



# Jogo sério sobre gestão de fábrica de *software*

Ramon Vaz de Mello Larivoir

JUIZ DE FORA  
SETEMBRO, 2021

# Jogo sério sobre gestão de fábrica de *software*

RAMON VAZ DE MELLO LARIVOIR

Universidade Federal de Juiz de Fora  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação  
Bacharelado em Sistemas de Informação

Orientador: Igor de Oliveira Knop

JUIZ DE FORA  
SETEMBRO, 2021

# JOGO SÉRIO SOBRE GESTÃO DE FÁBRICA DE *software*

Ramon Vaz de Mello Larivoir

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM SISTEMAS DE INFORMAÇÃO.

Aprovada por:

Igor de Oliveira Knop  
D.Sc. Modelagem Computacional

Priscila Vanessa Zabala Capriles Goliatt  
D.Sc. Modelagem Computacional

Jairo Francisco de Souza  
D.Sc. em Informática

JUIZ DE FORA  
02 DE SETEMBRO, 2021

*Dedico este trabalho à minha mãe Maria (in memoriam), que infelizmente não pode presenciar este momento, mas sempre se faz presente em todos os dias da minha vida. Ao meu pai Cláudio e irmãos Vanessa e João Cláudio pelo apoio, carinho e incentivo em todos estes anos. Também dedico à minha noiva Nathalia, que sempre esteve ao meu lado me apoiando e me aconselhando em cada escolha.*

## Resumo

Empresas e profissionais que desenvolvem projetos desejam que os mesmos sejam sempre bem sucedidos. Para se obter sucesso, um bom gerenciamento deve ser executado e a experiência do gerente é um fator importante. Existem vários modelos e métodos de gerenciamento de projetos, mas seu ensino é difícil por nem sempre existir a possibilidade de praticar em tempo hábil durante o curso. Para suprir essa demanda, os jogos têm sido utilizados ao longo de décadas para fornecer um ambiente controlado no qual o professor apresenta um conceito ou técnica sob a qual os alunos atuam e discutem sobre as situações e problemas encontrados. Este trabalho investiga o uso de jogos para simular a gestão de uma nova empresa de fábrica de *software* na qual os jogadores deverão empregar os recursos iniciais e interagir com os demais jogadores para contratação da equipe, realizar orçamentos, cumprir projetos com qualidade e no prazo. Para tanto, um novo jogo foi projetado e desenvolvido e, posteriormente, um protótipo *online* foi usado para avaliar sua eficácia junto aos alunos de uma disciplina de empreendedorismo. Os resultados apontam que o jogo atinge seus objetivos e direciona possíveis mudanças para novas iterações futuras.

**Palavras-chave:** jogos sérios, gerência de projetos, simulação.

## Abstract

Companies and professionals who develop projects want them to be always successful. To be successful, good management must be performed and the manager's experience is an important factor. There are several models and methods of project management, but their teaching is difficult as there is not always the possibility to practice in a timely manner during the course. To meet this demand, games have been used over the decades to provide a controlled environment in which the teacher presents a concept or technique under which students act and discuss the situations and problems encountered. This work investigates the use of games to simulate the management of a new software factory company, in which players must employ the initial resources and interact with other players to hire the team, make budgets, fulfill projects with quality and on time. For that, a new game is designed and developed and, later, an online prototype is used to assess its effectiveness with students of an entrepreneurship discipline. The results indicate that the game achieves its goals and directs possible changes to new future iterations.

**Keywords:** serious games, project management, simulation.

## Agradecimentos

Aos meus pais e meu irmãos por todo amor e apoio durante esta jornada.

À minha noiva, pelo seu amor, que me fortifica em cada momento difícil, e pela sua companhia que sempre foi tão presente e acolhedora.

Aos meus amigos pelos ótimos momentos vividos em todos estes anos que, sem dúvida, ajudaram a aliviar a pressão.

Ao Professor Igor Knop, pela orientação, dedicação e sua paciência comigo durante toda a elaboração deste trabalho.

À Oficina de Playtest e todos seus membros e participantes, que foram essenciais para a evolução e melhoria do jogo devido à disponibilidade para testar e conceder *feedbacks* sempre construtivos.

À Professora Priscila Capriles, por disponibilizar sua turma para que os alunos pudessem jogar e que, por conta disso, foi possível coletar todos os dados que formam os resultados deste trabalho.

# Conteúdo

<b>Lista de Figuras</b>	<b>7</b>
<b>1 Introdução</b>	<b>10</b>
1.1 Descrição do Problema e Justificativa . . . . .	11
1.2 Objetivos . . . . .	12
<b>2 Fundamentação Teórica</b>	<b>13</b>
2.1 Processos de <i>Software</i> . . . . .	13
2.2 Modelos de Processo de <i>Software</i> . . . . .	14
2.3 Metodologias Ágeis . . . . .	15
2.4 Empreendedorismo e <i>Software</i> . . . . .	19
2.5 Uso de jogos para ensino de Engenharia de <i>Software</i> . . . . .	20
2.5.1 SimSE . . . . .	20
2.5.2 <i>Software Kaihatsu Game</i> . . . . .	20
2.5.3 <i>Problems and Programmers</i> . . . . .	21
2.6 Projeto, desenvolvimento e teste de jogos . . . . .	22
2.6.1 Modelo MDA: <i>Mechanics, Dynamics and Aesthetics</i> . . . . .	22
2.6.2 Estética . . . . .	23
2.6.3 Dinâmica . . . . .	24
2.6.4 Mecânica . . . . .	25
2.6.5 Jogos modernos de tabuleiro . . . . .	25
2.7 Pesquisa Baseada em <i>Design</i> . . . . .	27
2.8 Avaliação de Jogos Educativos com o <i>Model for the Evaluation of Educational Games (MEEGA+)</i> . . . . .	27
<b>3 Método</b>	<b>29</b>
3.1 Estrutura da pesquisa . . . . .	29
3.2 Ambiente para teste de jogabilidade . . . . .	30
3.3 Avaliação dos ciclos de jogabilidade . . . . .	31
3.4 Formulário para aplicação do MEEGA+ . . . . .	31
3.5 Acompanhamento de versões do protótipo . . . . .	32
<b>4 Desenvolvimento</b>	<b>33</b>
4.1 Projeto do jogo de tabuleiro <i>Bootstrapping</i> . . . . .	33
4.1.1 Estética . . . . .	33
4.1.2 Mecanismos . . . . .	34
4.1.3 Dinâmica . . . . .	36
4.2 Implementação como jogo de tabuleiro . . . . .	38
4.3 Desenvolvimento do protótipo digital . . . . .	42
4.4 Avaliação do <i>Bootstrapping</i> com o MEEGA+ . . . . .	46
<b>5 Considerações Finais</b>	<b>55</b>
<b>Bibliografia</b>	<b>57</b>

<b>A</b>	<b>Evolução do <i>Bootstrapping</i></b>	<b>60</b>
A.1	Versão 1 . . . . .	60
A.2	Versão 2 . . . . .	61
A.3	Versão 3 . . . . .	61
A.4	Versão 4 . . . . .	62

## Lista de Figuras

2.1	Fases do modelo de processo de <i>software</i> em cascata. Fonte: Sommerville (2011). . . . .	14
2.2	Modelo Incremental. Fonte: Sommerville (2011). . . . .	15
2.3	Modelo de Prototipação. Fonte: Pressman e Maxim (2016). . . . .	16
2.4	Modelo Espiral. Fonte: Pressman e Maxim (2016). . . . .	16
2.5	<i>Extreme Programming</i> . Fonte: Pressman e Maxim (2016). . . . .	18
2.6	Interface do jogo SimSE . . . . .	21
2.7	Interface do jogo <i>Software Kaihatsu Game</i> (SKG) . . . . .	22
2.8	Exemplos de carta de projeto, carta de programador, carta de problema e carta de conceito do jogo <i>Problems and Programmers</i> (PnP) . . . . .	22
2.9	Associação de conceitos do modelo <i>Mechanics, Dynamics and Aesthetics</i> (MDA). Figura adaptada de Zaffari e Battaiola (2014). . . . .	23
2.10	Representação das perspectivas do <i>designer</i> e do jogador. Fonte: Hunicke, LeBlanc e Zubek (2004) . . . . .	24
2.11	Hierarquia dos fatores de qualidade em dimensões/subdimensões (Petri, Wangenheim e Borgatto (2016)). . . . .	28
4.1	Principais conceitos e inter-relações do jogo <i>Bootstrapping</i> . . . . .	35
4.2	Cartas de atividade: uma atividade de nível 2 e uma atividade de nível 4. . . . .	39
4.3	Cartas de funcionário: cada funcionário possui um nome, seu nível de habilidades e a representação de quais são estas habilidades, uma imagem ilustrativa de uma pessoa, seu valor salarial, e sua reputação. . . . .	40
4.4	Cartas de projeto: Os projetos contém título, uma faixa de valores que difere para cada projeto, a árvore de estrutura de atividades, e suas pontuações. . . . .	41
4.5	Cartas de projeto: Os projetos estão dispostos com suas respectivas atividades de acordo com a demanda de nível que cada projeto exige. . . . .	42
4.6	Trecho do código em <i>React</i> que faz a montagem do elemento da carta de atividade do <i>Bootstrapping</i> . . . . .	44
4.7	Trecho do código em <i>React</i> que faz a montagem do elemento da carta de funcionário do <i>Bootstrapping</i> . . . . .	44
4.8	Parte da página no navegador que contém todos os elementos do <i>Bootstrapping</i> . No quadro do navegador é possível ver os seguintes elementos: frente e verso da carta de área escondida; imagem do verso das cartas com o nome do <i>Bootstrapping</i> ; frente e verso da carta de guia; e cartas de funcionários. . . . .	45
4.9	Trecho do código utilizando a <i>Application Programming Interface</i> (API) do <i>Puppeteer</i> para fazer a geração das imagens de cada carta do <i>Bootstrapping</i> . . . . .	45
4.10	Legenda do gráfico referente às respostas das perguntas do formulário. . . . .	46
4.11	Resultado MEEGA+ Seção: Estética. . . . .	47
4.12	Resultado MEEGA+ Seção: Aprendizibilidade. . . . .	47
4.13	Resultado MEEGA+ Seção: Operabilidade. . . . .	48
4.14	Resultado MEEGA+ Seção: Acessibilidade. . . . .	48
4.15	Resultado MEEGA+ Seção: Confiança. . . . .	49

4.16	Resultado MEEGA+ Seção: Desafio. . . . .	49
4.17	Resultado MEEGA+ Seção: Satisfação. . . . .	50
4.18	Resultado MEEGA+ Seção: Interação social. . . . .	50
4.19	Resultado MEEGA+ Seção: Diversão. . . . .	51
4.20	Resultado MEEGA+ Seção: Atenção focada. . . . .	51
4.21	Resultado MEEGA+ Seção: Relevância. . . . .	52
4.22	Resultado MEEGA+ Seção: Percepção de Aprendizagem. . . . .	53

## Siglas

**API** *Application Programming Interface*. 7, 43–45

**CSS** *Cascading Style Sheet*. 43

**DBR** *Design Based Research*. 27, 29

**DCC** Departamento de Ciência da Computação. 46

**HTML** *HyperText Markup Language*. 43

**ITS** *Intelligent Tutoring Systems*. 11

**JS** *JavaScript*. 43

**MDA** *Mechanics, Dynamics and Aesthetics*. 5, 7, 22, 23, 29, 33

**MEEGA** *Model for the Evaluation of Educational Games*. 27

**MEEGA+** *Model for the Evaluation of Educational Games*. 5, 7, 8, 27, 30, 31, 46–53

**PnP** *Problems and Programmers*. 5, 7, 21, 22

**SKG** *Software Kaihatsu Game*. 5, 7, 20, 22

**UFJF** Universidade Federal de Juiz de Fora. 38, 46

**UI** *User Interface*. 43

**URL** *Uniform Resource Locators*. 44

**VC** *venture capital*. 19

**XP** *Extreme Programming*. 14, 17, 20

# 1 Introdução

Um projeto bem sucedido é aquele que atende às necessidades do cliente, dentro dos limites de custos e prazo. Executar um projeto de *software* sem erros e sem atrasos é o objetivo de todas as empresas de tecnologia e também de seus funcionários. A necessidade de tornar esse processo mais fácil e confiável é evidente a cada dia. Gerenciamento de projetos é a aplicação de conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto a fim de cumprir seus requisitos (PMBOK, 2017).

Para se obter sucesso no projeto, um bom gerenciamento deve ser executado e existem vários modelos e metodologias para auxiliar o gerente a atingir seus objetivos. Nesse contexto, as metodologias ágeis são uma alternativa ao modelo tradicional de gestão de projetos, onde entregas parciais do produto são feitas ao longo de todo o projeto. O número de empresas utilizando metodologias ágeis em seus projetos cresce a cada ano (KUMAR; BHATIA, 2012).

Instituições de ensino de nível técnico e superior dispõem de disciplinas de gerenciamento de projetos nas quais os alunos exercitam o conhecimento necessário para poder executar seus projetos de forma eficiente. Porém, o ensino muitas vezes pode ser cansativo e desinteressante para o estudante por ser uma disciplina com bastante carga teórica expositiva e nem sempre existir a possibilidade de praticar os diversos métodos e técnicas em tempo hábil durante o curso (CARVALHO et al., 2018; SOUZA, 2018).

Existem vários métodos de ensino diferentes para o gerenciamento de projetos, e a avaliação dos resultados já vem escolhida previamente para que o aluno possa entender da melhor forma, porém, como os resultados não se alteram, qualquer situação diferente do que foi aprendido pode se tornar um grande dificultador para a conclusão do projeto devido o profissional não conhecer todas as possibilidades de resultados (PETRI et al., 2016; CARVALHO et al., 2018).

Como o tempo para estudo de gerenciamento de projetos é finito e, muitas vezes pequeno se comparado às demais disciplinas, a montagem de projetos piloto durante a disciplina pode se tornar inviável devido ao número de alunos e a duração

mínima do projeto para que se possa colocar em prática toda a teoria do ensino (KNOP, 2009). Isso exige diversas abordagens alternativas para, ao mesmo tempo, dar aos alunos o protagonismo de seu estudo e mantê-los motivados. Entre as abordagens, estão os simuladores de empresa e jogos de empresas (MAJER; DUDUCHI, 2015).

Estudos relacionados ao *design* de um ambiente de aprendizagem aprimorado por tecnologia cresceu ao longo dos anos (MAJER; DUDUCHI, 2015; ALKHATLAN; KALITA, 2018). Estes ambientes contam com a ajuda de tecnologia por meio dos quais os alunos possam adquirir habilidades ou conhecimento utilizando ferramentas de apoio à aprendizagem e recursos tecnológicos. Segundo Wang e Hannafin (2005), os métodos de pesquisa baseados em *design* vêm sendo muito utilizados, incluindo aprendizagem de investigação apoiada por tecnologia e ambientes de aprendizagem intencional suportados por computador, estes sistemas também são chamados de Sistemas Tutores Inteligentes, ou *Intelligent Tutoring Systems* (ITS) (ALKHATLAN; KALITA, 2018).

## 1.1 Descrição do Problema e Justificativa

Muitos projetos apresentam problemas durante sua elaboração, seja por atraso na entrega ou funcionalidades incorretas que podem até levar ao cancelamento do projeto. Entre as diversas causas, a inexperiência do gerente pode ser um fator determinante (STRIEGNITZ; LICHTER, 1996).

As instituições de ensino buscam cada vez mais melhorar seu processo de ensino, engajando os alunos de formas alternativas para tomar toda sua atenção e conseguir estimulá-los a estudar com mais afinco (CARVALHO et al., 2018). A aprendizagem através de uma ação ativa é bem mais eficiente, pois o estudante se sente incluído no ensino e demonstra maior interesse sobre o assunto (FALKEMBACH; GELLER; SILVEIRA, 2006; TAROUÇO et al., 2004). Os jogos de empresas e simuladores têm obtido bastante sucesso ao longo dos anos (MAJER; DUDUCHI, 2015). Um jogo com um objetivo bem definido, externo aos valores intrínsecos do mesmo, é conhecido como *jogo sério* e são de especial interesse deste trabalho.

Adotar um jogo que proporciona desafios, diversão e competitividade como forma de ensino traz benefícios para o aprendiz pois proporciona uma experiência

agradável e envolvente aos alunos, além de incentivar a interação social estimulando a cooperação e compartilhamento de ideias entre os participantes (PETRI et al., 2018).

## 1.2 Objetivos

O objetivo principal deste trabalho é melhorar o ensino e a prática do gerenciamento de projetos, com especial destaque para simular uma fábrica de *software* em seus anos iniciais, através de um jogo sério<sup>1</sup> para estudantes e gerentes em treinamento.

Para alcançá-lo, alguns objetivos específicos se fazem necessários:

- Criar um modelo que replique, com certa liberdade de abstração, os principais comportamentos relacionados a gerenciamento de projetos observados na literatura e estudos de caso;
- O jogador deve atuar na maior parte do tempo como gerente de projetos enquanto que as decisões operativas devem ser abstraídas pelos mecanismos básicos de jogo;
- Conseguir ajustar os mecanismos, interações e conceitos de forma simples o suficiente para que possam ser utilizados como uma dinâmica dentro da sala de aula, com apoio de *software* ou não;
- Observar se durante uma partida completa o jogo permite aos jogadores experimentar estratégias que lembrem das diversas abordagens de gerência;
- Avaliar o jogo criado quanto à sua experiência, usabilidade e percepção de aprendizagem do ponto de vista dos estudantes.

Este trabalho está organizado em cinco capítulos. Esta introdução é nosso Capítulo 1 e é seguida pela fundamentação teórica no Capítulo 2. O Capítulo 3 descreve o método utilizado nesta investigação e o Capítulo 4 seu desenvolvimento. Por fim, o Capítulo 5 traz nossas considerações finais.

---

<sup>1</sup>Jogos que, além do entretenimento, são utilizados com um propósito de ensino ou treinamento (ROCHA; BITTENCOURT; ISOTANI, 2015).

## 2 Fundamentação Teórica

Neste capítulo iremos apresentar algumas informações relacionadas ao desenvolvimento de *software* e o uso de jogos como ferramentas educacionais e trabalhos relacionados ao tema. Além disso, também serão conceituados alguns métodos e modelos de gerenciamento de projeto.

### 2.1 Processos de *Software*

Segundo Sommerville (2011), um processo de *software* é um conjunto de atividades relacionadas que levam à produção de um produto de *software*. A diversidade de processos de *software* é grande e cada um possui suas diferentes interpretações, porém, todos devem conter quatro atividades fundamentais para a engenharia de *software*: especificidade de *software*; projeto e implementação de *software*; validação de *software*; evolução de *software*.

A especificação de *software* define as funcionalidades do *software* e suas restrições. O projeto e implementação de *software* garante que as especificações foram atendidas. A validação do *software* assegura que as demandas do cliente foram supridas. A evolução de *software* permite que o sistema evolua de acordo com as necessidades do cliente (SOMMERVILLE, 2011).

Os sistemas de *software* atuais normalmente são complexos de tal modo que precisam ser desenvolvidos em equipes. Portanto, para produzir *software* com qualidade e produtividade, um mínimo de ordenamento deve ser adotado. Dessa forma, processos de *software* são formas para coordenar, motivar, organizar e avaliar o trabalho dos desenvolvedores, garantindo que o trabalho seja feito com produtividade e que estejam de acordo com os objetivos da empresa. Além disso, processos são importante pois orienta os desenvolvedores quanto às tarefas a serem desempenhadas e mantém o alinhamento entre os membros da equipe (VALENTE, 2020).

## 2.2 Modelos de Processo de *Software*

Sommerville (2011) define que um modelo de processo de *software* é uma representação simplificada de um processo de *software*, onde cada modelo retrata um panorama individual de cada processo. O modelo de processo de *software* estabelece o fluxo das atividades, ações e tarefas, o grau de iteração, os artefatos e a organização do processo a ser realizado (PRESSMAN; MAXIM, 2016). Entre os modelos iniciais, tradicionais e mais atuais podemos listar: em cascata; incremental e evolucionário; processo unificado; e uma série de processos ágeis, como *Extreme Programming* (XP) e *Scrum*, que serão citados neste trabalho.

O modelo em cascata é um dos primeiros modelos da engenharia de *software* com o encadeamento entre as fases, como apresentando na Figura 2.1. Com uma forte dependência entre duas etapas, onde uma só poderá ser iniciada após o término da anterior, tem a vantagem de deixar o processo visível, de fácil monitoramento. Mas como a desvantagem de que a versão final do *software* só pode ser apresentada na última etapa do processo, ou seja, caso alguma alteração proposta pelo cliente seja necessária, esta só será atendida após a entrega (SOMMERVILLE, 2015).

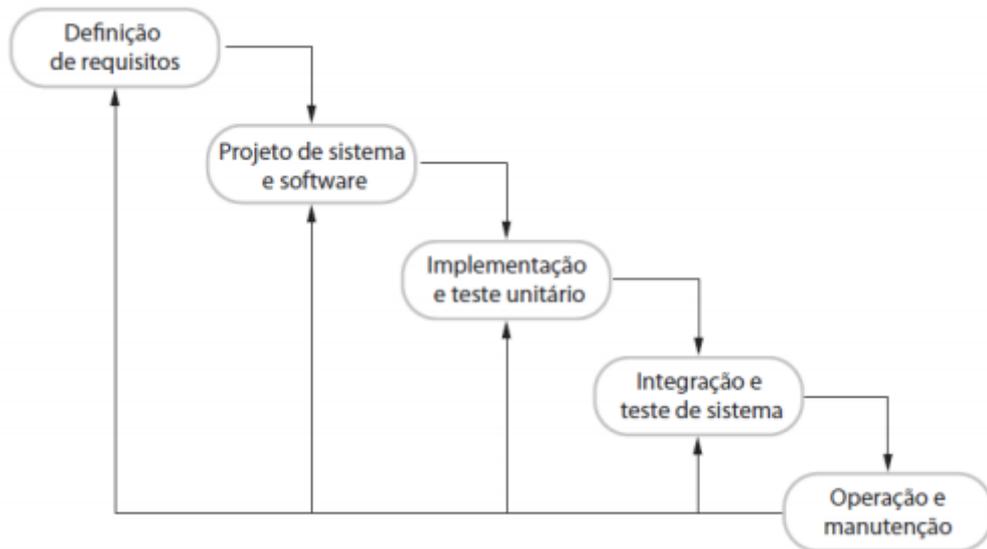


Figura 2.1: Fases do modelo de processo de *software* em cascata. Fonte: Sommerville (2011).

O modelo de desenvolvimento incremental (Figura 2.2) baseia-se em desenvolver uma implementação inicial e novas versões são elaboradas à partir de *feedbacks*

dos usuários (SOMMERVILLE, 2011). Em contrapartida, no modelo de desenvolvimento evolucionário o *software* é evoluído a cada iteração, adotando o modelo de Prototipação (Figura 2.3), onde é gerado um protótipo baseado nos requisitos iniciais e sua evolução ocorre posteriormente após a avaliação do cliente, ou no modelo Espiral (Figura 2.4), que faz a união da natureza iterativa da prototipação com os aspectos sistemáticos e controlados do modelo em cascata (PRESSMAN; MAXIM, 2016).

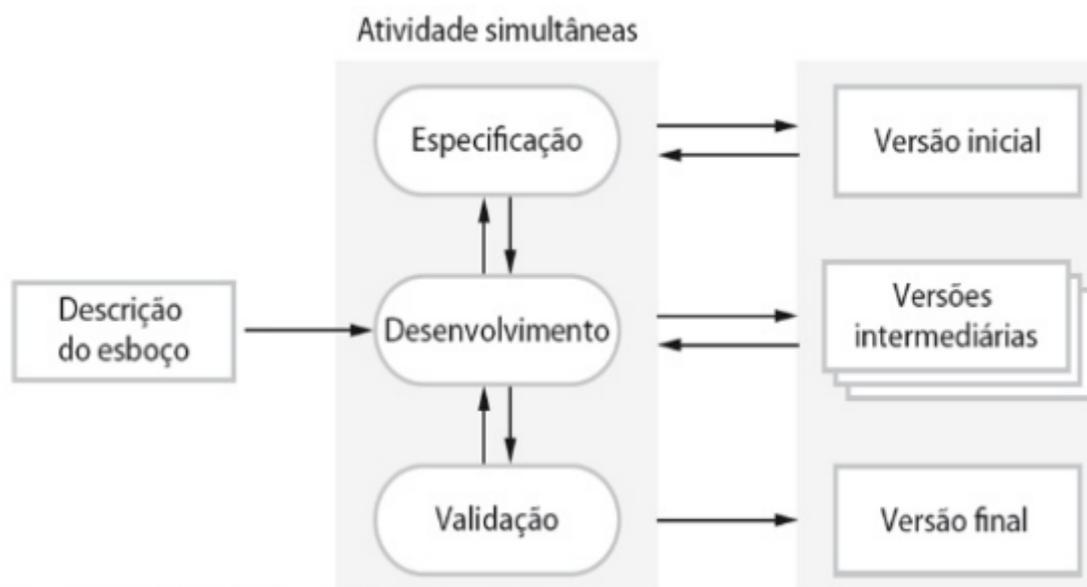


Figura 2.2: Modelo Incremental. Fonte: Sommerville (2011).

O Processo Unificado é uma tentativa de unir modelos tradicionais de processos de *software* com princípios de metodologias ágeis. Este modelo apresenta quatro fases: fase de concepção, que caracteriza-se pela identificação das entidades externas que irão interagir com o sistema; fase de elaboração, onde são definidos os requisitos do sistema; fase de construção, que é quando as partes do sistema são desenvolvidas; e a fase de transição, onde é feita a transferência do sistema do ambiente de desenvolvimento para o ambiente do cliente (SOMMERVILLE, 2011).

## 2.3 Metodologias Ágeis

Anos após o surgimento dos primeiros processos de desenvolvimento de *softwares* sequenciais, percebeu-se que a dinâmica de produção de um *software* é diferente ao observar que os cronogramas e orçamentos não são respeitados, e também a ocorrência de cancelamento.



Figura 2.3: Modelo de Prototipação. Fonte: Pressman e Maxim (2016).

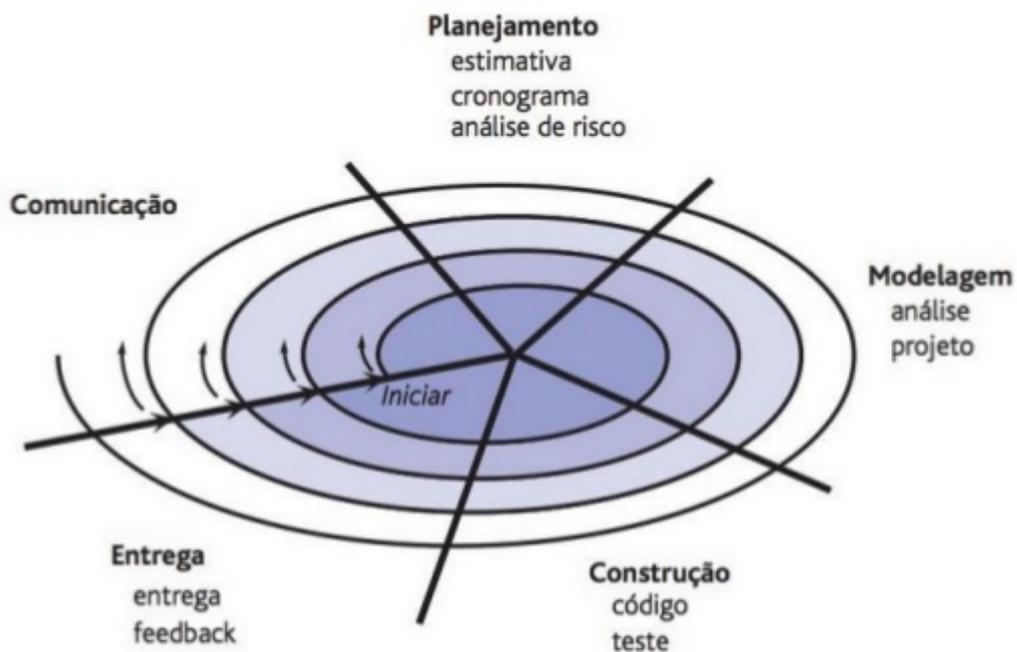


Figura 2.4: Modelo Espiral. Fonte: Pressman e Maxim (2016).

lamentos de projetos após dedicarem muito tempo e esforço ao projeto. Com isso, surgiu a necessidade de criação de uma alternativa a esses tipos de processos, surgindo assim os métodos ágeis (VALENTE, 2020).

Métodos ágeis caracterizam-se por pequenos incrementos no projeto onde, normalmente, novas versões do sistema são criadas e disponibilizadas a cada duas ou três semanas. A comunicação com o cliente é constante para que o mesmo dê retorno sobre a evolução e garanta que os requisitos sejam atendidos com maior rapidez (SOMMERVILLE, 2011).

O movimento teve início em 2001, com renomados desenvolvedores, autores e consultores da área de *software* ao assinarem o “Manifesto para o Desenvolvimento Ágil de *Software*”, Beck et al. (2001). O manifesto afirma:

Estamos descobrindo maneiras melhores de desenvolver *software*, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, passamos a valorizar:

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

*Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.*

Uma das metodologias ágeis existente é o *Extreme Programming* (XP), que é uma metodologia orientada a objetos e envolve uma série de regras e práticas abordadas em quatro atividades: planejamento, projeto, codificação e testes (Figura 2.5) (PRESSMAN; MAXIM, 2016).

A primeira atividade consiste em levantar os requisitos do sistema e na criação de histórias de usuários que descreve o resultado esperado, características e funcionalidades do sistema. A fase seguinte segue o princípio de ter um projeto simples ao invés de um projeto mais complexo. A fase de codificação consiste na criação do código, caracterizado pela *programação em pares*, de uma história. Por fim, a ultima fase é o momento onde todos os testes projetados para o sistema são executados.

Outra metodologia ágil muito utilizada é o Scrum, que é um *framework* para desenvolver, entregar e manter produtos complexos (SCHWABER; SUTHERLAND, 2017).

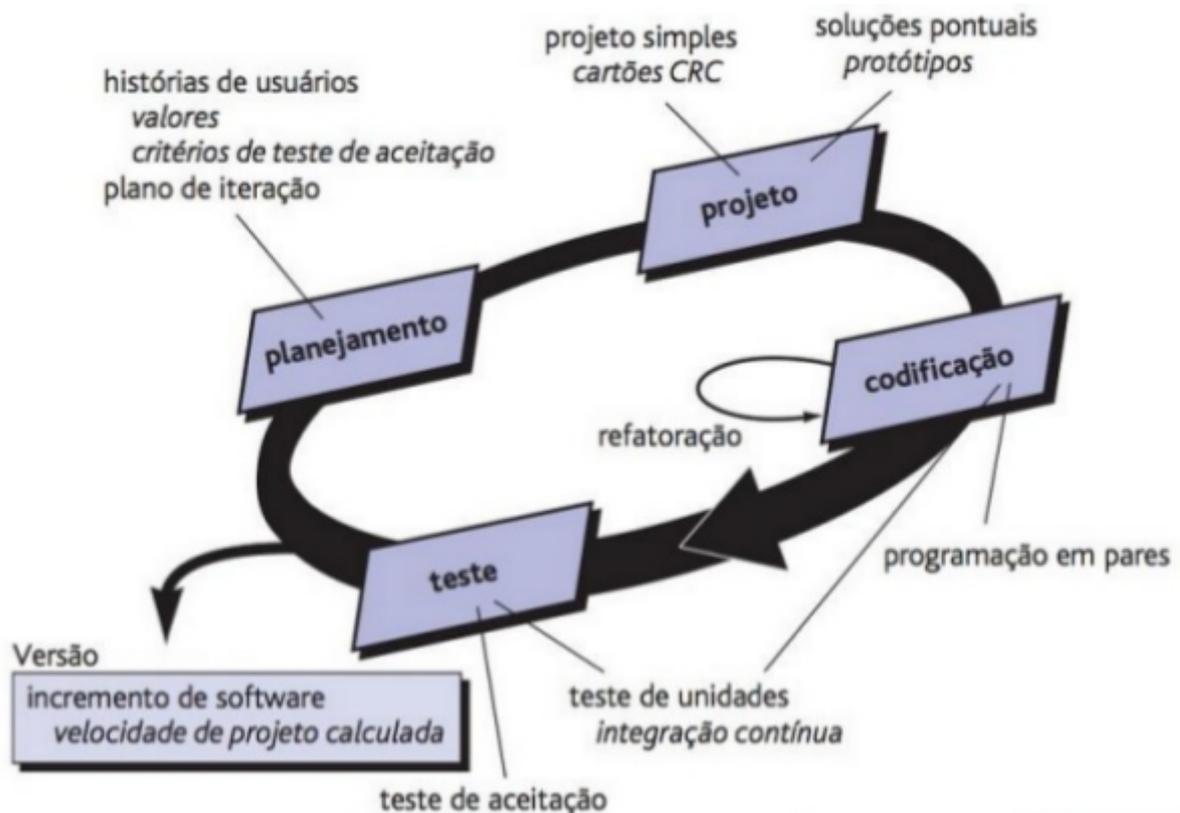


Figura 2.5: *Extreme Programming*. Fonte: Pressman e Maxim (2016).

Segundo Schwaber e Sutherland (2017), o Scrum não é um processo, técnica ou método, mas sim um arcabouço onde se pode empregar vários processos ou técnicas, com a característica de seguir uma abordagem iterativa e incremental para melhorar a previsibilidade e controle de riscos.

No Scrum é feito inicialmente um planejamento geral para que os objetivos e arquitetura do *software* sejam estabelecidos. Posteriormente, segue-se um ciclo, denominado de *sprints*, para que sejam desenvolvidos os recursos do sistema, onde uma funcionalidade completa é entregue ao final de cada ciclo. Por fim, quando se encerra os ciclos, conclui-se a documentação e, assim, o projeto se encerra (SOMMERVILLE, 2011). Além disso, o Time Scrum é composto por um *Product Owner*, que representa o dono do produto; o Time de Desenvolvimento, composto pelos profissionais que realizam o trabalho de entregar a funcionalidade ao final de cada *sprint*; e o *Scrum Master*, que atua como o líder do time (SCHWABER; SUTHERLAND, 2017).

## 2.4 Empreendedorismo e *Software*

O termo “empreendedor” tem origem francesa e significa “assumir riscos e começar algo novo”. Contudo, o termo “empreendedorismo” se originou ao distinguir o empreendedor do capitalista, ou seja, pessoas que assumem riscos das que fornecem capital (BRITO; PEREIRA; LINARD, 2013). O conceito de empreendedorismo vem sendo difundido e cresce a cada ano (BATTISTI; WEINZIERL, 2016; FERRARI, 2009), e, por conta dessa crescente, o seu ensino se torna cada vez mais presente em escolas e universidades onde seu propósito é ampliar a visão do estudante para novas perspectivas na carreira profissional (BATTISTI; WEINZIERL, 2016).

Seguindo o conceito de empreendedorismo, existem empresas que, de forma inovadora, em sua fase embrionária, possuem grandes chances de crescimento com recursos reduzidos, as chamadas *startups* (SILVEIRA; PASSOS; MARTINS, 2017). *Startups* são consideradas empresas de base tecnológica, que possuem estratégia competitiva devido suas inovações tecnológicas. Uma de suas principais características está no caráter de organização com potencial de rápido desenvolvimento, visando modelos de negócios que possam tornar-se repetível e escalável a baixo custo (SILVEIRA; PASSOS; MARTINS, 2017).

Muitas *startups* iniciam sua jornada contando com o suporte de investidores externos, como fundos de *venture capital* (VC), aceleradoras, *equity crowdfunding*, incubadoras, *corporate venturing* e grupos de investimento-anjo (SARFATI, 2018). Contudo, as fases iniciais da empresa, apesar de gerar receita, é muito comum gerar prejuízos, tornando-as empresas de alto risco, o que dificulta a captação de investimentos. Com isso, o empreendedor acaba optando pelo *Bootstrapping*.

O *Bootstrapping* é um termo usado quando o empreendedor inicia seu negócio com seus próprios meios financeiros, sem investimentos de capitais de risco (SARFATI, 2018). Portanto, o *Bootstrapping* acaba por ser o único meio que o empreendedor encontra para abrir seu empreendimento, visto que investimentos externos só começarão a surgir quando a empresa estiver em um estágio onde seus produtos estão minimamente viáveis, gerando receita e aumentando a quantidade de clientes (SARFATI, 2018).

## 2.5 Uso de jogos para ensino de Engenharia de *Software*

Jogos são atividades que possuem como principal objetivo o entretenimento (HUIZINGA, 1971). Durante o jogo, cria-se um mundo fantasioso, onde os comportamentos, valores e objetivos são definidos pelas regras previamente estabelecidas (EHRMANN; LEWIS; LEWIS, 1968).

Nesse contexto, existe a possibilidade de utilizar jogos de forma recreativa, visto que o interesse e participação na atividade é maior e isto se torna relevante para o ensino (ROCHA, 2018). Deste modo, os jogos sérios são utilizados de forma que seu propósito não seja puramente o entretenimento, mas sim o educacional (ABT, 1970; MICHAEL; CHEN, 2005).

### 2.5.1 SimSE

O SimSE é um jogo de simulação de engenharia de *software* educacional implementado com o intuito de ensinar aos alunos os processos de engenharia de *software* (NAVARRO, 2006). No jogo, o jogador assume o papel de gerente de projeto de uma equipe de desenvolvedores, assumindo funções como contratar e demitir funcionários, atribuir tarefas a seus funcionários, monitorar seu progresso e comprar ferramentas. O SimSE possui uma interface gráfica (Figura 2.6) onde é exibido o escritório do usuário, contendo mesas, cadeiras, computadores, entre outros itens. Além disso, a interface também contém as informações sobre funcionários, artefatos, clientes, projetos e ferramentas.

Um dos objetivos do SimSE é permitir que o usuário tenha a liberdade de personalizar os modelos de processos de *software*. Além disso, o jogo permite utilizar os seis modelos de processos de *software*, entre eles estão os modelos cascata, incremental, modelo unificado e XP.

### 2.5.2 *Software Kaihatsu Game*

O principal objetivo do *Software Kaihatsu Game* (SKG) é auxiliar, principalmente, engenheiros da área de *software* no entendimento do gerenciamento de projetos (HAMADA; HIJI; KANEKO, 2014).

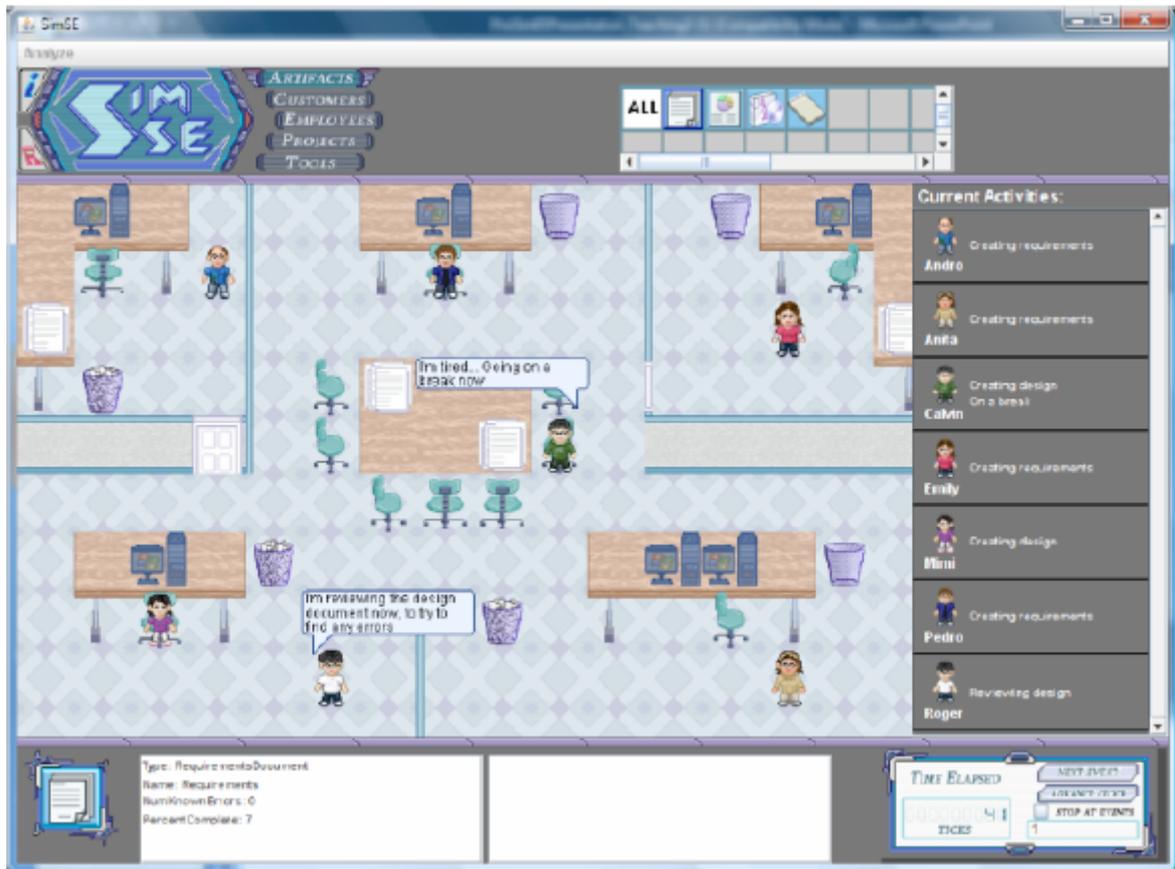


Figura 2.6: Interface do jogo SimSE

O jogo busca aumentar o entendimento em gerenciamento de projetos, alocação de recursos humanos e financeiros, e aumentar a experiência e habilidades envolvidas no gerenciamento de projetos. A Figura 2.7 apresenta alguns elementos da interface do jogo.

### 2.5.3 *Problems and Programmers*

O *Problems and Programmers* (PnP) é um jogo de cartas que visa o ensino de engenharia de *software* e tem como principal objetivo enriquecer o entendimento dos estudantes, simulando o processo de desenvolvimento de *softwares* desde a concepção até a fase final de entrega de forma competitiva. (BAKER; NAVARRO; HOEK, 2003).

Neste jogo, dois ou mais jogadores competem entre si com o objetivo de concluir primeiro o jogo. Para isso, os jogadores utilizam cartas de conceito buscando evitar os problemas, estas cartas podem ser vistas na Figura 2.8. O intuito é que os jogadores aprendam a relacionar as ações positivas executadas no jogo com situações reais de desenvolvimento de *softwares*.



Figura 2.7: Interface do jogo SKG



Figura 2.8: Exemplos de carta de projeto, carta de programador, carta de problema e carta de conceito do jogo PnP

## 2.6 Projeto, desenvolvimento e teste de jogos

Nesta seção será apresentado como o jogo pode ser projeto, desenvolvido e testado.

### 2.6.1 Modelo MDA: *Mechanics, Dynamics and Aesthetics*

Jogos são criados por desenvolvedores ou *designers* e são consumidos por jogadores. O MDA formaliza o consumo de jogos dividindo em componentes distintos: Regras, Sistemas e “Diversão”. E estabelece suas contrapartes de *design*: Mecânicas (*Mechanics*), Dinâmicas (*Dynamics*) e Estética (*Aesthetics*). A associação dos conceitos do modelo

estão ilustradas, de forma adaptada, na Figura 2.9. O MDA é uma abordagem formal para o entendimento de jogos, onde acadêmicos e profissionais consigam se entender (HUNNICKE; LEBLANC; ZUBEK, 2004). Os elementos associados são definidos pelos autores da seguinte forma:



Figura 2.9: Associação de conceitos do modelo MDA. Figura adaptada de Zaffari e Battaiola (2014).

**Mecânica** descreve os componentes específicos do jogo, no nível da representação de dados e algoritmos.

**Dinâmica** descreve o comportamento em tempo real da mecânica que atua nas entradas do jogador e nas saídas uns dos outros ao longo do tempo.

**Estética** descreve as respostas emocionais desejáveis ocorridas no jogador, quando ele interage com o sistema de jogo.

Os autores explicam que considerar as perspectivas tanto dos desenvolvedores quanto dos jogadores ajuda a perceber como pequenas mudanças em uma camada podem afetar outras. A Figura 2.10 ilustra a representação de perspectivas entre desenvolvedores e jogadores.

## 2.6.2 Estética

A ideia da definição de estética é fugir de palavras como “diversão” e “gameplay” mas se relacionar com um vocabulário mais direcionado. Tal vocabulário contém algumas palavras como:

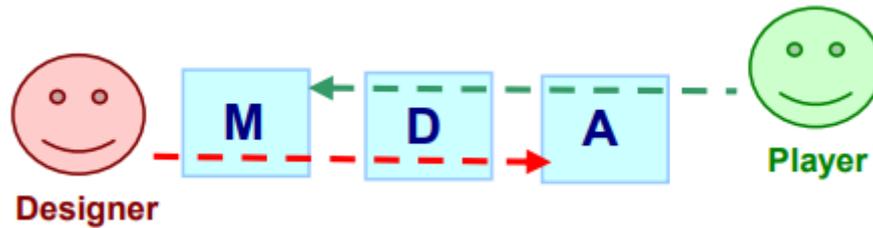


Figura 2.10: Representação das perspectivas do *designer* e do jogador. Fonte: Hunicke, LeBlanc e Zubek (2004)

1. **Sensação** Jogo como prazer dos sentidos.
2. **Fantasia** Jogo como faz de conta.
3. **Narrativa** Jogo como drama.
4. **Desafio** Jogo como corrida de obstáculos.
5. **Amizade** Jogo como ferramenta social.
6. **Descoberta** Jogo como território inexplorado.
7. **Expressão** Jogo como autoconhecimento.
8. **Submissão** Jogo como passatempo.

Usando a estética como um guia pode-se definir modelos de jogabilidade que auxiliam a descrever as mecânicas e as dinâmicas.

### 2.6.3 Dinâmica

Hunicke, LeBlanc e Zubek (2004) definem que a dinâmica trabalha para criar as experiências da estética. Zaffari e Battaiola (2014) usam o jogo de xadrez para exemplificar a experiência de competição para a dinâmica do jogo, onde uma mecânica de pontuação poderia ser atribuída às jogadas de captura, onde capturar um peão valeria 10 pontos, e uma rainha, 50 pontos. Assim, os jogadores poderiam elaborar estratégias mais agressivas para obter as peças do adversário.

## 2.6.4 Mecânica

A mecânica são as várias ações, comportamentos, e controle dos mecanismos oferecidas ao jogador dentro do contexto do jogo. Por exemplo, a mecânica de jogos de tiro inclui armas, munição e pontos de *spawn*<sup>2</sup>.

## 2.6.5 Jogos modernos de tabuleiro

Jogos de tabuleiros modernos tem como foco o perfil tático ou estratégico durante suas partidas, englobando uma série de relações e dinâmicas. Estes jogos também são conhecidos como *designer's games* por trazerem o nome do autor na capa (WOODS, 2012). Após o lançamento de Colonizadores de Catan (TEUBER, 1995), os jogos de tabuleiros modernos se tornaram populares e atualmente existem sites focados em armazenar estes tipos de jogos, como o BoardGameGeek (BOARDGAMEGEEK, 2021).

Existem diversas maneiras de definir a criação de um jogo, e a metodologia de criação utilizada varia de acordo com o objetivo de cada jogo. Além disso, deve-se levar em consideração, durante a elaboração, que o jogo deve despertar um sentimento de conexão do jogador com o próprio jogo para que, assim, os resultados de aprendizagem sejam mais efetivos (HUNICKE; LEBLANC; ZUBEK, 2004; NICHOLSON, 2011). Nicholson (2011) define uma sequência de passos para a criação de um jogo de tabuleiro moderno onde o objetivo do jogo é a aprendizagem do estudante.

**Gerar Resultados de Aprendizagem Centrados no Aluno:** O primeiro passo no desenvolvimento de um jogo para aplicação em sala de aula é determinar o conteúdo que será ensinado ao aluno.

**Decidir os Elementos do Jogo:** O próximo passo é definir os elementos de experiência do jogo. As experiências do jogo envolve o próprio jogo, as interações entre os jogadores e o cenário do jogo. Os principais elementos a serem considerados são os cenários desejados, o número de jogadores e o tipo de envolvimento dos jogadores.

**Cenários:** É essencial considerar os cenários desejados. Se o jogo for aplicado em um ambiente silencioso então a interação entre jogadores não deve ser muito explorada,

---

<sup>2</sup>Pontos onde se localiza o surgimento de itens, personagens e elementos de um jogo.

enquanto para ambientes abertos esta interação pode ser aproveitada. Portanto, a interação entre jogadores e escolhas de componentes devem ser analisadas neste ponto.

**Público e Número de Jogadores:** Deve ser pensado para quem o jogo será desenvolvido. A primeira decisão é determinar se o jogo será jogado por apenas um jogador ou se um grupo de pessoas jogarão ao mesmo tempo.

**Tipo de Interação do Jogador:** Deve ser considerado qual o tipo de envolvimento do jogador que o jogo demanda. Deve ser analisado como os jogadores irão interagir.

**Documentação:** Todas as decisões devem ser documentadas para que qualquer envolvido com o jogo consiga ver as principais decisões feitas sobre o jogo e os motivos.

O próximo passo é integrar o conteúdo, e os resultados de aprendizagem podem ser um guia para definir qual conteúdo será mais adequado para o jogo.

**Vincular Desafio e Conteúdo:** O desafio é o centro de qualquer jogo e, para fazer com que o jogador se envolva completamente com o jogo, cada conteúdo do jogo deve conter um desafio.

**Criar Funções Apropriadas para o Contexto e Conteúdo:** Jogos costumam conter gerência de recursos, que pode ser custo, tempo, pessoas, propriedades, entre outros. Os jogadores devem decidir como gerenciar estes recursos. Portanto, a construção das funções do jogador devem estar de acordo com o contexto do desafio.

**Explorar as Possibilidades para os Mecanismos:** Os mecanismos do jogo servem como ferramenta para conectar diferentes elementos. Estes mecanismos permitem ao jogador explorar os desafios dentro de sua função. Após estes itens, um protótipo deve ser criado.

**Testar e Revisar:** Nesta etapa deve ser feito os testes de jogabilidade, e algumas regras para o jogo podem ser definidas.

**Coletar Resultados e *Feedbacks* dos Jogadores:** Uma vez concluído o jogo, ele pode ser testado em sala de aula. O *feedback* dos jogadores é importante para que se possa tornar as regras mais claras e tornar a eficiência maior.

## 2.7 Pesquisa Baseada em *Design*

O método de Pesquisa Baseada em *Design*, ou *Design Based Research* (DBR) (WANG; HANNAFIN, 2005) permite melhorar as práticas educacionais através de um ciclo iterativo entre análise, projeto, desenvolvimento e implementação que aproxima pesquisadores, praticantes e desenvolvedores. Seu ciclo básico de aplicação pode ser resumido como: identificar as dinâmicas para sustentar a atenção e motivação do aluno; criação e aplicação do protótipo inicial para testes; ajustes relevantes oriundos dos *feedbacks* dos testadores; aplicação do novo protótipo aos alunos; coleta e análise dos dados.

## 2.8 Avaliação de Jogos Educativos com o MEEGA+

Avaliar sistematicamente um jogo para o ensino em computação é essencial para garantir a qualidade e provar que o impacto esperado e/ou engajamento que ele se propõe foram atendidos (CAULFIELD et al., 2011).

Neste contexto, o modelo de avaliação de jogo *Model for the Evaluation of Educational Games* (MEEGA) (SAVI; WANGENHEIM; BORGATTO, 2011) é apontado como um dos mais utilizados (PETRI; WANGENHEIM; BORGATTO, 2017). Este modelo é desenvolvido para a avaliação de jogos educacionais por meio da aplicação de um questionário padronizado após os estudantes utilizarem o jogo. Entretanto, o modelo MEEGA apresenta algumas limitações em sua concepção e avaliação (PETRI; WANGENHEIM; BORGATTO, 2017). Diante disso, Petri, Wangenheim e Borgatto (2017) apresentam uma evolução do modelo MEEGA voltado para avaliação da qualidade de jogos educacionais para o ensino de computação, o modelo MEEGA+.

O modelo MEEGA+ tem como objetivo “analisar jogos educacionais com o propósito de avaliar a percepção da qualidade em termos de experiência do jogador e percepção da aprendizagem do ponto de vista de alunos e instrutores no contexto de

“cursos superiores da área de computação”, Petri, Wangenheim e Borgatto (2017).

A partir de fatores de qualidade, são fragmentadas dimensões onde algumas possuem semelhanças. Os fatores de qualidades são relacionados a motivação, experiência do usuário, usabilidade, engajamento e satisfação. Já entre as dimensões encontram-se: atenção focada, diversão, desafio, interação social, confiança, relevância, satisfação e usabilidade (PETRI; WANGENHEIM; BORGATTO, 2016).

A decomposição hierárquica dos fatores de qualidade em dimensões/subdimensões podem ser vistas na Figura 2.11.

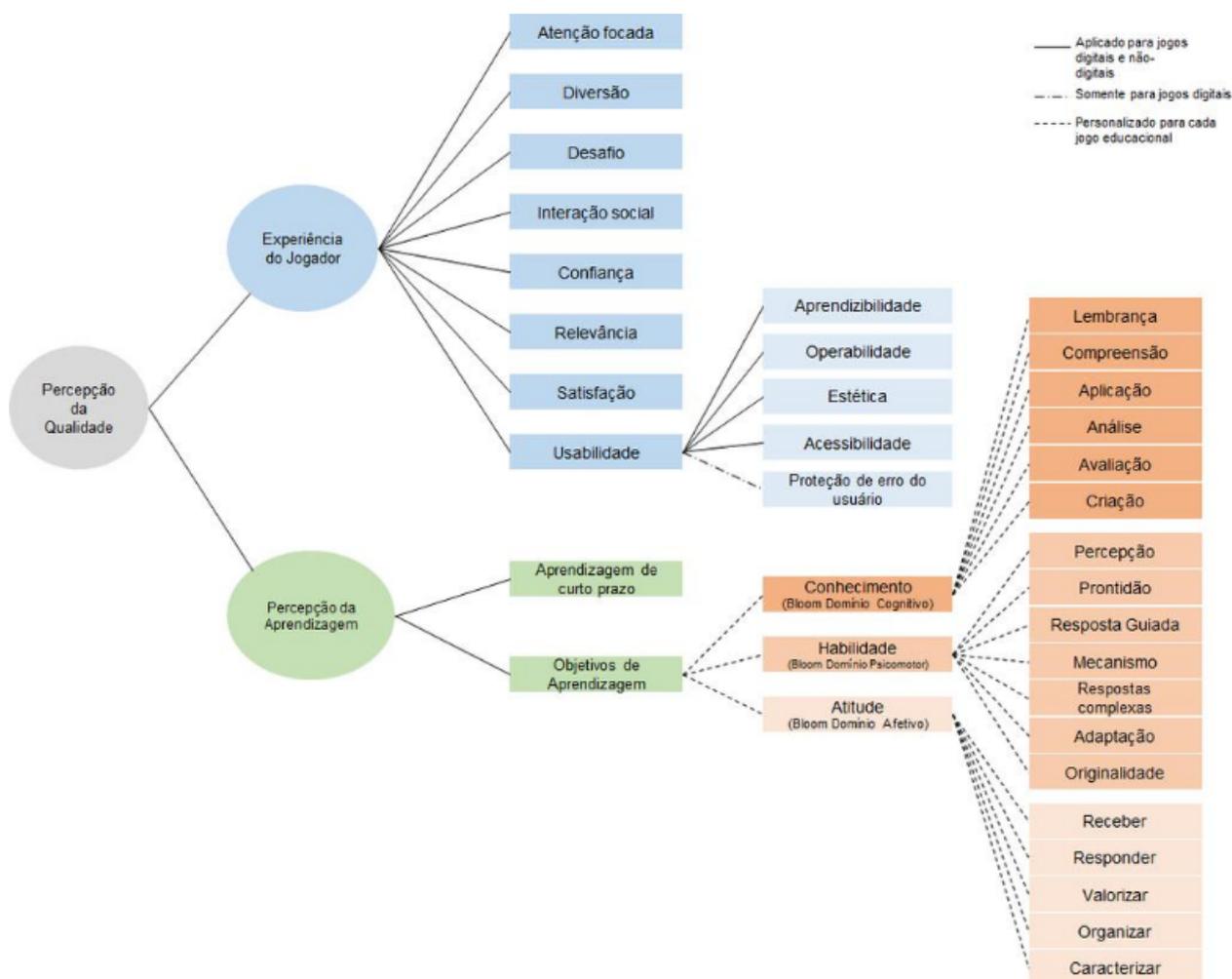


Figura 2.11: Hierarquia dos fatores de qualidade em dimensões/subdimensões (Petri, Wangenheim e Borgatto (2016)).

A avaliação deve ser realizada de forma rápida e não intrusiva, afetando minimamente o fluxo da aula. O jogo será aplicado aos alunos e, em seguida, um questionário deverá ser respondido. A meta de avaliação é avaliada com base nas percepções dos alunos através do questionário após a aplicação do jogo.

## 3 Método

Este capítulo descreve o método utilizado para o desenvolvimento das ferramentas, experimentos e sua respectiva condução para a conclusão deste trabalho.

### 3.1 Estrutura da pesquisa

Este trabalho foi desenvolvido em duas etapas distintas: a primeira focando em identificar um núcleo mínimo de jogabilidade; e a segunda nos aspectos educacionais. A condução foi baseada no método *Design Based Research* (DBR) (WANG; HANNAFIN, 2005) no qual o pesquisador também atua como *designer* e ciclos iterativos mesclam o projeto, refinamento e construção de conhecimento. O método é utilizado para práticas educacionais emergentes e foi utilizado para a montagem do protótipo de um jogo sério que foi denominado *Bootstrapping*.

A primeira etapa consistiu em identificar um conjunto de dinâmicas que manteriam a atenção e motivação dos alunos durante uma aplicação em sala de aula. Usando o MDA, uma visão partindo do ponto de vista do *designer*, para identificar mecanismos que conduziriam à dinâmica desejada.

1. Ideação do modelo de jogo;
2. Construção de um protótipo digital inicial;
3. Aplicação desse protótipo com um grupo de desenvolvedores para testes de jogabilidade;
4. Coleta de opinião imediata, ao final do teste;
5. Avaliação de ajustes necessário e reinício dos testes de jogabilidade;
6. Se o número de modificações urgentes ou médias não for significativo, iniciar um novo ciclo de testes;

A segunda etapa avançou para a aplicação mais próxima ao contexto educacional. O protótipo é aplicado e os conceitos reforçados. Novamente o ciclo de desenvolvimento iterativa se mantém. Nesta etapa, o método MEEGA+ é utilizado como coleta formal para avaliação do estado do projeto.

1. Aplicação do protótipo em ambiente educacional;
2. Coleta de dados, seguindo o método MEEGA+;
3. Análise e classificação das opiniões e sugestões;
4. Planejamento para alteração do protótipo em uma edição futura;
5. Se o número de modificações urgentes ou médias não for significativo, iniciar um novo ciclo de testes;

## 3.2 Ambiente para teste de jogabilidade

Para que fosse possível ter um ambiente de testes de forma rápida, optou-se por fazer o uso de uma plataforma digital para jogos digitais de tabuleiro, o Tabletopia<sup>3</sup>. Nesta plataforma é possível criar elementos para simular a física de um jogo de tabuleiro real, como marcadores, cartas e baralhos, fichas e moedas, entre outros. O ambiente permite uma prototipação rápida e também o acesso a jogadores remotamente, fundamental devido ao período de isolamento social em função da pandemia de COVID-19, no momento do desenvolvimento deste trabalho.

Com o protótipo criado, salas virtuais foram disponibilizadas para que os jogadores se conectassem e, assim, iniciar uma partida. Durante a partida, os jogadores puderam interagir com todos os elementos como se estivessem em uma mesa real. Entretanto, a manipulação digital impõe um custo na usabilidade, aumentando o tempo de partida e deixando a curva de aprendizado mais íngreme.

---

<sup>3</sup>Disponível gratuitamente em <http://tabletopia.com>.

### 3.3 Avaliação dos ciclos de jogabilidade

Para avaliar os ciclos de jogabilidade na etapa inicial, o autor aproximou-se de uma comunidade de desenvolvedores de jogos, a *Oficina do Playtest*. Os encontros são realizados semanalmente, e há um rodízio de quem apresentará o protótipo na semana. Apesar do encontro ser mais direcionado para a produção de produtos comerciais, vários dos participantes relataram estar envolvidos em projetos educacionais ou atuarem diretamente em atividade de fomento ao empreendedorismo com *startups*.

As sessões foram conduzidas com no máximo quatro jogadores e, ao final de cada encontro, coletamos os pontos negativos e positivos percebidos por cada jogador em relação à jogabilidade. Além disso, todas as partidas foram gravadas, com a autorização dos participantes, para que pudéssemos avaliar os pontos citados posteriormente. Um breve histórico das alterações nas regras estão disponíveis no Apêndice A.

### 3.4 Formulário para aplicação do MEEGA+

O MEEGA+ fornece um formulário padrão para ser aplicado aos participantes ao final da experiência com o jogo. Com isso, o participante responde perguntas que visam avaliar o jogo quanto alguns aspectos como: Estética; Aprendizabilidade; Operabilidade; Acessibilidade; Confiança; Desafio; Satisfação; Interação social; Diversão; Atenção focada; Relevância e Percepção de Aprendizagem.

Normalmente, este formulário é disponibilizado presencialmente, porém, como houve a necessidade da implementação digital para o *Bootstrapping*, o formulário foi adaptado para que a coleta das respostas ocorresse de forma *online*, com isso, foi criado um formulário digital contento todas as perguntas pertinentes ao MEEGA+. Com as respostas coletadas, fizemos o uso de uma planilha projetada<sup>4</sup> especificamente para o tratamento das respostas referente ao MEEGA+. Esta planilha dispõe de fórmulas para que sejam calculados os resultados, além de gráficos para tornar mais legível a visualização dos dados.

---

<sup>4</sup>Link para download dos resultados na planilha: ([https://1drv.ms/x/s!AsB-KRhKt8qJrizdnhDI-shiwED\\_](https://1drv.ms/x/s!AsB-KRhKt8qJrizdnhDI-shiwED_))

## **3.5 Acompanhamento de versões do protótipo**

O protótipo digital é montado a partir de um conjunto de imagens, porém, todos os componentes foram desenvolvidos à partir de programas e tecnologias para internet. Desta forma, todo o desenvolvimento foi gerenciado por controle de versão, permitindo experimentar caminhos, e retornar para versões anteriores caso necessário.

No Capítulo 4, a seguir, o desenvolvimento detalhado, das diversas etapas do projeto e os dados coletados são apresentados.

## 4 Desenvolvimento

Neste capítulo é detalhado todo o desenvolvimento das regras do jogo, da implementação digital, o ferramental computacional utilizado e o processo de avaliação e respectiva discussão dos resultados colhidos.

### 4.1 Projeto do jogo de tabuleiro *Bootstrapping*

O *Bootstrapping* foi desenvolvido usando uma visão do MDA, que apesar de bem simplificada, permite uma rápida descrição das intenções do projeto. Nas seções a seguir, essas decisões de projeto são explicadas.

#### 4.1.1 Estética

O *Bootstrapping* coloca o jogador no papel de um gerente de uma empresa de *software* com o objetivo de acumular reputação crescendo com a empresa e substituindo um investimento inicial por faturamento. As decisões principais giram em investir o dinheiro nos profissionais necessários para completar os projetos atuais, oferecer melhores preços observando a concorrência e cumprir as necessidades dos clientes dentro do prazo. Os prazos são de especial importância pois um atraso nas entregas gera gastos não previstos podendo ocasionar em dívidas para a empresa e penalidade na pontuação final.

Existirão funcionários com competências, habilidades e conhecimentos variados, e a alocação dos funcionários em cada tarefa afetará o tempo de execução desta atividade de acordo com a habilidade do funcionário, ou seja, se um funcionário possui uma maior habilidade em testes, então a alocação deste funcionário em uma função de desenvolvimento demandará mais habilidades deste funcionário para que a tarefa seja concluída, enquanto a sua alocação em uma atividade de teste, o esforço será menor e a conclusão poderá ser mais rápida.

A mão de obra é disputada ao oferecer melhores salários que os concorrentes. Cabe ao jogador observar a demanda de seus projetos e decidir se faz uma oferta com

valores altos ou baixos, pois um profissional com as qualificações necessárias pode ser a diferença entre um projeto entregue no prazo ou não.

Projetos são disputados por orçamentos, ou seja, os clientes trazem uma estimativa do tamanho do projeto e com uma faixa de valores que estão dispostos a pagar, contudo. O jogador que oferecer um orçamento mais baixo ganha a prioridade de escolha por um projeto que deseja trabalhar. Cabe aos jogadores estimar os valores que beneficiam suas empresas através da entrega do projeto a medida que são finalizados. A diferença entre um fazer um bom negócio ou não está na eficiência e adaptabilidade.

O objetivo final é ter uma empresa com boa reputação no mercado. A reputação é uma função direta do número, tamanho e qualidade dos projetos entregues, bem como tamanho da equipe, e dinheiro em caixa ao final da partida. Projetos entregues no prazo aumentam a reputação da empresa. Já projetos entregues fora do prazo baixam a reputação da empresa.

Dessa forma, o *Bootstrapping* compõe sua estética na qual o jogador poderá experimentar alguns elementos, ainda que muito simplificados, de como é criar uma empresa e administrar uma equipe em constante mudança, em um mundo competitivo e que exige rápida adaptação.

### 4.1.2 Mecanismos

O *Bootstrapping* possui os conceitos básicos que são utilizados ao longo do jogo: reputação, dinheiro, equipe, projeto, atividades, tarefas, habilidades, dependências, atrasos, empréstimos, custos. Esses conceitos podem ser mapeados de forma reduzida como na Figura 4.1.

Cada jogador possui uma quantidade de dinheiro inicial. Esse dinheiro é utilizado para pagar os custos, que vêm de salários dos funcionários e empréstimos. O dinheiro é recebido ao completar atividades e quando o projeto é finalizado.

O projeto é dividido em atividades, cada atividade possui um conjunto de tarefas para ser realizada, e cada tarefa pode ou não ser de tipos diferentes. Uma atividade pode ter outras atividades como pré-requisito, e o projeto só é finalizado quando todas as suas atividades são completadas.

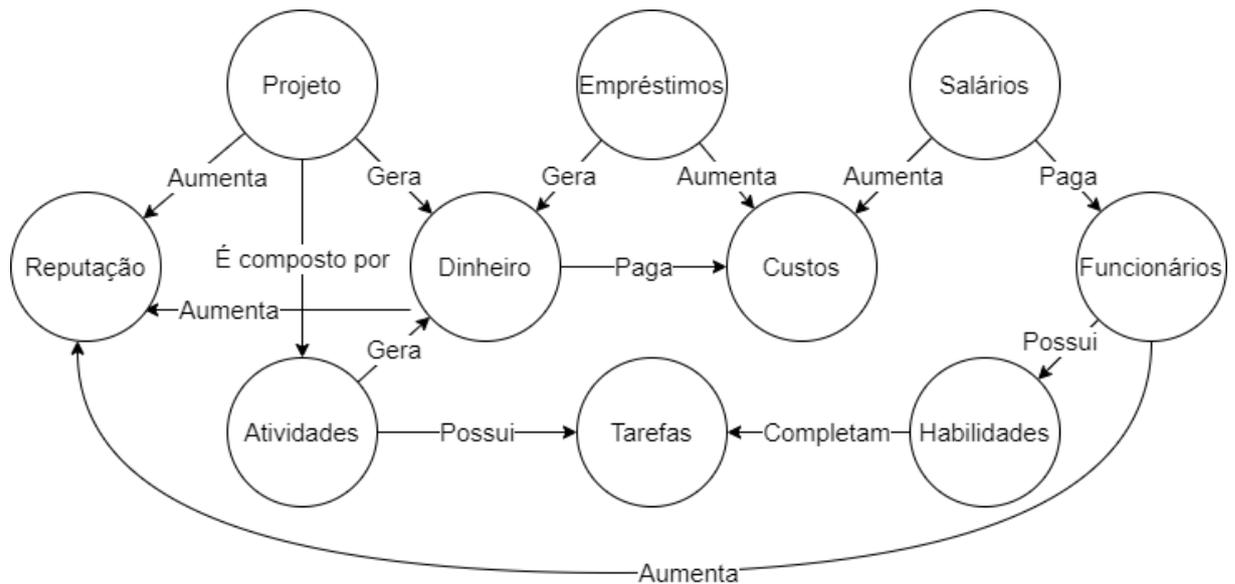


Figura 4.1: Principais conceitos e inter-relações do jogo *Bootstrapping*.

Os funcionários possuem habilidades, com os mesmos tipos das tarefas presentes nas atividades. Para completar uma tarefa, um funcionário com as habilidades de mesmo tipo deve ser alocado para trabalhar nela. Uma tarefa pode ser completada com três habilidades de qualquer tipo, fazendo a proporção 3:1, como uma espécie de “curinga”.

O jogo utiliza o mecanismo de alocação de trabalhadores: a cada rodada o gerente aloca seus funcionários para cumprir as tarefas nas atividades, de acordo com o tipo de habilidade que a tarefa demanda ou fazendo o uso da habilidade “curinga”.

O gerente pode contratar novos funcionários, aumentando os custos mensais e a quantidade de tarefas que é capaz de resolver. Quando necessário, pode adquirir empréstimos, que são quantidades de dinheiro imediata, mas adicionam um custo mensal extra até serem quitados.

Cada funcionário possui um valor mínimo para contratação, que é definido de acordo com suas habilidades. O gerente poderá fazer ofertas para um funcionário para poder contratá-lo. Todavia, o jogador que oferecer a maior quantia entre os lances terá a prioridade de escolha por um funcionário, desde que a quantia seja igual ou maior que o valor mínimo de contratação. Além disso, ao final de cada mês, o funcionário deverá receber seu salário, que é seu valor mínimo de contratação. Na atual versão do jogo, o excesso no salário do primeiro mês é pago como um bônus uma única vez durante a oferta

de emprego, ou seja, não é necessário fazer o pagamento salarial do funcionário recém contratado ao final do seu primeiro mês na empresa.

Cada projeto possui uma faixa de valores, possuindo um valor mínimo e um valor máximo disponível para orçamento, que é definido de acordo com a quantidade de atividades que possui. O gerente poderá fazer ofertas de orçamento para um projeto para poder trabalhar e receber o dinheiro orçado através da conclusão das atividades ou ao finalizar o projeto. Ao concluir uma atividade, o gerente receberá a quantia referente ao nível da atividade, ou seja, ao concluir uma atividade de nível 2, a empresa receberá \$2. Contudo, o jogador que oferecer a menor quantia entre os orçamentos terá a prioridade de escolha por um projeto, desde que a quantia seja maior ou igual ao valor mínimo e menor ou igual ao valor máximo do projeto. Caso o jogador não entregue uma atividade finalizada ao final do mês, receberá uma penalidade de atraso. Na versão atual do jogo, o jogador receberá um bônus ao adquirir um projeto no leilão, fazendo com que não seja necessário entregar uma atividade ao final do primeiro mês.

Existem quatro tipos de habilidades: *Desenvolvimento* (vermelho); *Testes* (verde); *Banco de Dados* (azul); e *DevOps* (rosa). Os funcionários podem ter de uma a três habilidades, em qualquer combinação. Alguns funcionários possuem mais de uma habilidade de mesmo tipo (especialistas), enquanto outros com mais de um tipo de habilidade (generalistas), tornando-o útil para vários tipos de tarefas.

### 4.1.3 Dinâmica

A dinâmica do jogo envolve em saber como propor um orçamento de acordo com o intervalo de valores definido para o projeto mas sem ter a informação completa sobre as tarefas das atividades necessárias. Os jogadores vão disputar os projetos e funcionários, utilizando uma oferta compartilhada exigindo a otimização dos recursos disponíveis. A importância do funcionário para cada empresa vai depender dos projetos que estiverem em desenvolvimento e sua equipe atual. Essa assimetria é fundamental para jogos de mercado compartilhado e a habilidade de valorar os elementos de jogo se desenvolve ao longo da partida.

Para contratar um funcionário, os jogadores disputarão entre si em um leilão se-

creto, ou seja, cada jogador deverá informar um valor que deseja, e não revelar a quantia. Após todos terem indicado seus valores, todos irão revelar simultaneamente suas ofertas e o jogador que tiver ofertado o maior valor terá a preferência de escolha por um funcionário que deseja contratar, ou seja, a ordem de escolha segue do maior lance para o menor. Porém, a oferta de valor deve ser igual ou maior que valor salarial do funcionário informado na carta, sendo assim, se um jogador ofertar uma quantia abaixo do mínimo indicado na carta de qualquer funcionário que está disponível para contratação, este não poderá contratar nenhum funcionário na rodada. Deste modo, o jogador poderá decidir entre tentar a contratação de um funcionário importante para o momento do jogo, onde possivelmente terá que fazer um lance de valor alto para conseguir a prioridade de escolha, ou fazer uma oferta para um funcionário que esteja mais adequado ao seu capital. Contudo, o jogador deve escolher sabiamente o valor a ser ofertado pois uma escolha equivocada de investimento pode ceder a prioridade para outro jogador e, por consequência, não conseguir a contratação de nenhum funcionário naquela rodada.

A disputa para um projeto ocorre de forma semelhante à contratação de funcionários, também sendo em formato de leilão secreto, porém, neste caso, o jogador que oferece o menor valor terá a preferência de escolha para um projeto, isto é, a ordem de escolha do projeto segue do menor lance para o maior. Da mesma forma, o valor ofertado deve estar dentro do intervalo de valores do projeto indicado na carta, isto é, maior ou igual ao valor mínimo ou menor ou igual ao valor máximo indicados. Sendo assim, se o jogador oferecer um valor menor que o mínimo ou maior que o máximo indicados na carta do projeto, não será possível adquirir aquele projeto para se trabalhar. Além disso, o valor que o jogador ofertar será a quantia de dinheiro total que ele receberá ao finalizar o projeto. Assim sendo, o jogador precisará ofertar um valor que o deixe em vantagem para escolher um projeto ao passo que também seja um valor rentável para sua empresa. Caso seja oferecido um valor que esteja fora do intervalo de valores de todos os projetos disponíveis para o leilão, o jogador não poderá pegar nenhum projeto naquela rodada.

Quando ocorre empate nas ofertas tanto para funcionários quanto para projetos, ou seja, quando dois ou mais jogadores ofertam valores iguais no leilão de contratação de funcionários ou no leilão de orçamento de projetos, o critério de desempate principal

se dará pelos pontos de reputação da empresa dos jogadores, sendo assim, o jogador que possui a menor reputação terá vantagem na escolha. Porém, pode ocorrer de os jogadores possuírem a mesma pontuação de reputação, então, neste caso, a vantagem vai para o jogador que está imediatamente à direita do jogador que obteve a prioridade no último leilão disputado, seja por funcionário ou por projetos. Contudo, ainda existe a possibilidade de empate na primeira rodada quando ainda não houveram lances e nem disputas, conseqüentemente, não existe ainda um jogador que já teve alguma prioridade, então, os próprios jogadores devem definir qual será o critério de desempate apenas para esta primeira rodada.

Ao final da rodada, quando deve-se entregar atividades concluídas de projetos e realizar o pagamento do salário dos funcionários da empresa, caso o jogador não possua capital suficiente em caixa para realizar os pagamentos, este deverá pegar um empréstimo com o banco. Contudo, deve-se levar em consideração que, a partir do momento que o empréstimo é arrecadado, haverá um pagamento por cada empréstimo ao avançar o mês à partir da próxima rodada. Porém, ao quitar um empréstimo, fazendo o pagamento do seu valor total, não haverá mais descontos mensais referente ao empréstimo quitado. Diante disso, o jogador deverá se atentar quanto aos seus rendimentos e o dinheiro que possui em caixa.

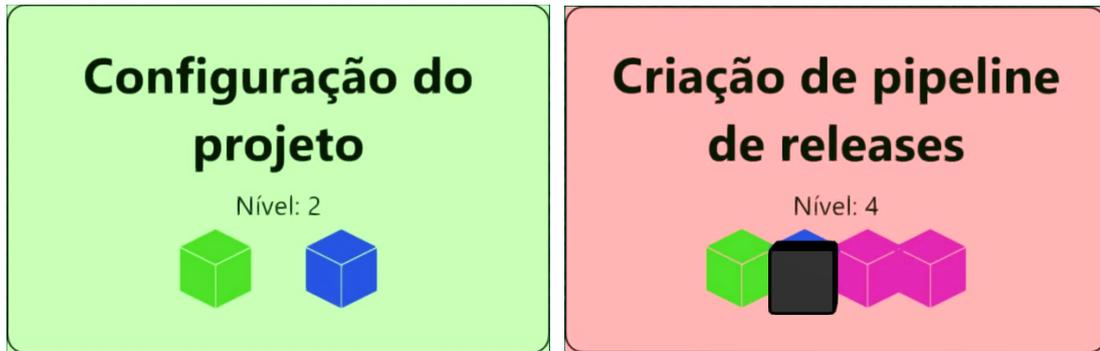
## 4.2 Implementação como jogo de tabuleiro

Para que todos os conceitos e dinâmicas pudessem ser experienciados em um ambiente de sala de aula, o jogo foi criado em forma de jogo de mesa. Essa mídia foi escolhida por experiências ao longo de 4 anos na disciplina de Empreendimentos em Informática, da Universidade Federal de Juiz de Fora (UFJF), onde tem sido observado grande participação e envolvimento dos alunos utilizando diversos jogos de tabuleiro diferentes, como *Power Grid* (FRIESE, 2004) e *StartUps: Assemble!* (MELEGATI et al., 2019). Desta forma, foi definido algumas restrições de projeto: o jogo seria limitado nos componentes a cartas diversas; com um conjunto de moedas (normalmente em papelão); um único tipo de marcadores (comumente cubos de madeira ou plástico); um manual com as regras; e uma caixa para transporte. Para as cartas, foram utilizados cinco tipos de cartas para

representar cada elemento no jogo.

O primeiro deles, é o tipo para as *cartas de atividade*, que são cartas que descrevem uma atividade contida em um projeto de desenvolvimento *software*. Cada carta possui: um título, que é o nome da atividade; seu nível de dificuldade; e um conjunto de tarefas, cada uma com cada tipo de habilidade necessária para que seja concluída.

Existem três níveis de atividades (2, 3 e 4), onde o valor do nível representa a quantidade de tarefas necessária para completá-la. Ainda, a atividade pode conter diferentes tipos de tarefas, não se limitando a um mesmo tipo, tornando-a mais complexa para ser concluída. A Figura 4.2 apresenta duas cartas de atividades, uma de nível 2 (Figura 4.2a) e outra de nível 4 (Figura 4.2b). A carta de nível 4 possui um cubo sobre uma tarefa para representar que a atividade está parcialmente completa.

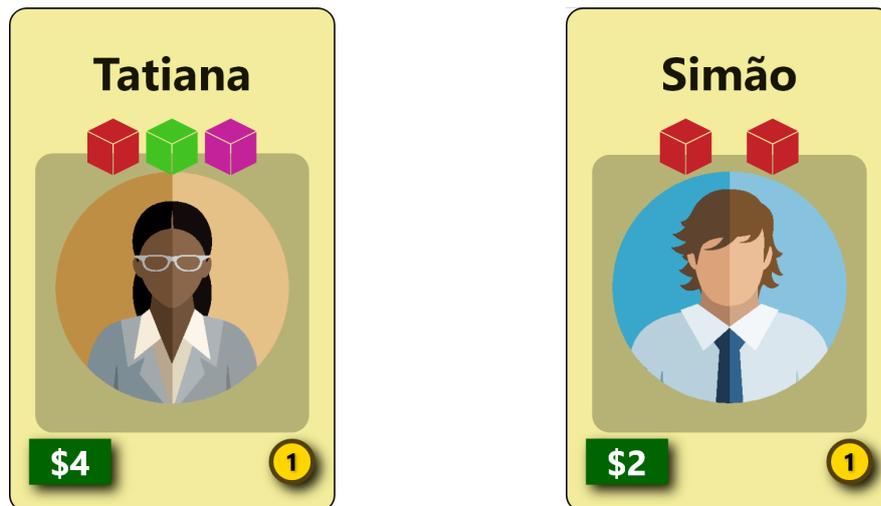


(a) Atividade de nível 2, o que significa que possui duas tarefas, representadas pelos cubos coloridos contidos na carta. O cubo de cor verde representa uma tarefa de *Testes*, enquanto o cubo de cor azul representa uma tarefa de *Banco de Dados*.  
(b) Atividade de nível 4, sendo o cubo de cor verde tarefa de *Testes*, o cubo de cor rosa representando tarefas de *DevOps*. A segunda tarefa está com um cubo preto, o que representa que já foi concluída.

Figura 4.2: Cartas de atividade: uma atividade de nível 2 e uma atividade de nível 4.

As cartas de funcionários representam uma pessoa, com suas respectivas habilidades para cumprir tarefas. Cada carta de funcionário possui um nome fictício para representar este funcionário, uma quantidade de cubos que representa seu nível de habilidades, uma imagem ilustrativa para representar uma pessoa, um valor salarial e sua reputação (pontos). A quantidade de cubos pode variar entre 1 e 3, e possuem cores para diferenciar entre as habilidades, além disso, um funcionário pode possuir mais de um cubo para a mesma habilidade ou até mesmo habilidades variadas. O salário pretendido é exibido em um quadro verde, no canto inferior esquerdo. Este também é o valor mínimo

para uma oferta de salário e varia de acordo com a quantidade e tipo de habilidades do funcionário. Sua reputação é exibida em um círculo amarelo no canto inferior direito na carta. A Figura 4.3 apresenta duas cartas de funcionários, um com três habilidades e outro com duas. A Figura 4.3a mostra uma carta de funcionário com três níveis de habilidades, onde suas habilidades são variadas, e com valor salarial igual a \$4. Um funcionário com dois cubos pode, por rodada, cumprir duas tarefas de mesma cor nas atividades liberadas de projetos, a Figura 4.3b apresenta um funcionário com esta característica.

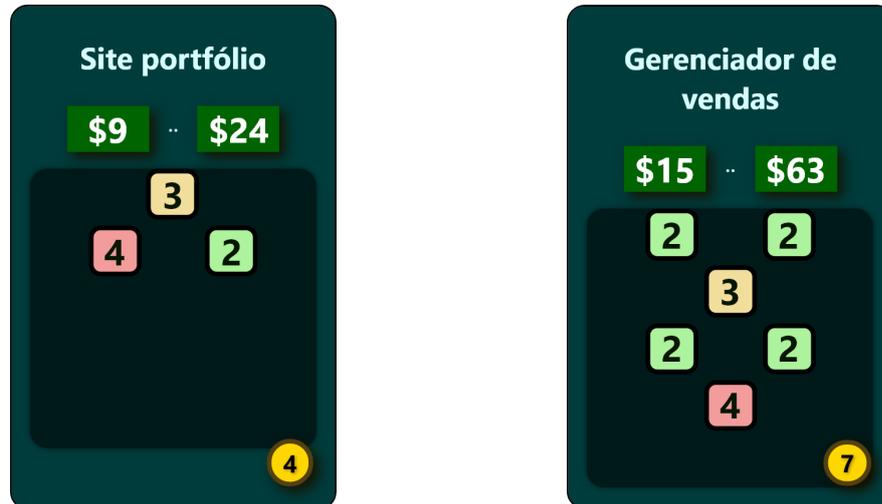


(a) Funcionário com nome “Tatiana”, possuindo três níveis de habilidades, onde uma habilidade é de *Desenvolvimento*, uma habilidade é de *Testes* e uma habilidade é de *DevOps*, valor salarial igual a \$4 e sua reputação igual a 1. (b) Funcionário com nome fictício de “Simão”. Este funcionário possui dois níveis de habilidades, sendo ambas as habilidades são referente à *Desenvolvimento*. Seu valor salarial corresponde a \$2 e sua pontuação vale 1.

Figura 4.3: Cartas de funcionário: cada funcionário possui um nome, seu nível de habilidades e a representação de quais são estas habilidades, uma imagem ilustrativa de uma pessoa, seu valor salarial, e sua reputação.

As cartas de projetos representam contratos com clientes e são a principal fonte de reputação para a empresa. Apresentam o nome do projeto, o intervalo de valores que o cliente está disposto a pagar, sua sequência e complexidade de atividades e sua quantidade de reputação (pontos) ao final do jogo, se completos. Como pode ser visto na Figura 4.4, o intervalo de valores varia para cada projeto, tomando como base a quantidade e complexidade de atividades que este possui. As atividades ficam dispostas em um quadro e cada projeto possui seu quadro de atividades únicos e com diferentes níveis de atividades. A Figura 4.4a mostra um projeto com faixa de valores entre \$9 e \$14, dois níveis de atividades, onde o primeiro nível requer uma atividade de nível 3 e o

segundo nível requer duas atividades, uma de nível 4 e outra de nível 2, e sua reputação vale 4 pontos. A Figura 4.4b apresenta um projeto com valores entre \$15 e \$63, quatro níveis de atividades, onde o primeiro nível requer duas atividades de nível 2, o segundo nível requer uma atividade de nível 3, o terceiro nível demanda duas atividades de nível 2, e o último nível exige uma atividade de nível 4, e sua reputação vale 7 pontos.

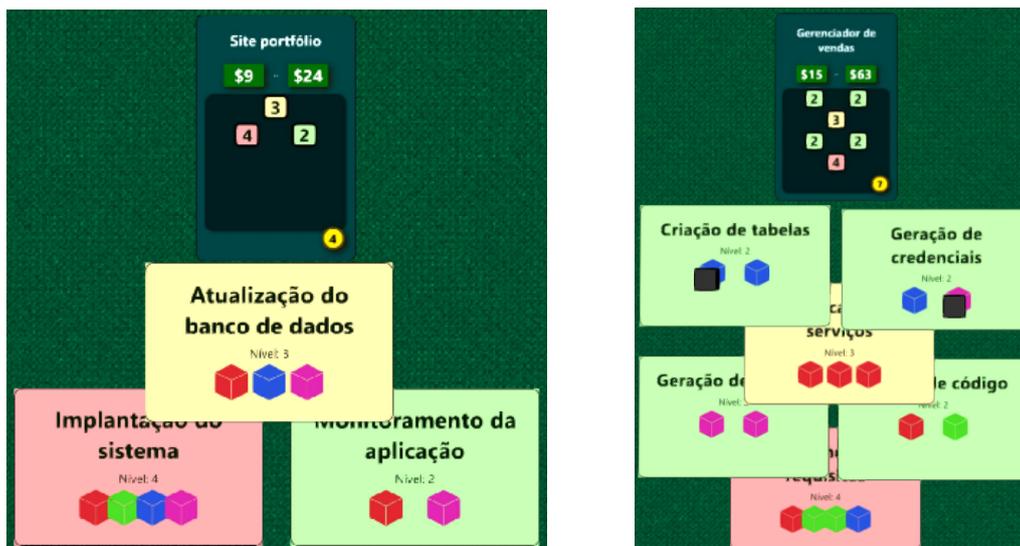


(a) Projeto com título “*Site portfólio*” com faixa de valores entre \$9 e \$14. Sua demanda é de três atividades: uma atividade de nível 3, uma atividade de nível 4 e uma atividade de nível 2. Sua pontuação é 4. (b) Um projeto com título “*Gerenciador de vendas*”, com faixa de valores entre \$15 e \$63, quatro níveis de atividades com um total de seis atividades, e com 7 de pontuação.

Figura 4.4: Cartas de projeto: Os projetos contém título, uma faixa de valores que difere para cada projeto, a árvore de estrutura de atividades, e suas pontuações.

O jogador, ao adquirir o projeto, deve montar a estrutura utilizando as cartas de atividade, seguindo exatamente como é disposto no quadro de atividades do projeto, posicionando as atividades abaixo da carta do projeto. A Figura 4.5 apresenta duas cartas de projetos com suas respectivas árvores de atividades montadas. Na Figura 4.5a contém um projeto contendo 2 níveis de atividades e com sua estrutura de atividades montada. Neste projeto nenhuma atividade foi iniciada. A Figura 4.5b apresenta um projeto com 4 níveis de atividades. Neste projeto já existem atividades parcialmente concluídas, que são as do primeiro nível, representadas pelos cubos pretos em cima da tarefa da atividade. Para que as atividades dos níveis inferiores sejam iniciadas, é necessário que todas as tarefas das atividades superiores sejam concluídas.

A carta de empréstimo é a que apresenta a informação referente a quantos empréstimos o jogador tomou. Dessa forma, é mostrado um quadro verde que representa



(a) Projeto contendo 3 atividades em sua árvore de representação, sendo que ainda não há tarefas concluídas em nenhuma atividade. (b) Projeto contendo 6 atividades em sua árvore de representação, onde uma tarefa está concluída para cada atividade do primeiro nível.

Figura 4.5: Cartas de projeto: Os projetos estão dispostos com suas respectivas atividades de acordo com a demanda de nível que cada projeto exige.

a quantia de valor monetário que o jogador recebe ao pegar um empréstimo, onde este valor é a quantia de \$10, e um quadro vermelho onde este valor é de \$1, que representa a quantia que o jogador deve pagar a cada mês enquanto o empréstimo não é quitado. O número de empréstimos tomados é representado por um cubo preto sobre a carta.

### 4.3 Desenvolvimento do protótipo digital

Para o protótipo digital, utilizamos o Tabletopia, que é uma plataforma digital para jogos digitais de tabuleiro. Esta plataforma possui vários tipos de componentes próprios, que podem ser adicionados ao jogo durante sua criação. Além disso, é possível criar novos componentes através de imagens que representam estes componentes, ou seja, para criar e adicionar uma carta de um funcionário foi necessário criar uma imagem representando a carta e adicioná-la ao novo componente, tornando assim um novo componente de carta para o *Bootstrapping*.

Normalmente, os criadores de jogos de tabuleiro digitais utilizam editores de imagens para gerar as imagens de seus componentes para, assim, adicionar à plataforma, porém, as imagens dos componentes do *Bootstrapping* foram geradas através de imple-

mentação de código utilizando *HyperText Markup Language* (HTML), *Cascading Style Sheet* (CSS) e *JavaScript* (JS) via *React*. Seguimos esta abordagem devido à algumas vantagens: versionamento de código, sendo possível retomar versões anteriores de forma rápida; geração automática das imagens após novas modificações; facilidade em atualizar e gerar as imagens para, assim, se tornarem os componentes do jogo.

Como os componentes foram criados utilizando *React*, que é uma biblioteca JS (REACT, 2021), também fizemos o uso do *Node.js* (NODE JS, 2021) para criar um servidor local para que a aplicação pudesse ser executada e os componentes JS pudessem ser compilados e gerados na página. Além disso, utilizamos o *Puppeteer* para gerar as imagens de cada elemento. O *Puppeteer* é uma biblioteca do *Node.js* que fornece uma API de alto nível para controle programático do *Chrome* ou *Chromium* por meio do protocolo *DevTools* (PUPPETTER, 2021). Esta ferramenta é comumente utilizada para criação e execução de testes de *User Interface* (UI) em navegadores, porém, a utilizamos devido sua facilidade em gerar uma imagem dos elemento individualmente através de *prints* da tela do navegador. Além disso, todos os elementos são gerados em uma única página, sendo assim, o código implementado pode ser utilizado para gerar uma versão impressa dos componentes.

A Figura 4.6 apresenta um trecho do código implementado para criar o elemento da carta de atividades na página do navegador. Neste trecho é criado a função “*AtividadeCard*” que recebe um objeto de carta por parâmetro e seu retorno é um página em HTML que utiliza os atributos do objeto para identificar cada elemento da carta. Já a Figura 4.7 representa um trecho do código para o elemento da carta de funcionários. Nesta parte do código está o trecho do retorno da função criada especificamente para esta carta. Sendo assim, é possível observar o trecho de código em HTML que representa a parte da carta onde são exibidas as habilidades do funcionário. Como pode ser visto, o código faz uma varredura em cada habilidade que consta na carta e inclui estas habilidades no elemento para que sejam mostrados na página.

Para que os elementos sejam montados na tela do navegador é utilizado o comando “*npm start*” em um *prompt* de comandos, como o *Windows PowerShell* ou o terminal do *Visual Studio Code*. Após a execução deste comando, uma página web é aberta no

```

export default function AtividadeCard({ card }) {
  return (
    <div className={`atividade a${card.nivel}`}>
      <div className="card">
        <div className="name">{card.name}</div>
        <div className="nivel">
          <span>Nível: </span>
          {card.nivel}
        </div>
      </div>
    </div>
  )
}

```

Figura 4.6: Trecho do código em *React* que faz a montagem do elemento da carta de atividade do *Bootstrapping*.

```

<div className="habilidadesLabel">Habilidades:
  <div className="habilidadesContainer">
    {card.habilidades.map((habilidade) => {
      const habs = [];
      for (let i = 1; i <= habilidade.nivel; i++) {
        habs.push(<div className={`nivelHabilidade ${habilidade.habilidade}`}
          key={`_${habilidade.id}_${i}`}>{habilidade.habilidade}</div>);
      }
      return habs;
    })}
  </div>
</div>

```

Figura 4.7: Trecho do código em *React* que faz a montagem do elemento da carta de funcionário do *Bootstrapping*.

navegador direcionada para a *Uniform Resource Locators* (URL) “<http://localhost:3000/>”, esta URL é previamente configurada no código do projeto. Nesta página são exibidos todos os elementos que representam as imagens das cartas, uma parte destas imagens é mostrada na Figura 4.8.

Na Figura 4.9 é possível ver um trecho de código que utiliza a API do *Puppeteer* onde, para cada elemento encontrado na tela que seja identificado como uma carta, é feito as definições de nome para o arquivo e a geração das imagens no repositório previamente configurado.

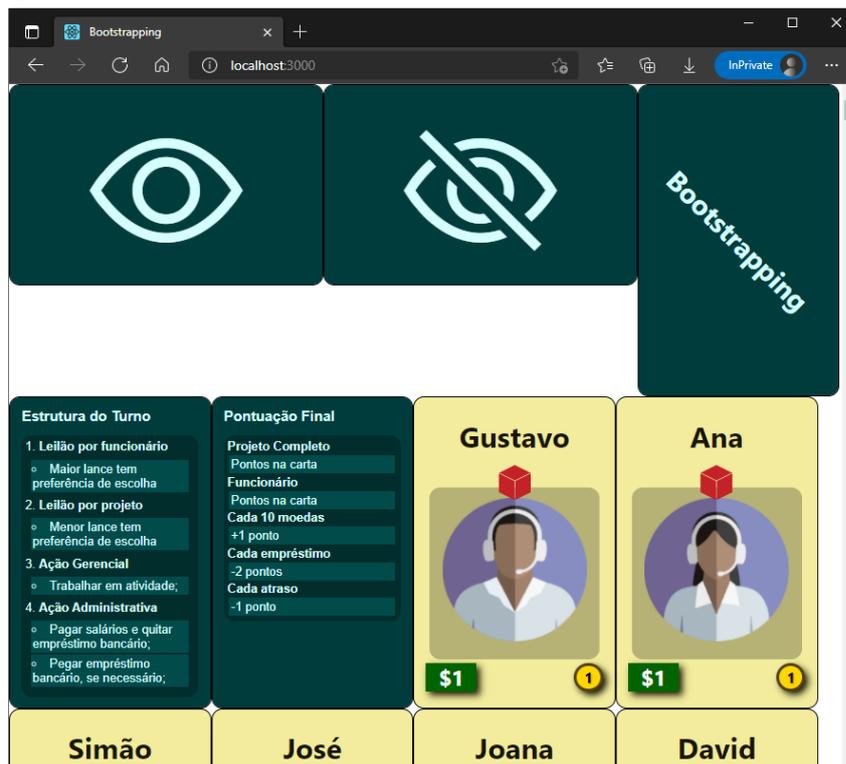


Figura 4.8: Parte da página no navegador que contém todos os elementos do *Bootstraping*. No quadro do navegador é possível ver os seguintes elementos: frente e verso da carta de área escondida; imagem do verso das cartas com o nome do *Bootstraping*; frente e verso da carta de guia; e cartas de funcionários.

```
const allCards = await page.$$(`.${tp}`);
for (let [c, card] of allCards.entries()) {
  const v = String(c + 1).padStart(2, 0);
  const fn = `${dir}/card-${tp}-${v}.png`;
  try {
    console.log(`Generated CARD ${v} ${fn}`);
    await (card).screenshot({
      path: fn,
      omitBackground: true,
    });
  }
}
```

Figura 4.9: Trecho do código utilizando a API do *Puppeteer* para fazer a geração das imagens de cada carta do *Bootstraping*.



Figura 4.10: Legenda do gráfico referente às respostas das perguntas do formulário.

## 4.4 Avaliação do *Bootstrapping* com o MEEGA+

O *Bootstrapping* foi aplicado em sua versão digital, via *Tabletopia*, na disciplina de Empreendedorismo em Informática do Departamento de Ciência da Computação (DCC) da UFJF, de forma remota em função das restrições de bio-segurança devido ao estado da pandemia em Julho de 2021. Os alunos foram agendados previamente e entre os dias 19 e 29 de Julho de 2021, 23 alunos compareceram, em 8 mesas diferentes.

Para coletar os dados e realizar a avaliação do MEEGA+, um formulário online foi enviado para os alunos após jogarem o *Bootstrapping*. Assim sendo, cada aluno respondeu o formulário do MEEGA+ de forma individual e informando suas percepções quanto ao jogo. O formulário apresenta a escala *Likert* de 5 pontos, com as resposta variando entre “Concordo totalmente” e “Discordo totalmente”. Deste modo, para cada item, o aluno atribuía uma resposta em relação à pergunta apresentada entre: concorda totalmente; concorda; é indiferente; discorda; ou discorda totalmente.

A Figura 4.10 representa a legenda do gráfico gerado de acordo com as respostas e auxiliará na visualização das imagens contidas neste documento que apresentam os gráficos com as respostas do formulário.

O questionário possui dois grupos de perguntas: Usabilidade e Experiência do jogador. Além disso, cada grupo possui seções para avaliar o jogo de forma mais direta e organizada. O grupo de Usabilidade possui as seções: Estética; Aprendizabilidade; Operabilidade; e Acessibilidade. Já o grupo de Experiência do jogador conta com as seções: Confiança; Desafio; Satisfação; Interação Social; Diversão; Atenção Focada; Relevância; e Percepção de Aprendizagem.

Presente no grupo de usabilidade, a Figura 4.11 apresenta o gráfico da seção estética. Entre as respostas dos alunos sobre se o *design* dos elementos de jogo são atraentes, tivemos a totalidade bem positiva, mesmo o jogo estando em um protótipo inicial. Mas como a maioria ficou em “concordo”, acreditamos que há bastante espaço para evoluir, mesmo que mantendo os elementos minimalistas da interface atual. Sobre os

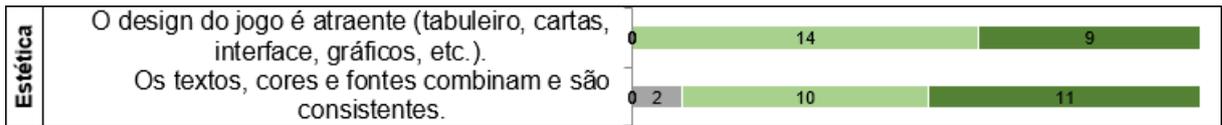


Figura 4.11: Resultado MEEGA+ Seção: Estética.



Figura 4.12: Resultado MEEGA+ Seção: Aprendizibilidade.

textos, cores e fontes estão consistentes, a avaliação se manteve positiva, mas ainda com espaço para melhorias. Com o esquema de cor e fontes sendo mantido por código, essa etapa seria de fácil adaptação pois bastaria alterar o estilo base, mantendo a coerência no projeto todo.

A seção aprendizibilidade procura saber se o jogador teve certa dificuldade em aprender a jogar e se outras pessoas poderiam aprender o jogo de forma rápida. A Figura 4.12 apresenta o gráfico desta seção e nela podemos observar que em relação a aprender poucas coisas para começar a jogar, a maioria das respostas são positivas, portanto percebemos que o jogo está relativamente simples mas ainda podemos torná-lo um pouco mais compreensível. Da mesma forma, a maior parte das resposta quanto a aprender a jogar foi fácil para os jogadores está em “concordo”, o que reforça a simplicidade do jogo em aprender o básico para se iniciar uma partida. Vemos também que as respostas foram positivas quanto ao que os jogadores acham sobre a facilidade em outras pessoas aprenderem a jogar o *Bootstrapping* de forma rápida, sendo assim, vemos que existe espaço para tornar o jogo ainda mais fácil de entender. Contudo, acreditamos que o fato do jogo ter sido acessado pelo *Tabletopia* pode ter trazido alguma dificuldade de entendimento inicial do jogo exatamente por ser uma plataforma digital, tornando a explicação um pouco menos dinâmica.

A seção operabilidade pergunta se o jogo é considerado fácil de jogar e possui regras claras e compreensivas. O gráfico da seção pode ser visto na Figura 4.13 e vemos que tivemos um resultado positivo quanto o jogo ser considerado fácil de jogar, isso mostra que conseguimos reduzir a complexidade do jogo. Sobre a clareza e compreensão das regras

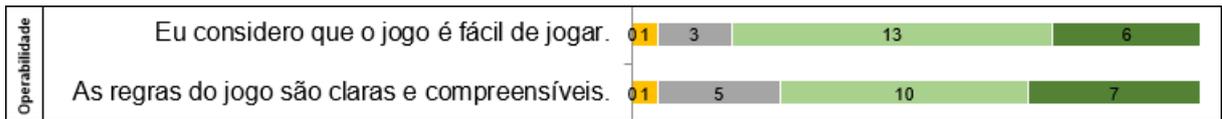


Figura 4.13: Resultado MEEGA+ Seção: Operabilidade.

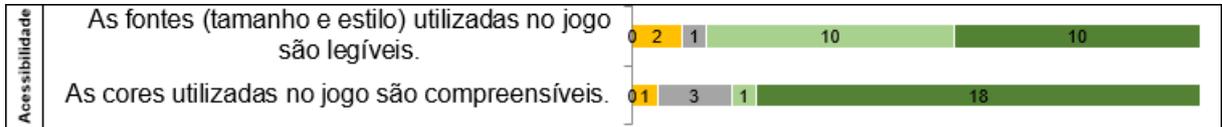


Figura 4.14: Resultado MEEGA+ Seção: Acessibilidade.

do jogo, a maioria das respostas também se mantiveram positivas. Estas respostas nos mostram que, apesar do *Bootstrapping* se mostrar um jogo fácil de entender e de iniciar a jogar, podemos melhorar nossa explicação quanto às regras do jogo.

A Figura 4.14 ilustra o gráfico da seção acessibilidade, que pergunta sobre o tamanho e estilo da fonte e vemos que as repostas também foram positivas, porém, existe um espaço de melhoria quanto a isso, sendo assim, poderíamos tentar aumentar o tamanho da fonte para tornas os elementos um pouco mais legível. Além disso, também existe a pergunta sobre as cores utilizadas serem compreensíveis, e a maioria das respostas são extremamente positivas, nos mostrando que uma adequação de cores não é um ponto preocupante, mas como os elementos foram criados via código, sua alteração se torna fácil até que se encontre as cores ideais.

Quanto ao grupo de experiência do jogador, a seção Confiança pergunta ao jogador se o jogo passa uma primeira impressão de ser fácil de se jogar e se a organização do conteúdo o ajudou neste aspecto. Podemos ver na Figura 4.15 que as respostas ficaram divididas, sendo algumas negativas, quando à facilidade de jogar o *Bootstrapping* à primeira vista. Portanto, percebemos que existe um grande espaço para melhorarmos este quesito, porém, acreditamos que por ter sido um jogo digital, então muitos elementos ficam dispostos na mesa desde a sua criação para facilitar o início do jogo, isso pode ter reduzido a experiência do jogador quanto à primeira impressão do jogo. Quanto a organização do conteúdo ter ajudado na confiança de aprendizagem do jogo, as respostas foram positivas, o que mostra que, apesar da primeira impressão não ter sido tão positiva, a forma que os elementos foram organizados na mesa contribuiu no aprendizado do jogo.

Já a seção desafio procura saber o quão desafiador é o jogo, além de oferecer

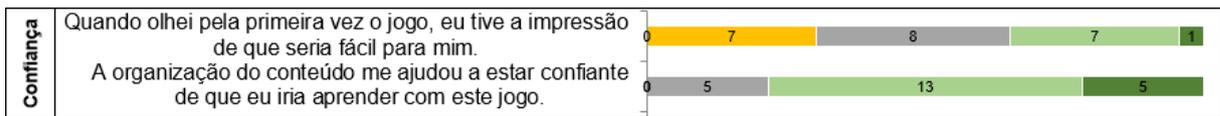


Figura 4.15: Resultado MEEGA+ Seção: Confiança.



Figura 4.16: Resultado MEEGA+ Seção: Desafio.

novos desafios, e se não se torna monótono durante o decorrer da partida. As respostas estão na Figura 4.16 e mostram que tivemos respostas positivas sobre o jogo ser adequadamente desafiador, sendo assim, percebemos que conseguimos tornar o jogo desafiador para o jogador. As respostas também foram positivas quanto ao *Bootstrapping* oferecer novos desafios em ritmo adequado, acreditamos que isso se dá pela dinâmica de leilão e a gerência de capital da empresa, que vai mudando a cada rodada e a cada ação. Porém, ainda sendo a maioria extremamente positiva, tivemos respostas divididas em relação ao jogo se tornar monótono nas tarefas. Como o jogo é dividido em quatro fases e ao final da quarta fase a rodada se inicia novamente, isso pode ter tornado de certa forma monótono para certos participantes, apesar de que a cada rodada o dinheiro de cada empresa muda e os funcionários e projetos que ficam disponíveis no leilão também não se repetem. Com estas respostas podemos perceber que o jogo está mais desafiador do que monótono, porém, adicionar novos desafios envolvendo os projetos e os funcionários pode ser agregar para estes quesitos.

A Figura 4.17 representa o gráfico da seção satisfação, onde o jogador responde sobre seu sentimento de realização ao completar tarefas, também se o avanço no jogo é devido ao seu esforço pessoal, além da satisfação sobre o que foi aprendido no jogo e se recomendaria o jogo para outros colegas. Sobre o sentimento de realização ao completar tarefas, as respostas foram muito positivas, mostrando que o *Bootstrapping* torna o jogador mais realizado ao finalizar suas ações com sucesso. As repostas também foram positivas quanto a sensação de que só é possível avançar no jogo devido ao esforço pessoal, o que destaca que o jogo exige empenho do jogador ao passo que traz um sentimo bom a cada avanço obtido. Quanto o sentimento de satisfação sobre o que foi aprendido

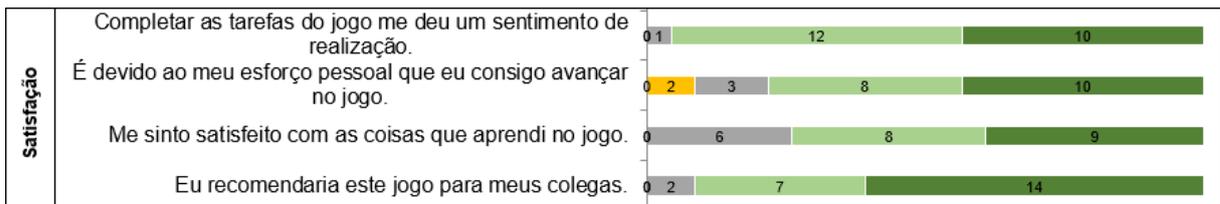


Figura 4.17: Resultado MEEGA+ Seção: Satisfação.

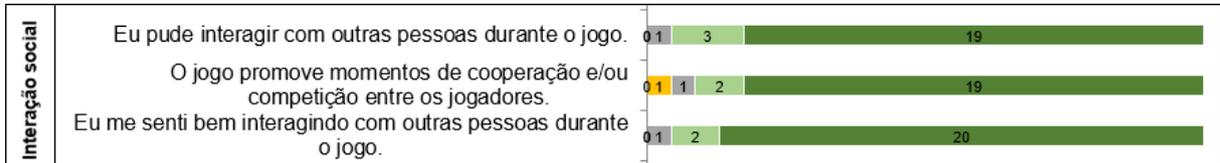


Figura 4.18: Resultado MEEGA+ Seção: Interação social.

no jogo, tivemos respostas positivas para a maioria do participantes, então percebemos que este quesito pode ser melhorado no futuro adicionando mais informações pertinentes relacionadas ao conteúdo da disciplina para que o jogador possa descobrir coisas novas ao jogar. A grande maioria dos participantes respondeu de forma positiva em relação a recomendar o jogo para seus colegas, mostrando que o *Bootstrapping* possui muito potencial e que outras pessoas também poderão se divertir ao jogar.

A interação social é a seção que pergunta se é possível interagir com outros jogadores, se o jogador se sentiu bem interagido com outras pessoas durante o jogo e se o jogo promove momentos de cooperação e/ou competição. A Figura 4.18 representa o gráfico com os resultados. Nesta seção tivemos a grande maioria das respostas como “Concordo totalmente” para todas as perguntas, sendo assim, a maioria dos participantes concordam totalmente que o *Bootstrapping* permite: a interação com outras pessoas; promove momentos de cooperação e/ou competição entre os jogadores; e a sensação de interação com outras pessoas durante o jogo. Estas respostas comprovam que o *Bootstrapping* não é um jogo sem interação social onde os jogadores se sentirão isolados durante as partidas, portanto, pode ser um forte aliado ao ensino.

Na seção diversão temos perguntas sobre a diversão do jogador durante o jogo e se ocorre situações durante o jogo que o fez sorrir. Temos as respostas representadas na Figura 4.19 e todas foram positivas, tendo como maioria em “Concordo totalmente”, quando questionados se se divertiram com o jogo e se aconteceu alguma situação durante o jogo que o fizeram sorrir. Sendo assim, percebe-se que o *Bootstrapping* promove diversão

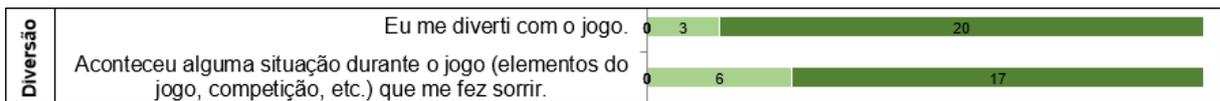


Figura 4.19: Resultado MEEGA+ Seção: Diversão.

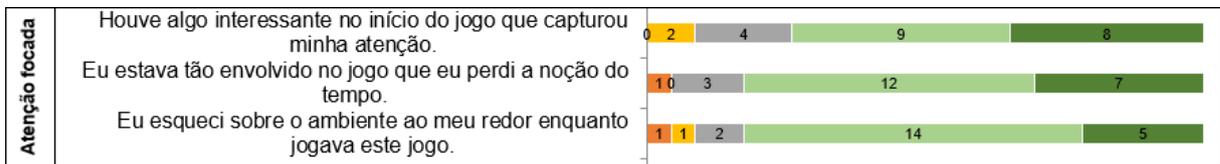


Figura 4.20: Resultado MEEGA+ Seção: Atenção focada.

e alegria para os jogadores, aliados a um grande interação social, citado na seção anterior.

Na Figura 4.20 temos os resultados obtidos na seção atenção focada, que avalia se houve algo interessante no jogo que captasse a atenção do jogador, também avalia se o jogador tem o sentimento de perder a noção do tempo e distração quanto ao ambiente ao redor devido seu envolvimento durante o jogo. Sobre haver algo interessante no início do jogo que captou a atenção do jogador, tivemos respostas positivas para a maior parte dos participantes, mostrando que desde seu início o *Bootstrapping* é um jogo com potencial para prender a atenção dos alunos. Também tivemos respostas positivas na pergunta referente ao jogador estar tão envolvido ao jogo que o fez perder a noção do tempo, assim como na pergunta sobre o participante se esquecer do ambiente ao redor enquanto jogava o jogo. Com estas respostas podemos perceber que o *Bootstrapping* mantém o foco do jogador exclusivamente para o jogo, porém, como a maioria das respostas positivas foram para “Concordo”, também existe uma certa necessidade de adicionar elementos e/ou dinâmicas ao jogo para o tomar ainda mais a atenção do jogador.

A seção relevância contém perguntas sobre a relevância do conteúdo do jogo quanto aos interesses do aluno, além de perguntas sobre o conteúdo do jogo estar relacionado com a disciplina, podendo ser um método de ensino adequado ao ensinamento da disciplina. Os resultados desta seção estão representados na Figura 4.21 e vemos que na pergunta sobre o conteúdo do jogo ser relevante para os interesses do aluno as respostas foram, em sua maioria, positivas, transparecendo que o *Bootstrapping* de fato tem relação com a disciplina de Empreendedorismo em Informática. Obtivemos a maioria das respostas positivas quanto a clareza do conteúdo do jogo estar relacionado com a disciplina, provando que os elementos do jogo e sua dinâmica não fogem do conteúdo da disciplina,

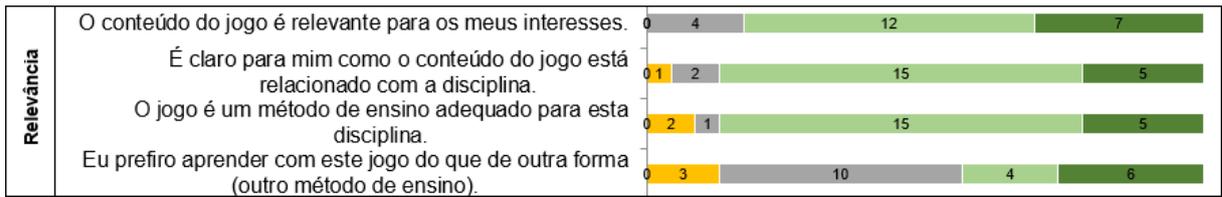


Figura 4.21: Resultado MEEGA+ Seção: Relevância.

mas como a maior parte das respostas são “Concordo”, percebemos é possível adicionar elementos ao jogo para agregar no seu conteúdo. Para a pergunta sobre o jogo ser um método de ensino adequado para a disciplina, as respostas também foram positivas com a maioria sendo “Concordo”, sendo assim, vemos que o *Bootstrapping* tem grande potencial para auxiliar no ensino de disciplinas que possuem relação com o assunto, porém, incluir algumas dinâmicas que evidenciem o ensinamento do jogo poderá melhorar ainda mais no aprendizado. Já na pergunta sobre o aluno preferir aprender com este jogo do que de outra forma, tivemos respostas variadas com a maioria sendo “Indiferente”. Acreditamos que para a última pergunta, o fato de só ser possível a aplicação do jogo de forma remota devido à pandemia pode interferir no interesse do aluno quanto à preferência de aprendizagem, e em um cenário onde as aulas são presenciais, a aplicação do jogo para o auxílio do ensino poderá ser uma ótima opção para o aprendizado.

Finalmente, a seção percepção de aprendizagem procura avaliar se o jogo contribuiu com a aprendizagem na disciplina e se esta aprendizagem foi eficiente em relação a outras atividades da disciplina. A Figura 4.22 ilustra estes resultados e podemos ver que na pergunta sobre o jogo contribuir para a aprendizagem na disciplina teve a maioria das respostas como positivas, comprovando que, apesar de ainda possuir muito espaço para melhorias, o *Bootstrapping* é uma opção viável como auxiliar no ensino da disciplina. Quanto a pergunta sobre o jogo ser eficiente para a aprendizagem, em comparação com outras atividades da disciplinas, as repostas foram divididas e a maior parte está em “Indiferente”. Percebemos que podemos melhorar ainda mais o jogo para que ele se torne efetivamente mais eficiente, porém, não podemos deixar de notar que o *Bootstrapping* se provou uma excelente opção se ensino para a disciplina.

Além das perguntas acima, onde se é possível selecionar uma opção, o formulário também contém questões para que pudesse ser dito os pontos fortes, o que poderia

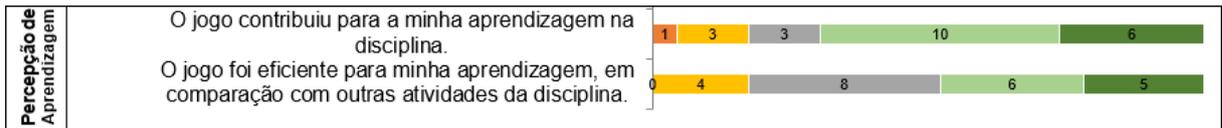


Figura 4.22: Resultado MEEGA+ Seção: Percepção de Aprendizagem.

ser melhorado segundo a visão do jogador e comentários adicionais.

Como pontos fortes do jogo, foram citados dentre as respostas: a sensação de planejamento a longo prazo; a mecânica de leilão, promovendo uma interação bacana entre os participantes; o desafio de dosar o dinheiro gasto e dinheiro ganho; o incentivo à estratégia e competição entre os jogadores; e que o jogo é educativo. Portanto, com estes comentários é possível perceber que o *Bootstrapping* promove uma boa interação entre os jogadores devido às dinâmicas e mecânicas empregadas ao jogo.

Como percepções de melhorias foram citados principalmente a dificuldade enfrentada para saber quais de seus funcionários já haviam sido alocados em alguma das atividades dos projetos. Também foi citado sobre uma possível automação em alguma das fases do jogo e em haver uma melhor mobilidade dos elementos. Porém, como utilizamos uma plataforma online para jogos de tabuleiro somente para que pudéssemos compartilhar o *Bootstrapping* com os alunos, não sendo ele parte do nosso desenvolvimento, há certa limitação em sua utilização que fogem do nosso domínio.

Quanto aos comentários adicionais, foi destacado o divertimento que os jogadores tiveram durante as partidas e a vontade de poder jogar novamente no futuro. Além disso, recebemos elogios quanto ao desenvolvimento do jogo.

Analisando os resultados obtidos percebemos que o *Bootstrapping* obteve muitas respostas positivas, evidenciando que o jogo está no caminho certo para se tornar um ótimo aliado para o ensino. Além disso, de acordo com as respostas obtidas foi possível imaginar algumas possíveis modificações para o *Bootstrapping*, como alterações na interface dos elementos, alterando cores e tamanho de fontes, ampliação nas dinâmicas do jogo adicionando novos desafios, como entregas de atividades com pouca qualidade fazendo com que o jogador seja penalizado, terceirização de funcionários, ou seja, o jogador poderá contratar temporariamente o funcionário de outro jogador, e também a demissão de funcionários após o pagamento de taxas de rescisão de contrato. Em suma, o *Boots-*

*trapping* possui espaços para melhorias que tornarão o jogo ainda mais completo, mas, já é possível notar o potencial do jogo para o auxílio na educação.

## 5 Considerações Finais

Instituições de ensino dispõem de disciplinas de gerenciamento de projetos e empreendedorismo para que os alunos tenham o conhecimento necessário para que saibam como aplicar seus conhecimentos de forma eficiente e que seus projetos sejam bem sucedidos. Entretanto, o ensino de gerenciamento de projetos, por muitas vezes, é um processo difícil ainda mais quando há a necessidade de gestão de recursos além de planejamento. A criação de projetos piloto é inviável durante uma disciplina e colocá-lo em prática pode deixar alguns alunos em situações que não envolvem comprometimento ou tomada de decisão.

Estar bem preparado para gerenciar um projeto é de suma importância para que ele seja concluído com sucesso, sem atrasos e sem gastos extras não previstos. Portanto, ter o conhecimento ou experiência em como atuar em todas suas etapas e adversidades, contribui para ele seja bem sucedido.

Este trabalho teve como objetivo principal auxiliar no ensino de gerenciamento de projetos e empreendedorismo de *software* através do planejamento e desenvolvimento de um novo jogo sério para uso educacional, o *Bootstrapping*. Utilizando mecanismos de jogos modernos, a sua ideia principal é poder simular as etapas e dinâmicas de gerenciamento de uma fábrica de *software* em seus primeiros anos, construindo uma equipe para realizar projetos, em um cenário de recursos limitados e em um ambiente competitivo.

O jogo foi desenvolvido e apresentado em uma plataforma remota ao longo de um ano em oficinas de desenvolvimento de jogos e teve boa aceitação dos especialistas em desenvolvimento. Em uma segunda etapa, o jogo foi aplicado em ambiente educacional e foi avaliado seguindo uma metodologia de avaliação de jogos educacionais. Com o bom desempenho em ambos os cenários, temos indícios que o objetivo principal foi atingido, mesmo que parcialmente pois resta criar e avaliar a sua versão física em uma dinâmica presencial quando os protocolos de bio-segurança para COVID-19 assim o permitirem.

O jogo deveria simular todas as etapas e, conseqüentemente, todas as adversidades de um projeto, como a demissão ou ausência de um funcionário, ocasionando um possível atraso ou gastos extras para o projeto. Entretanto, optou-se por manter o jogo

menos caótico, dando mais espaço para os jogadores estimarem melhor os valores que podem pagar e cobrar. Inserir um conjunto de eventos poderia complicar ainda mais o andamento da partida em função da ferramenta online.

Outro ponto que teve que ser imposta uma limitação foi em não permitir ao jogador, no papel de gerente, escolher ou adaptar a estrutura do projeto a um processo de *software* à sua escolha. Esse tipo de decisão aparece fixa nos projetos, pela topologia das atividades e fica assim até a sua conclusão. Alguns projetos são fortemente sequenciais enquanto outros permitem uma maior adaptação na sua execução. Mudar essa estrutura durante uma partida ainda é um ponto que deve ser avaliado futuramente e a complexidade inserida deve ser avaliada.

Uma outra limitação é manter todas as habilidades abstratas, elas não possuem um mecanismo associado a cada uma delas. Ou seja, por exemplo, *Desenvolvimento* e *Testes* só são diferenciadas pela cor e ordem com que são demandadas nas atividades. Um ponto de adaptação seria associar a cada uma um efeito para ser utilizado durante a partida. Mas novamente, isso geraria mais complexidade e poderia ser interessante para um ambiente fora da sala de aula ou uma versão “avançada” do jogo. Algo bem comum nos jogos modernos de tabuleiro, mas que inseriria uma complexidade extra em um momento que os mecanismos básicos é que estão em teste.

Após a conclusão deste trabalho, observou-se que ele poderia ser continuado em algumas frentes. Assim sendo, a adição de adversidades durante o andamento do jogo poderia tornar ainda mais real a simulação de um gerenciamento, uma vez que seria possível um funcionário se demitir ou algum outro jogador fazer uma proposta para contratar seu funcionário.

Entre as futuras aplicações está a possibilidade do jogo ser desenvolvido como forma de aplicativo para celulares ou web, no qual as regras seriam aplicadas e regidas pelo sistema ao invés dos próprios jogadores. Hoje esse cenário é bem comum: jogos de tabuleiro têm sido convertidos para jogos digitais com sucesso. Nesses jogos os jogadores podem jogar *online*, de seus próprios celulares, sozinhos contra uma inteligência artificial ou contra outros jogadores online.

## Bibliografia

- ABT, C. C. *Serious games*. [S.l.]: Viking Press, 1970.
- ALKHATLAN, A.; KALITA, J. Intelligent tutoring systems: A comprehensive historical survey with recent developments. *arXiv preprint arXiv:1812.09628*, 2018.
- BAKER, A.; NAVARRO, E. O.; HOEK, A. V. D. Problems and programmers: an educational software engineering card game. In: IEEE. *25th International Conference on Software Engineering, 2003. Proceedings*. [S.l.], 2003. p. 614–619.
- BATTISTI, P.; WEINZIERL, G. Empreendedorismo. 2016.
- BECK, K. et al. Manifesto for agile software development. 2001.
- BOARDGAMEGEEK. 2021. Disponível em: <https://boardgamegeek.com/>. Acesso em: 19 Agosto 2021.
- BRITO, A. M.; PEREIRA, P. S.; LINARD, A. Empreendedorismo. *Juazeiro do Norte: Instituto Federal de Educação, Ciência e Tecnologia do Ceará-IFCE*, 2013.
- CARVALHO, M. A. de et al. Desenvolvimento de um jogo de tabuleiro para o ensino de gerenciamento de projetos. 2018.
- CAULFIELD, C. et al. A systematic survey of games used for software engineering education. *Modern Applied Science*, Canadian Centre of Science and Education, v. 5, n. 6, p. 28–43, 2011.
- EHRMANN, J.; LEWIS, C.; LEWIS, P. Homo ludens revisited. *Yale French Studies*, JSTOR, n. 41, p. 31–57, 1968.
- FALKEMBACH, G. A. M.; GELLER, M.; SILVEIRA, S. R. Desenvolvimento de jogos educativos digitais utilizando a ferramenta de autoria multimídia: um estudo de caso com o toolkit instructor. *RENOTE*, v. 4, n. 1, 2006.
- FERRARI, R. *Empreendedorismo para computação: criando negócios de tecnologia*. [S.l.]: Elsevier, 2009.
- FRIESE, F. *Power Grid*. [S.l.]: 2F-Spiele, 2004.
- HAMADA, R.; HIJI, M.; KANEKO, T. Development of software engineering business board game. In: *Developments in Business Simulation and Experiential Learning: Proceedings of the Annual ABSEL conference*. [S.l.: s.n.], 2014. v. 41.
- HUIZINGA, J. *Homo ludens: o jogo como elemento da cultura*. [S.l.]: Editora da Universidade de S. Paulo, Editora Perspectiva, 1971. v. 4.
- HUNICKE, R.; LEBLANC, M.; ZUBEK, R. Mda: A formal approach to game design and game research. In: *Proceedings of the AAAI Workshop on Challenges in Game AI*. [S.l.: s.n.], 2004. v. 4, n. 1, p. 1722.

- KNOP, I. *Infraestrutura para simulação de processos de software baseada em meta modelos de dinâmica de sistemas*. Tese (Doutorado) — Dissertação (Mestrado em Modelagem Computacional)—Universidade Federal de . . . , 2009.
- KUMAR, G.; BHATIA, P. K. Impact of agile methodology on software development process. *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*, v. 2, n. 4, p. 46–50, 2012.
- MAJER, C. A.; DUDUCHI, M. Uso de simuladores de jogos empresariais na educação corporativa e acadêmica. 2015.
- MELEGATI, J. et al. A board game to teach team composition in software startups. In: SPRINGER. *International Conference on Software Business*. [S.l.], 2019. p. 321–335.
- MICHAEL, D. R.; CHEN, S. L. *Serious games: Games that educate, train, and inform*. [S.l.]: Muska & Lipman/Premier-Trade, 2005.
- NAVARRO, E. *SimSE: A Software Engineering Simulation Environment for Software Process Education DISSERTATION*. Tese (Doutorado) — Citeseer, 2006.
- NICHOLSON, S. Making gameplay matter: Designing modern educational tabletop games. *Knowledge Quest*, American Library Association, v. 40, n. 1, p. 60, 2011.
- NODE JS. 2021. Disponível em: <<https://nodejs.org/en/>>. Acesso em: 21 Agosto 2021.
- PETRI, G. et al. Um quiz game para a revisão de conhecimentos em gerenciamento de projetos. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2016. v. 27, n. 1, p. 320.
- PETRI, G. et al. Benefícios dos jogos não-digitais no ensino de computação. In: SBC. *Anais do XXVI Workshop sobre Educação em Computação*. [S.l.], 2018.
- PETRI, G.; WANGENHEIM, C. G. von; BORGATTO, A. F. Meega+: an evolution of a model for the evaluation of educational games. *INCoD/GQS*, v. 3, 2016.
- PETRI, G.; WANGENHEIM, C. G. von; BORGATTO, A. F. Evolução de um modelo de avaliação de jogos para o ensino de computação. In: SBC. *25º Workshop sobre Educação em Computação (WEI 2017)*. [S.l.], 2017. v. 25, n. 1/2017.
- PMBOK, G. Um guia do conhecimento em gerenciamento de projetos—6ª edição. 2017.
- PRESSMAN, R.; MAXIM, B. *Engenharia de Software-8ª Edição*. [S.l.]: McGraw Hill Brasil, 2016.
- PUPPETER. 2021. Disponível em: <<https://pptr.dev/>>. Acesso em: 21 Agosto 2021.
- REACT. 2021. Disponível em: <<https://pt-br.reactjs.org/>>. Acesso em: 21 Agosto 2021.
- ROCHA, P. S. V. d. S. *A importância do lúdico na educação infantil: uma análise a partir da concepção de professores*. [S.l.]: Universidade Federal da Paraíba, 2018.
- ROCHA, R. V. da; BITTENCOURT, I. I.; ISOTANI, S. Análise, projeto, desenvolvimento e avaliação de jogos sérios e afins: uma revisão de desafios e oportunidades. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2015. v. 26, n. 1, p. 692.

- SARFATI, G. Quem (e como) investe em startups brasileiras. *GV EXECUTIVO*, v. 17, n. 3, p. 12–16, 2018.
- SAVI, R.; WANGENHEIM, C. G. von; BORGATTO, A. F. A model for the evaluation of educational games for teaching software engineering. In: IEEE. *2011 25th Brazilian Symposium on Software Engineering*. [S.l.], 2011. p. 194–203.
- SCHWABER, K.; SUTHERLAND, J. *The Scrum Guide. November 2017*. 2017.
- SILVEIRA, T. S.; PASSOS, D. F. O.; MARTINS, I. Empreendedorismo x startup. *REMIPE-Revista de Micro e Pequenas Empresas e Empreendedorismo da Fatec Osasco*, v. 3, n. 2, p. 304–322, 2017.
- SOMMERVILLE, I. *Software engineering*. [S.l.]: Addison-Wesley/Pearson, 2011.
- SOMMERVILLE, I. *Software Engineering*. 10th. ed. [S.l.]: Pearson, 2015. ISBN 0133943038, 9780133943030.
- SOUZA, S. L. R. de. *Projeto e Desenvolvimento de um Jogo Digital para Auxílio no Ensino de Gerenciamento de Processos de Software*. 2018.
- STRIEGNITZ, P. M.; LICHTER, H. A case study on software project management in industry—experiences and conclusions. *Software, Ver*, v. 1, p. 305–313, 1996.
- TAROUCO, L. M. R. et al. Jogos educacionais. *RENOTE: revista novas tecnologias na educação [recurso eletrônico]*. Porto Alegre, RS, 2004.
- TEUBER, K. *Descobridores do Catan*. 1995.
- VALENTE, M. T. *Engenharia de Software Moderna (Livro Digital)*. [s.n.], 2020. ISBN 978-65-00-00077-1. Disponível em: <<https://engsoftmoderna.info>>.
- WANG, F.; HANNAFIN, M. J. Design-based research and technology-enhanced learning environments. *Educational technology research and development*, Springer, v. 53, n. 4, p. 5–23, 2005.
- WOODS, S. *Eurogames: The design, culture and play of modern European board games*. [S.l.]: McFarland, 2012.
- ZAFFARI, G.; BATTAIOLA, A. L. Integração do processo industrial de design de jogos com o modelo mda. *Proceedings do XIII SBGames-Trilha Indústria*, 2014.

## A Evolução do *Bootstrapping*

Ao longo do desenvolvimento do jogo, percebeu-se a necessidade de algumas melhorias. Dessa forma, o jogo foi evoluindo a medida que novas regras, mecanismos e dinâmicas foram adicionadas ou retiradas. Sendo assim, observamos a necessidade de separar cada etapa de criação em versões.

Em cada versão do jogo, será destacado os pontos positivos e negativos observados. Além disso, poderão ser apontadas algumas sugestões de modificações para as versões seguintes.

### A.1 Versão 1

Na primeira versão do jogo, notamos que a dinâmica do jogo estava no caminho certo. Onde, em cada rodada, o jogador possuía dois tipos de ações: uma administrativa e outra gerencial. No movimento administrativo, o jogador poderia optar por realizar um empréstimo, fazer uma oferta para funcionário ou fazer um orçamento para um projeto. Já na ação gerencial, o jogador deveria escolher entre alocar um funcionário em alguma atividade ou virar o mês.

O jogo inicia com três cartas de funcionários, projetos e atividades viradas para cima, dessa forma, os jogadores já podem começar a pensar em suas estratégias desde o início.

Com as cartas de atividades viradas para cima, chegamos a conclusão de que esta característica do jogo poderia simular o processo de reunião com o cliente para o levantamento de requisitos de um projeto, visto que estas reuniões ajudam na decisão contratual de um projeto real, tanto quanto poderá ajudar o jogador a decidir qual será o melhor projeto para orçar ou o melhor funcionário para contratar.

Ao decorrer do jogo, percebemos que poderia ser muito fácil juntar dinheiro com poucas rodadas, pois, ao virar o mês, o jogador apenas fazia o pagamento do empréstimo, ou seja, este era o único gasto do jogador durante todo o jogo. Diante

disso, vimos a necessidade de adicionar mais algum custo para o jogador, então, a solução encontrada, que foi adotada na versão seguinte, foi adicionar o custo de salário por cada funcionário ao final de cada mês.

Foram observados outros dois pontos de melhoria para as versões seguintes. A primeira foi o tamanho de algumas letras das cartas, pois, em alguns momentos havia uma certa dificuldade em ler algumas informações. A segunda foi o jogador começar com um empréstimo, visto que na primeira rodada todos os jogadores começam sem dinheiro, a ação de pegar empréstimo era basicamente uma jogada obrigatória.

## **A.2 Versão 2**

Na segunda versão testamos a primeira melhoria observada, que foi a adição de mais gastos para o jogador.

Esta nova dinâmica se fez muito útil, pois adicionou um novo grau de dificuldade ao jogo fazendo com que os jogadores precisassem agir com mais cautela pois, se contratassem muitos funcionários, poderiam ficar sem dinheiro para cobrir os gastos.

Ao adicionar este novo gasto, observamos que em uma próxima versão poderia ser adicionado a opção de atraso de pagamento de funcionário, o que acarretaria em algumas consequências, como o pagamento dobrado no próximo mês ou até mesmo a saída do funcionário da equipe.

Com a regra inicial de que a soma das habilidades dos funcionários sendo igual a três, seria possível realizar uma tarefa referente a qualquer habilidade, pensamos que seria interessante adicionar um ponto negativo referente qualidade ruim de entrega da atividade em questão, visto que o funcionário, ou funcionários, que concluiu a tarefa não possui a habilidade correta exigida para a execução da mesma. Porém, até esta versão, não foi pensado como este tipo de penalidade seria aplicada.

## **A.3 Versão 3**

Nesta versão, fizemos um *playtest* com 4 jogadores em uma oficina de jogos online onde vários desenvolvedores jogos de tabuleiro online apresentam seus jogos para que outros

jogadores possam jogar. Neste primeiro *playtest* participaram do nosso jogo duas pessoas que são da área de *startups* e, ao final da partida, todos deram suas considerações sobre o jogo.

Os jogadores observaram que a alocação e as regras do jogo estão simples e claras. Além disso, também destacaram que a montagem dos projetos está bem interessante, visto que a montagem segue a estrutura da imagem da carta de projetos, onde cada bloco representa uma atividade e sua complexidade.

Em contrapartida, os jogadores perceberam que a pontuação está desbalanceada e os ganhos por um projeto não estão muito adequados, sendo assim, poderemos fazer alguns ajustes nestes valores para que o jogo esteja mais balanceado.

Ainda, percebeu-se que o jogo está muito punitivo para um jogador iniciante que ainda não domina as regras e que, também, se torna muito difícil quando o jogador se encontra em uma situação ruim, como muitos empréstimos ou muitas dívidas, tornando o jogo, devido a estas condições, frustrante.

Além dos pontos positivos e negativos que foram observados, os jogadores também fizeram algumas sugestões de melhorias para as próximas versões. Uma das melhorias propostas foi a utilização de um montante de dinheiro inicial que não esteja relacionado com um empréstimo, mas sim como um 'investimento' inicial para a empresa.

## A.4 Versão 4

Nesta versão alteramos o cálculo do valor dos projetos. Devido aos *feedbacks* recebidos na versão passada, vimos a necessidade de ajustar os valores dos projetos, aumentando seu retorno financeiro para o jogador.

Nas versões anteriores haviam dois cálculos, um para o valor mínimo e outro para o valor máximo. Com isso, os valores eram calculados da seguinte forma:

- Valor mínimo do projeto = Soma do nível de cada atividade - Porcentagem de variação do projeto
  
- Valor máximo do projeto = Soma do nível de cada atividade + Porcentagem de variação do projeto

Nesta versão, os cálculos estão sendo feitos da seguinte forma:

- Valor mínimo do projeto = Soma do nível de cada atividade
- Valor máximo do projeto = Soma do nível de cada atividade + (Porcentagem de variação do projeto \* Quantidade de atividades no projeto)

Outra mudança aplicada nesta versão é a dinâmica de turnos. Anteriormente, os jogadores faziam suas ações de forma livre em relação à cronologia do jogo, ou seja, durante a ação administrativa, cada jogador poderia escolher entre fazer uma oferta a um funcionário, fazer um lance em um projeto ou pegar. Já na ação gerencial, o jogador poderia escolher entre trabalhar na atividade do projeto ou virar o mês, onde era realizado o pagamento dos funcionários e a dívida de empréstimo.

Devido às críticas recebidas com relação a dinâmica de turnos, nesta versão decidimos separar cada ação em turnos determinados. Dessa forma, o primeiro turno é onde os jogadores podem fazer a oferta para um funcionário. O segundo turno é reservado para os jogadores fazerem um lance em um projeto. No terceiro turno, os jogadores trabalham na atividade do projeto. Por fim, no quarto turno o mês é encerrado e os pagamentos realizados. Deste modo, todos os jogadores estarão atuando na mesma ordem cronológica e terão seus gastos ou lucros conquistados no mesmo momento.