



# **CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS**

*Recredenciado pela Portaria Ministerial nº 1.162, de 13/10/16, D.O.U nº 198, de 14/10/2016*  
ASSOCIAÇÃO EDUCACIONAL LUTERANA DO BRASIL

Raphael Araújo Bentes

JOGO MOBILE EDUCATIVO PARA AUXILIAR NA PRÁTICA E NO ESTUDO DA  
METODOLOGIA DE DESENVOLVIMENTO ÁGIL SCRUM

Palmas – TO

2021

Raphael Araújo Bentes

JOGO MOBILE EDUCATIVO PARA AUXILIAR NA PRÁTICA E NO ESTUDO DA  
METODOLOGIA DE DESENVOLVIMENTO ÁGIL SCRUM

Trabalho de Conclusão de Curso (TCC) II elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Ciência da Computação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Profa. Esp. Fernanda Pereira Gomes.

Palmas – TO

2021

Raphael Araújo Bentes

JOGO MOBILE EDUCATIVO PARA AUXILIAR NA PRÁTICA E NO ESTUDO DA  
METODOLOGIA DE DESENVOLVIMENTO ÁGIL SCRUM

Trabalho de Conclusão de Curso (TCC) II  
elaborado e apresentado como requisito parcial  
para obtenção do título de bacharel em Ciência  
da Computação pelo Centro Universitário  
Luterano de Palmas (CEULP/ULBRA).

Orientador: Profa. Esp. Fernanda Pereira  
Gomes.

Aprovado em: \_\_\_\_/\_\_\_\_/\_\_\_\_

BANCA EXAMINADORA

---

Profa. Esp. Fernanda Pereira Gomes.

Orientadora

Centro Universitário Luterano de Palmas – CEULP

---

Prof. Me. Fabiano Fagundes

Centro Universitário Luterano de Palmas – CEULP

---

Prof. Esp. Fábio Castro Araújo

Centro Universitário Luterano de Palmas – CEULP

Palmas – TO

2021

## RESUMO

BENTES, Raphael Araújo. **Jogo Mobile Educativo para auxiliar na prática e no estudo da Metodologia de Desenvolvimento Ágil Scrum**. 2021. 33 f. Trabalho de Conclusão de Curso (Graduação) – Curso de Ciência da Computação, Centro Universitário Luterano de Palmas, Palmas/TO, 2021<sup>1</sup>

A Metodologia de Desenvolvimento Ágil Scrum vem ganhando cada vez mais popularidade na área da computação, devido ao Scrum ser um método adaptável e de fácil implementação, sendo geralmente ensinada para alunos dos cursos da área de Computação ou em ambientes empresariais. Deste modo, há uma crescente demanda por profissionais capacitados nesta metodologia e, apesar de existirem diversos métodos para o estudo do Scrum, a maioria consiste em atividades com pouca prática, como leitura de livros e apostilas, o que acaba dificultando o entendimento rápido dos conceitos e aplicabilidades. Com isso, este trabalho apresenta o Knight Scrum um jogo mobile desenvolvido para auxiliar na prática da metodologia de Desenvolvimento Ágil Scrum, que de forma efetiva auxilie no aprendizado dos conceitos do Scrum, como os papéis, eventos e artefatos, bem como na prática da metodologia. Portanto este trabalho foi desenvolvido utilizando como metodologia um dos elementos do processo ENgAGED (*EducatioNAl GamEs Development*) desenvolvida por Battistella (2016) e elementos da adaptação Scrum4Games de Laubisch e Clua (2010).

**PALAVRAS-CHAVE:** Scrum, Jogo Educacional, Aprendizado, Metodologia de Desenvolvimento.

## **SUMÁRIO**

<b>1 INTRODUÇÃO</b>	<b>6</b>
<b>2 REFERENCIAL TEÓRICO</b>	<b>7</b>
2.1 METODOLOGIA DE DESENVOLVIMENTO ÁGIL SCRUM	7
2.1.1 O Time Scrum	8
2.1.2 Eventos e Artefatos Scrum	10
2.2 JOGOS EM AUXÍLIO DO APRENDIZADO	13
2.3 GÊNEROS DE JOGOS	145
2.4 TRABALHOS RELACIONADOS	167
2.4.1 Scrum'ed	168
2.4.2 SCRUM-Scape	189
2.5 PROCESSOS E ADAPTAÇÕES DE METODOLOGIAS PARA DESENVOLVIMENTO DE JOGOS	212
2.5.1 ENgAGED	223
2.5.2 Scrum4Games	256
<b>3 METODOLOGIA</b>	<b>268</b>
<b>4 RESULTADOS</b>	<b>31</b>
<b>5 CONSIDERAÇÕES FINAIS</b>	<b>67</b>
<b>5 REFERENCIAL</b>	<b>70</b>

## 1 INTRODUÇÃO

O Scrum é uma Metodologia de Desenvolvimento Ágil que desde 1990 ganha cada vez mais espaço dentre as empresas de desenvolvimento de software e em disciplinas de cursos da área da computação das instituições de ensino. Com isso, projetos tecnológicos como o SCRUM-Scape (CAMARGO, 2013) e Scrum'ed (SCHNEIDER, 2015), acabam sendo incluídos no processo de ensino, pois auxiliam no processo de aprendizado do Scrum.

Um dos temas de grande impacto na educação é a inclusão das inovações tecnológicas no processo de ensino. A Base Nacional Comum Curricular – BNCC apresenta 10 (dez) competências gerais da educação básica, dentre elas, 4 (quatro) instigam a utilização de tecnologia no processo de educação seja para o auxílio no aprendizado, para expressar e partilhar informações, produzir conhecimento, resolução de problemas entre outros fins, (BRASIL, 2018).

Em 2020 foi realizada a primeira edição da pesquisa Avaliação das Competências Digitais dos Docentes do Ensino Superior Brasileiro pela MetaRed Brasil, uma rede ibero-americana que no Brasil congrega docentes e profissionais de TI de Universidades. Segundo a MetaRed Brasil (2020), mais de 90% dos docentes usam recursos TIC no ensino e têm a percepção que 2/3 de seus estudantes também os usam nos seus processos de aprendizagem.

A utilização de aparelhos eletrônicos como *tablets* ou *smartphones* para o auxílio em algumas atividades escolares já é realidade, a pesquisa TIC Educação 2016, do Centro de Estudos sobre as Tecnologias da Informação da Comunicação (CETIC), diz que 52% das escolas utilizam o celular para realizar atividades com os alunos. De fato, a tecnologia está bastante presente na sociedade, o que se deve atentar em relação ao ambiente escolar é em como utilizar esses aparelhos de forma produtiva sem perder o foco no aprendizado.

A BNCC indica que a interação com a tecnologia é uma das fontes que estimulam a curiosidade (BRASIL, 2018). Desse modo, percebe-se que se utilizada de maneira correta, a tecnologia pode atuar significativamente no desenvolvimento e aprendizagem dos alunos. Veen e Vrakking (2009) afirmam que os jogos digitais também promovem o desenvolvimento de várias habilidades metacognitivas na aprendizagem, uma vez que levam ao desenvolvimento de estratégias e soluções para as dificuldades. Segundo Frosi (2010) “os jogos devem possuir certas características para que não percam sua identidade enquanto “game”, preservando, dessa forma, a atratividade e o interesse pelo jogo”.

Nesse sentido, com o intuito de inovar e facilitar o aprendizado dos conceitos Scrum e auxiliar na prática desses conceitos, o problema que o presente trabalho propõe resolver é: de que forma um jogo mobile pode facilitar o entendimento dos conceitos da Metodologia de

Desenvolvimento ágil Scrum e auxiliar na sua prática? Acredita-se que um jogo mobile, através de seus elementos gráficos e de seus mecanismos e dinâmicas, pode ser capaz de despertar a curiosidade do usuário/jogador proporcionando motivação ao estudo de conceitos e entendimento da prática da Metodologia de Desenvolvimento ágil Scrum.

Com base nisso, o objetivo do trabalho origina-se em desenvolver um jogo mobile para auxiliar no estudo dos conceitos e na prática da Metodologia de Desenvolvimento ágil Scrum, sendo assim para auxiliar no processo de desenvolvimento, foi definido alguns objetivos específicos como: definição dos aspectos do jogo tais como níveis, cenários, personagens e história; definir e modelar cenários do jogo de acordo com os conceitos do Scrum e definir gênero de jogos para o desenvolvimento do jogo.

A Metodologia de Desenvolvimento Ágil Scrum está presente em disciplinas das grades curriculares dos cursos de computação e, também, é utilizada por empresas de desenvolvimento de softwares por ser uma ótima ferramenta para gestão de projetos complexos. Contudo, esta metodologia possui vários papéis e cerimônias que acabam dificultando o aprendizado de todos os seus conceitos e dificultando a prática de seus recursos em uma sala de aula cuja carga horária é limitada.

Nesse sentido, como forma de minimizar tais problemas se faz necessário o estudo e desenvolvimento de ferramentas capazes de auxiliar na rápida compreensão dos conceitos dessa metodologia e conseqüentemente na sua prática. Desse modo, sabendo que um jogo mobile pode ser capaz de despertar o interesse de seus jogadores para um determinado assunto, acredita-se que o jogo proposto neste trabalho será capaz de auxiliar o estudante no desempenho do aprendizado dos conceitos e processos do Scrum, por meio de um jogo que proporcione de forma lúdica e interativa a possibilidade de assumir os papéis do Scrum, e com isso desenvolver um maior entendimento da metodologia.

Por fim, este trabalho está estruturado da seguinte forma: seção 2, Referencial Teórico, são apresentados os conceitos gerais sobre a Metodologia de Desenvolvimento Ágil Scrum, jogos em auxílio da do aprendizado, gêneros de jogos e trabalhos relacionados e jogos digitais e sua aplicação na educação; seção 3, Materiais e Métodos, fornece uma breve explicação referente aos materiais e tecnologias utilizadas e apresenta a metodologia realizada para o desenvolvimento do trabalho; seção 4, Resultados, apresenta de forma detalhada os resultados obtidos na execução do trabalho;

## 2 REFERENCIAL TEÓRICO

A Metodologia de Desenvolvimento Ágil Scrum é uma metodologia que nos últimos anos tem ganhado destaque entre os profissionais e pesquisadores por ser uma metodologia adaptável, ágil, além de seus benefícios como transparência, *feedback*, entre outros.

Sabendo disso, esta seção apresenta a fundamentação teórica utilizada na construção deste trabalho. A subseção 2.1 detalha os conceitos da Metodologia de Desenvolvimento Ágil Scrum; na subseção 2.2 são apresentadas os conceitos relevantes sobre o auxílio de jogos em prol do aprendizado; a subseção 2.3, expõe os gêneros de jogos; a subseção 2.4 apresenta trabalhos relacionados a este projeto; a subseção 2.5 apresenta métodos e processos utilizados para o desenvolvimento de jogos educativos.

### 2.1 METODOLOGIA DE DESENVOLVIMENTO ÁGIL SCRUM

Scrum é uma metodologia estrutural que vem sendo utilizada para gerenciar projetos complexos desde o início de 1990, Schwaber e Sutherland (2017). O Scrum não é um processo, ou técnica ou um método definitivo, com ele é possível aplicar várias técnicas ou processos. Além de possuir uma ótima eficácia em suas práticas de gestão, permite uma evolução contínua do produto, da equipe e do ambiente de trabalho.

Os autores Schwaber e Sutherland (2017), definem três pilares que apoiam a implementação de controle de processo do Scrum: transparência, inspeção e adaptação, detalhadas respectivamente a seguir:

- **Transparência:** os aspectos significativos do processo devem estar visíveis aos responsáveis pelos resultados;
- **Inspeção:** os usuários do Scrum devem frequentemente inspecionar os artefatos Scrum e o progresso em direção ao objetivo para detectar variações indesejáveis;
- **Adaptação:** se um inspetor determina que um ou mais aspectos de um processo desviou para fora dos limites aceitáveis, e que o resultado do produto será inaceitável, o processo ou o material sendo produzido deve ser ajustado.

Espinha (2021) diz que, uma das vantagens do *Scrum* é que ele auxilia na entrega de um produto final de qualidade, pois ao finalizar uma *Sprint* (detalhado na subseção 2.1.2) é necessário entregar um produto utilizável que será validado pelo Product Owner, essa prática impacta no resultado final do projeto.



Schwaber e Sutherland (2017) definem 5 (cinco) valores para o Scrum: comprometimento; coragem; foco; transparência e respeito, respectivamente abordados a seguir:

- **Comprometimento:** as pessoas se comprometem pessoalmente em alcançar os objetivos do *Time Scrum*;
- **Coragem:** o *Time Scrum* precisa ter coragem para fazer a coisa certa e trabalhar em problemas difíceis;
- **Foco:** todos focam no trabalho da Sprint e nos objetivos do *Time Scrum*;
- **Transparência:** o *Time Scrum* e seus *Stakeholders* concordam em estarem abertos a todo o trabalho e aos desafios com execução dos trabalhos;
- **Respeito:** os membros do *Time Scrum* respeitam-se uns aos outros para serem pessoas capazes e independentes.

Segundo os autores, se os valores apresentados acima forem vivenciados pelo time, os pilares do Scrum ganham vida e constroem confiança para a equipe, e que o sucesso do Scrum depende desta vivência.

### 2.1.1 O TIME SCRUM

Schwaber e Sutherland (2017) definem em seu guia do Scrum três papéis importantes para a Metodologia de Desenvolvimento Ágil Scrum, sendo o *Product Owner*, o Time de Desenvolvimento e o *Scrum Master*, detalhados a seguir:

- **Product Owner:** “é o responsável pela satisfação e pelo atendimento das necessidades do cliente, podendo ser interpretado como o papel mais importante nas definições do produto e no atendimento das expectativas do cliente” (Cruz, 2015). Portanto, para isso o *Product Owner* possui as seguintes atribuições:
  - Maximizar o valor do produto e do trabalho do Time de Desenvolvimento, gerenciando o *backlog* do produto (detalhado na subseção 2.1.2);
  - Expressar claramente os itens do *backlog* do produto;
  - Ordenar os itens do *backlog* do produto seguindo uma importância para alcançar as metas esperadas pelo cliente;
  - Garantir o valor do trabalho realizado pelo Time de Desenvolvimento;
  - Garantir que todo o *backlog* do produto seja visível, transparente e claro para todos os interessados, mostrando o que o *Time Scrum* deve buscar;
  - Garantir que o Time de Desenvolvimento entenda os itens do *backlog* do produto para que este seja corretamente construído.

- **Time de Desenvolvimento:** é a equipe de profissionais responsáveis por entregar um incremento funcional do produto ao final de cada *Sprint* (detalhado na subseção 2.1.2). Essa equipe deve ser: auto-organizada, para que ninguém possa interferir no desenvolvimento dos incrementos; multifuncional, possuindo as habilidades necessárias para criar o incremento do produto; não reconhecer títulos individuais, ou seja, não deve haver distinção de cargos, funções, títulos ou senioridades; o Time não deve ser dividido em sub-times para realizações de atividades específicas. O Time de Desenvolvimento é o papel mais importante para a qualidade técnica e funcional do sistema ou produto.
- **Scrum Master:** é responsável por promover e sustentar o Scrum, realiza isso ajudando todos do time a entenderem a teoria, as práticas, as regras e os valores do Scrum. O *Scrum Master* trabalha para o *Product Owner*, para o Time de Desenvolvimento e para a Organização, da seguinte forma:
  - **Trabalhando para o Product Owner:** é responsável por garantir e gerenciar o *Backlog* do Produto de forma clara e concisa maximizando o valor e facilitando o entendimento por todos do Time etc.;
  - **Trabalhando para o Time de Desenvolvimento:** deve ser capaz de treinar o Time em autogerenciamento e interdisciplinaridade; ajudar na criação de produtos de alto valor; remover impedimentos para o processo do Time; facilitar os eventos Scrum; e treinar o time de desenvolvimento em ambientes nos quais o Scrum não é adotado e compreendido;
  - **Trabalhando para a Organização:** deve atuar de forma a auxiliar a organização na adoção e adaptação ao Scrum, fornecendo treinamento e planejamento necessários para aumentar a produtividade do Time Scrum bem como a eficácia da aplicação do Scrum na organização.

Portanto, torna-se compreensível que estes papéis sejam definidos de forma a garantir não só a entrega de um produto final com qualidade, mas também a evolução e aperfeiçoamento da equipe e da organização.

Segundo Schwaber e Sutherland (2017), os artefatos Scrum são projetados para maximizar a transparência das informações, para que todos os membros da equipe tenham o mesmo entendimento. Ainda de acordo com os autores, a transparência implica no valor do produto, portanto quando um artefato não está completamente transparente as decisões podem ser falhas gerando riscos para o projeto e diminuindo os valores do produto.

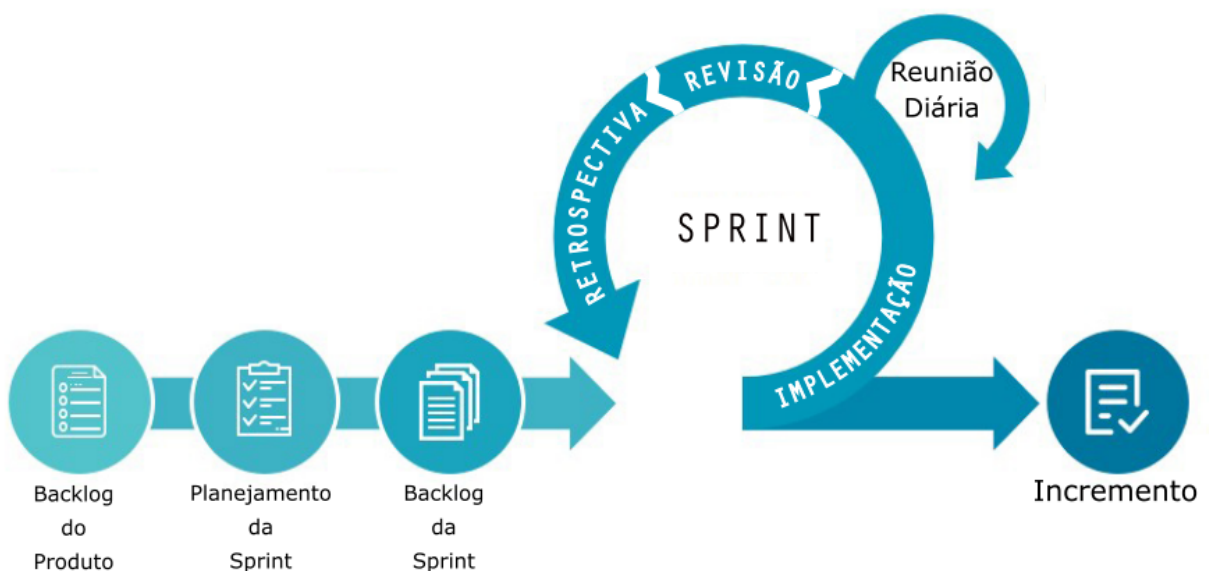
De acordo com (Cruz, 2015, p. 69) “o Time Scrum conta com o apoio de artefatos específicos e aplica regras que unem os eventos os papéis e os artefatos”, o artefato principal

do Scrum é o *backlog* que é composto de todos os requisitos do produto a ser entregue, ou seja, uma lista com todas as características, funções, tecnologias, melhorias e correções necessárias para a entrega de um produto. O *backlog* pode ser dividido em conjuntos que contribuem para objetivos específicos, alguns dos conjuntos que o backlog pode assumir é detalhado na subseção 2.1.2.

### 2.1.2 EVENTOS E ARTEFATOS SCRUM

Segundo Cruz (2015), o Scrum possui alguns eventos definidos com o intuito de minimizar a necessidade de reuniões, pois todo evento é uma oportunidade de inspecionar e adaptar o projeto. A Figura 1 apresenta os eventos e artefatos do Scrum.

Figura 1: Eventos e artefatos do Scrum



Fonte: Próprio autor

Na figura 1, são apresentados os eventos e os artefatos presentes no *Scrum*, detalhados a seguir:

- **Backlog do Produto:** corresponde a um artefato do Scrum cujo responsável é o *Product Owner*, esse artefato representa uma lista ordenada de todas as características, requisitos, funções e melhorias do produto, ele nunca está completo, mas está em evolução constante paralela ao desenvolvimento do produto. É somente através do *Backlog* do Produto que mudanças no projeto podem ser realizadas. O *Backlog* do

Produto é considerado um artefato vivo pois os requisitos nunca param de mudar, uma mudança nos requisitos de negócio, condições de mercado ou tecnologia podem causar mudanças no *Backlog* do Produto. Este artefato contém os itens que irão ocupar o Time de Desenvolvimento durante uma *Sprint*, ou seja, é através dele que será realizado o planejamento da *Sprint*;

- **Taskboard:** é um artefato que corresponde a uma grade contendo todas as histórias e suas respectivas tarefas, como se estivessem em uma linha de produção, cada história deve ser quebrada em tarefas a serem executadas para concluir a história;
- **Time-Boxed:** segundo Cruz (2015), são eventos com durações fixas, ou seja, todo evento tem uma duração fixa e deve ser encerrado ao término deste tempo, e há uma definição de trabalho a ser realizada, portanto um evento com um trabalho fechado e determinado para ser realizado dentro de uma duração fixa.
- **Sprint planning:** é um evento que acontece antes de cada *Sprint* e tem como objetivo definir o que pode ser entregue na próxima *Sprint* e como será realizado o trabalho para produzir a entrega. Este evento possui um *time-boxed* de no máximo oito horas para para *Sprint* de um mês de duração e para *Sprint* de duração menor o *time-boxed* também deve ser menor. O *Product Owner* inicia o evento apresentando os itens do *Backlog* do Produto e o Time de Desenvolvimento avalia quais itens poderão ser entregues no período e então é definido a meta da *Sprint* que fornece uma direção para o Time de desenvolvimento decidir como irá construir as funcionalidades necessárias e com isso é produzido o *Backlog* da *Sprint*;
- **Backlog da Sprint:** é um artefato do *Scrum* que contém o conjunto de itens do *Backlog* do Produto selecionados para a *Sprint*, junto com o plano para entrega do produto e atingir o objetivo da *Sprint*, é um artefato de responsabilidade do Time de Desenvolvimento, e sempre que um novo trabalho é necessário este é adicionado ao *Backlog* da *Sprint*;
- **Sprint:** é um evento com duração fixa, que deve durar de duas a quatro semanas e possuir meta e objetivo claro. De acordo com Schwaber e Sutherland (2017), durante uma *Sprint* não são feitas mudanças que possam pôr em perigo o objetivo da *Sprint*, as metas de qualidades não diminuem e o escopo pode ser melhorado e renegociado entre o *Product Owner* e o Time de Desenvolvimento. A *Sprint* contém cerimônias do *Scrum*, que são: as reuniões de planejamento da *Sprint*, a implementação, a revisão da *Sprint* e a retrospectiva da *Sprint*, detalhados a seguir:

- **Reuniões de planejamento da *Sprint*:** é um evento *time-boxed*, geralmente de oito horas para uma *Sprint* de um mês de duração, e deve ser utilizado para que o Time Scrum defina “o que será feito” na *Sprint* e “como será feito”;
- **Implementação:** etapa de execução, onde o Time de Desenvolvimento começa a trabalhar de acordo com o planejado, atendendo aos requisitos e metas definidas. Tem como duração no mínimo uma semana e no máximo quatro semanas, variando de acordo com a produtividade da equipe;
- **Revisão:** um evento realizado no final da *Sprint* para inspecionar o incremento e adaptar o *Backlog* do Produto, se necessário. Essa inspeção é realizada verificando quais itens do *Backlog* do Produto foram entregues e quais não foram, é avaliado os problemas encontrados e como foram resolvidos. É um evento importante pois a partir dele é levantado potenciais itens do *Backlog* do Produto para a próxima *Sprint*;
- **Retrospectiva:** evento realizado após a revisão, para que o Time *Scrum* possa inspecionar a si próprio e elaborar um plano de melhoria a ser aplicado na próxima *Sprint*. A inspeção deve avaliar o desenvolvimento da última *Sprint* em relação às pessoas, aos relacionamentos, aos processos e às ferramentas; identificar e ordenar os principais itens e as potenciais melhorias; criar um plano para implementar melhorias no trabalho do *Time Scrum*.
- **Reuniões Diárias:** é um evento *time-boxed* de até quinze minutos realizado todos os dias da *Sprint* com o objetivo de planejar o trabalho para as próximas vinte e quatro horas, verificando o que foi realizado no dia anterior e se há ou não impedimentos. Consiste em um evento realizado pelo Time de Desenvolvimento onde a participação do *Product Owner* e do *Scrum Master* estão restritas à condição de ouvintes. Para facilitar a realização do evento, o Time de Desenvolvimento pode utilizar perguntas pré-definidas, como por exemplo: "o que fiz ontem que ajudou o Time a atingir a meta da *Sprint*?"; "o que farei hoje para ajudar o Time a atingir a meta da *Sprint*?"; "qual obstáculo impede a mim ou o Time de atingir a meta da *Sprint*?".
- **Incremento:** é a soma de todos os itens do *Backlog* do Produto completados durante a *Sprint* e do valor de todas as *Sprint*'s anteriores. Um incremento é um passo em direção ao objetivo e sempre deve estar em condição de uso, independente do *Product Owner* decidir liberá-lo ou não.

Eventos tem o objetivo de avaliar a evolução do *Sprint* e do *Time Scrum*, e os artefatos são documentos com registros importantes para o desenvolvimento do incremento. Apesar do

Scrum ser uma Metodologia de Desenvolvimento Ágil, fornece o cuidado para ter uma documentação completa em paralelo com o produto, tendo no final um produto com os registros de todo o seu desenvolvimento, possibilitando também a avaliação da equipe com base nas metas.

## 2.2 JOGOS EM AUXÍLIO DO APRENDIZADO

Segundo Rodrigues (2013), um jogo é capaz de auxiliar no processo de aprendizagem, na tomada de decisões estratégicas, na resolução de desafios, na solução de conflitos, entre outros. Os jogos promovem a construção do pensamento do estudante, levando-o a adquirir novos conhecimentos e ativando sua imaginação, contribuindo para uma aprendizagem prazerosa. Rodrigues (2013) afirma que os jogos permitem que o jogador desenvolva, dentre outros, a representação do mundo, a linguagem, bem como a leitura e a escrita. Tais elementos promovem uma aprendizagem lúdica e corroboram para uma aprendizagem significativa, a qual caracteriza-se pela interação entre o cognitivo individual e as novas informações, que são ancoradas com os conceitos mais relevantes e inclusivos dos aprendizes, modificando assim, sua estrutura cognitiva inicial.

Segundo Savi (2011), jogos digitais educacionais proporcionam benefícios para o processo de ensino e aprendizagem, o autor cita os sete benefícios que os jogos digitais podem proporcionar, sendo eles:

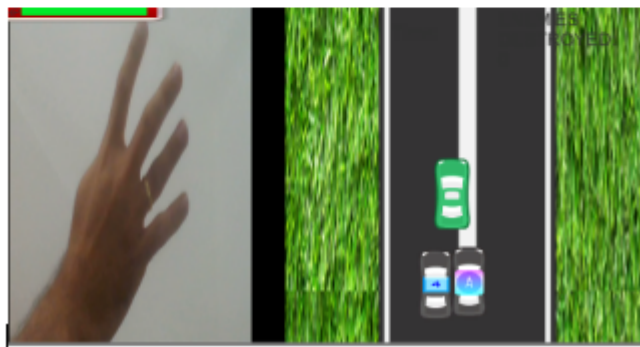
- **Efeito motivador:** segundo Savi (2011, p. 53, apud RITCHIE; DODGE, 1992), “as metas e desafios que precisam ser vencidos nos jogos geram provocações nas pessoas, mantendo-as motivadas e, em alguns casos, podem até recuperar o ânimo de quem perdeu o interesse pelo estudo”;
- **Facilitador do aprendizado:** jogos digitais possuem elementos gráficos e lúdicos que promovem o aprendizado em vários campos do conhecimento;
- **Aprendizado por descoberta:** através do *feedback* instantâneo e os cenários do jogo o jogador é estimulado a um processo de experimentação, exploração e colaboração, que promovem o aprendizado por descoberta, ou seja, o indivíduo tem um aprendizado duradouro pois aprende a partir das suas próprias explorações sobre o conteúdo em questão;

- **Experiência de novas identidades:** em jogos o jogador assume profissões como por exemplo ser um engenheiro, médico ou piloto, e assim realiza desafios correspondentes, fornecendo ao jogador conhecimentos relativos a essas profissões;
- **Socialização:** os jogadores podem interagir entre si para competirem, cooperarem ou trocarem informações por meio da Rede de Internet;
- **Coordenação motora:** alguns gêneros de jogos promovem o desenvolvimento da coordenação motora e habilidades espaciais, reeducando o comportamento do jogador;
- **Comportamento *expert*:** à medida que o estudante realiza os desafios impostos pelo jogo ele adquire experiência tornando-o *expert* no conteúdo proposto pelo jogo.

Portanto, com tantos benefícios, os jogos digitais já fazem parte do processo de ensino de diferentes níveis educacionais, desde o primário com jogos voltados para o auxílio da alfabetização da criança, como também em níveis mais avançados, por exemplo, na graduação. Um exemplo de jogo com o intuito educacional é o jogo *Libras Racer* (ARAUJO; GUIMARÃES; MENEZES, 2017) detalhado abaixo:

- ***Libras Racer*:** um jogo 2D com o objetivo de testar e auxiliar no conhecimento do jogador sobre a língua de sinais. O jogo possui uma dinâmica na qual o jogador deverá salvar um veículo de ataques inimigos, para isso, é necessário representar através de uma *webcam* o sinal em libras correspondente ao carro inimigo, conforme mostra a Figura 6 abaixo:

Figura 6: Apresentação do jogo *Libras Racer*



Fonte: (ARAUJO; GUIMARÃES; MENEZES, 2017).

Na Figura 6, é apresentado o jogo onde o carro verde representa o jogador e os demais veículos são os inimigos, cada inimigo possui uma identificação, podendo ser uma letra ou número. No trecho do jogo apresentado na figura o jogador apresenta o sinal em libras

correspondente ao número quatro, para então derrotar o inimigo que possui essa numeração e assim preservar a energia do veículo para que possa concluir o percurso.

## 2.3 GÊNEROS DE JOGOS

Para Silva et al. (2016), os jogos *mobiles* estão sendo jogados por todas as faixas etárias devido a grande acessibilidade proporcionada pelos dispositivos móveis, como *tablets* e *smartphones*. Silva et al. (2016) apresentam alguns jogos *mobiles* que foram desenvolvidos para o benefício da vida humana, como jogos que envolvem técnicas que auxiliam em terapias, fisioterapias, jogos educacionais voltados ao treinamento entre outros.

Segundo Villela (2021), os jogos eletrônicos possuem gêneros diferentes definidos por suas mecânicas e funcionalidades distintas. O autor apresenta ainda que um mesmo jogo pode se encaixar em mais de um gênero. Nesse sentido, Villela (2021) destaca os principais gêneros de jogo, são eles: Ação, Ação e Aventura, RPG (*Role-Playing Game*), Simulação, *Puzzle e Party Games*, Esportes, Estratégia, detalhados respectivamente a seguir:

- **Ação:** como o nome sugere, esse gênero tem como característica a ação, impondo aos jogadores desafios como batalhas diretas, outras modalidades de desafios são apresentadas de acordo com a escolha de um subgênero. Esse gênero engloba vários outros subgêneros, como: Plataforma, FPS (*First Person Shooter*), Luta, *Beat-em up*, Furtivo, Sobrevivência e Rítmico, detalhados abaixo:
  - **Plataforma:** como o nome sugere é um jogo baseado em plataforma onde o jogador controla um personagem através das ações de andar, pular, correr e cair, um exemplo clássico desse gênero é o Super Mario Bros;
  - **FPS:** são jogos de tiro em primeira pessoa, focados na perspectiva de um protagonista e com o objetivo de eliminar inimigos com o uso de armas de fogo;
  - **Luta:** a característica deste jogo está no combate corpo a corpo entre dois personagens. Geralmente esses jogos possuem vários personagens, cada um com suas particularidades;
  - **Beat-em up:** gênero focado no combate, contudo os inimigos são divididos em hordas, ou seja, o jogador terá vários inimigos simultaneamente;
  - **Furtivo:** o foco desse gênero está na estratégia para resolução de desafios para libertar o personagem;



- **Sobrevivência:** assim como o nome sugere, nesse gênero o jogador deverá elaborar estratégias para sobreviver, como construir casas, caçar animais, realizar plantações entre outros;
- **Rítmico:** esse gênero possui como principal elemento a música, ou seja, o jogador deverá executar o jogo de acordo com o ritmo musical escolhido.
- **Ação e Aventura:** este é um gênero combinado que possui duas mecânicas difundidas, possui como objetivos a exploração, a resolução de enigmas e a descoberta, envolvendo a ação como elemento adicional ao entretenimento. Esse gênero apresenta os seguintes subgêneros: Horror e Sobrevivência, e Metroidvania, detalhados a seguir:
  - **Horror e Sobrevivência:** esse gênero envolve elementos do sobrenatural, como monstros, fantasmas, zumbis entre outros, onde o personagem tem acesso a armas de fogo para auxiliar em sua sobrevivência;
  - **Metroidvania:** jogos desse gênero não possuem uma história linear. Deste modo o jogador precisa explorar o jogo descobrindo novas habilidades, armas e aliados, para então liberar um novo desafio.
- **RPG:** representado por suas características medievais ou fantasiosas, sua origem está ligada ao jogo de aventuras imaginárias *Dungeons & Dragons* publicado pela TSR, Inc em 1974. Esse gênero pode apresentar os seguintes subgêneros: RPG de Ação, MMORPG (*Massively Multiplayer Online Role-Playing Game*), *Roguelikes*, RPG *Sandbox*, detalhados respectivamente abaixo:
  - **RPG de Ação:** combina elementos dos jogos de ação e aventura, possui como principal característica o combate em tempo real, ou seja, o jogador deverá pensar e executar estratégias rápidas para sobreviver ao combate constante;
  - **MMORPG:** a particularidade deste subgênero está no número de jogadores envolvidos simultaneamente, devido a isso, os eventos do jogo ocorrem mesmo quando vários jogadores estão longe deles, pois os demais jogadores continuam a promover as mudanças dentro do jogo;
  - **Roguelikes:** a principal característica deste subgênero está na geração de calabouços com níveis gerados de forma procedural, ou seja, o jogo é programado para que um desafio tenha diferentes caminhos, isso evita com que o jogador decore sua resolução. Outra característica deste gênero é a “Morte Permanente”, onde, ao morrer, o jogador perde seu personagem e todas as suas conquistas, tendo então que iniciar uma nova jornada;

- **RPG *Sandbox*:** este subgênero está relacionado a capacidade de criação do jogador, pois ele terá liberdade para criar e modificar o mundo virtual do jogo de acordo com sua vontade.
- **Simulação:** tem como objetivo recriar situações do cotidiano em suas histórias envolvendo realidade virtual. Esse gênero pode apresentar os seguintes subgêneros: Simulação de Construção e Gestão, Simulação de Vida, Simulação de Veículos, detalhados a seguir:
  - **Simulação de Construção e Gestão:** é um subgênero em que o jogador podem construir, modificar e gerenciar as comunidades dentro do jogo;
  - **Simulação de Vida:** assim como o nome sugere, esse subgênero tem como objetivo simular a vida humana, onde o jogador vive ou controla um ou mais formas de vida dentro do jogo;
  - **Simulação de Veículos:** assim como os anteriores, o foco desse subgênero é proporcionar ao jogador a experiência de controlar um veículo, com todos os seus comandos e componentes.
- ***Puzzle e Party Games*:** envolve o conceito de resolução de enigmas, suas mecânicas podem ser bastantes variadas e se estende desde a resolução de exercícios de lógica até perguntas diversas;
- **Esportes:** possui como finalidade simular jogos da vida real, como: futebol, vôlei, basquete, corridas etc. Alguns buscam recriar o cenário do jogo com fidelidade ao real;
- **Estratégia:** inspirado em jogos de tabuleiro, a estratégia deve ser o principal foco do jogo, de modo a desenvolver táticas para a resolução dos desafios propostos.

Com isso, para a escolha do gênero de um de um jogo deve ser considerado um conjunto de elementos como por exemplo, a narrativa, os cenários, os personagens, os desafios, as recompensas, a portabilidade, a mecânica do jogo, entre outros.

## 2.4 TRABALHOS RELACIONADOS

Neste tópico são apresentados e detalhados dois trabalhos que possuem similaridade com o objetivo deste projeto.

### 2.4.1 Scrum'ed

O "Scrum'ed" é um jogo que convida o jogador a explorar um reino de fantasia, proposto por Schneider (2015). A ideia principal do jogo é fazer com que o jogador desenvolva o papel de Mestre Scrum para ajudar cavaleiros no planejamento e execução de um projeto solicitado pelo Rei. Para isso, o jogador deverá se aventurar pelos cenários do jogo solucionando impedimentos que estão fazendo com que a equipe fique ociosa e não conclua suas tarefas.

Segundo Schneider (2015), para o desenvolvimento do "Scrum'ed" foi definido o gênero de jogo RPG, pois esse gênero permite um alto índice de imersão e possibilita captar a atenção dos jogadores, de modo a permitir a aprendizagem dos conceitos básicos da metodologia de desenvolvimento ágil Scrum.

Para o ensino e aprendizagem dos conceitos básicos do Scrum, o autor utilizou os artefatos do Scrum da seguinte forma:

- **Product Backlog:** Lista acessível na interface contendo todas as missões existentes, suas prioridades e estimativas;
- **Sprint Backlog:** Lista acessível na interface contendo os desafios a serem realizados para que se possa concluir a *Sprint* proposta no jogo, com suas prioridades, estimativas e personagens responsáveis.
- **Taskboard:** Representa o quadro de missões do jogo, ou seja, uma lista acessível na interface contendo os desafios a serem realizados para que se possa concluir a *Sprint* atual do jogo, organizados por estados, sendo eles:
  - **To do:** desafios ainda não iniciados pelo jogador;
  - **In progress:** desafios aceitos e em andamento;
  - **Done:** desafios já concluídos;

A Figura 2 representa a narrativa do jogo onde o personagem principal está reunido com a equipe para realizar o *Sprint Planning*.

Figura 2 - Equipe reunida para o Planejamento da Sprint



Fonte: Schneider (2015).

Como mostra a Figura 2, ao chegar no palácio o jogador se encontra com a equipe para iniciar o planejamento da *Sprint*, para isso é apresentado o *Product Backlog*. Após a reunião cada membro retorna para suas respectivas atividades.

#### 2.4.2 SCRUM-Scape

O SCRUM-Scape é um jogo também de gênero RPG com temática medieval, onde o personagem principal precisa escapar de uma prisão, e para isso terá que realizar três missões. Segundo Camargo (2013), o jogo foi desenvolvido para ensinar metodologia de desenvolvimento ágil Scrum a estudantes dos cursos de computação, para isso o autor desenvolveu uma narrativa onde o personagem precisa realizar missões que abordam os seguintes conceitos do Scrum:

- **Responsabilidades de papéis:** *SCRUM Master, Development Team e Product Owner;*
- **Artefatos:** *Product Backlog, Sprint Backlog e Taskboard;*
- **Cerimônias:** *Daily SCRUM, Sprint Retrospective, Sprint Review e Sprint Planning.*

De acordo com Camargo (2013), o objetivo da primeira missão do jogo é apresentar os papéis do SCRUM ao jogador, para concluir essa missão o jogador deve resgatar todos os prisioneiros e para isso deve responder uma pergunta sobre os papéis Scrum. A Figura 3 apresenta a primeira missão do jogo.

Figura 3 - Primeira missão do jogo SCRUM-Scape

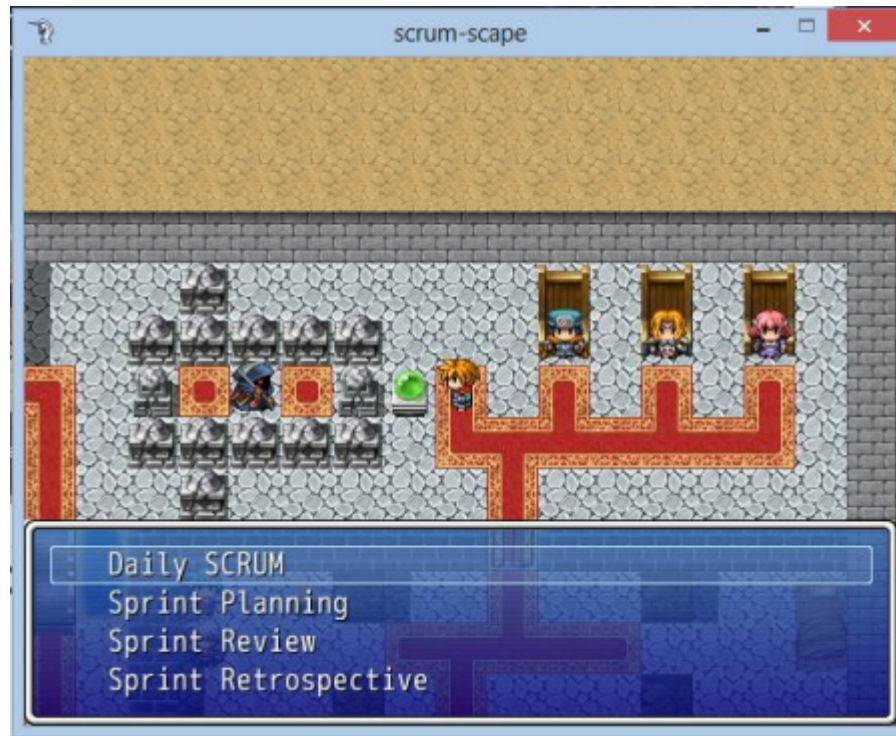


Fonte: Camargo (2013)

Na Figura 3, é apresentada a tela do jogo onde o jogador recebe sua primeira missão para aprender sobre os papéis Scrum. Esse cenário apresenta os três prisioneiros que precisam ser resgatados, assim como as informações necessárias para o resgate de cada um dos prisioneiros.

As cerimônias do Scrum são ensinadas ao jogador na Segunda missão do jogo, onde através de um cristal o jogador pode selecionar uma cerimônia do Scrum e assim responder uma pergunta sobre a mesma. A Figura 4 representa a segunda missão do jogo.

Figura 4 - Segunda missão do jogo SCRUM-Scape



Fonte: Camargo (2013)

Conforme a Figura 4, após resgatar os prisioneiros o jogador se direciona ao cristal verde, onde são apresentadas as cerimônias Scrum e o jogador deverá responder quatro perguntas de forma correta. Contudo, para cada resposta errada o jogador é enviado para outro cenário onde deverá derrotar um monstro para poder retornar e concluir as perguntas.

A terceira missão do jogo criado por Camargo (2013) conta com diversos cenários, em que cada cenário o jogador deverá responder uma pergunta sobre um artefato em específico, caso erre a resposta, o personagem é enviado para uma sala e deverá enfrentar um monstro para recuperar o artefato e devolver à equipe Scrum. A Figura 5 apresenta a terceira missão do jogo.

Figura 5 - Terceira missão do jogo SCRUM-Scape



Fonte: Camargo (2013)

Na Figura 5 o personagem se direciona ao cristal cinza para selecionar um dos seguintes artefatos: *Product Backlog Island*, *Burndown Chart Land* e *Taskboard Fields*. Ao escolher um artefato o personagem é enviado para o um cenário onde terá que responder a uma pergunta sobre o artefato escolhido. Para concluir o jogo é necessário que o jogador conclua os três artefatos de forma assertiva.

Portanto, como percebido nos trabalhos relacionados, ambos os jogos apresentados demonstram que a escolha do gênero RPG permite a construção de uma narrativa capaz de proporcionar o aprendizado ao jogador de forma interativa e construtiva. Com isso este trabalho também terá como gênero do jogo o RPG.

## 2.5 PROCESSOS E ADAPTAÇÕES DE METODOLOGIAS PARA DESENVOLVIMENTO DE JOGOS

Segundo o SENACRS (2018), o processo de criação de jogos está dividido entre: concepção, pré-produção, produção, testes e finalização. Ainda segundo o autor, essa divisão se adequa às etapas de metodologias de desenvolvimento ágeis já conhecidas como o *SCRUM*, *Extreme Programming - XP* e *Feature Driven Development - FDD*.

Nas seções seguintes serão apresentados exemplos de metodologias para desenvolvimento de jogos adaptadas de Metodologias Ágeis existentes.

### 2.5.1 ENGAGED

Battistella (2016) apresenta o ENgAGED (*EducatioNAl GamEs Development*) um processo de desenvolvimento de jogos educacionais, a ser aplicado em cursos de Computação em instituições de ensino superior. O autor modela o processo em cinco fases, sendo quatro referentes ao *design* instrucional e uma ao *design* de jogos, conforme apresentado na Tabela 1 a seguir:

Tabela 1–Visão geral do processo ENgAGED

<b>Fase 1. Análise da Unidade Instrucional (UI)</b>	
<b>A1.1 Especificar UI do jogo</b>	Especificar o contexto da UI, como curso, disciplina, pré-requisitos da disciplina, objetivos da disciplina e conteúdo programático.
<b>A1.2 Caracterizar aprendizes</b>	Caracterizar o público-alvo (alunos), p. ex. em termos de faixa etária, preferências de gênero de jogos, modo de interação, jogos favoritos etc. Caracterizar o ambiente, definindo a infraestrutura disponível para aplicação do jogo.
<b>A1.3 Definir objetivo(s) de desempenho</b>	Definir o(s) objetivo(s) que avaliam o desempenho do aluno ao final da UI. Ele(s) oferece(m) uma direção sobre a condução do conteúdo ao longo da UI.
<b>Fase 2. Projeto da Unidade Instrucional (UI)</b>	
<b>A2.1 Definir avaliação do aluno</b>	Definir como será estruturada a avaliação, para o aluno aprender com o jogo. Tipicamente a avaliação é inserida no próprio jogo, por meio de regras e apresentando feedbacks quando o aluno acerta ou erra um determinado conceito.
<b>A2.2 Definir conteúdo da estratégia instrucional</b>	A estratégia instrucional é Jogo Educacional. Assim, nesta atividade define-se o conteúdo e o sequenciamento do conteúdo ao longo do jogo.
<b>A2.3 Decidir pelo desenvolvimento ou utilizar jogo desenvolvido</b>	Decidir pelo desenvolvimento de um jogo educacional, ou pela utilização de um jogo existente. Caso a opção seja o uso de um jogo existente, a atividade orienta a busca por jogos disponíveis. Caso a opção seja pelo desenvolvimento é necessário seguir para Fase 3 – Desenvolvimento do Jogo Educacional.
<b>A2.4 Revisar o</b>	Revisar o modelo utilizado para avaliar o jogo educacional. Como



<b>modelo de avaliação do jogo</b>	padrão, propõem-se o modelo de avaliação de jogos educacionais MEEGA (Savi, Wangenheim e Borgatto, 2011).	
<b>Fase 3. Desenvolvimento do Jogo Educacional</b>		
<b>Fase 3.1. Análise do Jogo</b>		
<b>A3.1.1 Levantar requisitos do jogo</b>	Levantar os requisitos para identificação das funções e funcionalidades do jogo. Nesta atividade deve-se também definir como o conteúdo da UI, definidos na fase de projeto, serão distribuídos nos níveis do jogo.	
<b>Fase 3.2. Concepção do Jogo</b>		
<b>A3.2.1 Conceber o jogo</b>	Conceber o jogo, descrevendo as principais características, como objetivos do jogo, narrativa, regras, mecânica, elementos do jogo, pontuações e feedback educacional.	
<b>Fase 3.3. Design do Jogo</b>		
	Jogo digital	Jogo não-digital
<b>A3.3.1 Definir linguagem de programação ou game engine</b>	Definir a linguagem de programação ou game engine que será utilizada para o desenvolvimento do jogo. Nesta atividade são realizados testes empíricos com linguagens ou engines a fim de decidir qual será realmente utilizada para o desenvolvimento do jogo.	Não se aplica.
<b>A3.3.2 Produzir ilustrações ou imagens dos elementos do jogo</b>	Produzir as ilustrações que representam os elementos do jogo. Normalmente os elementos são personagens, cenários, objetos, artefatos, menus ou janelas de opções/configurações do jogo.	Produzir as imagens que representarão os elementos do jogo não-digital, como tabuleiros, cartas, fichas, dados, peças de montar (lego), etc.
<b>A3.3.3 Modelar o jogo</b>	Modelar os níveis do jogo, as bibliotecas adicionadas à game engine ou à linguagem de programação, os feedbacks educacionais e os diálogos dos personagens.	Modelar as etapas realizadas no jogo não-digital apresentando um fluxograma com a sequência de passados das jogadas realizadas pelo jogador, do início ao fim do jogo. Também é necessário modelar o feedback educacional do jogo.

<b>Fase 3.4. Implementação do Jogo</b>		
<b>A3.4.1 Produzir elementos do jogo</b>	Produzir os elementos criados na atividade A3.3.2 Produzir ilustrações ou imagens dos elementos do jogo.	
	Jogo digital	Jogo não-digital
	Codificação ou programação dos cenários, personagens, artefatos e objetos do jogo por meio de linguagem de programação ou game engine.	Produção dos elementos não digitais
<b>Fase 3.5. Testes do Jogo</b>		
<b>Fase 3.5. Testes do Jogo</b>	Realizar testes para detecção de erros e feedbacks para melhoria do jogo. Os testes são realizados pelos criadores do jogo e especialistas (conhecedores do domínio de aplicação), permitindo assim testar os níveis e funcionalidades do jogo, além de verificar os possíveis problemas antes de sua execução (Fase 4).	
<b>Fase 4. Execução da Unidade Instrucional (UI)</b>		
<b>A4.1 Planejar a execução do jogo</b>	Planejar a execução do jogo definindo data para jogar, local onde será jogado, e equipamentos que serão utilizados ou materiais que devem ser impressos.	
<b>A4.2 Instalar o jogo digital</b>	Jogo digital	Jogo não-digital
	Instalar o jogo conforme a sua plataforma. Caso seja um jogo com arquitetura cliente-servidor, é necessário instalá-lo no servidor e disponibilizá-lo para acesso via internet. Caso seja stand-alone é necessário disponibilizá-lo nos computadores que os alunos utilizarão no momento da execução do jogo. Caso seja um jogo de smartphone deve-se disponibilizá-lo para download.	Não se aplica.
<b>A4.3 Executar o jogo</b>	Executar o jogo em sala de aula, laboratório ou extraclasse.	
<b>Fase 5. Avaliação da Unidade Instrucional (UI)</b>		

<b>A5.1 Conduzir avaliação</b>	Conduzir a avaliação após a execução do jogo utilizando o instrumento para coleta de dados definidos na atividade A2.4 Revisar o modelo de avaliação do jogo.
<b>A5.2 Analisar dados da avaliação</b>	Realizar a análise dos dados coletados por meio de estatística descritiva, respondendo às questões de análise definidas na atividade A2.4 Revisar o modelo de avaliação do jogo.

Fonte: Battistella (2016)

Na tabela 1, são apresentadas as etapas desenvolvidas por Battistella (2016) necessárias para o desenvolvimento de um jogo educacional, portanto para o desenvolvimento deste trabalho será utilizado os processos da fase 3 -Desenvolvimento do Jogo Educacional, pois abrange todo o desenvolvimento do jogo deste a sua análise até a entrega.

### 2.5.2 SCRUM4GAMES

Scrum4Gmaes é uma adaptação da Metodologia de Desenvolvimento Ágil Scrum que apresenta alterações na estrutura do Time de Desenvolvimento, para atender as necessidades do desenvolvimento de um jogo digital, onde segundo Laubisch e Clua (2010), todo projeto de jogos digitais deve possuir duas equipes independentes e paralelas, pois no desenvolvimento de um jogo além da equipe de programação é necessário uma equipe de criação de materiais artísticos e sonoros, com isso Laubisch e Clua (2010) dividiram o time em Time de Desenvolvimento e o Time de *Game Design*, além da adição do papel de *Game Design Master*, do artefato *Game Design Wiki* e da cerimônia *Weekly Game Design Meeting*.

De acordo com Laubisch e Clua (2010), o Time de Desenvolvimento deverá possuir um *Scrum Master* e um *Product Owner*, para auxiliar o time no desenvolvimento do projeto. Já o Time de *Game Design* é composto por um *Game Design Master* e *Games Design's* que irão exercer a função de “cliente”, garantindo o desenvolvimento do jogo de acordo com os objetivos do projeto.

Laubisch e Clua (2010), definem que no Time de *Game Design* o *Game Design Master* é responsável pelo valor agregado ao projeto, sua função é exclusivamente garantir a concepção do projeto, para isso ele realizará a comunicação entre o Time de Desenvolvimento e o Time de *Game Design*, garantindo que não haverá perda de informação no caminho.

Para uma melhor comunicação interna entre os membros do projeto Laubisch e Clua (2010) criaram o Artefato *Game Design Wiki* uma ferramenta de comunicação interna do

projeto dinâmica e receptiva a mudanças, onde o Game Design Master exerce a função de editor e os Games Design's atuam como redatores.

Com o objetivo de unir a motivação da equipe com o valor agregado do produto, Laubisch e Clua (2010), criaram a Cerimônia *Weekly Game Design Meeting*, nessa cerimônia os membros irão apresentar sugestões, críticas, comentários etc., independente do seu papel no Time Scrum.

### 3 METODOLOGIA

Nesta seção são apresentados os materiais selecionados para o desenvolvimento do jogo mobile bem como os métodos a serem utilizados.

#### 3.1 MATERIAIS

As tecnologias utilizadas no desenvolvimento deste trabalho envolvem a plataforma de desenvolvimento *Unity*, uma plataforma que permite a utilização das linguagens *Javascript* e *CSharp* para o desenvolvimento de *scripts* para jogos (UNITY, c2021). Além disso, fornece diversos elementos gráficos para o desenvolvimento mais completo de jogos. Por fim, também foi utilizada a ferramenta *Adobe XD*, um aplicativo da *Adobe Systems* que permite organizar o fluxo de trabalho para desenvolvimento de aplicativos móveis e *desktop*, e será utilizada neste projeto para a modelagem (prototipação) do jogo.

#### 3.2 MÉTODOS

Para realização deste projeto serão utilizados os elementos da Fase 3 (Desenvolvimento do jogo educacional) do processo de desenvolvimento criado por Battistella (2016), apresentados na seção 2.4. Esses elementos serão utilizados conforme indica a Figura 7, além desses elementos também será utilizado o artefato *Game Design Wiki* desenvolvido por Laubisch e Clua (2010), apresentados na seção 2.4. Esse artefato será um documento com instruções sobre o desenvolvimento do jogo.

Figura 6 - Etapas de desenvolvimento do jogo



Fonte: Próprio autor

Na figura 6 é apresentada as etapas do processo desenvolvido por Battistella (2016), utilizadas para o desenvolvimento deste trabalho. Contudo, foi realizada uma adaptação onde as etapas ocorrem em formato cíclico, por ser uma forma que facilita na manutenção do projeto como um todo. As etapas apresentadas na figura acima serão detalhadas a seguir:

- **Entender o problema** nesta etapa foi realizada uma pesquisa bibliográfica envolvendo os conceitos necessários para o entendimento e desenvolvimento do projeto;
- **Desenvolvimento do Jogo Educacional:** corresponde ao processo desenvolvido por Battistella (2016) que será utilizado neste trabalho, contendo as seguintes atividades:
  - **Análise do Jogo:** nesta atividade definiu-se os requisitos para identificação das funções e funcionalidades do jogo. Além de também definir como os conceitos do Scrum, serão distribuídos nos níveis do jogo;
  - **Concepção do Jogo:** nesta etapa definiu-se as principais características do jogo, como objetivos, narrativa, regras, mecânica, elementos do jogo, pontuações e *feedback* educacional;
  - **Design do Jogo:** atividade destinada a produção das ilustrações que representam os elementos do jogo, como personagens, cenários, objetos, artefatos, menus ou janelas de opções/configurações do jogo. Após definir as ilustrações será necessário decidir o método de codificação, modelar/prototipar as telas do jogo, os cenários, a jogabilidade etc.;

- **Implementação do Jogo:** a implementação do jogo está separada em módulos, sendo eles o *front-end* do jogo como as telas e os cenários do jogo, seguido do *back-end* com a codificação das funções e suas regras de negócios;
- **Teste do Jogo:** esta atividade será realizada com alunos dos cursos de computação do CEULP/ULBRA onde serão coletados *feedbacks* e sugestões para melhorias do jogo.
- **Conclusão:** etapa final do desenvolvimento deste trabalho, consistirá na integração dos módulos, para que se tenha um jogo funcional.

## 4 RESULTADOS

Esta seção tem por objetivo apresentar e descrever as etapas que foram realizadas no presente trabalho e os resultados obtidos. As etapas foram divididas em: subseção 4.1 Análise do jogo 4.2 Concepção do jogo, 4.3 Design do jogo e 4.4 Implementação.

### 4.1 PROJETO DO JOGO

O jogo desenvolvido neste trabalho aborda de forma didática os conceitos do SCRUM e fornece desafios que colocam os conceitos estudados em prática. A distribuição dos conceitos do SCRUM entre os cenários do jogo acontece na seguinte sequência:

1. **Introdução ao Scrum:** aborda os conceitos relacionados aos pilares e valores do Scrum;
2. **Time Scrum:** apresenta os papéis Scrum e suas respectivas responsabilidades;
3. **Artefatos e Eventos Scrum:** apresenta a importância de cada artefato e o funcionamento dos eventos Scrum;
4. **Avaliação geral:** realiza uma avaliação geral do aprendizado do jogador.

Para auxiliar no desenvolvimento do jogo foi definida uma relação de requisitos funcionais e não funcionais que o jogo deve assumir, conforme apresentado na tabela 2 a seguir.

**Tabela 2 - Requisitos funcionais e não-funcionais**

REQUISITOS FUNCIONAIS	Relevância		
	Desejável	Importante	Essencial
O primeiro cenário do jogo deve abordar os conceitos de Introdução ao Scrum			X
O segundo cenário do Jogo deve abordar os conceitos relacionados ao Time Scrum			X
O terceiro cenário do jogo deve abordar os conceitos de Artefatos e Eventos Scrum			X
O jogo deve apresentar desafios que avaliem o conhecimento adquirido ingame			X
O jogo deve fornecer <i>feedback</i> com as questões respondidas pelo jogador	X		
O jogo deve pontuar o desempenho do jogador		X	

<b>REQUISITOS NÃO-FUNCIONAIS</b>			
O jogo deve estar disponível para plataforma mobile em sistemas android			<b>X</b>
O jogo deve ter conexão com internet		<b>X</b>	
Deve ser desenvolvido com componentes visuais de licença gratuita			<b>X</b>
A aplicação/usabilidade do jogo não deve passar mais que duas horas/aula		<b>X</b>	

Na tabela 2 são apresentados os requisitos funcionais e os requisitos não-funcionais selecionados para auxiliar no desenvolvimento deste trabalho. Portanto os requisitos funcionais abordam as características conceituais que o jogo deve assumir, ou seja está relacionado a jogabilidade do jogo, já os requisitos não-funcionais tratam o comportamento do jogo como produto, ou seja suas particularidades necessárias para o funcionamento.

## **4.2 CONCEPÇÃO DO JOGO**

O jogo desenvolvido neste trabalho é do gênero RPG, onde o jogador controla todas as ações do personagem principal, a escolha deste gênero se justifica pela sensação de liberdade que o mesmo proporciona ao jogador. Liberdade essa considerada neste contexto para auxiliar os jogadores na compreensão dos conceitos do Scrum, apresentados nos cenários do jogo. Além do gênero principal escolhido, o jogo possui “Morte Permanente”, uma das características do subgênero *Roguelikes*. Portanto ao morrer o jogador perde todas as suas conquistas, sendo necessário iniciar uma nova jornada.

### **4.2.1 NARRATIVA**

O jogo desenvolvido para este trabalho recebe o nome de Knight Scrum, que conta a história do cavaleiro Draco que recebe do rei a missão de proteger o reino Karpa do terrível monstro chamado SCRUM-BUG. O monstro se aproveita da falta de conhecimento dos cavaleiros sobre o SCRUM e lança um feitiço hipnótico, onde a única forma de escapar da hipnose sem sofrer danos críticos é resolvendo alguns desafios o mais rápido possível. Esses desafios estão relacionados aos conceitos Scrum e apresentados nos cenários 1, 2 e 3 do jogo.



Para tal missão, Draco deverá realizar uma viagem pelos campos do reino em busca de conhecimento sobre a Metodologia de Desenvolvimento Ágil Scrum. Quanto mais informações obter em sua jornada, mais fácil será enfrentar o monstro e assim proteger o reino. Para ter acesso aos conceitos Scrum, Draco terá que trocar seus pontos de honra adquiridos ao eliminar um monstro na loja de conceitos scrum. Draco conta com um sistema de evolução onde evolui seu *level* através dos pontos de experiência adquiridos ao eliminar um monstro.

O primeiro monstro que o cavaleiro Draco terá contato é o *Mushroom* um monstro espécie de cogumelo, já o segundo NPC é o *Flying* um monstro da espécie dos morcegos, o terceiro é o Skeleton esse monstro são membros da guarda real que foram mortos pelo terrível e ultimo NPC o SCRUM-BUG, que é BOSS do jogo, ou seja o monstro Chefe, ele pertence a espécie dos goblins.

#### 4.2.2 PERSONAGENS

O jogo conta com um personagem principal, que é o protagonista da história desenvolvida para o jogo, e quatro *Non-player Character* (NPC), que são personagens não jogáveis (não podem ser controlados pelo jogador). Na narrativa os NPC's são monstros que obedecem ao monstro SCRUM-BUG, e estes estão devastando os campos e masmorras do reino. Os personagens do jogo são:

- **Draco:**

O personagem principal do jogo se chama Draco, ele é um cavaleiro leal ao reino de Karpa. Draco recebeu do rei a importante missão de proteger o reino dos monstros enviados pelo monstro chefe, para isso, ele partiu em uma viagem pelos campos do reino, onde deve lutar e eliminar os monstros pelo caminho até seu confronto final com o SCRUM-BUG.

Draco é um cavaleiro experiente em combate corpo a corpo, porém lhe falta conhecimento sobre a Metodologia de Desenvolvimento Ágil Scrum, o que lhe trará dificuldades em sua jornada. Portanto, para concluir sua missão com êxito ele precisa comprar informações sobre o Scrum à medida em que avança em seu destino, na Figura 7 é apresentada a *sprite*, ou seja, a imagem/objeto gráfico utilizado para o representar o personagem principal.

Figura 7 - Sprite utilizada para o personagem principal



Fonte: Unity Asset Store<sup>1</sup>

<sup>1</sup>Disponível em: <https://assetstore.unity.com/packages/2d/characters/hero-knight-pixel-art-165188>

A Figura 7 apresenta as características visuais do personagem Draco, um cavaleiro que deve manusear uma espada e um escudo. Além dessas características, Draco possui os seguintes sistemas: sistema de vida, sistema de experiência, sistema de honra, sistema de ataque e defesa que serão detalhados na subseção 4.2.3 Sistemas do jogo.

- ***Mushroom, Flying, Skeleton e SCRUM-BUG***

São os NPC's que estão distribuídos pelo jogo, todos são monstros que possuem um temperamento agressivo, atacando qualquer outro personagem que se aproxima.

O objetivo dos três monstros é a destruição do reino e impedir que Draco consiga informações sobre os conceitos do Scrum, dificultando sua jornada e impossibilitando sua chegada ao monstro SCRUM-BUG.

O NPC SCRUM-BUG é o monstro mais forte dentro do jogo, capaz de hipnotizar seus inimigos enquanto realiza ataques, Ele é conhecedor dos conceitos do SCRUM e durante a hipnose realiza perguntas envolvendo os conceitos Scrum. Nesse sentido, para enfrentar tal inimigo o jogador terá que estar preparado. Na figura 8 são apresentadas as *sprites* utilizadas em cada NPC.

Figura 8 - Sprites utilizadas nos NPC's do jogo



Fonte: Unity Asset Store<sup>2</sup>

Disponível<sup>2</sup> em: <https://assetstore.unity.com/packages/2d/characters/monsters-creatures-fantasy-167949>

Na Figura 8 é possível observar as características visuais de cada NPC. Assim como o personagem principal, os NPC's também possuem os seguintes sistemas: sistema de vida, sistema de perseguição e sistema de ataque, detalhados na subseção 4.2.3 Sistema do Jogo.

#### 4.2.3 SISTEMAS DO JOGO

O ciclo de interação de um jogo é determinado pelos sistemas envolvidos nele, com isso, para uma boa experiência dentro do jogo foram criados dois tipos de sistemas, sendo eles: sistemas de personagens e sistemas de jogo. A seguir são apresentados os sistemas de personagens presentes no jogo:

- **Sistema de vida:** é um sistema que controla a vitalidade do personagem ou seu o HP dos personagens dentro do jogo. Todo personagem é iniciado com sua vitalidade em cem por cento, e a medida que recebe algum golpe do inimigo é decrementado uma

porcentagem de seu HP, e assim se sucede até que o personagem perca todo seu HP, resultado em sua morte permanente;

- **Sistema de Experiência:** esse sistema atribui ao personagem principal uma pontuação sempre que um monstro for eliminado. É através deste sistema que o jogador evolui o *level* de seu personagem;
- **Sistema de Honra:** assim como o sistema de experiência, esse sistema atribui ao personagem pontos de honra sempre que um monstro é eliminado;
- **Sistema de Ataque:** esse sistema é dividido em dois tipos, sendo o primeiro o sistema de ataque do personagem principal, onde é necessário que o jogador realize um comando para ativar o sistema. O segundo é o sistema de ataque dos NPC's, esse sistema é ativado automaticamente assim que atingido o requisito necessário, ou seja sempre que o personagem principal estiver a uma distância próxima ao NPC;
- **Sistema de Defesa:** é um sistema particular do jogador principal, onde ao ativado protege o personagem de qualquer ataque por 5 segundos, além disso, esse sistema proporciona ao personagem uma regeneração de HP, ou seja, o sistema restaura a vitalidade do personagem enquanto ativo;
- **Sistema de perseguição:** como apresentado na seção 4.4.2, os NPC's são personagens não jogáveis, portanto, o sistema de perseguição é um sistema destinado apenas aos NPC's. Este sistema faz com que os NPC's persigam o personagem principal sempre que o requisito necessário para sua ativação seja cumprido. O requisito para ativação é: entrar no campo de visão do NPC, ou seja, estar em uma localização próxima, para desativar o sistema o personagem principal precisa sair do campo de visão do NPC, ao fazer isso NPC encerra a perseguição.;

Assim como os personagens, o jogo em geral possui sistemas que controlam toda a interação do jogador com os componentes visuais do jogo. Sendo eles detalhados a seguir:

- **Sistema de *Respawn*:** esse sistema se faz necessário pois sempre que um monstro morre no jogo ele é "apagado", ou seja, não é mais exibido no jogo. No entanto, com o sistema de *respawn*, um novo monstro aparece no mapa após 30 segundos da morte do anterior.
- **Sistema da Loja:** esse sistema controla as informações sobre o Scrum apresentadas ao jogador, assim como os requisitos para cada informação a ser comprada;
- **Sistema de Desafios:** controla os desafios, ou seja, é o sistema que testa o conhecimento do jogador sobre os conceitos do Scrum aprendidos dentro do jogo, assim como os requisitos de cada desafio.

- **Sistema Reiniciar:** esse sistema é responsável por fazer com que o jogador retorne ao início do jogo sempre que o mesmo é derrotado por algum monstro.

Portanto, os sistemas apresentados controlam toda a dinâmica do jogo, ou seja, são mecanismos indispensáveis para o funcionamento do jogo, pois cada ação do jogo ou interação com o jogo requer um controlador para executar o resultado esperado. Por exemplo, por ser um jogo que envolve combate, é esperado que um dos personagens envolvidos seja eliminado, com isso o sistema de vida entra em ação.

#### 4.2.4 CENÁRIOS

O jogo é composto por quatro cenários sendo o último a sala do BOSS, onde cada cenário contém conceitos do Scrum que o jogador deverá explorar e estudar para poder liberar o acesso a sala do BOSS. Cada cenário possui um portal de teleporte que ativa o sistema de desafios, esses desafios estão detalhados na subseção 4.2.5 Desafios. Os conceitos do Scrum estão distribuídos nos cenários conforme a tabela 3 a seguir.

**Tabela 3 - Cenários e Conceitos do Scrum**

<b>Cenário</b>	<b>Conceitos</b>
<b>1</b>	Introdução ao Scrum
<b>2</b>	Time Scrum
<b>3</b>	Artefatos e Eventos Scrum
<b>Sala do BOSS</b>	Avaliação geral

Fonte: Próprio autor

Conforme apresentado na tabela 3 acima, cada cenário do jogo apresentará ao jogador um determinado conceito referente ao SCRUM. No cenário 1, o jogador terá uma introdução ao SCRUM onde aprenderá sobre principais pilares e os valores do SCRUM, no cenário 2 o jogador terá contato com conceitos que envolvem o Time Scrum, ou seja, aprenderá sobre os papéis de cada membro e suas obrigações, já no cenário três é apresentado ao jogador os conceitos referentes aos eventos e artefatos do Scrum. Portanto, ao concluir os 3 cenários o jogador possuirá conhecimento necessário para avançar à sala do BOSS, para lutar contra o SCRUM-BUG e realizar a avaliação geral, que são desafios envolvendo os conceitos aprendidos em cada um dos cenários anteriores.

Contudo, para ter acesso aos conteúdos disponíveis em cada cenário o jogador precisará realizar uma troca equivalente, ou seja, precisará comprar informações com pontos de honra.

As informações são fornecidas de forma aleatória e não serão armazenadas, ou seja, o jogador deverá estudá-la com atenção. Os pontos de honra são adquiridos ao derrotar os monstros espalhados pelo cenário. A tabela 4 a seguir demonstra a quantidade de pontos necessários para comprar informações em cada cenário.

**Tabela 4 - Pontos necessários por informação**

<b>Cenário</b>	<b>Pontos por informação</b>
1	5 pontos de honra
2	10 pontos de honra
3	20 pontos de honra

Fonte: Próprio autor

Cada cenário cobra uma pontuação específica por informação fornecida, portanto, para cada informação Scrum adquirida na loja no cenário 1, será descontado do jogador 5 pontos de honra e não há limite para a quantidade de compras que ele pode realizar. A tabela 5 a seguir demonstra um modelo de informações sobre o Scrum que o jogador poderá adquirir em seu respectivo cenário.

**Tabela 5 - Demonstração de modelo de informação sobre os conceitos Scrum.**

<b>Cenário</b>	<b>Informações</b>
<b>1</b>	Os três PILARES que apoiam a implementação de controle de processo do Scrum são: TRANSPARÊNCIA, INSPEÇÃO e ADAPTAÇÃO
<b>1</b>	O Pilar ADAPTAÇÃO: se um inspetor determina que um ou mais aspectos de um processo desviou para fora dos limites aceitáveis, e que o resultado do produto será inaceitável, o processo ou o material sendo produzido deve ser ajustado
<b>2</b>	O Product Owner deve 'Maximizar' o valor do produto e do trabalho do Time de Desenvolvimento, gerenciando o backlog do produto.
<b>2</b>	O Scrum Master: é responsável por promover e sustentar o Scrum, realiza isso ajudando todos do time a entenderem a teoria, as práticas, as regras e os valores do Scrum
<b>3</b>	Backlog do Produto: corresponde a um artefato do Scrum cujo responsável é o Product Owner, esse artefato representa uma lista ordenada de todas as

	características, requisitos, funções e melhorias do produto, ele nunca está completo, mas está em evolução constante paralela ao desenvolvimento do produto.
<b>3</b>	Sprint: é um evento com duração fixa, que deve durar de duas a quatro semanas e possuir meta e objetivo claro. De acordo com Schwaber e Sutherland (2017), durante uma Sprint não são feitas mudanças que possam pôr em perigo o objetivo da Sprint, as metas de qualidades não diminuem e o escopo pode ser melhorado e renegociado entre o Product Owner e o Time de Desenvolvimento

Fonte: Próprio autor

Na tabela 5, é apresentado exemplos de informações sobre os conceitos Scrum que o jogador poderá adquirir de acordo com os conceitos determinados anteriormente para cada cenário.

Cada cenário possui um limite de pontos de honra que o jogador poderá acumular, esse limite motiva o jogador a realizar compras constantes sobre os conceitos Scrum e assim estar sempre adquirindo conhecimento sobre a metodologia. A tabela 6 a seguir apresenta o limite de pontos de honra de cada cenário.

**Tabela 6 - Limite máximo de pontos acumulados por cenário**

<b>Cenário</b>	<b>Limite de pontos de honra</b>
<b>1</b>	30 pontos de honra
<b>2</b>	60 pontos de honra
<b>3</b>	110 pontos de honra

Fonte: Próprio autor

Na tabela 6 é apresentada a quantidade de pontos de honra que podem ser acumulados em cada cenário, portanto, no cenário 1 o jogador poderá acumular um máximo de 30 pontos de honra, no cenário 2 até 60 pontos de honra, e no cenário 3 poderá atingir 110 pontos de honra.

Os pontos de honra possuem duas utilidades dentro do jogo, sendo que a primeira se refere a compra de informações sobre os conceitos Scrum apresentado anteriormente, e a segunda utilidade é para ativar os desafios realizados nos portais de teleporte presente em cada um dos cenários do jogo, exceto na sala do BOSS.

#### **4.2.5 DESAFIOS E PONTUAÇÕES**

Para avançar entre os cenários do jogo, o jogador deverá cumprir alguns desafios predeterminados para cada cenário. Esses desafios estão divididos em duas etapas, sendo a primeira etapa correspondente à evolução do personagem, e a segunda etapa referente à resolução de forma assertiva às perguntas de cada portal de teleporte de cada cenário.

A etapa de evolução do personagem é determinada pelo incremento do *level* inicial do personagem, onde inicia-se o jogo com *level* 1(um) e requer que o jogador acumule pontos de experiências eliminando os monstros de cada cenário. A tabela 7 a seguir demonstra a quantidade de pontos de experiência necessários para evoluir para um respectivo *level*.

**Tabela 7 - Level x Pontos de experiência**

<b>Cenário</b>	<b>Level</b>	<b>Pontos de experiências necessários</b>
<b>1</b>	<b>1</b>	0 pontos
	<b>2</b>	300 pontos
	<b>3</b>	1000 pontos
<b>2</b>	<b>4</b>	1700 pontos
	<b>5</b>	3000 pontos
	<b>6</b>	10000 pontos
<b>3</b>	<b>7</b>	25000 pontos
	<b>8</b>	50000 pontos
	<b>9</b>	90000 pontos
<b>Boss</b>	<b>10</b>	200000 pontos

Fonte: Próprio autor

Na tabela 7 é possível observar que cada cenário possui seu limite de evolução, portanto no cenário 1 o jogador poderá alcançar no máximo o *level* 3 (três), fazendo com que o jogador se sinta motivado a progredir dentre os cenários para poder alcançar o *level* posterior do jogo.

Para concluir a segunda etapa correspondente aos desafios do cenário, o jogador precisará de pontos de honra adquiridos ao eliminar os monstros espalhados pelo cenário, pois



o portal de teleporte requer duas condições para o seu perfeito funcionamento: a primeira, uma quantidade  $x$  de pontos de honra, a segunda, que o jogador responda no mínimo 60% de suas perguntas de forma assertiva.

**Tabela 8 - Requisito e penalidade, desafio do portal de teleporte**

<b>Portal de Teletransporte</b>	<b>Pontos de Honra requerido</b>	<b>Pontos cobrados por respostas</b>	
Cenário 1	<b>30</b>	3 Respostas corretas	<b>15</b>
		2 Respostas corretas	<b>20</b>
		$\leq 1$ Resposta correta	<b>30</b>
Cenário 2	<b>60</b>	3 Respostas corretas	<b>30</b>
		2 Respostas corretas	<b>45</b>
		$\leq 1$ Resposta correta	<b>60</b>
Cenário 3	<b>110</b>	3 Respostas corretas	<b>55</b>
		2 Respostas corretas	<b>75</b>
		$\leq 1$ Resposta correta	<b>110</b>

Fonte: Próprio autor

A tabela 8 apresenta a quantidade de pontos de honra necessários para ter acesso ao desafio de cada “portal de teleporte” em cada cenário. Também é apresentado o valor de pontos cobrados de acordo com o número de respostas corretas. Portanto, ao realizar o desafio no cenário 1 e adquirir um acerto inferior ou igual a uma resposta correta será descontado 30 pontos de honra do jogador, acertando duas respostas será descontado 20 pontos de honra e ao acertar três será descontado 15 pontos de honra.

O desempenho do jogador está relacionado aos pontos de honra adquiridos em cada cenário. Portanto o jogador poderá avançar do cenário 1 para o cenário 2 possuindo no máximo 15 pontos de honra, do cenário 2 para o cenário 3 com no máximo 30 pontos de honra, e do cenário 3 para a sala do BOSS máximo de 55 pontos, dessa forma o jogador pode concluir os 3 cenários com um aproveitamento máximo de 100 pontos de honra.

Para a sala do BOSS o jogador poderá concluí-la com pontuações variando entre 60 pontos a 100 pontos, onde será realizado a média geral entre os pontos dos 3 cenários, sendo que o somatório dos pontos dos 3 cenários possui peso 1 enquanto e os pontos da sala do BOSS possui peso 2.

Portanto a pontuação final do jogador ao concluir o jogo seguirá a seguinte somatória:  $[ \text{média cenários} + ( \text{média sala do BOSS} * 2 ) ] / 3$ . Essa pontuação é apresentada no *feedback* educacional do jogo, que consiste em um relatório contendo os desafios realizados no jogo e desempenho do jogador, detalhado na seção 4.5 Feedback educacional. A tabela 9 a seguir apresenta os formatos de perguntas apresentadas nos desafios.

**Tabela 9 - Demonstração de modelo de perguntas do cenário 3**

<b>Pergunta</b>	(FCC, 2011). Na metodologia Scrum, Sprint é uma iteração de duração menor ou igual a um mês, onde uma parte incremental e funcional do produto está potencialmente pronta para entrega. É INCORRETO afirmar que, nessa fase,
Alternativa 1	o escopo pode ser esclarecido e renegociado entre o time de desenvolvimento e o proprietário do produto.
Alternativa 2	nenhuma alteração que afetaria a meta do Sprint é efetuada.
Alternativa 3	a composição do time de desenvolvimento permanece constante.
Alternativa 4	as metas de qualidade não diminuem
Alternativa 5	o Sprint pode ser cancelado por decisão do Scrum Master.

Fonte: Próprio autor

Na tabela 9, é apresentado o modelo de pergunta realizada no desafio do cenário 3, portanto é possível observar que os desafios realizados são condizentes com os conceitos Scrum estudados no cenário em que o desafio é realizado.

#### 4.2.4 REGRAS

Para o funcionamento do jogo Knight Scrum desenvolvido neste trabalho foi definido o seguinte conjunto de regras:

- 1- Só é permitido avançar de cenário após cumprir os requisitos do cenário em que se encontra, a seguir é apresentado os requisitos de cada cenário:

- Para avançar ao cenário 2 o jogador deverá evoluir o seu personagem até o *level* 3 e obter no mínimo 2 respostas corretas no desafio do portal de teleporte do cenário 1.
- Para avançar ao cenário 3 o jogador deverá evoluir o seu personagem até o *level* 6 e obter no mínimo 2 respostas corretas no desafio do portal de teleporte do cenário 2.
- Para avançar à sala do BOSS o jogador deverá evoluir o seu personagem até o *level* 9 e obter no mínimo 2 respostas corretas no desafio do portal de teleporte do cenário 3.

2- Cada portal de teleporte possui seus requisitos e penalidades;

3- Para cada pergunta do portal de teleporte será descontados pontos de honra;

4- Não há tempo limite para cumprir os desafios do portal de teleporte de cada cenário;

5- Sempre que morrer, o jogador será enviado para o cenário 1, perdendo toda sua pontuação e *level* adquiridos.

O conjunto de regras apresentados nesta subseção complementam a dinâmica do jogo pois expressam como o jogador deve se comportar dentro do jogo, e como o jogo se comporta de acordo com a interação do jogador.

### 4.3 DESIGN DO JOGO

Com base nas definições iniciais do jogo, foi desenvolvido o *design* do jogo, onde são apresentados seus elementos visuais que estão disponíveis ao jogador. A figura 9 representa a tela “dados do jogador”, que será o primeiro contato do jogador com o ambiente do jogo.

Figura 9 - Tela dados do jogador

Fonte: Próprio autor

Na figura 9, são solicitadas algumas informações do jogador como seu nome, email e curso. Essas informações são importantes pois estão relacionadas ao *feedback* educacional do jogo.

Após preencher as informações solicitadas e clicar no botão “Jogar”, o jogador terá acesso aos principais componentes de informações e ações do jogo, conforme apresentado na figura 10 a seguir:

Figura 10 - Principais componentes do jogo



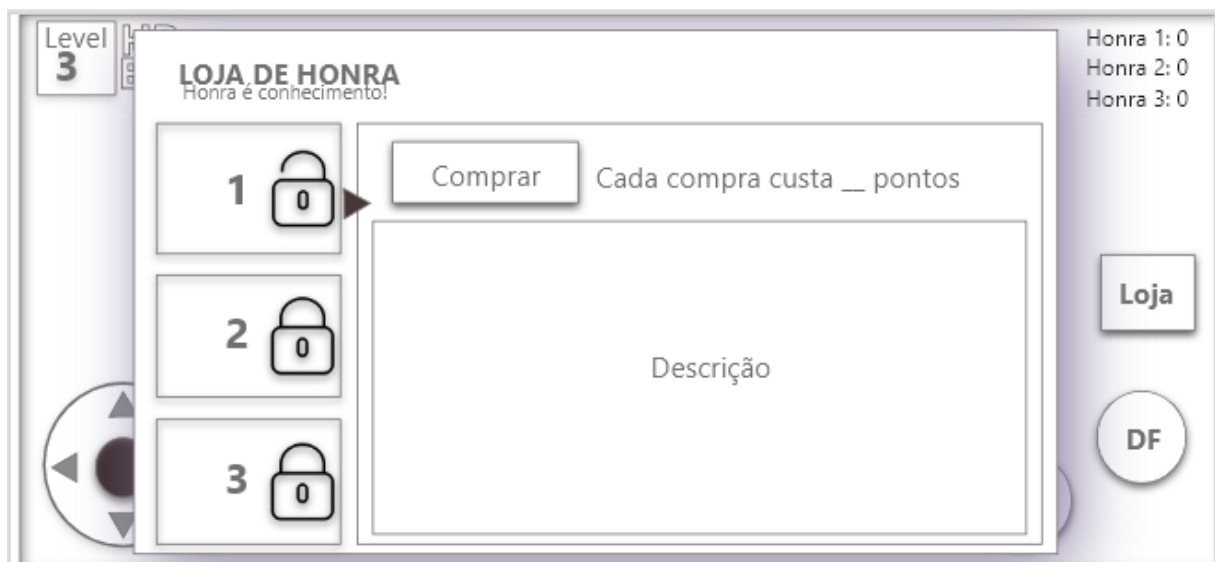
Fonte: Próprio autor

Os componentes apresentados na figura 10 correspondem às informações e mecanismos necessários para poder jogar. Esses elementos estão representados por:

1. **Estatísticas do personagem:** Apresenta para o jogador as estatísticas atuais do personagem como:
  - 1.1. **Level:** Corresponde ao *level* atual do personagem;
  - 1.2. **HP:** Barra de progressão da vida do personagem;
  - 1.3. **Exp:** Barra de progressão da experiência do personagem.
2. **Touch Controls:** é o controle de movimentação do personagem onde o jogador poderá realizar as ações de mover o personagem para a direita, esquerda, para cima e para baixo;
3. **Pontos de honra:** Esse elemento apresenta ao jogador os pontos de honra acumulados em cada um dos cenários do jogo;
4. **Ações do personagem:** são as ações de combate do jogador onde ele poderá executar as ações de atacar e defender;
5. **Loja de Honra:** Este é um elemento chave para o jogo, pois é onde o jogador pode comprar informações sobre os conceitos do SCRUM, ou seja, troca pontos de honra por conhecimento.

Caso o jogador selecione a opção “Loja”, lhe será apresentada a tela “loja de honra” conforme a figura 11 a seguir:

Figura 11 - Tela loja de honra



Fonte: Próprio autor

O *design* desenvolvido na figura 11 apresenta a loja de honra que poderá ser acessada em qualquer cenário do jogo. Contudo, seus elementos possuem restrição de acesso, onde o

jogador só terá acesso às informações respectivas ao cenário em que se encontra. Ao realizar uma compra, será apresentada no campo “Descrição” uma informação sobre os conceitos SCRUM daquele cenário, de forma aleatória.

A transição entre os cenários do jogo é executada através do elemento portal de teleporte, disponível no mapa de cada cenário. A disposição de cada portal é apresentada na figura 12 a seguir.

Figura 12 - Estrutura do mapa do jogo

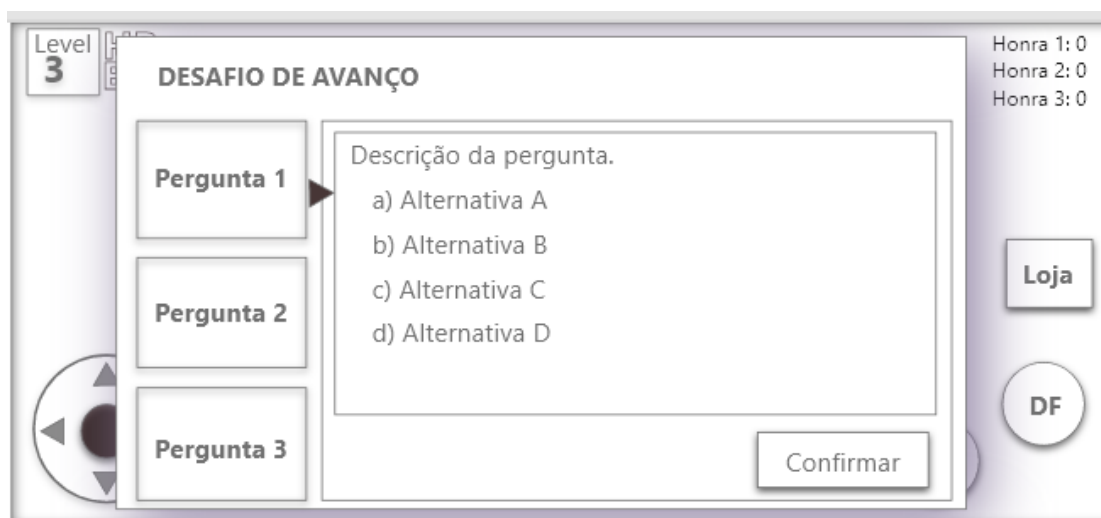


Fonte: Próprio autor

Conforme apresentado na figura 12, para ter acesso ao próximo cenário no jogo, o jogador deverá percorrer o cenário até localizar o portal de teleporte. O jogador poderá avançar e retroceder entre os cenários de 1 a 3, porém, ao ter acesso a sala do BOSS ele não terá opção de retornar ao cenário 3, sendo assim, ao finalizar os requisitos da sala do BOSS ele concluirá o jogo.

Ao entrar em contato com um portal de teleporte e possuir os requisitos necessários, ou seja, “*level* e pontos de honra”, será apresentado ao jogador a tela desafio de avanço, conforme demonstrado pela figura 11 a seguir.

Figura 13 - Tela desafio de avanço



Fonte: Próprio autor

A figura 13 apresenta a tela de desafio de avanço que é composta por 3 perguntas com o máximo de 4 alternativas cada. As perguntas são apresentadas de forma aleatória de modo a evitar a possibilidade do jogador realizar os desafios de forma intuitiva. Ao concluir as 3 perguntas e atingir o requisito do desafio, o jogador é enviado ao próximo cenário, caso contrário permanecerá no cenário em que está.

#### 4.4 IMPLEMENTAÇÃO

A implementação do jogo foi iniciada com o desenvolvimento da interface do jogo, tendo como base as telas apresentadas na seção 4.3 Design do jogo e seguindo os sistemas propostos conforme apresentado na seção 4.2.2 Sistemas do jogo.

A principal interface do jogo corresponde a Figura 10 da subseção 4.3 Design do jogo, onde apresenta os componentes visuais do jogo, ou seja, as principais interações do jogador com o jogo. Na figura 14 a seguir são apresentados os componentes gráficos desenvolvidos.

Figura 14 - Componentes gráficos desenvolvidos



Fonte: Próprio autor

Na figura 14 são apresentados os componentes gráficos desenvolvidos para o jogo, conforme proposto na seção 4.3 Design do jogo. O funcionamento de cada elemento é controlado pelos sistemas apresentados na seção 4.2.2 Sistemas do jogo. O componente “Loja” é controlado pelo “sistema da loja” presente no *script* Funcoes, na figura 15 a seguir é apresentado trecho do código presente no script Funcoes.



Figura 15 - Trecho de código presente no script Funcao

```

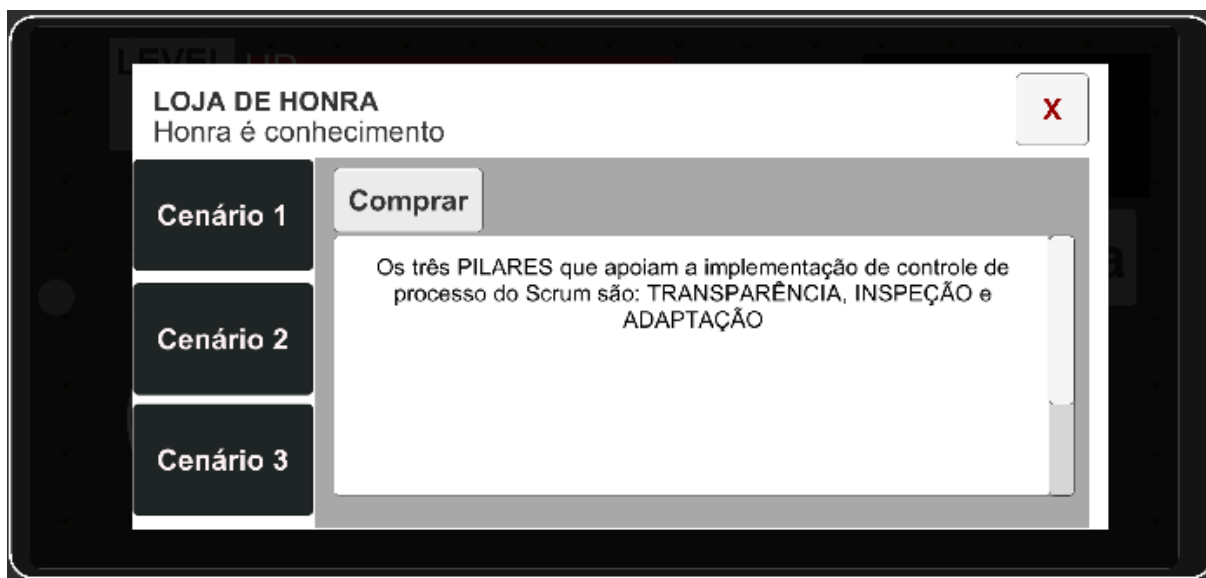
51 public void openLoja()
52 {
53     telaLoja.gameObject.SetActive(!telaLoja.gameObject.activeInHierarchy);
54 }
55
56 public void comprarInfo(Text descricao)
57 {
58     if(Player.cenario == 1)
59     {
60         if(float.Parse(HonraScore_1.text) >= 5 )
61         {
62             int aux = 0;
63             while(aux != 1)
64             {
65                 int _rand = Random.Range(0, ConceitosScrum.Count);
66                 if(ConceitosScrum[_rand]["cenario"] == Player.cenario+"")
67                 {
68                     descricao.text = ConceitosScrum[_rand]["descricao"];
69                     HonraScore_1.text = (float.Parse(HonraScore_1.text) - 5) +"";
70                     aux = 1;
71                 }
72                 else{
73                     aux = 0;
74                 }
75             }
76         }
77         else
78         {
79             descricao.text = "Voce não possui pontos de Honra suficientes.";
80         }
81     }

```

Fonte: Próprio autor

Na figura 15 é possível observar que ao pressionar o botão Loja na interface do jogo o sistema ativa a Loja na interface através da função `openLoja()` (linha 51), Esse é um dos principais trechos de código do jogo pois a partir dele o jogador terá contato com os conceitos da Metodologia de Desenvolvimento Ágil Scrum. Analisando a função `comprarInfo()` (linha 56) é possível notar que o código verifica se o jogador está no cenário 1 do jogo, através do condicional da (linha 58), caso a condição seja verdadeira e o jogador possuir os pontos de honras verificados na (linha 60), ele receberá de forma aleatória uma informação sobre o Scrum, que está presente em `ConceitosScrum[_rand]["descricao"]` que é uma lista com os conceitos Scrum presente no jogo. Na figura 16 é apresentado a tela loja de honra desenvolvida.

Figura 16 - Tela loja de honra desenvolvida



Fonte: Próprio autor

Na figura 16 é apresentada a visualização dos conceitos Scrum após o jogador realizar uma compra no cenário 1 do jogo. Caso o jogador realize uma segunda compra essa informação some da tela e é apresentada a nova informação comprada.

Após o desenvolvimento dos sistemas necessários para o funcionamento dos componentes gráficos foi desenvolvido os cenários e os portais de teleporte presentes no jogo, pois os portais de teleporte possuem um sistema muito importante para a dinâmica do jogo que é o sistema de desafios, onde o jogador coloca em prática os conceitos Scrum aprendido no cenário atual. Nas figuras 17, 18, 19 e 20 são apresentadas características de cada cenário, bem como a localização de cada portal de teleporte.

Figura 17 - Cenário 1 desenvolvido para o jogo



Fonte: Próprio autor

Figura 18 - Cenário 2 desenvolvido para o jogo

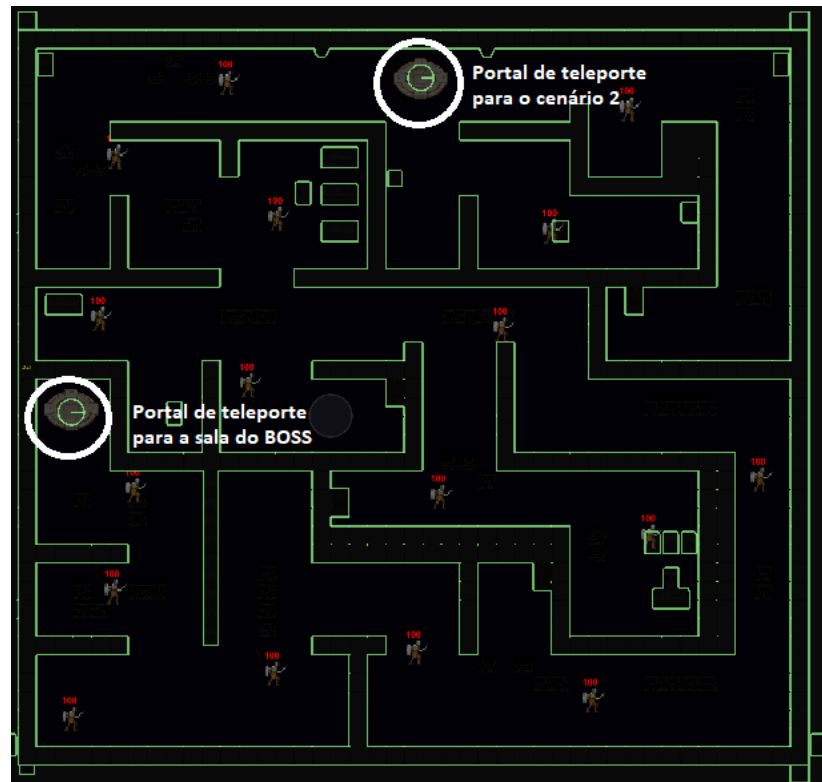


Fonte: Próprio autor

Nas figuras de 17 e 18, são apresentados os cenários 1 e 2 do jogo bem como as particularidades de cada um.

Sendo que o cenário 1 representa um campo aberto com vegetação, já o cenário 2 apresenta um campo árido.

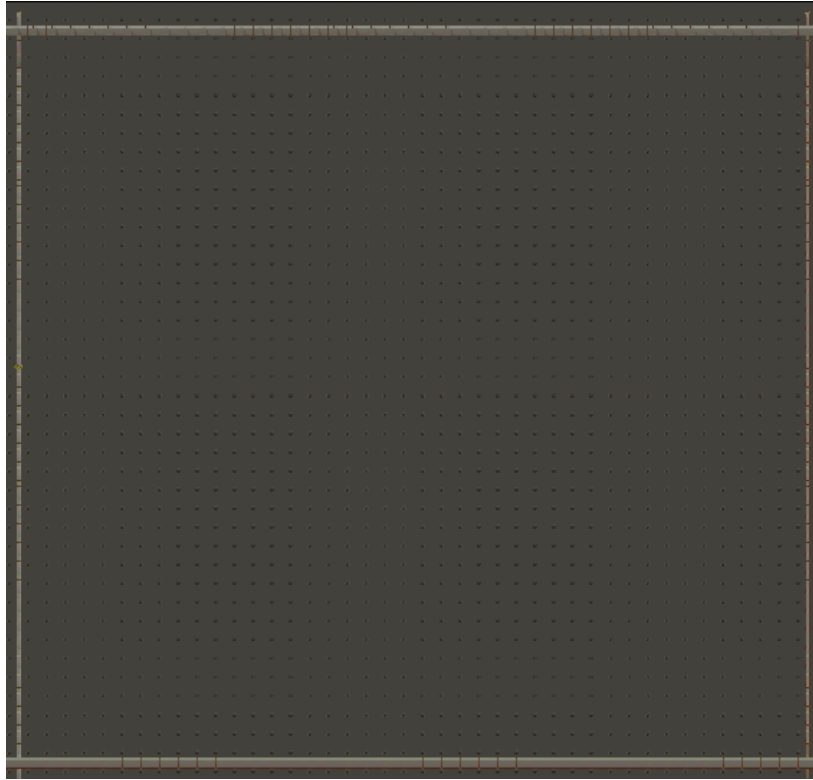
Figura 19 - Cenário 3 desenvolvido para o jogo



Fonte: Próprio autor

A figura 19 representa o terceiro cenário onde foi desenvolvido um labirinto escuro dificultando o avanço do personagem.

Figura 20 - Sala do BOSS, cenário 4 desenvolvido para o jogo



Fonte: Próprio autor

A figura 20 corresponde a sala do BOSS que é uma área aberta sem obstáculos onde o jogador terá contato com o SCRUM-BUG.

É possível notar que os cenários de 1 a 3 possuem um portal de teleporte, e está vinculado a cada portal de teleporte o *script* Portais, sendo ativado toda vez que o personagem principal passar por cima do portal de teleporte, esse script controla a abertura da tela de desafios. A figura 21 a seguir apresenta trecho do código do script Portais.

Figura 21 - Trecho de código do script Portais

```

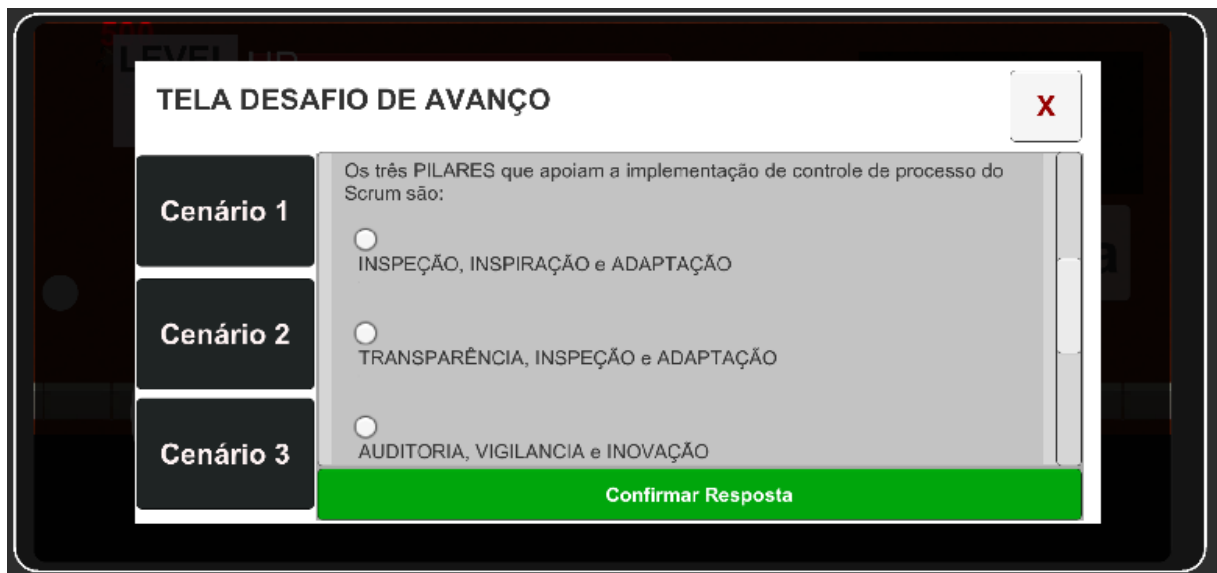
23 void OnTriggerEnter2D(Collider2D collider)
24 {
25     if(collider.gameObject.tag == "Player")
26     {
27         if(Target.position.x >= 22f && Target.position.x <= 26f)
28         {
29             if(float.Parse(honraScore_1.text) >= 30f && int.Parse(PlayerLevel.text) >= 3)
30                 TelaDesafio.SetActive(true);
31         }
32     }
33 
```

Fonte: Próprio autor

Na figura 21 é apresentado o trecho do código responsável por ativar a tela de desafio para o personagem Draco. O método `OnTriggerEnter2D` linha (23), é executado sempre que qualquer personagem fizer qualquer contato com o portal de teleporte, em seguida é

executado o condicional `if` (linha 25) que irá verificar através da `tag` de identificação "Player" se o personagem que realizou o contato é o personagem principal, caso seja uma condição válida é verificado qual portal de teleporte foi ativado, isso por meio da condição `if` (linha 27) que verifica as coordenadas X do personagem. Assim, é executado um último condicional (linha 29) que irá validar se o personagem possui os pontos de honra e o *level* necessário para abrir a tela de desafios do cenário. Na figura 22 a seguir é apresentada a tela de desafios desenvolvida para o jogo.

Figura 22 - Tela desafio de avanço desenvolvida para o jogo



Fonte: Próprio autor

A figura 22 demonstra a tela desafio de avanço onde é apresentado para o jogador perguntas relacionadas aos conceitos Scrum aprendidos no cenário. Ao responder o desafio e clicar em "Confirmar Resposta" é executado o sistema de desafios presente no *script* Desafios. A figura 23 a seguir apresenta trecho do código presente no script Desafios.

Figura 23 - Código que verifica a quantidade de acertos

```

273     if(Player.cenario == 1)
274     {
275
276         if(RespostasCorretas <= 1)
277             honraScore_1.text = (int.Parse(honraScore_1.text) - 30) +"";
278         else if(Portais.Target.position.y >= -1.2f && Portais.Target.position.y <= 1.6f)
279         {
280             if(RespostasCorretas == 2)
281                 honraScore_1.text = (int.Parse(honraScore_1.text) - 20) +"";
282
283             if(RespostasCorretas == 3)
284                 honraScore_1.text = (int.Parse(honraScore_1.text) - 15) +"";
285
286             Portais.Target.position = new Vector3(49f,0f,0);
287             Player.cenario = 2;
288         }
289     }

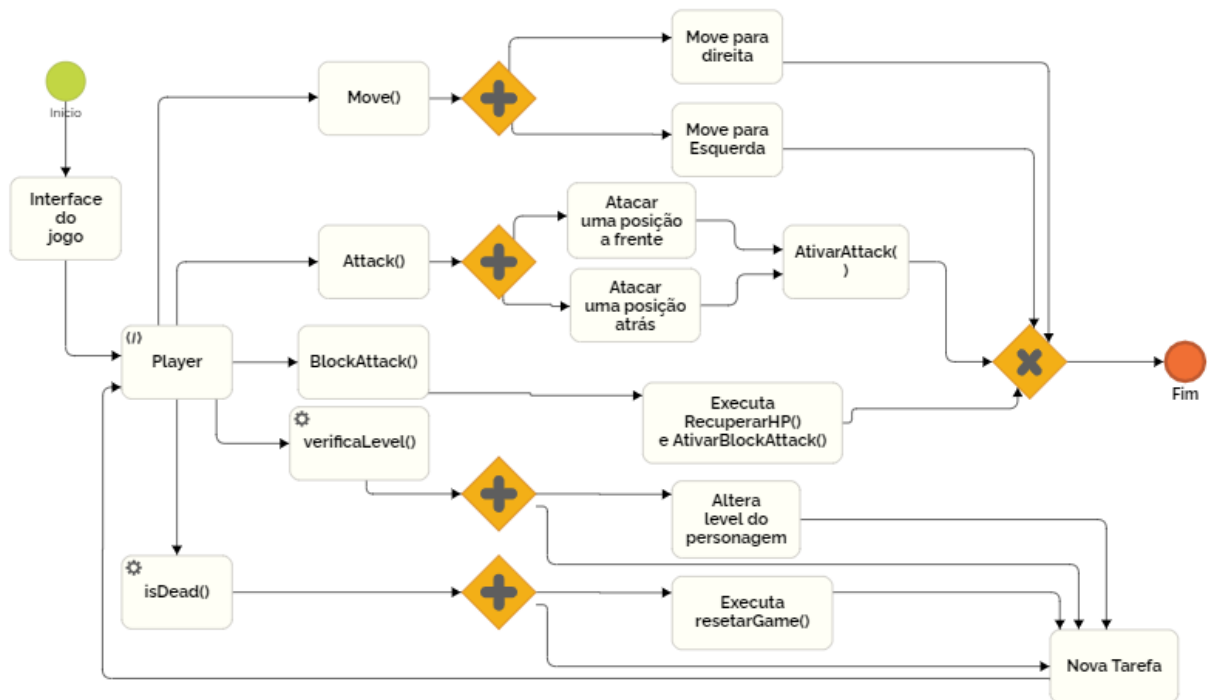
```

Fonte: Próprio autor

A figura 23 apresenta trecho do código do sistema de honra, onde é descontado pontos de honra do jogador de acordo com a quantidade de respostas corretas obtidas no desafio. A primeira validação do código é um condicional `if` que verifica em qual cenário o jogador está realizando o desafio, através do parâmetro `Player.cenario` (linha 273).

Caso o desafio tenha sido realizado no cenário 1 em seguida é verificado quantas respostas corretas o jogador obteve, caso tenha acertado apenas uma questão do desafio, o sistema desconta todos os pontos de honra do cenário 1 e mantém o personagem no cenário 1. Caso as respostas corretas sejam mais de uma, o sistema envia o personagem para o cenário seguinte e realiza os descontos do ponto de honra correspondentes com a quantidade de acertos.

Portanto, nota-se que a interação do jogador com o jogo não depende apenas de componentes gráficos, mas também de sistemas que controlam cada ação executada dentro do jogo. Nesse sentido, foi elaborado um diagrama de fluxo para apresentar a correlação existente entre os sistemas de personagens, sendo ele principal ou NPC 's, com os sistemas gerais do jogo. Na figura 15 a seguir é apresentado o diagrama de fluxos do *script Player*, que é um arquivo de código contendo as instruções correspondente a dinâmica do personagem principal dentro do jogo.

Figura 24 - Diagrama de fluxo do *script* Player

Fonte: Próprio autor

Na figura 24 é possível observar as funcionalidades destinadas ao personagem principal do jogo, onde os métodos *Move()*, *Attack()* e *BlockAttack()* dependem de interação do jogador com a interface do jogo para serem executados, já os métodos *verificarLevel()* e *isDead()* são métodos automáticos executados pelo jogo. Esses métodos são responsáveis pela dinâmica do personagem principal. O fluxo e seus respectivos elementos são detalhados a seguir:

- **Início:** O script Player é executado sempre que o jogador entrar no jogo, e assim executa os métodos *verificarLevel()*, *isDead()* e a animação *Idle* que é a primeira animação do personagem principal e representa o estado de ocioso.
- **Move():** É o método responsável por controlar os movimentos do personagem principal, sendo executado sempre que o jogador interagir com o joystick presente na interface do jogo, esse método movimenta o personagem para a direita, para a esquerda, para cima e para baixo, além disso ele também realiza o controle da animação *Move* que representa o “andar” do personagem.
- **Attack():** método executado através da interação com o “botão AT” presente na interface do jogo, esse método que controla o ataque do personagem principal, fazendo com que os ataques sejam desferidos em posições alternadas sendo uma posição para frente ou uma posição atrás do último golpe realizado, isso se faz necessário pois o



ataque possui um componente da Unity que só é ativado quando houver colisões em posições diferentes.

Esse método também realiza o controle da animação *Attack* que representa o “golpe de espada” do personagem principal.

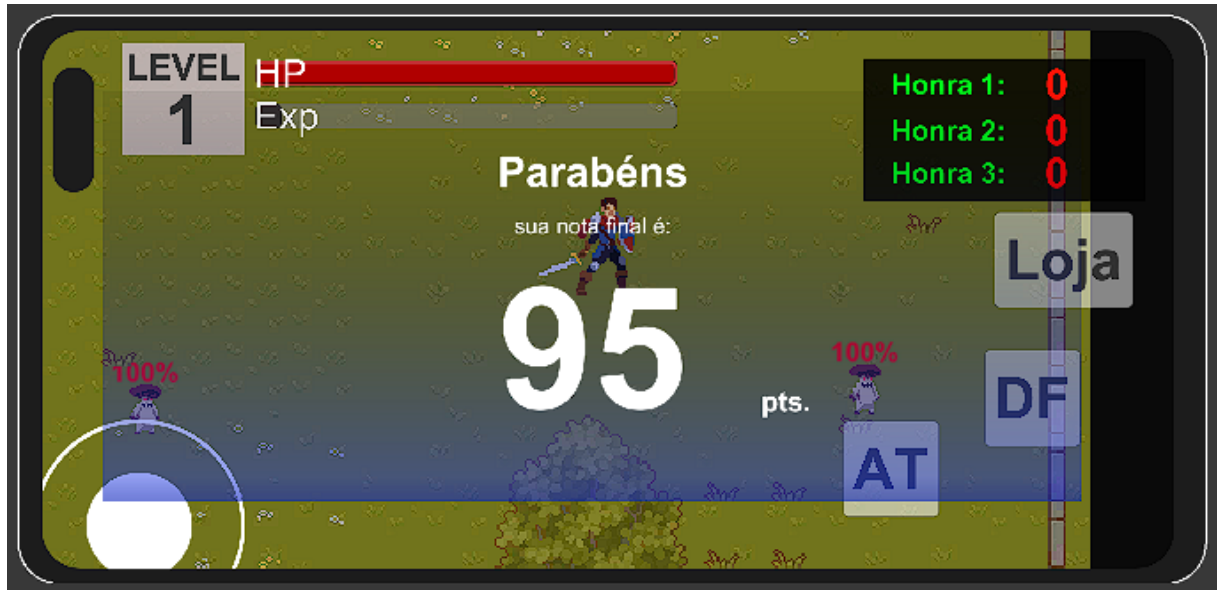
- ***BlockAttack()***: método executado sempre que houver interação do jogador com o “botão DF” presente na interface do jogo, esse método controla a defesa e a regeneração de HP do personagem. Ele controla a animação *Block* onde o personagem principal usa seu escudo para se defender dos ataques inimigos.
- ***isDead()***: esse método é responsável por reiniciar o jogo através da função *restarGame()* sempre que os pontos de vida/HP do personagem principal chegar a 0(zero), assim como os demais, este método controla a animação *Dead* que representa a “morte” do personagem principal. Esse método não requer interação do jogador com a interface do jogo para ser executado.
- ***verificarLevel()***: é um método que também não necessita de interação do jogador para ser executado. Sua função é verificar se o personagem principal possui pontos de experiência necessários para avançar para o próximo *level*.
- **Fim**: Os métodos *Move()*, *Attack()* e *BlockAttack()*, terão seus fluxos encerrados sempre que o jogador deixar de interagir com a interface do jogo, e assim o personagem retorna ao estado de ocioso executando a animação *Idle*.

#### 4.4.1 FEEDBACK EDUCACIONAL

O *feedback* educacional do jogo é apresentado em dois modelos sendo o primeiro em jogo, onde ao concluir o jogo será apresentado na interface do jogador a média geral obtida, conforme proposto na subseção 4.2.5 Desafios e Pontuações.

Para o segundo modelo de *feedback* educacional foi desenvolvido uma página web, onde ao realizar login o jogador terá acesso a um relatório contendo as seguintes informações: média do jogo; média de acertos e erros geral; médias de acertos e erros por cenário; e todas as respostas dos desafios realizados dentro do jogo. A figura 25 a seguir demonstra o *feedback* apresentado em jogo.

Figura 25 - Feedback educacional em jogo



Fonte: Próprio autor

Na figura 25 é apresentado ao jogador sua nota final que consiste na média dos pontos obtidos através dos desafios que realizou nos cenários 1,2 e 3 e na sala do BOSS, Como jogos este modelo de apresentação foi adotado devido a limitação de de tela que um jogo mobile proporciona. Contudo, no segundo modelo de feedback desenvolvido foi possível apresentar mais informações. A figura 26 a seguir demonstra o feedback educacional apresentado na página web.

Figura 26 - Página web, tela de feedback educacional



Fonte: Próprio autor

Na figura 26 é possível observar os seguintes elementos:

- **Média do jogo:** consiste na pontuação final adquirida ao concluir o jogo conforme apresentado na subseção 4.2.5 Desafios e Pontuações.
- **Média geral:** representa a média de acertos e erros de todas as vezes em que o jogador realizou um desafio dentro do jogo, independente de ter concluído ou não o jogo.
- **Média por cenário:** é a média de todas as vezes em que o jogador realizou um desafio em um respectivo cenário.
- **Minhas respostas:** corresponde a todos os desafios respondidos dentro do jogo, assim como as respostas apresentadas pelo jogador.

Desta forma o jogador pode acompanhar seu desempenho em um determinado cenário do jogo, e assim saber em qual conceito do Scrum precisa melhorar, além de acompanhar o histórico de suas respostas.

Para o funcionamento do *feedback* educacional em página web, foi necessário incluir ao jogo um sistema de cadastro e *login*, detalhados a seguir:

- **Sistema de Cadastro:** corresponde ao sistema integrado a “tela dados do jogador” apresentada na subseção 4.3. Esse sistema armazena os dados apresentados pelo jogador

em um banco de dados, por meio de conexão com a internet. A figura 27 a seguir demonstra a tela dados do jogador desenvolvida para o jogo.

Figura 27 - Tela dados do jogador desenvolvida

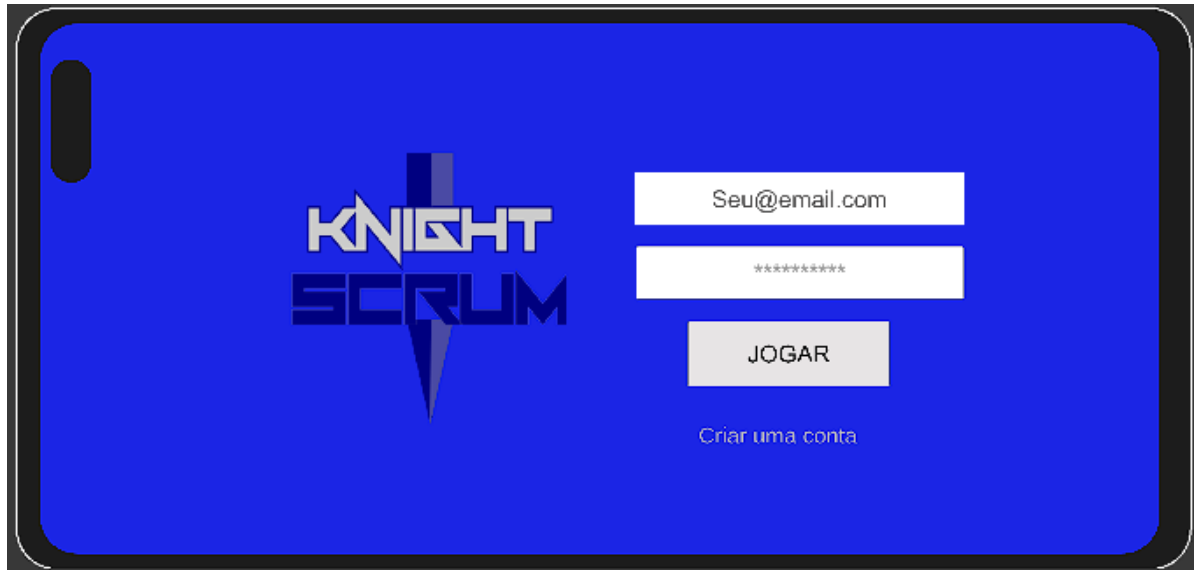


Fonte: Próprio autor

A figura 27 representa a tela dados do jogador onde é solicitado ao jogador nome completo, e-mail, curso e uma senha, o email e a senha serão utilizados no sistema de login. Ao clicar em cadastrar o sistema verifica se o email apresentado já está cadastrado, se não estiver ele armazena os dados na base de dados do jogo. Se o jogador clicar em “Voltar” o sistema apresenta a tela de login.

- **Sistema de Login:** é o sistema integrado a “tela de login”, através dessa tela o jogador informa seu e-mail e senha previamente cadastrados para poder ter acesso ao jogo. A figura XYY demonstra a tela de *login* desenvolvida para o jogo.

Figura 28 - Tela de login desenvolvida para o jogo



Fonte: Próprio autor

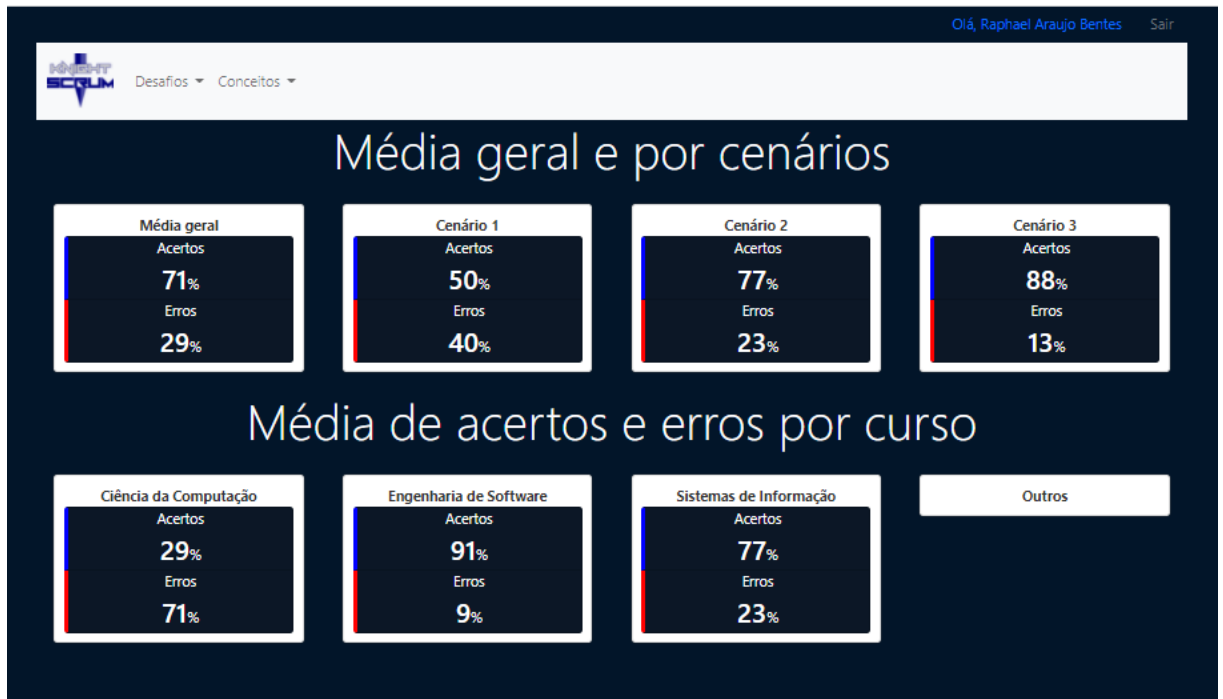
A figura 28 é representada a tela de login, onde é solicitado o e-mail e a senha do jogador, ao informar os dados e clicar em “Jogar” o sistema valida os dados, caso esteja cadastrado o jogo é iniciado. Ao clicar em criar uma conta, o sistema de cadastro é apresentado ao jogador.

#### 4.4.1.1 PAINEL ADMINISTRATIVO

Para este trabalho também foi desenvolvido um painel administrativo, por meio do qual um professor pode realizar a gestão dos conceitos Scrum abordados em cada cenário do jogo, assim como a gestão dos desafios de cada cenário.

Além disso, o painel conta com uma *dashboard* apresentando as médias de acertos e erros gerais dos desafios realizados em jogo, assim como as médias de acertos e erros dos cursos informados pelos jogadores. As figuras 29, 30, 31 e 32 representam os elementos que compõem o painel administrativo.

Figura 29 - Dashboard apresentada no painel administrativo



Fonte: Próprio autor

Na figura 29 é apresentada a tela dashboard disponível no painel administrativo, onde é apresentado a média geral de acertos e erros geral e por cenários, essas médias são compostas por todos os desafios já realizados dentro do jogo, além disso também é apresentado a média de acertos e erros por curso de formação, esses informados pelos jogadores ao realizarem o cadastro no jogo. A partir dessa tela, por meio do menu disponível é possível acessar as telas de gestão dos desafios e dos conceitos do jogo, conforme apresentado na figura 30 a seguir.

Figura 30 - Cadastro de desafios

The screenshot displays the 'Cadastro de desafio' (Challenge Registration) interface. At the top, a navigation bar includes a logo and a menu with 'Desafios' and 'Conceitos' dropdowns. A dropdown menu is open under 'Desafios', showing 'Cadastrar' and 'Listar' options. The main content area is titled 'Cadastro de desafio'. The form is divided into several sections: 'Pergunta' (Question) with a text input field, 'Cenário' (Scenario) with a checkbox, 'Alternativas' (Alternatives) with four text input fields labeled 'Alternativa 1' through 'Alternativa 4', and 'Gabarito' (Answer Key) with four radio buttons corresponding to the alternatives. A blue 'Cadastrar' button is located at the bottom of the form. The top right of the page shows the user's name 'Olá, Raphael Araujo Bentes' and a 'Sair' (Logout) link.

Fonte: Próprio autor

Conforme apresentado na figura 30, ao clicar no item “Desafios” do menu e selecionar a opção “Cadastrar”, será apresentado o formulário de cadastro de desafio, onde é possível adicionar novos desafios ao jogo preenchendo a descrição da pergunta, selecionando o cenário a que pertence, as alternativas para resposta e o gabarito onde é informado qual alternativa é a correta.

Como regra, cada desafio deve possuir no mínimo duas alternativas e no máximo quatro alternativas. Portanto, ao clicar em cadastrar o desafio é armazenado na base de dados ficando disponível no jogo. A figura 31 a seguir apresenta a lista dos desafios cadastrados.

Figura 31 - Lista de desafios cadastrados

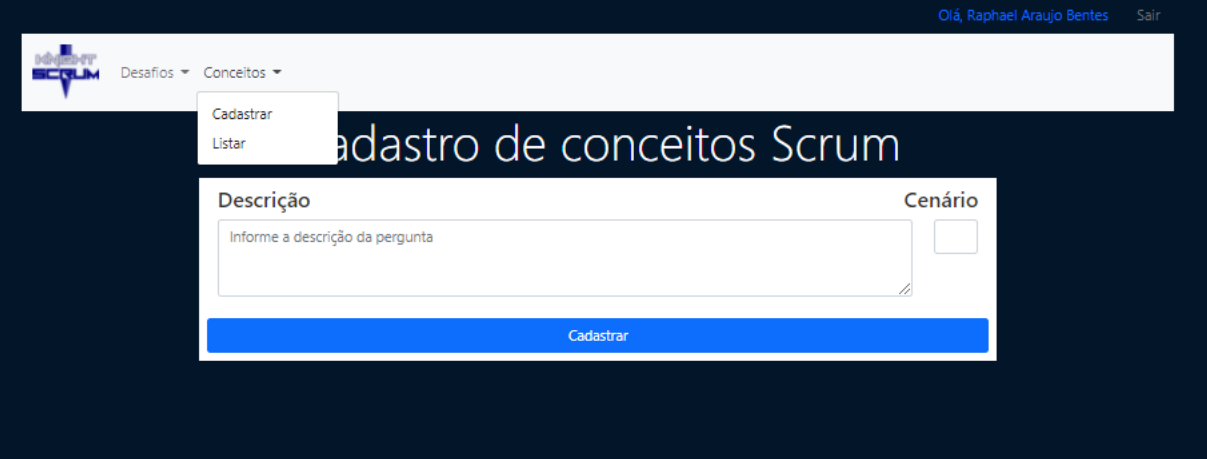
Cenário	Descrição da pergunta
1	Os cinco VALORES definidos para o Scrum são: <ul style="list-style-type: none"> <li>comprometimento, humildade, esforço, transparência e orgulho</li> <li>comprometimento, coragem, foco, transparência e respeito</li> <li>comprometimento, garra, foco, transparência e respeito</li> <li>comprometimento, humildade, foco, empatia e respeito</li> </ul>
1	Os três PILARES que apoiam a implementação de controle de processo do Scrum são: <ul style="list-style-type: none"> <li>inspeção, inspiração e adaptação</li> <li>transparência, inspeção e adaptação</li> <li>auditoria, vigilância e inovação</li> </ul>
1	Os aspectos significativos do processo devem estar visíveis aos responsáveis pelos resultados, representa o Pilar da <ul style="list-style-type: none"> <li>inovação</li> <li>adaptação</li> <li>transparência</li> <li>inspeção</li> </ul>
1	Os membros do Time Scrum respeitam-se uns aos outros para serem pessoas capazes e independentes, da significado ao Valor do(a) <ul style="list-style-type: none"> <li>comprometimento</li> <li>cortesia</li> <li>simpatia</li> <li>respeito</li> </ul>
2	(CESPE, 2012). Uma das atribuições do product owner, papel definido pelo Scrum, é a responsabilidade pelo gerenciamento do backlog. Tal atribuição pode ser delegada aos outros membros do time Scrum. <ul style="list-style-type: none"> <li>certo</li> <li>errado</li> </ul>

Fonte: Próprio autor

Na figura 31 é possível observar a lista de desafios cadastrados no jogo, apresentada ao clicar no item “Desafios” do menu e selecionar a opção “Listar”. A lista apresentada contém o cenário e a descrição da pergunta, assim como as alternativas de cada pergunta, sendo que a alternativa com gabarito correto é apresentada na cor verde. A figura 32 a seguir apresenta o formulário de cadastro de conceitos Scrum.



Figura 32 - Cadastro de conceitos

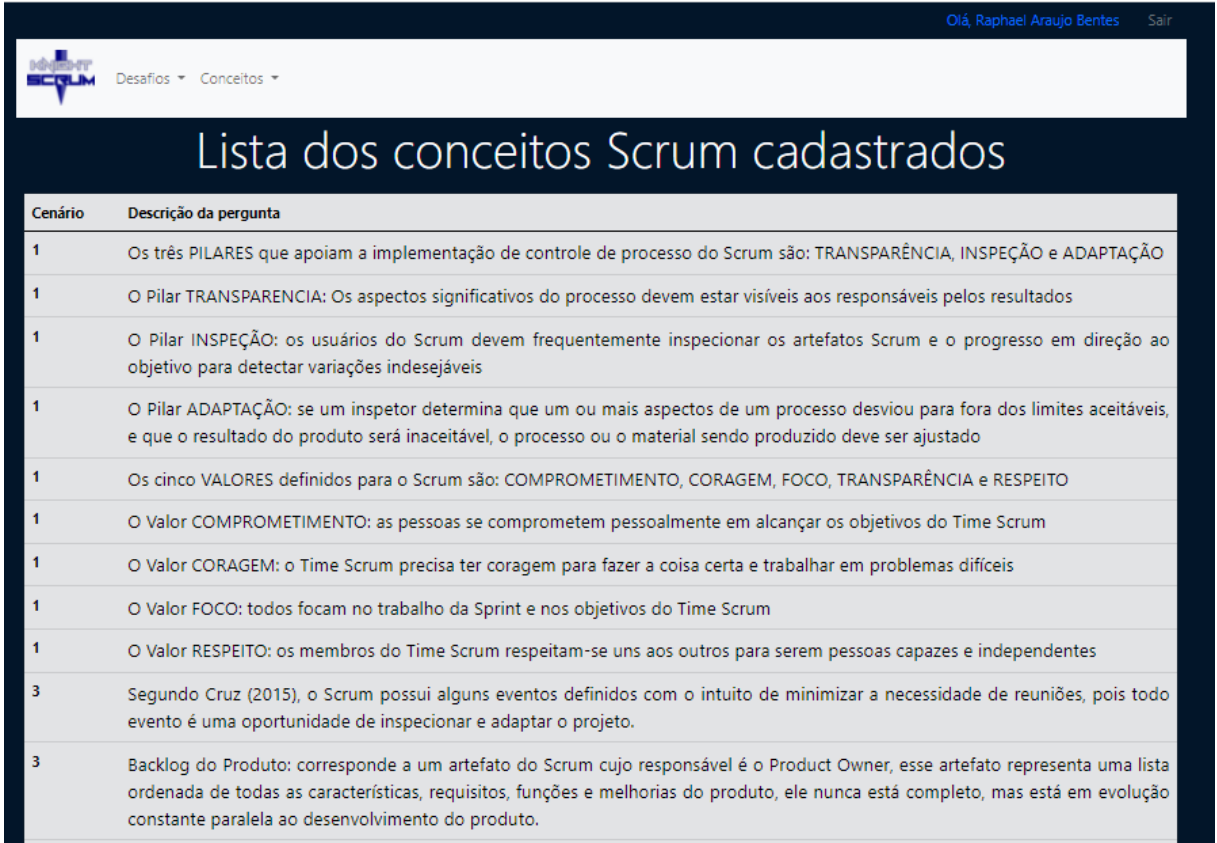


The screenshot shows a web application interface for registering Scrum concepts. At the top right, the user is logged in as 'Olá, Raphael Araujo Bentes' with a 'Sair' (Logout) link. The navigation menu includes 'Desafios' and 'Conceitos'. A dropdown menu under 'Conceitos' is open, showing 'Cadastrar' (Register) and 'Listar' (List). The main heading is 'Cadastro de conceitos Scrum'. The form contains two fields: 'Descrição' (Description) with a placeholder 'Informe a descrição da pergunta' and 'Cenário' (Scenario) with an empty input box. A blue 'Cadastrar' button is at the bottom of the form.

Fonte: Próprio autor

A figura 32 demonstra o formulário de cadastro de conceitos Scrum, apresentado ao clicar no item “Conceitos” e selecionar a opção “Cadastrar”. O formulário é composto pela descrição do conceito a ser cadastrado e o cenário do jogo a que pertence. Ao preencher o formulário e clicar no botão “Cadastrar”, o conceito é armazenado na base de dados ficando disponível no jogo. A figura 33 a seguir apresenta a lista de conceitos cadastrados.

Figura 33 - Lista de conceitos cadastrados



Cenário	Descrição da pergunta
1	Os três PILARES que apoiam a implementação de controle de processo do Scrum são: TRANSPARÊNCIA, INSPEÇÃO e ADAPTAÇÃO
1	O Pilar TRANSPARENCIA: Os aspectos significativos do processo devem estar visíveis aos responsáveis pelos resultados
1	O Pilar INSPEÇÃO: os usuários do Scrum devem frequentemente inspecionar os artefatos Scrum e o progresso em direção ao objetivo para detectar variações indesejáveis
1	O Pilar ADAPTAÇÃO: se um inspetor determina que um ou mais aspectos de um processo desviou para fora dos limites aceitáveis, e que o resultado do produto será inaceitável, o processo ou o material sendo produzido deve ser ajustado
1	Os cinco VALORES definidos para o Scrum são: COMPROMETIMENTO, CORAGEM, FOCO, TRANSPARÊNCIA e RESPEITO
1	O Valor COMPROMETIMENTO: as pessoas se comprometem pessoalmente em alcançar os objetivos do Time Scrum
1	O Valor CORAGEM: o Time Scrum precisa ter coragem para fazer a coisa certa e trabalhar em problemas difíceis
1	O Valor FOCO: todos focam no trabalho da Sprint e nos objetivos do Time Scrum
1	O Valor RESPEITO: os membros do Time Scrum respeitam-se uns aos outros para serem pessoas capazes e independentes
3	Segundo Cruz (2015), o Scrum possui alguns eventos definidos com o intuito de minimizar a necessidade de reuniões, pois todo evento é uma oportunidade de inspecionar e adaptar o projeto.
3	Backlog do Produto: corresponde a um artefato do Scrum cujo responsável é o Product Owner, esse artefato representa uma lista ordenada de todas as características, requisitos, funções e melhorias do produto, ele nunca está completo, mas está em evolução constante paralela ao desenvolvimento do produto.

A figura 33 demonstra a lista de conceitos Scrum cadastrados apresentada ao clicar no item Conceitos do menu e selecionar a opção “Listar”. Essa lista contém o cenário a que pertence o conceito e a descrição do conceito.

Portanto, com o desenvolvimento do painel administrativo foi possível apresentar por meio das médias, os resultados obtidos com os desafios propostos para o jogo. Assim como tornar os conceitos e os desafios dinâmicos, onde um professor pode controlar quais conceitos serão abordados e quais desafios serão apresentados para o jogador.

## 5 CONSIDERAÇÕES FINAIS

A BNCC indica que a interação com a tecnologia é uma das fontes que estimulam a curiosidade (BRASIL, 2018). Desse modo, percebe-se que se utilizada de maneira correta, a tecnologia pode atuar significativamente no desenvolvimento e aprendizagem dos alunos. Veen e Vrakking (2009) afirmam que os jogos digitais também promovem o desenvolvimento de várias habilidades metacognitivas na aprendizagem, uma vez que levam ao desenvolvimento de estratégias e soluções para as dificuldades. Segundo Frosi (2010), “os jogos devem possuir certas características para que não percam sua identidade enquanto “game”, preservando, dessa forma, a atratividade e o interesse pelo jogo”.

Assim, este trabalho teve como objetivo desenvolver um jogo mobile capaz de auxiliar no estudo e na prática dos conceitos da Metodologia de Desenvolvimento Ágil Scrum. Para alcançar o resultado, foram seguidas as etapas presentes na seção 3.2 onde foi realizado a análise dos conceitos necessários para o desenvolvimento de um jogo educacional. Através desta análise foi possível identificar artigos acadêmicos relacionados ao desenvolvimento de jogos educacionais. Portanto, por meio desses artigos foi possível conhecer e compreender técnicas e ferramentas, utilizadas para o desenvolvimento de jogos educacionais.

Para o desenvolvimento do jogo apresentado neste trabalho foram utilizadas duas ferramentas sendo a primeira o *Adobe XD*, onde foi realizado a prototipagem das telas do jogo; e a segunda a *Unity3D* que é uma ferramenta que fornece diversos elementos gráficos para o desenvolvimento de jogos e conta com um acervo de componentes gratuitos como *sprites* de personagens, *sprites* de texturas entre outros.

Como resultado obteve-se o jogo *Knight Scrum*, um jogo do gênero RPG com elementos do subgênero *RogueLikes* em formato 2D, no qual o jogador controla um personagem principal, onde no decorrer do jogo terá contato com os conceitos da Metodologia de Desenvolvimento Ágil Scrum. Assim adquirindo conhecimento sobre os pilares do scrum, os valores do scrum, o time de desenvolvimento, os artefatos e eventos do Scrum.

O objetivo principal do jogo é o aprendizado dos conceitos da Metodologia de Desenvolvimento Ágil Scrum. No decorrer da implementação foi estudado como o jogador teria acesso aos conceitos Scrum referente ao cenário em que estivesse presente, e em como avaliar o aprendizado do jogador, portanto para isso foram implementados os desafios de avanço de cada cenário. O jogo está disponível para *smartphones* com sistema android, sendo possível realizar seu *download* através do site [www.knightscrums.com.br](http://www.knightscrums.com.br).

O jogo está modulado para funcionar com um administrador e vários jogadores, ou seja, atualmente apenas um professor capacitado, possui acesso e poderá alimentar a base de dados

do jogo com os conceitos Scrum a serem abordados nos cenários e os desafios a serem realizados pelos jogadores. Ao fazer isso, o conteúdo cadastrado estará disponível a todos que o tiverem instalado em seu *smartphone*.

Portando, como trabalhos futuros, são apresentadas as seguintes sugestões que poderão melhorar a experiência didática, em relação aluno x professor, segue:

- **Sistema de turmas:** a proposta desse sistema é fornecer ao professor a possibilidade de criar suas turmas, onde terá acesso apenas aos jogadores que são seus alunos. Para esse sistema funcionar é necessário adaptações na tela de cadastro do jogo e no painel administrativo, sendo elas:
  - Painel administrativo: sistema de cadastro de professores, sistema de código de turma e sistema de gestão de turma.
  - Jogo: atribuir ao formulário de cadastro, um campo o código da turma, ou seja, será permitido cadastro no jogo apenas com código de turma.
- **Sprites:** o jogo foi desenvolvido com componentes de licença pública, com isso a distribuição do jogo acaba sendo limitada. Portanto, como melhoria seria necessário desenvolver sprites de autoria própria ou adquirir licenças sem limites de uso.
- **Dashboard:** analisar e incrementar ao painel administrativo do jogo, elementos que forneçam ao professor relatórios de desempenhos de seus alunos.

Com essas sugestões, acredita-se que a dinâmica e os resultados educacionais propostos para o desenvolvimento deste trabalho ficarão ainda mais completos. Com isso, o ambiente desenvolvido não ficará limitado somente ao ensino dos conceitos Scrum, podendo trabalhar disciplinas que possuem como foco elementos conceituais.

## 6 REFERENCIAL

ARAUJO, Victor, F. A.; GUIMARÃES, Adolfo, P.; MENEZES, Lucas, G. R. Libras Racer: Protótipo de um Jogo Sério para o Ensino da Língua Brasileira de Sinais (Libras). In: XVI Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames, 2017). Disponível em <https://www.sbgames.org/sbgames2017/papers/JogosGraduacao/176689.pdf>. Acesso em 12 de mai de 2021.

BATTISTELLA, Paulo E. ENgAGED: processo de desenvolvimento de jogos para ensino em computação. 2016. 401 f. Tese (Doutorado) - Curso de Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2016. Disponível em: <https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/175816/345443.pdf?sequence=1&isAllowed=y>. Acesso em: 13 mai 2021.

BRASIL. Ministério da Educação. Base Nacional Comum Curricular. Brasília, 2018. Disponível em: [http://basenacionalcomum.mec.gov.br/images/BNCC\\_EI\\_EF\\_110518\\_-versaofinal\\_sit\\_e.pdf](http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_-versaofinal_sit_e.pdf). Acesso em 16 de abr de 2021.

CAMARGO, André Stangarlin de. JOGO DE RPG PARA ENSINAR SCRUM. 2013. 103 f. TCC (Graduação) - Curso de Ciências da Computação, Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis, 2013. Disponível em: <http://www.gqs.ufsc.br/files/2020/02/TCC-Andre-Camargo-vf.pdf>. Acesso em: 13 maio 2021.

FARDO, Marcelo L. A GAMIFICAÇÃO COMO ESTRATÉGIA PEDAGÓGICA: Estudo de elementos dos games aplicados em processos de ensino e aprendizagem. 2013. Disponível em: <https://repositorio.ucs.br/xmlui/bitstream/handle/11338/457/Dissertacao%20Marcelo%20Luis%20Fardo.pdf?sequence=1&isAllowed=y>. Acesso em: 21 de abr de 2021.

FLOOD, K. Game Unified Process (GUP). GameDev.net Articles. 2003. Disponível em: <http://www.gamedev.net/reference/articles/article1940.asp>. Acesso em: 13 de mai de 2021.

FROSI, F. O. Jogos Digitais no Contexto Escolar: desafios e possibilidades para a Prática Docente. In: Simpósio Brasileiro de Jogos e Entretenimento Digital, 2010. Disponível em : [www.sbgames.org/papers/sbgames10/culture/full/full13.pdf](http://www.sbgames.org/papers/sbgames10/culture/full/full13.pdf). Acesso em 08 de Ago de 2021.

GODOY, A.; BARBOSA, E.F. Game-scrum: an approach to agile game development. In: Simpósio Brasileiro de Jogos e Entretenimento Digital, 2010. Disponível em: [http://sbgames.org/papers/sbgames10/computing/short/Computing\\_short19.pdf](http://sbgames.org/papers/sbgames10/computing/short/Computing_short19.pdf). Acesso em: 13 de mai de 2021

LAUBISCH, A.; CLUA, E. Scrum4Games: Uma aplicação do Scrum para projetos de games focada em game design. In: IX Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames, 2010), Florianópolis. 2010. Disponível em: [http://www.sbgames.org/papers/sbgames10/artanddesign/Full\\_A&D\\_21.pdf](http://www.sbgames.org/papers/sbgames10/artanddesign/Full_A&D_21.pdf). Acesso em: 13 de mai de 2021.

METARED BRASIL. Avaliação das Competências Digitais dos Docentes do Ensino Superior Brasileiro. SEMESP, 2020. Disponível em :<https://www.semesp.org.br/wp-content/uploads/2021/04/Avaliacao-de-Competencias-Digitais-1.pdf>. Acesso em 02 de Ago de 2021.

SILVA, Leonardo F.; PINHEIRO, M. D.; SANTOS, R. R.; SCHIMIGUEL, J. O CRESCIMENTO DOS JOGOS NO MERCADO MOBILE E SUAS ACESSIBILIDADES. Revista Caribeña de Ciencias Sociales, Fevereiro, 2016. Disponível em: <https://www.eumed.net/rev/caribe/2016/02/jogos.zip>. Acesso em 20 de abr de 2021.

SÁ, A. L.; SILVA, E. F.; MACHADO, M. C. UTILIZAÇÃO DOS JOGOS DIGITAIS COMO AUXÍLIO NO PROCESSO ENSINO APRENDIZAGEM NA DISCIPLINA DE PORTUGUÊS, XV Encontro Virtual de Documentação em Software Livre e XII Congresso Internacional de Linguagem e Tecnologia Online. Disponível em: [http://www.periodicos.letras.ufmg.br/index.php/anais\\_linguagem\\_tecnologia/article/viewFile/15066/1125612225](http://www.periodicos.letras.ufmg.br/index.php/anais_linguagem_tecnologia/article/viewFile/15066/1125612225). Acesso em 17 abr. 2021

SANTOS, Adam M. I Blue It: Um Jogo Sério para auxiliar na Reabilitação Respiratória. In: XVII Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames, 2018). Disponível em <https://www.sbgames.org/sbgames2018/files/papers/ArtesDesignFull/188093.pdf>. Acesso em 12 de mai de 2021.

SENAC. Planejamento de jogos digitais para multiplataformas. Rio Grande do Sul, 2018. Disponível em [https://www.senacrs.com.br/cursos\\_rede/planejamento\\_de\\_jogos\\_digitais\\_para\\_multiplataformas/html/metodologias\\_desenvol\\_%20jogos/index.html#:~:text=O%20processo%20de%20de%20de,em%20conjunto%20com%20outras%20t%C3%A9cnicas.&text=Utiliza%20de%20os%20processos%20que%20se%20adequam%20%C3%A0%20metodologia%20escolhida](https://www.senacrs.com.br/cursos_rede/planejamento_de_jogos_digitais_para_multiplataformas/html/metodologias_desenvol_%20jogos/index.html#:~:text=O%20processo%20de%20de%20de%20de,em%20conjunto%20com%20outras%20t%C3%A9cnicas.&text=Utiliza%20de%20os%20processos%20que%20se%20adequam%20%C3%A0%20metodologia%20escolhida). Acesso em 12 de maio de 2021

SCHNEIDER, Marcelo F, SCRUM'ed: um jogo de RPG para ensinar Scrum, 2015. Disponível em: [https://softwarequalitygroup.paginas.ufsc.br/files/2020/02/TCCfinal\\_SCRUMed.pdf](https://softwarequalitygroup.paginas.ufsc.br/files/2020/02/TCCfinal_SCRUMed.pdf). Acesso em: 19 de abr de 2021

VEEN, W.; VRAKING, B. Homo Zappiens: educando na era digital. Trad. de Vinícius Figueira. Porto Alegre: Artmed, 2009. 141 p.

VILLELA, M. MOBA, RPG, MMORPG, FPS e mais: entenda significado dos gêneros de games. TECHTUDO, 2021. Disponível em: <https://www.techtudo.com.br/noticias/2021/03/moba-rpg-mmorpg-fps-e-mais-entenda-significado-dos-generos-de-games.ghtml>. Acesso em 20 de abr de 2021.