

# Scrumie: Scrum Teaching Agent Oriented Game

## Scrumie: Jogo Orientado a Agentes para Ensino de Scrum

Leonardo Lima Marinho<sup>1</sup>, Suelen Regina C. dos Santos<sup>1</sup>, Leonardo Andrade<sup>2</sup>, Bruna Costa Cons<sup>1</sup>, Marcelo Schots<sup>1,2</sup>, Vera Maria B. Werneck<sup>1,2</sup>

**Abstract:** The use of agile methods has become essential in software development at the present time. Among the existing methods, Scrum is one of the major ones, and is used to manage projects in companies, even outside of the scope of software systems development. Considering the relevance of this subject and the success usually obtained in learning through educational games, Scrumie was proposed to teach the management of Scrum projects. Scrumie applies intelligence in multiagent architecture being developed with Agile Passi an agent oriented methodology. This paper contains a proposal, modeling, implementation and evaluation of the Scrumie game.

**Keywords:** Scrum — Serious Game — Multiagent System

**Resumo:** O uso de métodos ágeis se tornou imprescindível no desenvolvimento de software na atualidade. Dentre os métodos existentes, o Scrum é um dos principais utilizados para gerenciar projetos em empresas, mesmo fora do escopo de desenvolvimento de sistemas. Considerando a relevância deste assunto e o sucesso geralmente obtido no aprendizado por meio de jogos educacionais, foi proposto Scrumie para ensinar o gerenciamento de projetos Scrum. Scrumie aplica a inteligência numa arquitetura multiagentes, sendo desenvolvido com a metodologia orientada a agentes Agile Passi. Este artigo apresenta a proposta, modelagem, implementação e avaliação do jogo Scrumie.

**Palavras-Chave:** Scrum — Jogos — Sistemas Multiagentes

<sup>1</sup> Programa de Mestrado em Ciências Computacionais, Universidade do Estado do Rio de Janeiro, Brasil

<sup>2</sup> Departamento de Informática e Ciência da Computação, Universidade do Estado do Rio de Janeiro, Brasil

\*Corresponding author: vera@ime.uerj.br

DOI: <http://dx.doi.org/10.22456/2175-2745.98203> • Received: 13/11/2019 • Accepted: 08/03/2020

CC BY-NC-ND 4.0 - This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

## 1. Introdução

O modelo em cascata (*waterfall*), caracterizado por exigir especificações do sistema bem definidas e processos de trabalho longos e sequenciais [1] tem sido considerado um padrão para o desenvolvimento de software. Com a crescente complexidade dos sistemas e uma maior busca por eficiência nas empresas, este modelo começou a ser considerado inadequado em projetos de pequeno ou médio porte, ou que possuam requisitos dinâmicos. Portanto, surgiu a necessidade de incluir conceitos baseados no manifesto ágil [2], com etapas reduzidas e mais curtas, nas quais há entrega incremental para o cliente por meio de versões do sistema.

Os métodos ágeis possuem maior foco na entrega de código do que na documentação, havendo maior integração da equipe, interação com o cliente e aceitação para mudanças nas funcionalidades do sistema. Como o Scrum é o *framework* ágil mais utilizado no mercado [3], considerou-se relevante buscar a disseminação e aprendizado de seus conceitos.

Entre as formas de ensino mais eficazes, pode-se citar o uso de jogos educativos, pois, segundo Grando [4], estes proporcionam uma participação ativa de aprendizado pelo jogador, de forma que conceitos de difícil compreensão possam ser aprendidos e sedimentados. Ademais, os jogos permitem a exploração do conhecimento de forma prática e divertida, através do aspecto lúdico [5].

Neste sentido, o presente trabalho propõe o jogo Scrumie<sup>1</sup>, que foi definido com base no mecanismo proposto pelo jogo 2TScrum [6], um serious game de tabuleiro que simula o desenvolvimento de um software utilizando o *framework* ágil Scrum. Ao analisar o jogo 2TScrum identificou-se sua limitação separando as atividades entre os participantes da equipe Scrum, o que pode ser melhor abordado utilizando a arquitetura multiagentes proposta no Scrumie. No contexto de ensino-aprendizagem, o emprego dos agentes tem como propósito gerar maior qualidade e flexibilidade, tanto do ponto de vista do aluno quanto do professor. Além disso, a adaptação

<sup>1</sup>Disponível em: <https://scrumie-app.herokuapp.com/>

do jogo usando agentes pôde torná-lo mais acessível e prático, por não exigir o uso de um tabuleiro e peças físicas. O uso de agentes viabilizou a inclusão de um conceito inteligente, em que um agente provê auxílio a jogadores que precisem de dicas do jogo.

Além da introdução, este artigo é composto por mais 6 seções. A seção 2 aborda o método ágil Scrum, como este funciona, como a equipe é formada e quais são os seus eventos. A seção 3 apresenta o jogo Scrumie, definindo o modo de funcionamento e suas regras. A seção 4 descreve a metodologia utilizada no processo de desenvolvimento deste trabalho, o Agile PASSI, além de listar as etapas realizadas neste contexto, mostrando em seguida a modelagem realizada no desenvolvimento do jogo, enquanto a seção 5 relata a implementação do jogo. Na seção 6, a avaliação do Scrumie é apresentada, sendo aplicada em dois momentos diferentes. Por fim, a seção 7 apresenta as considerações finais.

## 2. Scrum

O Scrum é um *framework* de gestão e planejamento de projetos que segue os ideais ágeis. Ele recomenda a auto-organização da equipe, envolvimento constante do cliente no projeto, prazos de entrega curtos e flexibilidade de adaptação a mudanças.

As atividades básicas em um processo Scrum são: pré-planejamento, desenvolvimento e pós-planejamento. No pré-planejamento, as funcionalidades do sistema são relatadas em um documento chamado *backlog*. Cada requisito possui uma prioridade e um custo para ser implementado, e isto também é definido nesta fase, além dos riscos do projeto, ferramentas que serão utilizadas, definição da equipe e proposta de uma arquitetura de desenvolvimento. A fase de desenvolvimento é iterativa, na qual ocorrem as *sprints* (unidades de tempo de um mês ou menos que compartimentalizam o trabalho de um incremento), onde se observa e controla possíveis mudanças nas variáveis já estabelecidas previamente. Já a fase de pós-planejamento engloba reuniões realizadas para avaliar o progresso do projeto, testes finais e integração do sistema [7].

A equipe Scrum é formada pelos seguintes papéis: *Product Owner*, o responsável pelo gerenciamento do *backlog* do produto; o *Scrum Master*, que garante a compreensão do Scrum pela equipe, além de resolver problemas que impedem o sucesso da *sprint*; e a equipe de desenvolvimento, responsável pela criação dos incrementos utilizáveis do produto.

O Scrum define a realização de reuniões para manter a equipe integrada junto ao cliente. A Reunião de Planejamento da *sprint* possui o intuito de definir os itens a serem considerados na *sprint* atual. A Reunião Diária é uma reunião curta, realizada para acompanhar as atividades atuais e planejar as atividades do dia. Há também duas reuniões mais informais, a Revisão e a Retrospectiva da *Sprint*, com o objetivo de apresentar e avaliar os resultados obtidos ao fim da *sprint*. A partir das reuniões, artefatos são gerados, como o *Backlog* do Produto – uma lista de requisitos do sistema – e o *Backlog* da

*Sprint*, que relata as funcionalidades selecionadas do *backlog* do produto que precisarão ser desenvolvidas na *sprint* atual [6].

## 3. O jogo Scrumie

Scrumie possui um mecanismo de jogo semelhante ao 2TScrum, no qual um jogador deve seguir um fluxo que o permite gerenciar projetos de software, aplicando seus conhecimentos teóricos sobre Scrum e estimulando novos conhecimentos. Para isto, o jogo simula situações cotidianas de desenvolvimento de software, motivando o jogador a ter autonomia para tomar decisões a respeito de acontecimentos no projeto que devem ser gerenciados [6].

Com o objetivo de instigar o aprendizado no jogo, o jogador precisa finalizar o desenvolvimento do projeto dentro do prazo e orçamento estabelecidos pelo cliente no início da partida. No decorrer do jogo, o desempenho do jogador é avaliado para que dicas possam ser dadas a ele, auxiliando-o no gerenciamento do projeto.

Igualmente ao 2TScrum, o Scrumie apresenta os mesmos elementos que contribuem para a interação do jogador com o jogo: a carta do cliente, as cartas de *backlog*, a carta de validação do *backlog*, as cartas com perfil do desenvolvedor, as cartas surpresa e as cartas de eventos [6].

Os componentes que integram o jogo Scrumie são representados por quase todos os papéis da equipe Scrum, sendo o jogador responsável pelo papel do Gerente de Projeto e do *Product Owner*.

### 3.1 Partida do Jogo

As etapas do jogo são baseadas em Brito e Vieira [6]. A seguir, são detalhadas as etapas do fluxo principal que o jogador percorre durante uma partida.

No início ao jogo, é apresentado para o jogador a seguinte narrativa [6]: “Um novo projeto chega à empresa e você é alocado para gerenciá-lo utilizando o *framework* Scrum. Este projeto se trata de um sistema web para uma biblioteca, com um orçamento e prazo fixos que você deve cumprir, mas no decorrer do tempo há diversos acontecimentos que você deve gerenciar. Além desses acontecimentos, durante a construção do produto, o seu cliente acaba entendendo melhor o que necessita e mudanças podem acontecer. Você deve manter o projeto no prazo e no orçamento estabelecidos, mas será que é possível? Você aceita esse desafio?”.

Após a leitura da narrativa, o jogador inicia a partida. Em seguida, a carta do cliente é apresentada, informando o produto desejado, o orçamento e o prazo para o projeto. Posteriormente, o jogador segue para a criação do *backlog* do produto, onde são apresentados trinta e seis itens do *backlog*, exibidos com um identificador, descrição do item e estimativa de tempo para o desenvolvimento. O jogador deve escolher apenas os itens esclarecidos na carta do cliente.

Assim que o jogador montar o *backlog* do produto, Scrumie passa a calcular uma pontuação com base no prazo e no custo divulgados nos itens selecionados. Assim é mantido o

gerenciamento da pontuação do jogador ao longo do jogo, a partir dos acontecimentos e modificações presentes nas cartas das próximas etapas.

O jogador segue para a reunião com o cliente e, neste momento, o jogo passa a validar os itens do *backlog* do produto segundo a carta de validação do *backlog* concedida por ele. Scrumie ajusta o *backlog* do produto de acordo com essa carta, incluindo os itens dela que não estão nesse *backlog* e descartando os itens que não estão presentes nela. Se for verificado que o jogador acertou menos de oito itens no *backlog* do produto, ele aplica uma penalidade ao jogador. Caso contrário, oferece uma recompensa.

Depois de validado o *backlog* do produto, o jogador deve escolher três perfis de desenvolvedores para sua equipe de desenvolvimento. No caso, é exibido para o jogador cartas com os perfis dos desenvolvedores, informando a classificação, o custo monetário por *sprint* e o bônus de cada um, ou seja, sua produtividade. Nesse momento é verificada a quantidade de desenvolvedores escolhidos pelo jogador.

Com a equipe definida, o jogador faz o Planejamento da *Sprint*, onde cria o *backlog* da *sprint* escolhendo no máximo sete itens do *backlog* do produto. Scrumie verifica a quantidade de itens escolhidos para o *backlog* da *sprint*. Com a verificação feita e aprovada, assim que o *backlog* da *sprint* é criado, o prazo do projeto é atualizado, decrementando o total de tempo estimado para os itens do *backlog* da *sprint*.

Ainda no planejamento, duas cartas de Planejamento da *Sprint* são sorteadas e apresentadas para o jogador. Após a leitura das cartas, o jogador deve tomar decisões sobre os acontecimentos descritos nelas, considerando o prazo e o custo. Com as escolhas feitas, o desempenho do jogador é avaliado com base na média de desempenho aceitável de um jogador na partida. Caso seja verificado que a atuação do jogador atual não está boa, Scrumie prontamente apresenta dicas ao jogador em relação às cartas de Planejamento da *Sprint*. Depois, há um sorteio de duas cartas surpresa, que são apresentadas para o jogador e têm suas alterações aplicadas à partida.

Finalizada a etapa do planejamento, segue-se para a etapa de desenvolvimento, na qual duas cartas de Reunião Diária são sorteadas e apresentadas para o jogador. Novamente, o jogador deve considerar as informações nas cartas para tomar decisões. Feito isso, Scrumie faz mais uma avaliação de desempenho, repetindo o mesmo processo descrito anteriormente.

Ao ter gerenciado os acontecimentos presentes na Reunião Diária, o jogador segue para a Revisão da *Sprint*, onde uma carta de Revisão da *Sprint* é sorteada e apresentada ao jogador. Esta carta mostra dificuldades e soluções identificadas ao expor o incremento do sistema na reunião. Baseado na informação recebida, o jogador deve realizar uma escolha entre as opções oferecidas. Em seguida, o jogo atualiza essa nova configuração.

Depois da Reunião de Revisão, o jogador segue para a Retrospectiva da *Sprint*, onde há o sorteio de uma carta de

Retrospectiva da *Sprint* e esta é exibida para o jogador. Esta carta mostra acontecimentos positivos ou negativos da *sprint* que devem ser lidados pela equipe na próxima *sprint*. Da mesma forma que antes, o jogador deve tomar uma decisão e o Scrumie irá avaliá-la, tomando as medidas necessárias.

Finalmente, o jogo valida se todos os itens do *backlog* do produto foram desenvolvidos. Caso sobre itens no *backlog* do produto, o jogo ainda não pode ser finalizado. Este comunica ao jogador sobre o retorno para o Planejamento da *Sprint*, pois o produto não foi concluído, iniciando mais uma *sprint*. Caso contrário, Scrumie informa para o jogador que o jogo foi finalizado. Em seguida, são apresentadas ao jogador sua pontuação final e sua posição no ranking. O ranking indica um feedback de desempenho para o jogador, levando em consideração o prazo e o orçamento resultantes no final do projeto em relação aos outros jogadores. Por último, é atualizada a pontuação média de desempenho de um jogador com a pontuação final da partida.

## 4. Modelagem do sistema

Agile PASSI é a metodologia ágil orientada a agentes utilizada no desenvolvimento do jogo Scrumie. Ela surgiu a partir da metodologia PASSI, criada inicialmente para desenvolver um sistema robótico, incluindo processos ágeis oriundos do método *Extreme Programming*, onde há a liberação de versões iterativas [8].

É uma metodologia orientada a código, que valoriza mais o desenvolvimento do código do que a criação da documentação do mesmo. Há a identificação de agentes como um conjunto de funcionalidades expressas em casos de uso e a utilização de ontologias para descrever o domínio do sistema [9]. Os processos do Agile PASSI se resumem a quatro estágios: Requisitos, Sociedade, Código e Testes. A maioria das etapas de desenvolvimento são auxiliadas por automatizações e reutilização de código, por meio do *framework* disponibilizado, *Agent Factory*. Ele realiza um trabalho de engenharia reversa, construindo automaticamente um esboço das classes dos agentes a partir dos diagramas que os identificam [8], [10].

Originalmente, a metodologia recomenda a reutilização de padrões no código através da ferramenta *Agent Factory*, além da criação de ontologias e de histórias de usuário ao planejar as funcionalidades do sistema. No entanto, estas especificidades não foram utilizadas no desenvolvimento do trabalho.

A princípio, foi criado um fluxo principal do jogo, como uma lista de atividades, posteriormente utilizado para a criação dos diagramas. Algumas adaptações foram realizadas em relação às recomendações do Agile PASSI: manteve-se a sugestão de criação do diagrama de casos de uso para fase de Requisitos, enquanto os diagramas de identificação de agentes e o diagrama de identificação de papéis foram definidos na fase de Sociedade. Os estereótipos usados vêm do padrão UML [11].

Nesta etapa, também foi elaborada uma lista contendo as principais regras de negócio do jogo. Isso porque estas

regras detalham as funcionalidades particulares do software presentes nos diagramas, para satisfazer o cliente e o objetivo do negócio [12].

#### 4.1 Diagramas de Caso de Uso

Esses diagramas auxiliam no levantamento dos requisitos funcionais do sistema Scrumie. A Figura 1 apresenta os diagramas de caso de uso que descrevem um conjunto de funcionalidades do jogo. Nesse diagrama temos 5 papéis diferentes: Jogador, Cliente, Executador de Regras, *Scrum Master* e Gerenciador de Progresso. Os quatro últimos papéis são agentes. No diagrama de casos de uso é apresentado as interações entre o jogador e suas tomadas de decisão. O agente Cliente apresenta as cartas do cliente, os itens de backlog, além de recompensar ou penalizar o jogador invocando o Executador de Regras. Esse agente verifica se as listas de backlog estão de acordo com as regras em relação ao número de itens e itens relacionados entre si. O *Scrum Master* no jogo é o responsável pelo planejamento da *sprint*, além de disponibilizar dicas e cartas surpresas onde o jogo decide o caminho a ser seguido. O *Scrum Master* precisa do Gerenciador de Progresso para avaliar o desempenho do jogador.

Esses diagramas colocaram em prática o uso de regras de negócio. As regras de negócio RN1 e RN2 (Tabela 1) retratam os casos de uso Receber penalização e Receber recompensa, respectivamente, especificando com mais detalhes as restrições impostas pelo sistema para o jogador. Os demais diagramas de caso de uso também estabelecem restrições que permitem o funcionamento correto do jogo, além de determinar como são os agentes nos papéis que estão ativos, apoiando o jogador no caminho do desenvolvimento ágil do sistema.

#### 4.2 Diagramas de Identificação de Agentes

De acordo com Henderson-Sellers [13], a elaboração do diagrama de identificação de agentes começa a partir do diagrama de casos de uso já produzido. Isto é, ele é formado com base no agrupamento de um ou mais casos de uso em pacotes estereotipados. E, ao fazer isso, cada pacote passa a definir as funcionalidades de um agente específico. Além disso, os nomes dos pacotes são os nomes dos agentes representados.

Uma das principais características deste diagrama é o seu relacionamento, onde relacionamentos entre casos de uso de diferentes agentes são estereotipados como “communicate”, enquanto relacionamentos entre casos de uso do mesmo agente são modelados usando “include” e “extend” [13]. Foram identificados quatro agentes: (i) Cliente, (ii) Executor de Regras, (iii) *Scrum Master* e (iv) Gerenciador de Progresso. A Figura 2 representa o diagrama no cenário do jogo Scrumie para os agentes.

O agente Cliente é responsável pelo papel do cliente, sendo encarregado de criar cartas do cliente, cartas de backlog e cartas de validação do backlog, de acordo com o cenário elaborado para a partida.

O agente *Scrum Master* é responsável pelo papel do *Scrum Master* sendo encarregado de apresentar dicas ao jogador me-

**Table 1.** Exemplo de Regras de Negócio

	<b>DESCRIÇÃO DAS REGRAS DE NEGÓCIO</b>
1	O jogador que acertar menos de 8 itens escolhidos para o backlog do produto irá receber uma penalização de 200 reais.
2	O jogador que acertar 8 ou mais itens escolhidos para o backlog do produto irá receber uma recompensa de 400 reais.
3	A pontuação média esperada do jogador é autorregulada de acordo com as partidas previamente realizadas.
4	A média de desempenho aceitável do jogador é calculada a partir da metade do jogo e da seguinte forma: (percentual indicando nível de conclusão do jogo) * (pontuação média esperada).
5	Dicas são exibidas para o usuário caso o seu desempenho esteja abaixo da média a partir da metade do tempo de jogo.
6	O jogador deve escolher entre 3 perfis de desenvolvedores.
7	O jogador deve escolher no máximo 7 itens do backlog do produto para o backlog da <i>sprint</i> .
8	Quando o jogador viola alguma regra do jogo, a regra violada é exibida.
9	O jogo termina quando não houver + itens disponíveis no backlog do produto para o jogador.
10	O jogo retorna para a etapa de Planejamento de <i>Sprint</i> se ainda houver itens do backlog para serem desenvolvidos.

diante o seu desempenho, e selecionar aleatoriamente cartas das respectivas reuniões presentes no Scrum.

Os outros dois agentes foram adicionados com o objetivo de auxiliar a dinâmica do jogo. Um deles é o agente Gerenciador de Progresso, responsável pela avaliação de desempenho do jogador, mantendo uma comunicação com o agente *Scrum Master* a respeito do desempenho apresentado e gerenciando a pontuação do jogador ao longo do jogo. Há também o agente Executor de Regras, responsável por analisar e tratar as violações de regras.

#### 4.3 Diagramas de Identificação de de Papéis

O diagrama de identificação de papéis é um diagrama de sequência UML, onde cada objeto é utilizado para simbolizar o papel do agente, cuja sintaxe é <nome-do-papel>: <nome-do-agente>. Um agente pode desempenhar papéis distintos dentro do mesmo diagrama [13].

A Figura 3 mostra um exemplo do diagrama, que explora a troca de mensagens ao longo de parte do cenário do jogo Scrumie, envolvendo comunicação entre agentes e seus papéis. A elaboração deste diagrama também parte do princípio de que o diagrama de identificação de agentes já foi desenvolvido.

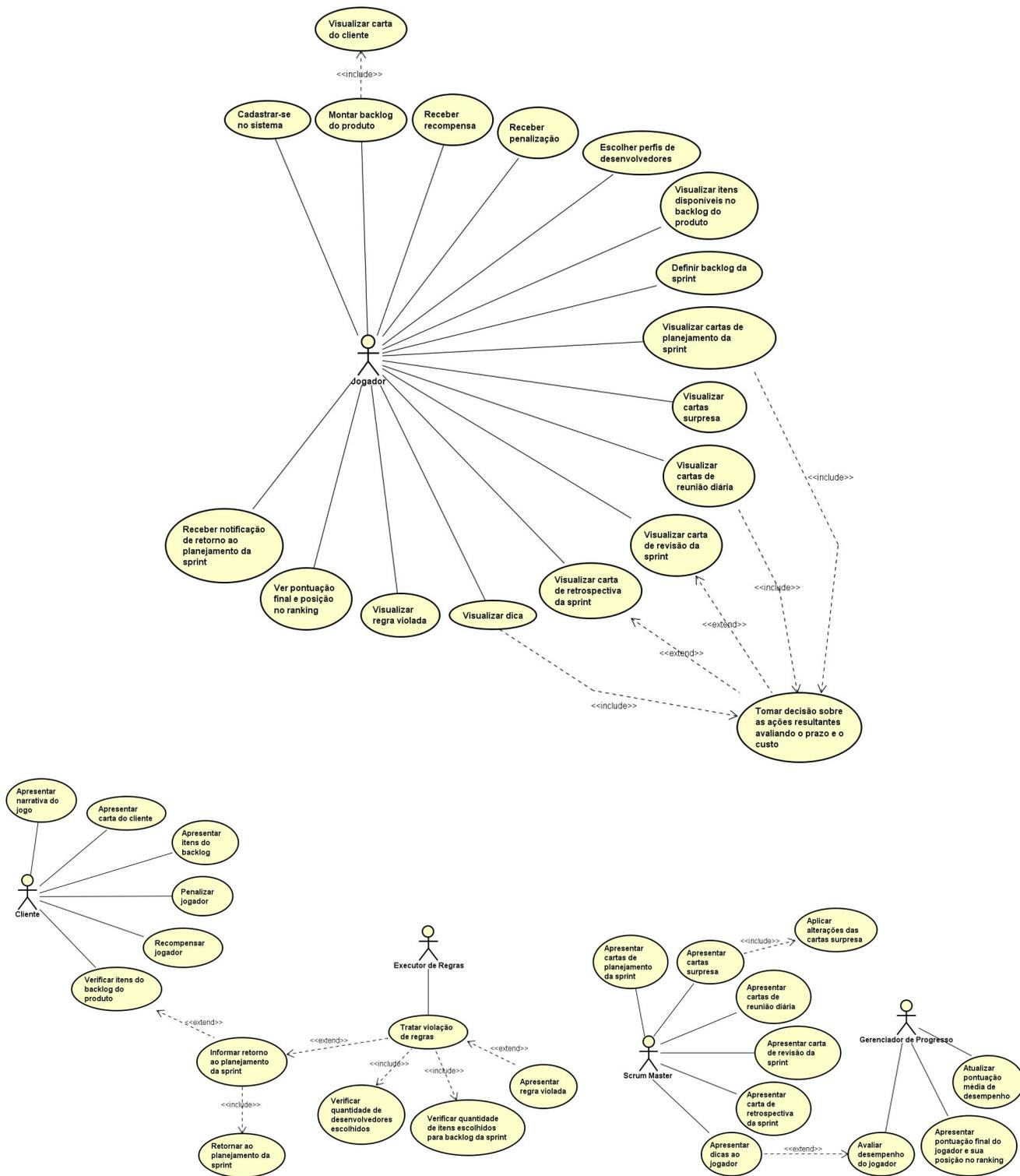


Figure 1. Diagramas de caso de uso do Scrumie

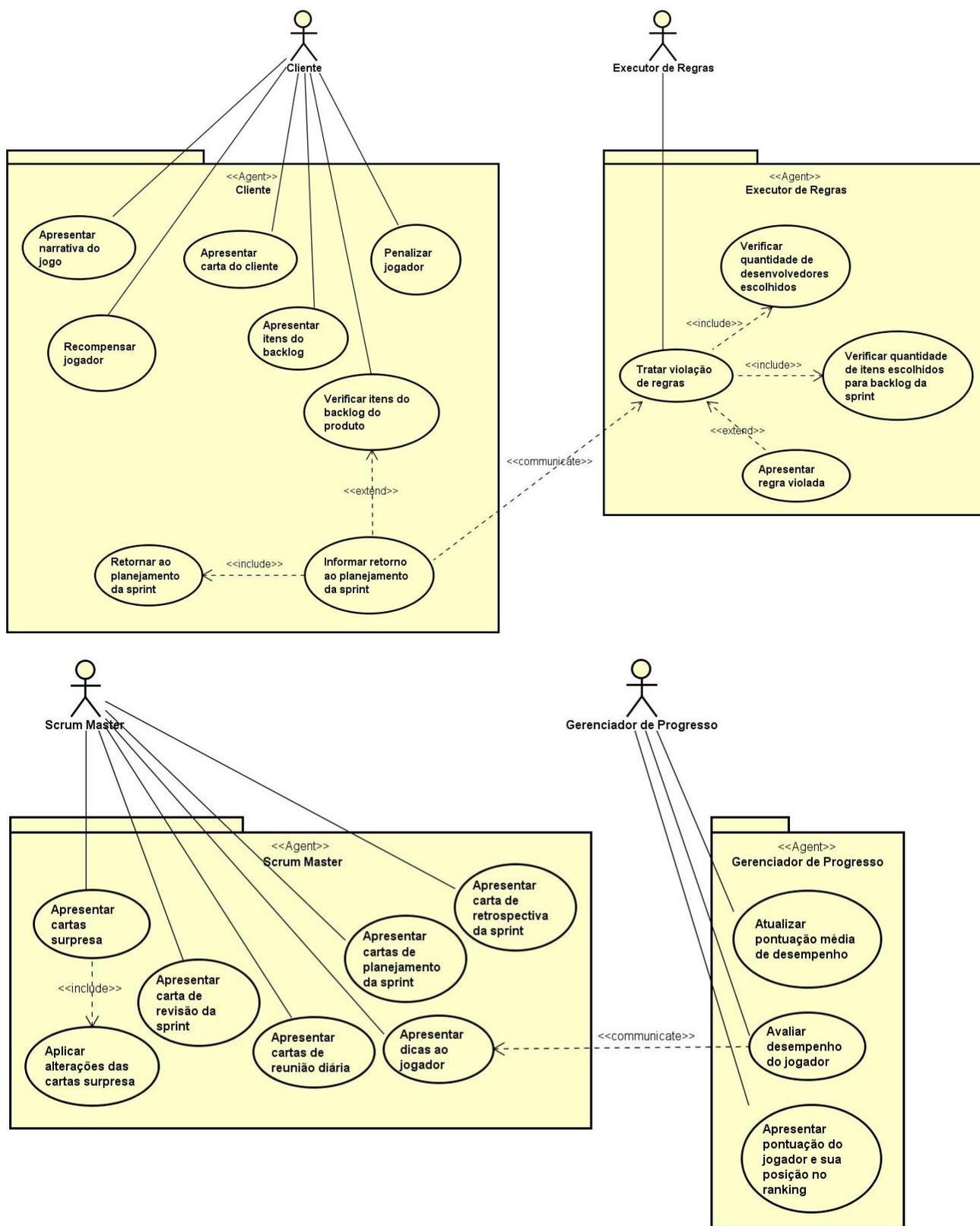


Figure 2. Diagrama de identificação de agentes

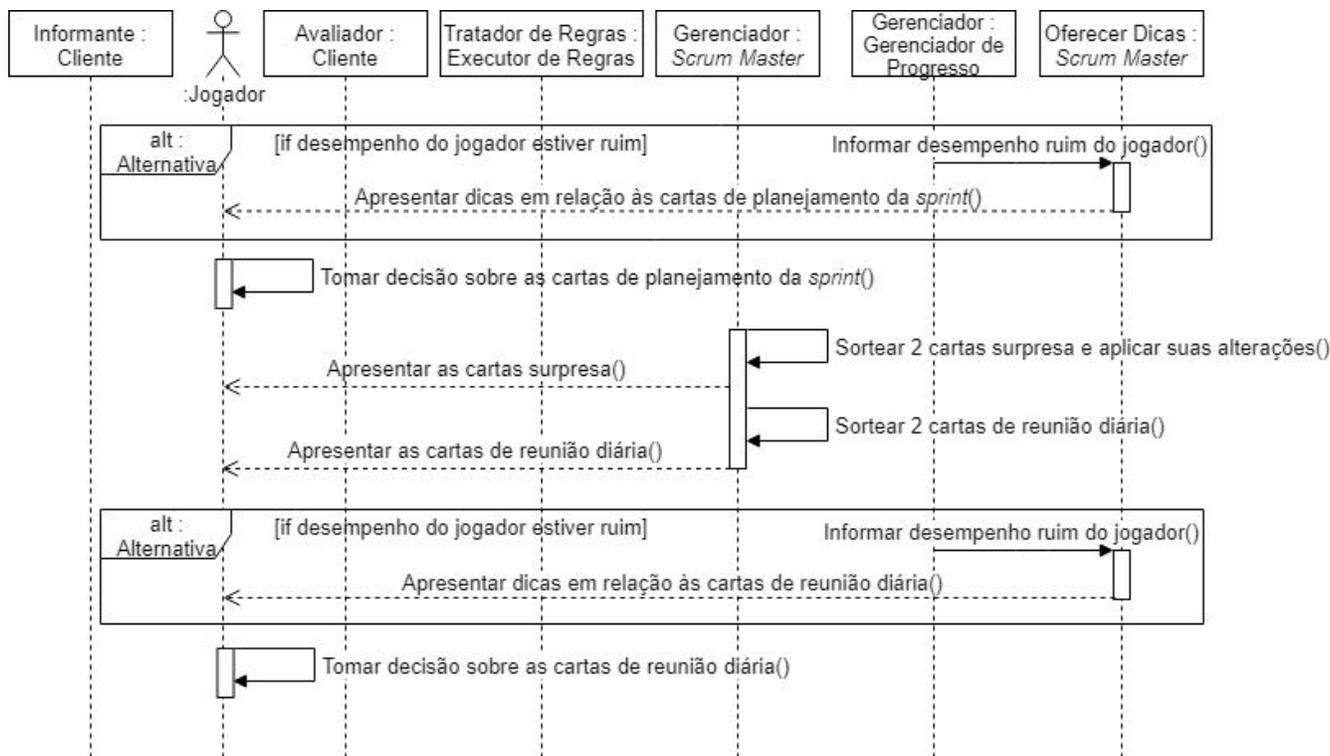


Figure 3. Diagrama de identificação de papéis

## 5. Implementação do jogo

Após a modelagem, deu-se início à implementação do jogo, que foi feita em duas versões. A primeira foi desenvolvida com uma interface de linha de comando e serviu para validar a ideia de tornar digital o jogo de tabuleiro. Como os resultados foram satisfatórios, seguiu-se para a implementação da segunda versão, feita com uma interface gráfica do usuário na forma de uma aplicação web, a qual foi posteriormente avaliada por alunos.

Em ambas as versões, ao se decidir a partir de quais tecnologias o jogo seria construído, considerou-se a necessidade da facilidade de desenvolvimento e da capacidade da tecnologia em representar bem a arquitetura multiagentes. Neste ponto, optou-se por não usar um *framework* de desenvolvimento agentes, uma vez que isso demandaria um período de estudos e compreensão do mesmo, podendo atrasar a entrega da aplicação, e indo além da simplicidade inicialmente proposta.

Com essas considerações, desde a primeira versão optou-se pela utilização da plataforma Node.js<sup>2</sup>, que é capaz de executar a linguagem de programação JavaScript em aplicações desktop. Dessa forma, o código foi escrito em TypeScript<sup>3</sup>, uma linguagem que é compilada para JavaScript e que fornece a versão mais recente dele, agregando funcionalidades, como um melhor suporte à tipagem de dados e à orientação a objetos. Como o JavaScript é a linguagem de programação utilizada

em todos os navegadores modernos e uma das mais usadas no mercado [14], o código escrito na primeira versão pôde ser facilmente portado para uma aplicação web na segunda.

Já na segunda versão do Scrumie foram utilizados os seguintes componentes: MongoDB 3<sup>4</sup>, Express.js<sup>5</sup>, Angular<sup>6</sup> e Node.js. MongoDB é um banco de dados não relacional, orientado a documentos e de código aberto, implementado em C++ com documentos em formato BSON (uma extensão do JSON). Express.js é um *framework* para aplicações web dentro da plataforma Node.js. É responsável pelas requisições do cliente ao servidor e gerenciamento e roteamento dos métodos HTTP (GET, POST, PUT, DELETE etc.). Angular é um *framework* JavaScript de código aberto para aplicações web e *back-end* baseado em TypeScript, liderado pela equipe Angular do Google.

Depois das tecnologias terem sido definidas, a organização da aplicação foi projetada com os objetivos de facilitar a utilização de boas práticas de desenvolvimento e de agilizar a implementação. Para tal, separou-se o sistema nos seguintes agentes: (i) Cliente, responsável pela inicialização e por apresentar o cenário da jogabilidade; (ii) Gerenciador de Progresso, responsável pelo controle do fluxo de interações entre o jogador e o sistema; (iii) *Scrum Master*, que gerencia todas as cerimônias do Scrum; (iv) Executor de Regras, que valida as atividades do jogador.

A implementação dos agentes proporciona a adição de

<sup>2</sup>Disponível em: <http://nodejs.org/en/about/>

<sup>3</sup>Disponível em: <https://www.typescriptlang.org/>

<sup>4</sup>Disponível em: <https://www.mongodb.com/>

<sup>5</sup>Disponível em: <https://expressjs.com/>

<sup>6</sup>Disponível em: <https://angular.io/>

inteligência ao jogo, além de ter facilitado a separação das responsabilidades e dos papéis dos agentes do sistema, permitindo uma melhor organização.

## 5.1 Cenários de Uso e Comportamento dos Agentes

Ao iniciar o jogo, a Tela da Carta do Cliente (Figura 4) é exibida com a narrativa do jogo, o sistema a ser desenvolvido e o orçamento em valor e dias.

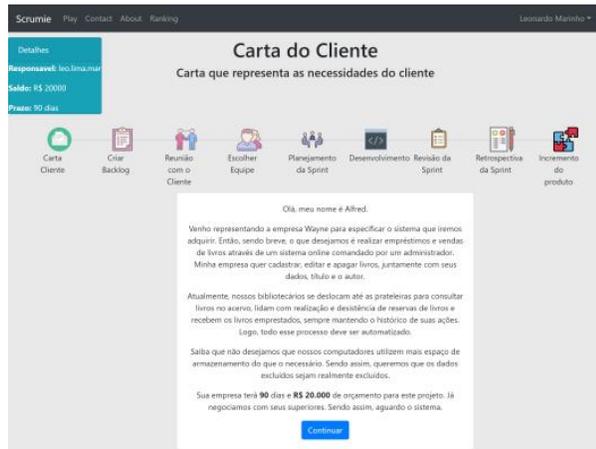


Figure 4. Tela da Carta do Cliente

A atividade a seguir é o planejamento do desenvolvimento com a seleção dos requisitos que estão no *backlog* do produto. A Tela de Selecionar *Backlog* do Produto (Figura 5) exibe todos os itens de *backlog* possíveis. O jogador monta *backlog* do produto, de acordo com a carta do cliente. Nesse momento, o agente Cliente valida a escolha de itens do *backlog* e o jogador é recompensado ou penalizado de acordo com as regras do negócio validadas pelo agente Executor de Regras. O requisitos do produto (itens do *backlog*) escolhido pelo jogador aparecem na Tela de Reunião com o Cliente (Figura 6).



Figure 5. Tela de Selecionar *Backlog* do Produto

A seguir o jogador tem que montar sua equipe, e o Scrumie exibe perfis de desenvolvedores disponíveis para o projeto



Figure 6. Tela de Reunião com o Cliente

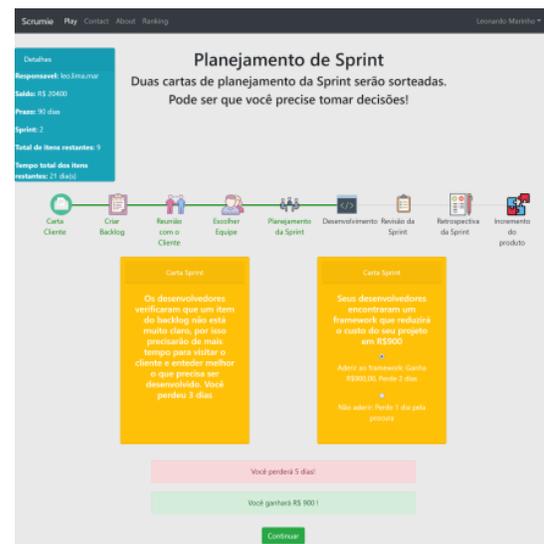


Figure 7. Tela de Planejamento de *Sprint*

para que sejam escolhidos 3 perfis de desenvolvedores. O agente Executor de Regras analisa essa escolha e avisa se há alguma violação de regras.

A próxima etapa é o Planejamento da *Sprint*, assim o agente *Scrum Master* sorteia 2 cartas de Planejamento de *Sprint* e exibe dicas ao jogador (Figura 7), que toma as decisões de acordo com as cartas selecionadas/dicas no passo anterior. Na continuação, o jogador deve escolher no máximo 7 itens do *backlog* do produto para compor o *backlog* da *sprint*. O agente Executor de Regras verifica a violação de regras em relação aos itens escolhidos para *backlog* da *sprint*, informando a regra violada.

Após a escolha do *backlog* da *sprint*, é apresentado o resumo da *Sprint* (Figura 8) e, ao continuar, o agente *Scrum Master* sorteia 2 cartas surpresa e as apresentam na Tela de Desenvolvimento (Figura 9), que contém também os itens de *backlog* incluídos e retirados da lista. Neste momento, as alterações de orçamento e prazo são aplicadas.

Na etapa de desenvolvimento, é realizada uma Reunião Diária. O agente *Scrum Master* sorteia 2 cartas de Reunião Diária e o jogador toma as decisões de acordo com as cartas

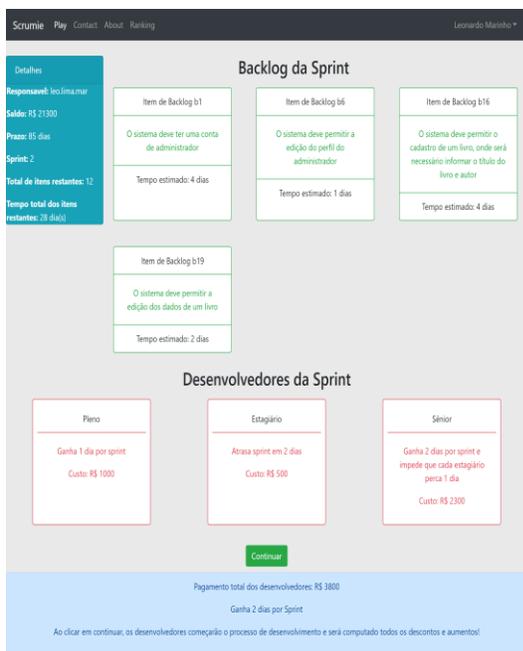


Figure 8. Tela de Backlog e Desenvolvedores da Sprint

selecionadas.

Após a reunião diária, o agente *Scrum Master* sorteia 1 carta de Revisão da *Sprint* e a exhibe ao jogador, que toma decisão de acordo com a carta selecionada e dicas do passo anterior.

Na Retrospectiva da *Sprint*, o agente Cliente valida se todos os itens do *backlog* do produto foram desenvolvidos. Caso não tenha finalizado o *backlog*, o agente Cliente é informado sobre retorno para o Planejamento da *Sprint* (Figura 7), pois o produto não foi concluído na *Sprint* (Figura 10).

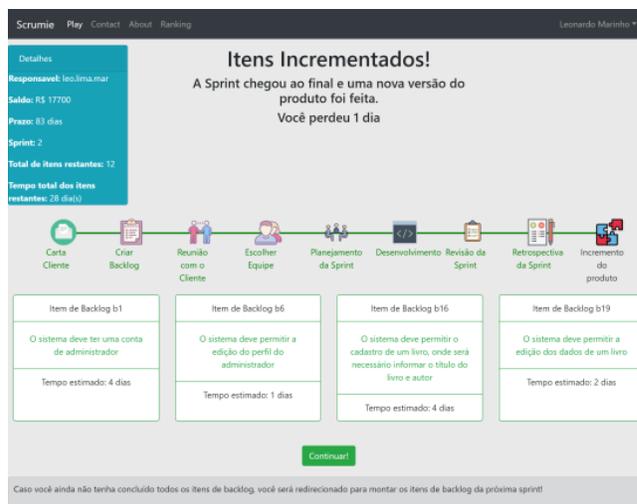


Figure 10. Tela de Itens Incrementados

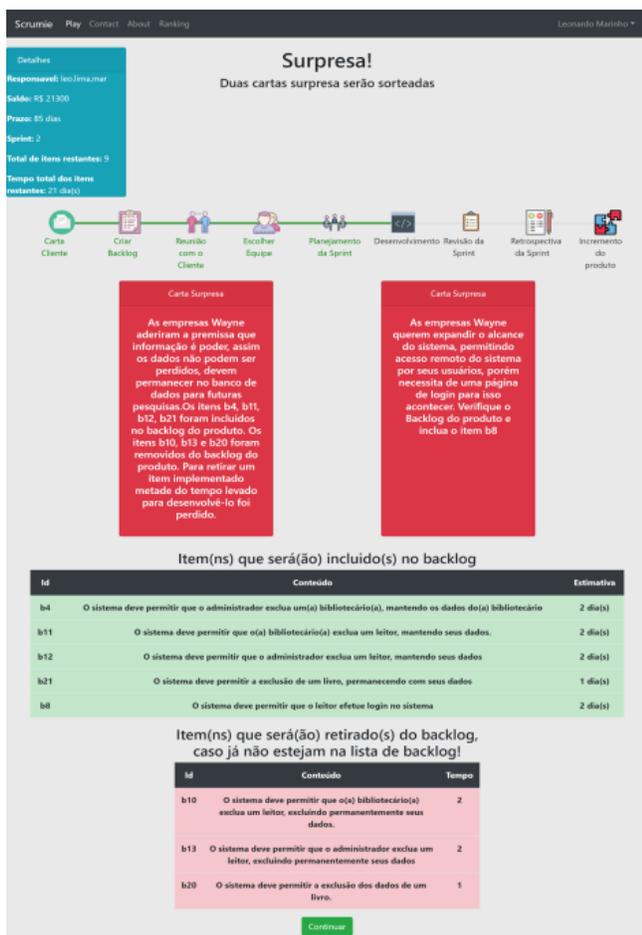


Figure 9. Tela de Desenvolvimento com as Cartas Surpresas

No final, assim que o *backlog* do produto é concluído, o agente Gerenciador de Progresso calcula a pontuação do jogador e sua posição no ranking, baseado no prazo e no orçamento restantes.

## 6. Avaliação do Scrumie

O Scrumie foi avaliado em dois momentos. No primeiro, foi utilizada uma versão inicial do jogo já com a interface nova, sendo aplicada em duas turmas no mesmo dia (Engenharia de Software e Metodologia Científica no Projeto Final), ambas do curso de Ciências da Computação do semestre 2019.2 de uma universidade pública do Rio de Janeiro. Ambas as turmas continham em torno de 7 alunos cada, mas somente metade compareceu a essa aula. A segunda avaliação foi realizada na turma de Interfaces Humano-Computadores desse mesmo curso e universidade, tendo somente um aluno em comum, e composta de 20 alunos.

Em ambas as avaliações foi aplicado um projeto cadastrado no sistema para ser desenvolvido, utilizando o *framework* Scrum, onde vários acontecimentos mudavam o saldo e o prazo do Gerente de Projetos. Antes do início de cada avaliação, cada aluno preencheu um termo de consentimento e um questionário de caracterização, com o objetivo de identificar a formação acadêmica, a formação geral e o nível de familiaridade em Engenharia de Software, Desenvolvimento de

Software, Desenvolvimento de Software com Método Ágil, e Desenvolvimento de Software com Scrum.

Após o uso do Scrumie, o questionário de satisfação do usuário denominado SUS (*System Usability Scale*) [15] foi utilizado, acompanhado de duas perguntas abertas. As perguntas abertas pediam para apontar o que o usuário mais gostou no Scrumie e as mudanças sugeridas no jogo.

SUS é uma ferramenta simples, que tem como objetivo medir a usabilidade de um sistema, com uma pontuação que varia de uma avaliação negativa (0) a uma avaliação positiva (100) [16]. Neste trabalho, optamos por adotar a escala Likert, utilizada em [17], onde uma declaração é feita e o respondente então indica o grau de concordância em uma escala de 0 a 4, pontos que podem ser interpretados da seguinte forma, respectivamente: (i) Discordo totalmente; (ii) Discordo levemente; (iii) Não discordo, nem concordo; (iv) Concordo levemente; e (v) Concordo totalmente. O questionário tem 10 declarações, sendo as ímpares positivas e, as pares, negativas [15].

Para cada questão, a pontuação gerada pelo método SUS foi considerada como: avaliação ruim quando a pontuação fosse inferior a 35,7; regular quando entre 35,7 e 71,4; e boa quando superior a 71,4.

Para calcular a pontuação geral do SUS de cada usuário, primeiro soma-se as contribuições de pontuação para cada item. A contribuição de pontuação de cada item varia de 0 a 4. Para os itens 1, 3, 5, 7 e 9, a contribuição da pontuação é a posição da escala menos 1. Para os itens 2, 4, 6, 8 e 10, a pontuação é 5 menos a posição da escala. Multiplica-se, então, a soma das pontuações em 2,5 [15].

Tenório *et al.* [18] propõe como a seguinte base temática para as questões: (i) facilidade de aprendizagem nas questões 3, 4, 7 e 10; (ii) eficiência nas questões 5, 6 e 8; (iii) facilidade de memorização na questão 2; (iv) minimização dos erros na questão 6 e; (v) satisfação nas questões 1, 4 e 9.

## 6.1 Análise dos Resultados

O questionário de caracterização das duas turmas mostrou que há pouca experiência em desenvolvimento de software utilizando algum método ágil, tendo em vista que mais de 50 percentuais dos participantes relataram não ter nenhum conhecimento.

A primeira avaliação do jogo foi realizada com apenas 7 alunos e todos simularam o desenvolvimento do projeto de uma biblioteca utilizando o *framework* Scrum. Ao término da avaliação, foi preenchido o questionário *follow-up* com as perguntas abertas sobre a qualidade do jogo e sugestão de melhorias, além do questionário SUS. O propósito era avaliar como Scrumie serve como ferramenta de ensino.

Individualmente, cada questão obteve a pontuação apresentada na Figura 11. Aplicando a média das pontuações obtidas em cada item do questionário, obtêm-se que para o tema facilidade de aprendizagem, resulta a pontuação de 60,71, indicando que a ferramenta pode ser utilizada para o aprendizado, mas poderia melhorar. No tema eficiência, obteve-se a pontuação de 7, concluindo que a aplicação é eficiente na

perspectiva dos participantes dessa avaliação. Na temática facilidade de memorização, resulta a pontuação de 71,43, logo, os participantes acreditam que a ferramenta provém satisfação no tema. Em relação à minimização de erros, obteve-se a pontuação de 85,71, evidenciando-se que a ferramenta possui mecanismos para minimização de erros. Na satisfação do usuário, a pontuação foi de 63,1, que indica que a aplicação é satisfatória na perspectiva do participante, mas ainda pode melhorar.

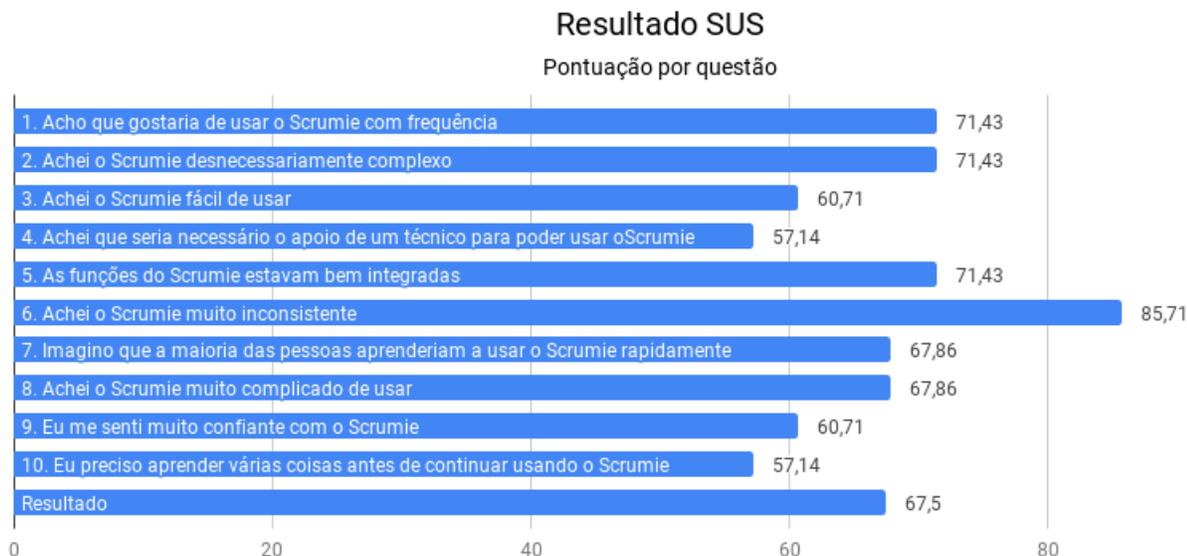
Na primeira avaliação, a pontuação geral do questionário SUS para o Scrumie, gerou os seguintes dados: 42,9 percentuais dos participantes classificaram a experiência como regular e, aproximadamente 57,1 percentuais como boa.

Analisando a avaliação da opinião dos usuários em relação ao Scrumie, nota-se que as respostas foram favoráveis a utilização do aplicativo, reiterando a importância do mesmo no processo de aprendizagem. Em relação a pergunta livre sobre o que não gostou ou mudaria no jogo Scrumie, foi constatado que o jogo poderia melhorar, indicando ao jogador, o que é feito na próxima etapa, isto é, eliciar ao jogador, o quanto ele perde ou ganha em cada etapa. Foi, também, identificado que havia uma inconsistência na contagem de dinheiro.

Devido a pontuação geral de 67,5 no questionário SUS, dessa avaliação pode-se concluir que a aplicação pode ser definida como regular e essa pontuação indica que o jogo precisava ser mais intuitivo.

Com o objetivo de aplicar melhorias de acordo com os resultados obtidos no primeiro teste do jogo, mudanças foram realizadas, principalmente para melhorar a facilidade de aprendizagem e a satisfação do usuário. A segunda avaliação contou com 12 alunos numa turma de 20 e, como na outra, eles simularam, na perspectiva de gerente de projeto, o desenvolvimento de um software de uma biblioteca utilizando o *framework* Scrum. Ao término da avaliação, também foi preenchido o mesmo questionário *follow-up*. Individualmente, nessa segunda avaliação cada questão obteve a pontuação apresentada na Figura 12. Vale ressaltar que foram encontrados problemas por alguns usuários em uma tela no programa durante o teste, o que afetou as notas da avaliação, principalmente na eficiência e diminuição de erros.

Nessa segunda avaliação, aplicando as temáticas definidas por Tenório *et al.* [18], considerando a média das pontuações obtidas em cada item do questionário SUS, obtêm-se que: para o tema facilidade de aprendizagem, resulta a pontuação de 64,58, indicando que a ferramenta é satisfatória na perspectiva do participante, porém, pode melhorar. Em relação ao tema eficiência, obteve-se a pontuação de 67,36, concluindo que a aplicação é satisfatoriamente eficiente na perspectiva do participante. Na temática facilidade de memorização, resulta a pontuação de 85,42. Logo, os participantes acreditam que a aplicação provém de facilidade de memorização. Para o tema minimização de erros, obteve-se a pontuação de 68,75, apesar de erros no jogo, a ferramenta possui mecanismos para minimização de erros, mas ainda precisa de melhoria.



**Figure 11.** Resultado do teste de usabilidade por questão da primeira avaliação

Na satisfação do usuário, resultou na pontuação de 65,28, que indica que a aplicação é satisfatória na perspectiva do participante, contudo, precisa de melhoria.

Nesta avaliação, o questionário SUS gerou os seguintes dados: 58,3 percentuais dos participantes regular; e aproximadamente 41,7 percentuais como boa.

A pontuação geral das repostas do questionário SUS nessa avaliação gerou a pontuação de 66,8, o que significa que a aplicação foi avaliada como regular no quesito usabilidade.

Comparando as duas avaliações e utilizando como base as temáticas definidas por Tenório *et al.* [18], obtivemos os seguintes resultados (Tabela 2): Na temática facilidade de aprendizado, houve uma pequena melhoria da nota, que pode estar relacionado ao fato de ter sido adicionado descrição em cada etapa no jogo, e também alertas resumindo os resultados de cada etapa. Em relação ao tema eficiência, a nota da segunda avaliação decresceu comparada com a primeira, este fato está relacionado com os problemas que ocorreram no jogo durante o teste. No tema facilidade de memorização, houve um grande aumento da nota, fato que tem relação com as melhorias propostas e implementadas após a primeira avaliação. Na minimização de erros, um decréscimo de em torno de 18 percentuais e com certeza foi devido aos problemas identificados. Na satisfação do usuário houve um pequeno incremento na nota, o que implica que, apesar dos problemas ocorridos durante o teste, a aplicação ainda é satisfatória na perspectiva dos usuários.

Avaliando as opiniões dos usuários em relação ao Scrumie, a análise de dados sugere que as repostas foram favoráveis ao jogo e sua utilização, reiterando que o jogo pode ser considerado importante no processo da aprendizagem. Já na análise dos pontos negativos observa-se que o jogo tem com problemas na tela “Escolher perfil de desenvolvedores”

**Table 2.** Comparativo das pontuações por temática das duas avaliações

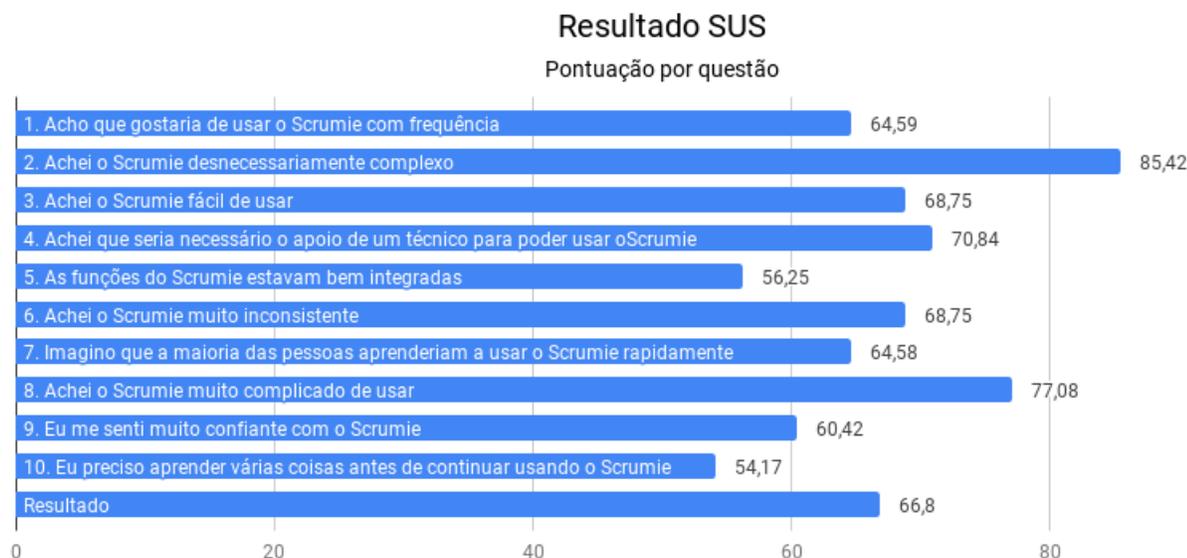
Temática	1a Avaliação	2a Avaliação
Facilidade de aprendizado	60.71	64.58
Eficiência	75	67.36
Facilidade de memorização	71.43	85.42
Minimização de erros	85.71	68.75
Satisfação	63.1	65.28

(alerta que se sobrepôs as opções) e na validação das cartas surpresas.

A pontuação geral das repostas do questionário SUS foram parecidas, na primeira gerou 67,5 e na segunda a pontuação de 66,8. Assim, significa que o Scrumie foi avaliado como regular no quesito usabilidade em ambas as avaliações, sendo que a segunda ficou prejudicada por apresentar inconsistências em duas telas.

As avaliações, embora tenham sido com poucos alunos, indicam que o jogo é regular. Por mais que problemas tenham ocorrido durante o último teste, o Scrumie agradou aos participantes. Assim, podemos inferir que o Scrumie - um jogo que simula o método ágil Scrum - tem potencial de atrair e auxiliar alunos interessados nesse aprendizado. Portanto, ajustes já estão sendo realizados de forma a retirar todos as inconsistências do sistema e deixar a interface mais intuitiva, além de migrar para um *framework* orientado a agentes para possibilitar a visibilidade da simulação dos agentes.

As avaliações foram planejadas em detalhes, mas a amostra é pequena não sendo significativa para evidenciar padrões consistentes que permitam resultados conclusivos. Entretanto, todos os alunos são de graduação e a maioria com pouco



**Figure 12.** Resultado do teste de usabilidade por questão da segunda avaliação do Scrumie

conhecimento de métodos ágeis, mas com experiência no mercado. Além disso, essas avaliações são importantes para correção e evolução do Scrumie.

## 7. Conclusões

O jogo Scrumie representa a maior contribuição deste artigo, como proposta de investir no aprendizado prático por meio do uso de jogos educativos em software. Considera-se um estudo relevante devido à busca crescente por eficiência em empresas no desenvolvimento de sistemas e à consequente importância dada ao conhecimento de como aplicar métodos ágeis.

O uso de agentes facilitou a separação de papéis envolvidos na equipe Scrum, apesar de trazer maior complexidade para a implementação. Por ser uma arquitetura diferente da convencional, utilizar agentes foi um desafio, no sentido de saber inserir a autonomia e a inteligência geralmente associadas aos agentes da maneira correta. Esse primeiro protótipo foi desenvolvido de forma bastante simples e no futuro pretende-se a incorporação de uma arquitetura BDI por meio de um *framework* orientado a agentes. Assim, a abordagem de ter agentes representando pessoas de uma equipe serviu como uma solução para este problema abstrato, gerando bons resultados. No futuro, esses agentes podem evoluir, estendendo o jogo para uso de múltiplos papéis. A arquitetura de agentes facilita a evolução, pois os comportamentos estão nos agentes criados, que podem tornar a evolução mais simples e a incorporação de regras mais dinâmicas no jogo.

Alguns pontos negativos originados no 2TScrum não foram abordados para melhoria no Scrumie, como o fato de possuir apenas um cenário de desenvolvimento simulado – o sistema web para uma biblioteca; e possuir um limite de participantes no jogo, sendo que o Scrumie admite apenas um jogador. Sendo assim, esses pontos ficam sugeridos como

melhorias futuras.

Scrumie irá evoluir considerando os aspectos sugeridos nessas avaliações e incorporação de inteligência a alguns agentes. Além disso, pretende-se realizar uma avaliação da aprendizagem com as turmas em 2020 aplicando pré e pós-testes conforme proposto no MAVA [19]. Foi iniciado, também uma avaliação por especialistas em desenvolvimento ágil.

## Agradecimentos

Os autores agradecem o apoio financeiro fornecido pela Fundação de Apoio a Pesquisa do Estado do Rio de Janeiro (FAPERJ), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e Universidade do do Estado do Rio de Janeiro (UERJ).

## Contribuições dos Autores

Este trabalho foi realizado no projeto da final da disciplina de Sistemas Multiagentes do mestrado em Ciências Computacionais por Leonardo Lima Marinho, Suelen Regina dos Santos e Bruna Costa Cons, elaborando a proposta, requisitos, modelagem e primeira versão do Scrumie. A segunda versão foi implementada por Leonardo Andrade, aluno de Iniciação Científica, tendo o apoio dos alunos de mestrado e dos professores orientadores. Orientação foi realizada pelos professores Marcelo Schots e Vera Maria Werneck.

## Referências

[1] SOARES, M. dos S. Comparação entre metodologias ágeis e tradicionais para o desenvolvimento de software. *IN-*

- FOCOMP Journal of Computer Science*, v. 3, n. 2, p. 8–13, 2004.
- [2] BECK, K. *et al.* *Manifesto for agile software development*. 2001. Acesso em 12 de dez. de 2018. Disponível em: <http://agilemanifesto.org/>.
- [3] SILLE, F.; BRAGA, J. C. Software educacional para prática do scrum. In: SBC. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. Santo André, São Paulo, 2013. v. 2, n. 1, p. 152–161.
- [4] GRANDO, R. C. O jogo na educação: aspectos didático-metodológicos do jogo na educação matemática. *Campinas: Unicamp*, 2001.
- [5] LEGEY, A. P. *et al.* Desenvolvimento de jogos educativos como ferramenta didática: um olhar voltado à formação de futuros docentes de ciências. *Alexandria: Revista de Educação em Ciência e Tecnologia*, v. 5, n. 3, p. 49–82, 2012.
- [6] BRITO, A.; VIEIRA, J. '2tscrum': A board game to teach scrum. In: ACM. *Proceedings of the 31st Brazilian Symposium on Software Engineering*. Fortaleza, Ceará, 2017. p. 279–288.
- [7] DEVMEDIA. *Processos Ágeis para desenvolvimento de Software – Parte 02*. 2008. Disponível em: <https://www.devmedia.com.br/processos-ageis-para-desenvolvimento-de-software-parte-02/9209>.
- [8] CHELLA, A. *et al.* Agile passi: An agile process for designing agents. *International Journal of Computer Systems Science & Engineering*, v. 21, n. 2, p. 133–144, 2006.
- [9] CHELLA, A. *et al.* From passi to agile passi: Tailoring a design process to meet new needs. In: IEEE. *Proceedings. IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2004.(IAT 2004)*. Beijing, China, 2004. p. 471–474.
- [10] CONS, B. *Comparação entre Metodologias Ágeis Orientadas a Agentes*. 2018. Monografia (Graduação em Ciência da Computação), Instituto de Matemática e Estatística, Universidade do Estado do Rio de Janeiro, Rio de Janeiro.
- [11] LARMAN, C. *Utilizando UML e padrões*. Porto Alegre: Bookman Editora, 2000. 492 p.
- [12] DALLAVALLE, S. I.; CAZARINI, E. W. Regras do negócio, um fator chave de sucesso no processo de desenvolvimento de sistemas de informação. *São Carlos: USP-EESC-Escola de Engenharia de São Carlos-Área: Engenharia de Produção*, 2000.
- [13] HENDERSON-SELLERS, B.; GIORGINI, P. *Agent-oriented methodologies*. Pennsylvania: Igi Global, 2005. 428 p.
- [14] DEVMEDIA. *Top 10 linguagens de programação mais usadas no mercado*. 2018. Disponível em: <https://www.devmedia.com.br/top-10-linguagens-de-programacao-mais-usadas-no-mercado/39635>.
- [15] BROOKE, J. *et al.* Sus-a quick and dirty usability scale. *Usability evaluation in industry*, v. 189, n. 194, p. 4–7, 1996.
- [16] BANGOR, A.; KORTUM, P.; MILLER, J. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, v. 4, n. 3, p. 114–123, 2009.
- [17] AGUIAR, T. V. de; SÁNCHEZ-PI, N.; WERNECK, V. M. B. Aide-vr: Extending a virtual living lab framework using virtual reality. In: SPRINGER. *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Toledo, Spain, 2018. p. 270–281.
- [18] TENÓRIO, J. M. *et al.* Desenvolvimento e avaliação de um protocolo eletrônico para atendimento e monitoramento do paciente com doença celíaca. *Revista de Informática Teórica e Aplicada*, v. 17, n. 2, p. 210–220, 2010.
- [19] FERREIRA, A. d. A. *MA-AVA: Modelo de Avaliação da Aprendizagem em Ambientes Virtuais*. 2017. Dissertação (Ciências Computacionais) - Instituto de Matemática e Estatística, Universidade do Estado do Rio de Janeiro, Rio de Janeiro.