



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE CIÊNCIAS**  
**DEPARTAMENTO DE COMPUTAÇÃO**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**  
**MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO**

**RUBENS ANDERSON DE SOUSA SILVA**

**D-CREEA: DSML PARA CRIAÇÃO DE JOGOS ANALÓGICOS EDUCACIONAIS DE  
CARTAS**

**FORTALEZA**

**2022**

RUBENS ANDERSON DE SOUSA SILVA

D-CREEA: DSML PARA CRIAÇÃO DE JOGOS ANALÓGICOS EDUCACIONAIS DE  
CARTAS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação.

Orientadora: Profa. Dra. Rossana Maria de Castro Andrade.

FORTALEZA

2022

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Sistema de Bibliotecas  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

S583d Silva, Rubens Anderson de Sousa.

D-CREEA: DSML para criação de jogos analógicos educacionais de cartas / Rubens Anderson de Sousa Silva. – 2022.

96 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2022.

Orientação: Profa. Dra. Rossana Maria de Castro Andrade.

1. Informática na Educação. 2. Jogos Educacionais. 3. Jogos Analógicos de Cartas. 4. MDE. 5. DSML.  
I. Título.

CDD 005

---

RUBENS ANDERSON DE SOUSA SILVA

D-CREEA: DSML PARA CRIAÇÃO DE JOGOS ANALÓGICOS EDUCACIONAIS DE  
CARTAS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação.

Aprovada em:

BANCA EXAMINADORA

---

Profa. Dra. Rossana Maria de Castro Andrade (Orientadora)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Ismayle de Sousa Santos  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Windson Viana de Carvalho  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Victor Travassos Sarinho  
Universidade Estadual de Feira de Santana (UEFS)

À minha família, por sua capacidade de acreditar em mim e investir em mim. Mãe, seu cuidado e dedicação foi que deram, em alguns momentos, a esperança para seguir.

## AGRADECIMENTOS

À Profa. Dra. Rossana Maria de Castro Andrade por me orientar em minha dissertação de mestrado.

À minha mãe, Maria Nilda Ramos de Sousa, que nunca me deixou faltar nada mesmo nos momentos mais difíceis.

Aos amigos de laboratório Belmondo Rodrigues, Joseane Paiva, Bosco Borges, Pedro Almir, Evilásio Jr., Fátima Pinheiro, Erick Barros, Nayana Carneiro e muitos outros que me acompanharam nessa jornada.

Agradeço a todos os professores por me proporcionar o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional, por tanto que se dedicaram a mim, não somente por terem me ensinado, mas por terem me feito aprender.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), na pessoa da Cláudia Queda de Toledo, pelo financiamento da pesquisa de mestrado via bolsa de estudos.

Aos bibliotecários da Universidade Federal do Ceará: Francisco Edvander Pires Santos, Juliana Soares Lima, Izabel Lima dos Santos, Kalline Yasmin Soares Feitosa e Eliene Maria Vieira de Moura, pela revisão e discussão da formatação utilizada neste *template*.

Ao aluno Thiago Nascimento do curso de ciência da computação da Universidade Estadual do Ceará que elaborou o *template* do qual este trabalho foi adaptado para Universidade Federal do Ceará.

“O sonho é que leva a gente para frente. Se a gente for seguir a razão, fica aquietado, acomodado.”

(Ariano Suassuna)

## RESUMO

Os jogos educativos digitais e analógicos podem promover uma experiência lúdica de aprendizagem por meio da resolução de problemas relacionados à matéria de uma disciplina e colocar o aluno como principal ator no processo de aprendizagem. Além disso, na utilização de jogos educacionais analógicos, essa experiência proporciona uma discussão presencial sobre o assunto, com interação tátil e feedback rápido. No entanto, desenvolver um jogo educacional não é trivial, porque exige, além de uma base científica, conhecimento em educação e experiência em design de jogos. Os conceitos de *Model-Driven Engineering/Engenharia Orientada a Modelos (MDE)*, entre eles o de *Domain-Specific Modeling Language/Linguagem de Modelagem Específica de Domínio (DSML)*, têm sido usados no desenvolvimento de jogos educacionais digitais para facilitar o processo de desenvolvimento de forma padronizada e semiautomática. Sendo assim, o objetivo deste trabalho é primeiro desenvolver uma DSML para a criação de jogos analógicos educacionais de cartas, denominada D-CreEA Card Games e, em seguida, uma plataforma WEB baseada nesta DSML. A motivação deste trabalho é auxiliar professores da área de Computação na construção de metodologias de aprendizagem lúdica através de jogos. Para a avaliação da D-CreEA, professores de Ciência da Computação utilizaram a plataforma para desenvolver um jogo de cartas via manipulação de um jogo padrão, passando por todas as fases de desenvolvimento do mesmo. Esta avaliação revelou que a plataforma D-CreEA pode ser de fato útil para a criação de jogos analógicos, assegurando todos os quesitos de qualidade considerados em um nível satisfatório, segundo os participantes da avaliação.

**Palavras-chave:** informática na educação; jogos educacionais; jogos analógicos de cartas; MDE; DSML.

## ABSTRACT

Educational games can promote a playful learning experience by solving problems related to the educational content of a subject, placing the student as the main actor in the learning process. In addition, with analog educational games, this experience provides a face-to-face discussion on the subject, with tactile interaction and quick feedback. However, developing an educational game is not trivial, since it requires knowledge in education, game design and scientific background. To facilitate this process, one can use the concept of MDE (Model-Driven Engineering) and DSML (Domain-Specific Modeling Language), in a standard and semi-automated way. In this context, the objective of this work is first to develop a DSML to create educational analog card games, called D-CreEA Card Games, and then a WEB platform based on the DSML. The motivation of this work is to help professors in the Computing field to construct playful learning methodologies using games. To evaluate D-CreEA, Computer Science professors had used the platform to develop a card game through the manipulation of a standard game, at all stages of the game development. The evaluation revealed that the D-CreEA platform can indeed be useful for creating educational analog games, satisfying all aspects of quality considered, according to the participants.

**Keywords:** informatics in the education; educational games; analog card games; MDE; DSML.

## LISTA DE FIGURAS

Figura 1 – Metodologia de desenvolvimento da DSML D-CreEA . . . . .	19
Figura 2 – Metodologia de desenvolvimento da Plataforma D-CreEA . . . . .	21
Figura 3 – Relação entre custo e benefício de uma pontuação . . . . .	30
Figura 4 – Mapa do investimento cognitivo ou “Taste Maps”. Fonte: (O’SHEA; FREE-MAN, 2019) (adaptado) . . . . .	33
Figura 5 – Sintaxe Abstrata - <i>DSML To Create Educational Analog Card Games/DSML Para a Criação de Jogos Educacionais Analógicos de Cartas (D-CreEA)</i> . . .	44
Figura 6 – Modelo do Game Loop - Exemplo do <i>D-CreEA</i> . . . . .	46
Figura 7 – Game Loop - GREaTest Card Game . . . . .	48
Figura 8 – Exemplo de regra do GreaTest Card Game gerado pelo D-CreEA . . . . .	48
Figura 9 – As informações do jogo geradas por the <i>D-CreEA</i> . . . . .	48
Figura 10 – Cartas do GreaTest Card Game geradas pelo <i>D-CreEA</i> . . . . .	49
Figura 11 – Diagrama de Casos de Uso da D-CreEA . . . . .	56
Figura 12 – Arquitetura da plataforma D-CreEA . . . . .	58
Figura 13 – Diagrama de Classes - <i>D-CreEA</i> . . . . .	59
Figura 14 – <i>Landing page</i> da D-CreEA . . . . .	61
Figura 15 – Login da D-CreEA . . . . .	62
Figura 16 – Game Design Canvas da D-CreEA . . . . .	62
Figura 17 – Diagramas de um jogo da D-CreEA . . . . .	63
Figura 18 – Tela de edição de regras e estados da D-CreEA . . . . .	65
Figura 19 – Criação de Regras da D-CreEA . . . . .	66
Figura 20 – Tela de edição de Decks da D-CreEA . . . . .	67
Figura 21 – Tela de edição de cartas da D-CreEA . . . . .	68
Figura 22 – Tela de impressão da D-CreEA . . . . .	69
Figura 23 – Características escolhidas no questionário pré-uso . . . . .	75
Figura 24 – Características escolhidas no questionário pré-uso versus pós-uso . . . . .	76
Figura 25 – Relação entre características e satisfação da D-CreEA . . . . .	76
Figura 26 – Protótipo de alta fidelidade da tela de GDD da D-CreEA . . . . .	91
Figura 27 – Protótipo de alta fidelidade da tela de regras da D-CreEA . . . . .	92
Figura 28 – Protótipo de alta fidelidade da tela de criação de baralhos da D-CreEA . . .	93
Figura 29 – Protótipo de alta fidelidade da tela de criação de cartas da D-CreEA . . . . .	94

## LISTA DE QUADROS

Quadro 1 – PICO para a busca de trabalhos relacionados . . . . .	36
Quadro 2 – Trabalhos relacionados . . . . .	38
Quadro 3 – Jogos base para o domínio . . . . .	43
Quadro 4 – Trabalhos publicados ao longo desta pesquisa . . . . .	81

## LISTA DE ABREVIATURAS E SIGLAS

<i>BGG</i>	<i>Board Game Geek</i>
<i>CRUD</i>	<i>Create, Read, Update, Delete</i>
<i>D-CreEA</i>	<i>DSML To Create Educational Analog Card Games/DSML Para a Criação de Jogos Educacionais Analógicos de Cartas</i>
<i>DSML</i>	<i>Domain-Specific Modeling Language/Linguagem de Modelagem Específica de Domínio</i>
<i>FQAD</i>	<i>Framework for Qualitative Assessment of DSMLs</i>
<i>GBL</i>	<i>Game-Based Learning/Aprendizagem Baseada em Jogos</i>
<i>GDC</i>	<i>Game Design Canvas</i>
<i>GTCG</i>	<i>GreaTest Card Game</i>
<i>JSON</i>	<i>JavaScript Object Notation</i>
<i>MDE</i>	<i>Model-Driven Engineering/Engenharia Orientada a Modelos</i>
<i>MVC</i>	<i>Model-View Controller</i>
<i>PICO</i>	<i>Population-Intervention-Comparison-Outcome/População-Intervenção-Comparação-Resultado</i>
<i>RFs</i>	<i>Requisitos Funcionais</i>
<i>RNFs</i>	<i>Requisitos Não-Funcionais</i>
<i>RSs</i>	<i>Requisitos do Sistema</i>
<i>RUs</i>	<i>Requisitos do Usuário</i>
<i>SaaS</i>	<i>System as a Service</i>
<i>UML</i>	<i>Unified Modeling Language/Linguagem de Modelagem Unificada</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
<b>1.1</b>	<b>Contextualização</b>	<b>14</b>
<b>1.2</b>	<b>Motivação</b>	<b>15</b>
<b>1.3</b>	<b>Problema e Objetivo de Pesquisa</b>	<b>16</b>
<b>1.4</b>	<b>Público-alvo</b>	<b>18</b>
<b>1.5</b>	<b>Metodologia</b>	<b>19</b>
<b>1.6</b>	<b>Resumo e Estrutura da Dissertação</b>	<b>22</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>23</b>
<b>2.1</b>	<b>Domain-Specific Modeling Language</b>	<b>23</b>
<b>2.2</b>	<b>Jogos</b>	<b>24</b>
<b>2.2.1</b>	<i>Elementos de um Jogo</i>	<b>25</b>
<b>2.2.2</b>	<i>Tipos de Gameplay</i>	<b>26</b>
<b>2.2.3</b>	<i>Interação do jogador em jogos de equipe</i>	<b>27</b>
<b>2.2.4</b>	<i>Balanceamento de Jogos</i>	<b>28</b>
<b>2.2.5</b>	<i>Jogos de mesa, tabuleiro e cartas</i>	<b>30</b>
<b>2.2.6</b>	<i>Jogos Educacionais</i>	<b>31</b>
<b>2.2.7</b>	<i>Diversão e Experiência</i>	<b>32</b>
<b>2.3</b>	<b>Resumo do Capítulo</b>	<b>34</b>
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>35</b>
<b>3.1</b>	<b>Revisão da Literatura</b>	<b>35</b>
<b>3.2</b>	<b>Discussão sobre os resultados da busca</b>	<b>37</b>
<b>3.3</b>	<b>Resumo do Capítulo</b>	<b>39</b>
<b>4</b>	<b>DSML PARA A CRIAÇÃO DE JOGOS ANALÓGICOS DE CARTAS</b>	<b>40</b>
<b>4.1</b>	<b>Domínio</b>	<b>40</b>
<b>4.2</b>	<b>Metamodelo</b>	<b>42</b>
<b>4.3</b>	<b>Modelo-para-Artefato</b>	<b>45</b>
<b>4.4</b>	<b>Modelo Guia</b>	<b>46</b>
<b>4.5</b>	<b>Validação da adequação da solução: Instância de um Jogo</b>	<b>47</b>
<b>4.6</b>	<b>Resumo do Capítulo</b>	<b>49</b>
<b>5</b>	<b>A PLATAFORMA D-CREEA CARD GAMES</b>	<b>51</b>

<b>5.1</b>	<b>Requisitos da D-CreEA</b>	51
<b>5.1.1</b>	<i>Estudo de Viabilidade</i>	51
<b>5.1.2</b>	<i>Elicitação e Análise de Requisitos</i>	52
<b>5.1.3</b>	<i>Especificação de Requisitos</i>	53
<b>5.1.3.1</b>	<i>Requisitos Funcionais do Usuário</i>	53
<b>5.1.3.2</b>	<i>Requisitos Funcionais do Sistema</i>	54
<b>5.1.3.3</b>	<i>Requisitos Não Funcionais</i>	56
<b>5.2</b>	<b>Projeto: Arquitetura e Diagrama de Classes</b>	57
<b>5.3</b>	<b>Implementação</b>	58
<b>5.4</b>	<b>A plataforma D-CreEA</b>	60
<b>5.4.1</b>	<i>Home</i>	60
<b>5.4.2</b>	<i>Login</i>	60
<b>5.4.3</b>	<i>Game Design Canvas</i>	60
<b>5.4.4</b>	<i>Diagrama</i>	62
<b>5.4.5</b>	<i>Gameplay</i>	64
<b>5.4.6</b>	<i>Decks</i>	65
<b>5.4.7</b>	<i>Cards</i>	67
<b>5.4.8</b>	<i>Impressão</i>	68
<b>5.5</b>	<b>Resumo do Capítulo</b>	69
<b>6</b>	<b>AVALIAÇÃO</b>	71
<b>6.1</b>	<b>Método de Avaliação</b>	71
<b>6.2</b>	<b>Avaliação Piloto</b>	73
<b>6.3</b>	<b>Resultados da avaliação com professores</b>	74
<b>6.4</b>	<b>Resumo do Capítulo</b>	77
<b>7</b>	<b>CONCLUSÃO</b>	79
<b>7.1</b>	<b>Visão Geral</b>	79
<b>7.2</b>	<b>Resultados e Contribuições</b>	80
<b>7.3</b>	<b>Limitações e Ameaças à Validade</b>	82
<b>7.4</b>	<b>Trabalhos Futuros</b>	83
	<b>REFERÊNCIAS</b>	85
	<b>APÊNDICE A–PROTÓTIPO DA INTERFACE WEB DA D-CREEA</b>	91
	<b>APÊNDICE B–QUESTIONÁRIO PRÉ-USO DA D-CREEA</b>	95

<b>APÊNDICE C–QUESTIONÁRIO PÓS-USO DA D-CREEA . . . . .</b>	<b>96</b>
---	-----------

# 1 INTRODUÇÃO

Neste capítulo é apresentada uma introdução a esta pesquisa através de uma visão geral do contexto desta pesquisa (Seção 1.1), sua motivação (Seção 1.2), problema e objetivo (Seção 1.3), público alvo (Seção 1.4), metodologia (Seção 1.5) e, por último, um resumo do capítulo e a estrutura da dissertação (Seção 1.6).

## 1.1 Contextualização

O papel do aluno nas aulas tem sido bastante discutido na literatura, trazendo, de um lado, o modelo jesuíta, método mais tradicional de ensino, em que o aluno absorve o conteúdo filtrado e interpretado pelo professor, com provas refletidas apenas no conteúdo apresentado; e do outro lado, métodos que promovem o protagonismo do aluno. O modelo tradicional é problemático, pois coloca o aluno como um indivíduo passivo e pouco crítico, sem espaço para exploração, descoberta e estudo prático (ANASTASIOU, 2007). Por outro lado, as metodologias ativas de ensino advogam que o aluno é importante no ato de construir as aulas e, para o professor, é herdado o papel do mediador ou facilitador (ANASTASIOU, 2007).

Um desses métodos ativos é a dinâmica lúdico-pedagógica, que inclui a *Game-Based Learning/Aprendizagem Baseada em Jogos (GBL)*, sendo conceituada como o uso de jogos e simulações para fins educacionais e fins de treinamento (FREITAS, 2006), a qual possui o principal propósito a melhora da motivação e do engajamento do aluno. Pesquisas sobre jogos com crianças e adultos em antropologia, psicologia e educação indicam que brincar é um importante mediador para a aprendizagem e socialização ao longo da vida (CSIKSZENTMIHALYI, 1990).

No trabalho de Dempsey (1999), observou-se a preponderância de jogos com a intenção de promover o aprendizado de habilidades e atitudes intelectuais de nível superior em oposição aos resultados de conhecimento verbal. Os jogos cumpriam muitas funções como tutoria, diversão, ajuda na exploração de novas habilidades, promoção da autoestima, prática de habilidades existentes, treinamento de habilidades existentes, automatizando e buscando mudar uma atitude (DEMPSEY *et al.*, 1996a).

Dado o papel natural que a brincadeira e a simulação têm para o desenvolvimento intelectual, jogos como veículo para brincadeira e simulação não são apenas uma diversão para as crianças, mas uma parte integrante de sua aprendizagem e vida social (FENGFENG, 2009). Além disso, é essencial promover a aprendizagem ancorada em situações autênticas (BROWN

*et al.*, 1989; CHOI; HANNAFIN, 1995), visto que essa abordagem faz com que os alunos se envolvam com o material, invocando assim um estado de “mindfulness” ou “flow”, no qual os alunos empregam processos meticulosamente orientados (SALOMON *et al.*, 1991) para a resolução de problemas.

Pode-se classificar a plataforma dos jogos em digitais, analógicos e híbridos, e esta mesma classificação é usada nos jogos educacionais. Nos jogos digitais, o (*gameplay*) envolve apenas plataformas digitais, como computador e *smartphones*; nos jogos analógicos, a jogabilidade (*gameplay*) inclui apenas peças e objetos físicos, como cartas, tabuleiro, *tokens*, dados, etc; e nos híbridos há uma mistura dos dois tipos.

## 1.2 Motivação

Pesquisas mostram que o mercado de jogos educacionais e sérios está em expansão. Como aponta o BNDES, em 2017, para empresas do setor de jogos digitais no Brasil, houve um crescimento no nível de investimento, tanto no Brasil quanto no resto do mundo. São R\$ 1,3 Bi investidos a nível nacional, representando um aumento em 7,3% nesse setor - em comparação a 143,5 U\$ Bi a nível mundial <sup>1</sup>.

Já para o mercado de jogos analógicos, 28% da população do Brasil faz uso de *boardgames*, seja jogos de cartas ou tabuleiro. O segmento representou 9,7% das vendas do setor de brinquedos em 2018, de um total de R\$ 6,87 bilhões. Além disso, a indústria dos jogos analógicos teve um crescimento de 7,5% no ano, mostrado em levantamento da Associação Brasileira dos Fabricantes de Brinquedos (Abrinq).

Essa maior adoção da metodologia de aprendizado baseado em jogos está positivamente relacionada à melhoria da motivação e engajamento do aluno em contexto de aula, com cooperação ou competição dentro de um cenário-problema em torno de um assunto educacional que deve ser elucidado para buscar a vitória (FREITAS, 2006). No caso de jogos analógicos, ainda há o fator de interação cara-a-cara, que ajuda a integrar os alunos em sala (XU *et al.*, 2012). Além disso, o uso de jogos no ensino faz parte da metodologia ativa de ensino para resolução de problemas presencial, que, além de ajudar na interação social, promove o protagonismo do aluno no ato de construir a aula, aspecto crucial na motivação e engajamento tanto do aluno quanto do professor no contexto de aula (ANASTASIOU, 2007).

Em resumo, as principais vantagens dos jogos educacionais analógicos são a dis-

---

<sup>1</sup> fonte: <https://bit.ly/3ke207D>

cussão face-a-face, o rápido *feedback* e a interação entre os alunos jogadores e até professores. Entretanto, esse tipo de jogo traz desafios inerentes ao design e prototipação, por tratar de interação tátil e de menor controle do *gameplay* em comparação aos jogos digitais, visto que no reino digital, o controle é automatizado. Além disso, jogos digitais já possuem vários *frameworks* de desenvolvimento (ex: Unity<sup>2</sup>, Unreal Engine<sup>3</sup>), ao contrário dos jogos analógicos. Dessa forma, optou-se neste trabalho por jogos educacionais analógicos com o objetivo de facilitar o seu design e prototipação.

### 1.3 Problema e Objetivo de Pesquisa

A sequência de eventos que ocorrem durante o jogo e o resultado desses eventos é, até certo ponto, desconhecida durante a fase de produção de um jogo (HUNICKE *et al.*, 2004). Além disso, existem vários exemplos de jogos que tinham potencial promissor, investimento e equipe, mas simplesmente não foram bem aceitos/avaliados no mercado.

Para o caso de desenvolvimento de jogos educacionais existe um agravante: é necessário alinhar os objetivos educacionais com fatores motivacionais, como imersão, diversão e *flow*, de forma que o jogo consiga ensinar/treinar um conteúdo pedagógico de forma substancial e efetiva, mas ainda assim consiga ser divertido para o aluno (FENGFENG, 2009; FREITAS, 2006).

Essa dificuldade é aumentada em jogos analógicos, pois a interação do jogador com o jogo é multimodal, sendo necessário controlar a jogabilidade manualmente ao se manipular *decks*, contar pontos ou interpretar e aplicar regras, por exemplo, uma das formas de facilitar o controle do *gameplay* é através do uso de computador/*smartphone* para controlar entradas e saídas automaticamente (BORGES *et al.*, 2019). Há também o custo adicional de desenvolvimento, design de cartas e implementação, pois para criar um jogo de tabuleiro educacional, todos os componentes físicos devem ser prototipados e produzidos, em várias iterações.

Dessa forma, o problema de pesquisa que motiva o desenvolvimento deste trabalho é o seguinte: *como criar um jogo de cartas analógico educacional personalizável ao conteúdo educacional do educador e de forma facilitada?*

Para solucionar esse problema, usar ferramentas de software que possam auxiliar ou automatizar em algum nível o desenvolvimento desses jogos pode ser uma alternativa viável.

---

<sup>2</sup> unity.com

<sup>3</sup> unrealengine.com

Já existem vários trabalhos que propõem guias de criação, abstrações de tipos de jogos e até *frameworks* para este fim (ver Capítulo 3).

Por exemplo, uma maneira de promover uma abordagem de desenvolvimento de jogos de forma eficaz é por meio de *frameworks* de software e metamodelos. Essas estruturas são definidas através de conceitos de *MDE*, que consiste em métodos de criação de softwares baseados em modelos, específicos para resolver problemas de domínio específicos. Para o *MDE*, o foco é o desenvolvimento de modelos que servirá de base para o produto final. Esses modelos auxiliam no processo de concepção, construção e prototipagem de um jogo de forma padronizada e semi-automatizada, pois oferecem uma estrutura pré-moldada para a geração de jogos. Isso acarreta a melhoria do refinamento dos jogos gerados, padronizando o desenvolvimento e definindo formalmente o conhecimento sobre aquele nicho de jogo.

A utilização de *MDE* é adequada para o problema abordado nesta dissertação visto que para desenvolver um jogo via *MDE*, é necessário apenas que o modelo gerado possa de fato ser usado para gerar jogos e este processo possa ser avaliado em quesitos de qualidade, como funcionalidade, extensibilidade, etc. Além disso, a linguagem no domínio específico descreve o comportamento, a estrutura e os requisitos dessa classe de problemas e sua conversão formal entre modelos e para artefatos de jogo. Assim, é possível descrever a transformação de informações de alto nível de domínio de jogos analógicos de cartas em artefatos de um jogo físico.

Contudo, o desafio aqui consiste no fato de que há uma lacuna/falta de trabalhos baseados em *MDE* para jogos analógicos. Por outro lado, esta lacuna acaba sendo o diferencial deste trabalho, pois abre uma oportunidade para promover o uso de *MDE* para este tipo de jogo e escopo, expandindo a borda de conhecimento sobre esse assunto.

A diferença entre *MDE* para jogos analógicos e digitais é a transformação de modelo para artefato, pois, nesse caso, os artefatos não consistem em software, mas sim em elementos de jogos analógicos, como cartas impressas, manual, entre outros.

Dessa forma, o objetivo deste trabalho é propor uma abordagem *MDE* para auxiliar no processo de criação de um jogo analógico educacional de cartas, que possa ser adaptado ao conteúdo pedagógico do professor e, ainda, ser construído com base na literatura de *game design*. É necessário então desenvolver e testar uma *DSML*, a qual é intitulada *D-CreEA*, para auxiliar a criação de jogos analógicos educacionais de cartas. A ideia é descrever o domínio dos jogos de cartas educacionais analógicos e gerar um metamodelo que consiga gerar modelos para a criação

do jogo final.

Para isso, os seguintes objetivos específicos devem ser alcançados:

1. Definir a linguagem de modelagem para o desenvolvimento de jogos analógicos, aqui chamada de D-CreEA;
2. Definir os requisitos, o projeto (design) e implementar uma plataforma de software que, baseada na linguagem D-CreEA, seja capaz de gerar jogos analógicos, plataforma esta, também chamada de D-CreEA;
3. avaliar a plataforma D-CreEA com professores da área de Computação para verificar a eficácia da mesma.

Ao final deste trabalho, espera-se contribuir para promover a discussão sobre MDE para jogos analógicos, uma vez que não foi encontrado outro trabalho com escopo e objetivos parecidos. Além disso, espera-se promover a adoção de jogos analógicos como ferramenta de ensino/aprendizagem, e isso está atrelado a vários aspectos do contexto educacional, como motivação e evasão.

#### 1.4 Público-alvo

O público-alvo do *D-CreEA* é formado por professores que buscam implementar um jogo educacional analógico de cartas e desenvolvedores de jogos analógicos de cartas.

Para o professor utilizar a *D-CreEA*, ele deve:

- Ser (ou ter na equipe) professores/instrutores, pois o desenvolvedor deve ser capaz de organizar o conteúdo, elencar os objetivos educacionais e entender a dificuldade de cada desafio;
- Ter (ou ter na equipe) conhecimento em UML;
- Ter (ou ter quem possua) nível de pensamento computacional bem desenvolvido para solucionar o problema do *game loop* através de estados sequenciais que encapsulam as regras do jogo (estados do jogo).

Vale ressaltar que o desenvolvimento pode ser feito por mais de uma pessoa, não se limitando a um professor, podendo incluir desenvolvedores de jogos, designers, entre outros. Contudo, uma diretriz deste trabalho é que o professor consiga gerar jogos de independente, utilizando somente a ferramenta de software implementada, os guias de criação e sua imaginação. Assim, o professor evita a necessidade de um game designer para criar seu jogo, embora seja muito recomendada a presença de um para guiar o desenvolvimento.

## 1.5 Metodologia

Para alcançar o objetivo deste trabalho, a metodologia científica de criação da DSML foi baseada em trabalhos relacionados que também propõem DSML para o desenvolvimento de jogos educacionais. Esta metodologia de criação da DSML consiste em 4 fases principais: *i) Fase de Domínio*, *ii) Fase de Metamodelo*, *iii) Fase de Conversão para Artefato*, e *iv) Fase de Avaliação*. A Figura 1 ilustra a metodologia e suas fases, as quais são mais detalhadas no Capítulo 4.

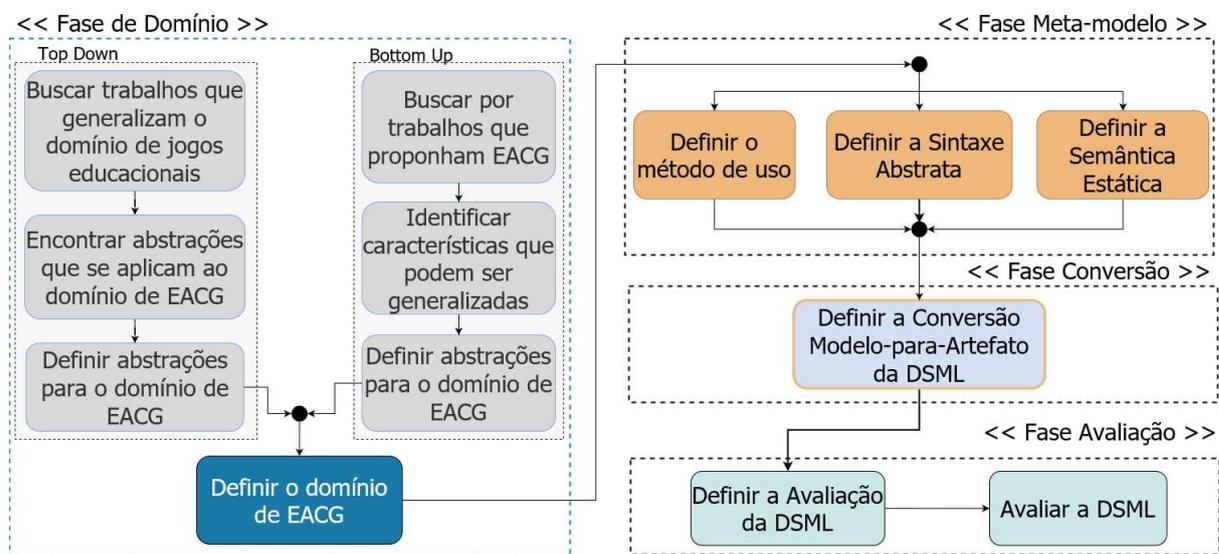


Figura 1 – Metodologia de desenvolvimento da DSML D-CreEA

A fase de domínio é o processo de identificação dos conceitos relevantes de um domínio de aplicativo, com foco em reaproveitar os conceitos identificados no domínio de *Educational Analog Card Games - EACG*. Os produtos do processo desta fase são as definições reutilizáveis dos conceitos do domínio de jogos analógicos de cartas, que são comuns para qualquer jogo desse tipo.

Para entender o domínio de jogos analógicos educacionais de cartas deste trabalho, pesquisas exploratórias paralelas foram desenvolvidas para encontrar trabalhos relacionados, métodos de avaliação de DSMLs e jogos educacionais analógicos. Para isso, utilizou-se duas abordagens: uma *Top-down* e uma *Bottom-up*.

O metamodelo de um domínio é formado por sintaxe abstrata de domínio, semântica estática e sintaxe concreta.

A sintaxe abstrata descreve os conceitos relevantes do domínio e as relações entre esses conceitos (ALTUMBAY *et al.*, 2009). Cada conceito de domínio é mapeado para uma meta

classe. A representação gráfica da Sintaxe Abstrata é apresentada na Figura 5, em notação *Unified Modeling Language/Linguagem de Modelagem Unificada (UML)* de diagramas de classes. Essa representação é adequada, pois o diagrama de classes tem o papel de representar conceitos, relações e entidades baseadas na realidade, indo ao encontro da definição de sintaxe abstrata.

A semântica estática de um metamodelo define o nível de precisão e completude das regras de formação do metamodelo (ALTUMBAY *et al.*, 2009). Essas regras de boa formação são usadas para definir as restrições sobre como os modelos podem ser formados, e validar os modelos construídos sobre um metamodelo específico.

A sintaxe concreta é usada para representar todos os conceitos visuais do domínio para um jogo específico, tendo como base os elementos identificados e descritos no sintaxe abstrata - basicamente, é a sintaxe abstrata aplicada a um jogo instanciado específico.

Na fase de conversão, são criados os algoritmos de conversão de modelo para artefato. Estes artefatos possuem quatro unidades: descrição de contexto, regras, descrição de gameplay e elementos físicos manipuláveis (i.e., cartas e manual do jogo). Estes algoritmos recebem a codificação da sintaxe concreta (ou sintaxe abstrata instanciada em um jogo) e retornam, de forma declarativa, um dos quatro artefatos de jogo.

Para a descrição do contexto do jogo, informações cruciais podem ser consultadas, como número de jogadores, descrição, contexto educacional, narrativa, etc. Estas informações devem ser inseridas pelo professor ao instanciar a sintaxe concreta, durante a criação do jogo.

Já para as regras, foram utilizados dois modelos de descrição: o simplificado e o estruturado. O simplificado consiste em texto aberto, onde é possível escrever o que quiser da forma que quiser. Já o estruturado segue a linha de *User Stories* (COHN, 2004) para definir qual jogador realiza a ação que afetará outro jogador/contexto de jogo, e sob quais condições. Escolhemos este modelo pois é amplamente conhecido na área da Computação, com a estrutura “Como <função>, eu quero <objetivo>, para que <efeito>” pode ser adaptada facilmente para descrever uma regra.

Para descrever o gameplay, utilizou-se diagrama de estados, cujas transições são regras associadas a cada estado do jogo dados acontecimentos de eventos. Assim, o professor pode indicar o estado inicial e relacionar as regras de mudança de estado para prosseguir com o gameplay até o estado final, de empate ou vitória de algum jogador. Vale ressaltar que as regras de criação e associação de estados e regras são descritas no Capítulo 4.

A fase de avaliação da DSML tem o objetivo de entender se a linguagem é realmente

capaz de gerar jogos analógicos a partir dos modelos criados pelo metamodelo. Para isso, buscou-se implementar uma versão do jogo *GreaTest Card Game (GTCCG)* (BEPPE *et al.*, 2018). GTCCG é um jogo de cartas analógico educacional que visa praticar o estudo de Teste de Software. O gameplay consiste em identificar o tipo de teste de software que, quando usado no cenário dado (ou seja, o desafio), iria detectar *bugs*.

Essa avaliação da DSML é importante pois revela aspectos cruciais da DSML, como adequação (ou seja, se a ferramenta realmente pode criar um jogo) ou se há falta de funcionalidade. Este jogo foi escolhido devido aos seus diferentes tipos de mecânicas, como negociação e bônus, já que esses não estão presentes no modelo inicial do D-CreEA, e ajudam a destacar possíveis limitações da DSML. Os resultados da avaliação são apresentados no Capítulo 6. Lá também estão apresentados os resultados da segunda avaliação - esta, mais relacionada à ferramenta de software (ver Capítulo 5).

Por outro lado, a avaliação da plataforma D-CreEA busca avaliar os níveis de qualidade atingidos pela plataforma, segundo a visão dos participantes da avaliação - estes, professores da área de Computação. O *framework* de avaliação foi o *Framework for Qualitative Assessment of DSMLs (FQAD)*, que tem o intuito de avaliar quesitos de satisfação dos usuários sobre a qualidade da DSML implementada.

Os participantes preenchem um formulário com suas expectativas sobre a plataforma, em seguida realizam tarefas padronizadas dentro da D-CreEA, e por último respondem um segundo questionário para saber se suas expectativas foram correspondidas ou frustradas, e em que nível. Além disso, eles opinam livremente sobre suas impressões sobre a D-CreEA. Os resultados dessa avaliação e o procedimento em detalhes podem ser vistos no Capítulo 6.

Já a metodologia de criação da plataforma D-CreEA segue o processo de criação Ágil de software, baseado na metodologia apontada em (SOMMERVILLE, 2015; PRESSMAN; MAXIM, 2021) e destacada na Figura 2. Cada fase da metodologia corresponde a uma etapa do modelo de processo de desenvolvimento de software: Elicitação e análise de requisitos, design e arquitetura, implementação e testes e avaliação (neste caso com o público-alvo). Cada etapa detalhada do modelo de processo adotado nesta pesquisa pode ser conferido no Capítulo 5.

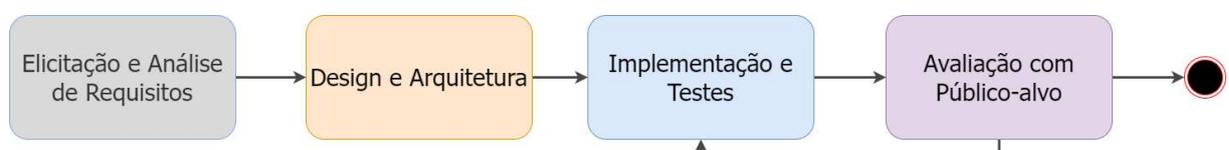


Figura 2 – Metodologia de desenvolvimento da Plataforma D-CreEA

## 1.6 Resumo e Estrutura da Dissertação

Este trabalho visa propor uma plataforma (i.e., ferramenta de software) baseada em uma DSML para a criação de jogos analógicos educacionais de cartas, denominada D-CreEA, capaz de prototipar jogos analógicos educacionais personalizados.

Neste capítulo foi apresentada uma visão geral do contexto desta pesquisa, sua motivação, problema, objetivo, resultados esperados, público-alvo e metodologia.

O restante desta dissertação é dividido da seguinte forma: no Referencial Teórico (Capítulo 2) discute-se o *background* necessário para o desenvolvimento do trabalho; nos Trabalhos Relacionados (Capítulo 3) apresenta-se trabalhos que tem relação a este em termos de contexto, resultados ou tipo de aplicação; no capítulo da DSML (Capítulo 4) é apresentada a aplicação da metodologia para a criação e avaliação da linguagem de domínio específico; a Plataforma D-CreEA é discutida no Capítulo 5; na Avaliação (Capítulo 6) são mostrados a metodologia de avaliação e seus resultados; e, por último, a Conclusão e os trabalhos futuros (Capítulo 7) são descritos.

## 2 REFERENCIAL TEÓRICO

Neste capítulo, o referencial teórico deste trabalho é discutido. Na Seção 2.1, fala-se sobre DSMLs e sua relação com MDE; em seguida, na Seção 2.2, discute-se os conceitos relacionados a jogos, incluindo jogos educacionais e de tabuleiro. Por último, a Seção 2.3 traz um resumo do capítulo.

### 2.1 Domain-Specific Modeling Language

Engenharia de Software Aplicada a Modelos (do inglês, Model-Driven Engineering, MDE) é uma metodologia de desenvolvimento de artefatos de software que busca explorar o domínio conceitual de problemas de software, representando todos os tópicos relacionados àquele problema (TANG *et al.*, 2008).

Para MDE, o foco está nos modelos gerados a partir de metamodelos criados via exploração do conhecimento do domínio. O conceito de Linguagem de Modelagem Específica de Domínio (do inglês, Domain Specific Modeling Language, DSML) pode ser definido como uma combinação de uma linguagem específica de domínio (DSL), um gerador de código específico de domínio (modelo para artefato) e uma estrutura de domínio para elevar o nível de abstração, além da codificação manual de software (por exemplo, diagramas). Assim, DSML é usada para aplicar MDE formalmente, cuja linguagem formal é a base da geração desses metamodelos. DSML usa definições de domínio para gerar um produto final em uma linguagem de baixo nível (por exemplo, programação, textual) escolhida a partir das especificações de alto nível (PRASANNA, 2012).

DSML é usada principalmente em situações em que sistemas semelhantes estão sendo desenvolvidos, pois através do uso de modelos padronizados, código e inteligência podem ser reutilizados, ajudando assim a aumentar a produtividade e auxiliar o especialista a formalizar o desenvolvimento prático de um tipo de aplicativo para outros desenvolvedores. Assim, as regras e diretrizes que foram incorporadas em uma ferramenta podem ser aplicadas a outras e podem até mesmo orientar ou alertar adequadamente os desenvolvedores sobre os requisitos e restrições desse domínio (PRASANNA, 2012). Essa linguagem pode ser utilizada em diversas áreas, como educação, saúde, *business*, entre outros.

Vale ressaltar que uma linguagem específica de domínio (Domain-Specific Language, DSL) é uma linguagem de implementação de software de alto nível que oferece suporte a

conceitos e abstrações que estão relacionados a um determinado domínio (aplicativo) (VISSER, 2007). Portanto, uma DSL possui uma coleção de frases em notação textual ou visual com sintaxe e semântica formalmente definidas. Uma gramática ou metamodelo define a estrutura das sentenças da linguagem e a semântica deve ser definida usando semântica matemática abstrata ou utilizando uma tradução para outra linguagem com semântica bem definida e já utilizada.

Uma DSL é de alto nível, abstraindo detalhes de implementação de baixo nível e, possivelmente, de particularidades da plataforma de implementação (VISSER, 2007). Nesta definição, uma DSL não necessariamente implementa software, mas artefatos práticos e utilizáveis que ajudam a resolver um problema de domínio específico. Para este trabalho, os artefatos são representações de jogos analógicos, como cartas, manuais, regras, entre outros.

A sintaxe abstrata de uma linguagem descreve o vocabulário de conceitos fornecidos pela linguagem e como eles podem ser combinados para criar modelos ou programas. Esta consiste em um conjunto de conceitos fornecidos e seus relacionamentos com outros conceitos, e também pode incluir regras que definem se um modelo é bem formado.

Por outro lado, a sintaxe concreta é o conjunto de notações que facilita a apresentação e construção dos construtos da linguagem. A sintaxe concreta pode ser formulada de maneira textual ou visual (HAHN, 2008).

A semântica estática tem o papel de restringir e representar informações que não podem (facilmente) ser mostradas por meio da representação sintática. Estas são as regras que governam a criação de todos os modelos de algum metamodelo (PRASANNA, 2012).

No contexto de jogos sérios (jogos com o intuito de instruir ou treinar um assunto não-lúdico), o uso de DSMLs auxilia no seu desenvolvimento por parte de pessoas com pouca ou nenhuma expertise em game design, como por exemplo, o educador. Este pode concentrar a função de projetar os jogos com pouco ou até sem o auxílio de um desenvolvedor de jogos, exigindo apenas os conhecimentos básicos. Essa facilidade ocorre porque todos os requisitos e restrições já foram formalmente definidos durante o projeto da DSML para aquele domínio, restando a tarefa de modelagem do jogo e a tradução em um artefato.

## 2.2 Jogos

Um jogo pode ser definido como “uma competição de habilidades e forças físicas ou mentais, exigindo que o(s) participante(s) siga(m) um conjunto específico de regras para atingir um objetivo”(HOGLE, 1996; DEMPSEY *et al.*, 1996b; MALONE, 1981).

### 2.2.1 Elementos de um Jogo

De acordo com Prensky et al. (PRENSKY, 2001), um jogo é uma estrutura de seis elementos: regras, metas e objetivos, resultados e *feedback*, conflito/competição/desafio/oposição, interação e representação ou história.

Para Battaiola et al. (BATTAIOLA; MUNHOZ, 2018), os elementos de um jogo também podem ser definidos por quatro elementos: Contexto, Artefato/Jogo, Atividade/Jogo e Agente/Jogador. Essa categorização leva em consideração o ambiente e a necessidade de uma oportunidade para que a jogabilidade ocorra.

O *framework* MDA (Mechanics, Dynamics & Aesthetics) (HUNICKE *et al.*, 2004) apresenta uma categorização direta de jogos, fundamentalmente dividida em três componentes:

- A *Mecânica* descreve os componentes específicos do jogo em termos de representação de dados e algoritmos. Eles são as várias ações, comportamentos e mecanismos de controle oferecidos ao jogador dentro de um contexto de jogo - a mecânica suporta a dinâmica do jogo;
- A *Dinâmica* descreve o comportamento do tempo de execução da mecânica agindo nas ações do jogador e nas reações dos outros jogadores longo do tempo - Uma Dinâmica funciona para criar experiências Aesthetics;
- *Aesthetics* (termo mais próximo da experiência) descreve as respostas emocionais desejáveis evocadas no jogador quando ele interage com o sistema de jogo. Cada jogo possui vários objetivos de experiência, em vários graus. A taxonomia definida nos ajuda a descrever os jogos para identificar a experiência desejada que o jogador deve ter com o jogo. Esta é definida da seguinte forma:
  - Sensação (jogo como prazer sentido),
  - Fantasia (jogo de faz-de-conta),
  - Narrativa (Jogo como um drama),
  - Desafio (jogo como uma pista de obstáculos),
  - Interação Social (jogo como estrutura social),
  - Exploração (jogo como um território desconhecido),
  - Expressão (Jogo como autodescoberta), e
  - Submissão (Jogo como um hobby - aqui, o termo mais correto seria Casual).

### 2.2.2 Tipos de Gameplay

Nesta seção são apresentados vários tipos e estratégias de gameplay, que impactam na estratégia de jogo e nas decisões do jogador. Entendê-las é necessário para evitar jogadas e estratégias infalíveis e evitar o desbalanceamento dos jogos. Existem duas estratégias principais: o tipo de jogada individualista e o multi-jogador.

No tipo de jogada individualista, às vezes chamada de “Bingo estratégico”, há decisões a serem tomadas, mas afetam apenas jogadores individuais (ENG, 2019). Neste caso, os jogadores estão em um jogo competitivo multi-jogador em tempo real, mas as ações que podem ser tomadas afetam apenas o seu próprio desempenho, mesmo que o objetivo final seja ganhar dos outros jogadores. Por exemplo, o clássico jogo Bingo, que é um jogo em que os jogadores realizam ações, mas afetam apenas seu próprio tabuleiro de jogo pessoal (ENG, 2019).

Já no tipo de jogo multi-jogador, toda ação afeta o desempenho de outro(s) jogador(es), ao contrário do jogo individualista. Contudo, existem tipos diferentes de impactos para cada ação. Soma-zero é um conceito em que a ação do jogador sempre gera desperdício ou demérito nos recursos de outro jogador. Por exemplo, se um jogador realizar uma ação e agora está em uma posição melhor ou "ganhando", então outro jogador perdeu sua posição e agora ele está "perdendo". O oposto também é verdade (ENG, 2019).

Em jogos de soma zero, seu sucesso deve vir às custas de outra pessoa. Não há ações ou recursos adicionais somados à economia do jogo, e isso implica que a perda de uma pessoa é o ganho de outra. Os tipos de jogada soma zero são os seguintes (ENG, 2019):

- *Atacando o líder*: Uma das ações mais populares desse tipo de jogo é atacar o líder ou, pelo menos, atacar a pessoa cuja posição é percebida como aquela que está ganhando. Isso geralmente é feito porque essa pessoa tem mais recursos, melhor posição ou outras vantagens que a tornam um alvo maduro. Caso contrário, existem as razões subjetivas para atacar o jogador que mais tem a perder sendo atacado.
- *Atacando o perdedor*: Da mesma forma, atacar o perdedor também pode ser uma estratégia viável, dado o design do jogo. Atacar o perdedor é algo que pode ser feito se a perda de posição ou recursos de um jogador valeria, para o jogador atacante, sua posição ou recursos. Atacar o perdedor - ou um jogador enfraquecido - pode ser vantajoso porque eles são um tipo de jogador que não pode montar uma defesa eficaz contra você.
- *Vencedor versus Classificação mais alta*: Não importa se um jogador decide atacar o jogador vencedor para fins subjetivos ou o jogador perdedor para fins estratégicos, a

principal motivação do jogador é continuar jogando para ganhar. No entanto, existem certas circunstâncias em que o jogador não consegue superar o jogador líder no jogo. Nestes casos, significa que o jogador pode melhorar a sua posição, mas não o suficiente para ganhar o jogo. É aí que o conceito de “jogar por posição” entra em jogo. Aqui, os jogadores realizam ações para ter um desempenho melhor do que os outros jogadores. Subjetivamente, isso significa que eles não irão terminar em último ou superar outro rival no jogo. Este tipo de decisão dá ao jogador a capacidade de redefinir sua posição. Eles podem não ser capazes de vencer, mas pelo menos eles podem fazer melhor do que outros jogadores.

- *Ajudar versus dificultar*: As ações de ajuda versus impedimento são determinadas pela natureza das ações do jogador. Alguns jogadores podem realizar ações que ajudarão a sua posição, a ganhar recursos ou a construir uma estratégia que os ajudará mais tarde no jogo. Caso contrário, eles podem ter que realizar uma ação contenciosa e ferir outro jogador no jogo. A primeira ação dá a eles as ferramentas necessárias para melhorar sua posição no jogo por meio de sua própria escala pessoal. Enquanto o último depende de contenção (e jogo de soma zero) para prejudicar a posição de outro jogador. No entanto, em certos jogos, você pode assumir a posição final de resolver ações que simultaneamente o ajudam enquanto magoam outra pessoa.

### **2.2.3 Interação do jogador em jogos de equipe**

Os jogos em equipe são especialmente importantes para a interação do jogador. Nesses jogos, os jogadores se envolvem em um jogo cooperativo com membros de sua própria equipe. Contudo, existem algumas vertentes de jogos nesse campo, como jogos equipe x equipe, equipe x objetivo e aliança.

Nos jogos equipe x equipe, como o próprio nome sugere, os jogadores formam equipes para derrotar a equipe adversária. Contudo, existem alguns padrões para formar equipes e gerar jogos desse tipo, como (ROCHA *et al.*, 2008):

- *Equipes Complementárias*: os jogadores encarnam papéis cujas habilidades possuem pontos fortes e fracos, e as equipes devem ser formadas por tipos de personagens cujas habilidades se completem.
- *Equipes Sinérgicas*: consiste na seleção de personagens que juntos conseguem ter habilidades que sozinhos não teriam, para alcançar um objetivo específico.

- *Equipes pedra-papel-tesoura*: equipes formadas por jogadores que servem para anular jogadores específicos da equipe rival.

Um exemplo que implementa esses padrões perfeitamente é o *Overwatch*<sup>1</sup>, em que equipes possuem jogadores com habilidades específicas, como curandeiro e escudo, e ainda, alguns personagens são mais fortes contra outros, e fracos contra terceiros.

Já nos jogos equipe x objetivo, todos os padrões de equipe x equipe se mantêm, mas agora o inimigo é um desafio em comum, que não envolve os jogadores diretamente. Nos jogos de tabuleiro, o desafio seria o próprio tabuleiro, que gera desafios randômicos para os jogadores resolverem.

Por último, sobre os jogos de aliança, os jogadores devem formar equipes mas ainda buscam ganhar dos seus companheiros. Isso significa que um jogador buscam ajudar seus companheiros a derrotar um objetivo em comum, seja uma equipe rival ou o “tabuleiro”, mas ainda busca individualmente fazer estratégias contra eles para sair vencedor.

#### **2.2.4 Balanceamento de Jogos**

Balanceamento das mecânicas de jogos é vista como essencial (ANDRADE *et al.*, 2006), mas muitas vezes é deixada de lado em vários jogos. Esta falta de cuidado pode impactar ainda mais quando o jogo é educacional, visto que uma mecânica “quebrada” pode destruir a diversão e impedir o jogo de seguir seu fluxo normal e o treinamento do assunto educacional.

Balanceamento de jogos consiste em criar a aparência de equidade (fairness) em um jogo (SCHREIBER, 2010). Um dos métodos mais tradicionais é o de experiência própria ou intuição, onde o designer “acha” ou “sente” que as escolhas para o jogo parecem certas para ele. Isso é muitas vezes baseado em *playtest*, um ciclo onde o designer projeta um protótipo, testa com jogadores e modifica de acordo com os relatos e experiências - contudo, esse método falta em conceitos formais. O método estatístico utiliza dados e metadados de jogos são capturados e interpretados por análise estatística para concluir a respeito do jogo. Um outro método é por modelagem matemática, de forma que o designer deve entender as relações entre os números presentes no jogo (ex.: pontuações) e ajustá-los para que os eventos que ocorram em *gameplay* possam ser previstos e ajustados para manter a equidade.

Para jogos digitais, existe o conceito de balanceamento de jogo dinâmico (ANDRADE *et al.*, 2006), em que o jogo (1) deve se adaptar ao nível inicial de qualquer jogador,

<sup>1</sup> jogo *overwatch*: <https://playoverwatch.com/pt-br/>

podendo variar de novato a expert, (2) se adaptar a evolução das habilidades do jogador e (3) o jogo deve manter essa adaptação transparente, para que o jogador não perceba que suas ações estão sendo observadas de perto.

Para alcançar o balanceamento dinâmico, o jogo deve constantemente medir o desempenho do jogador ao capturar e interpretar dados relacionados ao gameplay, e dessa forma, prover a intervenção. Essa relação entre os dados e a intervenção é chamada de “função desafio” (DEMASI; ADRIANO, 2003) apud (ANDRADE *et al.*, 2006), que deve mapear todas as ações do jogador em intervenções na dificuldade do jogo. Porém, é possível alterar elementos do jogo mais sutis, além da dificuldade, como o próprio ambiente simulado, deixando os recursos disponíveis ao jogador mais escassos ou abundantes, ou ainda, alterando o comportamento dos Non-Player Characters (NPCs).

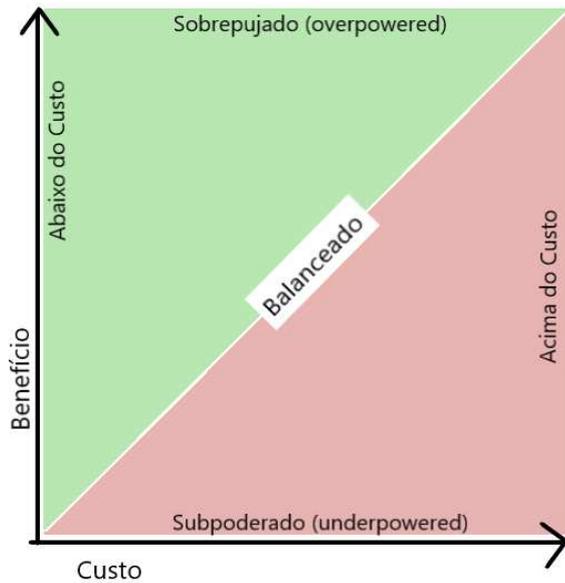
Já para os jogos analógicos, o balanceamento dinâmico não pode ser implementado em sua forma original, dado que não existe o elemento do computador que captura dados em tempo real e assim altera o gameplay. Dessa forma, o balanceamento deve ser feito no momento da criação e refinamento das mecânicas e regras, seguindo as técnicas de balanceamento tradicionais, como a pedra-papel-tesoura ou o playtest.

Por exemplo, em jogos com sistemas de recursos, o designer deve entender sobre como a oferta x demanda e inflação influenciam no preço dos recursos, trocas diretas entre jogadores e a mecânica de leilão. Muitas vezes, para casos em que há modelos matemáticos e estatísticos, o uso de gráficos (SCHREIBER, 2010) pode auxiliar a entender a relação entre custo x benefício de uma carta, por exemplo. A Figura 3 exemplifica um gráfico de custo x benefício de uma pontuação de um jogo.

Note que os pontos no plano acima da linha  $y=x$  são tidos como *overpowered* ou “boas” e as abaixo, como *underpowered* ou “ruins”, mas exatamente em cima dessa linha está o equilíbrio entre custo e benefício. Repare que quanto mais tipos de pontuações e tipos de custos forem adicionados no jogo, mais difícil é balancear o jogo.

Para o caso dos jogos educacionais, um problema cabal é ajustar o nível dos jogadores para que aqueles que sabem mais sobre o assunto não tenham uma vantagem extrema sobre os novatos. Uma estratégia usada em (BEPPE *et al.*, 2018) é adotar sorte para determinar quem ganha os pontos, mesmo que o jogador acerte a pergunta relacionado ao conteúdo pedagógico do jogo. Isso em si traz problemas, pois pode levar a sentimentos negativos como frustração, mas é uma opção que pode ser utilizada se bem ajustada.

Figura 3 – Relação entre custo e benefício de uma pontuação



Fonte: (SCHREIBER, 2010) (adaptado)

Para o caso dos jogos criados com o D-CreEA, o ideal é utilizar técnicas de modelagem matemática para ajustar os três tipos de pontuação presentes nas cartas. Esse ajuste é descrito na Seção 5.4.6.

### 2.2.5 Jogos de mesa, tabuleiro e cartas

O termo "jogo de mesa" é o termo mais amplo que cobre a maioria dos jogos geralmente jogados em uma mesa. Este termo inclui Jogos de tabuleiro, Jogos de cartas, RPGs de papel e caneta (Dungeons and Dragons entre outros.), e Jogos de batalha em miniatura (variando de grandes jogos de guerra a pequenos jogos corpo a corpo).

O que diferencia um jogo de tabuleiro de um jogo de cartas é o uso do tabuleiro como uma mecânica ativa, que não está incluída nos jogos de cartas. Assim, para jogos de cartas, a mesa onde o jogo de cartas é jogado não faz parte da jogabilidade, apenas do contexto do jogo.

Para classificar os jogos de tabuleiro, o trabalho de (KRITZ EDUARDO MANGELI, 2017) organiza e propõe uma ontologia baseada na classificação BGG<sup>2</sup> para tipos de mecânica de jogos de tabuleiro. O número de mecânicas se estende a mais de 50 - por exemplo, Take-That, onde um jogador utiliza um efeito para prejudicar outro, Apostas (*gambling*), Expressão, Dramatização, entre outros.

<sup>2</sup> BGG: <https://boardgamegeek.com/>

Além disso, os jogos de tabuleiro também podem ser divididos em duas grandes famílias (BOARD. . . , 2019): Hobby Games e Mainstream Games. Os jogos convencionais são amplamente populares e estão disponíveis nas lojas convencionais. Esses jogos tendem a ter menos regras, focando mais na facilidade de jogo. Geralmente são de baixo custo e não incluem expansões.

Por outro lado, os jogos Hobby (jogos de passatempo) são direcionados a pessoas que consideram o jogo um hobby dedicado. Eles são vendidos principalmente em lojas especializadas em jogos, sendo a maioria dos títulos difícil de encontrar. Esses jogos tendem a ter regras mais rígidas e oferecem uma gama mais ampla de dificuldades de jogo e faixas etárias-alvo do que os jogos convencionais. Frequentemente, os jogos de passatempo podem incluir expansões para adicionar à experiência de jogo para jogadores estabelecidos, mas podem custar um investimento considerável. Portanto, os jogos de passatempo costumam ser intimidantes ou pouco atraentes para o público em geral (BOARD. . . , 2019).

### **2.2.6 Jogos Educacionais**

Os jogos educacionais têm as seguintes propriedades:

- o fator facilitador associado aos jogos reduz a carga cognitiva e permite que os alunos usem sua preciosa memória de trabalho para tarefas de ordem superior (GREDLER, 2004).
- contextos utilizáveis: aprendizagem informal, jardim de infância/pré-escola, escola primária, escola secundária, educação de adultos, gestão de negócios, militar e saúde (FENG-FENG, 2009).
- resultados na motivação, o trabalho de (GULUMBAY; TURKEY, 2006, p. 69) aponta 13 componentes principais da motivação facilitada pelo jogo: apresentação da identidade, relações sociais, brincadeira, aprendizagem, realização, recompensas, imersão, contexto, fantasia, singularidade, criatividade, curiosidade, controle e propriedade.
- os gêneros incluem jogos de aventura, jogos de simulação, jogos de tabuleiro, jogos de quebra-cabeça, jogos de simulação de negócios, jogos de ação e jogos de estratégia. O estudo analisou a eficácia dos jogos através da realização de uma meta-análise da literatura sobre metodologia educacional e jogos relacionados. (FENGFENG, 2009).
- as áreas de aprendizagem incluem educação em ciências, matemática, artes da linguagem, leitura, física, saúde, ciências naturais, ciências sem conteúdo e habilidades sociais e desenvolvimento de habilidades gerais para resolução de problemas (FENGFENG, 2009).

Em relação aos jogos de tabuleiro educacionais, a principal vantagem sobre os jogos educacionais digitais é o contato e discussão face a face, promovendo a reflexão crítica e experimental, a interação social entre os alunos e o rápido *feedback* promovido pelo professor que pode estar supervisionando o partida. Além disso, os jogos educativos podem trazer um maior vínculo entre professores e alunos durante a aplicação do jogo. Em um ambiente mais descontraído, alunos e professores podem interagir de forma mais espontânea. Em um cenário de jogo digital, essa interação seria mais ampla, pois o aluno interage muito mais com o computador/*smartphone*.

Para experimentar este tipo de jogo, (YUSOF *et al.*, 2016) conduziu um experimento para testar a eficácia de aprendizagem, retenção e preferências dos alunos ao longo de três semestres comparando o jogo educacional de tabuleiro PHARM Game® com outros métodos de ensino. O estudo considerou duas variáveis de resultado: a primeira quantifica a compreensão e o conhecimento das informações nas sete áreas de conteúdo de cada aluno abordado no jogo. O segundo mede a percepção do impacto de vários exercícios laboratoriais e a eficácia e o prazer do aprendizado do jogo.

Os resultados demonstraram que o desempenho pós-teste foi significativamente maior do que as pontuações médias alcançadas após as aulas ou trabalhos de biblioteca. No entanto, não foi encontrada diferença significativa na pontuação média entre os dois pré-testes. O jogo era regularmente classificado em primeiro lugar pelos alunos como a maneira mais eficaz e divertida de aprender. Jogar melhorou o desempenho em testes de conhecimento e retenção de longo prazo mais do que outras ferramentas de ensino utilizadas.

### ***2.2.7 Diversão e Experiência***

Para entender o que é diversão e como avaliar se um jogo é divertido, é necessária uma abordagem psicológica, descompilando as emoções e o estado de diversão/prazer. O modelo de cinco fatores de personalidade (ou "Big 5") (SAUCIER; GOLDBERG, 1996; GOLDBERG, 1993) ajuda nesta tarefa. É um método psicológico para avaliar a personalidade de um indivíduo ao longo de cinco dimensões: Abertura à Experiência, Consciência, Amabilidade, Extroversão e Neuroticismo. No entanto, o modelo Big 5 é baseado em 30 facetas diferentes de traços de personalidade, então é considerado muito difícil de aplicar na prática. Para simplificar este processo de avaliação, foram criados Os 5 Domínios do Jogo (VANDENBERGHE, 2012) baseado no Big 5, que descreve as cinco dimensões das personalidades como Abertura à Experiência, Consci-

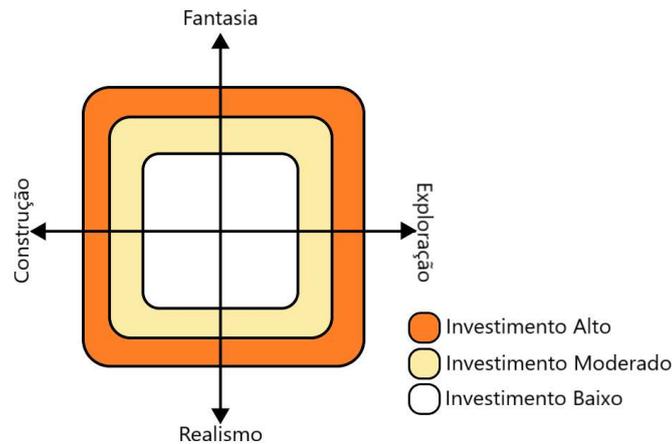


Figura 4 – Mapa do investimento cognitivo ou “Taste Maps”. Fonte: (O’ SHEA; FREEMAN, 2019) (adaptado)

enciosidade, Amabilidade, Extroversão e Neuroticismo - que foi "traduzido" em experiências ludológicas equivalentes.

Então, “Os motores do jogo”(The Engines of Play) condensou este sucessor (VANDENBERGHE, 2016), mantendo as principais facetas do Big 5 para game design, que busca abordar o problema de por que as pessoas jogam e quais motivações os designers se envolvem com a jogabilidade. Os componentes, ou “Motores”, do modelo incluem Mapas do Taste (VANDENBERGHE, 2013); The Consumers Journey; and Motivation and Satisfaction (through the application of A jornada dos consumidores; e Motivação e Satisfação (por meio da aplicação da Teoria da Autodeterminação (Self-Determination Theory).

O mapa do investimento cognitivo (Figura 4) representa a profundidade do investimento que os jogadores terão em relação a uma experiência particular. Ao usá-lo, os desenvolvedores têm o entendimento de que o meio do mapa representa uma distribuição normal (onde a maioria dos jogadores reside) e que eles terão níveis médios de investimento para aquela experiência de jogo ou mecânica em particular. No entanto, conforme o jogador se aproxima das bordas externas do gráfico através de maior profundidade ou especialização no jogo, o nível de paixão dos jogadores por essa experiência aumenta, enquanto o número de jogadores diminui proporcionalmente (MORA-ZAMORA; BRENES-VILLALOBOS, 2019).

Sobre o sentimento de diversão em jogos, foram propostos quatro elementos para categorizar diversão (LAZZARO, 2004):

- *Easy Fun* está associado a emoções como *admiração*, *curiosidade* e *surpresa*. Ele estimula a criatividade e é inerentemente envolvente, sem a adição de pontuações ou pontos. O

- fracasso pode ser tão recompensador quanto o sucesso.
- *Hard Fun* está ligado ao processo de progressão e desafio significativo. Os jogadores podem sentir frustração, fiero (“triunfo pessoal sobre a adversidade (LAZZARO, 2004, p. 6)”) e, em seguida, alívio. *Hard Fun* mantém os jogadores engajados através do desenvolvimento contínuo de suas habilidades sobre obstáculos e objetivos; no entanto, depende em manter um equilíbrio entre a relação desses estados, frustração e recompensa, ou pode irritar os jogadores.
  - *People Fun* leva em conta o efeito da interação social na jogabilidade, promovendo competição, cooperação, desempenho e espetáculo enquanto gera diversão ou experiências de união. Das Quatro Chaves, *People Fun* está associada à maior diversidade de emoções, incluindo *schadenfreude* (“regozijar-se com infortúnios dos outros (LAZZARO, 2004, p. 6)”) e *amici* (“amizade (DESPAIN, , p. 24)”).
  - *Serious Fun* é uma expressão de valor para o jogador. Eles se envolvem com ele para encontrar alívio, relaxamento ou excitação - enquanto seus processos criam significado e permitem que eles se envolvam com pensamentos, sentimentos e comportamento. Pode assumir a forma de interação física (por exemplo, jogos de dança) ou relaxamento do estresse por meio de ação ou jogo meditativo.

### 2.3 Resumo do Capítulo

Neste Capítulo apresentou-se conceitos relacionados a MDE, DSML e desenvolvimento de jogos. Sobre MDE, viu-se que o foco está nos modelos gerados a partir de um domínio específico de problemas, e que para implementá-la é utilizado DSML como metodologia. Viu-se ainda que DSML é a combinação de uma linguagem formal, um gerador de código via modelo, e uma estrutura de abstração. Os contextos aplicáveis dependem da existência de padrões e repetições que as soluções para esses problemas específicos apresentam e assim possam ser abstraídos.

Depois disso, foi apresentado o conteúdo sobre desenvolvimento de jogos, discutindo as categorizações de elementos de jogos já propostos na literatura, os tipos de gameplay (individualista x multi-jogador), os tipos de interação entre jogador e jogo, balanceamento de jogos, jogos analógicos (tabuleiro x cartas), jogos educacionais e seu papel de diversão versus educação, e, por último, os tipos de diversões e emoções envolvidas nos jogos.

### 3 TRABALHOS RELACIONADOS

Este capítulo apresenta os trabalhos relacionados ao tema desta dissertação. Aqui buscou-se entender quais técnicas de desenvolvimento de jogos são mais utilizadas através de uma revisão da literatura rigorosa (Seção 3.1) e quais são as mais indicadas para o caso dos jogos educacionais de cartas (Seção 3.2). Finalmente, o capítulo é resumido na Seção 3.3.

#### 3.1 Revisão da Literatura

Os trabalhos relacionados discutidos neste capítulo são aqueles que propõem ou utilizam uma estrutura MDE para auxiliar a concepção de jogos educacionais digitais ou analógicos. Para encontrar esses trabalhos, promoveu-se uma pesquisa exploratória dentro do tema de desenvolvimento de jogos educacionais, sob duas abordagens: de forma manual e de forma automática.

Para a busca manual, foi utilizada principalmente a ferramenta *Google Scholar*, onde os trabalhos eram escolhidos, baseando-se nos seus títulos. Foram encontrados trabalhos que apresentavam no título *frameworks*, estruturas, *DSMs*, *DSLs*, *DSMLs*. Destes, viu-se que uma abordagem que ia ao encontro dos objetivos desta pesquisa era através do uso de *DSMLs*, pois nos trabalhos encontrados sempre há a generalização do domínio de jogos.

Com esta constatação, focou-se em trabalhos relacionados a *DSMLs* e, a partir disso, criou-se o protocolo de pesquisa automática baseado em mapeamentos sistemáticos (PETERSEN *et al.*, 2008), apresentado a seguir.

A questão de pesquisa foi: RQ1 - “*Quais são as linguagens de domínio específico usadas para a criação de jogos educacionais, sejam analógicos ou digitais?*” e uma subpergunta: a: “*Quais destas são usadas para a criação de jogos educacionais de cartas/tabuleiro/de mesa?*”.

A base de trabalhos científicos escolhida para buscar por trabalhos relacionados foi a *Scopus*, por indexar artigos de vários anais de eventos.

Para a criação da *string* de busca, utilizou-se a estrutura *Population-Intervention-Comparison-Outcome/População-Intervenção-Comparação-Resultado (PICO)* (YENSEN, 2013). Esta serviu de guia para estruturar os termos de busca nas máquinas de pesquisa das bases científicas. O Quadro 1 mostra a estrutura PICO da *string* de busca.

A PICO resultante da pergunta RQ1 foi:

- População: *modelos de desenvolvimento de jogos analógicos ou digitais;*

Quadro 1 – PICO para a busca de trabalhos relacionados

Termo em português	sinônimos	termos em inglês	PICO
modelo	modelo, padrão	model, template, pattern	P
linguagem	linguagem	language, domain specific modeling language, domain specific language, DSL, DSML	I
jogo educacional	jogo sério, jogo instrucional, jogo educativo	educational game, serious game, instructional game	O

Fonte: elaborado pelo autor.

- Intervenção: *DSLs, DSMLs, e frameworks ou plataformas que implementem DSMLs para a criação de jogos educacionais;*
- Comparação: -
- Resultado: *jogo educacional;*

Os Critérios de Inclusão usados para selecionar os trabalhos encontrados foram: (i) estar escrito em português ou inglês e (ii) reportar um estudo primário ou secundário, i.e., ser baseado em pesquisa empírica ou revisão da literatura.

Já o critério de exclusão foi estar indisponível para consulta.

A *string* de busca resultante, na sintaxe da Scopus, é dada no Código-fonte 1.

Código-fonte 1 – string de busca - Scopus

```

1 TITLE((language OR DSL OR DSML OR DSM OR "modeling language
  " OR "domain specific") AND ("Educational Game" OR "
  serious game" OR "instructional game")) AND ( LIMIT-TO (
  DOCTYPE,"cp" ) OR LIMIT-TO ( DOCTYPE,"ar" ) ) AND (
  LIMIT-TO ( LANGUAGE,"English" ) OR LIMIT-TO ( LANGUAGE,"
  Portuguese" ) )
2 }

```

Com a execução da busca, 64 trabalhos foram identificados como potenciais relacionados, organizados e separados em suas categorias. Os trabalhos foram classificados por título como potencialmente irrelevantes ou potencialmente relevantes, pois alguns apresentavam resultados que fogem do interesse central da pesquisa, como jogos educacionais, aprendizado de idiomas/linguagens, dentre outros.

Após a execução, apenas 5 foram classificados como relacionados. A partir desses,

foi feito um processo de *snowballing* para encontrar mais trabalhos, que consiste em avaliar a lista de referências e a lista de citações de estudos já conhecidos para identificar novos estudos relevantes para a pesquisa. Além disso, os trabalhos encontrados manualmente ajudaram a solidificar a base de estudos relacionados. Ao final desse processo, um total de 21 trabalhos foram selecionados. Os trabalhos relacionados são mostrados no Quadro 2.

Contudo, vários artigos, mesmo não sendo relacionados diretamente, são relevantes por apresentarem resultados de interesse, que ajudam a solidificar o referencial teórico e ampliar a visão sobre o assunto, como *frameworks*, *guidelines*, revisões da literatura em temas paralelos e jogos educacionais.

### **3.2 Discussão sobre os resultados da busca**

Em resumo, os trabalhos encontrados possuem mecânicas diversas, incluindo Quiz, Point-and-Click, Sandbox, etc., o que é positivo, pois não vincula o tema do jogo educativo a uma pequena gama de jogos. Além disso, os conteúdos pedagógicos buscam ser mais gerais, não focando apenas em um assunto específico.

Alguns trabalhos propõem uma figura do desenvolvedor do jogo como necessária no desenvolvimento do jogo, o que é uma abordagem correta. Contudo, difere da diretriz deste trabalho, que é simplificar o processo de desenvolvimento para que o professor que, possivelmente tem uma base teórica de jogos baixa, consiga criar seus jogos de forma eficaz.

Este trabalho se assemelha aos elencados anteriormente por estar inserido no contexto de desenvolvimento do MDE, mas difere deles, pois o tipo de jogo gerado é analógico.

Por outro lado, o resultado desta pesquisa na literatura não identificou nenhum trabalho que propusesse um modelo dentro do tema MDE para o desenvolvimento de jogos analógicos, e desta forma, constatou-se que pode haver uma carência de trabalhos dentro desse domínio. A falta de eficácia na cobertura da pesquisa deste artigo é mantida em mente, mas a lacuna ainda se mantém, dada a dificuldade de encontrar este tipo de trabalho.

Assim, este trabalho contribui para ajudar a preencher essa lacuna ao propor uma abordagem MDE por meio de uma DSML para o desenvolvimento de jogos de cartas analógicos educacionais. Com isso, promovemos a discussão sobre a adoção de MDE para jogos analógicos.

Por último, apresentamos um resumo dos trabalhos que mais se aproximam à esta pesquisa, seja por ter como tema jogos analógicos ou por propor uma DSML.

O trabalho de (ALTUMBAY *et al.*, 2009) propõe um modelo para o desenvolvimento

Quadro 2 – Trabalhos relacionados

Título	Contribuição	Mecânica (HUNNICKE <i>et al.</i> , 2004)	Quem desenvolve	Plataforma	Validação / Avaliação
Este Trabalho	DSML	Variável	Professor or GD	Analog	Estudo de Caso
(ALTUMBAY <i>et al.</i> , 2009)	DSML	Tabletop Game	Não descrito	digital	Estudo de Caso
(LIMA, 2018)	framework	Card Game	Card Trading	digital	Estudo de Caso
(ZAHARI <i>et al.</i> , 2020)	DSML	Adventure	Não descrito	digital	FQAD
(TLILI <i>et al.</i> , 2016)	DSGL	RPG	Não descrito	digital	Avaliação com GDs
(GARCÍA <i>et al.</i> , 2019)	DSL	Quiz	Professor	digital	Estudo de Caso
(PRASANNA, 2012)	DSML	Story telling	Não descrito	digital	Estudo de Caso
(DUVAL <i>et al.</i> , 2015)	DSGL	Simulation	Instructor	digital	Estudo de Caso
(BELLOTTI <i>et al.</i> , 2009)	modelo	RPG	GD ou professor	digital	Estudo de Caso
(MARCHIORI <i>et al.</i> , 2011)	DSVL	Point and Click	Professor	digital	Estudo de Caso
(TROYER <i>et al.</i> , 2017)	DSML	Narrative	professor	digital	Estudo de Caso
(ZIATDINOV; KHAFIZOV, 2016)	DSL	Não descrito	GD	digital	Estudo de Caso
(MOSSMANN <i>et al.</i> , 2016)	DSL	Não descrito	GDs e Profissionais da saúde	digital	Estudo de Caso
(HAMIYE <i>et al.</i> , 2019)	Framework	Não descrito	professor	digital	Estudo de Caso
(CANTERI <i>et al.</i> , 2019)	Framework	Não descrito	professor	digital	Estudo de Caso
(TANG <i>et al.</i> , 2008)	DSML	Adventure RPG Puzzle Games	Professor	digital	Estudo de Caso
(HIRDES; LEIMEISTER, 2013)	DSL	Não descrito	GD	digital	Não descrito
(THILLAINATHAN, 2013)	Framework	Não descrito	Não descrito	digital	Não descrito

Fonte: elaborado pelo autor.

de jogos de tabuleiro digitais, cujo domínio é formado por *GameEngine*, *GameElement*, *Player*, *Event*, *Action*, *GameState*, *Goal*, *Sub-Goal*, *Non-MovableElement*, *MovableElement* e *Regras*. Este trabalho se relaciona ao D-CreEA por propor um modelo relacionado a jogos de tabuleiro. Contudo, ele tem como foco os jogos de tabuleiro digitais e, conseqüentemente, possui elementos que não se aplicam aos jogos analógicos de cartas, como *Movable Elements* e *GameEngine*.

O trabalho de (ZAHARI *et al.*, 2020) apresenta uma extensão da Linguagem de Modelagem *GLiSMo* (ZAHARI *et al.*, 2016), uma DSML para organizar a lógica e a estrutura de um jogo educacional. Este adiciona elementos da Teoria do *Flow* e do gênero de jogo *Aventura*. Para a avaliação, foi utilizado o *framework* FQAD (do inglês, *Framework for Quality Assessment of DSMLs*) (KAHRAMAN; BILGEN, 2013), que tem como objetivo avaliar DSMLs. O trabalho se assemelha ao D-CreEA por propor uma DSML para jogos, e por utilizar o FQAD para avaliação, mas se distancia por tratar de jogos digitais.

O trabalho de (PRASANNA, 2012) propõe um DSL para o desenvolvimento de jogos digitais educacionais, com foco na história do jogo (enredo). Seu domínio tem elementos como *Start Element*, *Mission Elements*, *Gameplay Elements* e *Stop Elements*. Este trabalho difere de D-CreEA pois, primeiro, não usa UML, o professor precisa criar uma história de fundo e, segundo, foca em jogos digitais. O trabalho se assemelha ao D-CreEA por propor uma linguagem para descrever jogos educacionais, mas se distancia por tratar de jogos digitais.

O último artigo selecionado é o trabalho de (TLILI *et al.*, 2016) propõe um DSGL (linguagem gráfica de domínio específico) para Jogos Digitais Educacionais do tipo RPG. Resume-se a um diagrama de classes sem a implementação do mecanismo de modelo para artefato. Para avaliação, utilizou-se desenvolvedores de jogos (GD). Não está claro quem deve desenvolver os jogos, se educadores ou GDs. Este trabalho se assemelha a D-CreEA por descrever o domínio dos jogos educacionais, mas se dissocia por tratar de jogos digitais.

### **3.3 Resumo do Capítulo**

Este capítulo abordou os trabalhos relacionados desta dissertação, apresentando os métodos de busca - manual e automática - e seus resultados. A busca manual focou em encontrar trabalhos que propusessem técnicas/métodos de criação de jogos analógicos educacionais via Google Scholar, a fim de entender, de forma preliminar, o que já existe na literatura para esse fim. Com esse entendimento, focou-se em linguagens de domínio específico para a criação de jogos.

Com isso, a string para a busca automática foi desenvolvida e aplicada. Nesta etapa, foram encontradas várias DSMLs e suas variações, apresentadas no Quadro 2. Como conclusão, percebeu-se que existe uma lacuna na literatura em relação a trabalhos com foco em MDE para jogos analógicos.

## 4 DSML PARA A CRIAÇÃO DE JOGOS ANALÓGICOS DE CARTAS

Este capítulo apresenta os resultados do desenvolvimento da *DSML* para a criação de jogos analógicos de cartas, seguindo a metodologia apresentada na Seção 1.5. A Seção 4.1 mostra como aconteceu a definição do domínio. A Seção 4.2 apresenta o metamodelo e a Seção 4.3 mostra a transformação deste modelo para artefato. Já a Seção 4.4 apresenta um modelo guia para o desenvolvimento dos jogos e a Seção 4.5 mostra a fase de validação da adequação da linguagem. Por fim, a Seção 4.6 apresenta um resumo do capítulo.

### 4.1 Domínio

Para início da definição do domínio, é necessário fechar o escopo de desenvolvimento, descrevendo de forma clara qual o nicho de jogos será adotado para o domínio. Sabendo que existem dois tipos principais de jogos analógicos, de cartas e de tabuleiro, tomou-se como base o catálogo com os tipos de mecânicas de jogos de tabuleiro (KRITZ EDUARDO MANGELI, 2017), para entender, a nível de *gameplay*, quais as diferenças entre eles. A partir daí, notou-se que existem muitas variações de mecânicas que dependem exclusivamente do tabuleiro, como *tile placement*, e várias outras que independem dele, como *take that* ou *gambling*.

A partir desta constatação, nota-se que o domínio de jogos de tabuleiro no geral engloba o domínio de jogos de cartas, dado que um jogo de tabuleiro pode ter cartas no *gameplay*. Com isso em mente, adotou-se o domínio de jogos de cartas analógicas como escopo principal, por questões de simplificação. Contudo, levou-se em consideração que em trabalhos futuros pode-se expandir o escopo para jogos de tabuleiro no geral a partir deste trabalho. A escolha por simplificar o escopo é vantajosa, dado que este trabalho se insere em um campo pouco explorado, no desenvolvimento de jogos analógicos baseados em MDE.

O *gameplay* dos jogos gerados tem como mecânica básica o Quiz - perguntas e respostas descritas pelo professor. No entanto, outras mecânicas principais e secundárias podem fazer parte do repertório, como *Trade*, *Take That*, *Card Drafting*, *RPG*, *Gambling*, (KRITZ EDUARDO MANGELI, 2017), dentre outros, uma vez que o *D-CreEA* suporta descrição livre de regras e extensão do *gameloop*.

Com isso, a partir dos resultados da busca detalhada no Capítulo 3 e do fechamento do escopo, levantou-se que o domínio da *DSML* possui as seguintes entidades:

- Jogo (*Game*): representa a entidade de um jogo, encapsulando todas as suas componentes

- e detalhes, como descrição, *decks*, estados, funções, etc.
- Estado (*State*): é o estado momentâneo do jogo, com suas respectivas regras. O jogo deve ter pelo menos um caminho do estado inicial ao estado de final de jogo.
  - Regra (*Rule*): As regras são os reguladores do jogo definidos pelo professor e possuem dois tipos:
    - Regra de declaração (*Statement Rule*): explica uma situação de jogo, no sentido de 'como fazer'. Uma *Statement Rule* tem sua descrição simples e outra complexa, que é opcional. Esta última segue uma estrutura formal: eu (o ator que participa da ação), dado que (a condição de jogo necessária), quando (um fato ocorrer), então (fazer algo). Este tipo de descrição ajuda a guiar o usuário que não tem muita intimidade na descrição de regras.
    - Regra Condicional (*Conditional Rule*): Esta regra é formada por um conjunto de condições. Se todas falharem, deve existir um estado padrão para ser seguido. Esse tipo de regra é indicado para uma situação de jogo em que existem  $N = \{A, B, \dots, K\}$  estados que podem ser alcançados a partir do estado atual. A decisão de avançar para um estado X depende do cumprimento de uma condição Cx, gerando um Efeito Ex, sendo possível determinar qual condição será verdadeira somente durante a execução do jogo.
  - Baralho (*Deck*): representa o conjunto de cartas de um determinado tipo definido pelo professor. Cada baralho tem um propósito (por exemplo, desafios, bônus) e deve ser diferente de todos os outros baralhos para evitar redundância. O *deck* tem um leiaute frontal (*Front Layout*) e, possivelmente, um leiaute posterior (*Back Layout*), sendo possível escolher quais campos devem aparecer na carta.
    - Leiaute frontal (*Front Layout*) define todos os campos que uma carta deste *deck* possui, como título, arte, descrição, efeito e tipos de pontos, como custo, nível e ganho.
    - O Leiaute posterior (*Back Layout*) é semelhante ao leiaute frontal, mas tem um título, respostas, efeito e tipos de pontos, como custo, nível e ganho.
  - Carta (*Card*): unidade de informação que representa um cartão de jogo genérico. Segue o padrão de seu respectivo *deck*.
  - Papel (*Role*): representação da persona adotada por um jogador, que garante efeitos positivos ou negativos.

- *Token*: representa unidades palpáveis que possuem alguns efeitos positivos ou negativos no jogo.
- Efeito (*Effect*): é um ato que altera o status de um ator durante o jogo. Um jogador inicia um efeito, e a ação pode ser para o próprio jogador ou para o outro e pode durar alguns turnos ou o resto do jogo.
- Habilidade (*Skill*) é um efeito de um ator que tem um efeito bom e, opcionalmente, um efeito ruim.
- Declaração (*Statement*) é uma descrição usada para explicar alguma regra. É usado no contexto da *Statement Rule*.
- Condição (*Condition*) é usada para aplicar algum efeito e alcançar algum estado se um teste de expressão for verdadeiro. É usado no contexto da Regra Condicional.

## 4.2 Metamodelo

O metamodelo tem como função principal gerar modelos que residem dentro do domínio do trabalho. Para isso, são definidas a sintaxe abstrata e a semântica estática da linguagem de criação de jogos. A sintaxe abstrata define os papéis dos elementos do domínio e como eles devem se relacionar entre si e com sistemas externos.

Seguindo a metodologia descrita na Seção 1.5, utilizou-se de duas abordagens distintas: top-down e bottom up.

Na ótica *Top-down*, estruturas de generalização de jogos são usadas para fomentar a base da arquitetura da DSML. Delas foram retirados o entendimento sobre esse tipo de estruturas, seus pontos fortes e suas limitações, a fim de comparar com a D-CreEA e entender o posicionamento do presente trabalho na comunidade. As estruturas com maior destaque foram o *framework* MDA (Mechanics, Dynamics, and Aesthetics) e as estruturas de (BATTAIOLA; MUNHOZ, 2018), (O'SHEA; FREEMAN, 2019) e (PRENSKY, 2001).

Por outro lado, usando a abordagem *bottom up*, buscou-se jogos analógicos educacionais em anais de eventos de jogos (por exemplo, SBGames) e Google Scholar de forma não sistematizada para fazerem parte da base de jogos que contribuíram para o domínio da DSML. Os jogos foram filtrados por título e resumo para que ficasse claro que se tratavam de jogos dentro do escopo. Os jogos foram classificados em termos de tipo de jogo (cartas, tabuleiro ou híbrido), número de jogadores e seus tipos de mecânica, seguindo o catálogo *Board Game Geek* (BGG). O resultado é mostrado na Tabela 3.

Quadro 3 – Jogos base para o domínio

Trabalho	Tipo de Jogo	Tipo de mecânica	# jogadores
GREaTest Card Game (BEPPE <i>et al.</i> , 2018)	Cartas/Híbrido	Trading, dice rolling, deck, take that, card drafting, RPG	2 a 7
Heredograma Sem Mistério (CAMPOS <i>et al.</i> , 2003a)	Tabuleiro	Tile placement, dice rolling, RPG	5 equipes ou 5 pessoas
Evolução: A Luta Pela Sobrevivência (CAMPOS <i>et al.</i> , 2003b)	Tabuleiro	Moviment/point-to-point, RPG	5 equipes ou 5 pessoas
ARBattle (MALHEIROS <i>et al.</i> , 2012)	Cartas	Trading, Take that, Deck, RPG	Não descrito
Computasséia (SANTOS; FIGUEIREDO, 2016)	Cartas	Trading, Cooperativo, Tile placement, Story-telling	2 a 6
SimulES-W (MONSALVE <i>et al.</i> , 2010)	Tabuleiro e Cartas	Take That, Deck, Set Collection, RPG	Não descrito
O Baralho das Variáveis (FRANÇA <i>et al.</i> , )	Cartas	Deck, Tile Placement	1
Desafio de Design do Google (DARIN <i>et al.</i> , 2019)	Tabuleiro e Cartas	Deck, Take That, Random Draw, RPG	Não descrito
JEEES (FIGUEIREDO <i>et al.</i> , 2010)	Tabuleiro e Cartas	Deck, Set Collection, RPG	2 a 3
Dengueside Survival (AZEVEDO <i>et al.</i> , 2020)	Tabuleiro e Cartas	Deck, Movement, Random Draw, RPG	1 a 6
Gente (GUERREIRO <i>et al.</i> , 2020)	Cartas	Deck, Trading, RPG	2 a 8
O Jogo do Método (MATTAR <i>et al.</i> , 2017)	Tabuleiro e Cartas	Area movement, Cooperativo, Collection, Dice Rolling, RPG	Não descrito

Fonte: elaborado pelo autor.

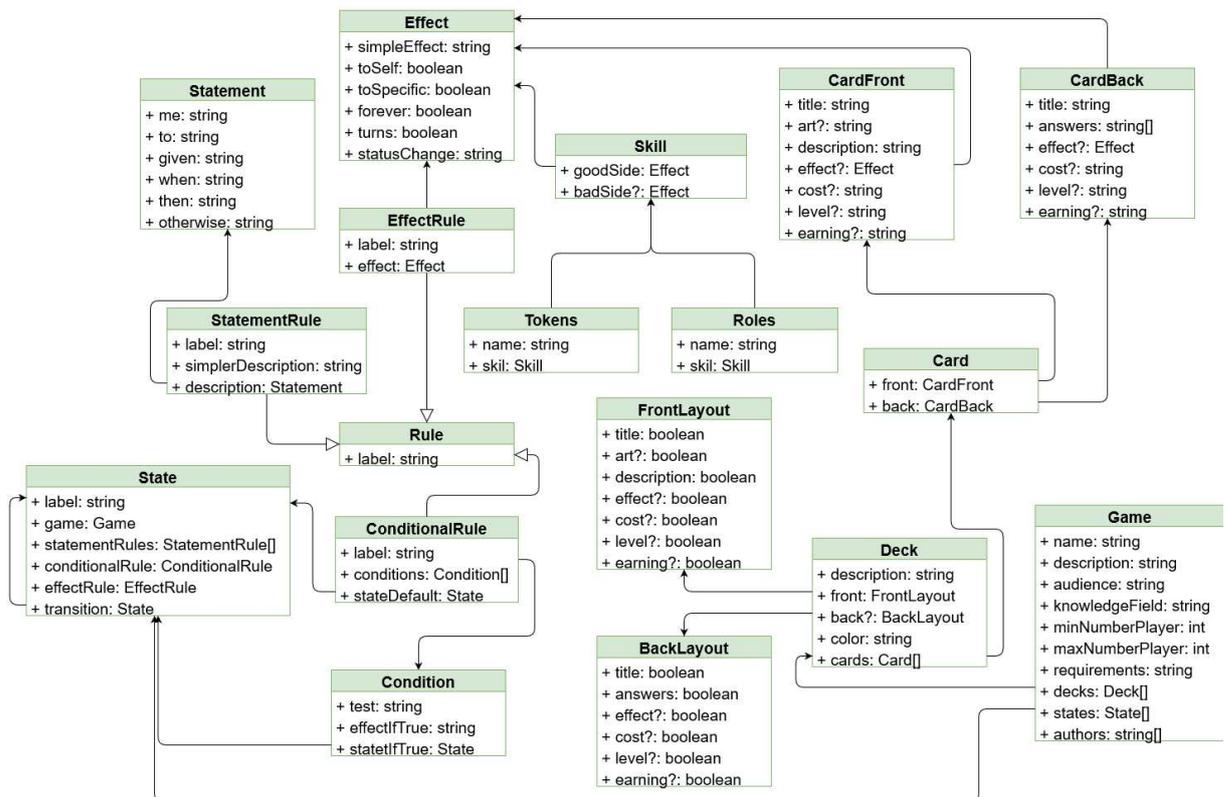
Após a análise dos jogos analógicos de tabuleiro encontrados, foram levantados os seguintes pontos:

- Os jogos abordam uma ampla variedade de temas educacionais, dentro e fora da computação. Na Computação, os temas foram: Desenvolvimento de Sistemas, Design e IHC, Programação, Software (e.g., Engenharia de Configuração, Teste de Software) e história da Computação. Já fora da Computação, os temas foram: Enfrentamento do Aedes Aegypti, Biologia, diversidade e método científico.
- Os jogos de tabuleiro situam o jogador em simulações dentro do tema educacional proposto através de mecânicas de RPG. Isso vai ao encontro da discussão apresentada no Capítulo 2, em que um dos requisitos para se alcançar objetivos educacionais é situando o aluno em situações palpáveis dentro do contexto educacional.
- Há muitos jogos de tabuleiro que usam como mecânica central as cartas, sendo possível converter alguns para jogos puramente de cartas, sem a necessidade de um tabuleiro como mecânica ativa. Assim, estes trabalhos foram úteis para o o escopo deste trabalho.
- Em relação aos jogos de cartas, a característica em comum é que há desafios envolvendo o conteúdo educacional que devem ser resolvidos através de uma mecânica simulada pelas cartas e seus efeitos.
- Geralmente, os jogos se baseiam em turnos ou rodadas para cada jogador, mas como

sugere a mecânica de *Simultaneous Action Selection*, optar por ações simultâneas entre os jogadores não é incomum.

Com isso, formou-se a base para a sintaxe abstrata. Para representá-la, utilizou-se UML, mais especificamente o diagrama de classes, pois este tem a função de representar de forma gráfica as relações entre entidades de um domínio. A sintaxe abstrata da D-CreEA é mostrada na Figura 5.

Figura 5 – Sintaxe Abstrata - *D-CreEA*



Fonte: Autor

A semântica estática da linguagem tem a função de restringir as relações da sintaxe abstrata, para que esta possa instanciar modelos de forma precisa e detalhada. Para o caso da *D-CreEA*, a semântica estática é um conjunto de regras que governam todos os jogos de cartas analógicas para este metamodelo e serve para representar relações que não podem ser (facilmente) mostradas por meio de diagramas de classes. As regras de definição de modelos é apresentada a seguir:

1. Cada jogo deve ter um único estado inicial e um único fim de jogo;
2. O estado inicial deve ter apenas uma transição, o estado *Game Over* não deve ter nenhuma transição, e nenhum estado pode transitar para o estado inicial;
3. Cada jogo deve ter pelo menos um caminho do estado inicial para o estado de final de jogo

- definido como *caminho do jogo* - com pelo menos um estado no meio; e
- 4. Cada caminho de jogo deve ter como objetivo implementar uma nova mecânica/tipo de jogo.

### 4.3 Modelo-para-Artefato

O Modelo-para-Artefato, ou Model-to-Artifact (M2A), é um conjunto de algoritmos cuja entrada é um modelo de um certo domínio, e a saída é uma implementação daquele modelo, que resolve um problema relacionado. No caso do *D-CreEA*, o modelo para artefato recebe um modelo em *JavaScript Object Notation (JSON)*, gerado via o metamodelo descrito anteriormente, e gera os artefatos de jogos analógicos educacionais: cartas e o manual do jogo. Um exemplo é mostrado na Seção 4.5.

Em muitos casos, é necessário ter um modelo intermediário, para facilitar a transformação em artefatos. No caso do *D-CreEA*, o modelo intermediário utilizado é o *JSON*, pois pode ser traduzido para Javascript e esta linguagem pode ser usada em qualquer plataforma, como *desktop*, *WEB* ou *mobile*.

Para gerar os artefatos de jogo, uma interface WEB foi implementada, e nela reside o algoritmo M2A, implementado como um serviço. Assim, este conjunto de algoritmos é *declarativo*, pois as saídas só dependem dos parâmetros de entrada. Isso é positivo, pois, por não depender de contexto externo, pode ser portado para outras implementações, independentemente de qual linguagem e ambiente.

O M2A tem quatro componentes principais: *Regras-para-texto (R2T)*, *Descrição-para-texto (D2T)*, *Baralhos-para-texto (B2T)* e *Cartas-para-artefato (C2A)*.

O algoritmo R2T começa no estado Game Start e percorre todos os caminhos do game loop até que todas as regras sejam visitadas exatamente uma vez, sendo traduzidas em texto formatado. Aqui, o percurso é feito em uma espécie de busca em largura, onde para cada estado, suas regras são apresentadas antes das regras do próximo estado. O passo é dado via regras condicionais ou regra de transição (ver Seção 4.1).

O D2T traduz os detalhes do jogo, tais como nome e descrição, em texto em linguagem natural.

No B2T, os detalhes de cada baralho são transformados em texto, destacando os significados dos campos em uma descrição de texto em linguagem natural.

O C2A traduz todas as cartas de todos os baralhos em cartas formatadas, seguindo o

padrão dos campos que aparecem em cada baralho, e com o *aspect ratio* de uma carta de Tarô, 7 x 12. Este formato foi adotado devido ao formato 'alongado', que pode caber mais conteúdo.

Os artefatos resultantes são apresentados em páginas HTML, customizadas com CSS, proporcionando uma boa procura de cartões, *tokens* e manuais. Além disso, uma página HTML pode ser impressa por qualquer navegador da web. Como o código de implementação é extenso, fornecemos um repositório de código online para salvá-lo e disponibilizá-lo para consulta<sup>1</sup>.

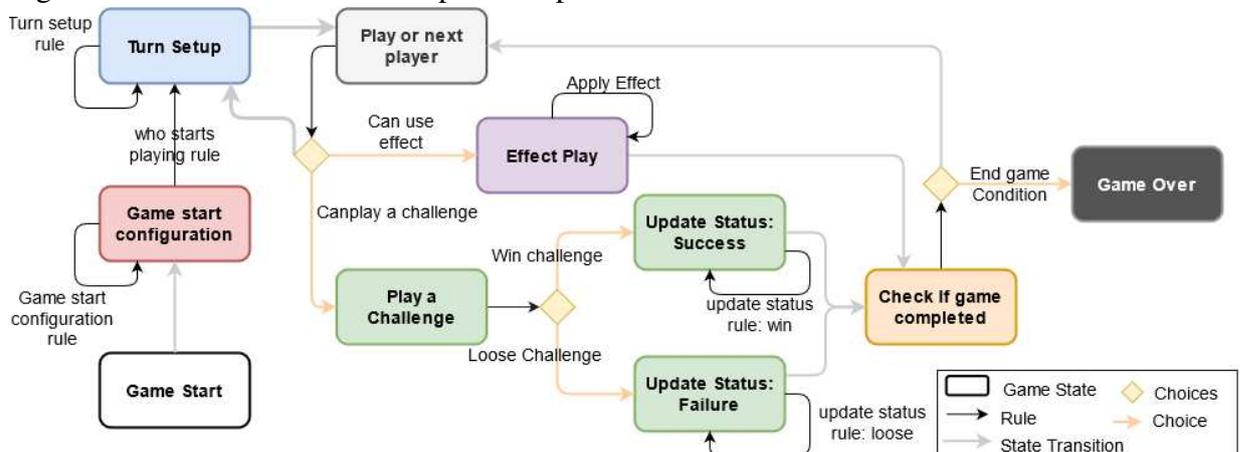
#### 4.4 Modelo Guia

Para guiar o desenvolvimento, é fornecido um modelo de jogo que implementa os elementos do domínio, as relações da sintaxe e as restrições da semântica. O modelo é definido como um ciclo de jogo que representa a configuração padrão de um jogo de cartas analógico educacional que possui as mecânicas do Quiz e do Bônus. No modelo, cada caminho do estado *Play or Next Player* a *Verificar se o jogo está completo* representa uma mecânica de jogo diferente.

A representação deste modelo é baseado em um diagrama de atividades *UML*, onde cada retângulo representa um estado de jogo do domínio e cada seta representa uma regra de transição. Este modelo tem o papel de exemplificar o uso da *DSML* para o usuário, facilitar o entendimento da dinâmica de instanciação dos modelos e serve de base para a implementação do modelo de jogo, através da extensão pelo usuário.

O modelo do *D-CreEA* pode ser visto na Figura 6.

Figura 6 – Modelo do Game Loop - Exemplo do *D-CreEA*



Fonte: Autor

<sup>1</sup> model-to-text repository: <https://bit.ly/3x1qXa1>

#### 4.5 Validação da adequação da solução: Instância de um Jogo

Para validar a adequação da solução, foi proposto re-criar o jogo GreaTest Card Game (*GTCG*) (BEPPE *et al.*, 2018), um jogo já existente e já publicado, através do método da *D-CreEA*. Esta avaliação visa responder à seguinte questão: “A linguagem *D-CreEA* é capaz de gerar jogos analógicos?”

O jogo *GTCG* é um jogo de cartas educacional multi-jogador voltado à prática de teste de software. O jogo é indicado para partidas com 3 a 7 jogadores de forma presencial, e possui as mecânicas de Quiz, onde o jogador escolhe uma carta de jogo com o tipo de teste de software que mais se adequa ao cenário de uso de software proposto em uma carta desafio posta sobre a mesa. As respostas ficam no verso da carta, cabendo a um dado decidir se o jogador que acertou a resposta irá ganhar os pontos da pergunta. Além disso, há uma mecânica de negociação, onde os jogadores podem conversar entre si para trocar cartas de acordo com as regras definidas na hora. Por último, existe a mecânica de carta bônus, com cartas que possuem efeitos para alterar os status dos jogadores e da mesa.

Esse jogo foi escolhido por estar dentro do domínio da *D-CreEA* e por ter mecânicas que não estão presentes originalmente no exemplo de jogo apresentado na Seção 4.4, como a negociação. Com esta avaliação, é possível detectar os pontos fortes e os pontos de melhoria do *D-CreEA*. Uma vez que a implementação completa do *GTCG* seja possível, há evidências sobre a adequação da *DSML*. Além disso, devido ao jogo base ter diferentes tipos de mecânica de jogo, como negociação e um loop de jogo diferente, é possível inferir evidência da extensibilidade do *DSML*.

O primeiro passo para desenvolver esse jogo via *D-CreEA* foi entender bem as regras do jogo e, dessa forma, tentar reproduzir o fluxo de gameplay do GreaTest no diagrama de game loop. O resultado disso é mostrado na Figura 7. Cada mecânica é representada por um caminho entre a condicional de *Play or Go To Next Player* e o estado que escolhe qual tipo de jogada irá ser adotada nesse momento da partida. Ele tem três saídas: para *Bonus Play*, para *Play a Challenge* e para *Negotiate with players*. Cada caminho segue até *Game Over* ou ao próprio estado *Play or Go To Next Player*, onde o ciclo é retomado.

O passo seguinte é modelar o *JSON* para implementar o gameloop do GreaTest e as informações do jogo original, como nome, descrição, etc. Com o jogo codificado, foi possível testar o código do modelo como *input* para o M2A. Como resultado, foram gerados três baralhos com as cores originais do jogo, os campos que aparecem em cada tipo de carta e algumas imagens

Figura 7 – Game Loop - GREaTest Card Game



Fonte: Autor

template pré-implementadas na plataforma *D-CreEA* (Figura 10), exemplos de regras geradas a partir do diagrama (Figura 8) e as informações do jogo organizadas em quadrados (Figura 9).

\* 'Play a challenge': Describes the way to resolve a challenge.

- The player combines 1 Game Card with 1 Challenge Card to make it's play
- the player beats the challenge if S/He uses a game card that answers correctly the challenge card that s/he chose from table
- the player throw the dice if S/He beats the challenge
- If the player beats the challenge AND the dice throw shows a number present in the answers, Then S/he wins the challenge. With this, go to 'Update status: success'
- If the player beats the challenge but the dice throw shows a number NOT present in the answers, Then S/he wins the challenge. With this, go to 'Play or next player'
- If the player used a game card that DO NOT answers correctly the challenge, Then S/he looses the challenge. With this, go to 'Update status: failure'

Fonte: Autor

Figura 8 – Exemplo de regra do GreaTest Card Game gerado pelo D-CreEA

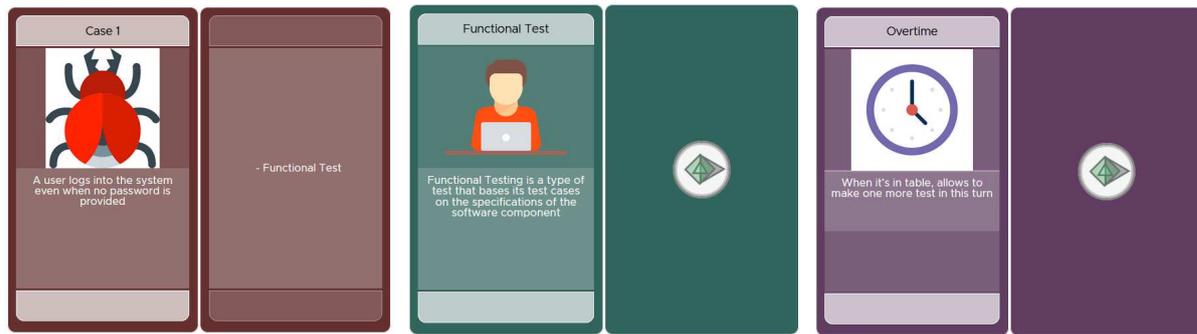
<p><b>GreaTest Card Game</b></p> <p>A game to practice software testing. In this game, the players must resolve challenges by combining types of software testing with software application scenarios. The objective is to choose the type of test that best suits the software scenario</p>	<p><b>Game Details</b></p> <ul style="list-style-type: none"> <li>• target audience: computer science students</li> <li>• knowledgeField: Software Testing</li> <li>• requirements: Software testing types</li> <li>• authors: Rubens Silva</li> <li>• NumberPlayers: from 3 up to 7</li> </ul>	<p><b>Game Material</b></p> <p>Decks</p> <ul style="list-style-type: none"> <li>• Challenges Deck: This deck has all the challenges of this game. Each challenge is a software use description. This deck has 2 cards</li> <li>• Game Deck: This deck has the types of software testing, e.g., Unit Test, Usability Test. This deck has 2 cards</li> <li>• Bonus Deck: This deck has all bonus cards. Every bonus card has an effect. This deck has 2 cards</li> </ul>	<p><b>Game Basics</b></p> <ul style="list-style-type: none"> <li>• This is a turn-based educational card game</li> <li>• We have 3 decks, and each one has a purpose</li> <li>• In its turn, the player must solve challenges to get victory points</li> <li>• Wins the game the player who has more victory points</li> </ul>
--	---	--	--

(a) Detalhes

(b) Materiais de Jogo

Figura 9 – As informações do jogo geradas por the *D-CreEA*

Como conclusão desta validação, foi possível gerar um jogo analógico publicado



(a) Carta Desafio

(b) Carta de Jogo

(c) Carta Bônus

Figura 10 – Cartas do GreaTest Card Game geradas pelo *D-CreEA*

bem semelhante ao original, diferindo apenas em detalhes de design das cartas, com as mecânicas se mantendo fieis às originais. Observa-se então que há aderência da linguagem *D-CreEA* ao domínio de jogos analógicos de cartas. Com isso, há respaldo para a continuação da implementação da *DSML* para uma plataforma que facilite a manipulação e visualização das regras de criação de jogos via *D-CreEA*.

#### 4.6 Resumo do Capítulo

Este capítulo apresentou a *DSML* para criação de jogos analógicos de cartas. Foram apresentados o domínio detalhado, o metamodelo, com sua sintaxe e semântica, a transformação de modelo para artefato e, por último, um modelo guia para auxiliar no desenvolvimento dos modelos de jogos pelo professor/usuário final.

O domínio definiu que o escopo do projeto é o de jogos de cartas, que independem de tabuleiro como mecânica principal. Além disso, definiu-se todas as entidades presentes no domínio, como Estado de Jogo, Tipos de Regras, Baralhos, etc.

O metamodelo baseado no domínio apresentou a sintaxe abstrata da linguagem de criação de jogos, que define como as entidades do domínio se relacionam e foi representada através de um diagrama de classes UML. Já a semântica estática serve para definir as restrições de criação de jogos para o domínio, que para o caso deste trabalho, serve também para definir relações que não podem ser inferidas facilmente do diagrama de classes. Um exemplo de restrição foi “Todo jogo deve ter exatamente um estado chamado *Game Start* e um chamado *Game Over*”.

Já o modelo para artefato definiu como é realizada a transformação do modelo gerado a partir do metamodelo para artefatos de jogo (M2A), que podem ser impressos. O M2A primeiro

transforma os dados de *input* para um modelo intermediário em *JSON* e, em seguida, transforma esse código em elementos *HTML* que podem ser impressos via navegador.

Por último, foi apresentado o modelo guia para auxiliar a implementação do modelo de jogo pelo usuário final. Este modelo implementa duas mecânicas básicas de Quiz e Jogada Bônus, além de respeitar as restrições da semântica estática do metamodelo. Assim, o usuário não precisa implementar o modelo do zero, basta estender e modificar o modelo guia para gerar um resultado personalizado.

## 5 A PLATAFORMA D-CREEA CARD GAMES

Este capítulo apresenta a plataforma Web *D-CreEA* Card Games. A Seção 5.1 apresenta o processo de engenharia de requisitos da plataforma, com as restrições, decisões de desenvolvimento e requisitos da *D-CreEA*. Depois, na Seção 5.2, a arquitetura da aplicação é mostrada. Na Seção 5.3, os detalhes da implementação e funcionalidades de cada tela são destacados e a Seção 5.4 mostra os resultados da implementação. Por último, a Seção 5.5 traz um resumo do capítulo.

### 5.1 Requisitos da D-CreEA

Nesta seção são apresentadas as fases da engenharia de requisitos da aplicação *D-CreEA*: estudo de viabilidade, elicitação e análise de requisitos e especificação de requisitos.

É importante destacar que, neste trabalho, sempre que for citada *D-CreEA* sem especificar se é falado sobre a *DSML* ou a plataforma, assume-se que o contexto se refere à plataforma.

#### 5.1.1 Estudo de Viabilidade

No estudo de viabilidade, segundo (SOMMERVILLE, 2015), deve ser feita uma estimativa acerca da possibilidade de se satisfazerem as necessidades do usuário, usando-se tecnologias atuais de software e hardware. O estudo deve considerar se o sistema proposto será proveitoso a partir de um ponto de vista de negócio e se ele pode ser desenvolvido no âmbito das atuais restrições.

O objetivo principal da plataforma *D-CreEA* é prover uma interface para a linguagem *D-CreEA*, a fim de auxiliar a criação de jogos analógicos de cartas via *DSML*. A plataforma é necessária para facilitar o uso da *DSML*, como mostrado no Capítulo 4, intermediando a interação do usuário da *DSML* e evitando que os usuários finais (professores de Computação) precisem manipular código *JSON*<sup>1</sup> diretamente.

Para o desenvolvimento da plataforma, foi adotado o modelo de processo incremental por prototipagem (SOMMERVILLE, 2015), pois esse modelo se adequa a um desenvolvimento de um sistema não-crítico, cujos requisitos podem evoluir a cada iteração de desenvolvimento.

Sobre a viabilidade técnica, adotou-se tecnologias que possuem vasta gama de

---

<sup>1</sup> JSON: <https://www.json.org>

bibliotecas *open source* para auxiliar no desenvolvimento e evitar codificação de muito baixo nível, proporcionando um rápido desenvolvimento. Em relação a plataforma de desenvolvimento, viu-se que gerar um sistema para a WEB é viável, pois não há restrições específicas de hardware que demandem que o *D-CreEA* seja instalado no computador. Além disso, o autor principal deste trabalho, que é responsável pela codificação e desenvolvimento, possui experiência com aplicações para Web e Mobile, o que contribui para a viabilidade do projeto.

### 5.1.2 *Elicitação e Análise de Requisitos*

Segundo (SOMMERVILLE, 2015), elicitación e análise de requisitos é o processo de derivação dos requisitos do sistema por meio da observação dos sistemas existentes, além de discussões com os potenciais usuários e compradores, análise de tarefas, entre outras etapas. Segundo (PRESSMAN; MAXIM, 2021), elicitación consiste nas atividades relacionadas com a descoberta e entendimento dos requisitos de um sistema. Essa parte do processo pode envolver o desenvolvimento de um ou mais modelos de sistemas e protótipos, os quais nos ajudam a entender o sistema a ser especificado. Os requisitos devem ser examinados para que sejam encontradas inconsistências, ambiguidades e omissões.

Desta forma, desenvolveu-se um protótipo para ilustrar as possíveis funcionalidades que o sistema precisa suprir. O protótipo, feito na ferramenta Adobe XD<sup>2</sup>, apresenta uma possível interface com o usuário em formato WEB para computador, e pode ser visto no Apêndice A. Com esse protótipo, entendeu-se alguns dos aspectos fundamentais para o usuário em termos de funcionalidade. O resumo de cada uma das funcionalidades é mostrado a seguir:

- Login: é necessário realizar login no sistema para proteção dos dados de cada jogo.
- Game Design Canvas: provê um game design canvas para definir as propriedades do jogo, como nome, autores, número de jogadores, etc.
- Diagrama: dispõe de uma interface de visualização do diagrama de *gameloop* do jogo, onde são exibidas as relações entre Estado de Jogo e Regras de Transição.
- Gameplay: é responsável pela interface de *Create, Read, Update, Delete (CRUD)* dos estados do jogo. Também deve prover o CRUD de regras.
- Deck: fornece o CRUD de *decks*.
- Cartas: provê o CRUD de cartas de cada *deck*.
- Resumo & Impressão: apresenta o manual do jogo em tempo real, com o resultado do

<sup>2</sup> Adobe XD: <https://www.adobe.com/products/xd.html>

algoritmo de modelo para artefato, explicação dos *deck* e modelos para impressão das cartas.

- Ajuda: fornece uma página de ajuda para guiar o desenvolvimento.

### 5.1.3 Especificação de Requisitos

Segundo (SOMMERVILLE, 2015), especificação de requisitos é a atividade de traduzir as informações obtidas durante a atividade de análise em um documento que defina um conjunto de requisitos. O objetivo é garantir que eles estejam corretos, precisos, completos, consistentes e verificáveis (PRESSMAN; MAXIM, 2021). Dois níveis de detalhamento de requisitos podem ser incluídos neste documento: *Requisitos do Usuário (RUs)*, que são declarações abstratas dos requisitos do sistema para o cliente e usuário final do sistema; e *Requisitos do Sistema (RSs)*, que são uma descrição mais detalhada da funcionalidade a ser provida.

Os tipos de requisitos do sistema, por sua vez, se dividem em *Requisitos Funcionais (RFs)*, que tratam cada requisito do sistema tal qual uma função, com entradas e saídas bem definidas, e *Requisitos Não-Funcionais (RNFs)*, que tratam de requisitos macros, relacionados à qualidade do sistema ou de componentes, através de objetivos e métricas para aferição de tais objetivos.

#### 5.1.3.1 Requisitos Funcionais do Usuário

Seguindo tanto as restrições definidas na elicitação de requisitos quanto o protótipo, chegou-se ao seguinte conjunto de Requisitos Funcionais do Usuário:

- RU1** Login - O usuário deve ser capaz de realizar login com e-mail e senha no sistema.
- RU2** Criar, Editar, Deletar e Ver Jogos - O usuário deve ser capaz de criar e modificar jogos dentro do sistema.
- RU3** Todo jogo deve seguir o padrão da linguagem *D-CreEA* (ver Capítulo 4).
- RU4** Todo jogo deve ter um Game Design Canvas (GDC) (SARINHO, 2017) para descrever as propriedades de um jogo.
- RU5** o usuário deve ser capaz de alterar o Game Loop do seu jogo ao alterar os estados, as regras e transições, sob as restrições da semântica estática (ver Seção 4.2).
- RU6** Em todo jogo, deve ser possível criar, editar e remover *decks* do jogo.
- RU7** Em todo jogo, deve ser possível criar, editar e remover cartas de um de seus *decks*.
- RU8** Em todo jogo, deve ser possível imprimir o conteúdo do jogo, como manual de jogo, *decks*,

cartas e regras.

**RU9** Para guiar a criação de jogos, deve ser provida uma página de ajuda contendo informação útil sobre a ferramenta.

### 5.1.3.2 *Requisitos Funcionais do Sistema*

Com os requisitos do usuário em mãos servindo de base para as funcionalidades, definiu-se o conjunto de requisitos funcionais do sistema a seguir:

**RF1** Login: O usuário deve ser capaz de criar um usuário e realizar login no sistema com e-mail e senha únicos.

**RF1-a** O e-mail de cada usuário deve ser único, e a tentativa de criar um usuário com nome repetido, deve ser impedido.

**RF2** CRUD de jogos: o usuário deve ser capaz de criar, alterar, remover e ver os jogos que criou;

**RF2-a** O nome de cada jogo deve ser único, e a tentativa de criar um jogo com nome repetido, deve ser impedido.

**RF2-b** Todo jogo criado deve ser o padrão da linguagem *D-CreEA*, seguindo as restrições da semântica estática (ver Seção 4.2).

**RF2-c** Todo jogo criado deve ter pré-implementado um exemplo de jogo, contendo um *game loop* básico de um jogo com mecânica Quiz.

**RF3** GDC: Todo jogo deve ter um *Game Design Canvas (GDC)*;

**RF3-a** O *GDC* deve possuir os seguintes campos: nome do jogo, autores, gameplay simplificada, descrição do jogo, público-alvo, campo de conhecimento, conhecimentos necessários e número mínimo e máximo de jogadores.

**RF3-a** todos os campos de FR3a devem ser preenchidos.

**RF4** Game Loop: o usuário deve ser capaz de alterar o Game Loop do seu jogo;

**RF4-a** o game loop possui estados, regras e transições, e o usuário deve ser capaz de criar, alterar, remover e ver todas elas.

**RF4-b** o estado possui *nome, propósito, regras e transição*.

**RF4-c** uma regra possui dois tipos *Regra Condicional e Regra Statement*. Uma regra condicional possui as condições de sucesso e uma condição de fracasso, caso todas as outras condições falhem - uma condição possui o teste, o efeito se verdadeiro e o estado se verdadeiro. Já uma regra *statement* possui uma descrição do que deve

acontecer quando esta for invocada.

**RF4-d** uma transição define qual o estado a seguir, quando terminar de aplicar as regras do estado atual.

**RF4-e** para visualizar estas relações de estados e transições, o usuário deve dispor de um diagrama semelhante ao da Figura 6, onde ele pode arrastar os estados para a posição que lhe convém dentro do canvas do diagrama.

**RF5** Decks: o usuário deve ser capaz de criar, editar e remover *decks* do jogo;

**RF5-a** cada *deck* possui *nome*, *descrição*, *cor primária* e *os campos da parte da frente e de trás de cada carta*

**RF5-b** os campos da parte da frente são: título, arte, descrição, efeito e pontuações. Já os campos da parte de trás são: título, respostas, efeito e pontuações.

**RF5-c** as pontuações são: custo - quanto se deve pagar para usar a carta; nível - nível do jogador no jogo atual necessário para usar esta carta; e ganhos - a recompensa por jogar com esta carta. Qualquer pontuação pode ser retirada.

**RF6** Cartas: o usuário deve ser capaz de criar, editar e remover cartas de um *deck*;

**RF6-a** cada carta possui um único *deck*, os campos da parte da frente e de trás de cada carta e o número de repetições naquele *deck*.

**RF6-b** os campos da parte da frente e de trás são os mesmos escolhidos no seu *deck*.

**RF6-c** cada carta define os valores de sua pontuação e de cada campo.

**RF7** Imprimir o conteúdo do jogo: o usuário deve ser capaz de imprimir o conteúdo do jogo.

**RF7-a** a impressão deve conter o manual do jogo, as cartas e qualquer *token* disponível implementado.

**RF7-b** o manual do jogo deve conter o GDC, a explicação dos *decks* e a explicação das regras em linguagem natural.

**RF7-c** as cartas devem ser impressas em um arquivo separado, para facilitar a impressão. Cada carta deve aparecer o mesmo número de vezes que indica o seu campo "repetições".

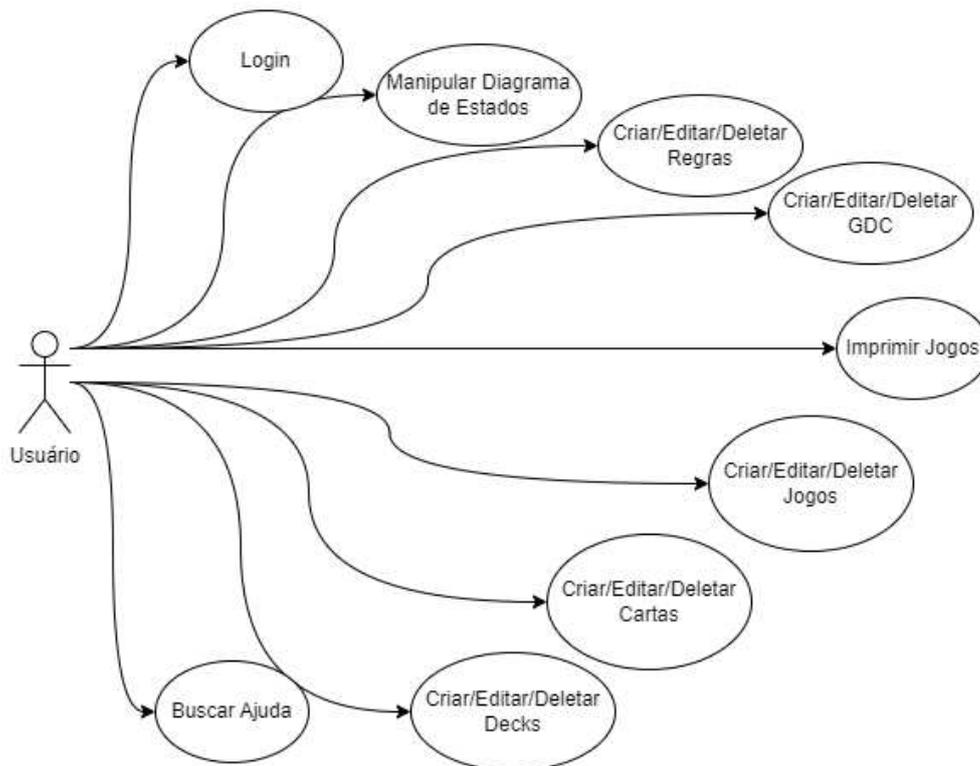
**RF7-d** a impressão deve ocorrer em arquivo PDF para que o usuário armazene os artefatos do seu jogo em disco e possa imprimi-lo independentemente do sistema, tendo assim mais controle.

**RF8** Tela de ajuda: o usuário deve ter disponível uma tela de ajuda, que o guie no desenvolvimento dos jogos;

**RF8-a** a tela de ajuda deve prover um tutorial, e exemplificar o uso da ferramenta com o exemplo de jogo implementado.

A seguir, na Figura 11, pode ser observado o diagrama de casos de uso da D-CreEA. Estão presentes oito funcionalidades principais, sendo elas: Login, Buscar Ajuda, Manipular Diagrama de Estados, Imprimir Jogo, Buscar Ajuda, e os CRUDs de Decks, Cartas, Jogos, GDC e Regras.

Figura 11 – Diagrama de Casos de Uso da D-CreEA



Fonte: Autor

### 5.1.3.3 Requisitos Não Funcionais

Segundo (SOMMERVILLE, 2015), requisitos não funcionais são requisitos que não estão diretamente relacionados com os serviços específicos oferecidos pelo sistema a seus usuários. Eles especificam ou restringem as características do sistema como um todo e são frequentemente mais críticos que requisitos funcionais individuais.

Dessa forma, seguindo as restrições definidas anteriormente, os requisitos não funcionais da *D-CreEA* são os seguintes:

**RNF1** Segurança: segundo (ISO/IEC/IEEE-24765, 2017), segurança é a preservação da confidencialidade, integridade e acessibilidade da informação. Dessa forma, o sistema deve

manter a segurança da informação de seus usuários em relação aos jogos e login.

**RNF2** Usabilidade: seguindo as heurísticas de (NIELSEN, 1994), o sistema deve informar de forma simples e direta, manter baixa a carga de memória do usuário, manter a consistência, o *feedback* constante, prevenir erros e oferecer ajuda quando tais erros acontecerem.

**RNF3** Velocidade: o sistema deve executar operações a partir da interação do usuário até sua resposta em menos de 4 segundos.

## 5.2 Projeto: Arquitetura e Diagrama de Classes

Um projeto de software é uma descrição da estrutura do software a ser implementado, dos modelos e estruturas de dados usados pelo sistema, das interfaces entre os componentes do sistema e, às vezes, dos algoritmos usados. (SOMMERVILLE, 2015).

O projeto de arquitetura está preocupado com a compreensão de como um sistema deve ser organizado e com a estrutura geral desse sistema. No modelo do processo de desenvolvimento de software, o projeto de arquitetura é o primeiro estágio no processo de projeto de software e é o elo crítico entre o projeto e a engenharia de requisitos, pois identifica os principais componentes estruturais de um sistema e os relacionamentos entre eles (SOMMERVILLE, 2015).

O resultado do processo de projeto de arquitetura é um modelo de arquitetura que descreve como o sistema está organizado em um conjunto de componentes de comunicação (SOMMERVILLE, 2015).

No caso da *D-CreEA*, foi utilizado o padrão de arquitetura baseado em *Model-View Controller (MVC)* (GOLDBERG; ROBSON, 1983) para WEB e em camadas (BUSCHMANN *et al.*, 2007). A *View* exibe a situação atual do modelo no *browser* do usuário, e troca eventos com o *Controller* via REST API.

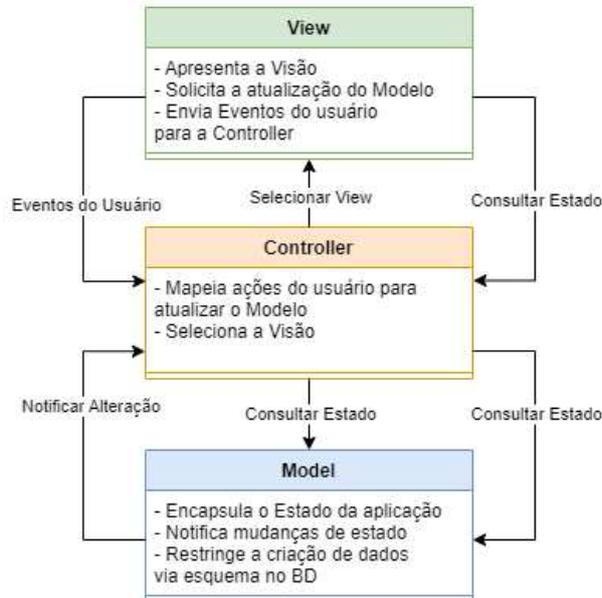
A camada *Controller* mapeia e valida as ações do usuário para retornar a *View* correta e para alterar o estado atual no *Model*. Sendo assim, a *Controller* troca eventos com a *View* e com a *Model*.

Por último, a camada *Model* é responsável por encapsular e armazenar o estado da aplicação, notificar mudanças no estado e restringir a criação dos dados via esquema. Esta, troca eventos com a camada *Controller*.

Há camadas, pois não há interação direta entre *view* e *model*, ao contrário do que sugere o *MVC* tradicional. Além disso, cada camada implementa seus mecanismos independentes

de segurança e autenticação, onde as operações realizadas são autorizadas caso o usuário esteja logado. A arquitetura é apresentada na Figura 12.

Figura 12 – Arquitetura da plataforma D-CreEA



Fonte: Autor

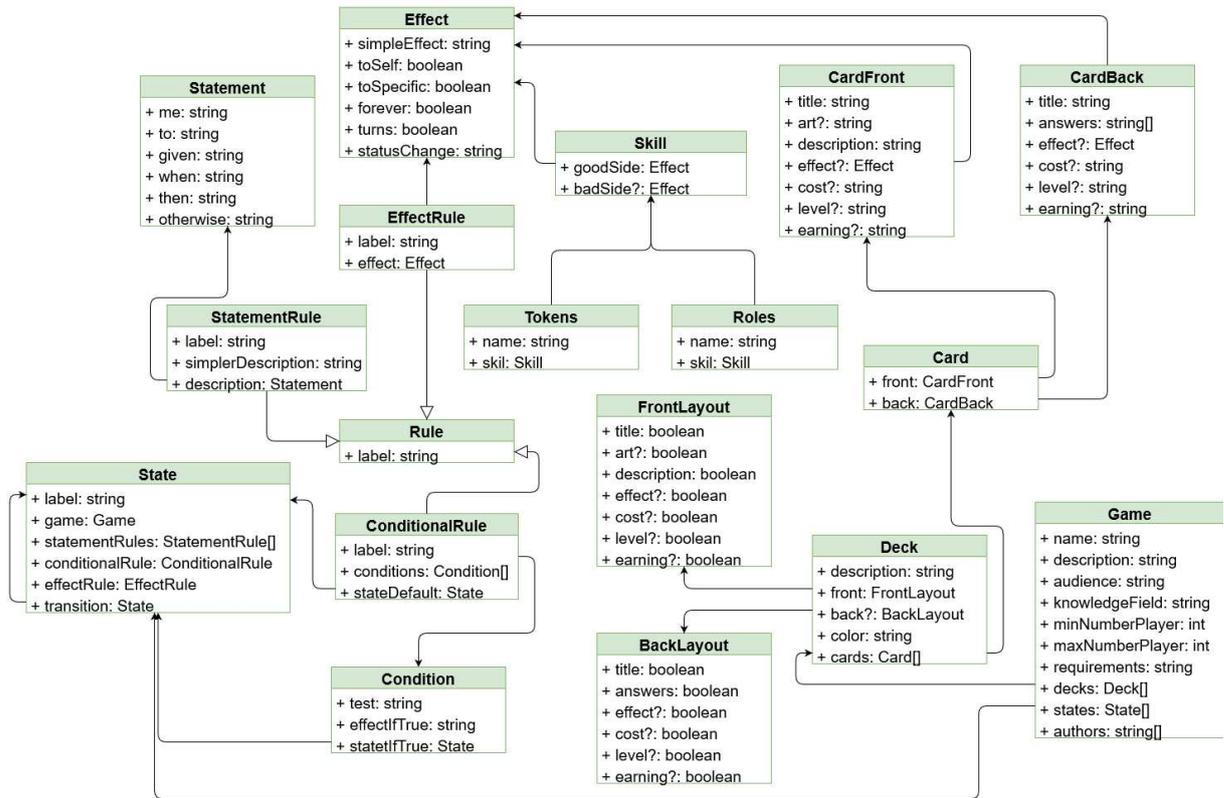
Já na Figura 13, pode ser observado o diagrama de classes da D-CreEA, idêntico ao que é visto na no Capítulo 4. Nesse diagrama, optou-se por ocultar os métodos.

### 5.3 Implementação

O estágio de implementação do desenvolvimento de software é o processo de conversão de uma especificação do sistema em um sistema executável. esta etapa sempre envolve processos de projeto e programação de software, mas, se for usada uma abordagem incremental para o desenvolvimento, também pode envolver o refinamento da especificação do software (SOMMERVILLE, 2015).

Para as tecnologias, escolheu-se *frameworks* JavaScript amplamente utilizados no mercado e com muitas opções de bibliotecas gratuitas disponíveis para o desenvolvimento<sup>3</sup>. Baseando-se nos requisitos do sistema, decidiu-se por um *System as a Service (SaaS)*, onde o usuário acessa o sistema via navegador para acessar o serviço de criação e impressão de jogos; seguindo a arquitetura do sistema, implementou-se um *MVC* com servidores para *front end*, *back end* e de banco de dados independentes. As tecnologias para o desenvolvimento do *D-CreEA* na plataforma WEB foram:

<sup>3</sup> <https://www.npmjs.com/>

Figura 13 – Diagrama de Classes - *D-CreEA*

Fonte: Autor

- na parte cliente (*view*) foi utilizado AngularJS<sup>4</sup>, projetado para construir interfaces com o usuário extensíveis;
- no *Controller* foi utilizado NodeJS<sup>5</sup>, um framework assíncrono baseado em eventos feito para aplicações WEB;
- no *Model*, optou-se por um gerenciador de bancos de dados disponível online, o MongoDB<sup>6</sup>, que possui sistema de *cloud* gratuito e de fácil uso.

Além disso, a plataforma possui sistema de entrega contínua (*Countinous Delivery*) através do serviço Heroku<sup>7</sup>, uma plataforma como serviço baseada em um sistema de container gerenciado para implantação e execução de aplicações na internet, que possui planos gratuitos.

Os componentes *View* e *Controller* do sistema foram hospedados no Heroku em dois servidores separados, enquanto o *Model* se encontra no serviço Atlas MongoDB. Todos os componentes atuam de forma independente e se comunicam via REST API. Cada plataforma possui suas políticas de segurança implementadas, cabendo ao usuário apenas estar autenticado para utilizar os serviços. Ainda, sobre os padrões de projeto usados no projeto, citamos: *builder*;

<sup>4</sup> AngularJS: <https://angularjs.org/>

<sup>5</sup> NodeJS: <https://nodejs.org/en/>

<sup>6</sup> Atlas MongoDB: <https://www.mongodb.com/>

<sup>7</sup> Heroku: <https://www.heroku.com>

*singleton, decorator, facade* (GAMMA *et al.*, 1995).

O processo de desenvolvimento seguiu por quatro meses, em constante revisão pelo autor e orientadora, e após esse período os primeiros resultados foram avaliados por professores que fazem parte do público-alvo da aplicação (ver Seção 1.3).

As telas da plataforma *D-CreEA* são apresentadas na Seção seguinte e o sistema está disponível no link <<https://dcreea.herokuapp.com/>>.

## 5.4 A plataforma D-CreEA

Esta seção apresenta as telas da plataforma D-CreEA, implementada neste trabalho, com foco nos detalhes de cada tela.

Cada subseção representa uma tela principal da aplicação, sendo elas: *Home, Login, GDC, Diagrama, Gameplay, Decks, Cards, Impressão e Ajuda*. A ordem das subseções não é por acaso, elas representam um fluxo de uso do usuário, que começa na tela *Home* e termina na tela de *Impressão*, cujas saídas são o manual do jogo e as cartas, acrescida da tela de *Ajuda*.

### 5.4.1 Home

A tela *Home* funciona como uma *landing page* para a plataforma. Aqui, vários detalhes das funcionalidades da aplicação são mostrados, a fim de despertar a curiosidade, criar interesse e provocar desejo no público-alvo, e assim gerar interesse em utilizar a ferramenta. A Figura 14 apresenta a *landing page*, separada em 4 setores principais.

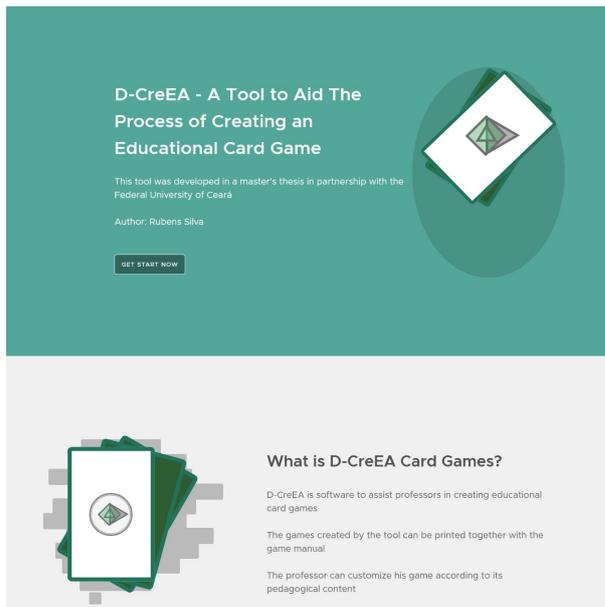
### 5.4.2 Login

A página de *Login* está contida dentro da *landing page* e é uma ponte entre a parte pública e restrita da aplicação. Aqui, o usuário pode criar um novo usuário com e-mail e senha ou se logar utilizando um usuário existente.

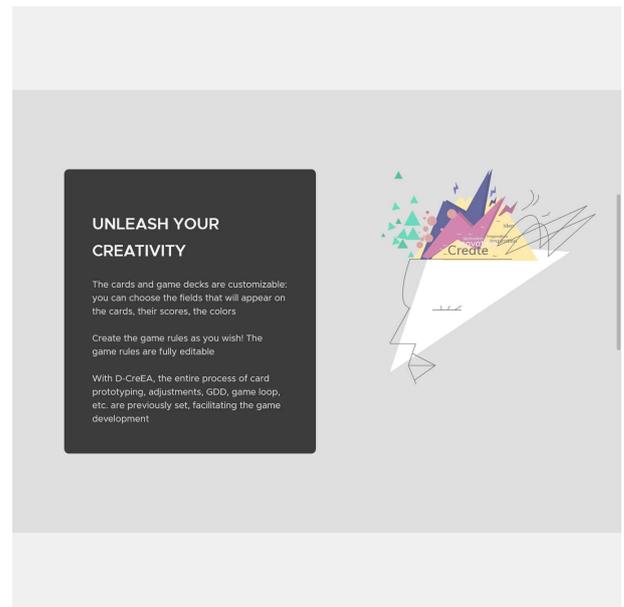
As restrições são: usuário deve ser único, então não pode existir duas contas para o mesmo e-mail, e ambos, e-mail e senha, não podem ser vazios.

### 5.4.3 Game Design Canvas

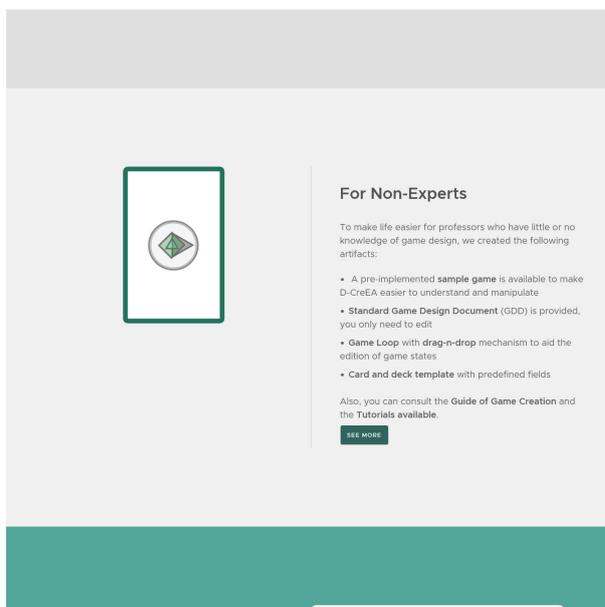
A partir dessa tela, todos os campos de formulário, cujo preenchimento possa despertar dúvida, possuem *tooltips* que aparecem ao lado do rótulo do campo e explicam com



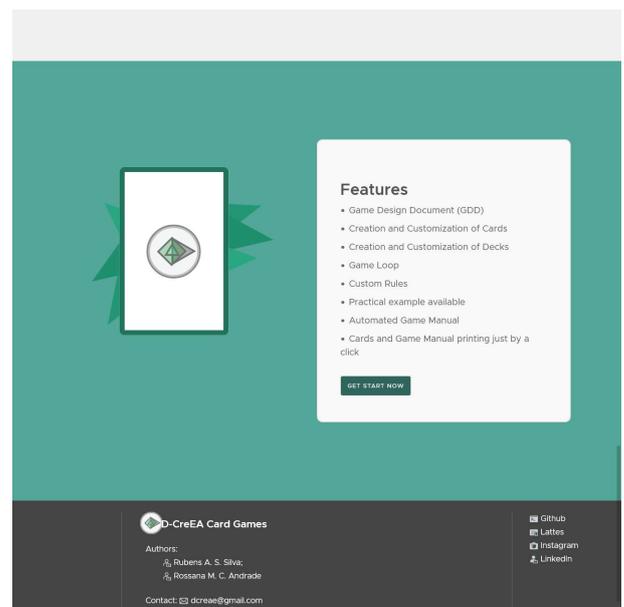
(a) Landing page 1-4



(b) Landing page 2-4



(c) Landing page 3-4

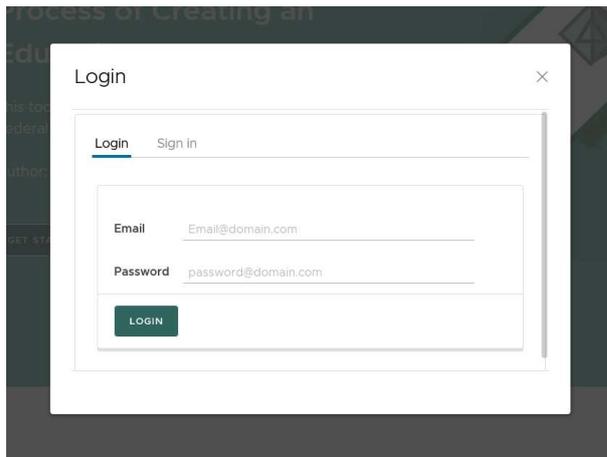


(d) Landing page 4-4

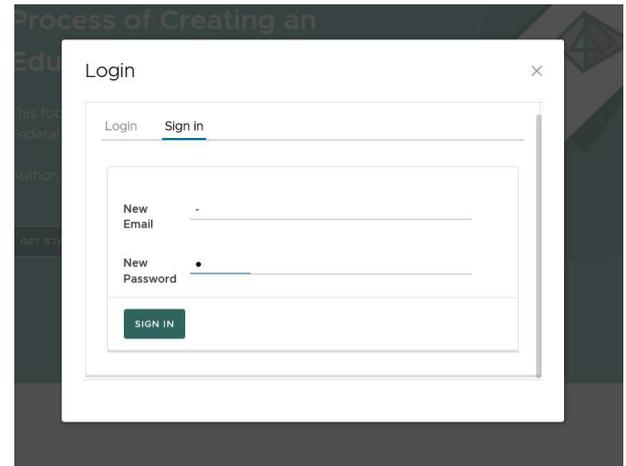
Figura 14 – Landing page da D-CreEA

maiores detalhes as especificidades desse campo. Por exemplo, na Figura 16, vemos um *tooltip* em verde.

Na tela de *GDC* há o formulário que representa o game design canvas do jogo (SARINHO, 2017), que consiste em um conjunto de informações relacionadas, como nome, autores, descrição do jogo, gameplay simplificado, e número mínimo e número máximo de jogadores. Aqui, todos os campos devem ser preenchidos, pois são as informações básicas do jogo que aparecerão no manual. A Figura 16 apresenta a imagem dessa tela.



(a) Login



(b) Criar usuário

Figura 15 – Login da D-CreEA

Figura 16 – Game Design Canvas da D-CreEA

Fonte: Autor

Em comparação com a interação sem a plataforma, o usuário precisaria alterar tais informações diretamente em código *JSON* e não iria dispor de ajuda ou dicas presentes nos *tooltips*. Além disso, suas alterações só seriam vistas no momento da criação do PDF para impressão.

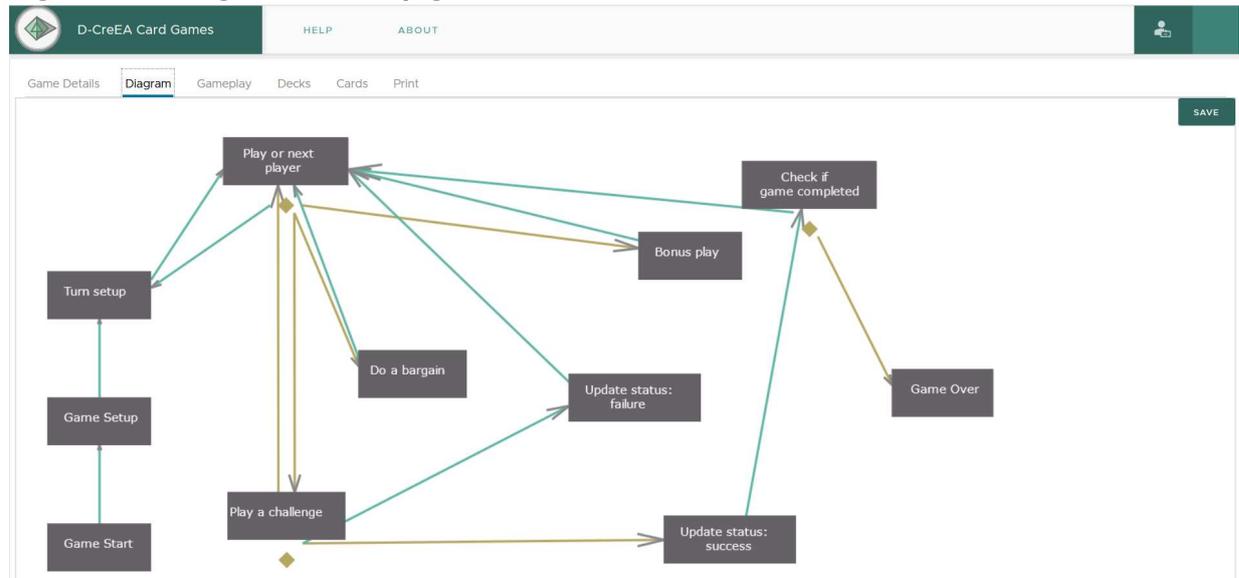
#### 5.4.4 Diagrama

Na tela de Diagrama, o usuário pode visualizar os estados do jogo e as transições e assim entender o fluxo de jogo e os tipos de jogada que existem durante a criação do jogo de cartas.

Na Figura 17 pode-se observar os estados do jogo exemplo, que vem implementado

ao criar um novo jogo. Note que existem dois tipos de setas: amarelas e verdes. As verdes representam as transições de estado para estado, enquanto as amarelas representam as transições por condição, provenientes de regras condicionais. Como dito no Capítulo 4, os estados que podem ser seguidos são escolhidos a partir de decisões que dependem do contexto atual de jogo, como, por exemplo, se o jogador já jogou ou se tem um efeito que o impeça de jogar.

Figura 17 – Diagramas de um jogo da D-CreEA



Fonte: Autor

As cores são por padrão cinza, mas podem ser editadas na aba *Gameplay*. Isso facilita na identificação de grupos de estados com significados semelhantes, como, por exemplo, um caminho para um certo tipo de jogada. Também podem ser editados os estados e as regras de cada um. Decidiu-se separar estas duas ações pela complexidade da edição/criação de regras e estados. Contudo, qualquer alteração feita na aba *Gameplay* é imediatamente atualizada nesta aba. Esta interação não é a ideal, mas foi adotada para atenuar a complexidade e se encaixar no cronograma de desenvolvimento deste trabalho.

Em comparação com a interação sem a plataforma WEB, o usuário não teria a disposição nenhum diagrama e, caso quisesse consultar o game loop do jogo, precisaria utilizar uma ferramenta de diagramas para criar manualmente tal diagrama, cuja edição seria passiva de erros, tanto de lógica quanto de restrições da linguagem. Na plataforma, esta criação é feita de forma automatizada e, além disso, as restrições são respeitadas, pois a parte de criação e edição de estados impediria a edição de estados e regras fora das normas da linguagem.

### 5.4.5 *Gameplay*

A tela de *Gameplay* é responsável pela criação e edição de estados e regras, sempre respeitando as restrições de criação da linguagens. Nesta tela, como pode ser visto na Figura 18, temos três setores: o primeiro, acima à esquerda, serve para criar/editar um estado de jogo; logo abaixo, o segundo serve para criar/editar as regras do dado estado; e, no terceiro, à direita, é mostrada a lista de estados de jogo, onde se pode consultar um estado e selecionar para editar ou apagar algum deles.

Na primeira seção, em “Create New State” (Figura 19), pode-se definir o rótulo (*label*) do estado, o propósito (*purpose*) e a cor do estado que aparecerá no diagrama na aba *Diagram*. Assim que apertar “Save”, o estado aparecerá na lista à direita, mas, para isso, este precisa ter pelo menos uma transição para poder ser salvo. Logo, o próximo passo é criar as regras deste estado.

Logo abaixo, na seção “Rules for this state”, o usuário pode criar e editar as regras do estado atual. O setor possui duas abas, uma para regras *Statement* (Figura 19c) e uma para regras de transição (Figura 19d). A aba de regras *Statement* cria regras que alteram os status dos jogadores, mas não provocam mudança no estado atual do diagrama - esse é o papel das regras de transição, as quais possuem dois tipos: as transições simples e as regras condicionais. As transições simples ocorrem quando todas as regras *Statement* já foram aplicadas no contexto atual, e servem apenas para dizer aonde seguir. Já as regras condicionais indicam para onde seguir baseando-se em decisões definidas nestas regras.

Vale ressaltar que um estado não pode ter regra de transição simples e condicionais ao mesmo tempo, são excludentes pela própria natureza e estrutura de tais regras: você escolhe qual estado baseado em contexto ou uma rota padrão independente. De toda forma, é impossível estagnar em um estado de jogo por erro de design, o fluxo sempre segue.

Por último, observe que, se o usuário simplesmente criar um estado, ele não é alcançável por qualquer outro. Então se faz necessário ter em mente de qual estado este recém criado será o próximo passo. Por exemplo, se eu quero criar mais um tipo de jogada, o meu novo estado deve ser alcançado a partir do estado em que se escolhe qual o tipo de jogada será feita - no caso do jogo exemplo, esse estado é o *play or go to next player*, e uma das condições dele deve escolher o estado recém criado.

Figura 18 – Tela de edição de regras e estados da D-CreEA

The screenshot shows the 'Gameplay' tab of the D-CreEA Card Games interface. It features a top navigation bar with 'D-CreEA Card Games', 'HELP', and 'ABOUT'. Below the navigation bar are tabs for 'Game Details', 'Diagram', 'Gameplay', 'Decks', 'Cards', and 'Print'. The 'Gameplay' tab is active, showing three main panels:

- Create New State Form:** A form for creating a new state. It includes a 'Label' field with 'New State', a 'Purpose' field with 'to be new', and a 'Color' field. There are 'CLEAR' and 'SAVE STATE' buttons at the bottom.
- Rules For This State:** A section for defining rules for the current state. It has tabs for 'Statement Rules' and 'Transition Rule'. A table below shows a message: 'We couldn't find any rule' and '0 rules'.
- States List:** A table listing existing states with columns for 'Label', 'Purpose', and 'Conditional?'. The table contains 11 entries.

Label	Purpose	Conditional?
Turn setup	sets the setup of every beginning of a turn	false
Game Start	sets the beginning of the game	false
Play or next player	Decides what is the play	true
Game Over	The end of the game. Lets see the winners	false
Bonus play	To use the effect of a Bonus card	false
Game Setup	sets the setup of every beginning of a game	false
Update status: success	Describes the way to change the points in case of win a challenge	false
Play a challenge	Describes the way to resolve a challenge	true
Do a bargain	Negotiate with another player	false
Update status: failure	Describes the way to change the points in case of miss a challenge	false
Check if game completed	if someone won, the game must end	true

Fonte: Autor

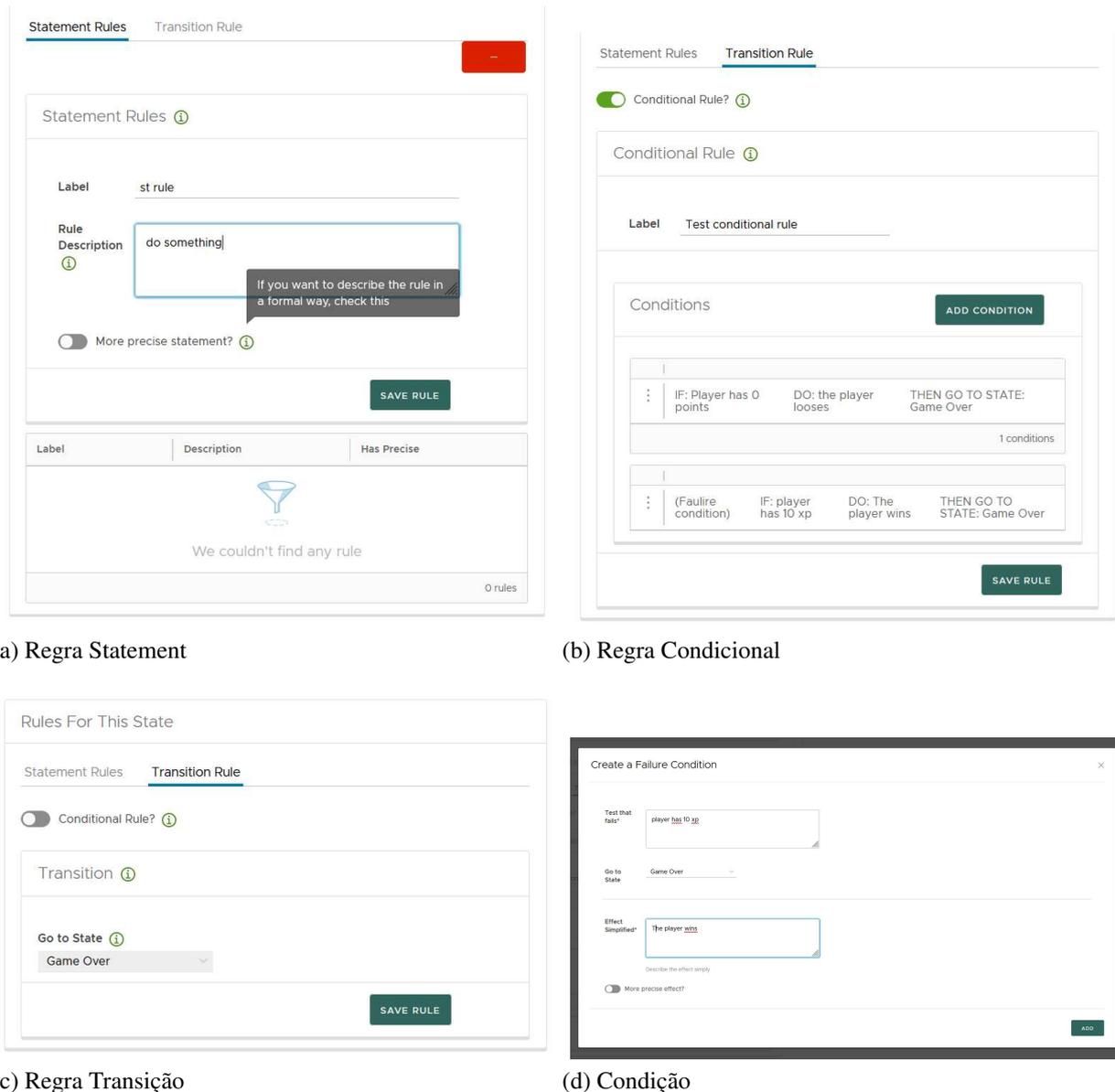
#### 5.4.6 Decks

A tela de *Decks* é responsável pela criação e edição de Baralhos do jogo. Nela, o usuário pode escolher quantos baralhos deseja inserir no jogo e, em cada um, quais os campos que devem aparecer em todas as cartas, sua cor, seu nome e sua descrição.

Como pode ser visto na Figura 20, o design do *deck* pode ser conferido em tempo real, e qualquer mudança é mostrada nas cartas exemplo à esquerda. Nas cartas exemplo, a parte da frente da carta é vista mais à esquerda e a de trás, à direita. A parte da frente pode ter os campos *título*, *arte da carta*, *descrição*, *efeito da carta* e *pontuações*. Na parte de trás, pode-se ter *título*, *respostas*, *efeito da carta* e *pontuações*. Ainda, há a cor base das cartas e que varia nos campos para destacar onde começa e termina cada um.

Observe que a carta não necessariamente precisa ter efeitos ou pontuações, dependendo do jogo. Assim, os campos de efeitos e pontuações (frente e trás) não são obrigatórios. Além disso, nem toda carta precisa ter respostas, então a parte de trás pode vir completamente vazia, onde aparece apenas o logo da *D-CreEA*.

Por último, as pontuações que uma carta pode ter são três: custo (*cost*) - losango amarelo, *level* - círculo azul, e recompensa (*earnings*) - triângulo verde. Seguindo técnicas de



(c) Regra Transição

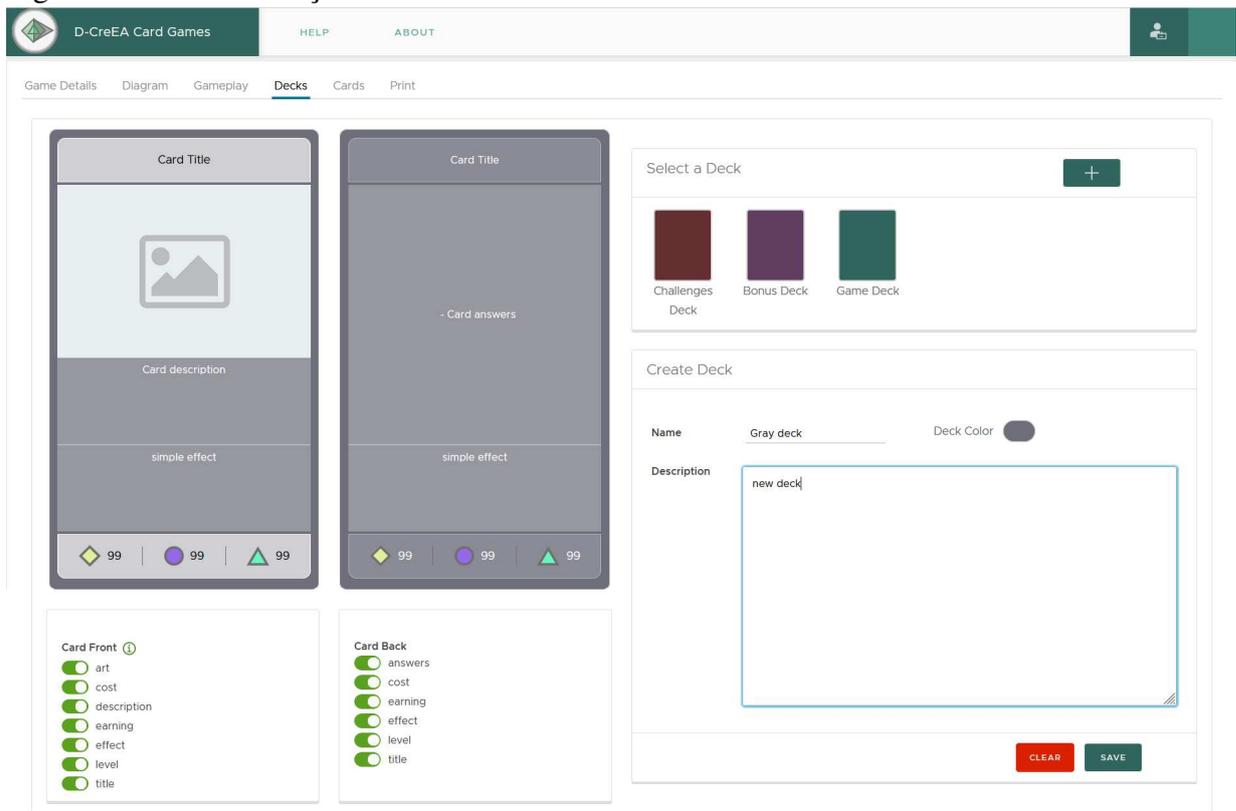
(d) Condição

Figura 19 – Criação de Regras da D-CreEA

balanceamento de jogo discutidas no Capítulo 2, é possível balancear o quão boa uma carta é balanceando entre custo de uso da carta e recompensa. Assim, se uma carta tem um custo alto e uma recompensa baixa, esta carta pode ser considerada uma carta “ruim” e vice-versa.

Por outro lado, o criador do jogo pode controlar quando uma carta pode ser usada ao introduzir a pontuação de *level*. Cada usuário pode ter um *level* que é melhorada no decorrer das partidas, a partir do cumprimento de certos desafios (definidos pelo criador do jogo). Assim, algumas cartas só poderão ser usadas após o usuário ter o *level* necessário, sendo esse *level* alcançado após um certo número de rodadas. Por exemplo, se a cada rodada o jogador aumentar seu level em 1, a cada rodada cartas mais difíceis (com level maior) e consequentemente, com maior recompensa, poderão ser resolvidas. Isso evita cartas que podem acabar com o jogo

Figura 20 – Tela de edição de Decks da D-CreEA



Fonte: Autor

muito cedo, e ainda ajuda a organizar os tipos de jogadas de acordo com a fase de jogo e grau de dificuldade. Contudo, caso o criador não queira utilizar nenhum tipo de pontuação, basta desabilitar, e definir as regras de vitória/derrota de forma independente.

#### 5.4.7 Cards

Esta tela é responsável pela criação e edição das cartas de cada baralho criado. Nela, o usuário pode escolher em qual deck deseja inserir/editar a carta, e irá preencher todos os campos que aparecem, de acordo com os escolhidos para aquele deck. Note que na Figura 21, apenas os campos de título, repetições, descrição e respostas aparecem, indicando que o deck em questão não possui os outros campos destravados.

Ainda, a carta possui um campo para uma imagem, onde o usuário poderá escolher dentre as imagens pré-definidas qual mais se adequa ao contexto daquela carta. Não é possível realizar o *upload* de uma imagem qualquer, pois um sistema de validação de imagens complexo é necessário, tanto para validar tamanhos e formatos, quanto para censurar imagens “problemáticas”.

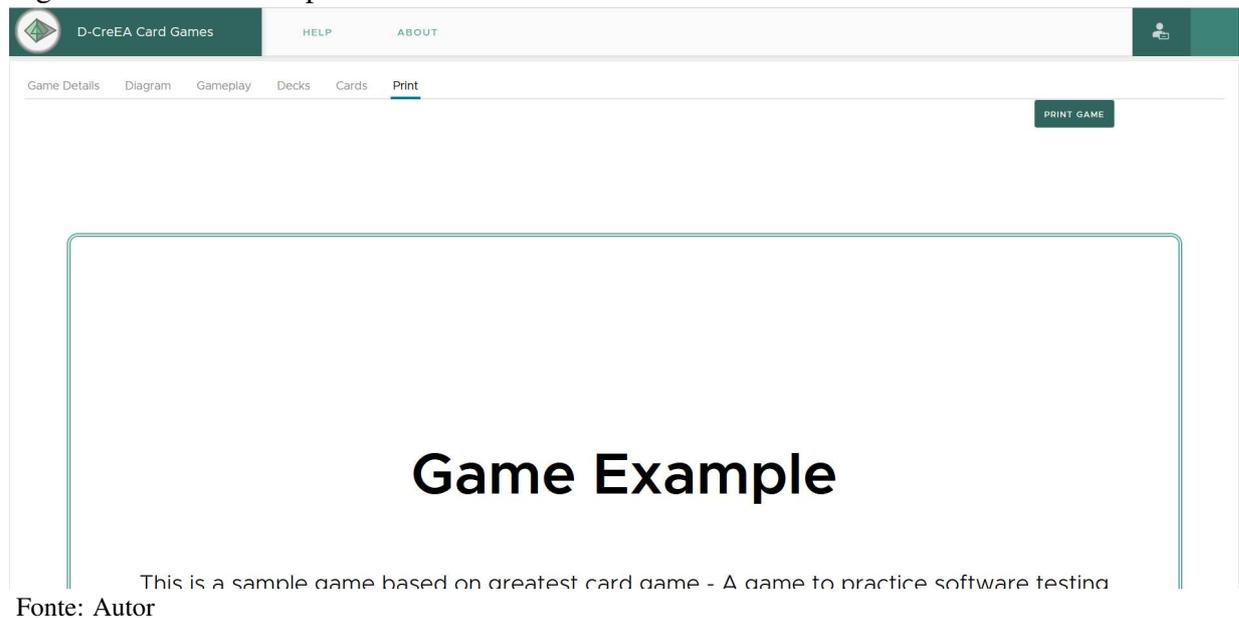
Figura 21 – Tela de edição de cartas da D-CreEA

Fonte: Autor

#### 5.4.8 Impressão

A tela de impressão é responsável por apresentar o manual do jogo e as cartas prontas para serem baixadas pelo navegador, em formato PDF. Ela apresenta todo o layout do manual, as informações do *GDC*, dos *decks* e tudo relacionado ao jogo até então. Vale ressaltar que essa página pode ser consultada a qualquer momento durante o desenvolvimento, mas as informações só serão mostradas se o *gameloop* estiver de acordo com as restrições da linguagem *D-CreEA*, mostrando um erro, caso contrário. A Figura 22 ilustra a tela e o manual impresso do jogo pode ser consultado neste link: [shorturl.at/LOTXZ](http://shorturl.at/LOTXZ). É possível ver alguns *bugs* na impressão, e esse é um dos principais pontos de melhoria da ferramenta. Contudo, isso não impede a checagem das informações e isso será priorizado nas melhorias futuras da *D-CreEA*.

Figura 22 – Tela de impressão da D-CreEA



## 5.5 Resumo do Capítulo

Este capítulo tratou de apresentar o desenvolvimento da plataforma *D-CreEA*, desde os requisitos, passando pelo projeto e implementação até os detalhes de cada tela implementada. O capítulo foi então dividido para tal: dos detalhes do processo de desenvolvimento até a apresentação da plataforma *D-CreEA* implementada.

Primeiro apresentou-se o processo de engenharia de requisitos, para definir o que a plataforma deve e não deve fazer. Seguindo o fluxo de definição de requisitos apresentado em (SOMMERVILLE, 2015), os passos seguidos foram: um estudo de viabilidade, elicitação e análise de requisitos e especificação de requisitos funcionais e não funcionais. Nesta fase, houve inúmeras revisões e alterações, como é natural em um processo de criação de software incremental.

O estudo de viabilidade arguiu sobre, como o próprio nome sugere, a viabilidade do projeto, procurando saber quais as dificuldades e expertises que seriam mais relevantes para projeto. Daí, definiu-se o tipo de aplicação (nesse caso, WEB) e as tecnologias que seriam usadas para o desenvolvimento.

Com o estudo de viabilidade concluído, partiu-se para a elicitação e análise de requisitos, onde definiu-se o conjunto de funcionalidades, escrito em linguagem natural.

Após isso, apresentou-se a especificação de *RUs*, formalizando as funcionalidades definidas na fase anterior. A partir dos *RUs*, o conjunto de requisitos funcionais do sistema foi

fechado, especificando detalhes nos *RFs*.

Depois, os requisitos não funcionais (*RNFs*) foram definidos baseando-se nas restrições do estudo de viabilidade e requisitos funcionais do sistema.

Em seguida, criou-se a arquitetura da *D-CreEA*, baseada em *MVC* e em Camadas, com uma camada transversal de segurança, apresentado na Figura 12.

Por último em relação ao projeto, os detalhes da implementação foram mostrados, como separação de projetos, provedor, sistema de gerenciamento de banco de dados, *continuous delivery*, entre outros.

Já para a parte de apresentação da ferramenta, mostrou-se os detalhes de cada tela, apresentando suas funcionalidades, suas restrições e os resultados esperados de cada uma. As telas apresentadas foram: *Home*, *login*, *Game Design Canvas*, *diagramas*, *gameplay*, *decks*, *cartas e impressão*. Também ressaltou-se os pontos de melhoria, visto que este projeto poderá ser continuado e melhorado após a defesa desta dissertação.

## 6 AVALIAÇÃO

Neste Capítulo são apresentados o método e os resultados da avaliação da *D-CreEA*. A plataforma foi avaliada como um todo, entendendo não só a adequação funcional para a criação de jogos, mas quais requisitos de qualidade são mais importantes para o grupo focal e quão bem a *D-CreEA* se adequa a eles. Na Seção 6.1 é apresentado o procedimento de avaliação da *D-CreEA*; na Seção 6.2 são expostos os resultados da avaliação piloto da plataforma; por último, na Seção 6.3 são mostrados os resultados da avaliação da *D-CreEA* com os professores.

### 6.1 Método de Avaliação

A avaliação de uma *DSML* é de certa forma peculiar, visto que é necessário avaliar vários aspectos da linguagem e também da plataforma, entretanto, esses aspectos muitas vezes diferem diametralmente. Por exemplo, para avaliar a *DSML* pode-se abordar extensividade, eficiência, completude e outros requisitos não-funcionais (ou características de avaliação de qualidade de software (ISO, 2011-13)).

Para realizar a avaliação, as abordagens que dependem de interação direta entre pesquisadores e grupo focal, tais como entrevista e teste dos jogos criados (MARCHIORI *et al.*, 2011; MOSSMANN *et al.*, 2016), não puderam ser realizadas dadas as circunstâncias de distanciamento social enfrentadas no contexto da pandemia de COVID-19 que ocorreu durante toda esta pesquisa. Para contornar esta situação, o método escolhido deveria ter como *target* principal a avaliação de *DSMLs* e a avaliação não deveria demandar interação social direta entre os participantes e/ou os autores. Por isso, adotou-se o método de avaliação *FQAD* (KAHRAMAN; BILGEN, 2013).

O *FQAD* tem por objetivo avaliar quesitos de qualidade de *DSMLs*, em termos de adequação funcional, usabilidade, extensividade, dentre outros. Os avaliadores escolhem os quesitos que mais se aplicam ao contexto da *DSML* e atribuem notas na escala likert, refletindo suas satisfações para aquele quesito. Assim, ao final, somente os quesitos mais relevantes para a *DSML* (sob o ponto de vista do público-alvo) recebem uma nota e, dessa forma, os avaliadores podem focar nos elementos de qualidade mais relevantes.

No caso da *D-CreEA*, após a sua fase de implementação, a próxima etapa é a avaliação com professores. O objetivo desta avaliação é receber *feedback* a partir do público-alvo da plataforma (professores dos cursos de Computação) de forma prática e gerar entendimento

sobre os pontos fortes e de melhoria do uso da *DSML* e da ferramenta de software.

Sendo assim, o método de avaliação, FQAD, foi utilizado e o seguinte passo-a-passo aplicado ao *D-CreEA*:

1. *Briefing*: Os avaliadores recebem um breve *briefing* da ferramenta *D-CreEA* para entender seus conceitos básicos e como usá-la;
2. Formulário pré-uso: aqui, deve-se acessar um formulário inicial (chamado aqui de *formulário pré-uso*), para captar as expectativas dos avaliadores sobre a ferramenta. Em relação a este formulário, mais detalhes a seguir.
  - a) No formulário pré-uso, os avaliadores devem elicitare os requisitos não-funcionais (ou características de qualidades).
  - b) os requisitos não funcionais elicitados, seguindo o padrão do *FQAD*, foram: usabilidade, confiabilidade, adequação funcional (*suitability*), extensividade, produtividade, expressividade, compatibilidade, reusabilidade, integrabilidade e manutenibilidade. Vale ressaltar que não é necessário escolher todos se não julgar pertinente. Ainda, essa seleção de requisitos não-funcionais será feita antes de ter qualquer contato com a ferramenta, tendo em mãos apenas o *briefing*.
3. Utilizar a ferramenta: neste passo, deve-se acessar o link para a *D-CreEA* implementada e modificar um jogo *template* utilizando a ferramenta como auxiliar. O jogo será padrão para todos os avaliadores e estes terão os passos a executar de antemão. Assim, o desenvolvimento terá um *setup* inicial comum, variando apenas a experiência individual de desenvolvimento de cada avaliador. Os passos para o uso da ferramenta são os seguintes:
  - a) Visitar a página de ajuda;
  - b) Criar usuário;
  - c) *Gameplay*: adicionar autor (opcional: alterar outras informações);
  - d) *Diagram*: alterar a posição de qual(is)quer estado(s) e salvar;
  - e) *Decks*: criar um novo ou alterar qualquer *deck* existente
  - f) Criar cartas do jogo;
  - g) Imprimir o jogo.
4. formulário pós-uso: após o uso da ferramenta, os avaliadores devem responder a um segundo formulário, onde avaliam a correspondência das expectativas iniciais citadas no formulário pré-uso.
  - a) Para cada requisito não-funcional elencado, o avaliador deve dar a nota que corres-

- ponde a sua opinião sobre o suporte que a *D-CreEA* oferece ao requisito;
- b) As notas para cada um variam de 1 a 5, onde 1 é insatisfatório e 5 é totalmente satisfatório;
  - c) Ainda, os avaliadores poderão opinar de forma aberta, através de elogios ou sugestões de melhorias sobre a ferramenta e sua experiência de uso.

A distribuição do formulário foi feita após a conclusão da primeira versão da plataforma *D-CreEA* e após a execução dos testes manuais em ambiente online. A participação na pesquisa foi feita de forma totalmente independente e individual, de forma que cada participante não tinha conhecimento sobre os outros participantes, e não tiveram contato direto ou indireto com os outros. Os questionários foram enviados via link do *Google Forms*. Ao abrirem o link com o formulário, os participantes recebiam um texto informando os objetivos da pesquisa, os pesquisadores responsáveis e que seus dados estariam protegidos e só seriam usados para fins desta pesquisa. O processo de avaliação foi feito sem limite de tempo e o *feedback* dos professores revelou que a duração variou entre 20 e 35 minutos. As perguntas do questionário estão presentes nos Apêndices B e C.

## 6.2 Avaliação Piloto

Antes de aplicar a avaliação foi realizado um piloto com um(a) pesquisador(a) pertencente ao público-alvo, a fim de entender se a avaliação estava pronta para aplicação com todos os professores. O piloto seguiu o mesmo procedimento da avaliação final.

Sobre a avaliação do piloto da plataforma, o(a) participante respondeu que tem nível 3 de familiaridade em uso de jogos e *game design*, em uma escala de 1 a 5. Sobre os requisitos não funcionais, no questionário pré-uso, ele(a) marcou todos exceto Expressividade como relevantes para o sistema. Já no pós-uso, os que apresentaram desempenho “Bom” ou “Muito Bom” foram usabilidade, confiabilidade, produtividade e reusabilidade. Já os com resultado “Regular” foram: adequação funcional, extensividade (marcado como irrelevante no pré-uso), integrabilidade e manutenibilidade. O marcado como “Ruim” foi apenas compatibilidade e não houve respostas para “Muito Ruim”.

O *feedback* nas perguntas abertas revelou os seguintes pontos:

- Segundo o(a) pesquisador(a), a plataforma pode ser usada inclusive para jogos não educacionais, e foi vista como “bem interessante e útil para auxiliar a criação de jogos de cartas”.

- Na versão do piloto, existiam *bugs* na criação de regras na tela de gameplay.
- Houve dificuldade em questões de usabilidade, onde o(a) pesquisador(a) não entendeu alguns pontos de como usar a ferramenta, principalmente na parte de diagramas. Contudo, segundo ele(a), o *bug* na criação de regras pode ter impactado no entendimento dessa funcionalidade.
- Por último, foi sugerida a inserção de imagens nas cartas, para que elas fiquem mais chamativas e interessantes aos jogadores.

O piloto mostrou-se muito importante, pois aspectos positivos da plataforma foram destacados, como relevância e aplicabilidade, e ainda revelou pontos de melhoria que só puderam ser vistos a partir da utilização real de terceiros, como *bugs*, dificuldades de uso e novas funcionalidades. Estes dados guiaram a evolução da plataforma para a avaliação final com professores, apresentada na próxima seção.

### 6.3 Resultados da avaliação com professores

Com os resultados obtidos no piloto, trabalhou-se para consertar os *bugs* e implementar melhorias e modificações na interface do sistema, a fim de melhorar a experiência do usuário e evitar falhas críticas, como as enfrentadas pelo(a) participante do piloto.

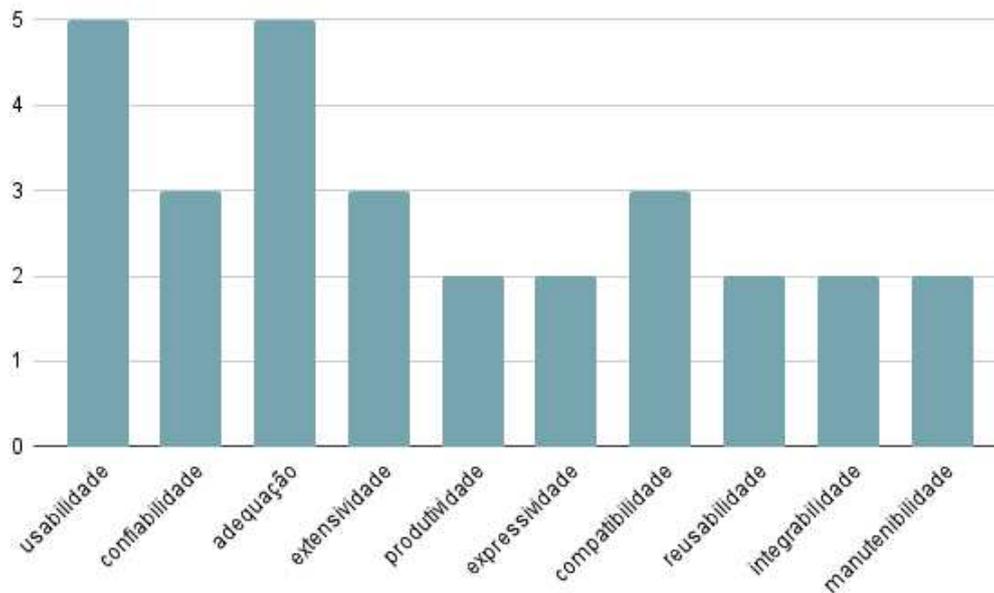
Com as alterações implementadas e testadas, partiu-se para a avaliação real. Ao todo, houve a participação de 5 professores, todos contatados de forma individual pelo próprio autor, que aceitaram participar de livre e espontânea vontade, sem promessa de recompensa ou brinde.

Sobre o questionário pré-uso, no quesito nível de familiaridade em uso de jogos, 3 (três) revelaram ter nível "Regular" de conhecimento, enquanto 1 (um) respondeu ter nível "Bom" e 1 (um) respondeu "Muito Bom". Já no quesito nível de familiaridade em design de jogos, 2 (dois) revelaram ter nível "Ruim" de conhecimento, 1 (um) revelou ter nível "Regular", 1 (um) respondeu ter nível "Bom" e 1 (um) respondeu "Muito Bom".

Sobre as características, pode-se observar na Figura 23 que as mais relevantes para a plataforma *D-CreEA*, segundo os avaliadores, são usabilidade e adequação funcional, escolhidas por todos. Em seguida, temos confiabilidade, extensividade e compatibilidade, escolhidos por pelo menos 3 participantes.

Sobre o questionário pós-uso, observou-se um fenômeno interessante: no item “*Para cada característica de software marcada no questionário anterior, atribua nota que reflète o nível de adequação da ferramenta com a característica*”, houve professor que marcou características

Figura 23 – Características escolhidas no questionário pré-uso



Fonte: Autor

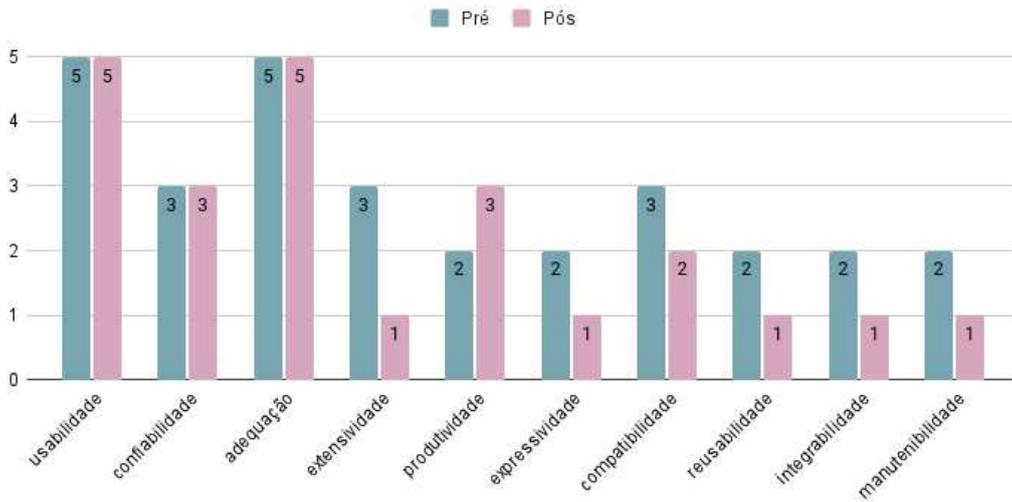
no formulário pré-uso como relevante e no pós-uso não atribuiu nota, e vice-versa, embora existisse o aviso *“Esta seção está relacionada com as respostas do questionário anterior, onde você marcou as características que você espera estejam presentes no D-CreEA. Por favor, tenha em mãos estas respostas, que lhe foram enviadas por email”*.

Com isso, interpretamos que os participantes mudaram de ideia após o uso da *D-CreEA* sobre as características relevantes, escolhendo novamente quais as mais relevantes e atribuindo nota a elas, e ignorando as menos relevantes. A comparação entre as características escolhidas é apresentada na Figura 24.

Tendo isso em mente, pode-se observar a relação entre os requisitos não-funcionais avaliados no questionário pós-uso e o nível de satisfação apontado por cada pesquisador, mostrado na Figura 25. Pode-se notar que não houve notas abaixo de "Muito Bom", "Bom" ou "Regular", o que indica um sucesso para a plataforma, destacando Confiabilidade, Adequação e Produtividade, em termos de respostas "Bom" ou acima. Todavia, a nota predominante foi "Regular", com 9 respostas, indicando que há pontos de melhoria no desempenho de várias características de qualidade, principalmente em Usabilidade.

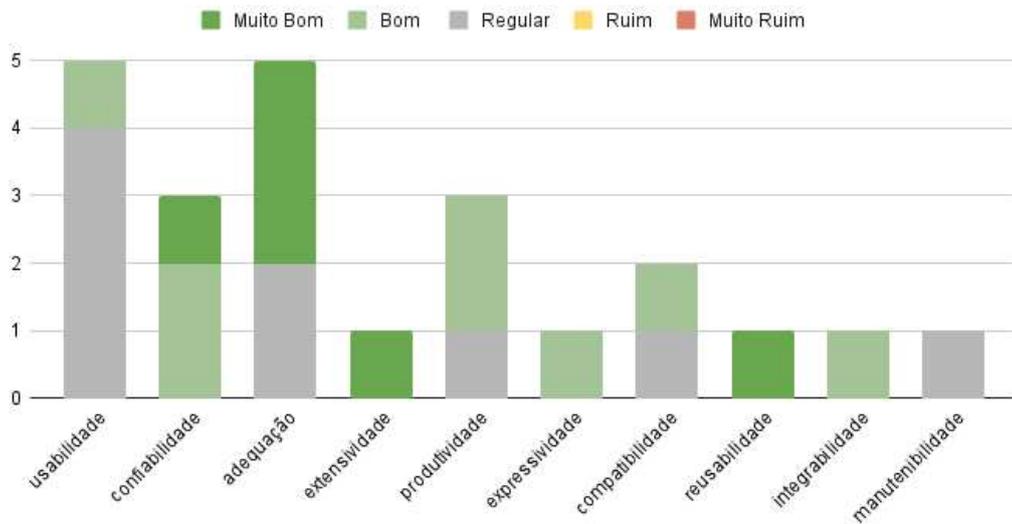
Por outro lado, observando os relatos nas respostas abertas, podemos notar vários aspectos mais detalhados do desenvolvimento via *D-CreEA*. Nota-se que os professores aprovaram a ideia, relatando que *“a ferramenta proposta é uma excelente iniciativa para apoiar educadores na criação de jogos”* e *“pode facilitar o processo de criação de cartas para uso didático por*

Figura 24 – Características escolhidas no questionário pré-uso versus pós-uso



Fonte: Autor

Figura 25 – Relação entre características e satisfação da D-CreEA



Fonte: Autor

professores”.

Contudo, houve dificuldades na criação dos jogos em alguns aspectos e os seguintes pontos de melhoria foram levantados a partir da avaliação:

- Melhorar usabilidade, principalmente em relação ao *feedback* visual na manipulação do diagrama, ao *feedback* da criação de cartas, à disposição dos elementos de criação dos *decks*.
- Para a maioria dos participantes, a tela de *gameplay* e diagramas gerou confusão e dificuldade, tanto em usabilidade, quanto funcionalidade.
- Houve dificuldade com relação a parte de ajuda e tutorial, onde os participantes tiveram

dificuldade em realizar as atividades propostas.

- Melhorar a usabilidade de uma forma geral, principalmente em questão de *feedback* e ajuda;
- Implementar tutoriais mais efetivos, interativos, por vídeo, que de fato demonstrem a utilização da ferramenta. Há atualmente implementado apenas a explicação textual do funcionamento e *prints* das telas;
- implementar mecanismos de balanceamento de jogos de forma explícita.
- Prover meios mais claros para que o usuário consiga definir o conceito do jogo em alto nível, como “objetivo, regras do jogo e relações entre os decks”, antes de implementá-lo a nível de regras e cartas.

Pode-se observar que a maioria dos pontos de melhoria relatados estão relacionados a usabilidade, e este deve ser o ponto principal na evolução da *D-CreEA*. Contudo, houve pedidos de melhorias quanto à funcionalidades, tanto para criar novas quanto para melhorar as existentes. Além disso, a questão da ajuda foi um ponto bastante relatado e isso deve também ser priorizado na evolução da mesma.

Já sobre a avaliação objetiva, vimos que todas as características de qualidade estão assegurados em um nível Regular ou acima, o que pode ser encarado como satisfatório.

Concluindo, pode-se considerar esta avaliação positiva, pois foi possível que os professores conseguissem concluir as atividades e entender o objetivo da *D-CreEA* como ferramenta, apesar de terem dificuldades no desenvolvimento sob alguns aspectos. Contudo, há pontos de atenção que devem ser melhorados para que *D-CreEA* seja de fato uma ferramenta utilizável e comercializável.

## 6.4 Resumo do Capítulo

Este Capítulo apresentou a avaliação da *D-CreEA* com os professores de Computação.

Primeiro, foi apresentado o procedimento da avaliação, onde pode-se observar as restrições, escolhas e os passos para avaliar a plataforma.

Após isso, pode-se conferir os resultados do estudo piloto, realizado com um(a) pesquisador(a), seguindo o procedimento original de avaliação, e que mostrou que o *setup* de fato era aplicável para o contexto de avaliação proposto. Além disso, pontos de melhoria foram identificados, para que a avaliação pudesse de fato ser executada.

Com as melhorias implementadas, foi realizada a avaliação com os professores, na qual obteve-se 5 respostas. Com isso, os resultados indicaram que a plataforma *D-CreEA* é uma ferramenta relevante ao que se propõe, mas necessita de evolução principalmente em relação a usabilidade e módulos de ajuda. Isso deve ser priorizado em trabalhos futuros para a sua evolução.

## 7 CONCLUSÃO

Este trabalho apresentou a *D-CreEA*, uma linguagem de modelagem específica de domínio para auxiliar no desenvolvimento de jogos de cartas educacionais analógicos, e a plataforma desenvolvida baseada na linguagem.

Este Capítulo apresenta uma visão geral dos resultados obtidos com este trabalho, as limitações e os trabalhos futuros advindos desta dissertação. A Seção 7.1 faz um apanhado geral dos resultados, A Seção 7.2 fala das contribuições do trabalho na área de Computação, em particular da linha de pesquisa de Engenharia de Software, a Seção 7.3 lista as limitações e ameaças à validade do trabalho e, por último, a Seção 7.4 apresenta os trabalhos futuros como continuidade desta pesquisa.

### 7.1 Visão Geral

Cada vez mais jogos educacionais vêm ganhando espaço como um método eficaz de ensino e prática de conhecimento e treinamento. Nos últimos anos, como é mostrado no Capítulo 2, vários trabalhos vêm sendo publicados com resultados promissores. A forma de ensinar está em constante mudança e metodologias que priorizam o ensino de forma prática e lúdica contribuem de forma positiva nesse advento. A utilização de jogos digitais para esse fim é algo bem estabelecido e os jogos analógicos educacionais vem ganhando espaço, mas de forma mais lenta quando comparados aos jogos digitais. Há espaços inexplorados nesse campo de jogos analógicos que poderão trazer frutos e é nesse contexto que este trabalho se insere.

A criação de um jogo educacional, seja digital ou analógico, não é uma tarefa fácil. É necessário balancear mecânicas e dinâmicas com conteúdo pedagógico para possivelmente obter uma experiência agradável e educativa. É justamente este problema que o *D-CreEA* busca ajudar a remediar.

*D-CreEA* tem como objetivo facilitar e guiar o desenvolvimento de jogos analógicos educacionais de cartas. Para isso, buscou-se implementar uma linguagem de modelagem de domínio específico (*DSML*) para o domínio dos jogos analógicos de cartas e, a partir dela, criar uma plataforma que provenha a interface entre a *DSML* e o usuário.

Com o intuito de validar a *D-CreEA* em relação ao seu grau de adequação funcional, implementou-se um jogo analógico de cartas educacional publicado, o *GreaTest Card Game (GTCCG)*. Com essa validação inicial foi possível verificar se a *DSML* realmente era capaz de

gerar jogos analógicos de cartas e, ainda, entender os pontos fortes e de melhoria em relação a linguagem.

A partir dessa validação e da constatação de adequação da linguagem, iniciou-se o processo de implementação da plataforma *D-CreEA*, um sistema como serviço (do inglês, *SaaS*) online para a criação de jogos analógicos que implementa a linguagem *D-CreEA*, cujo processo de concepção foi apresentado no Capítulo 5.4. Foi feito todo o processo de engenharia de requisitos, arquitetura, prototipação e implementação da plataforma até que fosse atingido o nível de aceitação necessário, com a *D-CreEA* online e funcional.

Com a plataforma *D-CreEA* implementada, partiu-se para a fase de avaliação com professores. A avaliação teve como base o *framework* de avaliação de DSMLs *FQAD* (KAHRAMAN; BILGEN, 2013), que avalia através de requisitos não-funcionais a qualidade da implementação. Os professores tiveram que, antes de usar a ferramenta, elencar, através de um questionário, quais requisitos não-funcionais esperavam que estivessem satisfatoriamente implementados na ferramenta. Depois tinham que utilizar a ferramenta, sendo guiados por um *briefing* da plataforma e a tela de ajuda para realizar uma série de atividades padrão, no intuito de implementar um jogo simples. Por último, eles deveriam responder a outro questionário a fim de comparar as experiências pré e pós uso, ao avaliar, via escala likert, os requisitos não-funcionais previamente selecionados como relevantes no questionário pré-uso.

A avaliação revelou que a plataforma *D-CreEA* pode ser de fato útil para a criação de jogos analógicos, segundo os participantes da avaliação, mas ainda precisa de ajustes e melhorias, principalmente em questões de usabilidade e ajuda.

## 7.2 Resultados e Contribuições

Os principais resultados deste trabalho são a linguagem e a plataforma *D-CreEA*. A linguagem é importante, pois se insere no contexto de criação de jogos analógicos ao definir e generalizar o domínio desse tipo de jogos. Com isso, foi possível implementar a plataforma *D-CreEA* e ainda introduzir a engenharia de software baseada em modelos (*MDE*) no contexto de jogos analógicos, algo que não foi encontrado na literatura.

A plataforma *D-CreEA* é importante, ainda, por implementar a linguagem e demonstrar que é, de fato, possível criar jogos analógicos com essa abordagem. Além disso, a plataforma tem grande potencial de se tornar um produto comercializável por estar implementada em ambiente WEB e ser estendível juntamente à linguagem *D-CreEA*.

Além disso, quatro (4) publicações resultaram desta dissertação como mostra a Tabela 4.

O trabalho (SILVA, 2021), publicação diretamente relacionada a *D-CreEA*, apresenta o desenvolvimento e a validação da linguagem. Além disso, várias outras pesquisas foram desenvolvidas ao longo da realização dessa dissertação, apresentados também na Tabela 4. O trabalho (SILVA *et al.*, 2021) apresenta um *survey* sobre evasão no curso de computação na UFC. Já o trabalho (BORGES *et al.*, 2019) apresenta o processo de hibridização do GreaTest via introdução de um aplicativo como ferramenta no *gameplay*. Em (LELLI *et al.*, 2020), é apresentado o processo de gamificação da disciplina de engenharia de software da UFC no contexto de ensino remoto emergencial.

Quadro 4 – Trabalhos publicados ao longo desta pesquisa

Autores	Título	Evento	Ano
Rubens Silva, Rossana Andrade	D-CreEA: DSML for Creating Educational Analog Card Games.	SBGames. IEEE	2021
Rubens Silva, Bosco Borges, Maria de Fátima Ferreira, Ismayle Santos, Rossana Andrade	Evasão em Computação na UFC sob a perspectiva dos alunos	WEI	2021
Rubens Silva, Rossana Andrade, Bosco Borges, Bruno Sabóia, Ismayle Santos, Joseane Paiva	Design and Evaluation of a Mobile Application for an Educational Card Game	JIS	2020
Valéria Lelli, Rossana Andrade, Lavinia Matoso, Rubens Silva, Guttemberg, Renata Farias, Jan	Gamification in Remote Teaching of SE Courses: ExperienceReport	SBES	2020
Rubens Silva, Bosco Borges, Ismayle Santos, Joseane Paiva, Rossana Andrade	Design e avaliação de um aplicativo móvel complementar para um jogo de cartas educacional	WiPlay	2019

Fonte: elaborado pelo autor.

Sendo assim, a contribuição efetiva deste trabalho é a criação de uma linguagem e sua derivada plataforma de criação de jogos analógicos, que propicia a criação de jogos por pessoas com pouca experiência em *game design*. Isso abre portas para professores de diversas áreas poderem criar seus jogos de forma embasada na literatura porém sem a carga de estudar a fundo o domínio dos jogos analógicos. Isso incentiva amplamente a adoção de jogos para a sala de aula e, conseqüentemente, incentiva a melhoria da motivação e engajamento do aluno.

Além disso, este trabalho se insere na literatura como um possível pioneiro na adoção de DSMLs para jogos analógicos, e isso incentiva a melhoria do campo de conhecimento sobre esse assunto. Dessa forma, a metodologia utilizada pode ser replicada e os resultados podem ser testados para verificar a eficiência da ideia.

Por último, destaca-se a possibilidade de estender a linguagem *D-CreEA* para jogos

de tabuleiro no geral, bastando definir o domínio de interação de jogos que envolvem o tabuleiro como mecânica.

### 7.3 Limitações e Ameaças à Validade

As limitações de uma pesquisa são os pontos que, dadas as adversidades, poderiam ser melhorados ou situações que prejudicaram a excelência do trabalho. Assim, os pontos de limitação deste trabalho foram:

- dificuldade em avaliar o trabalho de forma presencial, provocando limitações na amplitude da avaliação dada a pandemia global de Covid enfrentada durante praticamente todo o período da pesquisa de mestrado. Com isso, duas principais avaliações presenciais se tornaram inviáveis: avaliar os jogos criados pela *D-CreEA*; e avaliação de usabilidade assistida em ambiente controlado. Isso foi contornado utilizando técnicas de avaliação remota, porém de forma menos controlada.
- dificuldade para encontrar trabalhos relacionados. Durante a revisão da literatura sobre o tema de DSMLs para jogos analógicos, viu-se dificuldade em encontrar trabalhos que o abordassem. Assim, foi necessário buscar trabalhos que mais se aproximassem do tema, mesmo não aderindo exatamente ao mesmo.

No âmbito desta pesquisa, observaram-se também ameaças à validade (WOHLIN *et al.*, 2003), as quais são classificadas em quatro tipos: validade interna, validade de conclusão, validade de construção e validade externa.

Na validade interna é analisado se o relacionamento observado entre o tratamento e o resultado é causal, ou se o resultado pode ter sofrido influência de outro fator que não é controlado ou sequer foi medido. Na validade externa, são analisadas as condições que limitam a habilidade de generalizar os resultados do estudo para outros contextos. Já na validade de construção, são analisados os relacionamentos entre a teoria e a observação, ou seja, se o tratamento reflete causa e o resultado reflete o efeito. Por último, a validade de conclusão relaciona a habilidade de extrair uma conclusão correta a respeito dos relacionamentos entre o tratamento e o resultado do experimento.

Neste trabalho, as ameaças à validade internas identificadas estão detalhadas a seguir. Sobre o processo de criação da DSML, vale citar o problema da definição do domínio que é um trabalho subjetivo que depende dos trabalhos encontrados. Então, algumas interpretações pessoais podem ter ocorrido durante a extração de dados e análise de domínio. A fim de

minimizar esses vieses, a orientadora revisou os conceitos do domínio de forma independente. Outra ameaça é o design da interface da *D-CreEA* que foi feito de forma que agradasse o próprio autor desenvolvedor, de forma a priorizar o desenvolvimento em vez da usabilidade e, dessa forma, os gostos pessoais podem ter interferido na criação da interface da plataforma, ou seja, para o autor estava simples e claro, mas para outrem, não. Para minimizar essa ameaça, foi realizada a avaliação piloto e ajustes de usabilidade foram feitos após o mesmo.

Já as ameaças à validade externas identificadas neste trabalho tem relação ao fato de que não foram encontrados trabalhos apresentando processo de criação de jogos analógicos, apenas digitais. Portanto, a amostra de trabalhos coletados não está totalmente completa ou o campo de MDE para jogos analógicos não está maduro. Porém, o domínio da DSML está em constante evolução à medida que mais trabalhos são publicados e este trabalho pode gerar mais discussões a respeito. Além disso, outra ameaça é que os professores participantes da avaliação possuem diferentes expertises e diferentes conhecimentos sobre jogos, portanto, desenvolver um jogo pode ser mais fácil para uns e mais difícil para outros, e isso pode ter um impacto na experiência de uso da plataforma.

Por último, as ameaças à validade de conclusão identificadas neste trabalho referem-se às limitações em questão de avaliação, pois apenas 5 professores participaram e sob o ponto de vista estatístico, esse número não é ideal. Portanto, os resultados podem ser considerados apenas indícios. É também necessário avaliar a *D-CreEA* sob vários outros aspectos para atestar a eficiência da ferramenta. Além disso, as conclusões alcançadas são parciais, visto que não houve o desenvolvimento de jogo por um grupo de professores em contexto profissional. E, ainda, os jogos criados via *D-CreEA* não foram testados com uma audiência real de alunos. Sendo assim, várias outras avaliações devem ser feitas sob vários aspectos para atestar a eficiência da *D-CreEA*.

#### **7.4 Trabalhos Futuros**

Como trabalhos futuros, deve-se implementar melhorias na plataforma para que ela corresponda às expectativas de desenvolvedores de jogos e professores. Para isso, deve-se melhorar a usabilidade de algumas telas, consertar *bugs*, melhorar algumas funcionalidades e criar outras, como o balanceamento de jogos e tutoriais interativos de criação.

Além disso, é necessário avaliar os jogos gerados via *D-CreEA* com alunos reais em contexto de sala, para avaliar a jogabilidade dos jogos. Ainda, é necessário comparar o esforço

de criação de um jogo na forma tradicional e via *D-CreEA*, a fim de entender se há ganhos em termos de produtividade.

Após isso, é possível começar a estender a linguagem para dar suporte a jogos de tabuleiro gerais, ao incrementar o domínio da DSML e a linguagem em si. Depois, basta implementar a interface na plataforma *D-CreEA* e assim retornar ao ciclo de avaliação.

Uma outra evolução natural seria a expansão da DSML para inclusão do domínio de jogos analógicos de tabuleiro. Além disso, pode-se implementar outras plataformas finais de jogo, como jogos digitais e híbridos, visto que um princípio da DSML é a modularidade. Sendo assim, é possível implementar um novo algoritmo de modelo para artefato para integrá-lo a uma *engine* de desenvolvimento de jogos digitais (ex., Unity).

Além disso, pode-se estender esta pesquisa no sentido de testar a hipótese de que a *D-CreEA* é eficiente na geração de jogos, ao considerar um grupo de avaliadores significativo, contendo professores, desenvolvedores de jogos, pessoas sem experiência e outros, e comparar resultados para vários grupos em relação a vários aspectos do uso, desde usabilidade a resultados dos jogos.

Por último, como pesquisas futuras, é possível prover um modelo que sugere mecânicas de jogos mais adequadas ao tipo de necessidade do usuário (ex: tipo de jogo, dificuldade, assunto educacional) com base em um catálogo de mapeamentos de jogos existentes, implementados e classificados no domínio da *D-CreEA*.

## REFERÊNCIAS

- ALTUMBAY, D.; CETINKAYA, E.; METIN, M. Model driven development of board games. 2009.
- ANASTASIOU, L. Ensinar, aprender, apreender e processos de ensinagem. In: **Processos de ensinagem na universidade: pressupostos para as estratégias de trabalho em aula**. 7th. ed. Joinville: [S. n.], 2007. p. p.15–43.
- ANDRADE, G.; RAMALHO, G. L.; GOMES, A. S.; CORRUBLE, V. Dynamic game balancing: An evaluation of user satisfaction. **AIIDE**, v. 6, p. 3–8, 2006.
- AZEVEDO, G. S. de; LIMA, L. S.; SARINHO, V. T. Dengueside survival: Produzindo um jogo de tabuleiro estilo rpg para o combate ao mosquito aedes aegypti. **SBC - Proceedings of XIX SBGames 2020**, 2020. ISSN 2179-2259.
- BATTAIOLA, A.; MUNHOZ, D. Regras e mecânicas em jogos. a, p. 22–41, 12 2018.
- BELLOTTI, F.; BERTA, R.; GLORIA, A. D.; PRIMAVERA, L. A task annotation model for sandbox serious games. In: . [S. l.: s. n.], 2009. p. 233–240.
- BEPPE, T. A.; ARAÚJO, I. Linhares de; ARAGÃO, B.; SANTOS, I. D. S.; XIMENES, D.; ANDRADE, R. M. C. Greatest: a card game to motivate the software testing learning. In: . [S. l.: s. n.], 2018. p. 298–307.
- BOARD Game Types, A Quick Classification. 2019. Disponível em: <https://tableknight.com/board-game-genres/>. Acesso em: 20 dezembro 2020.
- BORGES, B.; SILVA, R.; PAIVA, J.; ARAGÃO, B.; SANTOS, I. D. S.; ANDRADE, R. Design e avaliação de um aplicativo móvel complementar para um jogo de cartas educacional. In: . [S. l.: s. n.], 2019. p. 21–30.
- BROWN, J. S.; COLLINS, A.; DUGUID, P. Situated cognition and the culture of learning. **Educational researcher**, Sage Publications Sage CA: Thousand Oaks, CA, v. 18, n. 1, p. 32–42, 1989.
- BUSCHMANN, F.; HENNEY, K.; SCHMIDT, D. C. **Pattern-oriented software architecture, on patterns and pattern languages**. [S. l.]: John wiley & sons, 2007.
- CAMPOS, L. M. L.; BORTOLOTO, T. M.; FELÍCIO, A. K. C. *et al.* A produção de jogos didáticos para o ensino de ciências e biologia: uma proposta para favorecer a aprendizagem. **Caderno dos núcleos de Ensino**, São Paulo, v. 47, p. 47–60, 2003.
- CAMPOS, L. M. L.; BORTOLOTO, T. M.; FELÍCIO, A. K. C. *et al.* A produção de jogos didáticos para o ensino de ciências e biologia: uma proposta para favorecer a aprendizagem. **Caderno dos núcleos de Ensino**, São Paulo, v. 47, p. 47–60, 2003.
- CANTERI, R. dos P.; GARCÍA, L. S.; FELIPE, T. A. de; GALVÃO, L. F. O.; ANTUNES, D. R. Conceptual framework to support a web authoring tool of educational games for deaf children. In: **CSEU (2)**. [S. l.: s. n.], 2019. p. 226–235.
- CHOI, J.-I.; HANNAFIN, M. **Situated Cognition and Learning Environments: Roles, Structures, and Implications for Design**. 1995. 53–69 p. Disponível em: <https://www.learntechlib.org/p/164884>. Acesso em: 30 agosto 2022.

- COHN, M. **User stories applied: For agile software development**. [S. l.]: Addison-Wesley Professional, 2004.
- CSIKSZENTMIHALYI, M. Flow: The psychology of optimal experience. In: \_\_\_\_\_. [S. l.: s. n.], 1990.
- DARIN, T.; ROCHA, F.; MOTTA, D.; ANGELO, P. V. Desafio de design google: Um jogo de cartas para apoio ao ensino do design de interação e conceitos básicos de interação humano-computador. In: SBC. **Anais Estendidos do XVIII Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais**. [S. l.], 2019. p. 100–105.
- DEMASI, P.; ADRIANO, J. d. O. Online coevolution for action games. **Int. J. Intell. Games & Simulation**, v. 2, n. 2, p. 80–88, 2003.
- DEMPSEY, J.; LUCASSEN, B.; RASMUSSEN, K. **The Instructional Gaming Literature: Implications and 99 Sources**. University of South Carolina, College of Education, 1996. Disponível em: <https://books.google.com.br/books?id=RrArmwEACAAJ>.
- DEMPSEY, J.; RASMUSSEN, K.; LUCASSEN, B. Instructional gaming: Implications for instructional technology. In: . [S. l.: s. n.], 1996.
- DESPAIN, W. **100 Principles of Game Design**. [S. l.]: New Riders; Pearson Education. v. 2012.
- DUVAL, Y.; PANZOLI, D.; REYMONET, A.; PLANTEC, J.; THOMAS, J.; JESSEL, J.-P. Serious games scenario modeling for non-experts. In: **CSEDU**. [S. l.: s. n.], 2015.
- ENG, D. **Player Interaction**. University XP, 2019. Disponível em: <https://www.universityxp.com/blog/2019/9/17/player-interaction>. Acesso em: 03 outubro 2021.
- FENGFENG, K. Chapter i a qualitative meta-analysis of computer games as learning tools. **Handbook of Research on Effective Electronic Gaming in Education**, v. 1, 01 2009.
- FIGUEIREDO, K.; FERREIRA, J.; MURTA, L.; CLUA, E. Um jogo de estratégia de gerência de configuração. **III Fórum de Educação em Engenharia de Software**, p. 1–8, 2010.
- FRANÇA, E. L. de; NUNES, R. C.; SANTOS, R. C. dos. Jogo baralho das variáveis uma proposta de utilização de jogos para disciplina de programação. XXI Workshop sobre Educação em Informática (WEI 2013).
- FREITAS, S. de. **Learning in immersive worlds: a review of game-based learning**. [S. l.], 2006.
- GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J.; PATTERNS, D. **Elements of reusable object-oriented software**. [S. l.]: Addison-Wesley Reading, Massachusetts, 1995. v. 99.
- GARCÍA, C. G.; NÚÑEZ-VALDEZ, E. R.; MORENO-GER, P.; CRESPO, R. G.; G-BUSTELO, B. C. P.; LOVELLE, J. M. C. Agile development of multiplatform educational video games using a domain-specific language. In: . [S. l.: s. n.], 2019. p. 599–614.
- GOLDBERG, A.; ROBSON, D. **Smalltalk-80: the language and its implementation**. [S. l.]: Addison-Wesley Longman Publishing Co., Inc., 1983.

- GOLDBERG, L. R. The structure of phenotypic personality traits. In: . [S. l.: s. n.], 1993. p. 26–34.
- GREDLER, M. Games and simulations and their relationships to learning. **Handbook of Research on Educational Communications and Technology**, 01 2004.
- GUERREIRO, F.; MOURA, F. S.; RODRIGUES, L.; FONTOURA, M. M. Design case do jogo de cartas "gente". SBC - Proceedings of XIX SBGames 2020, 2020. ISSN 2179-2259.
- GULUMBAY, A.; TURKEY. Affective and emotional aspects of human-computer interaction: Game-based and innovative learning approaches. **The Turkish Online Journal of Distance Education**, 01 2006.
- HAHN, C. A domain specific modeling language for multiagent systems. In: CITeseer. **Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1**. [S. l.], 2008. p. 233–240.
- HAMIYE, F.; SAID, B.; SERHAN, B. A framework for the development of serious games for assessment. In: SPRINGER. **International Conference on Games and Learning Alliance**. [S. l.], 2019. p. 407–416.
- HIRDES, E. M.; LEIMEISTER, J. M. A modeling language to describe reusable learning processes to achieve educational objectives in serious games. In: SPRINGER. **European Conference on Technology Enhanced Learning**. [S. l.], 2013. p. 454–459.
- HOGLE, J. G. Considering games as cognitive tools: In search of effective "edutainment". In: . [S. l.: s. n.], 1996.
- HUNICKE, R.; LEBLANC, M.; ZUBEK, R. Mda: A formal approach to game design and game research. **AAAI Workshop - Technical Report**, v. 1, 01 2004.
- ISO. Iso/iec 25010:2011 systems and software engineering – systems and software quality requirements and evaluation (square) – system and software quality models. **System and software quality models**, ISO, p. 34, 2011–13.
- ISO/IEC/IEEE-24765. **Systems and software engineering — Vocabulary**. [S. l.]: ISO/IEC/IEEE 2017, 2017. Acesso em: 02 outubro 2023.
- KAHRAMAN, G.; BILGEN, S. A framework for qualitative assessment of domain-specific languages. **Software Systems Modeling**, v. 14, 01 2013.
- KRITZ EDUARDO MANGELI, G. X. J. Building an ontology of boardgame mechanics based on the boardgamegeek database and the mda framework. In: **XVI Brazilian Symposium on Computer Games and Digital Entertainment**. Curitiba: [S. n.], 2017. p. 182–191.
- LAZZARO, N. Why we play games: Four keys to more emotion without story. **Game Dev Conf**, 01 2004.
- LELLI, V.; ANDRADE, R. M.; FREITAS, L. M.; SILVA, R. A.; FILHO, F. G. S.; GOMES, R. F.; SEVERO, J. S. de O. Gamification in remote teaching of se courses: experience report. In: **Proceedings of the 34th Brazilian Symposium on Software Engineering**. [S. l.: s. n.], 2020. p. 844–853.

LIMA, G. d. O. Framework para construção de jogos de cartas colecionáveis com ra. Universidade Federal do Maranhão, 2018.

MALHEIROS, T.; REIS, L.; CARVALHO, J. E. R. de. A utilização da realidade aumentada em jogos de cartas colecionáveis. SBC - Proceedings of SBGames 2012, 2012.

MALONE, T. What makes computer games fun?(abstract only). In: **Conference on Human Factors in Computing Systems: Proceedings of the joint conference on Easier and more productive use of computer systems.(Part- II): Human interface and the user interface.** [S. l.: s. n.], 1981. v. 1981.

MARCHIORI, E. J.; BLANCO, Á. D.; TORRENTE, J.; MARTINEZ-ORTIZ, I.; FERNÁNDEZ-MANJÓN, B. A visual language for the creation of narrative educational games. **Journal of Visual Languages & Computing**, Elsevier, v. 22, n. 6, p. 443–452, 2011.

MATTAR, J.; ALMEIDA, F. D.; SOUZA, L. M.; BEDUSCHI, J. d. O.; SILVA, C. C. da; SANTOS, B. R. dos; AMARAL, J. W. R.; SPROVIERI, R. Gamificação e jogos para metodologia científica: proposta de jogo de tabuleiro e game. SBC - Proceedings of XIX SBGames 2017, 2017. ISSN 2179-2259.

MONSALVE, E.; WERNECK, V.; LEITE, J. Simules-w: Um jogo para o ensino de engenharia de software. **III Fórum em Educação de Engenharia de Software (FEES), Simpósio Brasileiro de Engenharia de Software (SBES), Salvador**, p. 17–26, 2010.

MORA-ZAMORA, R.; BRENES-VILLALOBOS, E. Integrated framework for game design. In: . [S. l.: s. n.], 2019. p. 1–6.

MOSSMANN, J. B.; RIEDER, R.; FLORES, C. D.; PINHO, M. Project and preliminary evaluation of vr-med, a domain-specific language for serious games in family medicine teaching. In: **IEEE. 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)**. [S. l.], 2016. v. 2, p. 663–667.

NIELSEN, J. Enhancing the explanatory power of usability heuristics. In: **Proceedings of the SIGCHI conference on Human Factors in Computing Systems**. [S. l.: s. n.], 1994. p. 152–158.

O'SHEA, Z.; FREEMAN, J. Game design frameworks: where do we start? In: . [S. l.: s. n.], 2019. p. 1–10. ISBN 978-1-4503-7217-6.

PETERSEN, K.; FELDT, R.; MUJTABA, S.; MATTSSON, M. Systematic mapping studies in software engineering. In: **12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12**. [S. l.: s. n.], 2008. p. 1–10.

PRASANNA, A. T. **A Domain Specific Modeling Language for Specifying Educational Games**. Tese (Doutorado) – Master's Thesis. Vrije Universiteit Brussel, Department of Computer Science, 2012.

PRENSKY, M. **Digital game-based learning**. [S. l.]: New York: McGraw-Hill, 2001.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software-9**. [S. l.]: McGraw Hill Brasil, 2021.

ROCHA, J. B.; MASCARENHAS, S.; PRADA, R. Game mechanics for cooperative games. **ZON Digital Games 2008**, Citeseer, p. 72–80, 2008.

- SALOMON, G.; PERKINS, D. N.; PERKINS, T. Partners in cognition: Extending human intelligence with intelligent technologies. **Educational Researcher**, v. 20, n. 3, p. 2–9, 1991. Disponível em: <https://doi.org/10.3102/0013189X020003002>.
- SANTOS, J. C. O.; FIGUEIREDO, K. d. S. Computasseia: Um jogo para o ensino de história da computação. In: SBC. **Anais do XXIV workshop sobre educação em computação**. [S. l.], 2016. p. 31–40.
- SARINHO, V. T. Uma proposta de game design canvas unificado. **XVI Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)**, 2017.
- SAUCIER, G.; GOLDBERG, L. R. Evidence for the big five in analyses of familiar english personality adjectives. In: . [S. l.: s. n.], 1996.
- SCHREIBER, I. **Game Balance Concepts - a continued experiment in game design and teaching**. 2010. Disponível em: <https://gamebalanceconcepts.wordpress.com>. Acesso em: 06 dezembro 2021.
- SILVA, R.; F., B. A.; FERREIRA, M. de F.; SANTOS, I.; ANDRADE, R. Evasão em computação na ufc sob a perspectiva dos alunos. In: **Anais do XXIX Workshop sobre Educação em Computação**. Porto Alegre, RS, Brasil: SBC, 2021. p. 338–347. ISSN 2595-6175. Disponível em: <https://sol.sbc.org.br/index.php/wei/article/view/15925>.
- SILVA, R. A. d. S. D-creea: Dsml for creating educational analog card games. **XVI Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)**, 2021.
- SOMMERVILLE, I. Software engineering 10th edition. **ISBN-10**, v. 137035152, p. 18, 2015.
- TANG, S.; HANNEGHAN, M.; HUGHES, T.; DENNETT, C.; COOPER, S.; SABRI, M. A. *et al.* Towards a domain specific modelling language for serious game design. In: **6th International Game Design and Technology Workshop, Liverpool, UK**. [S. l.: s. n.], 2008.
- THILLAINATHAN, N. A model driven development framework for serious games. **Available at SSRN 2475410**, 2013.
- TLILI, A.; ESSALMI, F.; AYED, L. J. B.; JEMNI, M.; KINSHUK. Towards a generic uml model to support designing educational role playing games. In: **2016 IEEE 16th International Conference on Advanced Learning Technologies (ICALT)**. [S. l.: s. n.], 2016. p. 153–157.
- TROYER, O. D.; BROECKHOVEN, F. V.; VLIEGHE, J. Linking serious game narratives with pedagogical theories and pedagogical design strategies. **Journal of Computing in Higher Education**, Springer, v. 29, n. 3, p. 549–573, 2017.
- VANDENBERGHE, J. **The 5 Domains of Play: Applying Psychology's Big 5 Motivation Domains to Games**. GDC Presentation Video (2012), 2012. Disponível em: <http://www.gdcvault.com/play/1015595/The-5-Domains-of-Play>. Acesso em: 24 março 2020.
- VANDENBERGHE, J. **Applying the 5 Domains of Play: Acting Like Players**. GDC Presentation Video (2013), 2013. Disponível em: <http://www.gdcvault.com/play/1017876/Applying-the-5-Domains-of>. Acesso em: 24 março 2020.
- VANDENBERGHE, J. **Engines of Play: How Player Motivation Changes Over Time**. GDC Presentation Video (2016), 2016. Disponível em: <http://www.gdcvault.com/play/1023329/Engines-of-Play-How-Player>. Acesso em: 24 março 2020.

VISSER, E. Webdsl: A case study in domain-specific language engineering. In: SPRINGER. **International summer school on generative and transformational techniques in software engineering**. [S. l.], 2007. p. 291–373.

WOHLIN, C.; HÖST, M.; HENNINGSSON, K. Empirical research methods in software engineering. In: **Empirical methods and studies in software engineering**. [S. l.]: Springer, 2003. p. 7–23.

XU, Y.; BARBA, E.; RADU, I.; GANDY, M.; MACINTYRE, B. Chores are fun: Understanding social play in board games for digital tabletop game design. **Proceedings of DiGRA 2011 Conference: Think Design Play**, 05 2012.

YENSEN, J. Pico search strategies. v. 17, 10 2013.

YUSOF, S. A. M.; RADZI, S. H. M.; DIN, S. N. S.; KHALID, N. A study on the effectiveness of task manager board game as a training tool in managing project. In: AIP PUBLISHING LLC. **AIP Conference Proceedings**. [S. l.], 2016. v. 1761, n. 1, p. 020074.

ZAHARI, A. S.; RAHIM, L. A.; MEHAT, M. A review of modelling languages for adventure educational games. **2016 3rd International Conference on Computer and Information Sciences (ICCOINS)**, p. 495–500, 2016.

ZAHARI, A. S.; RAHIM, L. A.; NURHADI, N.; ASLAM, M. A domain specific modeling language for adventure educational games and flow theory. **International Journal on Advanced Science Engineering and Information Technology**, v. 10, 06 2020.

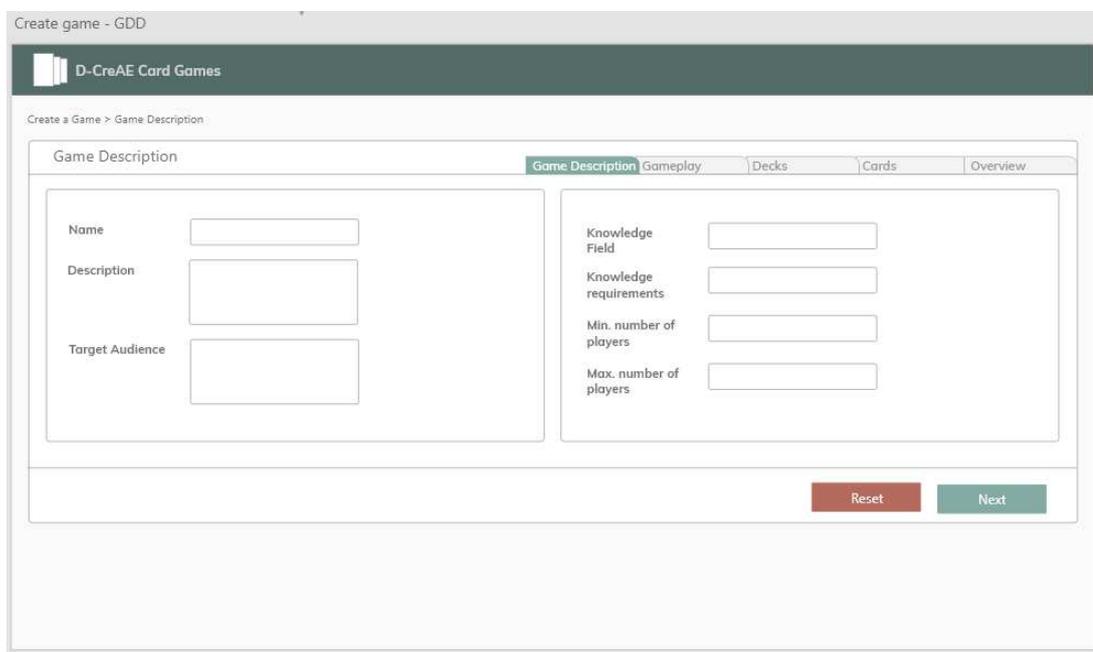
ZIATDINOV, M.; KHAFIZOV, M. New automata definition of language for game development. In: CRC PRESS. **Social Sciences and Interdisciplinary Behavior: The 4th International Congress on Interdisciplinary Behavior and Social Science (ICIBSoS 2015), Kazan Federal University, Kazan, Russia, 22-23 October 2015 & Arya Duta hotel, Jakarta, Indonesia, 07–08 November 2015**. [S. l.], 2016. p. 61.

## APÊNDICE A – PROTÓTIPO DA INTERFACE WEB DA D-CREEA

Neste apêndice são apresentadas as imagens do protótipo das telas em alta fidelidade da plataforma WEB D-CreEA.

Os protótipos foram desenvolvidos com a ferramenta Adobe XD na fase de requisitos e o design final das telas foi alterado e adaptado diretamente na plataforma ao longo do desenvolvimento. Os protótipos foram úteis para guiar o modo como as funcionalidades seriam apresentadas ao usuário logo na fase pré-implementação e, assim, auxiliar a análise de requisitos.

Nas figuras a seguir são mostradas essas imagens dos protótipos de telas da D-CreEA.



The image shows a high-fidelity prototype of the 'Game Description' screen in the D-CreEA system. The interface is titled 'Create game - GDD' and 'D-CreAE Card Games'. It features a breadcrumb trail 'Create a Game > Game Description' and a navigation menu with tabs for 'Game Description', 'Gameplay', 'Decks', 'Cards', and 'Overview'. The main content area is divided into two columns of input fields. The left column contains 'Name', 'Description', and 'Target Audience'. The right column contains 'Knowledge Field', 'Knowledge requirements', 'Min. number of players', and 'Max. number of players'. At the bottom right, there are 'Reset' and 'Next' buttons.

Figura 26 – Protótipo de alta fidelidade da tela de GDD da D-CreEA

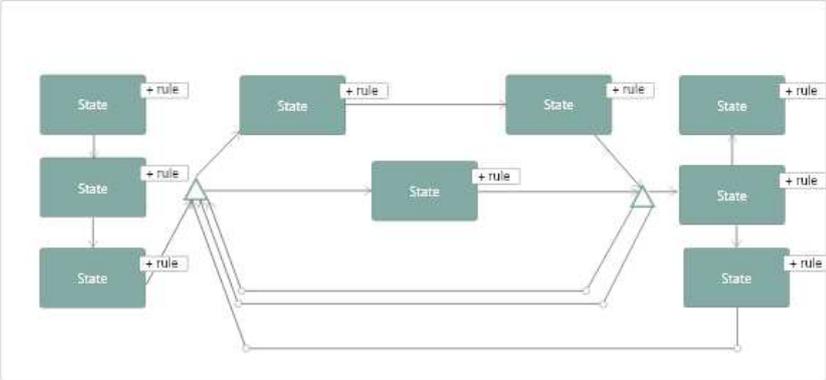
Create game - Conditional Rule Creation

D-CreAE Card Games

Create a Game > Gameplay

Gameplay Edition

Game Description | **Gameplay** | Decks | Cards | Overview



State X Save State

Label for this state

Rules from this state

Title	Cost	Benefit	Challenge	Description			
Card Name	2	2	2	4	low	The longest description...	<input type="button" value="O"/>
Card Name	2	2	2	4	low	The longest description...	<input type="button" value="O"/>

Search

**Conditional Rule** | Effect Label | Statement Rule | Transition Rule

Conditional Rule

Label for this Rule

Conditions Add Condition

IF regular text	DO my effect	GO TO State y	<input type="button" value="O"/>
IF regular text	DO my effect	GO TO State y	<input type="button" value="O"/>
IF regular text	DD my effect	GO TO State y	<input type="button" value="O"/>

Otherwise go to State

Save Rule

Details

Item Details Edit

Label: my rule

Game Description Edit

Label: my rule

Figura 27 – Protótipo de alta fidelidade da tela de regras da D-CreEA

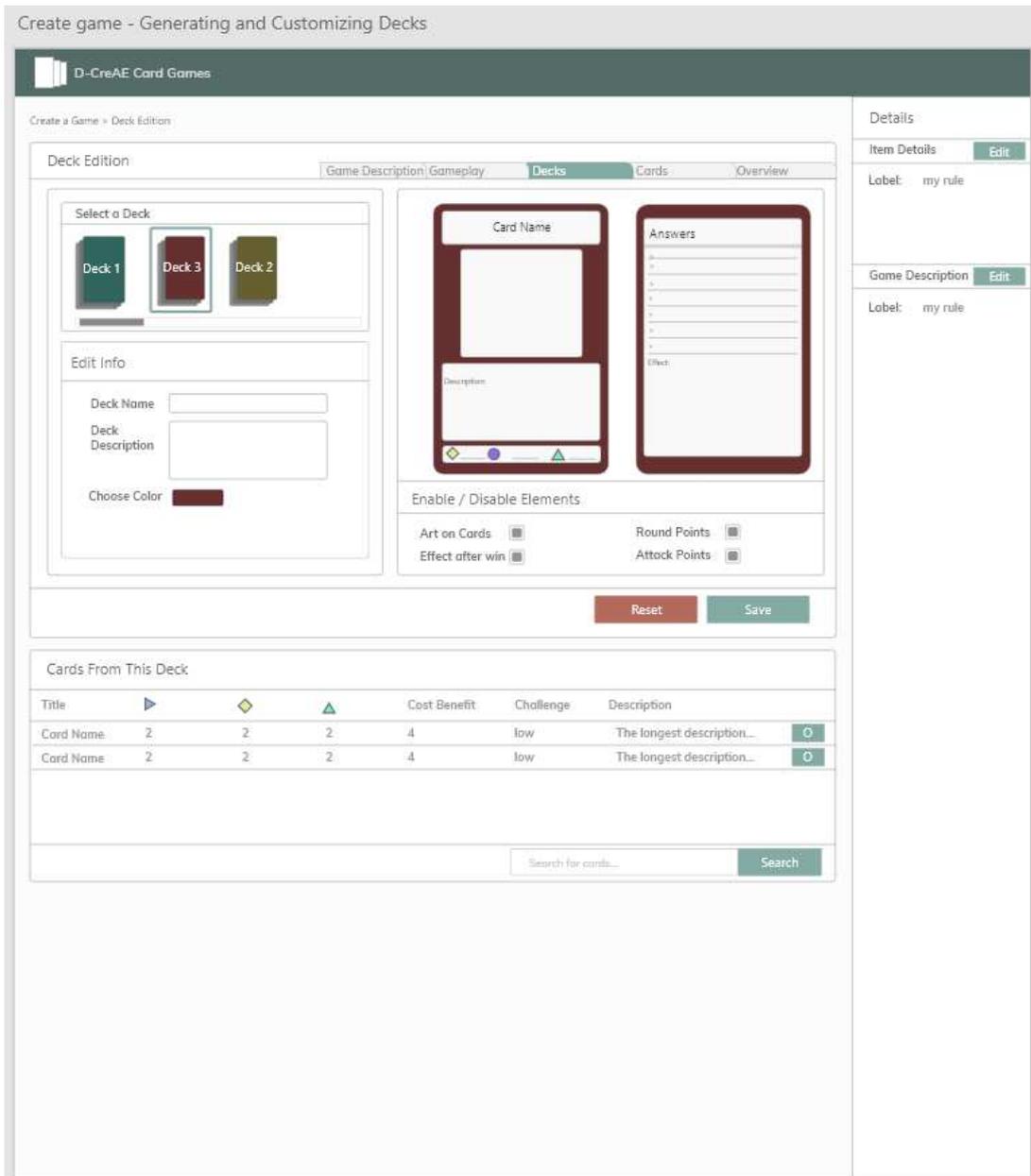


Figura 28 – Protótipo de alta fidelidade da tela de criação de baralhos da D-CreEA

Create game - Generating and Customizing Cards [challenge deck]

D-CreAE Card Games

Create a Game > Cards Edition

Cards Edition | Game Description | Gameplay | Decks | **Cards** | Overview

Cost Benefit x Attack - Cards from Deck 1

Cost Benefit x Turn - Cards from Deck 1

Challenge level:

Card Name:   
 Description:   
 Answers:   
  
  
  
  
  
  
 Effect:

Select a Deck

Deck 1 | Deck 3 | Deck 2

Reset Save

Cards From This Deck

Title				Cost Benefit	Challenge	Description
Card Name	2	2	2	4	low	The longest description...
Card Name	2	2	2	4	low	The longest description...

Search for cards... Search

Details

Item Details [Edit](#)

Label: my rule

Game Description [Edit](#)

Label: my rule

Figura 29 – Protótipo de alta fidelidade da tela de criação de cartas da D-CreEA

## APÊNDICE B – QUESTIONÁRIO PRÉ-USO DA D-CREEA

**Questão 1.** Qual seu nível de familiaridade em uso de jogos educacionais? (ex: aplicação em aula, treinamento, etc.)

**Questão 2.** Qual seu nível de familiaridade em design de jogos?

**Questão 3.** Você deve elencar quais características de software (qualidade/requisitos não-funcionais) você espera que estejam presentes (em nível satisfatório) no D-CreEA. Assim, no próximo passo, ao testar a ferramenta, você vai avaliar qual o nível de satisfação de tais características. Para consultar as características e definições, consulte: <https://miro.com/app/board/o9JlxqVY7U> = /

- (a) usabilidade
- (b) confiabilidade
- (c) adequação funcional
- (d) extensividade
- (e) produtividade
- (f) expressividade
- (g) compatibilidade
- (h) reusabilidade
- (i) integrabilidade
- (j) manutenibilidade

## APÊNDICE C – QUESTIONÁRIO PÓS-USO DA D-CREEA

**Questão 1.** Para cada característica de software marcada no questionário anterior, atribua nota que reflete o nível de adequação da ferramenta com a característica

- (a) usabilidade (muito ruim a muito bom)
- (b) confiabilidade (muito ruim a muito bom)
- (c) adequação funcional (muito ruim a muito bom)
- (d) extensividade (muito ruim a muito bom)
- (e) produtividade (muito ruim a muito bom)
- (f) expressividade (muito ruim a muito bom)
- (g) compatibilidade (muito ruim a muito bom)
- (h) reusabilidade (muito ruim a muito bom)
- (i) integrabilidade (muito ruim a muito bom)
- (j) manutenibilidade (muito ruim a muito bom)

**Questão 2.** Deixe sua opinião sobre a D-CreEA Card Games

**Questão 3.** Deixe sugestões para a evolução/melhoria da D-CreEA Card Games