses eventos é uma oportunidade para inspecionar e adaptar os artefatos do Scrum. Os eventos do Scrum são (SCHWABER e SUTHERLAND, 2020):

- **Sprint:** possui duração fixa de um mês, ou menos. Cada Sprint possui um objetivo próprio, para o qual o esforço do time Scrum deve estar voltado. Todo o trabalho necessário para atingir o objetivo do produto é realizado dentro das Sprints;
- **Planejamento da Sprint:** inicia a Sprint, onde o Time Scrum define: (i) qual é o objetivo da Sprint; (ii) a estimativa das próximas atividades do backlog do Produto; (iii) quais itens do Backlog do Produto devem ser selecionados para o Backlog da Sprint;
- **Reunião Diária:** evento de 15 minutos para que o Time Scrum inspecione o progresso em direção ao Objetivo da Sprint e adapte o Backlog da Sprint conforme necessário, melhorando a comunicação, identificando impedimentos e promovendo a agilidade na tomada de decisões;
- **Revisão da Sprint:** tem como objetivo a inspeção do resultado da Sprint, apresentado pelo Time Scrum para as demais partes interessadas pelo projeto;
- **Retrospectiva da Sprint:** visa planejar maneiras de aumentar a qualidade e a eficácia, inspecionando como foi a última Sprint em relação aos indivíduos, interações, processos, ferramentas e sua definição de pronto. O Time Scrum discute o que correu bem ou mal durante a Sprint, comentando sobre problemas encontrados.

Embora o Scrum não especifique nenhum método para realizar a estimativa das tarefas durante a reunião de planejamento da Sprint, muitas vezes o Planning poker é utilizado. Esta técnica busca estimar cada atividade por meio da colaboração de todos os desenvolvedores com suas estimativas pessoais, em que cada desenvolvedor vota usando uma carta contendo o valor correspondente à sua estimativa (MAHNIČ, 2012).

Os artefatos do Scrum existem para representar o trabalho ou o valor, provendo transparência, oportunidade de inspeção e adaptação. Cada artefato possui um compromisso para garantir que ele forneça informações que aumentem a transparência e o foco em relação ao qual o progresso pode ser medido. Os artefatos são (SCHWABER e SUTHERLAND, 2020):

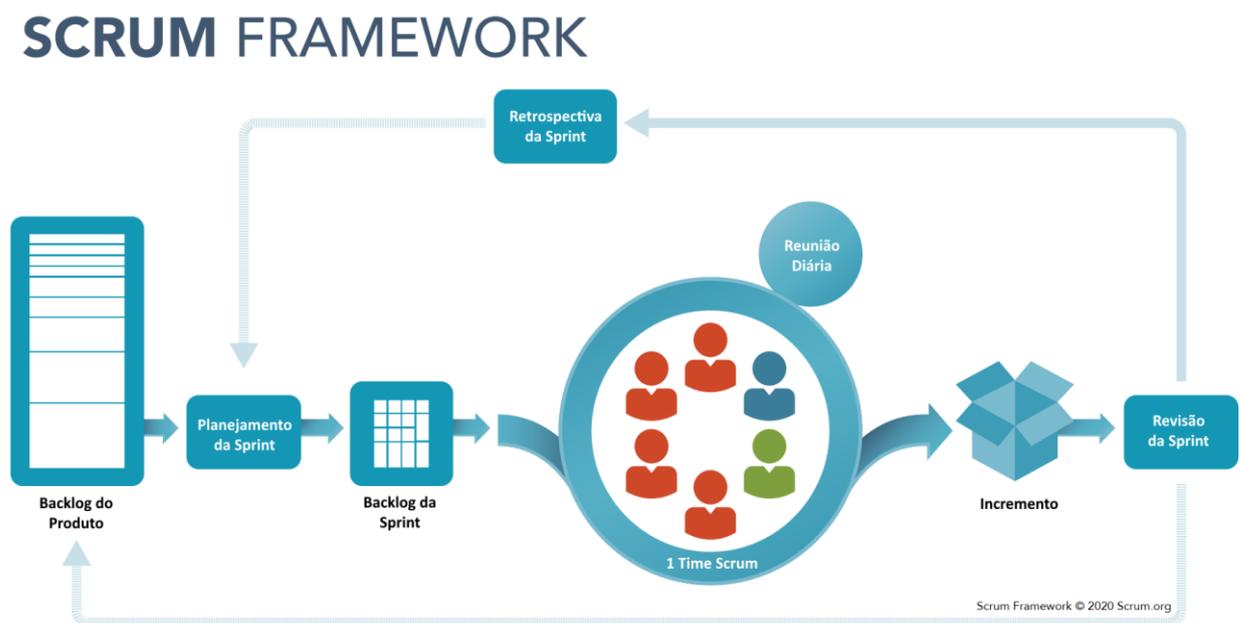
- **Backlog do Produto:** lista ordenada do que é necessário para melhorar o produto, sendo a única fonte do trabalho realizado pelo Time Scrum. Seu compromisso é o objetivo do produto, que descreve um estado futuro do produto, servindo como alvo para o Time Scrum;

- **Backlog da Sprint:** é composto pelo objetivo da Sprint (porque), o conjunto de itens do backlog do produto selecionados para a Sprint (o quê) e um plano acionável para a entrega do incremento (como). É uma imagem em tempo real e altamente visível do trabalho que deve ser realizado durante a Sprint. Seu compromisso é o objetivo da Sprint, gerando coerência e foco para o time Scrum trabalhar em conjunto;
- **Incremento:** é um trampolim concreto em direção ao objetivo do produto. Cada incremento é um aditivo a todos os incrementos anteriores, devendo ser utilizável. O trabalho só pode ser considerado parte de um incremento se atender ao seu compromisso, que é a definição de pronto: uma descrição formal do estado do incremento que atende aos critérios de qualidade exigidos para o produto.

Para visualizar o andamento das atividades presentes no backlog da Sprint, muitas vezes um quadro Kanban é utilizado. O Kanban é considerado um sistema para visualizar o trabalho, fazendo ele fluir, reduzindo o desperdício e maximizando o valor para o cliente (CORONA e PANI, 2013). Usualmente, é utilizado um quadro com três raias: fazer, fazendo e feito. Essas raias abrigam a descrição das atividades que estão em cada um desses estados, possibilitando uma rápida visualização da situação atual do trabalho em evidência.

A figura 1 ilustra o funcionamento geral do Scrum, englobando os eventos, artefatos e o time Scrum.

Figura 1 – O framework Scrum



Fonte: adaptado de SCRUM.ORG (2020)

1.3. Serious Games em Engenharia de Software

Jogos, além de terem como objetivo o entretenimento, podem ser utilizados para treinamento, publicidade, simulação, ou educação, sendo estes denominados de Serious Games, ou Jogos Sérios (SUSI et al., 2007). De acordo com Corti (2006), os jogos sérios "são todos que têm a vantagem do poder dos jogos de computador para cativar e engajar usuários finais para um propósito específico, como para desenvolver novos conhecimentos e habilidades". Zyda (2005) propõe uma definição mais formal: "Uma disputa mental, jogada com um computador em acordo com regras específicas, que utiliza entretenimento para promover treinamentos, educação, saúde, políticas públicas ou objetivos de comunicação estratégica de agências governamentais ou corporações."

Jogos e ambientes simulados permitem que estudantes experienciem situações que, por razões de segurança, custo, tempo etc., seriam impossíveis no mundo real (CORTI, 2006). Nesse contexto, os jogos sérios de simulação destacam-se por permitir a obtenção de conhecimentos teóricos em conjunto com a prática, embora muitas vezes esta seja limitada por não levar em consideração todos os inúmeros fatores, que podem impactar um cenário real.

Além disso, diversas análises já foram feitas, frequentemente mostrando que jogos promovem o aprendizado (SUSI et al., 2007). Tratando especificamente de serious games para o ensino de metodologias ágeis, Rodríguez (2021) realizou uma revisão da literatura que indicou que esses jogos parecem servir como os instrumentos de suporte mais bem sucedidos para complementar outros métodos de ensino.

Restringindo ainda mais o escopo, abordando apenas o ensino da metodologia ágil Scrum, diversos trabalhos obtiveram resultados positivos, entre eles alguns simuladores, conforme pode ser visto na seção de revisão bibliográfica. Sendo assim, podemos concluir que a utilização de serious games para o ensino do Scrum é uma abordagem com grande potencial para tornar o aprendizado dessa metodologia mais acessível e eficiente.

2. REVISÃO BIBLIOGRÁFICA

Buscando identificar quais serious games já haviam sido criados com a finalidade de auxiliar o aprendizado do Scrum, uma revisão bibliográfica foi realizada. A partir dessa pesquisa foram encontrados e selecionados para análise 10 trabalhos com essa proposta.

2.1. Planejamento e Condução

Antes da busca ser iniciada, foram formuladas uma questão primária (QP), que denota o objetivo principal do estudo, e 11 questões secundárias (QS), que auxiliam uma melhor compreensão do panorama existente, de acordo com alguns parâmetros:

- **QP:** Quais serious games já foram criados com a finalidade de auxiliar o ensino do Scrum desde 2010?
- **QS1:** Em qual ano o artigo foi publicado?
- **QS2:** Qual é a quantidade necessária de jogadores?
- **QS3:** Qual é o gênero do jogo?
- **QS4:** O jogo foi feito para qual plataforma?
- **QS5:** Como funciona a pontuação ou a vitória no jogo?
- **QS6:** O jogo aborda outros métodos ágeis?
- **QS7:** O jogo permite personalização?
- **QS8:** O jogo possui algum sistema de classificação de jogadores?
- **QS9:** O jogo permite que o jogador atue em todos os papéis do time Scrum?
- **QS10:** Quantos participantes realizaram a avaliação?
- **QS11:** Qual foi o resultado da avaliação?

A QS1 foi definida visando verificar qual é a tendência na elaboração de trabalhos sobre o tema nos últimos anos. Já as questões secundárias de 2 a 9 visam entender quais são as principais características dos jogos desenvolvidos nos trabalhos apresentados. Por fim, as questões QS10 e QS11 buscam retratar a abrangência das avaliações e as percepções dos avaliadores.

Com as questões escolhidas, seguiu-se com a definição do termo de busca. O objetivo da pesquisa era encontrar artigos em inglês ou em português, sendo utilizado o termo de busca

nesses dois idiomas. Alguns termos diferentes foram testados, onde os listados abaixo trouxeram os melhores resultados:

- **Em inglês:** (“scrum” OR “agile method”) AND (“serious game” OR “educational games”) AND (“learning” OR “teaching”)
- **Em português:** (“scrum” OR “método ágil”) AND (“serious game” OR “jogo sério” OR “jogo educativo”) AND (“aprendizado” OR “ensino”)

Para guiar a seleção dos artigos, foram utilizados 4 critérios de inclusão e 1 critério de exclusão. Dentre os critérios de inclusão, optou-se por todos os trabalhos desde 2010 porque a quantidade de artigos retornados na pesquisa antes dessa data foi irrisória.

- **Crítérios de Inclusão (CIs):**
 - **CI1:** Artigos devem estar escritos em português ou inglês.
 - **CI2:** Artigos publicados entre 2010 e 2022.
 - **CI3:** Artigos disponíveis para acesso livre em bibliotecas digitais.
 - **CI4:** Todos os artigos que desenvolvem, avaliam ou descrevem jogos com a finalidade de auxiliar o ensino do Scrum.
- **Crítérios de Exclusão (CEs):**
 - **CE1:** A existência de um artigo mais recente ou mais completo (tratando do mesmo trabalho) elimina o artigo anterior.

Em seguida, optou-se por utilizar o buscador Google Scholar (2022) para realizar a pesquisa. Ele foi escolhido por retornar resultados de diversas bases de trabalhos acadêmicos, permitindo a centralização da busca usando uma única ferramenta. O ponto negativo deste buscador é o grande volume de trabalhos retornados, que não possuem relação com o tema procurado, principalmente após as primeiras páginas da navegação. Para minimizar esse problema, a pesquisa foi interrompida na página 20 da busca de ambos os termos, limite onde já se havia passado algumas páginas sem nenhum trabalho considerado relevante para a resposta da questão primária.

2.2. Análise dos Resultados da Revisão

Através da pesquisa realizada, 10 trabalhos foram encontrados: 7 com o termo em inglês e 3 com o termo em português. As respostas às questões de pesquisa, obtidas após a análise de todos os trabalhos citados, estão disponíveis na tabela 1. A seguir, encontra-se a análise dos resultados.

Tabela 1 – Comparativo entre trabalhos correlatos encontrados na revisão da bibliografia (continua)

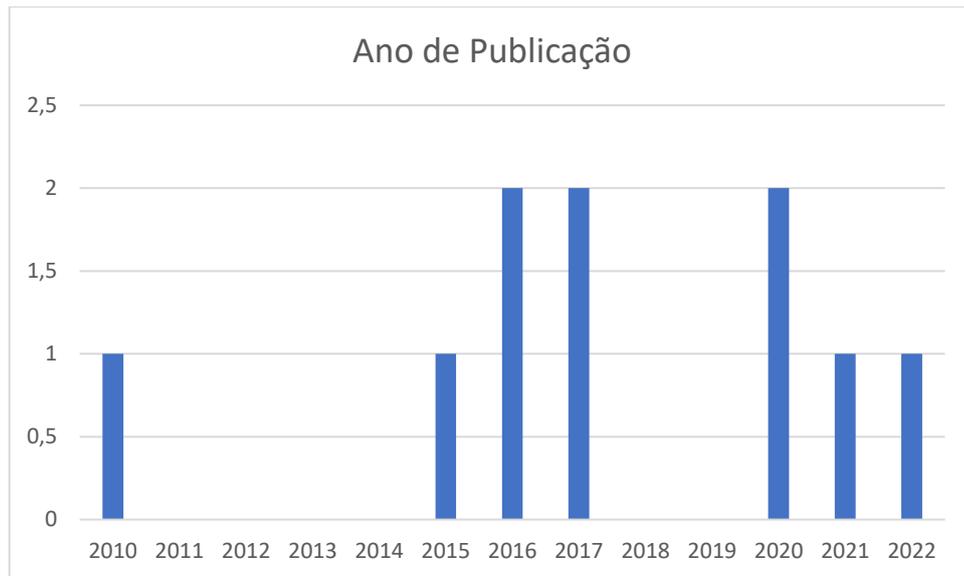
Trabalho	Ano	Jogadores	Gênero	Plataforma	Pontuação	Outros Métodos	Personalização	Classificação	Papéis do Scrum	Participantes	Resultado
Scrum Xperience: um protótipo de jogo voltado para o ensino de metodologias ágeis (ARAGÃO, 2022)	2022	3	Simulador	Desktop	De acordo com a quantidade de erros e acertos da equipe	XP	Configuração do jogo	Sim	Escolher entre SM, PO ou TD	30 alunos e 6 professores	Positivo
Comparing Different Serious Games Strategies for Teaching the SCRUM Framework (SANTOS et al., 2021)	2021	1	Runner / Sobrevivência	Desktop	De acordo com o tempo, quantidade de erros e de acertos	Não	Não	Não	Não	73 alunos	Positivo
ScrumGame: A Serious Game to Initiate Software Trainees in Scrum (RODRÍGUEZ, 2020)	2020	1	Quiz	Mobile	Não possui	Não	Não	Não	Não	10 profissionais	Positivo
Scrumie: Scrum Teaching Agent Oriented Game (MARINHO et al., 2020)	2020	1	Simulador	Desktop	De acordo com o prazo e o orçamento restantes	Não	Não	Não	Não	23 alunos	Regular
AGILITY SCRUM-Um Jogo para Ensino da Metodologia SCRUM (DE CASTRO et al., 2017)	2017	4 a 6	Protobord	Físico	Ganha a equipe que tiver finalizado as sprints e esteja com o circuito funcionando	Kanban	Não	Não	Jogador: escolher SM ou TD Professor: PO	26 alunos	Positivo

Tabela 1 – Comparativo entre trabalhos correlatos encontrados na revisão da bibliografia (conclusão)

Trabalho	Ano	Jogadores	Gênero	Plataforma	Pontuação	Outros Métodos	Personalização	Classificação	Papéis do Scrum	Participantes	Resultado
SCRUMI: A Board serious virtual game for teaching the SCRUM framework (DE SOUZA et al, 2017)	2017	1	Tabuleiro	Desktop	De acordo com o tempo para responder as perguntas, acertos e erros	Não	Configuração das fases do projeto	Não	Não	34 alunos	Positivo
SCRUM-X: An interactive and experiential learning platform for teaching Scrum (LEE, 2016)	2016	3	Simulador	Desktop	Ganha o time que finalizar o projeto em menos sprints	Não	Não	Não	Não	15 alunos	Positivo
SCRUM-Scape: Jogo educacional de Role-Playing Game (RPG) para ensinar SCRUM (BATTISTELLA et al., 2016)	2016	1	RPG	Desktop	De acordo com a quantidade de acertos	Não	Não	Não	Não	10 alunos e 7 profissionais	Positivo
Virtual Scrum: A teaching aid to introduce undergraduate software engineering students to Scrum (RODRÍGUEZ, 2015)	2015	7	Simulador	Desktop	Equipe que terminar o projeto com menos erros	Não	Não	Não	Escolher entre SM, PO ou TD	45 alunos	Positivo
PlayScrum - A card game to learn the scrum agile method (FERNANDES et al., 2010)	2010	2 a 5	Tabuleiro	Físico	Ganha a equipe que tiver finalizado todas as tarefas primeiro ou a que finalizar mais tarefas	Não	Não	Não	Scrum Master	13 alunos	Positivo

Em relação ao ano de publicação, foi encontrado apenas 1 trabalho publicado no ano de 2010, onde os demais foram publicados entre 2015 e 2022. Neste período, os anos com mais trabalhos publicados foram 2016, 2017 e 2020, com 2 publicações em cada, seguidos de 2015, 2021 e 2022, com uma publicação. A figura 2 mostra essa distribuição ao longo dos anos.

Figura 2 – Trabalhos correlatos – Ano de publicação



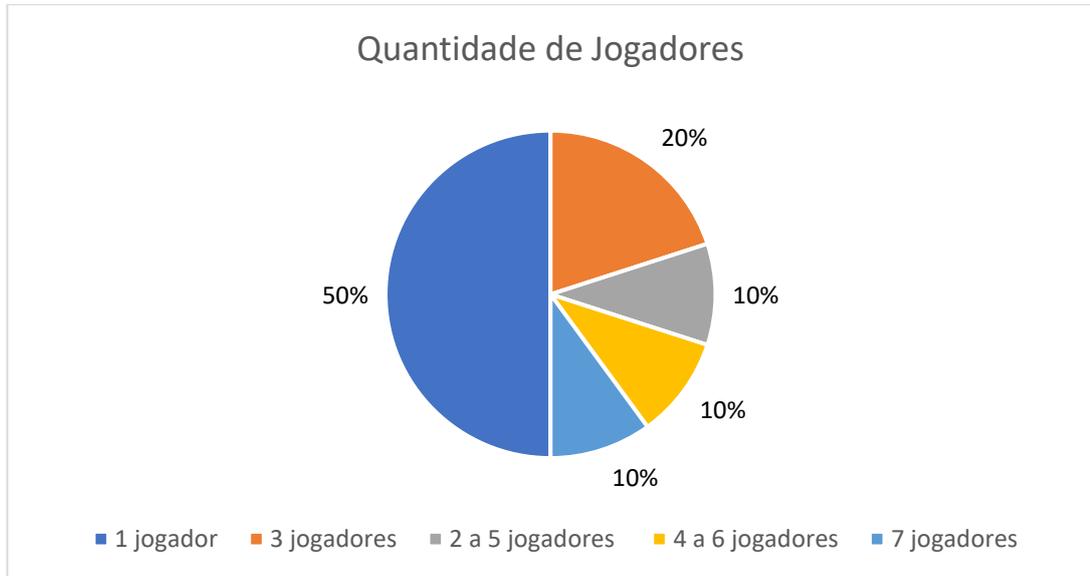
Fonte: Autor

Sobre a quantidade de jogadores, a figura 3 exibe os resultados obtidos. Metade dos jogos foram feitos para serem jogados individualmente, enquanto 2 foram feitos para serem jogados exatamente com 3 jogadores cada. Dentre os demais, um é feito para ser jogado com entre 2 e 5 jogadores, outro aceita entre 4 e 6 participantes, e outro com exatamente 7 pessoas.

Os jogos dos trabalhos encontrados dividem-se entre 7 gêneros, porém um desses jogos (SANTOS et al, 2021) é composto por dois jogos menores, de gêneros diferentes, onde para a pesquisa foi considerado como um único jogo pertencente a dois gêneros: runner e sobrevivência. Além deste, há 1 jogo de RPG, outro de quiz e outro realizado em uma protoboard física para circuitos elétricos. Os gêneros que tiveram maior frequência foram 2 jogos de tabuleiro e 4 simuladores. A figura 4 mostra essa distribuição.

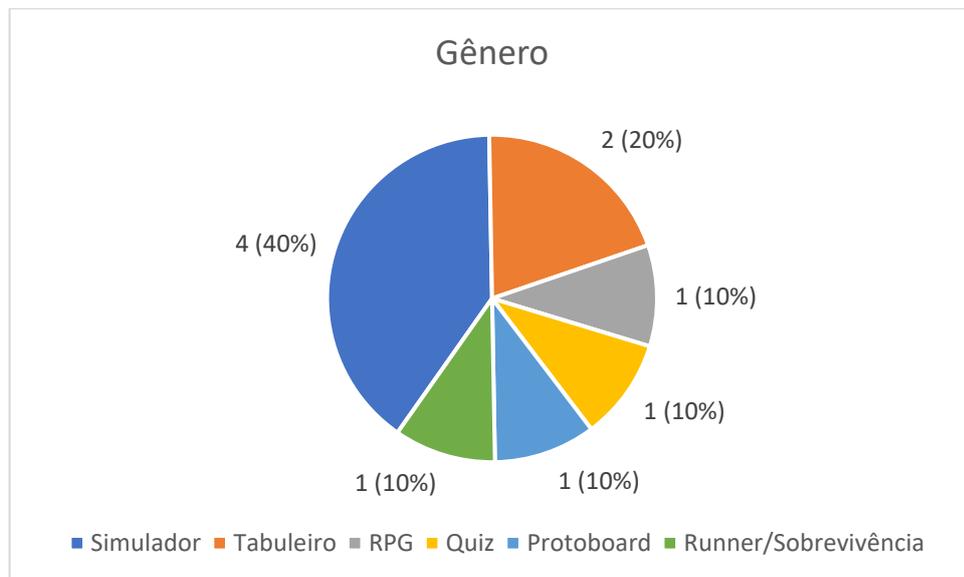
Analisando a plataforma para qual cada jogo foi disponibilizado, verificou-se que a maioria deles (70%) foram feitos exclusivamente para desktops, enquanto 2 são jogos físicos e 1 é exclusivo para dispositivos móveis que utilizam o sistema operacional iOS. A figura 5 traz o gráfico que ilustra esses números.

Figura 3 – Trabalhos correlatos – Quantidade de jogadores



Fonte: Autor

Figura 4 – Trabalhos correlatos – Gênero



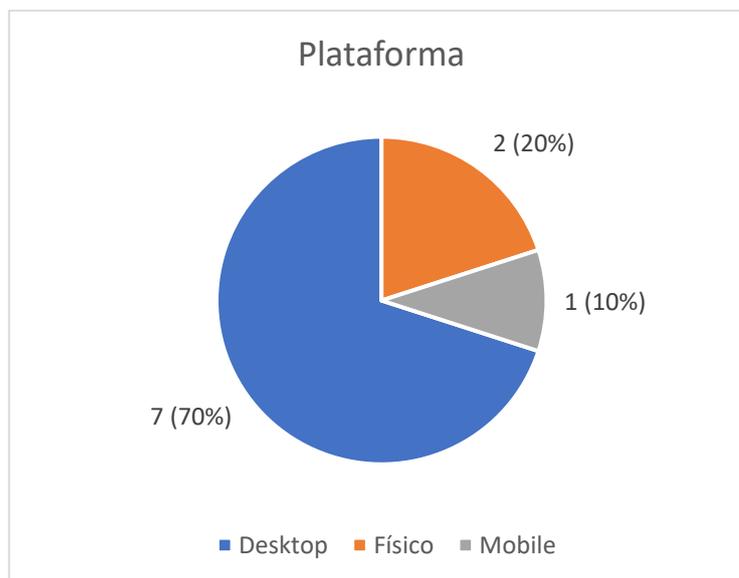
Fonte: Autor

Abordando os detalhes mais específicos dos jogos, a pontuação ou a vitória é definida de forma distinta em cada um dos trabalhos. Além disso, somente dois jogos abordam outras metodologias além do Scrum: Aragão (2022) aborda a metodologia ágil XP e De Castro et al. (2017) utiliza o Kanban.

Sobre a possibilidade de personalização, apenas Aragão (2022) e De Souza et al. (2017) trazem essas funcionalidades, exclusivamente para usuários administradores. Dentre

todos os jogos verificados, apenas Aragão (2022) possui um sistema de classificação, para que os jogadores possam verificar as pontuações uns dos outros e competir para ficar nas primeiras colocações.

Figura 5 – Trabalhos correlatos – Plataforma



Fonte: Autor

Em relação à possibilidade de que o jogador atue em todos os papéis do Scrum, somente Aragão (2022) permite essa escolha. Rodríguez (2015) e De Castro et al. (2017) possibilitam aos alunos a escolha apenas entre os papéis de Scrum master e time de desenvolvimento, enquanto o professor atua como product owner. Já no trabalho de Fernandes et al. (2010) o jogador atua exclusivamente no papel de Scrum master. Os demais trabalhos não relacionaram as ações do jogador com nenhum papel do time Scrum em específico.

As avaliações de cada trabalho foram realizadas por entre 10 e 73 participantes, onde 7 indicaram a participação apenas de alunos, 1 relatou a participação de alunos e professores, 1 informou a participação de alunos e profissionais, e 1 indicou a participação apenas de profissionais. Dessa forma, 8 jogos foram avaliados unicamente no meio acadêmico, onde os dois outros tiveram uma baixa amostragem de profissionais: 10 em Rodríguez (2020) e 7 em Battistella et al. (2016).

Dentre todas as informações obtidas, destaca-se o resultado positivo que todos os trabalhos obtiveram em suas avaliações, com exceção de um (MARINHO et al., 2020), que

indicou um resultado regular no único teste conduzido, referente à usabilidade do sistema. Entretanto, mesmo este trabalho relatou que o jogo agradou os avaliadores, de maneira geral.

Também merecem ser mencionadas algumas características encontradas nos trabalhos correlatos que inicialmente não haviam sido abordadas pelas questões de pesquisa. No trabalho de Aragão (2022), há a possibilidade do usuário administrador (professor) cadastrar as escolhas a serem realizadas por cada papel do time Scrum em cada um dos eventos Scrum ao longo da partida, além de permitir comentar sobre o desempenho individual de cada aluno.

No trabalho de Rodríguez (2015), o *planning poker* (MAHNIČ, 2012) é utilizado para a realização das estimativas das atividades. Já em Rodríguez (2020), o progresso do jogador é salvo a cada etapa finalizada da partida. Por último, o trabalho de Marinho et al. (2020) foi desenvolvido orientado a agentes, que adicionaram inteligência ao jogo.

2.3. Considerações Finais da Revisão

A partir dos dados analisados, pode-se concluir que os resultados obtidos por jogos para o aprendizado do Scrum têm sido bastante positivos, o que serve como justificativa para novos trabalhos com finalidades similares e propostas diferenciadas. Nesse contexto, como consequência complementar da pesquisa realizada, foi possível identificar lacunas dos trabalhos relatados.

A mais notável delas foi a inexistência de um jogo multiplataforma, acessível por diversos tipos de dispositivos. Dentre os jogos digitais analisados, todos foram desenvolvidos para serem jogados em desktops, com exceção de Rodríguez (2020), criado exclusivamente para dispositivos móveis, que utilizem o sistema operacional iOS, o que já é uma limitação. Considerando que em 2020 cerca de 71% dos acessos à internet no mundo foram realizados por dispositivos móveis, proporção que vem aumentando ao longo dos últimos anos (ENGE, 2021), fica evidente a relevância de também existirem jogos com a mesma finalidade para estes dispositivos.

Outra lacuna identificada foi a falta de uma base de conhecimento teórico disponibilizada internamente ao sistema, onde o jogador possa acessar quaisquer informações sobre o jogo ou sobre o Scrum a qualquer momento. Apenas Aragão (2022) permite que usuários administradores cadastrem conteúdos didáticos, que são exibidos em pontos específicos durante a partida, mas que não podem ser acessados fora do contexto previamente

estabelecido. Sendo assim, o jogador precisa recorrer a recursos externos para elucidar qualquer dúvida que surja, seja a um professor, ou a algum material escrito.

Observou-se também que, mesmo entre os jogos do gênero simulador, nenhum desses trabalhos apresentou uma quantidade significável de conceitos complementares, que ajudem a representar outros aspectos e desafios do desenvolvimento de projetos de software, capazes de maximizar o realismo. Eles se limitaram a abordar conteúdos sobre o Scrum, sem considerar temas como a qualidade de cada tarefa finalizada em relação à senioridade do desenvolvedor, as ações que podem ser tomadas para o aumento dessas qualidades, a produtividade do time de desenvolvimento, ou a satisfação do cliente que contratou o projeto. E esses são fundamentais para determinar o sucesso de um projeto.

Em relação aos três papéis do time Scrum, a maioria dos jogos encontrados não permite que o jogador atue diretamente em um ou mais papéis. Apenas o jogo de Aragão (2022) possibilita que o jogador escolha entre essas atribuições, porém apenas uma delas pode ser interpretada por partida. Por este motivo, caso o jogador deseje atuar em cada uma das dessas funções para melhor entender quais são suas responsabilidades, ao menos três partidas precisam ser jogadas. Considerando o contexto de uma aula, o tempo necessário para jogar todas essas partidas pode ser um fator impeditivo para a adoção e a disseminação desse jogo. Sendo assim, buscando que o jogador pratique e vivencie as responsabilidades de cada papel, considera-se relevante que isso seja possível em uma única partida, onde o jogador revezaria entre os papéis do time Scrum ao longo do jogo.

Notou-se também que nenhum dos jogos permite que o jogador realize, no mesmo jogo, a priorização, a estimativa e a seleção das tarefas que irão compor o backlog da Sprint. Considerando que essas são as principais atividades da reunião de planejamento da Sprint, servindo como fatores determinantes para que o objetivo da Sprint seja alcançado até o final da iteração, seria interessante que o jogador tivesse a possibilidade de participar de todas essas ações.

Sobre a avaliação, verificou-se que apenas dois dos jogos foram avaliados por profissionais, e em uma quantidade bastante restrita. Dessa forma, conclui-se que um trabalho com a mesma finalidade, mas a ser avaliado por profissionais da indústria ao invés de alunos e professores, poderia trazer informações complementares relevantes. Além disso, abriria portas

para que o jogo também fosse utilizado em treinamentos em empresas, não apenas em salas de aula nas universidades.

Por fim, é possível chegar à conclusão de que diversas abordagens podem ser utilizadas na criação de um serious game com o propósito descrito, havendo espaço para mais experimentos do gênero. A revisão realizada foi capaz de identificar características importantes (presentes ou faltantes) nos jogos avaliados, embasando a escolha dos requisitos a serem utilizados no ScrumAge, visando tornar o jogo mais interessante e eficiente no cumprimento de sua proposta.

3. PROPOSTA: SCRUMAGE

Após a finalização da revisão bibliográfica sobre serious games para o aprendizado de Scrum, algumas das características dos trabalhos correlatos foram reunidas, em conjunto com as possíveis melhorias identificadas nas considerações finais da revisão. Em seguida, esses itens foram analisados e priorizados, de forma a delimitar um escopo inicial para a proposta de um jogo, transformando-se na lista de requisitos do ScrumAge. Todos os itens nela contidos visam tornar este jogo mais eficiente para o aprendizado do Scrum do que seus similares.

A primeira seção deste capítulo lista os requisitos funcionais e não funcionais do sistema. Já a segunda seção mostra o planejamento e a modelagem realizada. Por fim, a terceira seção exibe uma visão geral do jogo, incluindo seu objetivo e seus conceitos mais importantes.

3.1. Requisitos

Segundo Wieggers e Beatty (2013), requisitos são uma especificação do que deve ser implementado em um sistema, podendo atuar como uma restrição ao processo de desenvolvimento, ao descrever como o sistema deve se comportar, ou sobre quais propriedades ou atributos devem ter. Eles são úteis para listar de maneira simples e objetiva o que se espera do sistema após o término de sua construção.

3.1.1. Requisitos não funcionais

Requisitos não funcionais são aqueles que ditam as propriedades comportamentais do sistema (WIEGERS E BEATTY, 2013), indicando padrões e direcionando o desenvolvimento. Dentre os requisitos não funcionais do ScrumAge, destacam-se:

- **RN1: O jogo deve estar disponível na forma de um website:** com isso, poderá ser acessado de qualquer lugar com acesso à internet e a qualquer momento, não necessitando de qualquer outra instalação ou infraestrutura específica para ser acessado;
- **RN2: O jogo deve ser acessível tanto por dispositivos móveis quanto por computadores de mesa:** como é desejado que o ScrumAge possa ser utilizado também fora da sala de aula e do meio corporativo, por qualquer pessoa que se

interesse pelo aprendizado do Scrum, o acesso a partir de dispositivos móveis também deve ser possível e proporcionar uma boa experiência de uso;

- **RN3: O jogo deve ser *single player*:** dessa forma qualquer pessoa poderá jogar o ScrumAge sem precisar formar um grupo para isso;
- **RN4: O jogo deve apresentar uma linguagem clara e atrativa:** visa tornar o jogo mais divertido e interessante;
- **RN5: O jogo deve ter uma interface simples, intuitiva e agradável:** dessa forma a quantidade de textos será reduzida e a experiência do jogador fluirá de maneira mais natural;
- **RN6: O jogo deve atender a critérios usuais de qualidade técnica:** dentre esses critérios, destacam-se: (i) a segurança dos dados dos usuários; (ii) ter um tempo de resposta aceitável para cada ação do jogador; (iii) ser fácil de ser alterado e evoluído, conforme a necessidade.

3.1.2. Requisitos funcionais

Os requisitos funcionais representam quais ações o sistema deve permitir que seus usuários realizem (WIEGERS E BEATTY, 2013). No ScrumAge, dentre os principais requisitos funcionais, encontram-se:

- **RF1: O jogo deve simular o desenvolvimento de um projeto de software por uma equipe utilizando o Scrum:** chegou-se à conclusão de que o jogo deveria ser da categoria simulação, provendo a imersão do jogador em uma equipe de desenvolvimento utilizando o Scrum;
- **RF2: O jogo deve permitir que o jogador retorne ao ponto que parou a qualquer momento:** com isso o jogador não perderá seu progresso e poderá retornar ao jogo quando achar melhor;
- **RF3: O jogo deve disponibilizar um quadro Kanban para a visualização do andamento das atividades do Backlog da Sprint:** por ser um conceito amplamente utilizado em conjunto com o Scrum (DIGITAL.AI, 2021), é importante que seja apresentado através do jogo;
- **RF4: O jogo deve possuir um repositório de conhecimento sobre suas regras, sobre o Scrum, e sobre conceitos complementares, sendo facilmente acessível a qualquer momento:** através dessa Wiki o jogador poderia consultar um material feito para responder a maior parte das dúvidas que poderiam surgir durante o jogo, seja em

relação ao seu funcionamento ou ao Scrum, além de disponibilizar um guia rápido de consulta sobre o Scrum mesmo fora de uma partida;

- **RF5: O jogo deve fazer com que o jogador reveze entre os três papéis do Time Scrum:** dessa forma o jogador poderá aprender sobre as responsabilidades e obrigações de cada um dos três papéis do time Scrum;
- **RF6: O jogo deve ter um sistema de pontuação e de classificação de partidas:** sendo assim, o engajamento no jogo tende a ser aumentado, uma vez que o jogador competirá diretamente com os demais, em busca de conseguir uma melhor classificação no ranking.

Conforme proposto pelos métodos ágeis, o desenvolvimento do ScrumAge iniciou-se a partir desta lista enxuta de requisitos, onde os demais foram sendo descobertos e implementados ao longo do processo de desenvolvimento, descrito a seguir.

3.2. Planejamento e Modelagem

Com a lista de principais requisitos definida, o iniciou-se a elaboração da modelagem e do planejamento do jogo. Primeiramente, definiu-se que a duração média de uma partida deveria ser de aproximadamente 30 minutos, considerando que o jogador já tenha algum conhecimento teórico prévio sobre o Scrum. Essa duração foi escolhida para que o jogo possa ser utilizado dentro de um tempo de aula usual (que costuma variar de 40 a 50 minutos), após uma aula de introdução ao Scrum. Caso o jogador não tenha nenhum conhecimento sobre métodos ágeis ou sobre o Scrum, é esperado que ele faça um maior uso da Wiki, o que pode estender a partida.

Observando a duração média pretendida para cada partida, foram definidas quais ações necessárias para o desenvolvimento do projeto simulado deveriam ser realizadas pelo jogador e quais deveriam ser realizadas automaticamente pelo sistema. Caso a duração esperada da partida fosse desconsiderada, a princípio seria possível que o jogador fosse responsável por praticamente todas as ações necessárias durante a execução do Scrum. Porém, verificou-se que o jogo provavelmente se tornaria muito mais difícil, monótono e repetitivo, ao menos no formato em que estava sendo projetado.

Em conjunto com a concepção das atividades a serem realizadas pelo jogador, foram definidos quais impactos positivos ou negativos elas poderiam causar no andamento do

projeto simulado, sempre buscando uma aproximação com a realidade de projetos de desenvolvimento software. Para ser possível calcular a pontuação do jogador ao final da partida, esses impactos causados pelas ações do jogador foram traduzidos em métricas, que o jogador pode acompanhar ao longo da partida e verificar se suas decisões estão tendo bons ou maus resultados. Mais detalhes sobre essas métricas são abordados na seção sobre visão geral do ScrumAge e no capítulo sobre o protótipo do jogo.

Uma vez definidas as ações que o jogador deveria realizar, seus impactos e suas métricas de avaliação, a organização geral da aparência e usabilidade do jogo foi escolhida. Isso inclui quais seriam as telas do jogo, como seria a navegação entre elas e como as informações sobre o ScrumAge seriam apresentadas ao jogador.

Em seguida, o fluxo do jogo foi elaborado, baseado no caminho de desenvolvimento proposto pelo Scrum (figura 1). Este fluxo está descrito nos diagrama de atividades UML (Unified Modeling Language), descritos na figura 6, enquanto as figuras 7, 8 e 9 detalham a reunião de planejamento de Sprint, as reuniões diárias e as reuniões de final de Sprint (revisão e retrospectiva), respectivamente. A única etapa definida que não é explicitamente descrita no Scrum é a escolha do time de desenvolvimento, que foi inserida como uma parametrização extra, para trazer mais dinâmica ao jogo.

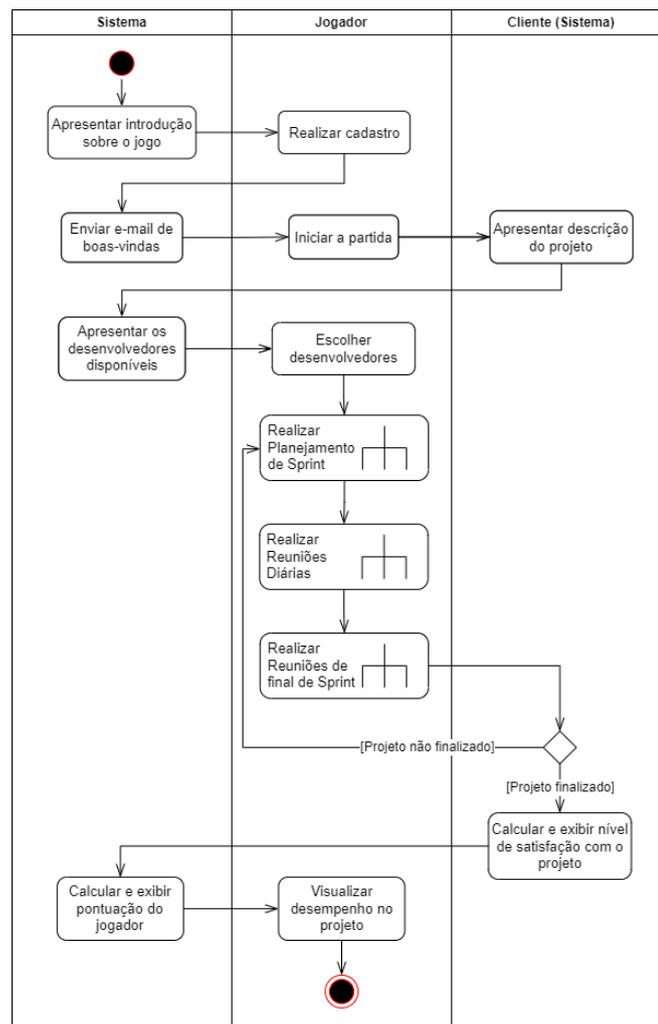
No fluxo central descrito na figura 6, o sistema inicia o fluxo ao apresentar uma introdução sobre o jogo. Em seguida o jogador deve realizar o cadastro, recebendo um e-mail de boas-vindas enviado pelo sistema. Após o cadastro, o cliente (sempre controlado pelo sistema) apresenta a descrição do projeto a ser desenvolvido. Em sua primeira ação do jogo, o jogador deve escolher os desenvolvedores, para então prosseguir com o desenvolvimento do projeto. Todo o desenvolvimento ocorre ao longo de Sprints, em que todas as demais ações do jogador são realizadas (diagramas presentes nas figuras 7, 8 e 9).

Assim como ilustrado na figura 7, o cliente inicia a reunião de planejamento de Sprint indicando as suas prioridades. A partir delas, o product Owner (controlado pelo sistema), apresenta o objetivo da Sprint e as tarefas mais prioritárias. Logo depois, esse papel é assumido pelo jogador, que precisa priorizar as tarefas listadas. As tarefas priorizadas são então estimadas pelo time de desenvolvimento, também controlado pelo jogador nessa etapa. Como última ação do jogador nessa reunião, o Scrum master escolhe o backlog da Sprint. Por

fim, o papel de Scrum master é assumido pelo sistema, que realiza a alocação de tarefas para os desenvolvedores ociosos.

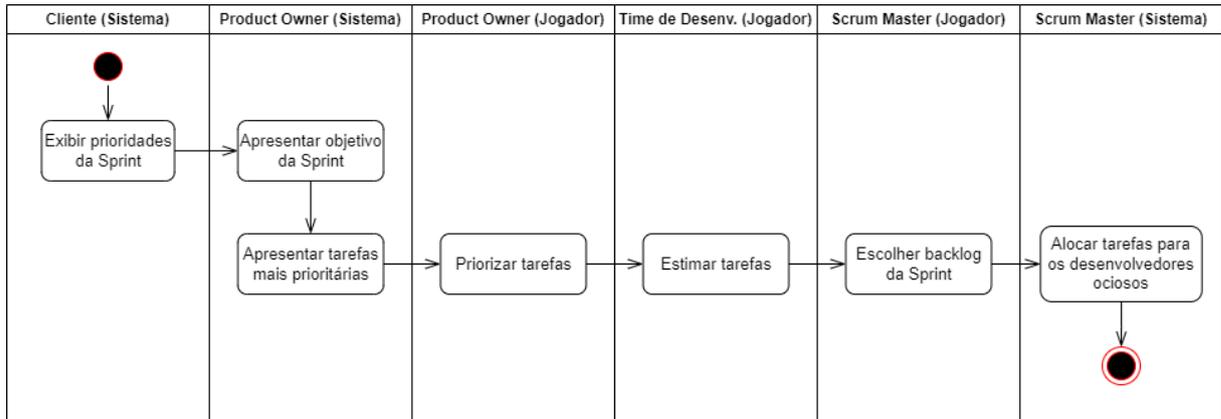
As atividades realizadas na etapa de reuniões diárias são descritas na figura 8. O sistema inicia a reunião avançando o projeto em 1 dia. Assumindo o papel do time de desenvolvimento, ele finaliza as tarefas que estejam em seu último dia. Se houver alguma tarefa com qualidade baixa, o jogador assume o papel de Scrum master para realizar alguma decisão sobre melhorar sua qualidade. O sistema então atualiza o progresso da Sprint, passando a representar o Scrum master para verificar se alguma tarefa apresentou problema. Caso positivo, ele exibe o problema e retorna a tarefa para o backlog da Sprint. Em seguida, o Scrum master aloca as tarefas para os desenvolvedores ociosos. Por fim, o sistema verifica se é o último dia da Sprint, retornando ao início deste diagrama se não for, e seguindo para o final, caso contrário.

Figura 6 – Diagrama de Atividades UML: fluxo do jogo ScrumAge



Fonte: Autor

Figura 7 – Diagrama de Atividades UML: Realizar Planejamento de Sprint



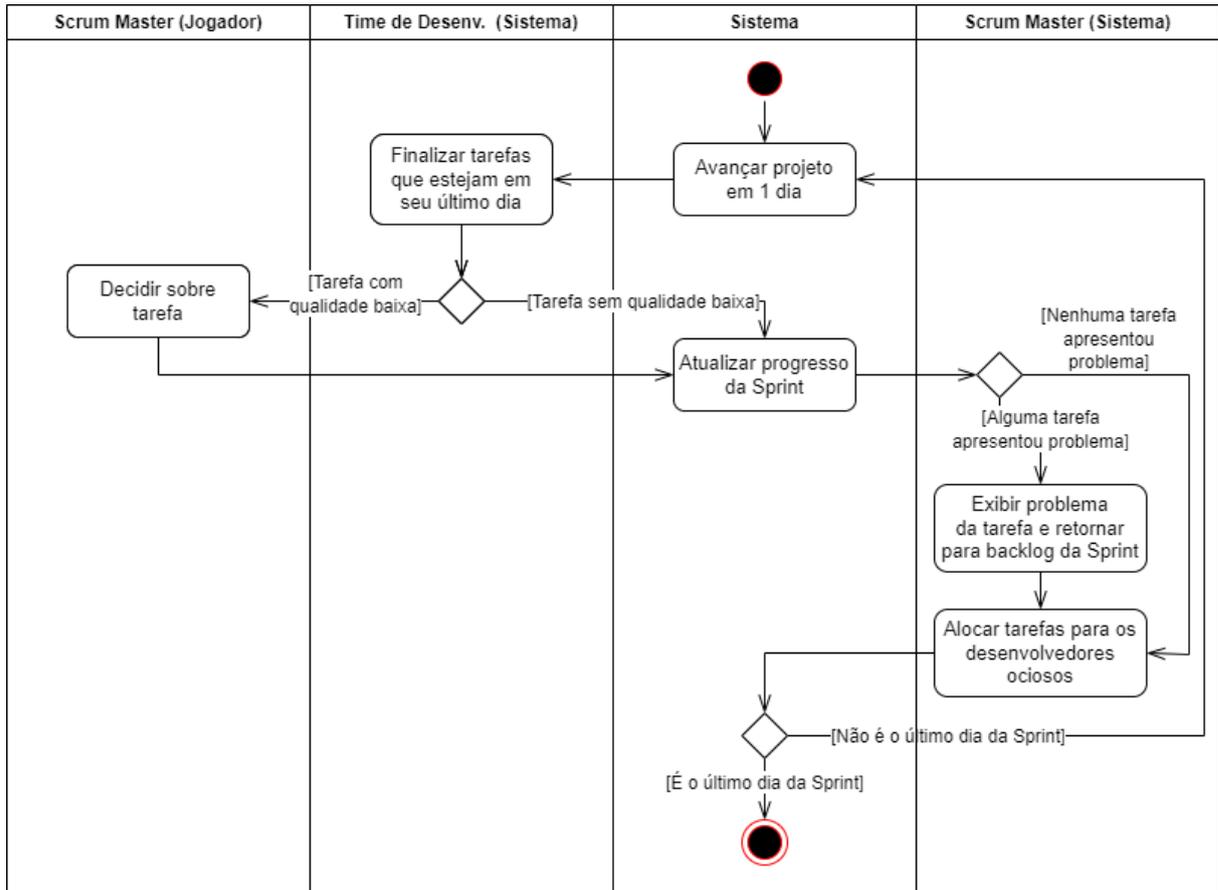
Fonte: Autor

O último diagrama de atividades é apresentado na figura 9, representando as reuniões de revisão e de retrospectiva da Sprint. O fluxo da reunião de revisão de Sprint é iniciado pelo cliente, que primeiramente solicita melhorias em tarefas, depois calcula e exibe seu nível de satisfação com a Sprint. Em seguida, o sistema atua como product owner para exibir a qualidade funcional média na Sprint. Ao fim dessa reunião, o jogador, agindo como product owner, analisa as informações exibidas.

Já a reunião de Revisão de Sprint começa com o sistema, no papel de Scrum master, atualizando o progresso e o orçamento do projeto, para em seguida exibir as tarefas que apresentaram problema na Sprint. No próximo passo, o time de desenvolvimento, controlado pelo sistema, exibe a qualidade técnica média na Sprint e a produtividade que terão na próxima Sprint. Por fim, o jogador, como Scrum master, analisa esses dados.

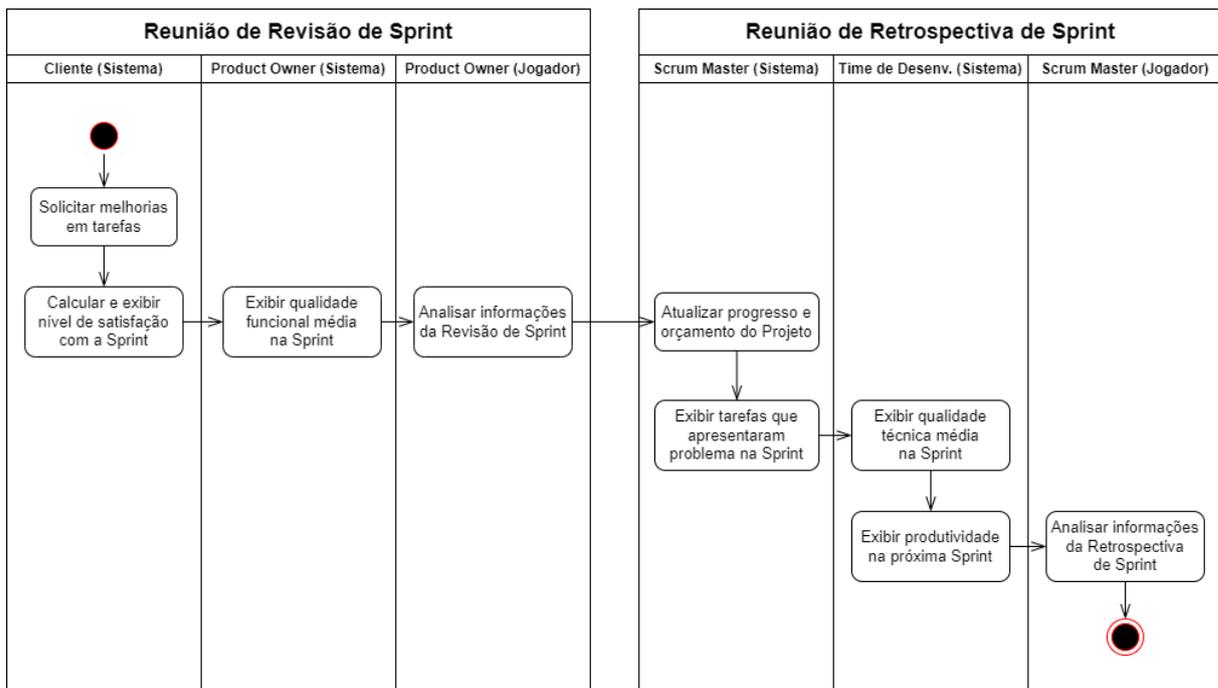
Finalizado o ciclo de desenvolvimento através das Sprints, o cliente exibe sua satisfação com o projeto, utilizada a seguir no cálculo efetuado pelo sistema para definir a pontuação do jogador. Por último, o jogador visualiza um *feedback* sobre seu desempenho, encerrando a partida.

Figura 8 – Diagrama de Atividades UML: Realizar Reuniões Diárias



Fonte: Autor

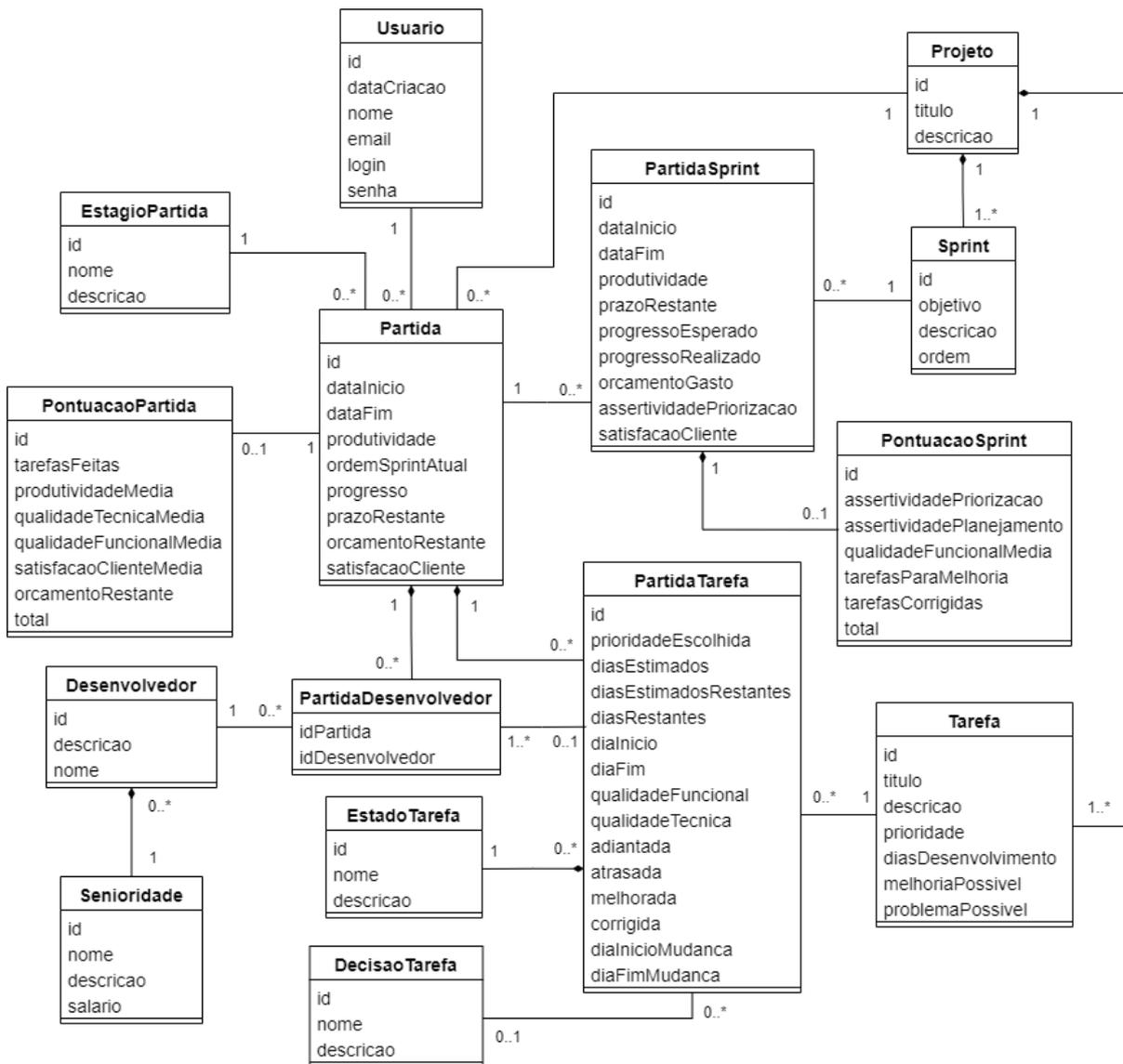
Figura 9 – Diagrama de Atividades UML: Realizar Reuniões de final de Sprint



Fonte: Autor

Como última etapa desta fase, foi realizada a criação do modelo conceitual de classes UML (figura 10). As principais classes definidas foram: usuário e partida, que compõem a parte central do jogo; projeto, sprint e tarefa, que abordam os dados dos projetos que podem ser simulados; e desenvolvedor, que possui as informações sobre os desenvolvedores que podem ser escolhidos pelo jogador.

Figura 10 – Modelo Conceitual UML: ScrumAge



Fonte: Autor

3.3. Visão Geral

O objetivo do ScrumAge é que o jogador finalize o projeto de desenvolvimento de um software de forma que o cliente fique o mais satisfeito possível, o que resultará em uma alta pontuação. Para isso, o jogador precisará tomar boas decisões gerenciais, considerando o tempo disponível e as tarefas que ainda precisam ser finalizadas.

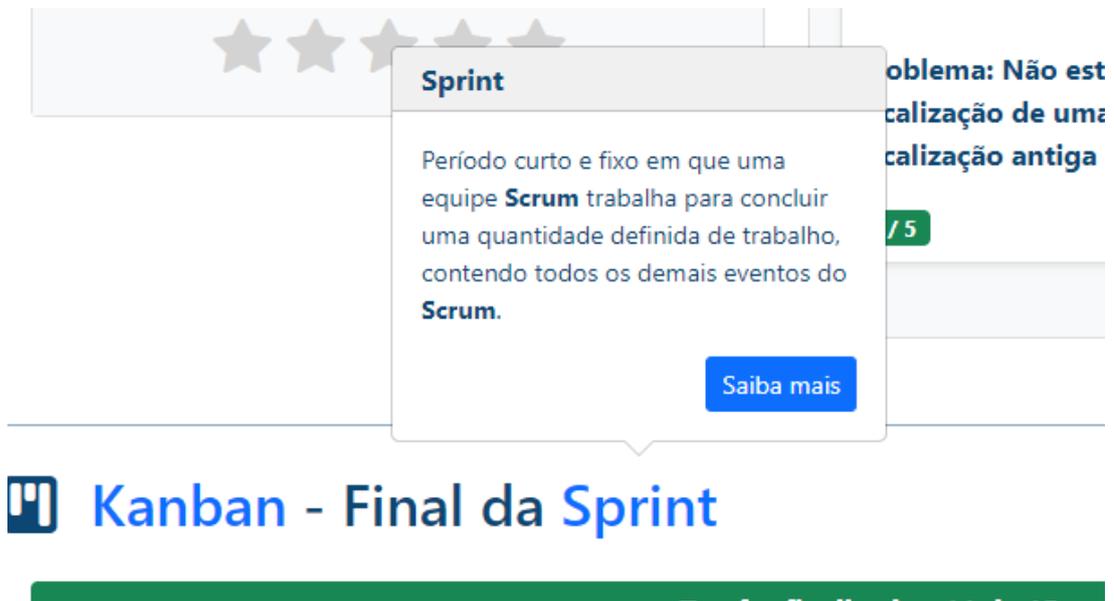
Para atender aos requisitos funcionais **RF2** (retornar ao ponto em que parou) e **RF6** (classificação), o sistema deve exigir que o jogador realize um cadastro e se autentique para jogar uma partida. Dessa forma, o estado da partida atual do jogador ficará salvo no servidor e poderá ser recuperado a qualquer momento e de qualquer dispositivo cliente, enquanto é garantido que dois usuários não aparecerão na classificação utilizando o mesmo nome de usuário.

Atendendo ao requisito funcional **RF4**, a página inicial Wiki pode ser acessada a qualquer momento a partir do menu superior. Além disso, em qualquer momento ao longo do jogo, os termos mencionados e disponíveis na Wiki serão destacados, onde o jogador poderá dar um clique e visualizar um resumo da definição daquele termo (figura 11). Caso ele clique em “Saiba mais”, o artigo completo da Wiki será exibido, contendo informações mais completas sobre aquele conceito, disponibilizando uma fonte ao final (figura 12). Dentro de um artigo, os termos abordados em outros artigos também estarão destacados e acessíveis, da mesma forma que foi descrita para aqueles exibidos durante o jogo.

A Wiki possui três seções principais, que possuem subseções cada: (i) Scrum, cujo conteúdo será traduzido do guia do Scrum (SCHWABER e SUTHERLAND, 2020) e inserido na íntegra; (ii) ScrumAge, que abordará os principais termos específicos utilizados no jogo; (iii) Conceitos Complementares, que trarão outras informações consideradas relevantes, incluindo o conteúdo completo do Manifesto Ágil (BECK, 2001), um artigo sobre o Kanban e outro sobre o planning poker (MAHNIČ, 2012). Para que o usuário consiga encontrar rapidamente informações sobre qualquer dos conteúdos disponíveis, uma caixa de pesquisa estará presente na Wiki (figura 12).

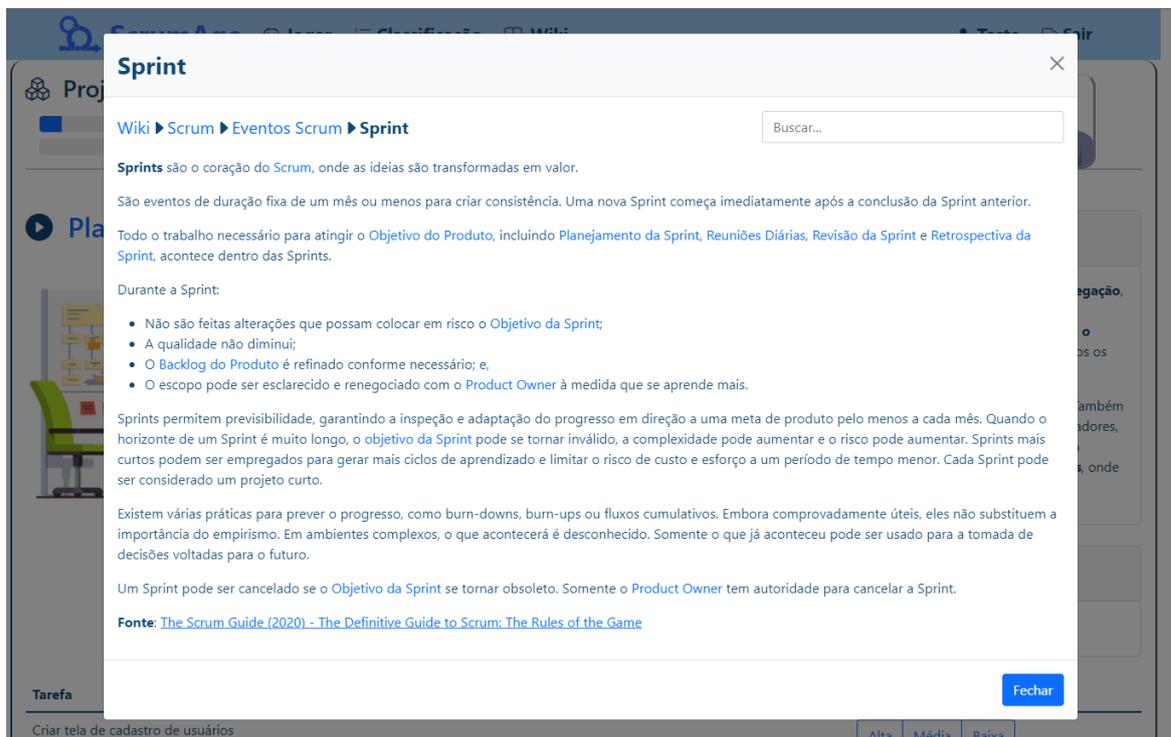
O requisito funcional **RF5** é atendido pela exibição do papel atual do jogador na partida, como uma imagem fixa no canto superior direito da tela (figura 13). Dessa forma, a atenção do jogador é voltada para qual ponto de vista ele deve seguir em cada etapa do jogo, facilitando seu entendimento sobre cada papel do Time Scrum.

Figura 11 – Resumo de conceito da Wiki no protótipo do ScrumAge



Fonte: Autor

Figura 12 – Artigo completo da Wiki no protótipo do ScrumAge



Fonte: Autor

Conforme pode ser visto na figura 13, além do papel atual, durante toda a partida serão exibidas duas barras de progresso no topo da página: uma indicando o dia atual do projeto e outra indicando a quantidade de tarefas finalizadas. Com isso, o jogador conseguirá visualizar como está seu progresso em relação às atividades necessárias para a finalização do projeto,

podendo tomar melhores decisões ao longo do jogo. Ao lado das barras de progresso, também se encontra o indicador da Sprint atual em que o jogo está.

Figura 13 – Papel atual no protótipo do ScrumAge



Fonte: Autor

Já a classificação de partidas (figura 14), exigida pelo requisito funcional **RF6**, é acessível a qualquer momento por meio do menu superior do sistema. Nela, o jogador poderá verificar os dados de todas as partidas que já foram finalizadas no ScrumAge, trazidas em ordem decrescente de acordo com a pontuação final, juntamente com o nome de usuário do jogador e a data de finalização da partida. Também, é possível filtrar a lista digitando um trecho do nome do jogador buscado. Acessando os detalhes de uma partida, o usuário conseguirá visualizar a composição da pontuação final, juntamente de todas as métricas que levaram a esta pontuação, assim como o *feedback* do cliente e do Time Scrum em relação ao projeto simulado. Esta tela deve ser a mesma exibida ao final da partida (figura 30).

Figura 14 – Tela de classificação no protótipo do ScrumAge

Posição	Pontuação	Jogador	Data
1º	9.800	Brunacons	28/07/2022
2º	9.400	Fsra1	29/07/2022
3º	8.600	Fsra1	29/07/2022
4º	7.700	verawerneck2	29/07/2022
5º	7.300	leolm	28/07/2022
6º	5.600	verawerneck	29/07/2022

Total: 6

Fonte: Autor

Visando tornar o jogo mais dinâmico e realista, cada tarefa cadastrada no jogo possui um possível problema que pode ocorrer e uma possível mudança necessária, que o cliente

pode solicitar, ações que são explicadas na seção sobre o cenário de uso. Além disso, alguns conceitos foram criados para o ScrumAge, inspirados em métricas reais (MIGUEL, 2014):

- **Qualidade Funcional:** indica a qualidade de uma tarefa finalizada, na opinião do product owner, tentando expressar o ponto de vista do cliente. Esse conceito pode ser entendido como um resumo de métricas reais, como: eficiência, facilidade de aprendizagem, aparência, entre outras. Essa qualidade influencia na probabilidade de o cliente solicitar alguma melhoria e na satisfação do cliente com o sistema. Seus valores possíveis são baixa (1), média (2) ou alta (3);
- **Qualidade Técnica:** indica a qualidade de uma tarefa finalizada, na opinião do time de desenvolvimento. Esse conceito pode ser entendido como um resumo de métricas reais, como: desempenho do software (escalabilidade, disponibilidade, velocidade de resposta), qualidade do código (reuso, manutenibilidade, organização), cobertura de testes automatizados, entre outras. Essa qualidade influencia na probabilidade de ser necessária a correção de algum problema em uma tarefa já finalizada. Além disso, também implica diretamente na produtividade do time de desenvolvimento, pois um software com melhor qualidade técnica possui maior facilidade de manutenção e de evolução. Assim como a qualidade funcional, seus valores possíveis também são baixa (1), média (2) ou alta (3);
- **Satisfação do Cliente:** indica o quanto o cliente está satisfeito com o resultado do projeto. Assume valores entre 0 e 5. Na revisão de cada Sprint ela é calculada considerando: (i) a assertividade da priorização realizada para as tarefas, em relação ao solicitado pelo cliente; (ii) a assertividade do planejamento realizado, de acordo com a quantidade de tarefas do backlog da Sprint que foram finalizadas; (iii) a quantidade de problemas ocorridos durante a Sprint; (iv) a quantidade de melhorias solicitadas pelo cliente; e (v) a qualidade funcional média das tarefas finalizadas na Sprint. Já no final do projeto ela é calculada levando em conta: (i) a quantidade de tarefas finalizadas; (ii) a qualidade funcional média de todas as tarefas finalizadas; (iii) a média das satisfações do cliente em todas as sprints; e (iv) o orçamento restante;
- **Produtividade:** indica qual a chance de uma tarefa ser finalizada de forma adiantada ou atrasada. Quanto maior a qualidade técnica das tarefas finalizadas na Sprint anterior, maior é a produtividade do time de desenvolvimento na próxima Sprint. Seus valores possíveis variam entre muito baixa (1), baixa (2), média (3), alta (4) e muito alta (5).

Todo o jogo foi planejado para poder ser facilmente traduzido para outros idiomas, necessitando apenas da utilização de arquivos de configuração contendo os textos nos idiomas desejados. Além disso, o sistema permite que diversas parametrizações sejam facilmente feitas diretamente no código-fonte do sistema, ou no banco de dados.

Dentre as parametrizações possíveis, estão aquelas relacionadas ao projeto, como o orçamento, a quantidade de desenvolvedores, a quantidade de tarefas, a quantidade de Sprints e de dias em cada Sprint etc. Outras parametrizações incluem os valores utilizados para o cálculo das métricas, como as probabilidades e as influências de cada métrica.

A partir da visão apresentada nesta proposta, o protótipo do ScrumAge foi desenvolvido, o que é relatado no próximo capítulo.

4. PROTÓTIPO DO SCRUMAGE

Uma vez finalizada a elaboração da proposta do ScrumAge, realizou-se a criação de um protótipo do jogo. A primeira seção deste capítulo mostra as etapas do processo de desenvolvimento. Já a segunda seção aborda os aspectos técnicos do protótipo. Por fim, a última seção descreve o cenário de uso do jogo, explicando seu fluxo.

4.1. Etapas do Desenvolvimento

O desenvolvimento do jogo foi dividido em três principais etapas: (i) o desenvolvimento do sistema de acordo com o definido no planejamento e na modelagem; (ii) a escrita do conteúdo textual, incluindo os textos de introdução ao jogo, os textos sobre o projeto a ser simulado e o conteúdo da Wiki; e (iii) os testes manuais e balanceamento da dificuldade.

4.1.1. Desenvolvimento do sistema

Após a finalização da modelagem, buscou-se determinar qual arquitetura seria a mais adequada para a implementação do jogo. Em seguida, foi iniciado o processo de descoberta e escolha das tecnologias específicas que poderiam ser utilizadas para implementar essa arquitetura, como o sistema gerenciador de banco de dados, as linguagens de programação e os *frameworks*. Para cada escolha considerou-se a necessidade de que a tecnologia em questão fosse gratuita, atual e bem conhecida, com grande disponibilidade de material de apoio, componentes e bibliotecas prontas na internet, visando facilitar o desenvolvimento e eventuais atualizações do sistema.

Uma vez escolhidas as tecnologias que seriam utilizadas na implementação do jogo, que serão descritas na seção sobre os aspectos técnicos do trabalho, o processo de desenvolvimento foi iniciado, sendo realizado em quatro iterações. Na primeira iteração, o módulo de gestão de usuários foi criado, incluindo a estrutura de autenticação. Essa parte foi priorizada porque todos os demais módulos exigem que o cliente esteja autenticado para acessá-los, evitando que o fator de autenticação precisasse ser adicionado posteriormente a módulos que já haviam sido finalizados.

Em seguida, na segunda iteração, os módulos de projetos e desenvolvedores foram criados, por serem necessários para a criação do módulo de jogo. Ambos foram criados com

uma estrutura que permite facilmente o cadastro de novos registros, bastando realizar inserções no banco de dados. Embora estes dois módulos sejam pequenos, optou-se pela sua separação dos demais porque entende-se que ambos possuem grande potencial de evolução em trabalhos futuros, o que será abordado nas considerações finais deste trabalho.

Na terceira iteração o módulo de jogo foi criado. É nele que todo o fluxo do jogo está contido, incluindo todos os eventos do Scrum, as regras do jogo e o cálculo das métricas e da pontuação da partida. Ao término desta iteração, a estrutura principal do jogo estava finalizada, atendendo aos requisitos funcionais **RF1** (simulador do Scrum), **RF2** (retornar de onde parou) e **RF3** (quadro Kanban).

Cada uma das três iterações descritas acima seguiu o mesmo processo de desenvolvimento, na seguinte ordem: (i) criação das estruturas do banco de dados, a partir do modelo entidade-relacionamento; (ii) programação da aplicação do servidor; (iii) programação da aplicação cliente; (iv) criação da camada de apresentação da aplicação cliente; (v) testes manuais e correções de eventuais problemas.

A quarta iteração foi focada nas funcionalidades restantes do sistema, incluindo os requisitos funcionais **RF4** (Wiki), **RF5** (revezar entre os papéis do Scrum) e **RF6** (classificação). Embora tenham sido deixadas para o fim, por não comprometerem o funcionamento do fluxo básico do jogo, todas representam diferenciais do ScrumAge em relação aos trabalhos correlatos encontrados. Dessa forma, sua existência é essencial para justificar este trabalho.

Esta última iteração foi iniciada com o desenvolvimento da Wiki, que representou um desafio de usabilidade para ser disponibilizada no sistema de forma onipresente e não invasiva, não impactando o andamento do jogo e ao mesmo tempo estando facilmente acessível. Nesta primeira versão, seu conteúdo é estático e restrito ao considerado essencial para a compreensão dos conceitos que o jogo visa ensinar, onde em trabalhos futuros espera-se permitir a edição e a criação dinâmica de seus artigos por usuários administradores.

Em seguida, realizou-se o desenvolvimento da exibição do revezamento entre os papéis do Scrum. Tanto esta funcionalidade quanto a Wiki foram realizadas exclusivamente na aplicação cliente, uma vez que todas as informações necessárias para seu funcionamento já estavam presentes na própria aplicação, sem a necessidade de acessar o banco de dados ou a

aplicação no servidor. Ou seja, foram executados apenas os três últimos passos do processo de desenvolvimento das três primeiras iterações.

Por fim, foi desenvolvida a funcionalidade que exibe a classificação de todas as partidas já realizadas no jogo, além da exibição dos detalhes de uma partida finalizada. Para isso, foi necessário buscar no banco de dados as informações sobre as partidas. Dessa forma, o processo de desenvolvimento foi o mesmo das três primeiras iterações, com exceção do primeiro passo, pois o banco de dados já estava pronto para retornar os dados desejados.

Com o fim do desenvolvimento do sistema, prosseguiu-se para a etapa de escrita do conteúdo textual. Mais detalhes sobre as tecnologias utilizadas encontram-se na seção sobre os aspectos técnicos.

4.1.2. Escrita do conteúdo textual

Finalizado o desenvolvimento do sistema, a etapa seguinte foi a escrita do conteúdo textual. Nesta etapa, o principal objetivo foi atender ao requisito não funcional **RN4** (linguagem clara e atrativa), onde buscou-se escrever todo o conteúdo de forma objetiva e leve. Os textos foram divididos em três categorias, cada uma com seu respectivo grau de formalidade: (i) textos de introdução ao jogo, com uma linguagem mais informal; (ii) textos sobre o projeto a ser simulado, reproduzindo um ambiente profissional, mas não completamente formal; e (iii) conteúdo do repositório de conhecimento (Wiki), com uma linguagem formal.

Os textos de introdução ao jogo (tela inicial, figura 16) e de explicação de regras (tela de escolha do time de desenvolvimento, figura 18), por representarem o primeiro contato do usuário com o sistema, foram escritos em um tom mais informal, buscando prender a atenção do jogador para a atividade que virá adiante. Dessa forma, a ideia é mostrar que se trata de um jogo sério, que tem o potencial de ensinar um conteúdo relevante ao jogador, mas ao mesmo tempo tenta fazer isso de forma natural e divertida.

Já os textos do projeto de desenvolvimento proposto para ser simulado pelo jogo buscam representar uma interação em um ambiente profissional, seja entre o cliente e o time Scrum ou dentro do próprio time. Sendo assim, embora o ambiente possa ser descontraído, exige certo grau de formalidade, que é simulado pelo tom utilizado na escrita. Os textos dessa categoria incluem: (i) a introdução ao projeto a ser simulado (figura 17); (ii) as descrições das

Sprints e as tarefas listadas, com seus possíveis problemas ou melhorias; e (iii) o resumo de desempenho ao final do jogo. O apêndice A lista os textos das descrições das Sprints e das tarefas do projeto, enquanto o apêndice B enumera os textos relativos ao desempenho, exibidos ao final do jogo.

Por último, os artigos da Wiki possuem uma linguagem formal. Isso se deve à sua finalidade de servirem como material de consulta aos conhecimentos que se pretende passar durante o jogo. Primeiramente, foram criados os artigos sobre o Scrum, que foram integralmente traduzidos da versão original mais recente do The Scrum Guide, em inglês (SCHWABER E SUTHERLAND, 2020). Em seguida, os artigos sobre os conceitos complementares foram escritos, como por exemplo sobre o Manifesto Ágil (BECK, 2001), também trazido na íntegra, e sobre o Kanban (CORONA e PANI, 2013). A lista com os nomes dos artigos presentes na Wiki está disponível no apêndice C. Com o conteúdo textual do jogo pronto, seguiu-se para a fase de testes e balanceamento.

4.1.3. Balanceamento

A última fase do desenvolvimento consistiu no balanceamento dos parâmetros do ScrumAge. Com esta finalidade, dezenas de partidas de teste foram jogadas até o final, onde seus parâmetros foram anotados e seus resultados comparados, de forma a refinar a experiência de jogo até que ela estivesse dentro das expectativas. Os parâmetros alterados afetam quatro aspectos: duração da partida; dificuldade do jogo; semelhança com a realidade; e a pontuação final do jogador. Nesta seção serão detalhados quais balanceamentos precisaram ser feitos e com quais objetivos, enquanto a seção sobre o cenário de uso apresentará em quais pontos esses parâmetros são utilizados.

O primeiro balanceamento necessário foi relacionado com a duração da partida. Inicialmente, estavam configuradas 6 sprints de 10 dias úteis cada, o equivalente à simulação de um projeto com 3 meses de duração. Entretanto, com tantos dias de projeto, verificou-se que o jogo estava repetitivo, sem que o jogador se deparasse com nenhuma nova situação que o levasse a um novo aprendizado ou desafio nas últimas sprints. Além disso, cada partida estava demorando em média 45 minutos para ser finalizada, enquanto o objetivo era que sua duração fosse de aproximadamente 30 minutos, conforme descrito na seção sobre a modelagem do jogo.

Sendo assim, optou-se por reduzir a duração do projeto simulado para 4 sprints. Com esta duração, observou-se que o jogador já teria contato suficiente com todos os conceitos apresentados pelo jogo. Acompanhando esta redução, os demais parâmetros relacionados também foram proporcionalmente reduzidos: de 60 tarefas a serem finalizadas para 40 e de um orçamento inicial de R\$ 60.000,00 para R\$ 40.000,00.

Com a duração da partida ajustada conforme os objetivos do trabalho, seguiu-se para o balanceamento da dificuldade do jogo. Visando tornar o projeto desafiador o suficiente para prender a atenção do jogador até o fim, ao mesmo tempo que não fosse difícil demais para que o jogador desistisse antes de finalizar a partida, diversas alterações foram realizadas.

A primeira alteração relacionada à dificuldade foi a quantidade total de dias necessários para o desenvolvimento de todas as tarefas previstas para o projeto. Como 4 desenvolvedores devem ser selecionados, trabalhando durante os 40 dias do projeto, idealmente seria possível que realizassem até 160 dias de desenvolvimento. Entretanto, buscando simular a realidade, imprevistos ocorrem durante a partida, fazendo com que diversas tarefas levem mais tempo do que o inicialmente esperado.

Dessa forma, a soma total de dias necessários para a finalização de todas as tarefas iniciou-se com 140 dias, sendo reduzida para 113 dias ao final do balanceamento deste parâmetro. Com este novo valor, notou-se que boas decisões provavelmente levarão o jogador a conseguir finalizar todas as tarefas nos últimos dias do projeto, enquanto más decisões farão com que muitas tarefas não consigam ser finalizadas dentro do tempo disponível para o projeto.

Sequenciando o balanceamento da dificuldade e da semelhança com a realidade, outros parâmetros mais simples foram ajustados, onde seus valores finais são descritos na seção sobre o cenário de uso. São eles: a probabilidade de ser solicitada uma melhoria para uma tarefa na reunião de Revisão da Sprint; o cálculo da produtividade do time de desenvolvimento na reunião de retrospectiva da Sprint; e a probabilidade de uma tarefa já finalizada apresentar um problema em uma reunião diária e precisar ser corrigida.

Como última etapa do balanceamento da dificuldade e do realismo do ScrumAge, foram realizados diversos ajustes no cálculo das qualidades de uma tarefa, baseado na senioridade do desenvolvedor que a realizou. Optou-se pelo formato de um vetor ponderado de três posições, onde cada posição indica a probabilidade de a qualidade técnica ou funcional

da tarefa ser baixa, média ou alta, para um desenvolvedor com aquela senioridade. Diversos ajustes foram feitos nesses vetores, com o objetivo de justificar a diferença salarial entre as senioridades e mais uma vez aumentar o realismo do jogo. Os valores finais para essas probabilidades são exibidos na seção sobre o cenário de uso.

Por fim, a última etapa do balanceamento em geral lidou com o cálculo da pontuação do jogo. Como até então só haviam sido definidas quais métricas deveriam ser levadas em consideração na pontuação, mas não quais seriam os cálculos feitos e seus respectivos pesos, esta foi a fase mais demorada do balanceamento. Diversos ajustes foram feitos até que os resultados fossem coerentes com o esperado no gerenciamento de um projeto: melhores decisões levam a maiores pontuações ao final da partida, enquanto decisões piores levam a uma baixa pontuação.

Uma vez finalizado o balanceamento, o desenvolvimento do ScrumAge foi dado como concluído. Em seguida, foi realizada a avaliação com profissionais da área de tecnologia da informação. A metodologia e os resultados dessa avaliação estão descritos no próximo capítulo deste trabalho.

4.2. Aspectos Técnicos

O jogo ScrumAge foi criado utilizando diversas boas práticas de programação, facilitando seu desenvolvimento e futura evolução. Entre essas práticas, destacam-se: modularização e separação em camadas com responsabilidades próprias, conforme descrito por Richards (2015), assim como nomes bem definidos e a escrita de funções pequenas e objetivas, de acordo com o recomendado pelo livro *Clean Code* (MARTIN, 2009), e a utilização de tecnologias recentes, porém bem disseminadas e reconhecidas.

Visando atender aos requisitos não funcionais **RN1** (website) e **RN2** (dispositivos móveis e computadores), além do requisito funcional **RF6** (classificação), um website foi criado na *World-Wide Web* (BERNERS-LEE, 1994a) utilizando uma arquitetura cliente-servidor (OLUWATOSIN, 2014). Dessa forma, uma única aplicação no servidor armazena e processa todas as informações relacionadas aos jogadores e às partidas, permitindo que o jogo e a classificação de partidas estejam disponíveis para qualquer aplicação cliente (website no navegador) em qualquer dispositivo com acesso à internet. A figura 15 mostra a visualização de uma reunião diária ao acessar o jogo através de um smartphone.

Já para realizar a autenticação no sistema, atendendo ao requisito funcional **RF2** (retornar ao ponto em que parou) e também ao **RF6** (classificação), o usuário é identificado por meio de um JWT (JSON Web Token). JWT é um padrão aberto e amplamente utilizado para transmitir informações com segurança e praticidade, pois as informações são trafegadas de forma criptografada (JWT, 2022).

A comunicação entre cliente e servidor é realizada através de uma API (*Application Programming Interface*) REST (*Representational State Transfer*). Uma API é um conjunto de ferramentas de programação, que permitem a troca de informações entre diferentes aplicações (OXFORD, 2022). REST é um estilo arquitetural a ser utilizado na *World-Wide Web* (BERNERS-LEE, 1994a) que tem foco nos componentes, conectores e dados de um sistema (FIELDING, 2000).

Figura 15 – Acesso ao ScrumAge usando um *smartphone*



Fonte: Autor

O sistema gerenciador de banco de dados utilizado para o sistema foi o MySQL (2022). Ele foi escolhido por atualmente ser gratuito, de código aberto, de fácil aprendizado e utilização, além de ser o segundo mais utilizado no mundo (STATISTA, 2022).

O desenvolvimento do servidor foi realizado com a plataforma Node.js (2022), que possibilitou que todo o sistema fosse escrito na mesma linguagem de programação: TypeScript (2022). Esta linguagem pertence à Microsoft (2022), fornecendo todas as funcionalidades da versão mais recente do JavaScript (ou ECMAScript) (ECMA INTERNATIONAL, 2022) e adicionando recursos, como um melhor suporte à orientação a objetos e à tipagem de dados. Utilizou-se também o *framework* NestJS (2022), feito para construir aplicações *server-side* eficientes e escaláveis.

Já para a implementação do código da aplicação cliente, a linguagem TypeScript foi utilizada com o *framework* Angular (2022). Ele é utilizado para a criação de SPAs (*Single Page Applications*), que são aplicações que se comportam como aplicativos de smartphones, tendo apenas parte do conteúdo recarregado a cada navegação, sem transições completas de telas (JADHAV et. al, 2016).

Por fim, ainda no cliente, utilizou-se o *framework* Bootstrap (2022) para a padronização de estilo e para permitir que o *website* fosse responsivo. Responsividade é a característica das aplicações que se adaptarem a diferentes resoluções e tamanhos de telas, permitindo que a aplicação mantenha uma boa usabilidade em diversos dispositivos, sejam computadores, tablets ou smartphones.

4.3. Cenário de Uso

Ao acessar o ScrumAge pela primeira vez, o usuário irá se deparar com a tela inicial (figura 16), já possuindo acesso completo à base de conhecimento do sistema (Wiki) e à classificação dos jogadores. Em um primeiro momento, o jogador deve criar seu cadastro, onde receberá um e-mail de boas-vindas. Após se cadastrar e se autenticar no sistema, o usuário terá disponível a explicação do funcionamento do jogo e poderá iniciar a partida a qualquer momento. Note que o menu superior do sistema continua visível durante todo o jogo.

Ao decidir iniciar o jogo, o usuário visualiza um texto sobre o projeto cujo desenvolvimento será simulado através da partida (figura 17). Esta primeira versão do

ScrumAge conta com apenas um projeto disponível, porém novos projetos podem ser facilmente inseridos diretamente no banco de dados, para que o jogador possa escolher entre os projetos cadastrados. Este projeto já cadastrado se trata de um sistema de gestão de estoque e de vendas, composto de 7 módulos principais, conforme pode ser visto na figura 17.

Figura 16 – Tela Inicial – Introdução ao ScrumAge

ScrumAge
O jogo para aprender Scrum de forma Ágil

Seja bem-vindo(a) ao ScrumAge!

Estamos iniciando neste momento uma jornada que irá **simular o desenvolvimento de um projeto de software utilizando a metodologia ágil Scrum**, uma das mais requisitadas no mercado de desenvolvimento. E mesmo que você não tenha interesse no desenvolvimento de softwares, saiba que o Scrum já mostrou **sucesso na gestão dos mais diversos tipos de projetos**, sendo aplicável a todos os segmentos do mercado. Então não perca tempo e aproveite esta oportunidade para aprender o máximo que puder!

Objetivo

O ScrumAge foi concebido para **facilitar o aprendizado do Scrum** através de um exemplo prático, realista (na medida do possível) e divertido, introduzindo também alguns **conceitos complementares** considerados importantes pelos autores.

Wiki

Como você já pôde notar, possuímos uma **Wiki integrada ao jogo**, de forma que um resumo de cada definição relevante para seu aprendizado sobre o Scrum ou sobre o ScrumAge está a um clique de distância! E basta apenas outro clique para acessar uma descrição mais completa de qualquer um desses tópicos. **Experimente agora!** Use e abuse dessa funcionalidade!

Instruções

Papéis

Ao longo do jogo você irá revezar entre todos os papéis do Time Scrum: Scrum Master, Product Owner e Time de Desenvolvimento. Dessa forma, para ter uma boa pontuação **você deve tomar suas decisões de acordo com o esperado para o seu papel atual**, seguindo a teoria do Scrum.

Qualidades e Produtividade

O ScrumAge utiliza três métricas importantes (**que não fazem parte do Scrum**) para tentar aumentar o realismo e a dinamicidade do jogo: **Qualidade Funcional**, **Qualidade Técnica** e **Produtividade**. Embora essas métricas possam ser chamadas de diversos nomes e abordadas das mais variadas formas no mundo real, o jogo usa uma abordagem própria bastante simples e prática, onde qualquer semelhança com a realidade é mera coincidência. **Essa abordagem não deve ser levada em consideração fora do escopo do jogo, pois foi inventada apenas para ele.**

Pontuação

A pontuação no jogo é calculada levando em consideração diversos fatores, mas o principal é a **Satisfação do Cliente**. Ela definirá a sua posição na **classificação do jogo**, que pode ser acessada através do menu superior.

Antes de começar...

Caso não consiga jogar até o final agora, saiba que o jogo sempre voltará ao mesmo ponto de onde você parou.

Esperamos que adquira o máximo de conhecimento possível e que se divirta no caminho!

Vamos jogar!

Fonte: Autor

Figura 17 – Apresentação do Projeto

ScrumAge **Jogar** **Classificação** **Wiki** **UserTest** **Sair**

Apresentação do Projeto

SISTEMA DE GESTÃO DE ESTOQUE E DE VENDAS

Olá, como vai?

Eu me chamo **Alice**, e este do meu lado é o **Lucas**. Somos os donos da VariScrum, uma pequena rede de lojas de variedades (atualmente com 3 unidades), e estamos com grandes ambições de crescimento. Para conseguirmos chegar até onde almejamos, chegamos à conclusão de que precisamos de um novo **sistema de gerenciamento de estoque e de vendas**, feito de maneira personalizada para nós, pois o atual é muito simples e não permite customizações. Sendo assim, **contamos com o trabalho da sua equipe no desenvolvimento desse software como parte importante do nosso projeto!**

Como a Alice já disse, me chamo Lucas, e agora vou te passar as informações que sei que você está mais ansioso para saber: o que imaginamos para esse sistema. Antes de mais nada, já digo que esperamos sua opinião sincera sobre nossas ideias, pois nosso objetivo é construir esse software com a maior qualidade possível, então sinta-se à vontade para se expressar caso não concorde com algo que vamos falar.

Vamos começar com o básico: queremos um **sistema web**, acessível por navegadores, para que nossos funcionários não percam tempo tendo que instalar o sistema. Além disso, nós gostaríamos de ter acesso a partir dos nossos smartphones, logo **o sistema precisa se adaptar a todos os tamanhos de tela**. As telas precisam ter um tempo adequado de carregamento, algo que nosso sistema atual não tem.

Em termos de funcionalidades, pensamos inicialmente nos seguintes módulos para o sistema:

- Gestão de Usuários
- Gestão de Produtos
- Gestão de Fornecedores
- Gestão de Lojas
- Realização de Vendas
- Dashboards estatísticos
- Gestão de Clientes

Alice falando: E aqui chegamos ao maior desafio: nosso orçamento é de **R\$ 40.000,00** para a contratação do time de desenvolvimento, e temos apenas **2 meses** para finalizarmos esse projeto. Isso se deve a estarmos com tudo certo para a abertura de duas novas lojas nessa data, e elas precisam ser inauguradas já com o novo sistema, para que não precisemos pagar pela implantação do antigo. **Se nosso projeto atrasar, a inauguração terá que ser adiada até que ele esteja pronto**. Mas já sabemos que isso não irá acontecer, porque ouvimos falar que a competência da sua gestão é lendária, e esse foi um dos motivos de termos procurado a sua empresa.

Então é isso. Ao longo do desenvolvimento iremos detalhando melhor cada uma das funcionalidades esperadas. Mas agora precisamos saber: você topa esse desafio?

Me tire daqui! **Eu aceito esse desafio!**

Fonte: Autor

Após aceitar o projeto proposto, é solicitada a primeira ação do jogador na partida: a escolha de um time de desenvolvimento adequado. Conforme pode ser visto na figura 18, um texto introdutório é apresentado, dando algumas dicas para essa escolha. Em seguida, são apresentados os conceitos de qualidade funcional e qualidade técnica, explicando que a senioridade dos desenvolvedores escolhidos influencia nestas métricas.

No mercado de trabalho, a senioridade é o nível de experiência de um trabalhador em certa profissão ou cargo. Não há uma definição formal sobre o que faz com que um profissional tenha uma ou outra senioridade, e isso pode depender de diversos fatores e variar bastante de acordo com cada área. Já no ScrumAge, existem 4 senioridades: estagiário, júnior, pleno e sênior. Nesta etapa, o jogador deve escolher 4 desenvolvedores dentre os 8 disponíveis.

Quanto maior a senioridade do desenvolvedor, maior o seu salário, logo o jogador precisará escolher desenvolvedores cuja soma dos salários não ultrapasse o orçamento do projeto, mas sabendo que quanto mais orçamento sobrar, maior também será a satisfação do

cliente ao final do projeto. O objetivo desta etapa é mostrar para o jogador como que a escolha de um time de desenvolvimento adequado tem grande influência em um projeto.

Após a escolha do time de desenvolvimento, o projeto será de fato iniciado, com o começo da primeira Sprint. Nesta primeira versão do ScrumAge, o único projeto cadastrado possui 4 Sprints de 10 dias úteis (2 semanas) cada, sendo que 40 tarefas devem ser desenvolvidas ao todo. Cada uma dessas sprints segue o fluxo descrito nas próximas seções.

Figura 18 – Escolha do Time de Desenvolvimento

ScrumAge
Jogar
Classificação
Wiki
UserTest
Sair

Projeto: Sistema de Gestão de Estoque e de Vendas

ESCOLHA DO TIME DE DESENVOLVIMENTO

Levando em consideração o **prazo (2 meses)** e o **orçamento (R\$ 40.000,00)** definidos pelo cliente para a contratação do **Time de Desenvolvimento**, você, com sua habilidade **lendária** de gestão utilizando o **Scrum**, chegou à conclusão de que é preciso montar um time com **4 desenvolvedores**, cuja soma dos salários seja **menor ou igual à R\$ 20.000,00 mensais**. Você também sabe que o cliente ficaria mais feliz quanto mais dinheiro do **orçamento sobrar ao final do projeto**, se isso for possível. Entretanto, você diz a si mesmo que seria bom ter um orçamento restante, mas que mais importante que isso é montar um time que conseguirá **finalizar o projeto dentro do prazo acordado**.

Com toda sua experiência, você sabe que um dos pontos mais importantes para a montagem de uma **equipe de desenvolvimento** com um bom custo-benefício é o **equilíbrio entre a senioridade dos desenvolvedores**. Como 4 pessoas devem ser escolhidas, não seria bom que mais de duas tivessem o mesmo nível de experiência, pois o time sairia desse equilíbrio. Dessa forma, você pediu ao departamento de recursos humanos da sua empresa para selecionar **8 desenvolvedores** do banco de vagas, sendo 2 de cada **senioridade**, de forma que você possa **realizar a escolha mais adequada para o projeto**.

Com tanto conhecimento sobre a gestão de projetos de software, você possui **métricas próprias para medir o sucesso do seu time**. Você as chama de **Qualidade Funcional** e de **Qualidade Técnica**, que são definidas como alta, média ou baixa para cada tarefa finalizada, podendo ser resumidas da seguinte maneira:

- Qualidade Funcional:** indica a qualidade de uma tarefa finalizada, na opinião do **Product Owner**, tentando expressar o ponto de vista do **Cliente**. Essa qualidade influencia na probabilidade do **Cliente** solicitar alguma **Melhoria** e na **Satisfação do Cliente** com o sistema.
- Qualidade Técnica:** indica a qualidade de uma tarefa finalizada, na opinião do **Time de Desenvolvimento**. Essa qualidade influencia na probabilidade de ser necessária a **Correção** de algum problema em uma tarefa já finalizada. Além disso, também implica diretamente na **Produtividade** do **Time de Desenvolvimento**, pois um software com melhor **qualidade técnica** possui maior facilidade de manutenção e de evolução.

Quanto maior a **senioridade** de um desenvolvedor, maior a probabilidade dele finalizar uma tarefa com as **qualidades altas**.

Com todo esse conhecimento em mente, **quem você escolhe para o seu Time de Desenvolvimento?**

	Nome	Senioridade	Salário
<input checked="" type="checkbox"/>	Eliana	Estagiário	R\$ 1.000,00
<input type="checkbox"/>	Elias	Estagiário	R\$ 1.000,00
<input checked="" type="checkbox"/>	Júlia	Júnior	R\$ 3.000,00
<input type="checkbox"/>	João	Júnior	R\$ 3.000,00
<input checked="" type="checkbox"/>	Paula	Pleno	R\$ 5.000,00
<input type="checkbox"/>	Pedro	Pleno	R\$ 5.000,00
<input checked="" type="checkbox"/>	Suzana	Sênior	R\$ 7.000,00
<input type="checkbox"/>	Sérgio	Sênior	R\$ 7.000,00

Custos mensais: **R\$ 16.000,00** / R\$ 20.000,00

[Continuar](#)

Fonte: Autor

4.3.1. Reunião de Planejamento de Sprint

O sistema, no papel do cliente, a cada reunião de planejamento de Sprint exibe uma mensagem contendo suas solicitações para a Sprint atual, como mostra a figura 19. Esse texto sempre possui o mesmo formato: primeiramente descreve as 5 solicitações essenciais, para em seguida trazer as 5 demandas secundárias para a Sprint, mas ainda desejáveis. Nele, cada tarefa existente é destacada em negrito, de forma que o jogador possa facilmente associar as solicitações com as tarefas listadas mais abaixo na mesma tela.

Figura 19 – Planejamento da Sprint – Parte 1: Mensagem do cliente e Objetivo da Sprint



The screenshot displays the ScrumAge interface for a project titled "Projeto: Sistema de Gestão de Estoque e de Vendas". The interface includes a navigation bar with "Jogar", "Classificação", and "Wiki" options, and a user profile section with "Teste" and "Sair" buttons. The main content area shows the current sprint progress: "Dia 1 de 40" and "Feitas: 0 de 40". The "Planejamento da Sprint" section is active, featuring an illustration of a meeting and a "Mensagem do Cliente" box. The message text reads: "Antes de mais nada, gostaríamos de ter a **tela inicial** do projeto pronta, com o **menu de navegação**, para já podermos ter uma noção da aparência do sistema. Sendo tão importante quanto isso, também desejamos ter logo o **cadastro de usuários**, que deverá **enviar um e-mail para que o usuário ative a sua conta**. Além disso, é imprescindível que desde já seja necessário que todos os usuários se **autentiquem** para utilizarem o sistema. Adicionalmente, o **perfil** de cada usuário deverá ser visualizado através de uma tela própria. Também é interessante termos **permissões** distintas para cada tipo de utilizador do sistema: administradores, vendedores e caixas, o que será importante conforme os próximos módulos do sistema forem desenvolvidos. Usuários administradores deverão ter acesso a uma **tela de busca de usuários**, onde poderão também **alterá-los** ou **excluí-los**." Below the message is the "Objetivo da Sprint" section, which states: "Implementar o módulo de cadastro e acesso de usuários".

Fonte: Autor

Abaixo do texto do cliente, o sistema, agora no papel de product owner, exibe o objetivo da Sprint (figura 19), para apresentar esse conceito ao jogador e auxiliá-lo ainda mais na priorização, que ele deverá realizar a seguir.

Logo em seguida, o sistema exibe as tarefas que foram criadas de acordo com a mensagem do cliente e com o conhecimento do product owner sobre as próximas necessidades do projeto, conforme mostra a figura 20. A cada planejamento de Sprint, o jogo exibe uma lista embaralhada contendo as 13 tarefas mais prioritárias ainda não desenvolvidas, com exceção do último planejamento, que traz todas as tarefas que ainda não foram

finalizadas. Essas tarefas são as candidatas a serem selecionadas para o backlog da Sprint na última etapa da reunião de planejamento de Sprint.

Na primeira Sprint, são trazidas as 10 tarefas mencionadas na mensagem do cliente e as 3 tarefas que serão as mais prioritárias na próxima Sprint. Nas demais sprints, as tarefas já priorizadas na Sprint anterior e ainda não finalizadas, são automaticamente marcadas com prioridade alta, independentemente da sua prioridade anterior.

Após finalizar a leitura da mensagem do cliente e do objetivo da Sprint, o jogador, no papel do product owner, é instruído a realizar sua primeira ação no planejamento da Sprint: selecionar as prioridades das tarefas listadas, como pode ser visto na figura 20. Para isso, ele deve utilizar o seguinte critério: alta para as principais atividades; média para as secundárias; e baixa para as que não foram citadas na mensagem do cliente.

Figura 20 – Planejamento da Sprint – Parte 2: Priorização de tarefas

Selecione a prioridade de acordo com o solicitado pelo Cliente: **alta** para as principais atividades, **média** para as secundárias e **baixa** para as que não foram citadas neste planejamento de Sprint:

Tarefa	Prioridade 
Criar tela inicial	Alta Média Baixa
Possibilitar exclusão de usuários	Alta Média Baixa
Enviar e-mail para ativação de conta	Alta Média Baixa
Criar tela de cadastro de usuários	Alta Média Baixa
Criar tela de perfil de usuário	Alta Média Baixa
Criar tela de cadastro de produtos	Alta Média Baixa
Possibilitar alteração de usuários	Alta Média Baixa
Criar autenticação de usuários	Alta Média Baixa
Possibilitar alteração de produtos	Alta Média Baixa
Criar tela de busca de usuários	Alta Média Baixa
Criar menu de navegação	Alta Média Baixa
Implementar controle de permissões por perfil: administrador, vendedor, caixa	Alta Média Baixa
Criar tela de busca de produtos	Alta Média Baixa

[Continuar](#)

Fonte: Autor

Feita a priorização, o jogador passa a representar o time de desenvolvimento, para então estimar quantos dias cada tarefa levará para ser finalizada (figura 21). Essa estimativa visa simular o uso do Planning poker (MAHNIČ, 2012) utilizando a sequência de Fibonacci,

de forma que cada tarefa pode ser estimada em 1, 2, 3, 5 ou 8 dias de desenvolvimento. Por ser difícil acertar estimativas sem ter um conhecimento prévio mais abrangente sobre o sistema que está sendo desenvolvido, optou-se por restringir as opções de cada tarefa de forma que apenas duas fiquem disponíveis: a estimativa mais aproximada com a real e outra mais afastada.

A última etapa da reunião de planejamento de Sprint é a seleção das tarefas do backlog do produto que irão compor o backlog da Sprint, conforme mostra a figura 22. Nesta fase, o sistema traz a lista ordenada das tarefas candidatas para seleção. A ordenação é feita listando primeiramente as tarefas de maior prioridade, onde as de menor prazo estimado são exibidas antes entre tarefas de mesma prioridade. Nesta lista, todas as tarefas com prioridade alta, incluindo as que ainda não foram finalizadas na Sprint anterior, são automaticamente selecionadas para o Backlog da Sprint.

Figura 21 – Planejamento da Sprint – Parte 3: Estimativa de tarefas

Indique quantos dias você acha que o desenvolvimento da tarefa levará. Para facilitar a estimativa, apenas duas opções estão disponíveis.

Tarefa	Prioridade	Estimativa (dias)
Criar tela inicial	Alta	<input checked="" type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3
Possibilitar exclusão de usuários	Média	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input checked="" type="checkbox"/> 5 <input type="checkbox"/> 8
Enviar e-mail para ativação de conta	Alta	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input checked="" type="checkbox"/> 5 <input type="checkbox"/> 8
Criar tela de cadastro de usuários	Alta	<input checked="" type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 5 <input type="checkbox"/> 8
Criar tela de perfil de usuário	Média	<input type="checkbox"/> 1 <input checked="" type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 5 <input type="checkbox"/> 8
Criar tela de cadastro de produtos	Baixa	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input checked="" type="checkbox"/> 5 <input type="checkbox"/> 8
Possibilitar alteração de usuários	Média	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input checked="" type="checkbox"/> 5 <input type="checkbox"/> 8
Criar autenticação de usuários	Alta	<input checked="" type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 5 <input type="checkbox"/> 8
Possibilitar alteração de produtos	Baixa	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input checked="" type="checkbox"/> 5 <input type="checkbox"/> 8
Criar tela de busca de usuários	Média	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input checked="" type="checkbox"/> 3 <input type="checkbox"/> 5 <input type="checkbox"/> 8
Criar menu de navegação	Alta	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input checked="" type="checkbox"/> 5 <input type="checkbox"/> 8
Implementar controle de permissões por perfil: administrador, vendedor, caixa	Média	<input checked="" type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 5 <input type="checkbox"/> 8
Criar tela de busca de produtos	Baixa	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input checked="" type="checkbox"/> 3 <input type="checkbox"/> 5 <input type="checkbox"/> 8

[Continuar](#)

Fonte: Autor

O jogador, no papel de Scrum master e analisando as tarefas candidatas, é orientado a selecionar quais devem ser desenvolvidas ao longo da Sprint atual. Neste momento, o desafio é que sejam selecionadas as tarefas de prioridade média e baixa que o jogador acredita que

poderão ser finalizadas dentro da Sprint. Para isso, a cada seleção realizada, o total de dias estimados para o desenvolvimento de todas as tarefas e para cada desenvolvedor individualmente são atualizados, sendo esses números exibidos em vermelho quando a seleção está com poucos ou muitos dias estimados, enquanto são exibidos em verde quando possuem valores realistas.

A produtividade da equipe também é exibida neste ponto, uma vez que influencia diretamente na quantidade de carga de trabalho que o time de desenvolvimento conseguirá realizar durante a Sprint. O jogador deve entender que, quanto mais baixa estiver a produtividade, menos tarefas devem ser selecionadas, pois maior é o risco de nem todas serem finalizadas dentro da Sprint. O pensamento análogo é válido para quanto mais alta a produtividade estiver. Na primeira sprint, como não há um histórico de tarefas realizadas, a produtividade é sempre média (valor 3).

Figura 22 – Planejamento da Sprint – Parte 4: Seleção de tarefas para o Backlog da Sprint

Selecione as tarefas que irão compor o Backlog da Sprint

Tarefa	Status	Prioridade	Dias Estimado
<input checked="" type="checkbox"/> Criar tela inicial	Backlog do Produto	Alta	1
<input checked="" type="checkbox"/> Criar tela de cadastro de usuários	Backlog do Produto	Alta	1
<input checked="" type="checkbox"/> Criar autenticação de usuários	Backlog do Produto	Alta	1
<input checked="" type="checkbox"/> Enviar e-mail para ativação de conta	Backlog do Produto	Alta	5
<input checked="" type="checkbox"/> Criar menu de navegação	Backlog do Produto	Alta	5
<input checked="" type="checkbox"/> Implementar controle de permissões por perfil: administrador, vendedor, caixa	Backlog do Produto	Média	1
<input checked="" type="checkbox"/> Criar tela de perfil de usuário	Backlog do Produto	Média	2
<input checked="" type="checkbox"/> Criar tela de busca de usuários	Backlog do Produto	Média	3
<input checked="" type="checkbox"/> Possibilitar exclusão de usuários	Backlog do Produto	Média	5
<input checked="" type="checkbox"/> Possibilitar alteração de usuários	Backlog do Produto	Média	5
<input checked="" type="checkbox"/> Criar tela de busca de produtos	Backlog do Produto	Baixa	3
<input checked="" type="checkbox"/> Criar tela de cadastro de produtos	Backlog do Produto	Baixa	5
<input type="checkbox"/> Possibilitar alteração de produtos	Backlog do Produto	Baixa	5

Tarefas Selecionadas: **12**
 Total de Dias Estimados: **37**
 Dias Estimados para cada Desenvolvedor: **9**
 Produtividade: **3**

Continuar

Fonte: Autor

Em sua escolha, o jogador deve evitar selecionar tarefas demais, de forma que o cliente não fique insatisfeito com a baixa assertividade do planejamento que lhe foi apresentado. Por outro lado, se poucas tarefas forem selecionadas, mais desenvolvedores ficarão ociosos ao final da Sprint, aumentando o risco de que nem todas as tarefas do projeto

sejam finalizadas até o seu fim, o que também causa grande impacto na satisfação final do cliente.

Após a seleção das tarefas que irão compor o backlog da Sprint, o jogo realiza a alocação automática de tarefas para os desenvolvedores que não estão com nenhum desenvolvimento em andamento. Nesta versão do ScrumAge, optou-se por realizar essa alocação de forma automática visando manter a duração média das partidas dentro do tempo esperado, pois o jogador levaria muito tempo para realizar uma boa alocação.

Tanto ao final do planejamento de Sprint quanto durante as reuniões diárias, essa atribuição de tarefas sempre segue o mesmo algoritmo:

- i. Primeiramente, as tarefas com estimativa de 8 dias são alocadas, seguidas pelas de 5 dias, independentemente da prioridade. Entre tarefas de mesma estimativa, a com maior prioridade é alocada primeiro.
- ii. Em seguida, as tarefas com 1, 2 ou 3 dias de estimativa são alocadas. Para elas, o critério é o oposto: as com maior prioridade são alocadas antes, e entre tarefas de mesma prioridade, as de maior estimativa primeiro.
- iii. Seguindo a ordenação descrita, a tarefa mais prioritária é atribuída ao desenvolvedor de menor senioridade que está disponível, sem estar desenvolvendo nenhuma outra tarefa.

A versão final desse algoritmo foi obtida de maneira empírica, onde diversas outras abordagens foram testadas e essa se destacou como a que trouxe maior efetividade no desenvolvimento das tarefas ao longo do projeto. Dessa forma, cada parte do algoritmo, respectivamente, leva à seguinte tendência:

- i. Que o risco de uma tarefa não ser finalizada na Sprint atual seja minimizado: se a ordem das tarefas a serem desenvolvidas considerasse apenas a priorização realizada, tarefas mais prioritárias seriam realizadas antes independentemente de sua estimativa. Sendo assim, haveria grande risco das tarefas com maior estimativa não serem finalizadas dentro da Sprint, uma vez que seriam iniciadas apenas no meio da Sprint, onde possivelmente não haveria tempo hábil restante para finalizá-las.
- ii. Que as tarefas mais prioritárias sejam finalizadas antes, o que tem maior relevância para a satisfação do cliente: mesmo realizando as tarefas de maior estimativa antes, ainda é alta a probabilidade de que todas as tarefas mais prioritárias sejam

finalizadas dentro da Sprint. Isso ocorre porque se as tarefas de maior prioridade forem as de maior estimativa, elas já estarão feitas. Por outro lado, se as mais prioritárias forem as de menor estimativa, elas provavelmente conseguirão ser finalizadas até o final da Sprint, justamente por serem mais rápidas de serem desenvolvidas.

- iii. Que as tarefas finalizadas tenham maior chance de terminarem com as qualidades funcional e técnica mais altas: como as tarefas mais longas são alocadas antes e a atribuição a desenvolvedores de menor senioridade também ocorre antes, a maior probabilidade é que tais tarefas sejam realizadas por esses desenvolvedores. Dessa forma, enquanto os desenvolvedores mais júniores estão focados em uma única grande tarefa (cuja tendência é que seja finalizada com menor qualidade), os desenvolvedores mais sêniores estarão realizando várias das demais tarefas, por serem mais rápidas, maximizando a quantidade de tarefas finalizadas com maior qualidade.

Finalizada a alocação automática de tarefas para os desenvolvedores que estavam ociosos, a reunião de planejamento de Sprint é finalizada e o jogo segue para as reuniões diárias daquela Sprint.

4.3.2. Reuniões Diárias

Após a reunião de planejamento de Sprint, é iniciada a primeira reunião diária da Sprint. Como cada Sprint possui 10 dias úteis, 10 reuniões diárias são realizadas. De maneira similar às barras de progresso do projeto que ficam no topo da página, a tela de reunião diária (figura 23) também possui duas barras de progresso: a primeira indica o dia atual da Sprint, enquanto a segunda indica quantas tarefas já foram finalizadas na Sprint, em relação à quantidade total de tarefas selecionadas na reunião de planejamento. Durante as reuniões diárias, o jogador representa o time de desenvolvimento, alternando para o Scrum master quando uma decisão precisa ser tomada.

Abaixo das barras de progresso encontra-se o quadro Kanban (CORONA e PANI, 2013) utilizado no ScrumAge para que o jogador consiga visualizar o andamento das tarefas selecionadas para a Sprint. Conforme pode ser visto na figura 23, neste jogo o quadro possui as colunas Fazer, Fazendo, Revisando e Feito, que representam os estados possíveis para uma

tarefa presente no Backlog da Sprint. Ao lado de cada coluna encontra-se um número indicando a quantidade de tarefas atualmente naquele estado.

Cada tarefa é exibida através de um cartão que contém os elementos visuais necessários para que o jogador consiga verificar rapidamente os atributos atuais daquela tarefa. Ao passar o cursor por cima desses elementos, uma pequena descrição aparece, conforme mostra a figura 24A. Desses elementos, todos aqueles que não possuem texto podem ser clicados, abrindo o resumo correspondente ao seu conceito. A tabela 2 mostra os conceitos representados pelos ícones e seus possíveis valores.

Figura 23 – Reunião Diária



Fonte: Autor

Há três formatos de cartão: um utilizado pelas tarefas presentes na coluna Fazer, outro pelas tarefas nas colunas Fazendo e Revisando, e um último formato para as tarefas na coluna Feito. Todos eles trazem o ícone que indica a prioridade, além do código identificador e da definição da tarefa. Exemplos desses cartões podem ser vistos na figura 24, sendo explicados a seguir.

Tabela 2 – Significado dos ícones dos cartões das tarefas

Prioridade	Qualidade Funcional	Qualidade Técnica	Decisão	Alteração	Efeito da Produtividade
Baixa Média Alta	Alta Média Baixa	Alta Média Baixa	Aprofundar Testes Refatorar/ Revisar Não fazer nada	Correção Melhoria	Adiantada Atrasada

As figuras 24A e 24B mostram exemplos de cartões de tarefas da coluna Fazer, onde o rodapé dos cartões possuem um texto indicando que ainda não houve progresso diante da quantidade de dias estimados para aquela tarefa. Note que a figura 24A mostra o cartão de uma tarefa que já havia sido finalizada na Sprint anterior, mas que retornou para o backlog da Sprint atual devido à uma melhoria solicitada pelo cliente, o que será mais bem explicado na seção sobre a reunião de revisão da Sprint. Por outro lado, a figura 24B representa uma tarefa cujo desenvolvimento ainda não foi iniciado, contendo apenas o ícone que indica sua prioridade.

Figura 24 – Exemplos de cartões de tarefas

Fazer	Fazendo / Revisando	Feito
<p>A</p> <p>FAZER</p> <p>Melhorar: Exibir na tela de registro de vendas o tipo de cartão utilizado (crédito ou débito).</p> <p> TAREFA-26 </p> <p>Realizar integração com sistema de pagamento com cartão</p> <p>0 de 2</p>	<p>C</p> <p> TAREFA-18 </p> <p>Possibilitar alteração de dados do fornecedor</p> <p>0 de 1 </p>	<p>F</p> <p> TAREFA-24 </p> <p>Criar componente para visualização do histórico de ações de um usuário</p> <p>4 / 5 </p>
<p>B</p> <p> TAREFA-27</p> <p>Realizar integração com sistema de pagamento Pix por QR Code</p> <p>0 de 3</p>	<p>D</p> <p> TAREFA-25</p> <p>Criar tela de busca de lojas</p> <p>2 de 3 </p>	<p>G</p> <p> TAREFA-29 </p> <p>Criar tela de cadastro de lojas</p> <p>7 / 5 </p>
	<p>E</p> <p> TAREFA-38</p> <p>Criar dashboard com estatísticas de vendas por produtos</p> <p>2 de 1 </p>	<p>H</p> <p> TAREFA-30 </p> <p>Permitir exclusão de lojas</p> <p>2 / 5 </p>

Fonte: Autor

Já as figuras 24C, 24D e 24E representam cartões das colunas Fazendo e Revisando. O lado esquerdo do rodapé do cartão indica há quantos dias o desenvolvimento está ocorrendo em relação ao tempo originalmente estimado. Quando o tempo de desenvolvimento ultrapassa a estimativa, a barra de progresso muda da cor azul para vermelho (figura 24E). Já no lado direito do rodapé encontra-se o nome do desenvolvedor que está atuando na tarefa, em azul se for o primeiro desenvolvimento, ou em cinza se for uma correção ou melhoria em uma tarefa que já havia sido finalizada.

O último formato de cartão de tarefa é o que pode ser visto nas figuras 24F, 24G e 24H, presentes na coluna Feito. Todas as tarefas finalizadas possuem à direita do cabeçalho de seu cartão os ícones representando a qualidade funcional e a qualidade técnica com a qual aquela tarefa foi feita. Além disso, se a tarefa tiver passado por uma decisão do jogador, o que será explicado adiante, ela ficará com um ícone indicando qual foi essa decisão, conforme mostra a figura 24H. Os outros dois ícones que podem aparecer no cabeçalho de uma tarefa indicam se ela já havia sido finalizada anteriormente e passou por uma correção ou melhoria.

Já o rodapé dos cartões da coluna Feito possui, à esquerda, a relação entre a quantidade de dias que o desenvolvimento da tarefa levou e a estimativa realizada, seguida pelo ícone que indica atraso ou adiantamento devido à produtividade do time de desenvolvimento, quando ocorrer. À direita encontram-se o nome dos desenvolvedores que atuaram na tarefa, seguindo a mesma lógica dos cartões das colunas Fazendo e Revisando.

O jogador deve analisar constantemente todos os elementos visuais mencionados para tentar tomar as melhores decisões gerenciais, quando necessário. Após familiarizar-se com essas informações, o jogador pode prosseguir com o jogo, indo para a próxima reunião diária. A cada início de reunião diária, o sistema diminui o prazo do projeto em 1 dia, atualizando a barra de progresso em todas as tarefas que estão em desenvolvimento.

Cada tarefa tem um tempo fixo de desenvolvimento, armazenado internamente e omitido do jogador. A produtividade do time de desenvolvimento (tabela 3) é utilizada primeiramente quando uma tarefa chega ao seu penúltimo dia de desenvolvimento, quando o sistema utiliza as probabilidades de adiantamento descritas na tabela 3 para definir se a tarefa será terminada de forma adiantada em 1 dia. Não tendo sido adiantada, em seu último dia, o sistema verifica novamente a produtividade do time e utiliza as probabilidades de atraso da tabela 3 para decidir se a tarefa será atrasada em 1 dia. A produtividade afeta apenas as tarefas

que estão sendo finalizadas pela primeira vez, não influenciando no prazo de tarefas sendo corrigidas ou melhoradas.

Tabela 3 – Relação entre produtividade e probabilidades de adiantamento e atraso

Produtividade	Probabilidade de Adiantamento	Probabilidade de Atraso
1	0%	30%
2	0%	20%
3	15%	15%
4	20%	0%
5	30%	0%

Quando uma tarefa é finalizada, o sistema realiza o cálculo de suas qualidades funcional e técnica, de acordo com a senioridade do desenvolvedor que a realizou. A relação entre a senioridade e a probabilidade de cada uma das qualidades ser baixa, média ou alta é descrita na tabela 4.

Tabela 4 – Relação entre a senioridade e a probabilidade de cada qualidade

Senioridade	Qualidade Baixa	Qualidade Média	Qualidade Alta
Estagiário	20%	70%	10%
Júnior	15%	60%	25%
Pleno	15%	40%	45%
Sênior	10%	25%	65%

Tarefas finalizadas com pelo menos uma das qualidades como baixa possibilitam que o jogador, no papel de Scrum master, decida realizar alguma ação para que a qualidade seja melhorada, conforme mostra a figura 25. Quando isso ocorre, as seguintes opções são disponibilizadas para o jogador, de acordo com as qualidades da tarefa finalizada:

- **Revisar:** opção disponível apenas quando a qualidade funcional está baixa. Estende o prazo de desenvolvimento da tarefa em 1 dia, mas aumenta a qualidade funcional para o valor máximo;
- **Refatorar:** opção disponível apenas quando a qualidade técnica está baixa. Estende o prazo da tarefa em 1 dia, mas aumenta a qualidade técnica para o valor máximo. Refatoração é um termo utilizado por desenvolvedores de software para designar a ação de reescrever ou reestruturar parte do código de um sistema, visando realizar alguma melhoria;
- **Aprofundar testes:** opção disponível apenas quando uma das qualidades está baixa e a outra está baixa ou média. Estende o prazo da tarefa em 2 dias, mas aumenta ambas as qualidades para o valor máximo;

- **Não fazer nada:** opção que sempre está disponível, mantendo a tarefa como está e não influenciando em seu prazo.

Dependendo da senioridade média do time de desenvolvimento, é esperado que o jogador precise realizar de 2 a 3 decisões por Sprint, em média. Quando uma tarefa é finalizada, o algoritmo de alocação descrito anteriormente é novamente utilizado, de forma que nenhum desenvolvedor ficará ocioso enquanto existir alguma tarefa no backlog da Sprint com o estado Fazer.

Figura 25 – Reunião Diária – Decidir sobre tarefa

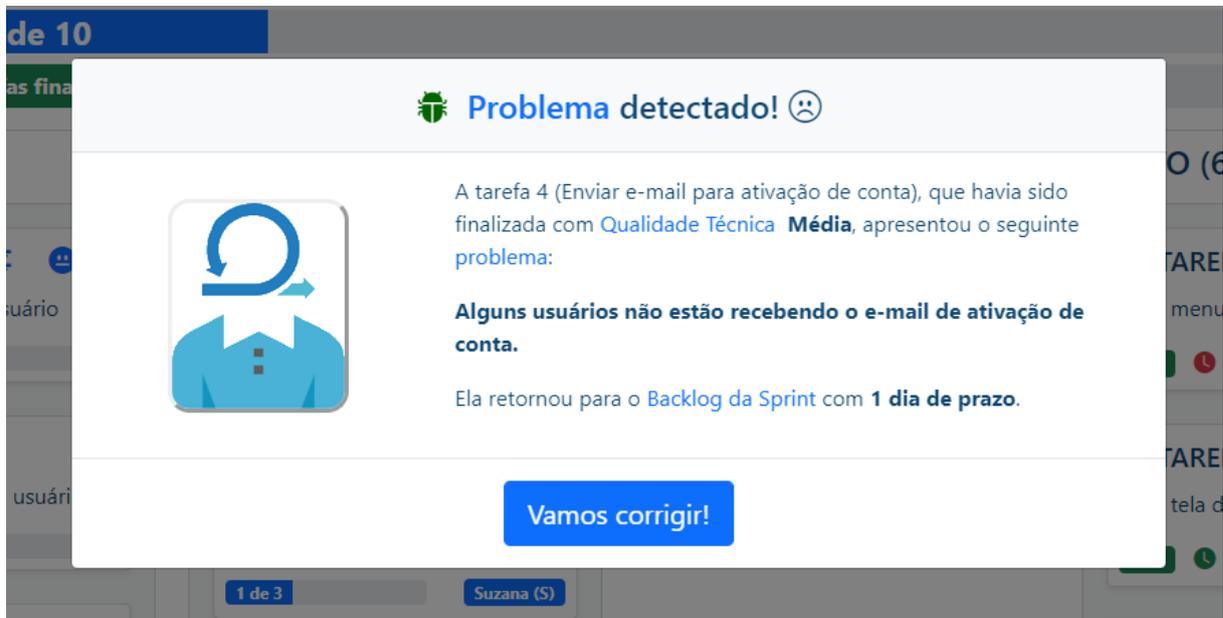


Fonte: Autor

Durante qualquer reunião diária a partir da segunda Sprint, existe uma chance de que uma tarefa da Sprint anterior que não tenha sido finalizada com qualidade técnica alta apresente algum problema, conforme mostra a figura 26. A cada reunião, cada tarefa com qualidade técnica baixa tem 40% de chance de necessitar de uma correção, enquanto cada tarefa com qualidade técnica média tem 15% de chance. Quando isso ocorre, a tarefa volta para o backlog da Sprint atual com 1 dia de prazo. De acordo com a senioridade média do time de desenvolvimento, é esperado que de 2 a 3 tarefas apresentem problema por Sprint.

Após avançar por todos os dias da Sprint, decidindo sobre todas as tarefas que ficaram com qualidade técnica baixa, a etapa de reuniões diárias é finalizada. Em seguida, o jogador irá para a reunião de revisão da Sprint.

Figura 26 – Reunião Diária – Problema detectado



Fonte: Autor

4.3.3. Reunião de Revisão da Sprint

Duas reuniões são simuladas no último dia de cada Sprint, após a última reunião diária. A primeira é a revisão da Sprint, descrita nesta seção, e a segunda é a retrospectiva da Sprint, que será abordada na próxima seção. Em ambas, a parte superior da tela exibe as informações específicas da reunião, enquanto a parte inferior exibe uma versão compacta do quadro Kanban, de forma que o jogador possa conferir o estado final de todas as tarefas do backlog da Sprint, conforme mostra a figura 27.

Na reunião de revisão da Sprint do ScrumAge, o jogador participa como product owner, organizando as informações mais relevantes para todas as partes interessadas (no ScrumAge, são o time Scrum e o cliente). As três principais informações exibidas são as melhorias solicitadas pelo cliente e as métricas de qualidade funcional média e satisfação do cliente na Sprint, conforme pode ser visto na figura 28.

Em todas as Sprints, com exceção da última, durante a reunião de revisão o cliente pode solicitar melhorias em até 2 tarefas que foram finalizadas naquela Sprint. Para selecionar

as tarefas que exigirão melhoria, o sistema se baseia na qualidade funcional das atividades finalizadas. Respeitando o limite de até 2 tarefas para serem melhoradas, tarefas com qualidade funcional baixa possuem 100% de chance de terem uma melhoria solicitada, enquanto tarefas com qualidade funcional média possuem 50% de chance. Tarefas com qualidade funcional alta não necessitarão de melhorias.

Figura 27 – Revisão e Retrospectiva da Sprint - Quadro Kanban compactado

Kanban - Final da Sprint

Tarefas finalizadas: 6 de 12

FAZER (3)

FAZENDO (3)

REVISANDO (0)

FEITO (6)

Tarefa	Descrição	Progresso	Atribuído a
TAREFA-6	Criar tela de cadastro de usuários	5 / 1	Suzana (S)
TAREFA-9	Criar autenticação de usuários	7 / 5	Júlia (J)
TAREFA-3	Implementar controle de permissões por perfil: administrador, vendedor, caixa	7 / 5	Paula (P)
TAREFA-1	Possibilitar exclusão de usuários	2 / 1	Elana (E)
TAREFA-16	Criar tela de cadastro de produtos	4 / 5	Suzana (S)
TAREFA-4	Enviar e-mail para ativação de conta	2 / 5	Elana (E)

Continuar

Fonte: Autor

Tarefas selecionadas para melhoria retornarão ao backlog da próxima Sprint com prioridade máxima e com exatos 2 dias de prazo. Dessa forma, na próxima reunião de planejamento de Sprint o jogador já poderá levar essas tarefas em consideração, diferentemente das tarefas que apresentam problemas durante as reuniões diárias, que afetam o planejamento inicialmente realizado.

As métricas de qualidade funcional e técnica servem para mostrar diretamente ao jogador as consequências de suas decisões, tanto na escolha do time de desenvolvimento quanto quando tarefas foram finalizadas com qualidade baixa. Dessa forma, ele pode adaptar suas estratégias no decorrer da partida.

Ambas as qualidades, funcional e técnica, variam de 1 para baixa a 3 para alta, porém devido às probabilidades que cada desenvolvedor tem de finalizar uma tarefa com cada um desses três níveis, e às escolhas do jogador, a média real pode variar bastante. Empiricamente, notou-se que as melhores escolhas de um jogador faziam com que a média da qualidade dentre todas as tarefas normalmente ficasse acima de 2,65, enquanto as piores escolhas faziam com que a média ficasse abaixo de 2,1.

Figura 28 – Revisão da Sprint



Fonte: Autor

Dessa forma, esses valores passaram a ser considerados o limite superior e o limite inferior da qualidade, respectivamente. Como o objetivo desta métrica é informar de maneira clara ao jogador qual foi o seu desempenho, optou-se por extrapolar esses valores para o intervalo de 0 a 5, calculando a qualidade a ser exibida. Portanto, se a média for abaixo de 2,1 a qualidade funcional será 0, enquanto uma média acima de 2,65 terá qualidade funcional 5. Já os valores de média entre 2,1 e 2,65 são extrapolados para valores entre 0 e 5, usando a seguinte fórmula:

$$\text{Qualidade Exibida} = \frac{\text{Qualidade Média} - 2,1}{2,65 - 2,1} \times 5$$

Tanto a qualidade funcional média, visualizada durante a reunião de revisão da Sprint, quanto a qualidade técnica média, visualizada durante a reunião de retrospectiva da Sprint, se utilizam da mesma fórmula para serem calculadas e exibidas.

Por fim, a métrica de satisfação do cliente é um dos pontos centrais no ScrumAge, sendo utilizada como parâmetro de maior relevância para o sucesso do projeto simulado. Na reunião de revisão de Sprint o seu cálculo leva em consideração diversos itens considerados relevantes para um maior realismo, listados a seguir:

- **Assertividade da priorização (de 0 a 1,5 pontos):** compara a prioridade definida pelo jogador com a prioridade desejada pelo cliente na reunião de planejamento da Sprint. Recompensa o jogador de acordo com o seu cuidado em entender o que foi solicitado pelo cliente e priorizar corretamente, sendo o item de maior importância.

$$\textit{Pontuação Priorização} = \frac{\textit{Tarefas priorizadas corretamente na Sprint}}{\textit{Total de tarefas priorizadas na Sprint}} \times 1,5$$

- **Assertividade do planejamento (de 0 a 0,5 ponto):** compara a quantidade de tarefas finalizadas na Sprint com a quantidade de tarefas existentes no backlog da Sprint, incluindo tarefas que voltaram para correção após a descoberta de um problema em uma reunião diária. Este parâmetro considera que quanto menos tarefas tiverem sido finalizadas, menor é a satisfação do cliente, avaliando se o jogador sobrecarregou seu time de desenvolvimento.

$$\textit{Pontuação Planejamento} = \frac{\textit{Tarefas finalizadas na Sprint}}{\textit{Total de tarefas no backlog da Sprint}} \times 0,5$$

- **Quantidade de problemas ocorridos durante a Sprint (0 a 1,0 ponto):** quanto mais problemas tiverem ocorrido ao longo da Sprint, menor é a satisfação do cliente. Como no máximo 3 tarefas podem apresentar problema em uma mesma Sprint, a pontuação é calculada conforme a fórmula abaixo:

$$\textit{Pontuação Problemas} = 1,0 - \frac{\textit{Tarefas com problema na Sprint}}{3}$$

- **Quantidade de melhorias solicitadas na revisão da Sprint (de 0 a 1,0 ponto):** quanto mais melhorias solicitadas na Revisão da Sprint, menor a satisfação do cliente. Como no máximo 2 tarefas podem ter melhorias solicitadas na mesma Revisão de Sprint, a pontuação é calculada conforme a fórmula abaixo:

$$\textit{Pontuação Melhorias} = 1,0 - \frac{\textit{Melhorias solicitadas na revisão da Sprint}}{2}$$

- **Qualidade Funcional média das tarefas finalizadas na Sprint (de 0 a 1,0 ponto):** quanto mais alta a média da qualidade funcional, maior é a satisfação do cliente. Este item utiliza o mesmo cálculo de qualidade média explicado acima, porém extrapolando de 0 a 1, ao invés de extrapolar de 0 a 5, como é exibido ao jogador.

A soma das pontuações de cada um dos itens anteriores indica qual é o valor da satisfação do cliente ao final da reunião de Revisão da Sprint. Todos os itens tiveram seus cálculos e pesos balanceados até que o resultado obtido fosse condizente com o esperado, fazendo seu papel educativo.

Finalizada a reunião de revisão da Sprint, onde todas as partes interessadas estiveram presentes para inspecionar o incremento do produto, segue-se para a reunião de retrospectiva da Sprint.

4.3.4. Reunião de Retrospectiva da Sprint

No ScrumAge, o jogador participa da reunião de retrospectiva da Sprint (figura 29) no papel do Scrum master, apresentando quais problemas foram encontrados na Sprint atual e as métricas internas do time: a qualidade técnica média e a produtividade. Para a produtividade, é exibida tanto aquela que foi considerada na Sprint atual, que está sendo finalizada, quanto a que será utilizada na próxima Sprint, com base na qualidade técnica média da Sprint atual.

Figura 29 – Retrospectiva da Sprint



Fonte: Autor

Os problemas ocorridos são apresentados apenas para que o jogador lembre por quais situações inesperadas passou durante a Sprint. Já a qualidade técnica média é calculada exatamente da mesma forma que a qualidade funcional média, conforme explicado na seção anterior.

Por fim, é definida a produtividade que o time de desenvolvimento terá na próxima Sprint. Esse parâmetro é baseado exclusivamente na qualidade técnica média, porém é um número inteiro ao invés de um número real. Ele leva em consideração os limites inferior e superior de qualidade média, conforme foi explicado na seção anterior, porém o limite superior 2,65 é arredondado para 2,7. A tabela 5 descreve como é feita essa definição.

Tabela 5 – Definição da produtividade

Produtividade	Qualidade Técnica Média
Mínima (1)	$\leq 2,1$
Baixa (2)	$> 2,1$ e $\leq 2,3$
Média (3)	$> 2,3$ e $\leq 2,5$
Alta (4)	$> 2,5$ e $\leq 2,7$
Máxima (5)	$> 2,7$

A reunião de retrospectiva de Sprint marca o fim da Sprint. Neste momento o sistema verifica se esta foi a última Sprint do projeto. Não sendo, o jogo segue para a reunião de

planejamento da próxima Sprint. Caso contrário, o sistema exibe a tela de fim de jogo, como pode ser visto na próxima seção.

4.3.5. Fim de Jogo

Ao término da última Sprint do projeto, o ScrumAge apresenta a tela de fim de jogo (figura 30). Conforme pode ser visto, esta tela é dividida em três seções: fim de jogo, considerações finais do cliente e considerações finais do time Scrum. Ela oferece um *feedback* textual para o jogador considerando seu desempenho em cada métrica do jogo e no seu desempenho em geral, além de calcular e exibir a pontuação em relação a cada uma dessas métricas e a pontuação total da partida. Essa é a pontuação que será exibida na tela de classificação (figura 14).

O cálculo da pontuação do jogador atribui um peso a cada uma das métricas obtidas ao final da partida, calculando sua pontuação individualmente. Ele é separado em duas partes. A primeira consiste na avaliação das quatro métricas que compõem a satisfação final do cliente: (i) quantidade de tarefas finalizadas; (ii) qualidade funcional média de todas as tarefas finalizadas; (iii) satisfação média do cliente em todas as Sprints; e (iv) orçamento restante ao final do projeto. Já a segunda aborda a única métrica que influencia as considerações finais do Time Scrum: a qualidade técnica média de todas as tarefas concluídas.

Figura 30 – Fim de Jogo

ScrumAge
Jogar
Classificação
Wiki
UserTest
Sair

Fim do Projeto "Sistema de Gestão de Estoque e de Vendas"

FIM DE JOGO

Pontuação Final

6.100 + 2.000 = 8.100 Pontos

Classificação: 18º lugar

Mensagem Final dos Autores

Muito obrigado por ter nos acompanhado até o fim nessa jornada! Esperamos ter contribuído de alguma forma para o seu aprendizado no **Scrum** e, talvez, em alguns outros conhecimentos úteis. Sabemos que ainda há muito a ser melhorado no jogo, e por isso contamos com o seu **feedback**, para que possamos seguir evoluindo.

CONSIDERAÇÕES FINAIS DO CLIENTE



Finalmente chegamos ao final do projeto! O caminho não foi fácil, mas temos nosso sistema funcionando, inclusive nas nossas duas novas lojas. Desde já gostaríamos de **agradecer por todo o esforço seu e da sua equipe**, pois sabemos que foi um grande desafio chegarmos até aqui.

Várias funcionalidades que pedimos ficaram faltando. Mesmo que a maior parte delas não seja essencial, esperávamos que vocês conseguissem deixar o sistema mais completo.

Adoramos a qualidade do que foi entregue! Sua equipe é realmente bastante profissional, pois tiveram muito cuidado com os detalhes, onde deram ótimas soluções para as funcionalidades que solicitamos! Só temos elogios para todos vocês. **Parabéns!**

Satisfação Final do Cliente

♥♥♥♥
♥♥

= 6.100 (de 10.000) pontos

Vocês **conseguiram gerenciar muito bem o orçamento** reservado para a contratação do time de desenvolvimento! Ficamos muito felizes em saber que sobramos mais de 10% do orçamento total do projeto, **muito obrigado!**

Em resumo, **o resultado final ficou bom.** Sabemos que o prazo era muito apertado e que seria difícil construirmos o sistema que havíamos imaginado neste tempo, mas **vocês fizeram um trabalho satisfatório.** No futuro entraremos em contato para evoluirmos esse sistema. Até a próxima!

Tarefas Finalizadas

35 de 40

+ 500 (de 3.000) pontos

Qualidade Funcional

★
★
★
★
★

+2.700 (de 3.000) pontos

Satisfação Média do Cliente

♥♥♥
♥♥

+2.200 (de 4.000) pontos

Orçamento Restante

R\$ 8.000,00

+700 (de 2.000) pontos (bônus)

CONSIDERAÇÕES FINAIS DO TIME SCRUM



O desenvolvimento deste projeto foi **bastante produtivo!** Utilizando o **Scrum** e de diversas boas práticas de desenvolvimento, conseguimos usar e aperfeiçoar nossas habilidades para construir um sistema de **qualidade exemplar!** Trabalhar com um time tão dedicado e capaz, resolvendo problemas de forma rápida e eficiente, é o sonho de qualquer desenvolvedor! Com um código tão bem escrito quanto esse, a evolução e manutenção do software tende a ser simples, então todos que lidarem com ele poderão nos agradecer no futuro!

Foi bom trabalhar com você. Com certeza **aprendemos muito sobre o gerenciamento de projetos** e aperfeiçoamos nossos conhecimentos sobre o **Scrum**. Esperamos poder trabalhar com você novamente em breve. **Muito obrigado, sucesso!**

Produtividade Média

⚡
⚡
⚡
⚡
⚡

Qualidade Técnica

★
★
★
★
★

= 2.000 (de 2.000) pontos

Fonte: Autor

Conforme foi descrito na seção da etapa do desenvolvimento que aborda o balanceamento realizado no ScrumAge, todos os valores e pesos para a pontuação foram definidos empiricamente visando tornar o jogo mais realista, após diversas partidas de teste terem sido jogadas e analisadas. A seguir, serão descritos os parâmetros finais utilizados para o cálculo da pontuação.

Primeiramente, as pontuações das métricas relacionadas à satisfação final do cliente são calculadas e exibidas. Elas podem somar um total de até 12.000 pontos, onde até 10.000 são decorrentes da satisfação direta do cliente com o desenvolvimento realizado, enquanto até 2.000 pontos são oferecidos como bônus pelo orçamento poupado pelo jogador. A seguir encontram-se os pesos e os cálculos realizados a partir de cada uma das métricas que compõem a pontuação da satisfação final do cliente:

- **Quantidade de tarefas finalizadas (de 0 a 3.000 pontos):** para obter a pontuação máxima, o jogador precisa que todas as 40 tarefas do projeto estejam concluídas. Cada tarefa não finalizada representa uma penalidade de 500 pontos para o jogador. Sendo assim, se até 34 tarefas tiverem sido finalizadas, o jogador ficará com pontuação 0 neste quesito. Para valores entre 34 e 40 tarefas finalizadas, a seguinte fórmula traduz essa lógica:

$$\textit{Pontuação Tarefas} = 3000 - \textit{Tarefas não finalizadas} \times 500$$

- **Qualidade funcional média das tarefas finalizadas (de 0 a 3.000 pontos):** o cálculo realizado é equivalente ao presente na reunião de revisão da Sprint, porém considerando um intervalo de 0 a 3.000 pontos:

$$\textit{Pontuação Qualidade Funcional} = \frac{\textit{Qualidade Média} - 2,1}{2,65 - 2,1} \times 3.000$$

- **Satisfação média do cliente em todas as Sprints (de 0 a 4.000 pontos):** este quesito é calculado através da média simples da satisfação do cliente nas Sprints do projeto. Como são 4 Sprints e a satisfação em cada Sprint varia entre 0 e 5, a fórmula que traduz esta média a seguinte:

$$\textit{Pontuação Satisfação Média} = \frac{\sum_{k=1}^4 \textit{Satisfação na Sprint } k}{4 \times 5} \times 4.000$$

- **Orçamento restante ao final do projeto (de 0 a 2.000 pontos bônus):** caso o jogador escolha o time de desenvolvimento com menor senioridade possível, composto por 2 estagiários com salário de R\$ 1.000,00 e 2 júniores com salário de R\$ 3.000,00 cada, o orçamento restante ao final do projeto de 2 meses terá seu valor máximo possível de R\$ 24.000,00. Dessa forma, a pontuação é máxima quando este é

o valor restante, enquanto é mínima quando todo o orçamento é utilizado. A fórmula a seguir representa a pontuação bônus:

$$\text{Pontuação Orçamento} = \frac{\text{Orçamento restante}}{24.000} \times 2.000$$

A soma das pontuações de cada um dos itens anteriores indica qual é a pontuação da satisfação do cliente ao final do projeto. Já o cálculo da única métrica do Time Scrum está descrito a seguir:

- **Qualidade técnica média das tarefas finalizadas (de 0 a 2.000 pontos):** o cálculo é similar ao realizado para a qualidade funcional média, porém considerando um intervalo de 0 a 2.000 pontos:

$$\text{Pontuação Qualidade Técnica} = \frac{\text{Qualidade Média} - 2,1}{2,65 - 2,1} \times 2.000$$

A última etapa do ScrumAge consiste na exibição do feedback textual ao jogador. Para isso, a tela possui 8 parágrafos distribuídos pelas três seções. Do 3º ao 7º parágrafo, os textos exibidos são personalizados, existindo três versões para cada um deles. A versão exibida depende do desempenho do jogador na partida: alto, médio ou baixo. Os parágrafos de 3 a 6 abordam as métricas que influenciam nas considerações finais do cliente. Já o parágrafo 7 fala sobre a métrica de qualidade técnica média ao final da partida, a única que afeta as considerações finais do time Scrum. A tabela 6 mostra qual versão do texto será exibida de acordo com o desempenho do jogador naquele tema, enquanto o apêndice B traz os textos na íntegra.

Tabela 6 – Textos de feedback: desempenho por tema

Parágrafo	Tema	Desempenho		
		Alto	Médio	Baixo
3	Quantidade de Tarefas Finalizadas	= 40	>= 36 e < 40	< 36
4	Qualidade Funcional Média	>= 4	>= 2,5 e < 4	< 2,5
5	Orçamento Restante	> 4.000	> 0 e <= 4000	= 0
6	Satisfação Final do Cliente	>= 4	>= 2,5 e < 4	< 2,5
7	Qualidade Técnica Média	>= 4	>= 2,5 e < 4	< 2,5

5. AVALIAÇÃO DO SCRUMAGE

Após a conclusão do desenvolvimento do ScrumAge, foi solicitado que profissionais da área de tecnologia da informação jogassem e avaliassem o jogo. Através dessa avaliação foi possível mensurar diversos aspectos do sistema, entre eles se ele atende ao objetivo proposto e quais melhorias podem ser realizadas.

A avaliação foi aprovada pela CONEP, na instância do Comitê de Ética da UERJ, sob número 5.577.977, com parecer emitido em 11/08/2022.

Para o recrutamento dos participantes, foi seguido o método Bola de Neve, que utiliza uma cadeia de referências para se obter uma amostra representativa (VINUTO, 2014). Alguns profissionais da área de tecnologia da informação que fazem parte da rede de contatos do autor foram contactados, e cada um recomendou de 1 a 2 indivíduos, onde foi obtida uma amostra com 39 profissionais.

Antes de realizarem a avaliação, todos os participantes receberam um termo de consentimento (apêndice D) e um formulário de caracterização (apêndice E), visando identificar a experiência acadêmica e profissional dos avaliadores. Nesse formulário, foram solicitados dados sobre a formação acadêmica e sobre a experiência do participante em relação a engenharia de software, desenvolvimento de software, métodos ágeis e Scrum.

Para a realização da avaliação, os participantes receberam uma URL para acessarem o website onde o ScrumAge foi disponibilizado. Em seguida, realizaram um cadastro no sistema inserindo seu nome, nome de usuário, e-mail e senha, obtendo acesso ao jogo.

Após o término da partida, cada participante foi instruído a responder a três questionários sobre sua percepção do jogo e a, opcionalmente, realizar comentários (apêndice F). Todos os três questionários foram respondidos através de uma escala de tipo Likert. Nela, há uma pontuação de 1 a 5, valores correspondentes a respostas que variam entre “discordo totalmente” e “concordo totalmente”, como apresenta a tabela 7.

O primeiro questionário solicitado trouxe 11 questões relacionadas ao conteúdo do jogo, enquanto o segundo realizou 4 questões sobre a sua interface. Já o último questionário conteve 10 questões relacionadas à usabilidade do sistema. Por fim, o avaliador teve a opção de realizar comentários sobre o que mais gostou e sobre o que não gostou, ou mudaria, no ScrumAge.

Tabela 7 – Escala tipo Likert

Pontuação	Resposta
1	Discordo totalmente
2	Discordo
3	Sem opinião
4	Concordo
5	Concordo totalmente

O questionário de conteúdo foi elaborado visando entender a percepção dos avaliadores sobre os principais objetivos do trabalho: a relevância para o aprendizado do Scrum, a precisão dos conceitos apresentados, o realismo e a jogabilidade em geral. Já o questionário de interface possui perguntas relacionadas ao formato de apresentação do conteúdo, buscando definir principalmente se o jogo atende aos requisitos não funcionais **RN4** (linguagem clara e atrativa) e **RN5** (interface simples, intuitiva e agradável). Por fim, para a avaliação da usabilidade foi utilizado o instrumento System Usability Scale - SUS (BROOKE, 1996), que será explicado na seção sobre a análise dos resultados de usabilidade.

A primeira seção deste capítulo aborda a análise dos resultados, iniciando pelas respostas do formulário de caracterização dos participantes; seguidas das respostas dos questionários sobre o conteúdo, a interface e a usabilidade do ScrumAge; e finalizando com os comentários dos avaliadores. Já a segunda seção traz uma análise dos resultados obtidos.

5.1. Análise dos Resultados

A análise dos resultados foi dividida em cinco partes. A primeira mostra uma síntese sobre as respostas ao questionário de caracterização dos participantes. A segunda, a terceira e a quarta partes abordam, respectivamente, os resultados dos três questionários sobre o ScrumAge: conteúdo, interface e usabilidade. Por fim, a última parte reúne os comentários realizados pelos avaliadores.

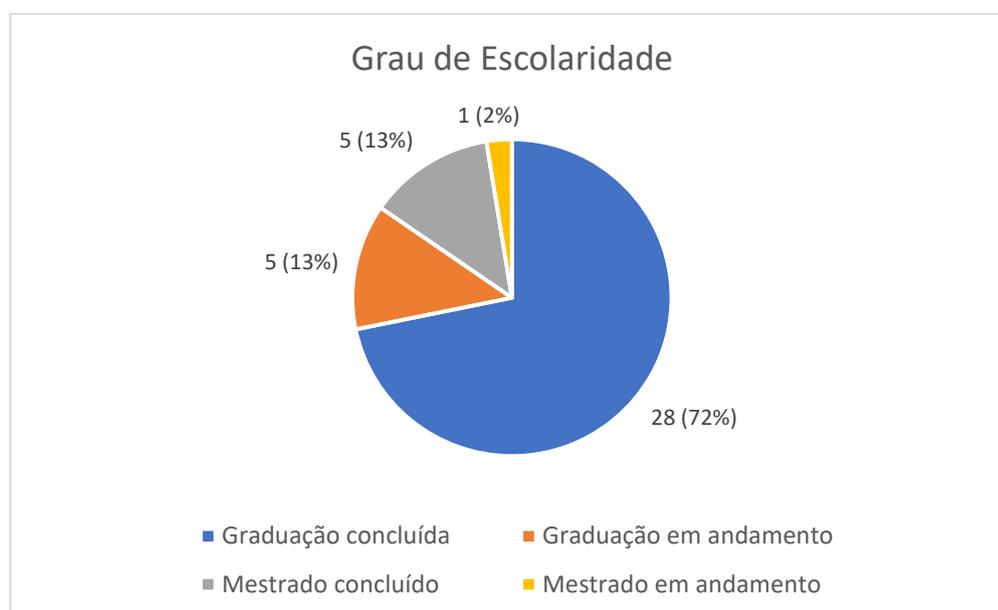
5.1.1. Caracterização dos Participantes

A primeira parte do formulário de caracterização dos participantes solicitou informações sobre a formação acadêmica do avaliador: grau de escolaridade, ano de ingresso,

ano de conclusão ou de previsão de conclusão e instituição de ensino. Estas informações têm como objetivo traçar o perfil acadêmico dos avaliadores.

A distribuição dos graus de escolaridade dentre os 39 participantes é mostrada pelo gráfico presente na figura 31. Conforme pode ser visto, a maior parte dos avaliadores (28 participantes ou 72% do total) possui a graduação completa. Além disso, 5 avaliadores (13%) estão com a graduação em andamento, com a mesma quantidade de participantes possuindo o mestrado concluído. Por fim, 1 avaliador (2%) estava com o mestrado em andamento no momento da pesquisa.

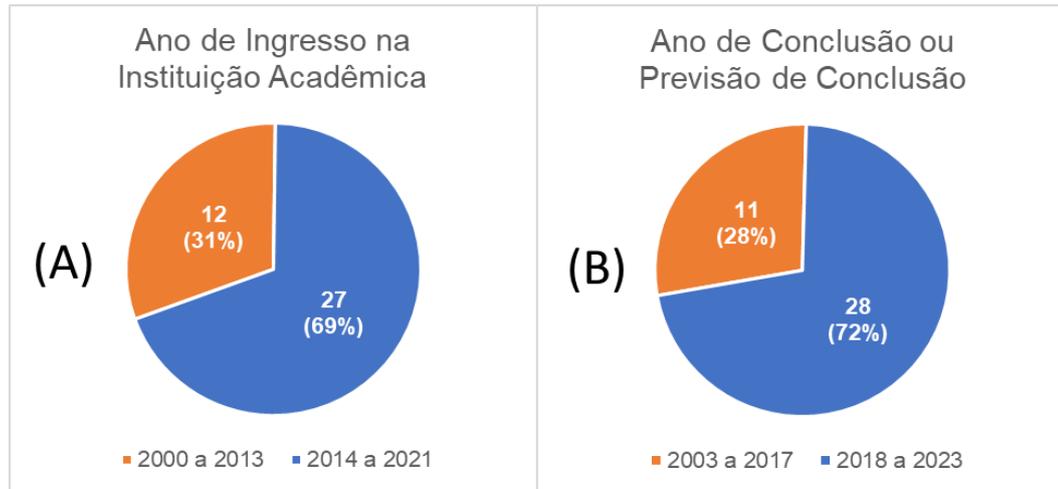
Figura 31 – Grau de Escolaridade



Fonte: Autor

Todos os avaliadores ingressaram em suas instituições acadêmicas entre 2000 e 2021, com conclusão ou previsão de conclusão do curso entre 2003 e 2023. A figura 32A exibe o ano de ingresso, onde 69% dos participantes iniciaram seus estudos no ensino superior nos últimos 8 anos. Já a figura 32B mostra que, embora todos os participantes tenham concluído sua formação acadêmica nos últimos 20 anos, ou irão concluir neste ano ou no próximo, 28 (72%) dos 39 avaliadores ainda estava estudando nos últimos 5 anos. Sendo assim, é visível que a maioria dos participantes possui uma formação bem recente, onde os conceitos de engenharia de software que aprenderam em sala de aula tendem a ser bem similares ao lecionado atualmente.

Figura 32 – Ano de Ingresso e de Conclusão de Curso



Fonte: Autor

Os 39 participantes disseram ter estudado em 21 instituições de ensino diferentes, evidenciando a diversidade de suas formações. Destas, 2 se destacaram: a UFRJ, com 8 participantes, e a UNESA, com 7 avaliadores ao todo. Além delas, 3 dos participantes disseram ter estudado na UERJ, enquanto 2 estudaram em cada uma das seguintes: UVA, UNILASALLE e UNICARIOCA. Os 15 outros participantes estudaram em instituições de ensino distintas, com uma ocorrência para cada. A figura 33 relaciona as instituições de ensino e a quantidade de participantes formados em cada uma delas.

A segunda parte do formulário solicitou informações sobre a experiência do participante com desenvolvimento ágil. Para esta questão, o avaliador poderia selecionar todas as opções que melhor descrevessem sua experiência. A figura 34 mostra a quantidade de pessoas que marcaram cada uma das opções.

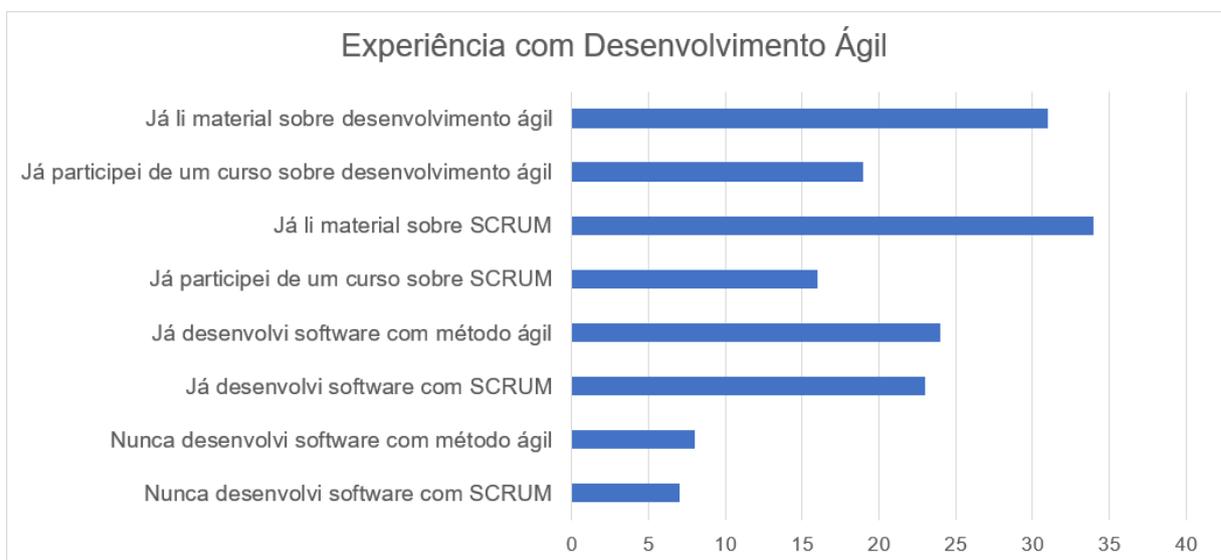
Dos 39 participantes, 34 (87%) já haviam lido material sobre o Scrum, onde 16 (41%) chegaram a participar de um curso sobre esta metodologia. Já 31 (79%) das pessoas haviam lido material sobre métodos ágeis em geral, com 19 (49%) tendo participado de cursos sobre o assunto. Em relação à experiência com desenvolvimento, 24 (62%) responderam que já haviam desenvolvido software com algum método ágil, onde 23 (59%) indicaram ter utilizado o Scrum para esta finalidade. Somente 8 participantes (21%) disseram nunca terem desenvolvido software com método ágil, e 7 (18%) disseram nunca terem desenvolvido software com Scrum. Sendo assim, os resultados reforçam o que já havia sido constatado no 15th Annual State Of Agile Report (DIGITAL.AI, 2021): o interesse por metodologias ágeis é grande, principalmente pelo Scrum.

Figura 33 – Instituições de Ensino



Fonte: Autor

Figura 34 – Experiência com Desenvolvimento Ágil



Fonte: Autor

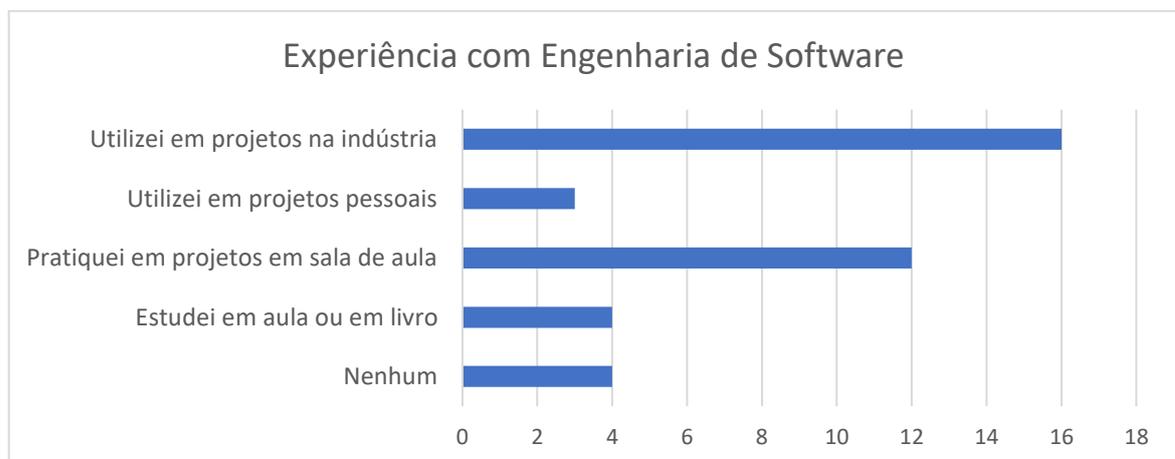
Como última seção do formulário de caracterização, foi solicitado que os participantes utilizassem a escala presente na tabela 8 para indicarem seu grau de experiência com os seguintes assuntos: (i) engenharia de software; (ii) desenvolvimento de software; (iii) desenvolvimento de software com método ágil; e (iv) desenvolvimento de software com Scrum.

Tabela 8 – Escala de grau de experiência

Pontuação	Resposta
1	Nenhum
2	Estudei em aula ou em livro
3	Pratiquei em projetos em sala de aula
4	Utilizei em projetos pessoais
5	Utilizem em projetos na indústria

Considerando a experiência com engenharia de software, a opção mais selecionada (16 de 39, ou 41% do total) pelos participantes foi a correspondente ao grau máximo de experiência disponível na escala, utilizando em projetos na indústria. A segunda alternativa mais selecionada foi a prática em projetos em sala de aula, indicada por 12 pessoas (31%). As opções de ter estudado em aula ou livro ou nunca ter estudado foram selecionadas por 4 participantes (10%) cada. Por fim, apenas 3 avaliadores (8%) indicaram terem utilizado a engenharia de software em projetos pessoais. Esses dados podem ser vistos na figura 35.

Figura 35 – Experiência com Engenharia de Software



Fonte: Autor

Já em relação à experiência do participante com desenvolvimento de software, a maioria das pessoas (27 de 39, representando 69% do total) indicou já ter utilizado em

projetos na indústria. Em segundo lugar, 6 participantes (15%) disseram ter praticado em projetos de sala de aula. Por fim, as demais opções foram selecionadas por apenas 2 pessoas (5% da amostra total) cada. A figura 36 mostra esses dados.

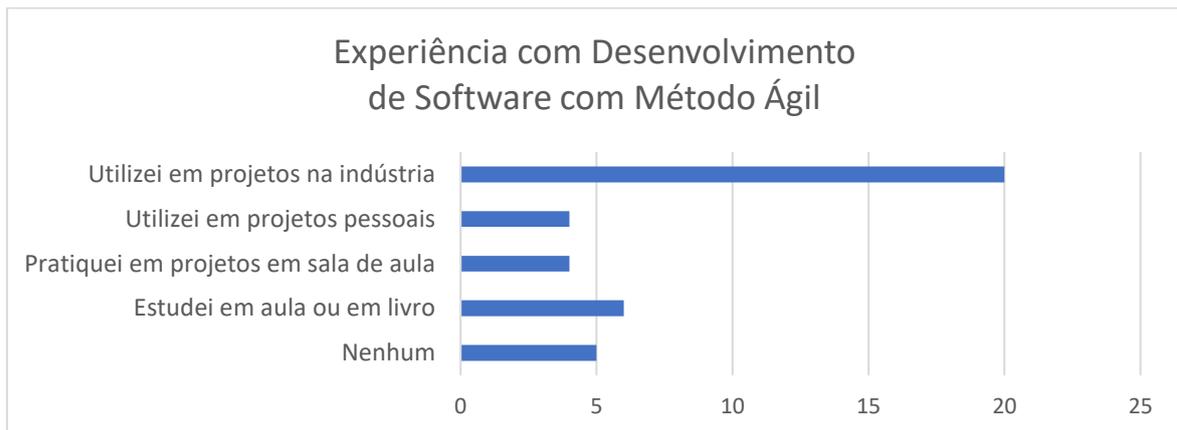
Figura 36 – Experiência com Desenvolvimento de Software



Fonte: Autor

A figura 37 mostra que 20 (51%) dos 39 participantes indicaram já terem realizado na indústria o desenvolvimento de software com método ágil. Na segunda opção mais selecionada, 6 participantes (15%) disseram ter estudado apenas em aula ou em livro. Além disso, 5 pessoas (13%) indicaram não ter nenhuma experiência com o assunto. Em último lugar, as duas opções restantes foram marcadas por 4 participantes (10%) cada.

Figura 37 – Experiência com Desenvolvimento de Software com Método Ágil

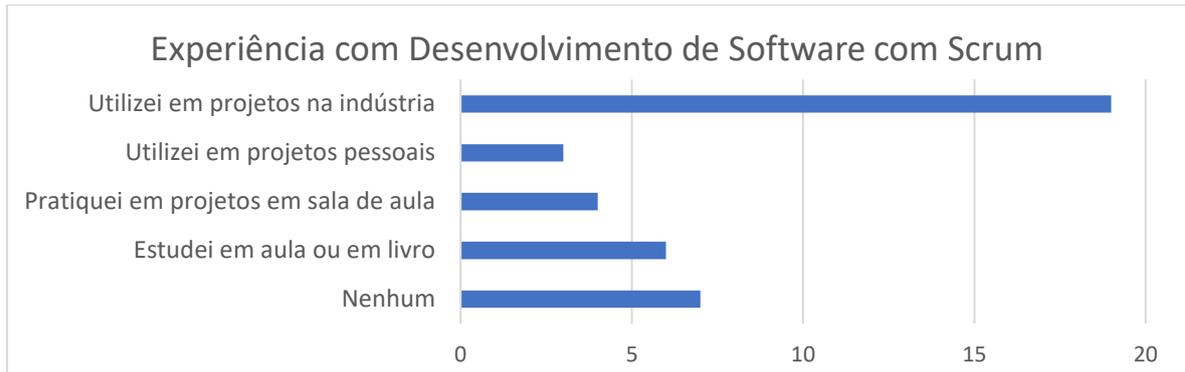


Fonte: Autor

A última questão do formulário de caracterização dos participantes solicitou o grau de experiência com desenvolvimento de software com Scrum, onde as respostas podem ser vistas na figura 38. Neste quesito, a opção mais escolhida (19 participantes, ou 49% do total) foi a

que o participante indica que utilizou em projetos na indústria. Em segundo lugar, 7 pessoas (18%) disseram não ter nenhuma experiência neste tópico. Além disso, 6 avaliadores (15%) marcaram a opção que estudaram em aula ou em livro, seguida de 4 pessoas (10%) que indicaram ter praticado em projetos em sala de aula. Por último, 3 participantes (8%) indicaram ter utilizado em projetos pessoais.

Figura 38 – Experiência com Desenvolvimento de Software com Scrum



Fonte: Autor

Ao fim da análise das respostas do formulário de caracterização, é possível concluir que a maior parte dos participantes finalizou seus estudos nos últimos anos e já possui experiência profissional nos temas abordados pelo ScrumAge. Entretanto, observa-se que ainda há muitos profissionais que não utilizaram os métodos ágeis na prática profissional. Isso reforça a importância de que o ensino desses temas tem para o desenvolvimento dos profissionais de tecnologia.

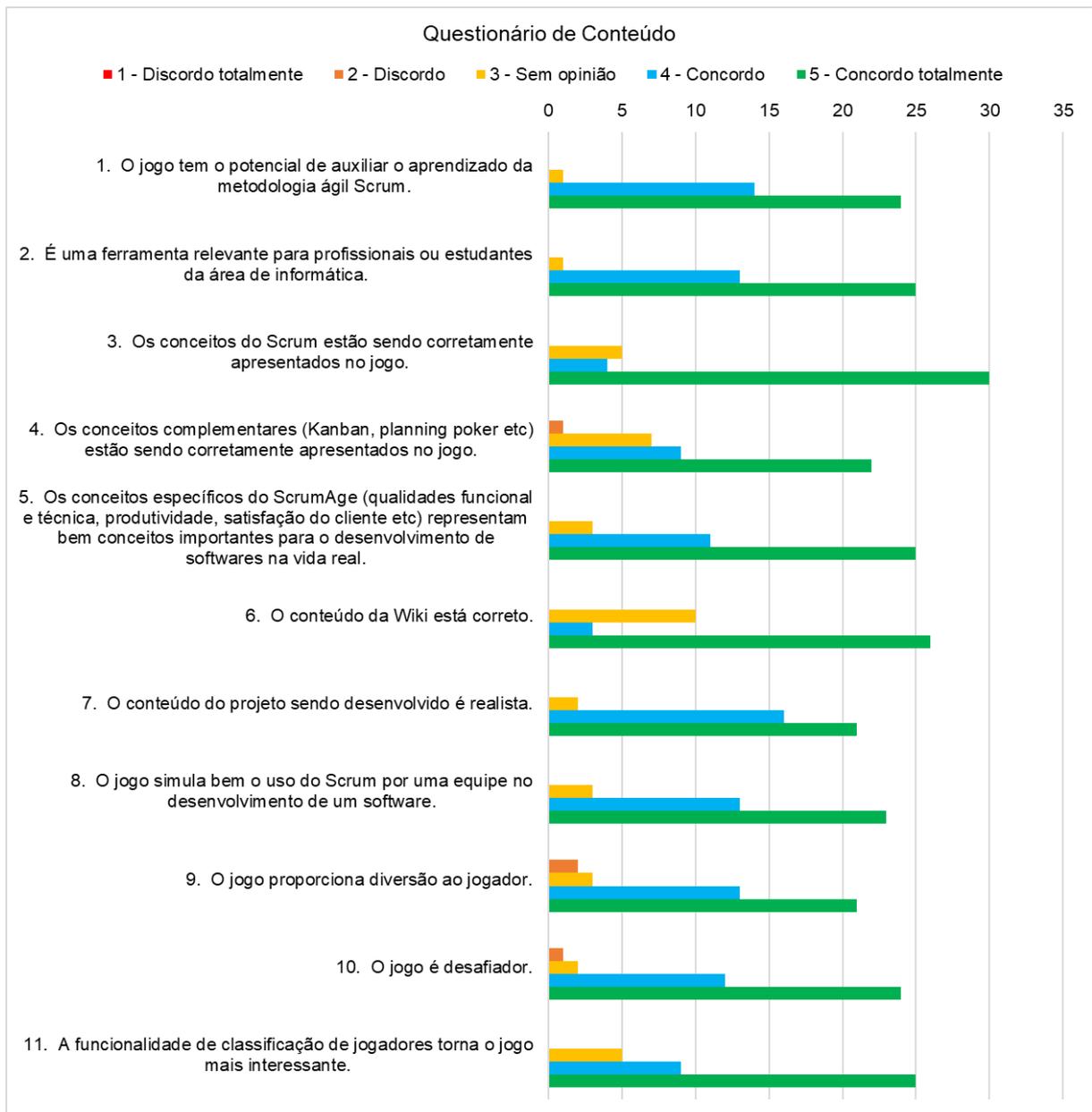
5.1.2. Conteúdo

O questionário de conteúdo consiste em 11 questões que visam identificar a opinião dos avaliadores sobre a relevância e a qualidade do conteúdo textual e conceitual apresentado para o aprendizado do Scrum. A figura 39 apresenta todas as questões e a quantidade de pessoas que responderam às questões.

A tabela 9 traz outra visão das respostas obtidas, as separando em três categorias: avaliações positivas, avaliações neutras e avaliações negativas. Utilizando esta separação, chega-se ao valor de que 89,3% das avaliações de conteúdo foram positivas, com 9,8% avaliações neutras e apenas 0,9% negativas.

Conforme observado na Figura 39, todas as questões obtiveram altos graus de concordância, apontando que o ScrumAge foi bem aceito pela grande maioria dos avaliadores. Como essas questões são relacionadas aos principais objetivos deste trabalho, como auxiliar o aprendizado do Scrum e de conceitos complementares de forma lúdica, as respostas trazidas nesta seção já indicam resultados positivos em relação aos objetivos educacionais do do ScrumAge.

Figura 39 – Gráfico das Respostas ao Questionário de Conteúdo



Fonte: Autor

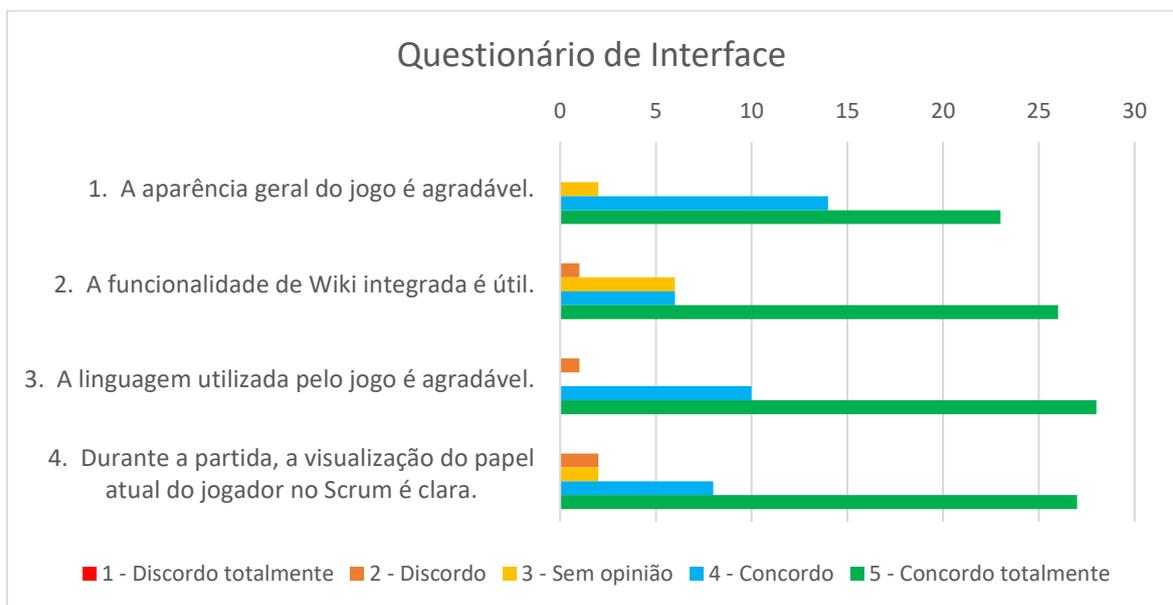
Tabela 9 – Dados das Respostas ao Questionário de Conteúdo

Questão	Avaliações Positivas	Avaliações Neutras	Avaliações Negativas
1	38 (97,4%)	1 (2,6%)	0 (0%)
2	38 (97,4%)	1 (2,6%)	0 (0%)
3	34 (87,2%)	5 (12,8%)	0 (0%)
4	31 (79,5%)	7 (17,9%)	1 (2,6%)
5	36 (92,3%)	3 (7,7%)	0 (0%)
6	29 (74,4%)	10 (25,6%)	0 (0%)
7	37 (94,9%)	2 (5,1%)	0 (0%)
8	36 (92,3%)	3 (7,7%)	0 (0%)
9	34 (87,2%)	3 (7,7%)	2 (5,1%)
10	36 (92,3%)	2 (5,1%)	1 (2,6%)
11	34 (87,2%)	5 (12,8%)	0 (0%)
Média:	34,8 (89,3%)	3,8 (9,8%)	0,4 (0,9%)

5.1.3. Interface

O questionário de avaliação da interface possui 4 questões sobre a forma pela qual a apresentação dos conteúdos do jogo é realizada, com foco na verificação dos requisitos não funcionais **RN4** (linguagem clara e atrativa) e **RN5** (interface simples, intuitiva e agradável). A figura 40 apresenta quais foram as questões abordadas e as respostas obtidas em cada uma delas.

Figura 40 – Gráfico das Respostas ao Questionário de Interface



Utilizando a mesma lógica aplicada na seção anterior, a tabela 10 mostra os resultados correspondentes ao questionário de interface. Para este, 91% das avaliações foram positivas, com 6,4% avaliações neutras e somente 2,6% negativas. Sendo assim, as respostas obtidas neste questionário demonstram uma ótima avaliação também em relação à interface do sistema.

Tabela 10 – Dados das Respostas ao Questionário de Interface

Questão	Avaliações Positivas	Avaliações Neutras	Avaliações Negativas
1	37 (94,9%)	2 (5,1%)	0 (0%)
2	32 (82,1%)	6 (15,4%)	1 (2,6%)
3	38 (97,4%)	0 (0%)	1 (2,6%)
4	35 (89,7%)	2 (5,1%)	2 (5,1%)
Média:	35,5 (91%)	2,5 (6,4%)	1 (2,6%)

5.1.4. Usabilidade

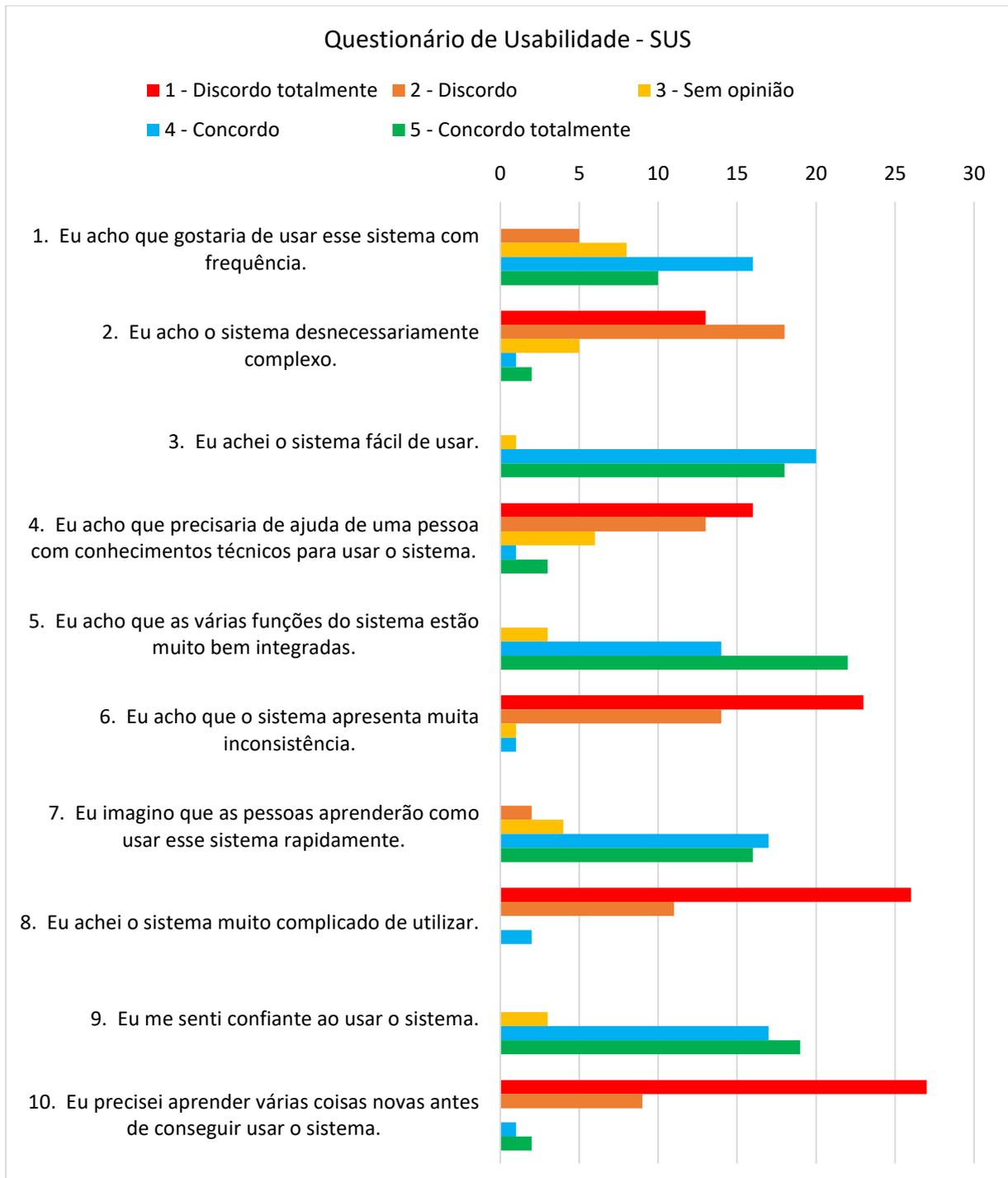
O último questionário visa avaliar a usabilidade do sistema. Para esta avaliação, optou-se pela utilização do questionário System Usability Scale - SUS (BROOKE, 1996). Segundo Brooke, usabilidade é difícil de se mensurar quantitativamente. Entretanto, quando se tenta medir esta propriedade há alguns pontos que devem ser abordados: efetividade, a capacidade dos usuários completarem tarefas utilizando o sistema; eficiência, a quantidade de recursos consumidos ao se realizar tarefas no sistema; e satisfação, as reações subjetivas dos usuários ao usarem o sistema. E o SUS avalia estes pontos de forma rápida e confiável, destacando-se dentre outras escalas que não possuem um modelo de classificação final para o software. O SUS determina um valor final, que indica o grau da qualidade da usabilidade do software, já tendo sido validado para o português de Portugal (MARTINS, 2015).

No SUS, são feitas dez afirmações e os avaliadores são instruídos a responder cada uma delas com a escala de tipo Likert, descrita na tabela 7, assim como os demais questionários. Entretanto, diferentemente dos outros, no SUS as frases são intercaladas entre positivas e negativas. As frases e suas respostas estão descritas na figura 41.

Analisando individualmente as respostas obtidas, é visível que todos os temas foram positivamente avaliados, o que justifica a boa pontuação final obtida na avaliação. O único item que se destaca negativamente dos demais é o primeiro, relacionado à frequência com que o usuário gostaria de usar o sistema. Ele foi o único que, embora ainda tenha uma maioria de avaliações positivas, apresentou uma quantidade relativamente alta de opiniões neutras e

negativas. Provavelmente isso se deve ao ScrumAge ser um jogo para aprendizado e treinamento, ainda não possuindo cenários variados o suficiente para atrair novamente jogadores que já tenham finalizado algumas partidas e como pode ser observado no questionário sobre o perfil dos participantes, a maioria já conhece o Scrum (28 participantes), ou já usa métodos ágeis nos seus projetos (26 participantes).

Figura 41 – Gráfico das Respostas ao Questionário de Usabilidade - SUS



Fonte: Autor

De acordo com Brooke (1996), uma vez obtidas as notas, a pontuação SUS de cada avaliador deve ser calculada individualmente, observando a pontuação contribuída por cada item da avaliação, da seguinte maneira:

- **Afirmações positivas:** A pontuação para afirmações de índice ímpar (1, 3, 5, 7 e 9) devem ser calculadas conforme a equação a seguir:

$$\text{Pontuação item ímpar} = \text{valor escolhido} - 1$$

- **Afirmações negativas:** A pontuação para afirmações de índice par (2, 4, 6, 8 e 10) são calculadas de acordo com a seguinte fórmula:

$$\text{Pontuação item par} = 5 - \text{valor escolhido}$$

Após o cálculo as contribuições individuais de cada item, a pontuação total de um único avaliador pode variar entre 0 e 100, e é obtida usando o cálculo a seguir:

$$\text{Pontuação total do avaliador} = 2,5 \times \sum_{k=1}^{10} \text{Pontuação item } k$$

Por fim, para obter a pontuação final do sistema no SUS, basta realizar a média aritmética da pontuação de todos os avaliadores.

Seguindo o procedimento descrito para as respostas apresentadas na figura 41, o ScrumAge obteve a pontuação final de 82 pontos. Bangor et al. (2008) propuseram uma associação entre a pontuação no SUS e adjetivos, conforme pode ser visto na tabela 11. Segundo ela, a pontuação obtida pelo ScrumAge está entre boa e excelente.

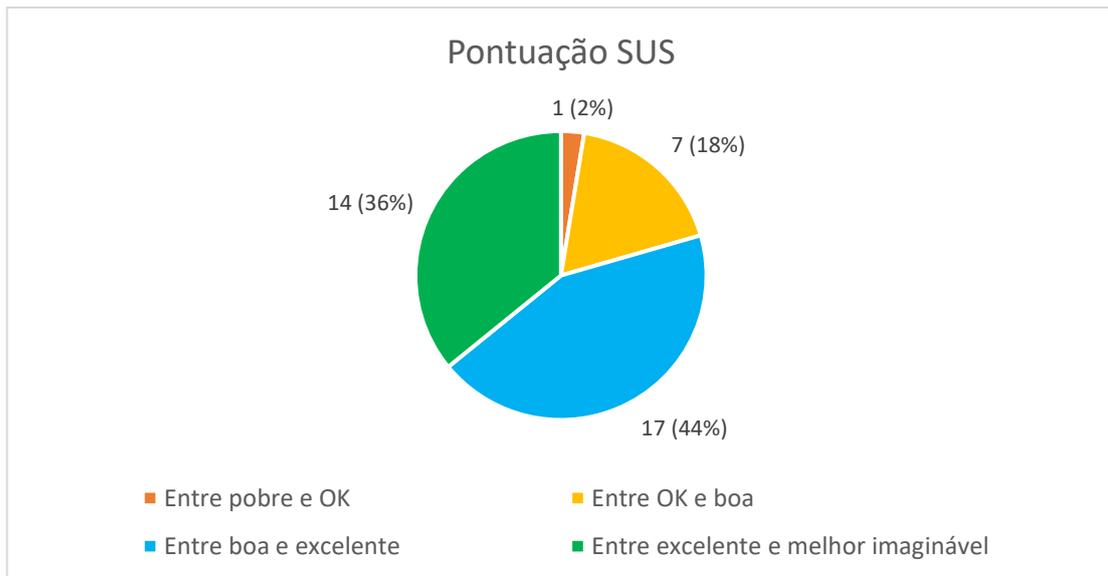
Tabela 11 – Pontuação SUS e adjetivos correspondentes, segundo Bangor et al. (2008)

Pontuação SUS	Adjetivo
De 0 a 25	Pior imaginável
De 26 a 38	Entre pior imaginável e pobre
De 39 a 52	Entre pobre e OK
De 52 a 73	Entre OK e boa
De 73 a 85	Entre boa e excelente
De 85 a 100	Entre excelente e melhor imaginável
100	Melhor imaginável

Contabilizando a pontuação SUS calculada para cada avaliador e seguindo o proposto por Bangor et al., obteve-se o gráfico que pode ser visto na figura 42. Ele mostra que 38

(98%) dos 39 participantes avaliaram o sistema com uma usabilidade positiva, definida como entre OK e melhor imaginável. Mesmo considerando apenas as duas faixas mais positivas, 31 (80%) dos avaliadores acharam a usabilidade do sistema entre boa e melhor imaginável. Por fim, apenas 1 participante (2%) registrou uma percepção negativa, entre pobre e OK.

Figura 42 – Adjetivo para a pontuação SUS de cada avaliador do ScrumAge



Fonte: Autor

Sendo assim, o ScrumAge também obteve sucesso na avaliação de usabilidade, assim como nas avaliações de conteúdo e interface. Como todos os questionários trouxeram resultados bastante positivos, consideramos que o jogo foi bem recebido e bem-sucedido em sua avaliação, mostrando-se apto para futuros testes em sala de aula.

5.1.5. Comentários dos participantes da pesquisa

Como última etapa da avaliação, dois campos livres e de preenchimento opcional foram deixados para que os participantes pudessem dizer o que mais gostaram no ScrumAge e o que não gostaram (ou mudariam) no jogo. Mesmo se tratando de campos opcionais, 30 (77%) dos 39 avaliadores preencheram o campo sobre o que mais gostaram, enquanto 22 participantes (56%) preencheram o campo sobre aspectos negativos e possíveis melhorias.

Seguem todas as respostas para a questão “Do que você mais gostou no jogo ScrumAge?”:

- “A forma como são trazidos cenários realistas do dia a dia para dentro do jogo / a dinâmica onde vc claramente vê o impacto de cada decisão tomada”

- “O cenário, bem realístico.”
- “A simulação de problemas ocorrendo durante as sprints foi bem bacana!! Bem similar a vida real. A identidade visual ficou muito boa!”
- “Simula de forma realista a entrada de demandas e desenrolar do scrum”
- “Das situações próximas com a realidade que encontrei na minha vivência trabalhando com scrum. Além disso, o jogo é bem intuitivo de se entender, creio que uma pessoa que não conheça scrum vá entender como se joga da mesma forma.”
- “Achei muito interessante a "tradução" de um projeto na vida real para um jogo, a forma de trazer as necessidades do cliente de forma clara para a priorização de atividades é um dos pontos que mais me interessou no jogo.”
- “Da realidade como foi sendo exibido os problemas durante o projeto e as dicas para melhorar a qualidade funcional e técnica.”
- “A dinâmica de se sentir dentro de um projeto, chegando até a se preocupar com os prazos e com os desenvolvedores demorando pra concluir uma task. Me coloquei na pele de um P.O e de um líder de time de desenvolvimento”
- “Ter que entender as prioridades dos clientes no papel do product owner.”
- “Pra mim foi bem divertido ver o cenário real de um desenvolvimento, principalmente nas tomadas de decisão como a de se refatorava o código ou não durante o desenvolvimento.”
- “Da competição, e da experiência em desenvolver um software”
- “Classificação”
- “Por em "prática" a metodologia ágil”
- “A parte mais legal é a simulação das sprints”
- “A dinâmica das Sprints em geral”
- “Ir vendo o progresso das sprints é prazeroso. Tomar as decisões sobre revisão também é uma mecânica bem interessante.”
- “De poder jogar a partir de vários perfis, tanto Scrum master como Product owner e desenvolvedor.”
- “Conteúdo sobre metodologia ágil didático e prático, e esquema de jogo interessante e desafiador.”
- “Eu curti o método usado para introduzir o princípio de desenvolvimento ágil. Se eu tivesse um sistema como esse, teria me salvado de muita dor de cabeça e tempo gastado à toa quando comecei a trabalhar em times ágeis.”

- “Achei o jogo extremamente fácil de usar e bem intuitivo. Textos simples e objetivos fluem bem no avanço do game. Acredito que da maneira proposta, pessoas sem conhecimento prévio estejam aptas ao que o game é proposto, ensinar o Scrum. Parabéns!”
- “O jogo é intuitivo, fácil de jogar e de aprender”
- “Aprendizado”
- “O jogo é desafiador e estimula a continuar jogando.”
- “Desafio”
- “Achei o jogo bem empolgante! Me deu vontade real de terminar todas as tarefas e entregar o projeto. De fato, muito bem pensado toda a dinâmica do jogo.”
- “Facilidade na utilização.”
- “A simplicidade e fluidez relacionadas à dinâmica”
- “O capricho. Imagens e textos estavam bem localizados. A escolha das paletas de cor estava agradável. Avisos pro usuário saber se tá tudo ok pra continuar. Textos em popovers. A dinâmica das tabelas de votação (prioridade e dificuldade). Board com informações sobre os sentimentos dos clientes. São tantos detalhes q fica difícil de lembrar de tudo”
- “A interface é bem amigável e as etapas são bem separadas. Isso facilita o entendimento de cada estágio das Sprints.”
- “a interface”

Como pode ser observado, as respostas para esta pergunta aberta reforçam o que foi verificado nos questionários, com a maior parte dos participantes tendo visões positivas sobre muitos aspectos do ScrumAge. Dentre os principais elogios realizados, destacam-se: o realismo ao simular um projeto de desenvolvimento; a dinâmica que o jogo usa para simular o uso do Scrum; a aparência, a interface e a usabilidade em geral; o aprendizado sobre os conceitos abordados; o desafio e a competição.

Já para a questão “Do que você não gostou (ou o que você mudaria) no jogo ScrumAge?”, embora menos respostas tenham sido preenchidas, 4 dos avaliadores escreveram respostas mais extensas com suas ideias de melhorias para o jogo. Sendo assim, primeiramente serão apresentadas as respostas menores na íntegra, enquanto as maiores terão seu resumo descrito em seguida:

- “Achei o tempo da partida um pouco extenso.”
- “Achei muito extenso, um pouco cansativo de ler as necessidades para responder. Além disso, poderia ter uma explicação do que deveria ter sido feito para ter um resultado melhor no final do jogo.”
- “Achei que tinha textos muito grandes a ponto de estar sempre tendo que ler com muita calma e atenção.”
- “Muito texto”
- “Os pop-ups e as modais aparecerem todas no mesmo lugar. Acho q os pop-ups tinham q ficar no canto superior ou inferior (lado esquerdo ou direito). Joguei no desktop”
- “Gostei de tudo! Talvez pensaria numa melhor maneira de exibir as notificações com relação aos papéis profissionais na parte de evolução das sprints. Ao clicar em próximo o sistema vai exibindo notificações dos papéis profissionais e elas acabam sendo um pouco "irritantes" cobrindo boa parte da tela ao somar com as notificações das tarefas.”
- “Algumas mensagens errôneas na transição de etapas.(Clico no botão de avançar etapa, a nova página é carregada porém aparece uma mensagem como se eu tivesse tentando avançar outra etapa sem preencher)”
- “A interface mobile às vezes Bugou um pouco. Uma imagem fora do lugar, notificações tomando todo o espaço da tela.”
- “Pouco intuitivo.”
- “Possibilidade de escolha na alocação das tarefas e repriorização no decorrer da sprint.”
- “Opção de atribuir tarefas a cada desenvolvedor”
- “Possibilidade de escolher as tarefas que irão para desenvolvimento e quem realiza cada tarefa”
- “Não dá para prever como os recursos seriam usados para atribuir a tarefa. Teria que jogar mais de uma vez para entender o jogo em si.”
- “Talvez poderia ter algum tipo de mecânica pra "refinar" as tarefas consideradas mais obscuras, o que poderia evitar revisões funcionais/técnicas.”
- “Acho que a visão do planning poker poderia trazer um pouco mais a visão de como cada um dos desenvolvedores interferem nessa estimativa ao invés de uma visão unificada já de início, mas não sei qual seria a melhor forma de trazer isso sem complicar a interface do jogo.”
- “número imutável de sprints”

- “Faltou a presença do tester no jogo, ajudaria a garantir a qualidade do produto”
- “Não ficou claro para mim como é calculada a pontuação final. Acredito que poderia estar mais destacado.”

Respostas resumidas da segunda questão:

- Resposta 1:
 - Não olhei o quadro Kanban, apenas ia clicando em continuar. Talvez por já ter familiaridade com a dinâmica deste quadro.
 - A parte menos legal foi priorizar as atividades, devido à necessidade de ficar subindo a tela para ler o texto e descendo para marcar a prioridade.
 - Não cheguei a utilizar a wiki.
- Resposta 2:
 - Jogo difícil para quem não está habituado aos termos utilizados.
 - Não entender sobre o processo de desenvolvimento faz com que o usuário não saiba qual é a melhor opção.
 - Seria bom um mini tutorial para a identificação do tempo de desenvolvimento de cada funcionalidade.
- Resposta 3 (sugestões):
 - Ter um módulo de treinamento.
 - Gerar um certificado de conclusão do jogo com horas, para ser utilizado por estudantes que precisem apresentar horas em atividades complementares.
 - Incluir outros papéis para ter times multidisciplinares, talvez em outros níveis de dificuldades, pois tornaria o jogo mais complexo.
 - Realizar cerimônias de refinamento.
 - Na tela de seleção do time, adicionar habilidades diferentes para os personagens, de forma que as tarefas realizadas possam ser influenciadas por essas habilidades pessoais, mesmo em desenvolvedores de mesma senioridade.
 - Ao final de cada sprint realizar uma avaliação do progresso do projeto, havendo a possibilidade de rever as prioridades.
 - Em relação às decisões para tarefas de qualidade baixa, adicionar opção de criar débito técnico, para resolução posterior.
- Resposta 4:
 - O algoritmo de alocação de tarefas é ingênuo, com sênior pegando tarefas de menor prioridade.

- Senti muita falta do jogo enfatizar a entrega de valor para o cliente. Se a aplicação não conseguir tangibilizar isso o jogo será jogado de forma errônea.
- Joguei duas partidas e a instrução de escolher um time balanceado me levou a uma pontuação pior do que escolher um time mais experiente.
- Seria interessante as tarefas de maior prioridade entregues agregarem maior pontuação ao final da sprint.
- Deveria haver uma ordem lógica para a entrega de certas tarefas, como por exemplo uma tela de busca só puder ser feita após a tela de cadastro da entidade.

As respostas acima mostram que todas as críticas foram bastante construtivas, contendo ótimas sugestões de melhorias para trabalhos futuros. Dentre os pontos negativos, destacam-se: a duração da partida sendo considerada extensa; a existência de muitos textos explicativos; a forma em que as notificações são exibidas; a falta de clareza sobre quais são as melhores ações do jogador e sobre o cálculo da pontuação; a impossibilidade de escolher a alocação de tarefas dentre os desenvolvedores; a falta da possibilidade de refinar e repriorizar tarefas; a dificuldade de ler a mensagem do cliente em cada planejamento de Sprint enquanto é feita a priorização.

Já em relação às sugestões de melhoria, seguem as principais: ter alguma forma de treinamento ou tutorial; realizar cerimônias de refinamento de backlog; possibilitar que o jogador escolha a alocação das tarefas; avaliar e exibir o progresso parcial do jogador no final de cada Sprint; enfatizar a entrega de valor para o cliente; incluir outros papéis, para ter equipes multidisciplinares; adicionar habilidades especiais diferentes para desenvolvedores de mesma senioridade; adicionar a opção de criar débito técnico, para que o jogador possa escolher melhorar posteriormente tarefas com baixa qualidade; gerar um certificado de conclusão com horas.

5.2. Conclusões da Avaliação

Diante de todos os dados apresentados, pode-se concluir que a avaliação do ScrumAge apresentou resultados positivos. Os 39 avaliadores jogaram um total de 54 partidas, onde a maior pontuação obtida foi de 11.000 pontos. Essa pontuação foi atingida pelo único participante que disse possuir a certificação PSM-I – Professional Scrum Master I (SCRUM.ORG, 2022). Sendo assim, a amostra obtida foi significativa, com uma boa

quantidade de participantes e partidas, gerando resultados mais confiáveis para a pesquisa. Além disso, a maior parte dos avaliadores se mostrou engajada também em comentar sobre sua experiência, sugerindo diversas melhorias, o que será de suma importância para a melhoria do jogo.

A primeira seção do formulário de caracterização dos participantes visou identificar a sua formação acadêmica. Como a rede de contatos do autor consiste principalmente de profissionais da área de tecnologia da informação, era esperado que a maioria dos participantes possuíssem a graduação concluída, conforme ocorreu. Da mesma forma, também era previsto que a maior parte dos avaliadores tivesse se formado nos últimos anos, o que também se confirmou. Por fim, as instituições de ensino onde cada participante estuda ou estudou se mostraram mais diversas do que o esperado, embora a maioria se situe no Rio de Janeiro/Brasil, estado de origem do autor.

Em relação às informações obtidas sobre a experiência dos participantes com desenvolvimento ágil, verificou-se uma grande demanda pelo aprendizado de métodos ágeis, com destaque para o Scrum, o que confirma os resultados disponíveis no 15th Annual State Of Agile Report (DIGITAL.AI, 2021). Isso se deve ao fato de que quase todos os participantes já tinham lido material sobre Scrum ou métodos ágeis, e cerca de metade deles já participou de algum curso sobre esses assuntos. Além disso, cerca de dois terços dos avaliadores indicaram já ter desenvolvido software com Scrum ou com métodos ágeis.

Na parte final do formulário de caracterização, os participantes indicaram seu grau de experiência com 4 temas relevantes no ScrumAge: (i) engenharia de software; (ii) desenvolvimento de software; (iii) desenvolvimento de software com método ágil; e (iv) desenvolvimento de software com Scrum. Para todos esses tópicos, o grau de experiência mais alto foi também o mais escolhido: a utilização em projetos na indústria. Essa informação evidencia a utilidade dos temas abordados pelo ScrumAge para desenvolvedores de software que estão no mercado de trabalho.

Na primeira parte da avaliação sobre o jogo, o primeiro questionário respondido foi o de conteúdo, com questões sobre a relevância e a qualidade do conteúdo conceitual e textual utilizado no ScrumAge. Neste questionário, considerado o de maior importância para o propósito do trabalho, quase 90% das respostas foram positivas, denotando uma aprovação

geral dos avaliadores. Apenas 0,4% das respostas foram negativas, e quase 10% foram neutras.

Dentre as avaliações neutras, é importante ressaltar que mais da metade (52%) foram relacionadas à precisão conceitual dos temas abordados (conceitos do Scrum, conceitos complementares e artigos da Wiki). Entretanto, verificou-se que a maioria dessas notas a essas questões foi dada por participantes que indicaram ter menor grau de experiência nas áreas abordadas, logo eles não poderiam emitir uma opinião sobre a exatidão dos conceitos abordados, levando-os a escolher a opção “Sem opinião”. Considerando essa informação extra, a taxa de aprovação do conteúdo do ScrumAge foi ainda maior do que a demonstrada numericamente.

Já para o questionário de interface, também foram obtidas notas bastante favoráveis, onde 67% delas corresponderam à nota máxima, e 24% à nota alta, somando 91% de respostas positivas. Em relação ao pouco mais de 6% de respostas neutras, 60% dessas foram recebidas na questão sobre a utilidade da Wiki integrada. Novamente correlacionando com o grau de experiência indicado no questionário de caracterização, foi possível verificar que essa resposta foi escolhida por avaliadores com maior experiência nos temas presentes na Wiki. Sendo assim, já era esperado que pessoas mais experientes realmente não apontassem a utilidade da Wiki, quanto àqueles que ainda estão aprendendo, pois não precisaram utilizá-la.

Para a última parte da avaliação do ScrumAge, relacionada à usabilidade do sistema, foi utilizado o questionário SUS (BROOKE, 1996). Seguindo a tendência demonstrada pelas respostas dos demais questionários, a percepção da usabilidade do sistema pelos avaliadores também foi muito boa, obtendo 82 pontos de média. Segundo a escala proposta por Bangor et al. (2008), a pontuação dada por 98% dos participantes foi positiva, com o sistema sendo classificado como com uma interface entre boa e excelente.

Na última etapa da avaliação, as duas únicas questões discursivas, embora opcionais, obtiveram uma grande quantidade de respostas: 77% dos avaliadores descreveram o que mais gostaram no ScrumAge, enquanto 56% indicaram o que menos gostaram, ou o que mudariam no jogo.

Considerando as respostas para a primeira questão, os participantes reafirmaram as percepções que já haviam demonstrado nos questionários objetivos. Os principais elogios

foram relacionados ao realismo da simulação, à dinâmica do jogo, à aparência e à usabilidade do sistema, ao aprendizado e ao desafio proporcionados pelo jogo.

Em relação ao retorno obtido para a segunda pergunta, embora menos participantes a tenham respondido, 4 deles realizaram diversas sugestões cada um, o que fez com que essa questão tivesse mais conteúdo escrito em suas respostas do que a questão anterior. Essas opiniões foram de grande influência para as melhorias futuras sugeridas nas considerações finais deste trabalho.

Ainda sobre a segunda questão, foi solicitado que os participantes realizassem tanto críticas quanto sugestões de mudança. Dentre as críticas realizadas, as principais indicaram que a duração da partida está extensa, que há textos demais no jogo e que as notificações estão sendo exibidas de forma desagradável. Como sugestões, destacaram-se a criação de um tutorial, a implementação de uma cerimônia de refinamento de backlog e dar a possibilidade de o jogador escolher a alocação das tarefas realizadas.

Comparando os resultados descritos com os trabalhos correlatos do gênero simulação (Aragão, 2022; Marinho, 2018; LEE, 2016; RODRÍGUEZ, 2015), o ScrumAge obteve as mesmas percepções positivas sobre o conteúdo apresentado e a relevância desse método para o aprendizado do Scrum. Entretanto, o jogo de Aragão (2022) foi avaliado como difícil, ao passo que Lee (2016) foi descrito como não sendo desafiador o suficiente, enquanto a avaliação do ScrumAge indicou um nível de desafio adequado. Assim como Aragão (2022), o ScrumAge também recebeu algumas críticas sobre a longa duração de uma partida, mesmo com ela sendo estimada com aproximadamente 10 minutos a menos do que o trabalho citado.

Sobre a interface, Aragão (2022) e Rodríguez (2015) relataram algumas críticas recorrentes entre boa parte dos avaliadores, relacionadas principalmente em relação à câmera da interface 3D, enquanto no ScrumAge as poucas críticas realizadas nesse aspecto foram menos frequentes e relacionadas em sua maioria com as notificações sobre a alteração de papéis e aos longos textos presentes.

Relacionando os resultados obtidos no quesito usabilidade, Marinho (2018) relatou uma percepção regular, enquanto os demais trabalhos não destacaram outras opiniões nesse tópico, exceto as já descritas mais especificamente sobre a interface. Esse trabalho também relatou alguns problemas técnicos durante sua avaliação, assim como Aragão (2022), o que não ocorreu nos testes do ScrumAge.

Por fim, a partir dos resultados descritos é possível concluir que o ScrumAge atingiu seus principais objetivos para esta avaliação, que envolviam validar se, na opinião dos avaliadores, o jogo é capaz de auxiliar o aprendizado do Scrum em seu formato atual. Além de ter conseguido responder a essa questão de forma bastante positiva, esta análise trouxe diversas sugestões importantes para futuras evoluções do sistema. Também, foi possível verificar que há grande tendência de que ele possa ser proveitoso em treinamentos corporativos ou em sala de aula, o que precisará ser confirmado em trabalhos futuros.

CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo principal propor e desenvolver um serious game de simulação para complementar o aprendizado da metodologia ágil Scrum. Para isso, foi realizada uma revisão bibliográfica visando identificar quais trabalhos correlatos já haviam sido publicados com a mesma motivação. Os resultados obtidos nessa revisão foram encorajadores, uma vez que as avaliações conduzidas nesses trabalhos retornaram percepções majoritariamente positivas, indicando potencial para o cumprimento de seu propósito.

Em contrapartida, a revisão bibliográfica também revelou lacunas relacionadas principalmente à disponibilidade em outros equipamentos e ao realismo dos jogos encontrados. Dessa forma, as características positivas e as limitações encontradas nesses trabalhos foram utilizadas como insumos para a proposta do ScrumAge.

Entre os diferenciais que este trabalho propôs, destacam-se: (i) a viabilidade de utilização em situações não cobertas pelos demais trabalhos; (ii) a integração do conhecimento prático com o conhecimento teórico facilmente acessível dentro do próprio sistema; e (iii) a busca pelo realismo da simulação ao fazer com que o jogador se depare com situações e conceitos usuais para uma equipe de desenvolvimento, mas não abordados pelos outros jogos encontrados.

Após a concepção da proposta do jogo, um protótipo foi implementado, contemplando todos os itens abordados na proposta. Finalizado o desenvolvimento, o protótipo foi avaliado por 39 profissionais de tecnologia da informação, que proveram feedbacks bastante positivos em relação à sua experiência com o ScrumAge.

A partir dos comentários efetuados pelos avaliadores, foram levantadas diversas possibilidades de melhoria para serem empreendidas por trabalhos futuros. Entre as melhorias de usabilidade e interface, encontram-se: (i) a criação de alguma forma de treinamento ou tutorial, possibilitando a redução de textos introdutórios; (ii) ajustes na forma de exibição da troca entre papéis, para se tornar menos intrusiva; (iii) exibir a pontuação parcial do jogador ao final de cada Sprint; entre outras.

Já entre as melhorias sugeridas para aumentarem o realismo e a complexidade do jogo, estão: (i) possibilitar que o jogador escolha alocar manualmente as tarefas aos desenvolvedores; (ii) adicionar habilidades especiais para diferenciar desenvolvedores de

mesma senioridade; (iii) adicionar a opção de criar débito técnico, para que tarefas de baixa qualidade possam ser melhoradas posteriormente, não apenas no dia em que foram finalizadas; dentre outras de menor significância.

Além das sugestões recebidas, espera-se que outros aspectos também possam ser evoluídos em futuras versões do ScrumAge. Entre os aspectos relativos ao sistema em geral, estão: (i) criar tipo de usuário administrador, com acesso a mais funcionalidades no sistema; (ii) disponibilizar telas de configuração para realizar as parametrizações, que nesta primeira versão precisam ser feitas diretamente no banco de dados, ou no código-fonte do sistema, como por exemplo a criação de projetos, ou de artigos na Wiki; e (iii) traduzir os textos do jogo para outros idiomas e permitir a seleção de idioma pela interface do sistema; (iv) permitir que o jogador visualize mais detalhes do histórico de suas partidas; entre outros.

Em relação ao aperfeiçoamento do realismo do jogo para futuras versões, pretende-se: (i) abordar mais claramente os artefatos do Scrum: backlog da Sprint, backlog do produto e incremento; (ii) adicionar mais inteligência e individualidade aos desenvolvedores, considerando características pessoais, como produtividade, satisfação, eventuais contratempos etc.; (iii) inserir eventos aleatórios que podem influenciar o desenvolvimento de tarefas, como mudança de prioridade pelo cliente, ou problemas internos à equipe; (iv) considerar a influência da senioridade também no tempo de desenvolvimento de tarefas; (v) criar outros tipos de tarefas para existirem em conjunto com as demandas do cliente, como por exemplo, tarefas de melhoria de processo, que poderiam reduzir a probabilidade de problemas, ou aumentar a produtividade.

Uma das limitações deste trabalho foi a revisão bibliográfica realizada, que poderia ter sido mais abrangente ao utilizar outras fontes de dados. Já dentre as limitações da avaliação deste trabalho, destaca-se a falta de testes de conhecimento para evidenciar a influência do jogo no aprendizado, mesmo que 98% dos participantes tenham concordado que o ScrumAge tem potencial de auxiliar o aprendizado da metodologia ágil Scrum. Também é importante mencionar que cerca de metade dos avaliadores eram funcionários de uma única empresa, muitos com experiências parecidas com o Scrum. Objetivando superar essas limitações, espera-se realizar uma nova avaliação com alunos e professores da disciplina de engenharia de software.

Por fim, é possível concluir que este trabalho contribui para a evolução dos serious games para o ensino do Scrum, apresentando uma abordagem diferenciada, voltada para o aumento da disponibilidade do sistema, do realismo e do acesso a informações teóricas durante a prática simulada.

REFERÊNCIAS

- ANGULAR. Disponível em: <http://angular.io>. Acesso em: 06 de agosto de 2022.
- ARAGÃO, Paulo André Pimenta. Scrum XPerience: um protótipo de jogo voltado para o ensino de metodologias ágeis. 2022.
- BANGOR, Aaron; KORTUM, Philip T.; MILLER, James T. An empirical evaluation of the system usability scale. **Intl. Journal of Human-Computer Interaction**, v. 24, n. 6, p. 574-594, 2008.
- BATTISTELLA, Paulo Eduardo; DE CAMARGO, André Stangarlin; VON WANGENHEIM, Christiane Gresse. Scrum-scape: Jogo educacional de role-playing game (rpg) para ensinar scrum. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. 2016. p. 330.
- BECK, K., et al. **The Agile Manifesto**. Agile Alliance, 2001. Disponível em: <http://agilemanifesto.org/>. Acesso em: 30 de julho de 2022.
- BERNERS-LEE, Tim et al. The world-wide web. **Communications of the ACM**, v. 37, n. 8, p. 76-82, 1994a.
- BERNERS-LEE, Tim; MASINTER, Larry; MCCAHERILL, Mark. **Uniform resource locators (URL)**. 1994b.
- BOOTSTRAP. Disponível em: <https://getbootstrap.com/>. Acesso em: 06 de agosto de 2022.
- BROOKE, John et al. SUS-A quick and dirty usability scale. **Usability evaluation in industry**, v. 189, n. 194, p. 4-7, 1996. Corti, Kevin. Games-based Learning; a serious business application. Informe de PixelLearning, v. 34, n. 6, p. 1-20, 2006.
- CORONA, Erika; PANI, Filippo Eros. A review of lean-kanban approaches in the software development. **WSEAS transactions on information science and applications**, v. 10, n. 1, p. 1-13, 2013.
- DE CASTRO, Ronney Moreira et al. AGILITY SCRUM-Um Jogo para Ensino da Metodologia SCRUM. In: **Anais do XXV Workshop sobre Educação em Computação**. SBC, 2017. De Souza, Adler Diniz et al. SCRUMI: a board serious virtual game for teaching the SCRUM framework. In: 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C). IEEE, p. 319-321, 2017.
- DIGITAL.AI. **15th Annual State of Agile Report**, 2021. Disponível em <https://digital.ai/resource-center/analyst-reports/state-of-agile-report>. Acesso em: 30 de julho de 2022.

ECMA INTERNATIONAL. **ECMA-262 - ECMAScript® 2022 language specification**, 2022. Disponível em <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>. Acesso em: 06 de agosto de 2022.

ENGE, Eric. **Mobile vs. Desktop Usage in 2020**, 2021. Disponível em <https://www.perficient.com/insights/research-hub/mobile-vs-desktop-usage>. Acesso em: 25 de agosto de 2022.

EXPRESS. Disponível em: <http://expressjs.com/>. Acesso em: 06 de agosto de 2022.

FERNANDES, João M.; SOUSA, Sónia M. Playscrum-a card game to learn the scrum agile method. In: **2010 Second International Conference on Games and Virtual Worlds for Serious Applications**. IEEE, 2010. p. 52-59. FIELDING, Roy et al. RFC2616: Hypertext Transfer Protocol--HTTP/1.1. 1999.

FIELDING, Roy Thomas. REST: architectural styles and the design of network-based software architectures. **Doctoral dissertation, University of California**, 2000.

GARCIA, Ivan et al. Experiences of using a game for improving learning in software requirements elicitation. **Computer Applications in Engineering Education**, v. 27, n. 1, p. 249-265, 2019.

GOOGLE SCHOLAR. Disponível em <http://scholar.google.com>. Acesso em 23 de agosto de 2022.

JADHAV, Madhuri A.; SAWANT, Balkrishna R.; DESHMUKH, Anushree. Single page application using angularjs. **International Journal of Computer Science and Information Technologies**, v. 6, n. 3, p. 2876-2879, 2015. LEE, Wee Leong. SCRUM-X: An interactive and experiential learning platform for teaching scrum, 2016.

JWT. Disponível em: <https://jwt.io/>. Acesso em: 06 de agosto de 2022.

MAHNIČ, Viljan; HOVELJA, Tomaž. On using planning poker for estimating user stories. **Journal of Systems and Software**, v. 85, n. 9, p. 2086-2095, 2012.

MARINHO, Leonardo Lima et al. Scrumie: Scrum Teaching Agent Oriented Game. **Revista de Informática Teórica e Aplicada**, v. 27, n. 2, p. 140-152, 2020. MARTIN, Robert C. Clean code: a handbook of agile software craftsmanship. Pearson Education, 2009

MARTINS, Ana Isabel et al. European Portuguese validation of the system usability scale (SUS). **Procedia computer science**, v. 67, p. 293-300, 2015.

MICROSOFT. Disponível em <https://www.microsoft.com/>. Acesso em: 06 de agosto de 2022.

MIGUEL, José P.; MAURICIO, David; RODRÍGUEZ, Glen. A review of software quality models for the evaluation of software products. **arXiv preprint arXiv:1412.2977**, 2014.

MYSQL. Disponível em: <https://www.mysql.com/>. Acesso em: 06 de agosto de 2022.

NESTJS. Disponível em <https://nestjs.com/>. Acesso em: 06 de agosto de 2022.

NODE.JS. Disponível em: <http://nodejs.org/en/about/>. Acesso em: 06 de agosto de 2022.

OLUWATOSIN, Haroon Shakirat. Client-server model. **IOSR Journal of Computer Engineering**, v. 16, n. 1, p. 67-71, 2014. Richards, Mark. Software architecture patterns. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Incorporated, 2015.

OXFORD. **Application Programming Interface**. Disponível em: <https://www.oxfordlearnersdictionaries.com/definition/english/application-programming-interface>. Acesso em: 06 de agosto de 2022.

RODRIGUEZ, Guillermo; SORIA, Álvaro; CAMPO, Marcelo. Virtual Scrum: A teaching aid to introduce undergraduate software engineering students to scrum. **Computer Applications in Engineering Education**, v. 23, n. 1, p. 147-156, 2015.

RODRÍGUEZ, Guillermo Horacio et al. ScrumGame: A Serious Game to Initiate Software Trainees in Scrum. In: **XXVI Congreso Argentino de Ciencias de la Computación (CACIC)(Modalidad virtual, 5 al 9 de octubre de 2020)**. 2020.

RODRÍGUEZ, Guillermo; GONZÁLEZ-CAINO, Pablo C.; RESETT, Santiago. Serious games for teaching agile methods: A review of multivocal literature. **Computer Applications in Engineering Education**, v. 29, n. 6, p. 1931-1949, 2021.

SANTOS, Djenane Cristina Silveira; DE SOUZA, Adler Diniz; PAULA, Melise Veiga. Comparing different serious games strategies for teaching the SCRUM Framework. **RENOTE**, v. 19, n. 1, p. 227-236, 2021.

SCHWABER, K., & Sutherland, J. **The Scrum Guide – The Definitive Guide to Scrum: The Rules of the Game**, 2020. Disponível em <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>. Acesso em: 30 de julho de 2022

SCRUM.ORG. **The Scrum Framework Poster**, 2020. Disponível em: <https://www.scrum.org/resources/scrum-framework-poster>. Acesso em: 30 de julho de 2022.

STATISTA. **Ranking of the most popular database management systems worldwide**, 2022. Disponível em: <https://www.statista.com/statistics/809750/worldwide-popularity-ranking-database-management-systems/>. Acesso em: 06 de agosto de 2022.

SUSI, Tarja; JOHANNESSON, Mikael; BACKLUND, Per. Serious games: An overview. 2007.

THEODORSON, George A.; THEODORSON, Achilles G. A modern dictionary of sociology. 1969.

TYPESCRIPT. Disponível em: <https://www.typescriptlang.org>. Acesso em: 06 de agosto de 2022.

VINUTO, Juliana. A amostragem em bola de neve na pesquisa qualitativa: um debate em aberto. **Temáticas**, v. 22, n. 44, p. 203-220, 2014.

VON WANGENHEIM, Christiane Gresse; SAVI, Rafael; BORGATTO, Adriano Ferreti. SCRUMIA—An educational game for teaching SCRUM in computing courses. **Journal of Systems and Software**, v. 86, n. 10, p. 2675-2687, 2013.

WIEGERS, Karl; BEATTY, Joy. **Software requirements**. Pearson Education, 2013.

ZYDA, Michael. From visual simulation to virtual reality to games. **Computer**, v. 38, n. 9, p. 25-32, 2005.

APÊNDICE A – Sprints e tarefas do projeto simulado no protótipo do ScrumAge

Sprints

Sprint	Objetivo	Mensagem do Cliente
1	Implementar o módulo de cadastro e acesso de usuários	<p>Antes de mais nada, gostaríamos de ter a tela inicial do projeto pronta, com o menu de navegação, para já podermos ter uma noção da aparência do sistema. Sendo tão importante quanto isso, também desejamos ter logo o cadastro de usuários, que deverá enviar um e-mail para que o usuário ative a sua conta. Além disso, é imprescindível que desde já seja necessário que todos os usuários se autentiquem para utilizarem o sistema.</p> <p>Adicionalmente, o perfil de cada usuário deverá ser visualizado através de uma tela própria. Também é interessante termos permissões distintas para cada tipo de utilizador do sistema: administradores, vendedores e caixas, o que será importante conforme os próximos módulos do sistema forem desenvolvidos. Usuários administradores deverão ter acesso a uma tela de busca de usuários, onde poderão também alterá-los ou excluí-los.</p>
2	Implementar módulos para gestão de estoque de produtos	<p>Agora queremos avançar para o módulo que, para nós, é o mais importante do sistema: o de gestão de produtos. Por isso, precisamos que sua equipe dê uma atenção especial a esta parte. Através dele deve ser possível buscar, cadastrar, alterar e excluir produtos.</p> <p>Além disso, nos pouparia bastante tempo se o sistema permitisse cadastrear um produto a partir de uma cópia de outro já cadastrado. Outra funcionalidade útil seria enviar um alerta para usuários administradores toda vez que o estoque de um produto estiver baixo.</p> <p>Por fim (por enquanto, pois ainda há muito a ser feito), gostaríamos de ter em mãos a gestão dos dados dos nossos fornecedores, sendo também possível buscar, cadastrar, alterar e excluir suas informações.</p>
3	Implementar módulo para a gestão de lojas	<p>Como você sabe, hoje temos apenas 3 unidades funcionando na nossa cidade. Porém, como estamos planejando grandes investimentos para expandir rapidamente essa rede, inclusive para outras cidades, precisamos da sua ajuda. Para que possamos gerenciar nossas lojas com maior eficácia, desejamos agora de um módulo para a gestão delas. Assim como alguns dos módulos já implementados, precisamos ter acesso a uma gestão completa, incluindo busca, cadastro, alteração e exclusão de lojas.</p> <p>Uma vez finalizados os módulos de gestão de usuários, estoque de produtos, fornecedores e lojas, podemos seguir para o módulo de vendas, aquele que o operador de caixa usará para registrar nossas vendas. Como não queremos ter problemas com a Receita Federal, precisamos realizar a emissão de uma nota fiscal para cada venda.</p> <p>Para aumentar a segurança e a agilidade no recebimento dos pagamentos, nosso sistema deve se integrar com o sistema de pagamento Pix e com as nossas maquininhas de cartão. A última funcionalidade que imaginamos para este módulo, por enquanto, é o registro da devolução de um produto (embora saibamos que ela será pouco utilizada, pois nossos produtos são de extrema qualidade).</p> <p>Também pensamos em uma funcionalidade adicional para um melhor monitoramento do sistema: ao acessar o perfil de um usuário, administradores deverão conseguir ver todo o histórico de ações daquela</p>

		pessoa.
4	Implementar módulos de dashboards e de gestão de clientes	<p>Agora estamos na reta final! Para fecharmos essa primeira versão do sistema com chave de ouro, precisamos ter uma forma de visualizarmos rapidamente diversas informações importantes para um melhor gerenciamento dos nossos recursos. Afinal, como seremos a maior empresa do ramo no país (e talvez no mundo) se não tivermos como analisar dados e tomar decisões sobre o nosso negócio?</p> <p>Até o momento, pensamos em um módulo contendo quatro dashboards estatísticos: fluxo de caixa, vendas por vendedor, vendas por produto e estoque. Todos eles são essenciais para o nosso crescimento. Então, por favor, não nos decepcionem!</p> <p>Por último, queremos um módulo para gerirmos os dados dos nossos clientes, visando proporcionar a eles uma experiência que irá nos diferenciar da concorrência. Neste módulo, inicialmente devemos ter acesso a busca, cadastro, alteração e exclusão de clientes. Em seguida, vamos utilizar esses dados para começarmos a mostrar a todos porque eles devem escolher nossa marca: queremos implementar um programa de fidelidade e realizar o envio de e-mails contendo promoções exclusivas para os clientes cadastrados, ideias quase nunca exploradas no nosso setor.</p>

Lista de Tarefas

Id	Título	Melhoria Possível	Problema Possível
1	Possibilitar exclusão de usuários	Adicionar uma pergunta de confirmação, para evitar exclusões indevidas.	Ao excluir um usuário, o sistema diz que o usuário foi excluído, mas ele continua visível para busca.
2	Criar menu de navegação	Modificar o estilo do menu para que os itens fiquem mais legíveis, melhorando a acessibilidade.	Alguns itens do menu não estão sendo exibidos em dispositivos móveis.
3	Implementar controle de permissões por perfil: administrador, vendedor, caixa	Possibilitar que administradores troquem o perfil de outros usuários.	Vendedores estão conseguindo ver informações de fornecedores que não deveriam estar disponíveis para seu perfil.
4	Enviar e-mail para ativação de conta	Melhorar formatação da mensagem de e-mail de ativação de conta.	Alguns usuários não estão recebendo o e-mail de ativação de conta.
5	Criar tela de busca de usuários	Tornar a busca por usuários mais rápida, com menor tempo de resposta.	A busca de usuários com acento no nome não está funcionando.
6	Criar tela de cadastro de usuários	Adicionar campos faltantes no formulário de cadastro de usuários.	Um erro genérico está ocorrendo ao tentar cadastrar alguns usuários.
7	Criar tela de perfil de usuário	Aumentar tamanho das fontes dos campos do perfil de usuário.	A tela de perfil de usuário não está sendo corretamente exibida em dispositivos móveis, pois alguns elementos ficam sobrepostos.
8	Possibilitar alteração de usuários	Impedir que o CPF de um usuário seja alterado.	Um erro genérico está ocorrendo ao tentar alterar qualquer usuário.

9	Criar autenticação de usuários	Facilitar o acesso ao botão de logout do sistema.	A senha digitada está sendo aceita tanto em letras maiúsculas como minúsculas, o que é uma falha de segurança.
10	Criar tela inicial	Tornar o carregamento da tela inicial mais rápido.	A tela inicial não está sendo corretamente exibida em dispositivos móveis, pois alguns elementos ficam sobrepostos.
11	Criar tela de cadastro de fornecedor	Remover campos desnecessários no formulário de cadastro de fornecedor.	A validação do campo CNPJ está incorreta.
12	Possibilitar exclusão de produtos	Adicionar uma pergunta de confirmação, para evitar exclusões indevidas.	Alguns produtos excluídos estão voltando a aparecer no sistema depois de algum tempo.
13	Criar tela de busca de produtos	Permitir buscar produto por fornecedor.	Alguns produtos não estão sendo exibidos na busca.
14	Criar alerta para estoque baixo de produto	Melhorar visibilidade do alerta de estoque baixo.	O alerta não está funcionando com os valores definidos.
15	Possibilitar alteração de produtos	Posicionar em local mais visível o botão de "Salvar alterações".	A validação dos campos não funciona ao tentar alterar os dados de alguns produtos, impossibilitando a atualização.
16	Criar tela de cadastro de produtos	Impedir o cadastro de produtos com o mesmo nome.	Um erro de "produto já existente" é exibido no cadastro de alguns produtos novos.
17	Criar tela de busca por fornecedor	Permitir buscar fornecedor também por CNPJ.	Fornecedores recém cadastrados não estão sendo exibidos na busca.
18	Possibilitar alteração de dados do fornecedor	Possibilitar alterar o CNPJ de fornecedores.	Está sendo possível salvar um fornecedor com o nome em branco ao realizar uma alteração.
19	Possibilitar exclusão de fornecedor	Melhorar a performance da busca de fornecedores, que está demorando um pouco demais.	A exclusão de fornecedores não está exibindo a pergunta de confirmação se o usuário deseja realmente excluir.
20	Permitir cadastrar produto a partir de uma cópia de um produto já cadastrado	Melhorar posicionamento do botão de cópia de produto.	Alguns campos não estão sendo copiados para o novo produto.
21	Realizar emissão de nota fiscal	Aumentar tamanho da fonte dos campos da nota fiscal.	O sistema trava quando o sistema de emissão de nota fiscal da Secretaria da Fazenda está inacessível.
22	Registrar venda	Destacar promoções aplicadas na tela de registro de vendas.	Produtos adicionados e depois removidos da venda continuam sendo contabilizados no valor total da venda.
23	Registrar devolução de produto	Adicionar atalho para esta funcionalidade no menu de navegação.	Produtos devolvidos não estão sendo contabilizados na quantidade em estoque.
24	Criar componente para visualização do histórico de ações de um usuário	Melhorar organização visual do histórico de ações de um usuário.	O histórico de ações de um usuário não está trazendo algumas ações.
25	Criar tela de busca de lojas	Melhorar exibição dos campos na busca de lojas, que ficam sobrepostos quando o nome do bairro é grande.	A busca de lojas está trazendo registros repetidos.
26	Realizar integração com sistema de pagamento com cartão	Exibir na tela de registro de vendas o tipo de cartão utilizado (crédito ou débito).	Para alguns cartões o pagamento é aprovado, mas o sistema exibe a mensagem de transação negada.

27	Realizar integração com sistema de pagamento Pix por QR Code	Aumentar tamanho do QR Code, para facilitar a utilização pelo cliente.	O QR Code do PIX não está sendo exibido em algumas compras, forçando o cliente a utilizar outro método de pagamento.
28	Possibilitar alteração de lojas	Melhorar o posicionamento dos elementos na tela de alteração de lojas.	Um erro genérico está ocorrendo ao tentar alterar os dados de alguma loja.
29	Criar tela de cadastro de lojas	Permitir alterar o código de uma loja.	Não está sendo possível alterar a localização de uma loja, pois ao salvar a localização antiga volta a ser exibida.
30	Permitir exclusão de lojas	Ao excluir uma loja, manter o histórico de registro de vendas daquela loja.	O sistema exibe uma mensagem de erro genérica ao tentar excluir uma loja.
31	Implementar programa de fidelidade	Enviar e-mail para o cliente a cada compra realizada, informando os pontos de fidelidade acumulados com aquela compra e os benefícios disponíveis para resgate.	Alguns clientes não estão tendo suas compras contabilizadas no programa de fidelidade.
32	Enviar e-mails promocionais aos clientes	Confirmar se o usuário tem certeza que quer enviar o e-mail antes do envio ser efetivado.	Nem todos os clientes cadastrados estão recebendo os e-mails promocionais.
33	Criar tela de cadastro de cliente	Enviar e-mail para o cliente confirmando o cadastro e informando os benefícios.	Um erro de "ação não permitida" está ocorrendo quando um vendedor tenta cadastrar um cliente.
34	Criar dashboard com estatísticas de estoque	Melhorar taxa de atualização do gráfico de estoque.	Produtos excluídos ainda estão sendo exibidos no dashboard de estoque.
35	Criar dashboard com estatísticas de vendas por vendedor	Destacar em verde os vendedores com mais vendas e em vermelho os com menos vendas.	Vendedores recém cadastrados não estão aparecendo no dashboard.
36	Criar tela de busca de cliente	Melhorar performance da tela de busca de cliente, que está demorando mais do que deveria.	Não está sendo possível buscar clientes pelo CPF.
37	Possibilitar exclusão de cliente	Solicitar um motivo ao tentar excluir um cliente.	Ao excluir um cliente os registros de vendas para ele também são excluídos, gerando uma inconsistência no histórico de vendas.
38	Criar dashboard com estatísticas de vendas por produtos	Destacar produtos com maior taxa de vendas.	Produtos com poucas vendas não estão sendo contabilizados no dashboard.
39	Criar dashboard com estatísticas de fluxo de caixa	Exibir valor médio de cada venda.	Reembolsos por produtos devolvidos não estão sendo contabilizados no fluxo de caixa.
40	Possibilitar alteração de dados do cliente	Impossibilitar alteração de CPF.	Ao alterar os dados de um cliente, o sistema exibe uma mensagem de erro.

APÊNDICE B – Textos de feedback ao final da partida

Tema	Valor (Desempenho)	Texto
Quantidade de Tarefas Finalizadas	= 40 (Alto)	Parabéns!!! Vocês conseguiram finalizar todas as funcionalidades que havíamos imaginado para o sistema. Isso era o que considerávamos mais importante neste projeto! Muito obrigado!
	>= 36 (Médio)	Algumas das funcionalidades que pedimos ficaram faltando , mas isso não é um problema tão grande, porque todas as principais funções estão prontas. As que restaram não eram essenciais, logo ficarão para uma próxima evolução do sistema.
	< 36 (Baixo)	Várias funcionalidades que pedimos ficaram faltando. Mesmo que a maior parte delas não seja essencial, esperávamos que vocês conseguissem deixar o sistema mais completo.
Qualidade Funcional Média	>= 4 (Alto)	Adoramos a qualidade do que foi entregue! Sua equipe é realmente bastante profissional, pois tiveram muito cuidado com os detalhes, onde deram ótimas soluções para as funcionalidades que solicitamos! Só temos elogios para todos vocês. Parabéns!
	>= 2,5 e < 4 (Médio)	A maior parte das funções do sistema foram bem-feitas, porém algumas deixaram a desejar . Talvez essas que ficaram piores tenham sido feitas pelos membros menos experientes da equipe, mas acreditamos que no geral foi um bom trabalho.
	< 2,5 (Baixo)	Uma parte considerável das funcionalidades do sistema não ficaram da forma que esperávamos . Talvez essas que ficaram piores tenham sido feitas pelos membros menos experientes da sua equipe, mas isso poderia ter sido evitado se houvesse um maior cuidado durante o desenvolvimento.
Orçamento Restante	> 4.000 (Alto)	Vocês conseguiram gerenciar muito bem o orçamento reservado para a contratação do time de desenvolvimento! Ficamos muito felizes em saber que sobraram mais de 10% do orçamento total do projeto, muito obrigado!
	> 0 e <= 4000 (Médio)	Vocês conseguiram utilizar bem o orçamento reservado para a contratação do time de desenvolvimento. Não sobrou muito, mas esperávamos que não sobraria nada. Então agradecemos por não terem gasto tudo! Utilizaremos o valor restante para futuras evoluções do sistema.
	0 (Baixo)	Vocês usaram todo o orçamento reservado para a contratação do time de desenvolvimento. Seria bom se tivessem conseguido poupar um pouco, mas isso não era obrigatório.
Satisfação Final do Cliente	>= 4 (Alto)	Em resumo, ficamos realmente satisfeitos com o resultado final! Mesmo com o prazo apertado, vocês conseguiram fazer um excelente trabalho, superando de longe nossas expectativas! Tenham certeza de que em breve vamos contratar sua empresa para realizar evoluções nesse projeto e também em outros que já temos em mente! Além disso, também vamos recomendar vocês para nossos parceiros, porque não é fácil encontrar empresas de desenvolvimento competentes como a de vocês! Até a próxima!
	>= 2,5 e < 4 (Médio)	Em resumo, o resultado final ficou bom . Sabemos que o prazo era muito apertado e que seria difícil construirmos o sistema que havíamos imaginado neste tempo, mas vocês fizeram um trabalho satisfatório . No futuro entraremos em contato para evoluirmos esse sistema. Até a próxima!

Satisfação Final do Cliente	$< 2,5$ (Baixo)	<p>Em resumo, o resultado final não foi como esperávamos. Sabemos que o prazo era apertado e que seria difícil construirmos o sistema que havíamos imaginado neste tempo, mas imaginávamos conseguir algo pelo menos um pouco melhor do que isto. Sendo assim, agradecemos pelo esforço durante o tempo que tivemos juntos, mas não voltaremos a fechar negócio com sua empresa.</p>
Qualidade Técnica Média	≥ 4 (Alto)	<p>O desenvolvimento deste projeto foi bastante produtivo! Utilizando o Scrum e de diversas boas práticas de desenvolvimento, conseguimos usar e aperfeiçoar nossas habilidades para construir um sistema de qualidade exemplar! Trabalhar com um time tão dedicado e capaz, resolvendo problemas de forma rápida e eficiente, é o sonho de qualquer desenvolvedor! Com um código tão bem escrito quanto esse, a evolução e manutenção do software tende a ser simples, então todos que lidarem com ele poderão nos agradecer no futuro!</p>
	$\geq 2,5$ e < 4 (Médio)	<p>Conseguimos construir um software de qualidade interessante. Olhando para trás, poderíamos ter feito algumas coisas de outra maneira, mas ainda assim o produto final foi um sucesso, considerando o prazo que tínhamos. O time ainda pode se aperfeiçoar mais em alguns aspectos, porém já sabem como fazer um bom trabalho. Quem trabalhar na manutenção deste sistema no futuro pode dizer que o código não é tão bom quanto poderia ser, mas não vai dizer que foi mal feito.</p>
	$< 2,5$ (Baixo)	<p>Construímos um software de qualidade questionável. O prazo era curto e o time não se conhecia, mas mesmo assim acreditamos que poderíamos ter feito um trabalho melhor. Não conseguimos mostrar nossas habilidades da maneira que gostaríamos, o que é um pouco desmotivador. As próximas pessoas que precisarem dar manutenção no sistema talvez fiquem um pouco confusas e nos critiquem, mas agora já está feito. Quem sabe em uma próxima oportunidade, com o time mais entrosado, consigamos desenvolver um projeto com maior qualidade técnica.</p>

APÊNDICE C – Lista de artigos da Wiki do ScrumAge

- Wiki
 - Scrum
 - Teoria do Scrum
 - Valores do Scrum
 - Time Scrum
 - Time de Desenvolvimento
 - Product Owner
 - Scrum Master
 - Eventos Scrum
 - Sprint
 - Planejamento de Sprint
 - Reunião Diária
 - Revisão da Sprint
 - Retrospectiva de Sprint
 - Artefatos do Scrum
 - Backlog do Produto
 - Backlog da Sprint
 - Incremento
 - Conceitos Complementares
 - Manifesto Ágil
 - Metodologias Ágeis
 - Planning Poker
 - Kanban
 - Cliente
 - ScrumAge
 - Qualidade Funcional
 - Qualidade Técnica
 - Satisfação do Cliente
 - Produtividade
 - Correção
 - Melhoria
 - Prioridade
 - Estimativa
 - Senioridade
 - Alocação de Tarefas
 - Decisão para Tarefa
 - Revisar
 - Refatorar
 - Aprofundar Testes
 - Não Fazer Nada

APÊNDICE D – Termo de Consentimento Livre e Esclarecido**TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO**

Você está sendo convidado(a) a participar, como voluntário(a), da pesquisa intitulada “Avaliação de um serious game de simulação para o aprendizado da Metodologia Ágil Scrum”, conduzida por Leonardo Lima Marinho, Vera Maria Benjamim Werneck e Rosa Maria Moreira Esteves Costa, pesquisadores do IME- UERJ. Esta pesquisa tem por objetivo avaliar a qualidade e a usabilidade do serious game ScrumAge, voltado para o aprendizado da metodologia ágil Scrum. Você receberá o link onde poderá acessá-lo e testá-lo, para que em seguida responda perguntas relacionadas à sua experiência com o jogo.

Você foi selecionado(a) por ser profissional ou estudante da área de informática. Sua participação não é obrigatória. A qualquer momento, você poderá desistir de participar e retirar seu consentimento, abandonando o preenchimento do questionário. Sua recusa, desistência, ou retirada de consentimento não acarretará prejuízo.

A pesquisa apresenta riscos mínimos para os participantes, com possibilidade de constrangimento por finalizar o jogo com uma baixa pontuação ou em responder alguma das questões. Para contornar esse problema, o jogador pode abandonar a qualquer momento e assim seu resultado não será guardado. No entanto, o benefício viria pela disseminação do jogo, que busca tornar mais acessível e agradável o aprendizado dessa metodologia ágil tão utilizada na atualidade.

Sua participação não é remunerada e nem implicará em gastos para você. Caso se sinta prejudicado, você tem direito de ser indenizado por danos decorrentes da pesquisa, nos termos da lei e de ser ressarcido de despesas decorrentes da participação na pesquisa, de acordo com os itens VI e VII do artigo 9º, do Cap. III, da Resolução 510/2016.

Os dados obtidos por meio desta pesquisa serão confidenciais e não serão divulgados em nível individual, visando assegurar o sigilo de sua participação. Neste caso, os pesquisadores se comprometem a tornar públicos nos meios acadêmicos e científicos os resultados obtidos de forma consolidada, sem qualquer identificação de indivíduos, ou instituições participantes.

Caso você concorde em participar desta pesquisa, assinale ao final deste documento, que possui duas vias, sendo uma delas sua, que será enviada como recibo de sua resposta, e a outra, do pesquisador responsável/coordenador da pesquisa. Seguem os telefones e o endereço institucional do pesquisador responsável e do Comitê de Ética em Pesquisa – CEP, onde você poderá tirar suas dúvidas sobre o projeto e sua participação nele, agora ou a qualquer momento.

Contatos dos pesquisadores responsáveis: Leonardo Lima Marinho, telefone (21) 96835-1857, e-mail: leonardomarinho_10@hotmail.com; Vera M. B. Werneck, e-mail: vera@ime.uerj.br; Rosa Maria E. M. da Costa, telefone (21) 98125-6905, e-mail: rcosta@ime.uerj.br. Caso você tenha dificuldade em entrar em contato com os pesquisadores responsáveis, comunique o fato à Comissão de Ética em Pesquisa da UERJ: Rua São Francisco Xavier, 524, sala 3018, bloco E, 3º andar, - Maracanã - Rio de Janeiro, RJ, e-mail: etica@uerj.br - Telefone: (021) 2334-2180. O CEP COEP é responsável por garantir a proteção dos participantes de pesquisa e funciona às segundas, quartas e sextas-feiras, de 10h às 12h e 14h às 16h.

Declaro que entendi os objetivos, riscos e benefícios de minha participação na pesquisa, e que concordo em participar.

Assinatura do (a) participante: _____ (Obs.: neste caso, haverá um quadradinho de aceitação de participação, já que o instrumento estará disponível via web).

Assinatura do pesquisador: _____

APÊNDICE E – Questionário de Caracterização do Participante

Questionário de Caracterização do Participante

Prezado(a) participante,

Você foi selecionado para participar da avaliação de um jogo para o aprendizado da metodologia ágil Scrum. Este convite foi realizado por você ser profissional ou estudante da área de informática.

Preencha o formulário abaixo com os dados sobre sua experiência acadêmica e profissional e, após finalizar ao menos uma partida no ScrumAge, preencha os questionários seguintes conforme a sua percepção do jogo.

1. Formação acadêmica

- Doutorado concluído
 Doutorado em andamento
 Mestrado concluído
 Mestrado em andamento
 Graduação concluída
 Graduação em andamento

Instituição acadêmica: _____

Ano de ingresso: _____ Ano de conclusão / Ano previsto de conclusão: _____

2. Formação geral

a) Qual é sua experiência com desenvolvimento ágil? (marque todos os itens que se aplicam)

- Já li material sobre desenvolvimento ágil.
 Já participei de um curso sobre desenvolvimento ágil.
 Já li material sobre SCRUM
 Já participei de um curso sobre SCRUM
 Já desenvolvi software com método ágil
 Já desenvolvi software com SCRUM
 Nunca desenvolvi software com método ágil
 Nunca desenvolvi software com SCRUM

Por favor, detalhe sua resposta. Inclua o número de semestres ou número de anos de experiência relevante em desenvolvimento de software. (por exemplo, “Eu trabalhei por 2 anos como programador de software na indústria”)

b) Por favor, indique o grau de sua experiência nas áreas a seguir, com base na escala abaixo:

- 1 = Nenhum
 2 = Estudei em aula ou em livro
 3 = Pratiquei em projetos em sala de aula
 4 = Utilizei em projetos pessoais
 5 = Utilizei em projetos na indústria

Área de conhecimento	Grau de experiência				
Engenharia de software	1	2	3	4	5
Desenvolvimento de software	1	2	3	4	5
Desenvolvimento de software com método ágil	1	2	3	4	5
Desenvolvimento de software com SCRUM	1	2	3	4	5

Desde já, agradecemos a sua colaboração.

Leonardo Lima Marinho
 Vera Maria Benjamin Werneck
 Rosa Maria Moreira Esteves da Costa

APÊNDICE F – Questionário de Avaliação do ScrumAge

Questionário de Avaliação do ScrumAge

Após finalizar uma partida no jogo, responda os questionários de conteúdo, interface e de usabilidade, de acordo com a escala abaixo. Em seguida, sinta-se livre para comentar sobre percepções positivas e negativas sobre o ScrumAge.

Pontuação	Respostas
1	Discordo totalmente
2	Discordo
3	Sem opinião
4	Concordo
5	Concordo totalmente

Questionário de Conteúdo

Afirmação	Nota
1. O jogo tem o potencial de auxiliar o aprendizado da metodologia ágil Scrum.	1 2 3 4 5
2. É uma ferramenta relevante para profissionais ou estudantes da área de informática.	1 2 3 4 5
3. Os conceitos do Scrum estão sendo corretamente apresentados no jogo.	1 2 3 4 5
4. Os conceitos complementares (Kanban, planning poker etc) estão sendo corretamente apresentados no jogo.	1 2 3 4 5
5. Os conceitos específicos do ScrumAge (qualidades funcional e técnica, produtividade, satisfação do cliente etc) representam bem conceitos importantes para o desenvolvimento de softwares na vida real.	1 2 3 4 5
6. O conteúdo da Wiki está correto.	1 2 3 4 5
7. O conteúdo do projeto sendo desenvolvido é realista.	1 2 3 4 5
8. O jogo simula bem o uso do Scrum por uma equipe no desenvolvimento de um software.	1 2 3 4 5
9. O jogo proporciona diversão ao jogador.	1 2 3 4 5
10. O jogo é desafiador.	1 2 3 4 5
11. A funcionalidade de classificação de jogadores torna o jogo mais interessante.	1 2 3 4 5

Questionário de Interface

Afirmação	Nota
1. A aparência geral do jogo é agradável.	1 2 3 4 5
2. A funcionalidade de Wiki integrada é útil.	1 2 3 4 5
3. A linguagem utilizada pelo jogo é agradável.	1 2 3 4 5
4. Durante a partida, a visualização do papel atual do jogador no Scrum é clara.	1 2 3 4 5

Questionário de Usabilidade

Afirmação	Nota
1. Eu acho que gostaria de usar esse sistema com frequência.	1 2 3 4 5
2. Eu acho o sistema desnecessariamente complexo.	1 2 3 4 5
3. Eu achei o sistema fácil de usar.	1 2 3 4 5
4. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.	1 2 3 4 5
5. Eu acho que as várias funções do sistema estão muito bem integradas.	1 2 3 4 5
6. Eu acho que o sistema apresenta muita inconsistência.	1 2 3 4 5
7. Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.	1 2 3 4 5
8. Eu achei o sistema muito complicado de utilizar.	1 2 3 4 5
9. Eu me senti confiante ao usar o sistema.	1 2 3 4 5
10. Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.	1 2 3 4 5

Comentários

Do que você mais gostou no jogo ScrumAge?

Do que você não gostou (ou o que você mudaria) no jogo ScrumAge?