



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CAMPUS DE RUSSAS**  
**CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE**

**LUIS OTÁVIO LIMA CAMINHA**

**D-LEARN: UM JOGO DIGITAL BASEADO NO JOGO DE TABULEIRO LEARN PARA  
ENSINO DE ARQUITETURA DE SOFTWARE**

**RUSSAS**

**2023**

LUIS OTÁVIO LIMA CAMINHA

D-LEARN: UM JOGO DIGITAL BASEADO NO JOGO DE TABULEIRO LEARN PARA  
ENSINO DE ARQUITETURA DE SOFTWARE

Trabalho de Conclusão de Curso apresentado ao  
Curso de Graduação em Engenharia de Software  
da Universidade Federal do Ceará, como  
requisito parcial à obtenção do grau de bacharel  
em Engenharia de Software.

Orientadora: Profa. Dra. Anna Beatriz  
dos Santos Marques.

RUSSAS

2023

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)

---

C191d Caminha, Luis Otávio Lima.  
D-LEARN: UM JOGO DIGITAL BASEADO NO JOGO DE TABULEIRO LEARN PARA ENSINO  
DE ARQUITETURA DE SOFTWARE RUSSAS 2023 / Luis Otávio Lima Caminha. – 2023.  
72 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas,  
Curso de Ciência da Computação, Russas, 2023.  
Orientação: Profa. Dra. Anna Beatriz dos Santos Marques.

1. arquitetura de software. 2. aprendizagem baseada em jogos. 3. LEARN Board Game. I. Título.  
CDD 005

---

LUIS OTÁVIO LIMA CAMINHA

D-LEARN: UM JOGO DIGITAL BASEADO NO JOGO DE TABULEIRO LEARN PARA  
ENSINO DE ARQUITETURA DE SOFTWARE

Trabalho de Conclusão de Curso apresentado ao  
Curso de Graduação em Engenharia de Software  
da Universidade Federal do Ceará, como  
requisito parcial à obtenção do grau de bacharel  
em Engenharia de Software.

Aprovada em: 04/07/2023.

BANCA EXAMINADORA

---

Profa. Dra. Anna Beatriz dos Santos  
Marques (Orientadora)  
Universidade Federal do Ceará (UFC)

---

Prof. Ms. José Osvaldo Mesquita Chaves  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Williamson Alison Freitas Silva  
Universidade Federal do Pampa (UNIPAMPA)

A Deus.

Aos meus familiares.

## AGRADECIMENTOS

Em primeiro lugar, gostaria de agradecer a Deus por ter me sustentado até aqui durante toda essa caminhada e por ter me dado forças para levantar quando eu caí. Obrigado pelas pedras no caminho e por me fazer mais forte em cada uma delas, hoje entendo que sem elas eu não estaria aqui e entendo que sem o senhor eu não conseguiria lidar com elas. Obrigado por nunca me abandonar.

A minha mãe Maria Zuleide Ferreira de Lima por ter depositado em mim sua confiança e por ter cuidado de mim esse tempo todo, mesmo com todas as nossas dificuldades. Obrigado por ter segurado a barra desde o momento que nasci, a senhora foi e é fundamental pra mim.

A minha irmã Anna Ravenna Lima Caminha que esteve ao meu lado desde pequena, compartilhando nossa infância, juventude e agora vida adulta. Foi muito bom dividir todas essas fases com você, e vai continuar sendo. Ao meu irmão, Duylho Ramon de Lima Costa, por ser meu amigo e irmão e por ter me ensinado muitas das coisas que sei hoje.

Em memória dos meus falecidos avós paternos, Maria José Caminha e Raimundo Lopes Caminha, gostaria de agradecer por terem feito parte da minha infância e por terem me ensinado sobre trabalho e esforço desde pequeno, e minha falecida avó materna, Senhorinha Trajano de Sousa, gostaria de agradecer por ter acreditado que minha mãe iria lhe dar muitas alegrias, e com nossa chegada, sentiu que as alegrias também viriam de nós.

Em memória do meu falecido padrasto, José Leudo Costa da Silva, que foi companheiro da minha mãe por muito tempo e que foi uma figura paterna na nossa infância e juventude, gostaria de agradecer por todo cuidado, momentos felizes e conselhos.

A minha atual companheira, que me apoiou durante a escrita deste trabalho e que vem sendo uma das melhores pessoas que Deus colocou na minha vida. Você tornou o processo bem mais leve e é muito bom dividir os dias com você.

Ao meu cachorro Abraão De La Vega, que foi meu companheiro de todos os dias e em muitas noites de estudo. Sua chegada em nossas vidas nos deu forças e nos ensinou sobre o amor verdadeiro. A todos os meus familiares (tias, tios, primos e primas) e amigos que de alguma maneira me ajudaram nessa caminhada, com palavras de força e incentivo.

A Profa. Dra. Anna Beatriz dos Santos Marques, pela excelente orientação e por ter me guiado nessa jornada.

A banca avaliadora que aceitou participar da avaliação deste trabalho.

Por fim, gostaria de agradecer a todos que de alguma maneira me impulsionaram a chegar até aqui, seja por bem ou por mal.

“Não fui eu que ordenei a você? Seja forte e corajoso! Não se apavore nem desanime, pois o Senhor, o seu Deus, estará com você por onde você andar.” (Josué 1:9)

## RESUMO

A utilização de jogos para o ensino vem sendo cada vez mais usado visando o aprendizado lúdico, interativo e menos massivo para os discentes. Essa abordagem é chamada de game-based learning. No âmbito da Engenharia de Software, vários jogos podem ser encontrados com o propósito de tentar ajudar discentes a entenderem melhor situações do mundo real, impacto de suas decisões e trabalho em equipe, por meio de um ambiente simulado. Tais jogos utilizam ferramentas como cartas, tabuleiro, ou interfaces digitais para promover uma dinâmica de jogo. No tocante ao ensino de decisões arquiteturais de software, estes jogos visam fornecer uma abordagem mais familiar aos discentes do século XXI - que já nasceram na era digital e que tem grande familiaridade com jogos - tentando fazer com que os discentes abstraíam melhor a grande quantidade de conceitos necessários para um profissional arquiteto de software. Por esse prisma, jogos como o LEARN Board Game, que é um jogo não-digital que visa auxiliar no ensino da disciplina de arquitetura de software por meio de um jogo de cartas e tabuleiro, tem papel importante na lista de ferramentas utilizadas por docentes de arquitetura de software para tentar resolver os problemas de abstração de conteúdos. No entanto, o LEARN tem suas limitações por ser um jogo não-digital. Como exemplo, pode-se citar a maneira não ecológica que o jogo é confeccionado (por meio de papel e plástico) e as dificuldades para fazer alterações simples como alterar o texto de uma carta ou alterar o tabuleiro. Logo, mesmo sendo uma ferramenta que ajuda os docentes, faz com que seu manuseio não seja tão satisfatório para quem o usa. Com isso, o objetivo deste trabalho foi construir uma versão digital do LEARN, denominada D-LEARN, melhorando seus pontos negativos e maximizando seus pontos positivos, através da metodologia de pesquisa e desenvolvimento Design Science Research (DSR), que propõe ciclos de atividades para construção ou atualização de um artefato que visa resolver um problema do mundo real com uma abordagem pragmática e satisfatória. O jogo digital obteve uma nota de qualidade considerada excelente segundo o cálculo sugerido pelo questionário MEEGA+, superando a nota de qualidade do jogo não-digital.

**Palavras-chave:** arquitetura de software; aprendizagem baseada em jogos; LEARN Board Game.

## ABSTRACT

The use of games for teaching has been increasingly used to make learning fun, interactive, and less massive for the students. This approach is called game-based learning. In the field of Software Engineering, several games can be found with the purpose of trying to help students better understand real-world situations, the impact of their decisions, and teamwork, through a simulated environment. Such games use tools such as cards, board, or digital interfaces to promote a game-like dynamic. When it comes to teaching software architectural decisions, these games aim to provide a more familiar approach to 21st century students - who were born in the digital age and are very familiar with games - trying to make students better abstract the large amount of concepts needed to a professional software architect. From this perspective, games such as LEARN Board Game, which is a non-digital game that aims to help teaching software architecture through a board and card game, has an important role in the list of tools used by software architecture professors to try to solve abstract problems. However, LEARN has its limitations because it is a non-digital game. As an example, we can mention the non-ecological way the game is made (using paper and plastic) and the difficulties to make simple changes such as changing the text of a card or changing the board. Therefore, even though it is a tool that helps teachers, its handling is not so satisfactory for those who use it. Thus, the objective of this work was to build a digital version of LEARN, called D-LEARN, improving its negative points and maximizing its positive points, through the methodology of research and development Design Science Research (DSR), which proposes cycles of activities for building or updating an artifact that aims to solve a real world problem with a pragmatic and satisfactory approach. The digital game obtained a quality score considered excellent according to the calculation suggested by the MEEGA+ questionnaire, surpassing the quality score of the non-digital game.

**Keywords:** software architecture; game-based learning; LEARN board game.

## LISTA DE FIGURAS

Figura 1 – Exemplo da pirâmide de William Glasser . . . . .	20
Figura 2 – Tabuleiro do jogo LEARN . . . . .	22
Figura 3 – Demonstração do fluxo dos papéis do jogo . . . . .	23
Figura 4 – Exemplo de carta (frente e verso): Atributo de qualidade . . . . .	23
Figura 5 – Arquitetura do D-LEARN . . . . .	40
Figura 6 – Protótipo da Modal de Configuração do D-LEARN . . . . .	42
Figura 7 – Protótipo do tabuleiro e pilha de cartas do D-LEARN . . . . .	42
Figura 8 – Página de Login do painel do D-LEARN . . . . .	44
Figura 9 – Página de CRUD de cartas do painel do D-LEARN . . . . .	44
Figura 10 – Link para o D-LEARN na Landing . . . . .	47
Figura 11 – Modal de escolha do modo do jogo . . . . .	47
Figura 12 – Modal de configuração do jogo . . . . .	48
Figura 13 – Início do jogo . . . . .	49
Figura 14 – Carta selecionada no jogo . . . . .	49
Figura 15 – Carta respondida errada . . . . .	50
Figura 16 – Carta respondida certa . . . . .	50
Figura 17 – Carta de sorte ou revés . . . . .	51
Figura 18 – Modal de fim de jogo . . . . .	51
Figura 19 – Página de CRUD de cartas do painel do D-LEARN . . . . .	52
Figura 20 – Página de listagem de jogos . . . . .	52
Figura 21 – Modal de criação de carta . . . . .	52
Figura 22 – Aplicação do D-LEARN em sala de aula . . . . .	53
Figura 23 – Pergunta sobre frequência em jogos digitais . . . . .	55
Figura 24 – Respostas sobre frequência de uso de jogos não-digitais . . . . .	55
Figura 25 – Respostas sobre experiência em jogos de tabuleiro voltados para o ensino . . . . .	56
Figura 26 – Respostas sobre experiência em jogos de tabuleiro voltados para o ensino . . . . .	57
Figura 27 – Dados sobre a experiência de usuário do D-LEARN . . . . .	59
Figura 28 – Dados sobre o aprendizado de arquitetura de software após jogar o jogo . . . . .	61

## LISTA DE TABELAS

Tabela 1 – Comparação do D-LEARN com os trabalhos relacionados . . . . .	29
Tabela 2 – Requisitos funcionais do projeto . . . . .	35
Tabela 3 – Requisitos não funcionais do projeto . . . . .	38
Tabela 4 – Regras de negócio do projeto . . . . .	39
Tabela 5 – Fatores de qualidade do MEEGA+. . . . .	54
Tabela 6 – Fatores analisados no formulário MEEGA+ . . . . .	62

## **LISTA DE ABREVIATURAS E SIGLAS**

AWS	Amazon Web Services
DSR	Design Science Research
GCP	Google Cloud Platform
LGPD	Lei Geral de Proteção de Dados Pessoais

## SUMÁRIO

1	<b>INTRODUÇÃO</b>	14
2	<b>OBJETIVOS</b>	17
2.1	Objetivo geral	17
2.2	Objetivos específicos	17
3	<b>FUNDAMENTAÇÃO TEÓRICA</b>	18
3.1	Arquitetura de software	18
3.2	Aprendizagem baseada em jogos	19
3.3	<i>LEARN Board Game</i>	22
3.4	<i>Design Science Research</i>	24
4	<b>TRABALHOS RELACIONADOS</b>	26
5	<b>PROCEDIMENTOS METODOLÓGICOS</b>	30
5.1	Investigação do problema	30
5.1.1	<i>Revisão da literatura</i>	30
5.1.2	<i>Aplicação e observação da adoção do LEARN em sala de aula e coleta de dados</i>	31
5.1.3	<i>Entrevista com a professora responsável pela disciplina e aplicação do LEARN</i>	31
5.1.4	<i>Definição do problema central e questões de pesquisa</i>	32
5.1.5	<i>Definição da classe de problema</i>	32
5.1.6	<i>Definição dos objetivos</i>	32
5.2	Design e Desenvolvimento	33
5.2.1	<i>Definição dos requisitos</i>	33
5.2.2	<i>Definição da arquitetura do D-LEARN</i>	33
5.2.3	<i>Criação do Design e Desenvolvimento do jogo</i>	33
5.3	Validação	33
6	<b>PROCESSO DE CRIAÇÃO DO D-LEARN</b>	35
6.1	Requisitos	35
6.1.1	<i>Requisitos Funcionais</i>	35
6.1.2	<i>Requisitos Não-funcionais</i>	38
6.1.3	<i>Regras de Negócio</i>	39

<b>6.2</b>	<b><i>Design da Arquitetura</i></b> . . . . .	<b>40</b>
<b>6.2.1</b>	<b><i>Definição do ambiente</i></b> . . . . .	<b>41</b>
<b>6.2.2</b>	<b><i>Definição de tecnologias</i></b> . . . . .	<b>41</b>
<b>6.3</b>	<b>Design da interface do usuário</b> . . . . .	<b>41</b>
<b>6.4</b>	<b>Desenvolvimento</b> . . . . .	<b>43</b>
<b>7</b>	<b>APRESENTANDO O D-LEARN</b> . . . . .	<b>46</b>
<b>7.1</b>	<b>Sobre o jogo</b> . . . . .	<b>46</b>
<b>7.2</b>	<b>Iniciando e entendendo o jogo</b> . . . . .	<b>47</b>
<b>7.3</b>	<b>Painel administrativo do D-LEARN</b> . . . . .	<b>50</b>
<b>8</b>	<b>AVALIAÇÃO DO D-LEARN</b> . . . . .	<b>53</b>
<b>8.1</b>	<b>Planejamento e condução da avaliação</b> . . . . .	<b>53</b>
<b>8.2</b>	<b>Informações demográficas dos estudantes</b> . . . . .	<b>55</b>
<b>8.3</b>	<b>Informações sobre a usabilidade do D-LEARN</b> . . . . .	<b>56</b>
<b>8.4</b>	<b>Informações sobre experiência de usuário do D-LEARN</b> . . . . .	<b>58</b>
<b>8.5</b>	<b>Percepção de aprendizagem sobre Arquitetura de Software</b> . . . . .	<b>61</b>
<b>8.6</b>	<b>Respostas abertas do questionário</b> . . . . .	<b>61</b>
<b>8.7</b>	<b>Nota de qualidade do D-LEARN segundo o cálculo do MEEGA+</b> . . . . .	<b>63</b>
<b>8.8</b>	<b>Validação com a professora da disciplina</b> . . . . .	<b>63</b>
<b>8.9</b>	<b>Desenvolvimento das melhorias</b> . . . . .	<b>63</b>
<b>9</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS</b> . . . . .	<b>65</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>68</b>
	<b>APÊNDICE A –PERGUNTAS DO QUESTIONÁRIO MEEGA+</b> . . . . .	<b>70</b>

## 1 INTRODUÇÃO

Arquitetura de *software* é uma área da Engenharia de *Software* que engloba as definições para construção de um *software*, tais como definições de componentes internos e externos, sua comunicação com outros *softwares*, métricas de qualidade como escalabilidade, segurança e eficiência, entre outros aspectos (MEDVIDOVIC; TAYLOR, 2010).

Os conceitos de arquitetura de *software* são conceitos relevantes para a indústria de *software*. Decisões relacionadas à performance, segurança, disponibilidade, testabilidade, eficiência entre outras características de qualidade, são concebidas por um arquiteto durante a modelagem de um projeto de *software*. Por esse prisma, o domínio do arquiteto nesses conceitos e suas decisões arquitetônicas implica diretamente na vida útil do *software*, sua manutenção e a aprovação dos *stakeholders*.

Em relação aos profissionais de arquitetura de *software*, tomar boas decisões no momento de modelar sistemas, entender as necessidades dos *stakeholders*, sugerir melhorias e ser um pragmático para fazer o melhor com que se tem em mãos são algumas das principais habilidades que um arquiteto de *software* necessita (BASS, 2018). Sendo assim, as habilidades interpessoais do arquiteto são imprescindíveis para que o ciclo de vida do projeto e do *software* ocorram da melhor maneira possível, pois o arquiteto se relaciona com praticamente todos os *stakeholders* do projeto a fim de coletar o máximo de informações possíveis para tomar decisões concisas na arquitetura que será proposta (MEDVIDOVIC; TAYLOR, 2010).

Nesse sentido fica claro que ainda no seu período de formação, o profissional que virá a ser arquiteto deve conseguir aproveitamento máximo nas disciplinas relacionadas à arquitetura de *software* para conseguir desempenhar um papel satisfatório na indústria. Contudo, estudantes têm um certo grau de dificuldade na resolução de problemas com vários caminhos para uma solução, que é o caso da criação de um *design* para uma arquitetura de *software*, pois não há uma única forma de ser construído. Paralelo a isso, as formas de captação de saber dessa área utilizam poucos meios lúdicos, sendo em grande parte adquirida em cursos de universidade, que por sua vez tem o formato de aula expositiva como padrão (OLIVEIRA *et al.*, 2022). Este formato de aula expositiva, adotado desde o século XIX, tem foco na memorização e avaliação por meio de provas teóricas e têm se tornado cada vez mais difícil de adoção por parte dos estudantes. Problemas como falta de atenção, sentido e propósito são alguns dos problemas relativamente comuns nos estudantes do século atual (HEMMERT *et al.*, 2021).

Com isso, surgem algumas ferramentas de auxílio à docência para tentar ajudar

os docentes a transmitirem os conteúdos de maneira mais diversificada e até mesmo divertida, fazendo com que os discentes tenham um meio a mais para aprender e mais familiar para abstrair conhecimentos. Jogos de tabuleiro com uso de cartas são exemplos de jogos que auxiliam no aprendizado. Tais jogos visam auxiliar nas aulas por meio de atividades dinâmicas, promovendo interação entre os estudantes e fomentando o aprendizado diversificado. Como exemplo de um jogo não-digital com uso de cartas e tabuleiro, pode-se citar o *LEARN(LEarning software ARchitecture fundameNtals) Board Game*, que tem o propósito de transmitir de forma lúdica os conteúdos de arquitetura de *software* (SOUSA; MARQUES, 2020). Nesses tipos de jogos, os jogadores podem se reunir em grupos (promovendo a interação) e por meio de um tabuleiro podem realizar a atividade do jogo, onde cada um terá sua vez e poderão compartilhar com os outros companheiros suas decisões em cada etapa, permitindo assim o compartilhamento de conhecimento.

Jogos que trazem o propósito de aprendizado lúdico têm em maioria um *feedback* positivo por parte dos seus participantes e um indicador também positivo em relação à abstração dos conteúdos de maneira mais satisfatória do que a maneira tradicional de repasse de conteúdo entre docente e discente. Pode-se verificar tais resultados com as análises dos jogos não-digitais voltados para áreas da engenharia de *software* como *Problems and Programmers* e de jogos digitais como o SimSE. Em relação a jogos não-digitais para ensino focado em arquitetura de *software* pode-se citar os resultados do *LEARN, Smart Decisions e DecidArch*.

Contudo, jogos não-digitais como LEARN trazem consigo dificuldades de caráter físico que dificultam sua aplicação, como confecção e alteração. Ainda sobre o LEARN, atualmente é muito difícil confeccionar uma nova cópia do jogo, já que é necessário um processo de impressão do tabuleiro e compra dos pinos. Além disso, para fazer pequenas alterações no tabuleiro, seria necessário confeccionar um novo. Alterações em cartas também são um problema, tendo em vista que ao alterar a carta de uma unidade do jogo, todas as outras unidades teriam que ter essa carta alterada. Aspectos do ambiente como vento e luminosidade também podem ser um problema.

Logo, esse trabalho tem como objetivo a construção de uma versão digital do LEARN, denominado D-LEARN: *Digital LEarning software ARchitecture fundameNtals*, que visa resolver alguns desafios enfrentados pelo atual jogo em formato não-digital, tais como uso de papel para sua construção, dificuldade para fazer alterações na dinâmica do jogo e uma alta quantidade de tempo/esforço para confeccionar uma nova versão com novas regras, por exemplo.

A metodologia adotada para a pesquisa foi o *Design Science Research (DSR)*, baseado nas etapas de um ciclo de DSR proposto por Wieringa (2010), dividida em 3 etapas: Investigação do problema (etapa 1), design de uma solução (etapa 2) e validação da solução (etapa 3).

A importância deste projeto de pesquisa se dá além da continuação do trabalho feito no LEARN, que já tem muitos feedbacks positivos, na resolução dos problemas citados anteriormente e possíveis implementações de novas funcionalidades, fazendo com que os pontos positivos do LEARN sejam maximizados, e os pontos negativos sejam minimizados e/ou resolvidos. Portanto, o D-LEARN visa, objetivamente, melhorar o projeto atual por meio de uma versão digital, para amplo uso por docentes e auxílio no ensino de arquitetura de *software*.

A fim de apresentar o trabalho, o Capítulo 2 trata dos objetivos gerais e específicos, o Capítulo 3 disserta a fundamentação teórica utilizada e seus conceitos chaves. No Capítulo 4, os trabalhos relacionados são trazidos com intuito de embasar sobre a importância do projeto de pesquisa, trabalhos semelhantes e problemas presenciados, seguido do Capítulo 5 que detalha os procedimentos metodológicos utilizados para construção do artigo. O Capítulo 6 descreve o processo de criação do D-LEARN. O Capítulo 7 apresenta o jogo e por fim, o Capítulo 8 expõe os resultados de acordo com *feedbacks* coletados após aplicação do jogo em sala de aula.

## 2 OBJETIVOS

Neste capítulo, são delineados os objetivos gerais e específicos que guiarão o desdobramento da pesquisa subsequente. O propósito do objetivo geral é indicar o resultado almejado com a pesquisa, ao passo que os objetivos específicos representam os passos intermediários a serem seguidos para atingir o objetivo geral.

### 2.1 Objetivo geral

Auxiliar docentes e discentes no processo de ensino de arquitetura de software, por meio do desenvolvimento de uma versão digital do LEARN, o D-LEARN (*Digital LEarning software ARchitecture fundameNtals*).

### 2.2 Objetivos específicos

- Apoiar docentes de arquitetura de software por meio de uma versão digital do LEARN, com mais portabilidade e facilidade de aplicação.
- Melhorar o nível de satisfação e aprendizado do jogo não-digital LEARN, por meio de uma análise da nota de qualidade do jogo.
- Fazer com que o D-LEARN seja uma ferramenta auxiliar no ensino de arquitetura de software.

### 3 FUNDAMENTAÇÃO TEÓRICA

Para construção desse projeto de pesquisa, alguns conceitos fundamentais foram estudados e utilizados. O primeiro deles foi o conceito de arquitetura de *software*, seus problemas e principais aplicações (MEDVIDOVIC; TAYLOR, 2010; BASS, 2018). Além disso, também foi utilizado o conceito de *game-based learning*, a fim de desenvolver uma solução com alto grau de aprendizado no artefato criado (PINTO; SILVA, 2017; SOUSA; MARQUES, 2020; JABBAR; FELICIA, 2015). Como o D-LEARN será baseado no LEARN, este também foi apresentado como fundamentação teórica. Por último, o DSR, que foi a metodologia de pesquisa escolhida para este trabalho (HEVNER; CHATTERJEE, 2010; WIERINGA, 2010).

#### 3.1 Arquitetura de software

No centro de cada bom software e de cada série de boas decisões de design tomadas, há sempre uma boa arquitetura por trás (MEDVIDOVIC; TAYLOR, 2010). A arquitetura de software, é uma área da engenharia de *software* que não trata somente de um diagrama simples para o desenvolvimento e *deploy* de um *software*, mas sim de vários componentes interligados e com atributos próprios, se comunicando com outros *softwares* por meio de interfaces para fornecer um bom uso para o usuário final. Pode-se dizer que, o sucesso de um *software* depende de uma boa arquitetura.

Além de características funcionais como disponibilidade, segurança, compatibilidade, portabilidade, eficiência e integridade, também existem as características não-funcionais, que englobam os atributos de qualidade, que podem ser definidos seguindo a ISO/IEC 25010 (STANDARDIZATION, 2011) e que tangem de maneira maior o usuário final, por envolver aspectos como usabilidade, interface e experiência do usuário. Logo, montar uma arquitetura de um *software* envolve várias decisões arquitetônicas que devem ser tomadas em conjunto com os *stakeholders* do projeto, a fim de desenvolver a melhor solução para o problema pelo qual a arquitetura está sendo concebida.

Ademais, em relação aos profissionais de arquitetura, pode-se citar algumas vertentes: profissionais que trabalham de maneira mais focada na criação e documentação da arquitetura e profissionais que criam, implementam e atualizam a arquitetura criada, seja na nuvem utilizando alguma plataforma de *cloud* como *Amazon Web Services (AWS)* e *Google Cloud Platform (GCP)* ou em servidores comuns. Tais profissionais são chamados de *DevOps* e cuidam de outras tarefas

como provisionamento de ambientes, *deploy* de artefatos, construção de pipelines de integração contínua, monitoramento dos servidores, resolução de problemas em produção entre outras, a fim de tornar o processo de ciclo de vida do software mais fluído e automático, melhorando assim os processos de desenvolvimento, entrega e manutenção (BASS, 2018). Por esse prisma, pode-se definir um profissional *DevOps* como um profissional com sólidos conhecimentos em arquitetura de *software*, desenvolvimento e infraestrutura de *software* e que participa de todo ciclo de vida do *software* para garantir o bom andamento da infraestrutura e arquitetura pré-estabelecida.

Torna-se claro que, a arquitetura de *software* de um sistema é feita por uma série de decisões que impactam diretamente o usuário final e a vida dos funcionários que irão eventualmente trabalhar com o *software* desenvolvido, já que o *software* será desenvolvido com base na arquitetura criada no início do projeto, logo, uma boa arquitetura evitará alterações desnecessárias o projeto e surpresas na implantação e manutenção da mesma.

Portanto, nos dias de hoje com serviços como *on-demand* e *streaming*, disponibilidade próxima à 100%, milhões de usuários simultâneos em plataformas ao mesmo tempo e leis que proíbem qualquer vazamento de dados, como a Lei Geral de Proteção de Dados Pessoais (LGPD) (Presidência da República, 2018), é endossado mais ainda que bons conjuntos de decisões arquiteturais são fundamentais para o sucesso de um *software* tanto em sua entrega como em sua manutenção.

### **3.2 Aprendizagem baseada em jogos**

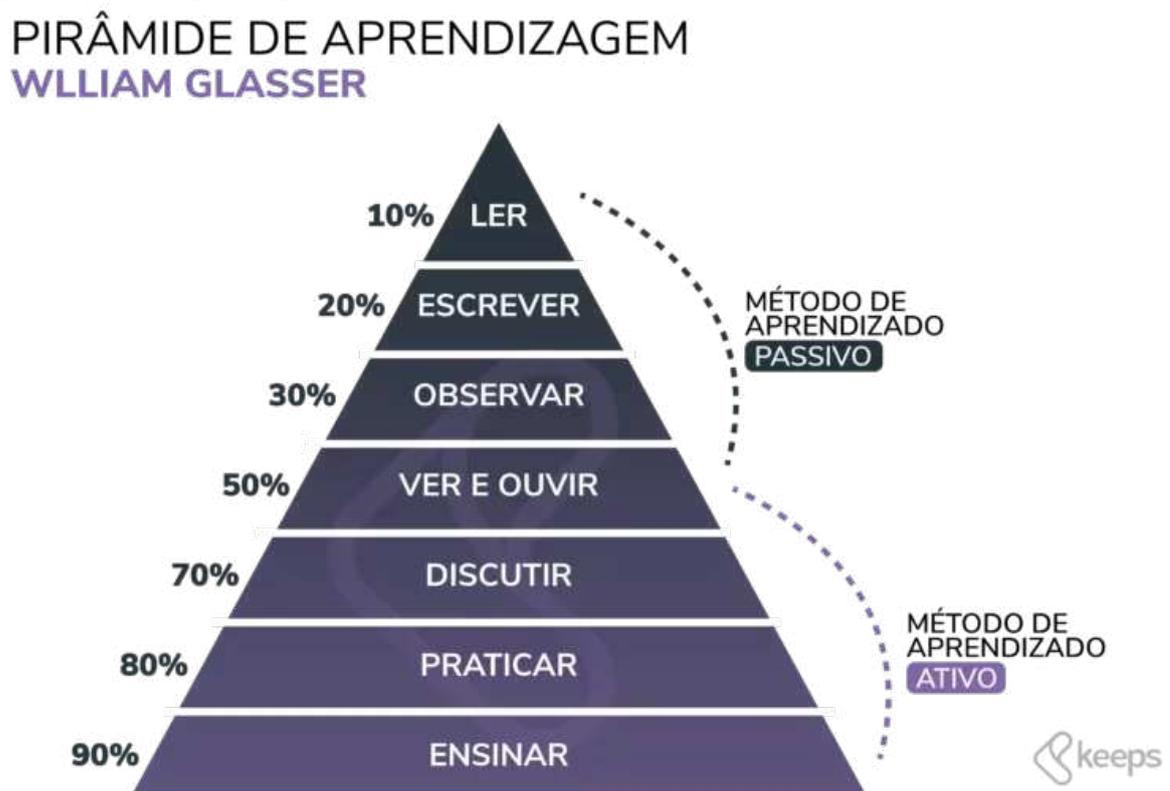
Uma das formas que vem sendo utilizada para atingir níveis maiores de aprendizado e de satisfação dos discentes é o *game-based learning* (aprendizado baseado em jogos). Esse tipo de aprendizado tende a promover maior engajamento, interação e entretenimento entre os participantes (JABBAR; FELICIA, 2015). Além disso, a forma de aprendizagem vem ganhando força nos últimos anos tanto pela consolidação da era digital como pela familiaridade dos jovens com jogos. Por isso, a utilização o *game-based learning* de maneira ativa ou passiva tende a ser um bom recurso para os discentes e docentes do século XXI, por trazer aspectos digitais e não-digitais comumente utilizados por eles em seu dia-a-dia e por tornar as aulas mais atraentes em algumas situações.

Paralelo a isso, as formas de ensino e aprendizado como a de formato meramente expositivo, onde o docente é detentor do conhecimento e o transmite de maneira unidirecional para os discentes, vem se tornando cada vez menos atraentes para os discentes por não envolver

atividades ativas que captem sua atenção. Estima-se que a atenção dos estudantes vai se dissipando a partir dos primeiros 15 minutos de aula (PINTO; SILVA, 2017). No entanto, esse formato ainda é amplamente utilizado nos dias atuais.

Por esse prisma, para conter essa perda de atenção algumas atividades que visam dinamizar o aprendizado precisam entrar em ação. Além disso, de acordo com o psiquiatra americano William Glasser, existem algumas formas de aprendizado e elas podem ser representadas por meio de uma pirâmide, separadas por dois tipos de aprendizagem, que são a aprendizagem ativa e passiva, e cada tipo contém seus sub-níveis, conforme mostrado no Figura 1 (FORMIGHIERI, 2022).

Figura 1 – Exemplo da pirâmide de William Glasser



Fonte: Blog Keeps (2021).

Entre o meio e o topo da pirâmide, representando os tipos de aprendizagem passiva, estão elencados tipos de aprendizagem como “ver”, “ouvir”, e o “ver e ouvir”, que representam um nível de aprendizado de até 50%. No nível “ver e ouvir”, é onde as aulas somente expositivas podem se enquadrar, pois os estudantes estão somente vendo e ouvindo o professor escrever na lousa e falar, respectivamente. Algumas atividades como escrever o que está sendo falado e elaborar mapas mentais no momento da aula, estão no meio da pirâmide, pois envolvem a

aprendizagem passiva e ativa por estar escrevendo e interligando os assuntos ouvidos. Nesse caso, assumindo que o estudante faça todas essas atividades, ele poderá chegar até 80% de aprendizado. No entanto, é difícil manter atenção em aulas expositivas após um longo período de tempo.

Logo, eventualmente o discente pode somente estar vendo e ouvindo, e parar de escrever por alguma razão. No nível entre o meio e a base da pirâmide, estão as aprendizagens ativas que englobam atividades em grupos, debates e atividades práticas com compartilhamento de ideias, onde o aprendizado pode chegar até 95%. É exatamente na aprendizagem ativa que as estratégias de ensino dinâmicas como jogos se enquadram, pois envolvem interação entre participantes, necessidade maior de pensamento analítico, debates e esclarecimento de idéias (SOUSA; MARQUES, 2020). Além disso, nesses tipos de ensino com auxílio de jogos um outro fator bem interessante pode entrar em campo, a adrenalina, pois o jogo instiga os participantes e promove uma competição (que deve ser saudável para que o andamento da atividade ocorra bem).

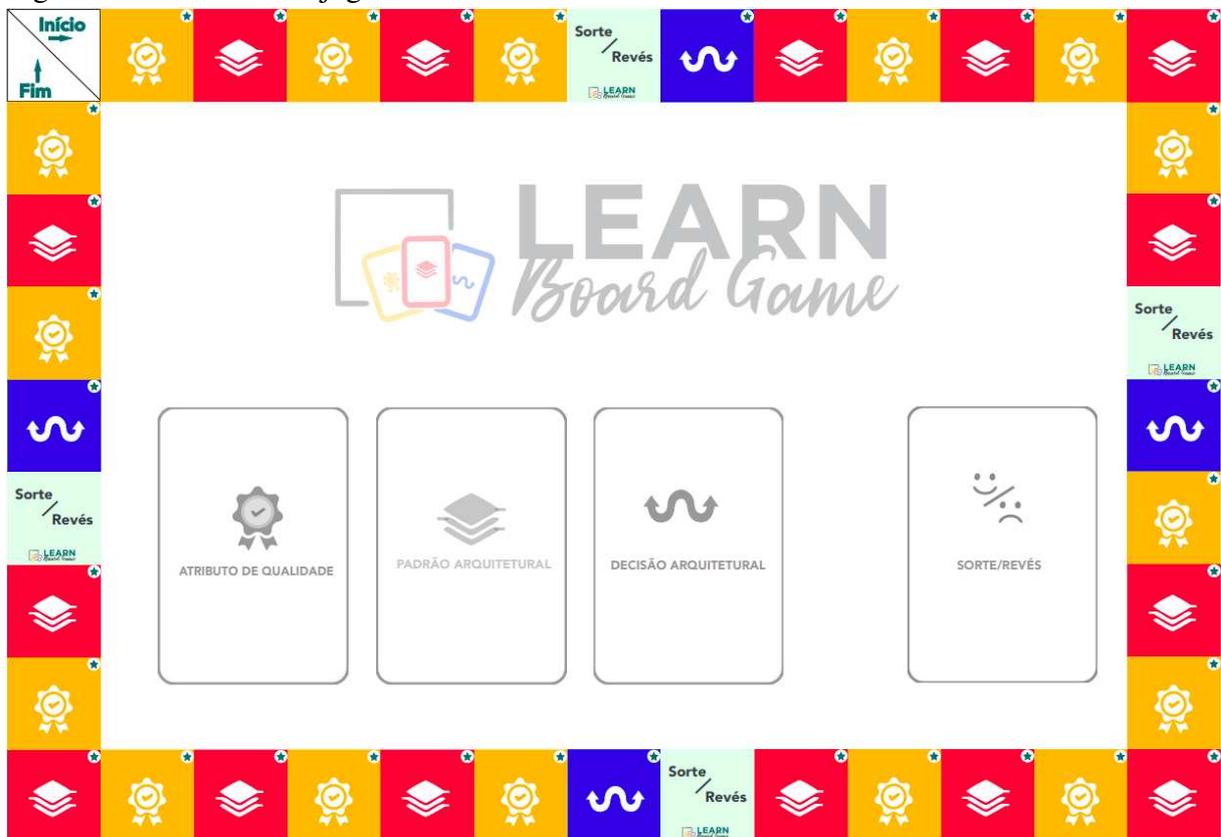
No cenário atual brasileiro, alguns jogos focados no auxílio do aprendizado vem se tornando cada vez mais comuns entre os estudantes. Geralmente esses jogos tem um cenário gamificado, ou seja, um cenário onde o usuário aprende sobre algum assunto, resolve questões sobre o mesmo e vai obtendo progresso por meio de seus acertos e subindo no *ranking* do jogo. Quanto maior o ranking, mais difíceis tendem a ser as perguntas. Isso faz com que o usuário comece com assuntos triviais e vá progredindo até dominar assuntos relativamente difíceis. Como exemplo, pode-se citar o *Graphogame* Brasil, aplicativo para celular que visa ensinar conteúdos como letras do alfabeto, sílabas e números para crianças de até seis anos (MINISTÉRIO DA EDUCAÇÃO DO BRASIL, 2023; GRAPHO GAME, 2023). Além deste, pode-se citar o *Duolingo*, que visa o ensino de línguas estrangeiras (SALUTES, 2022).

No tocante ao aprendizado de disciplinas relacionadas a área da computação, aplicação da *game-based learning* tende a fornecer gradualmente para os estudantes universitários dessa área *soft-skills* muito requisitadas na indústria como comunicação, trabalho em equipe e liderança (GARCIA *et al.*, 2020). Como exemplo, pode-se citar o SimSE, jogo digital para ensino de processos de engenharia de *software* e que ensina gradualmente como melhorar a gestão de um projeto de *software*, por meio de um ambiente que simula um projeto real contendo orçamento, desenvolvedores, demandas e problemas (NAVARRO *et al.*, 2004).

### 3.3 LEARN Board Game

O LEARN (*LEarning software ARchitecture fundameNtals*) é um jogo não-digital criado por Tamires Ariane, ex aluna da Universidade Federal do Ceará — Campus de Russas e visa auxiliar na aprendizagem de conceitos e padrões voltados para a Arquitetura de *Software* (SOUSA; MARQUES, 2020). O jogo utiliza um tabuleiro para controle do progresso dos participantes, e cartas contendo perguntas que quando respondidas corretamente fazem com que o jogador avance no tabuleiro.

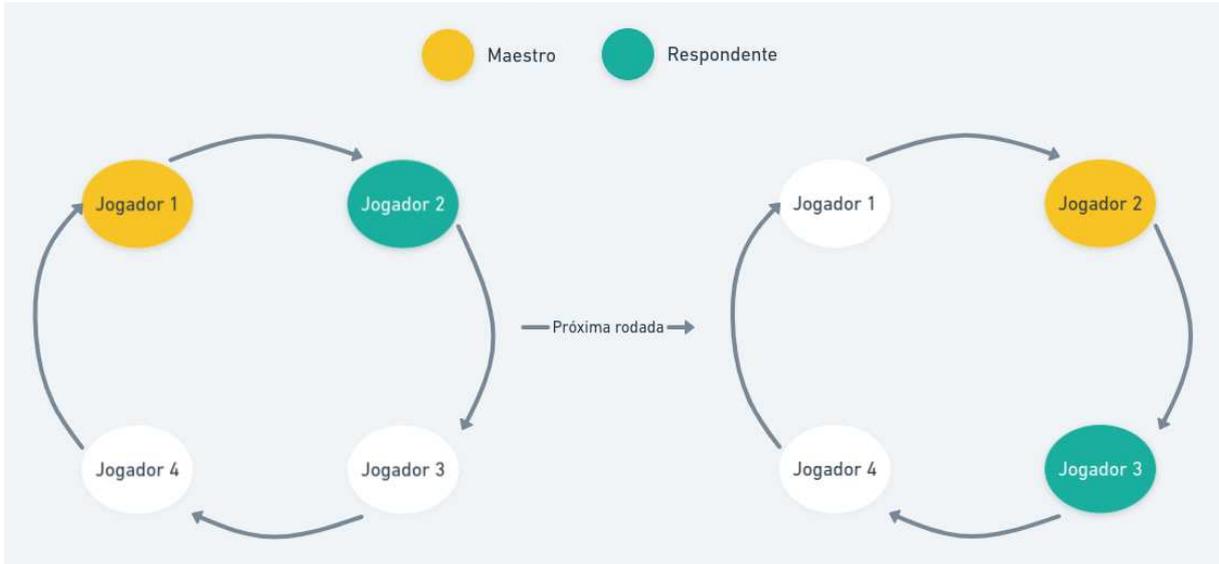
Figura 2 – Tabuleiro do jogo LEARN



Fonte: Sousa e Marques (2020)

O jogo funciona da seguinte maneira: dois ou mais jogadores escolhem um pino/marcador e todos iniciam no marco zero do tabuleiro. Em relação aos papéis do jogo, existem dois: o maestro e o respondente. O maestro é aquele que irá fazer a pergunta, enquanto o respondente é quem vai responder. Após o respondente concluir sua vez, ele assumirá o papel de maestro e o respondente será o próximo jogador e assim por diante.

Figura 3 – Demonstração do fluxo dos papéis do jogo



Fonte: elaborado pelo autor.

A primeira rodada poderá começar com qualquer um sendo o maestro, e o respondente sendo o participante imediatamente a sua esquerda ou direita, dependendo do sentido escolhido em consenso pelos jogadores. O maestro escolherá dentre alguns tipos de cartas — que ficam viradas para baixo —, que pertencem aos temas de atributos de qualidade, decisões arquiteturais e padrões arquiteturais, e irá ler a pergunta contida na carta para o respondente. Cada carta contém um número de estrelas (1 a 6), e em caso de acerto o jogador que foi indagado avançará seu pino a quantidade de casas representada pelo número de estrelas.

Figura 4 – Exemplo de carta (frente e verso): Atributo de qualidade



Fonte: elaborado pelo autor.

Ademais, quando todos saírem do marco zero, as cartas deverão ser puxadas de onde o respondente está no tabuleiro. Além disso, existe uma casa do tabuleiro de sorte ou revés e se um jogador parar nela após uma jogada, terá que puxar uma carta de sorte, sem relação com conteúdos, na qual poderá avançar mais casas por meio da “sorte”, ou poderá retroceder casas em caso de “revés”. Essa carta deixa o jogo mais dinâmico e divertido.

Geralmente, o jogo é aplicado no meio do semestre após o ensino teórico dos conceitos que serão utilizados no jogo. O jogo foi aplicado em algumas turmas de arquitetura de *software* desde o seu lançamento e sempre obteve resultados positivos em *feedbacks* coletados após a experiência.

### 3.4 *Design Science Research*

O DSR é um método de pesquisa utilizado para construção ou atualização de artefatos que visam resolver problemas existentes, e se baseia na ideia de que quanto maior a base de conhecimento sobre um problema, melhor será o resultado final do artefato (HEVNER; CHATTERJEE, 2010).

Para garantir um bom andamento do DSR, pode ser utilizado uma abordagem pragmática e satisfatória para garantir a viabilidade da pesquisa e contribuir com inovações em problemas reais existentes da humanidade (pragmatismo) e também contribuir de maneira a entregar um artefato que satisfaça os critérios definidos no início da pesquisa (satisfatoriedade). No entanto, para o bom andamento do DSR é necessário uma boa harmonia entre o pragmatismo e satisfatoriedade (HEVNER, 2007).

Além disso, para entender o DSR, é necessário entender que há uma distinção entre problemas práticos e questões de conhecimento. Problemas práticos se referem ao problema que os *stakeholders* desejam resolver, como por exemplo “diminuir o índice de evasão de escolas do ensino médio”, enquanto as questões de conhecimento centram na relação entre o problema prático e algum aspecto do ambiente, como por exemplo “relação entre a estrutura do ensino médio e estudantes que causam a evasão escolar” (WIERINGA, 2010).

No tocante aos artefatos gerados pelo DSR, podem ser do tipo: conceitos, modelos, métodos e instâncias. Tais artefatos podem resolver um problema existente sem solução (inovação) ou melhorar um artefato já existente (melhoria) (GEERTS, 2011).

Por último, pode-se falar sobre o ciclos do DSR. Segundo Wieringa (2010), um ciclo DSR pode conter 3 etapas: Investigação do problema (etapa 1), *design* de uma solução (etapa 2)

e validação da solução (etapa 3). Na etapa de investigação do problema (etapa 1), pode-se incluir atividades de revisão da literatura, definições de problemas e questões de pesquisa, coletas de dados, entendimento do ambiente, critérios dos *stakeholders*, entre outras atividades voltadas para o enriquecimento da base da pesquisa. Na etapa de *design* (etapa 2), são trabalhados aspectos como arquitetura da solução (caso necessário), protótipo da solução e implementação. Por último, na etapa de validação (etapa 3) utiliza-se os artefatos concluídos da etapa 2 para validação dos *stakeholders*. Nesta etapa estão incluídas atividades como aplicação da solução, coleta de *feedbacks*, análise de uso e os critérios de satisfatoriedade e pragmatismo.

## 4 TRABALHOS RELACIONADOS

A aplicação de jogos educacionais para auxiliar no ensino de algum conteúdo relacionado à engenharia de *software* tem sido utilizada muitas das vezes sob a perspectiva que somente o curso de computação em si, com suas aulas teóricas e práticas não são suficientes para formar profissionais capacitados para atuar na indústria (NAVARRO *et al.*, 2004). Logo, tem se tornado cada vez mais comum presenciar o ensino de conteúdos relacionados à engenharia de *software* com o apoio de ferramentas educacionais não-digitais como por exemplo o LEARN, *Problems and Programmers*, *Smart Decisions e DecidArch* e digitais onde pode-se citar o SimSE (SOUSA; MARQUES, 2020; BOER *et al.*, 2019; CERVANTES *et al.*, 2016; NAVARRO *et al.*, 2004).

O LEARN (*LEarning software ARchitecture fundameNtals*) é um jogo não-digital que visa abordar conteúdos relacionados a arquitetura de *software* de uma maneira lúdica, utilizando um jogo com tabuleiro e cartas. A relação entre o D-LEARN e o LEARN é direta, uma vez que o D-LEARN será uma versão digital aprimorada do LEARN. Logo, aspectos como *design* do jogo e regras poderão ser comuns entre as duas versões. No entanto, alguns aspectos como ambiente de uso, forma de interação e forma de criação serão distintos entre os dois jogos, dado que o D-LEARN será construído em formato digital e com isso vários aspectos provenientes do mundo digital irão estar presentes nele. O jogo foi avaliado por meio de um questionário utilizando a abordagem MEEGA+ (LUND, 2001) e teve resultados satisfatórios em suas avaliações promovidas em cada semestre, consolidando-o na disciplina de arquitetura de *software* e mostrando que o jogo obteve sucesso nos seus objetivos.

Também no âmbito de conceitos e decisões arquiteturais de *software* pode-se citar o *Smart Decisions*, um jogo não-digital com utilização de cartas e que aborda conceitos sobre decisões arquiteturais utilizando a abordagem *Attribute-Driven Design* (ADD) (CERVANTES *et al.*, 2016). O jogo é dividido em iterações, onde cada iteração contém um conjunto de controladores principais - como custo e escalabilidade - chamados de *drivers*, um elemento para refinamento que pode ser considerado como a “meta” da iteração e um conjunto de cartas para serem escolhidas contendo alternativas de solução para atingir a meta. Cada carta contém informações de consequências em cada *driver* e são medidas em quantidades de estrelas. Após escolher uma carta, os valores das estrelas de cada *driver* na carta são somados formando um *fitness* da escolha. Essas informações são colocadas em um *board* da pontuação de cada iteração. Além disso, os *fitness* são considerados da seguinte forma: se o *fitness* for entre 4 e 9, a decisão é

considerada como razoável. Se o *fitness* for menor que 4, a decisão é considerada ruim e há uma penalização de 2 pontos. Por último, *fitness* entre 10 e 12 são consideradas decisões excelentes e tem uma bonificação de 2 pontos. Por meio dessa dinâmica de iterações, decisões, consequências e pontuações o jogo acontece. Ao fim, a pontuação é somada e é definido o jogador vencedor. O jogo teve um piloto voltado para um projeto focado em *Big Data* com resultados satisfatórios e foi testado junto a educadores, estudantes e profissionais da área com intuito de validar a proposta e coletar *feedbacks* acerca de possíveis melhorias. Por fim, os resultados da pesquisa feita com 41 pessoas foram em maioria positivos, indicando que o jogo é uma ferramenta que pode auxiliar no aprendizado de arquitetura de *software*.

Além do LEARN e *Smart Decisions* pode-se citar o DecidArch, um jogo não-digital que visa auxiliar no ensino de arquitetura de *software* por meio de um jogo de cartas simples, jogado em grupo e que tem foco em três objetivos de aprendizado: conhecimento sobre *trade-offs* e decisões arquiteturais, apreciação sobre decisões tomadas por outros membros da equipe e conhecimento da dinâmica de um design arquitetural (BOER *et al.*, 2019). Neste jogo, existem 4 tipos de cartas: cartas de projeto — contém uma breve descrição sobre o projeto —, carta de *stakeholders* — contém informações sobre um *stakeholder* —, cartas de interesse — contém informações sobre interesses do projeto e opções de *design*, onde cada opção pode ser avaliada como muito negativa, negativa, boa e muito boa — e cartas de eventos — eventos que podem impactar o projeto, como por exemplo a saída de um membro. O jogo encerra-se após 30 minutos ou quando todas as cartas de interesse foram trabalhadas.

A dinâmica do jogo se dá em rodadas, onde em cada rodada um jogador escolhe uma carta de interesse e os outros jogadores definem quais opções de *design* estão disponíveis para esta carta escolhida. Ao fim da rodada, é decidido em grupo qual a decisão arquitetural será tomada e é escolhido uma carta de evento para acrescentar uma situação não esperada no projeto, a fim de dinamizar ainda mais o jogo. No fim do jogo, os jogadores podem conferir as opções de *design* escolhidas e quantas foram positivas e negativas para entender se foram bem. O jogo foi avaliado por meio de um questionário realizado com 83 pessoas e teve resultados positivos, concluindo-se que os participantes gostaram de jogar o jogo e demonstraram que o jogo ajudou-os a entender melhor conceitos teóricos difíceis de abstrair em sala de aula, no entanto, questões relacionadas ao tempo do jogo foram pontuadas como pontos de melhoria para o jogo.

No âmbito mais amplo de engenharia de *software*, tem-se o *Problems and Program-*

mers. Este é um jogo também não-digital e que utiliza cartas. Nele, é visado o ensino do processo de engenharia de *software* por meio de uma abordagem competitiva e que envolve aspectos de requisitos, *design*, implementação, manutenção e recursos humanos e financeiros (NAVARRO *et al.*, 2004). Além disso, os jogadores passam por todos esses aspectos mencionados utilizando a metodologia de desenvolvimento em cascata. A dinâmica do jogo se dá em grupo, onde dois jogadores assumem o cargo fictício de líderes em um projeto e escolhem 5 cartas do baralho principal, que envolve cartas de conceitos, problemas e programadores. Como exemplo, nas cartas de conceitos pode-se ter cartas relacionadas a código limpo, nas de problemas pode-se ter relacionadas a falta de recursos e na de programadores cartas relacionadas ao perfil de um programador. Por meio delas a dinâmica do jogo acontece, onde os jogadores utilizam as cartas que favorecem o projeto para mitigar ou evitar problemas e ter um bom desempenho no projeto, enquanto o outro jogador tenta jogar cartas que podem prejudicar o andamento do projeto do seu adversário, simulando os aspectos reais de um projeto onde problemas inesperados podem acontecer como por exemplo na saída de um programador. O jogo segue nessa dinâmica e o vencedor é o que conseguir concluir o projeto. Este jogo se assemelha ao D-LEARN por utilizar cartas e um jogo competitivo e se diferencia por não possuir um tabuleiro e por ser mais abrangente no conteúdo de engenharia de *software*, uma vez que o D-LEARN é focado em arquitetura de *software*. Em avaliação, o *Problems and Programmers* obteve resultados satisfatórios. Os respondentes da pesquisa acharam o jogo moderadamente fácil de usar e responderam que o jogo ajuda no aprendizado do processo de engenharia de *software* como foi proposto.

Por último, tem-se o SimSE que é um jogo digital e que visa o ensino do processo da engenharia de *software* por meio de um ambiente que simula o cenário real de um projeto (NAVARRO *et al.*, 2004). A metodologia do jogo permite apenas um jogador, que irá atuar como um gerente de um projeto, comandando um time de desenvolvedores e realizando tarefas como atribuição de atividades, monitoramento de progresso e compra de ferramentas necessárias para o desenvolvimento. O jogo inicia com um projeto a ser realizado, um orçamento máximo a ser usado e um tempo máximo para entrega do projeto. O tempo é dado em ciclos do relógio, ciclo representando 5 segundos por exemplo. O jogador tem opção de quais desenvolvedores irá pôr na equipe e pode tomar decisões no decorrer do jogo para cada empregado escolhido, como comprar um computador melhor, uma nova ferramenta de documentação, uma nova IDE (Ambiente de desenvolvimento integrado) ou até mesmo dar-lhe uma promoção. Obviamente todas essas ações irão ter um custo, que será retirado do orçamento do projeto. Também há ações

Tabela 1 – Comparação do D-LEARN com os trabalhos relacionados

<b>Trabalho</b>	<b>Utilizou a metodologia DSR?</b>	<b>É um jogo digital?</b>	<b>Há utilização de tabuleiro?</b>	<b>Há utilização de cartas?</b>	<b>É voltado para o ensino de arquitetura de software?</b>
<b>LEARN</b>	X		X	X	X
<b>Smart Decisions</b>				X	X
<b>Problems and Programmers</b>				X	
<b>SimSE</b>		X			
<b>DecidArch</b>				X	X
<b>D-LEARN</b>	X	X	X	X	X

Fonte: elaborado pelo autor.

que preservam o orçamento, como diminuir o salário de um empregado, porém esta ação pode acarretar na insatisfação e auto-demissão do empregado. Para cada decisão, há uma reação que favorece ou desfavorece o andamento do projeto, portanto é importante que o jogador tome boas decisões para evitar perda de tempo, recursos humanos e financeiros. Os resultados do SimSE lhe forneceram o *Premier Award for Excellence in Engineering Education Courseware* (Prêmio de excelência em cursos de formação em engenharia) de 2009, promovido pela Conferência Fronteiras na Educação em San Antonio, Texas. Contudo, desde então o jogo não teve mais tantas atualizações e segue com sua versão inicial. O jogo se assemelha ao D-LEARN por ser concebido para um ambiente digital, porém se diferencia por não se aprofundar nos aspectos relacionados à arquitetura de *software*.

Com exceção do SimSE, os demais jogos são não-digitais e somente o LEARN, *Smart Decisions* e *DecidArch* focam em aspectos da arquitetura de software, já que o *Problems and Programmers* e o SimSE focam no processo de engenharia de *software* como um todo. No Quadro a seguir é mostrada uma relação entre os trabalhos.

## 5 PROCEDIMENTOS METODOLÓGICOS

O DSR é um método de pesquisa utilizado para construção de artefatos que visam resolver problemas existentes, por meio de uma abordagem satisfatória e pragmática. Satisfatória no tocante a não obrigatoriedade do ótimo e pragmática no tocante a necessidade de criar algo que tenha sua aplicação viável. Além disso, o DSR trabalha com os conceitos de classes de problemas, a fim de categorizar as pesquisas que a utilizam. Para um trabalho seguindo as etapas do DSR, é necessário entender qual seu domínio, classe, referencial teórico e trabalhos semelhantes. Tal entendimento pode ser feito por meio de pesquisas e revisões literárias. Após isso, caso o artefato já exista pode-se fazer uma melhoria aplicando o DSR, e caso não exista pode-se criá-lo também utilizando a metodologia DSR (WIERINGA, 2010).

No presente trabalho, o domínio do DSR é o de ensino, e sua classe de problema é relacionada ao ensino tradicional nas disciplinas de arquitetura de software. Em relação ao artefato que será gerado ao fim do ciclo DSR, será uma melhoria do artefato já existente (LEARN) e não um artefato criado do zero. Logo, a metodologia DSR seguirá pelo viés de melhoria ao invés de concepção e criação.

Este capítulo apresenta as atividades que foram necessárias para realizar essa pesquisa e para atingir os objetivos propostos. Com o objetivo de detalhar melhor as atividades foram divididas em cinco subseções: (1) Etapas de pesquisa, (2) Procedimentos de entrevistas realizadas com os estudantes e professora responsável, (3) Análises das respostas, (4) Identificação dos principais problemas e questões de pesquisa, (5) Metodologia de desenvolvimento e (6) Validação da proposta.

### 5.1 Investigação do problema

Na presente seção, é realizado uma investigação do problema. Este processo é fundamental para compreender a complexidade, identificar adversidades e, por fim, constatar soluções eficazes.

#### 5.1.1 Revisão da literatura

No primeiro ciclo do DSR que engloba revisões de literatura e investigações sobre projetos relacionados ao problema de pesquisa, foram feitas pesquisas sobre jogos digitais para o ensino de arquitetura de software, no entanto não foram encontrados resultados satisfatórios

que abordassem de maneira semelhante os padrões e estilos arquiteturais tal como o LEARN aborda. A pesquisa foi feita utilizando as palavras chave “digital game”, “software architecture”, “learning” e/ou “teach”, nas bases de dados IEEE, ACM Digital Library e Google Scholar.

Então foram feitas pesquisas menos específicas tratando de jogos digitais para o ensino de engenharia de software, utilizando as palavras chave “game”, “software engineering”, “learning” e/ou “teach” também nas bases de dados IEEE, ACM Digital Library e Google Scholar, onde foram encontrados alguns resultados interessantes como o SimSE. Além disso, com relação a jogos não digitais, foi encontrado o jogo não-digital Problems and Programmers, que utiliza uma abordagem de cartas e tabuleiro, assim como o LEARN.

Em relação ao LEARN, ele é o artigo base para este trabalho, logo, com os critérios de pesquisa mencionados anteriormente, ele sempre esteve no topo dos resultados.

### ***5.1.2 Aplicação e observação da adoção do LEARN em sala de aula e coleta de dados***

Com finalidade de enriquecer a etapa de investigação do problema do ciclo DSR, foi necessário verificar o estado da prática, e nenhuma situação seria melhor para isso do que acompanhar a aplicação do LEARN Board Game da turma de arquitetura de software no semestre 2022.2 no Campus da Universidade Federal do Ceará (UFC) - campus de Russas. Nesta etapa, o jogo foi aplicado em conjunto com a então professora de arquitetura, Profa. Dra. Anna Beatriz dos Santos Marques. Por meio disso, foi possível perceber alguns problemas na prática por parte da professora e observar dificuldades encontradas pelos estudantes.

Após a dinâmica do jogo, um questionário foi aplicado a fim de coletar informações sobre a experiência dos estudantes em relação ao jogo, além de algumas informações em relação à familiaridade deles em relação a jogos.

### ***5.1.3 Entrevista com a professora responsável pela disciplina e aplicação do LEARN***

Após aplicação do jogo e do questionário, também foi feita uma entrevista com a professora responsável pela aplicação da dinâmica. Nesta entrevista, foi perguntado sobre quais as maiores dificuldades para aplicação do LEARN. O método utilizado para coleta de dados foi uma entrevista curta, onde foram realizadas algumas perguntas a fim de tentar entender quais os principais problemas do LEARN no seu formato atual.

#### **5.1.4 Definição do problema central e questões de pesquisa**

Seguindo a etapa 1 do ciclo DSR e com alguns dados de estudantes e da professora coletados, foram definidos o problema e questões de pesquisa. O problema central deste trabalho é como melhorar a aplicação do LEARN por meio de uma versão digital melhorando assim a dinâmica de ensino da disciplina de arquitetura de software, embasado nos pontos negativos encontrados. Tais pontos são relacionados a aplicação do LEARN como: tempo de confecção, manuseio, preparação e alteração do jogo, e relacionados ao aprendizado e familiaridade dos discentes, que indicaram um baixo uso de jogos não-digitais em relação a jogos digitais. Além disso, nas perguntas subjetivas os discentes sugeriram que o jogo poderia ter mais cartas para abranger mais conteúdos. No tocante às questões de pesquisa, foram definidos:

- Como uma versão digital do LEARN poderia resolver os seus problemas mais pertinentes?
- Como uma versão digital do LEARN poderia continuar a auxiliar no ensino de maneira lúdica e objetiva conteúdos relacionados à arquitetura de software?
- Que novas funcionalidades promoveriam uma interação melhor entre os estudantes e o jogo?

#### **5.1.5 Definição da classe de problema**

Seguindo os conceitos do DSR, é necessário definir qual classe de problema este trabalho irá utilizar por meio da metodologia. A classe de problema foi relacionada ao ensino tradicional nas disciplinas de arquitetura de software. Tal classe foi embasada nos problemas atuais do ensino tradicional de arquitetura de software como: falta de atenção dos estudantes, dificuldade para abstrair os conteúdos e falta de dinamicidade na disciplina. Esses problemas são em maioria gerados pelo ensino tradicional meramente expositivo unidirecional, que vem se tornando cada vez menos aceito pelos estudantes do século XXI.

#### **5.1.6 Definição dos objetivos**

Fechando a etapa 1 do ciclo DSR, foram definidos os objetivos de pesquisa. O objetivo central do trabalho é desenvolver uma versão digital do LEARN, o D-LEARN (*Digital Learning software ARchitecture fundameNtals*). Os objetivos específicos se encontram detalhados no Capítulo 2 deste trabalho.

## 5.2 Design e Desenvolvimento

Esta etapa consiste na criação da solução idealizada para o problema identificado. Por se tratar de um software, as atividades realizadas são baseadas em atividades do ciclo de vida de desenvolvimento de software.

### 5.2.1 Definição dos requisitos

Nesta etapa, foram definidos os requisitos funcionais, não funcionais e regras de negócio do D-LEARN. Para isso, foram realizadas sessões de conversa com a professora da disciplina a fim de elicitare os requisitos do projeto, sempre focando nas funcionalidades já existentes do jogo não-digital e as possíveis novas funcionalidades que agregam valor para o jogo digital. Os requisitos são apresentados no Capítulo 6.

### 5.2.2 Definição da arquitetura do D-LEARN

Nesta etapa, foi definido a arquitetura do jogo. Foram considerados os requisitos elicitados anteriormente para tomada de decisões. No fim, uma arquitetura híbrida foi escolhida, sendo um cliente-servidor com dependências de serviços externos relacionados à arquitetura em nuvem. Essa arquitetura é melhor detalhada no Capítulo 6.

### 5.2.3 Criação do Design e Desenvolvimento do jogo

Nesta etapa, foi criado o *Design System* do projeto e foi iniciado o desenvolvimento do jogo. Para a criação do *Design System*, foi utilizado a ferramenta de design Figma. Nesse *Design System*, todos os componentes que iriam ser utilizados no jogo foram criados em forma de protótipo, o que facilitou a criação deles quando o desenvolvimento foi iniciado. Quanto ao desenvolvimento, as tecnologias utilizadas foram da *stack* JavaScript tanto para o *Frontend* como para o *Backend*. O protótipo é mostrado no Capítulo 6.

## 5.3 Validação

Nesta etapa foram realizados os procedimentos de validação junto aos *stakeholders*, a fim de coletar *feedbacks* sobre o D-LEARN e em caso de melhorias, desenvolvê-las em próximas versões. A validação foi feita por meio de um questionário chamado MEEGA+, que é um

questionário de avaliação que tem como objetivo avaliar jogos para computação por meio da percepção dos jogadores (PETRI *et al.*, 2018). Uma validação com a professora da disciplina foi realizada, a fim de validar se as funcionalidades de gerenciamento do D-LEARN foram feitas de maneira satisfatória. Validação, resultados e análises são detalhados no Capítulo 6.

## 6 PROCESSO DE CRIAÇÃO DO D-LEARN

Este capítulo apresenta os artefatos desenvolvidos durante o processo de criação do D-LEARN. Os artefatos apresentados constituem a documentação técnica do jogo digital.

### 6.1 Requisitos

Nesta etapa, foram definidos os requisitos funcionais listados na Tabela 2, não-funcionais na Tabela 3 e regras de negócio do projeto D-LEARN na Tabela 4. Além das funcionalidades já existentes no LEARN (que podem ser vistas na seção 3.3 do Capítulo 3 e no Capítulo 4) que se transformarão em requisitos funcionais, outras funcionalidades para o jogo também foram adicionadas.

#### 6.1.1 Requisitos Funcionais

Tabela 2 – Requisitos funcionais do projeto

ID	Requisito	Prioridade
RF01	O sistema deve ter uma landing page de apresentação do projeto, contendo seções de apresentação, informações e contribuidores.	Alta
RF02	O sistema deve ter um tabuleiro contendo todas as casas do LEARN Board game.	Alta
RF03	O sistema deve ter “pilhas” de cartas no meio do tabuleiro, contendo os quatro tipos de cartas (atributos de qualidade, decisões arquiteturais, padrões arquiteturais e sorte ou revés) do LEARN.	Alta
RF04	O sistema deve abrir um modal quando o usuário clicar em uma “pilha”, o modal deve conter os metadados da carta (descrição, pergunta, quantidade de estrelas), a vez do jogador e botões de ações “Verdadeiro” ou “Falso”.	Alta
RF05	O sistema deve mostrar ao usuário qual jogador possui a vez.	Alta

RF06	Ao entrar na página do jogo, o sistema deve permitir através de um modal que o usuário escolha o modo de jogo(logar-se ou jogar como “Guest”).	Alta
RF07	O sistema deve possibilitar o login e logout de um usuário.	Alta
RF08	O sistema deve possibilitar o cadastro de um usuário.	Alta
RF09	Após escolher o modo de jogo(logado ou como “Guest”), o sistema deve mostrar um Modal de configuração, onde será informado o nome dos jogadores. Cada jogador irá assumir uma cor.	Alta
RF10	O sistema deve mostrar em qual casa do tabuleiro cada jogador está.	Alta
RF11	Ao passar o mouse em cima de um “pino” de jogador, o sistema deve mostrar o nome dele .	Alta
RF12	Ao acertar a pergunta de uma carta, o sistema deve mostrar um feedback e mover o jogador da sua casa atual, até a próxima casa (que é determinada pela quantidade de estrelas na carta).	Alta
RF13	Ao errar a pergunta de uma carta, o sistema deve mostrar um feedback e deixar o jogador da sua casa atual.	Alta
RF14	Após cada jogada(certa ou errada), o sistema deve passar a vez para o próximo jogador.	Alta
RF15	Ao “cair” em uma casa de sorte ou revés, o sistema deve abrir uma Modal com a carta de sorte ou revés automaticamente.	Alta
RF16	Ao “cair” em uma carta de sorte ou revés que tenha ações de ganho/perda em próximas jogadas, o sistema deve assignar esse ganho/perda ao jogador e executá-lo na próxima jogada automaticamente.	Alta
RF17	O sistema deve ter uma Modal com o Ranking do jogo.	Alta

RF18	O sistema deve possibilitar que os jogadores encerrem o jogo antes de chegar ao fim do tabuleiro, por meio de um botão.	Alta
RF19	Ao encerrar o jogo, o sistema deve mostrar a Modal de Ranking automaticamente.	Alta
RF20	O sistema deve mostrar um botão para reiniciar o jogo na Modal de Ranking quando o jogo for encerrado.	Alta
RF21	O sistema deve ter um botão para abrir a Modal de Ranking a qualquer momento.	Alta
RF22	O sistema deve encerrar o jogo automaticamente e mostrar a Modal de Ranking quando um jogador chegar na última casa.	Alta
RF23	O sistema deve possibilitar login/logout de um administrador, em um lugar diferente do login de um usuário comum.	Alta
RF24	Ao logar-se como administrador, o sistema deve entrar em um painel administrativo com os menus de cartas e jogos.	Alta
RF25	Ao logar-se como administrador, o sistema deve possibilitar que o administrador tenha acesso ao CRUD (Criar, Remover, Atualizar e Listar) de cartas, exceto para as cartas de sorte ou revés que terão somente visualização.	Alta
RF26	No cadastro de cartas, o sistema deve possibilitar o upload de uma imagem.	Alta
RF27	Ao logar-se como administrador, o sistema deve possibilitar a visualização de jogos realizados.	Alta
RF28	Na página de CRUD de cartas, o sistema deve possibilitar o filtro por tipo de carta.	Alta

RF29	O sistema deve mostrar uma Modal de confirmação para toda ação que possa ser perigosa, como exclusão de dados e encerramento do jogo de maneira precoce.	Alta
RF30	O sistema deve utilizar uma estratégia de token de sessão para manter os usuários logados e rejeitar requisições em caso de token inválido.	Alta
RF31	O sistema não deve permitir que sejam escolhidas cartas das pilhas que não são do mesmo tipo da casa onde o jogador atual está.	Alta

Fonte: elaborado pelo autor.

### 6.1.2 Requisitos Não-funcionais

Tabela 3: Requisitos não funcionais do projeto

ID	Requisito	Prioridade
RNF01	Na Landing Page, deve conter uma âncora (link) para acessar o jogo e as seções Ranking e de contribuidores.	Alta
RNF02	O sistema deve mostrar feedbacks enquanto os recursos do jogo são carregados.	Alta
RNF03	O sistema deve “escurecer” a cor das pilhas de cartas que não são do mesmo tipo da casa onde o jogador atual está.	Alta
RNF04	O sistema deve emitir sons quando o jogador acerta ou erra uma pergunta e quando o jogo é finalizado.	Desejável
RNF05	O sistema deve ser responsivo.	Desejável
RNF06	Ao entrar na página do jogo, o sistema deve permitir através de um modal que o usuário escolha o modo de jogo(logar-se ou jogar como “Guest”).	Alta
RNF07	O sistema deve utilizar as cores predominantes do jogo não-digital LEARN (azul, vermelho e amarelo) para os botões e design do jogo.	Desejável
RNF08	O sistema deve possibilitar o cadastro de um usuário.	Desejável

RNF09	O sistema deve utilizar Node.js para construção do servidor e Express como framework.	Alta
RNF10	O sistema deve utilizar o MongoDB como banco de dados.	Desejável
RNF11	O sistema deve se conectar com o AWS S3 para armazenar as imagens das cartas.	Desejável
RNF12	O sistema deve utilizar React JS para construção da interface da aplicação.	Desejável
RNF13	O sistema deve ser hospedado em uma hospedagem gratuita.	Desejável
RNF14	O sistema deve utilizar TypeScript tanto no cliente como no servidor.	Desejável
RNF15	O sistema deve adotar o padrão API Rest para construção da API.	Desejável
RNF16	No cadastro de usuário comum, a senha deve conter no mínimo 6 caracteres.	Alta
RNF17	As fontes do jogo devem ser legíveis e adaptáveis ao tamanho da tela por meio da utilização de rem (“pontos da tela”) ao invés de px (pixels).	Desejável
RNF18	O prazo de expiração do token de sessão deve ser de 7 dias.	Alta

Fonte: elaborado pelo autor.

### 6.1.3 Regras de Negócio

Tabela 4: Regras de negócio do projeto

ID	Regra
RN01	As cartas de sorte ou revés não devem ser escolhidas pelos jogadores, apenas quando o mesmo “cai” em cima de uma casa de sorte ou revés.
RN02	O Log do jogo só é armazenado no servidor quando o jogo é finalizado e o usuário está logado.

Fonte: elaborado pelo autor.

## 6.2 Design da Arquitetura

O D-LEARN foi definido como a versão digital do LEARN. Logo, algumas variáveis precisaram ser consideradas para que a arquitetura do D-LEARN conseguisse atender todos os requisitos elicitados anteriormente.

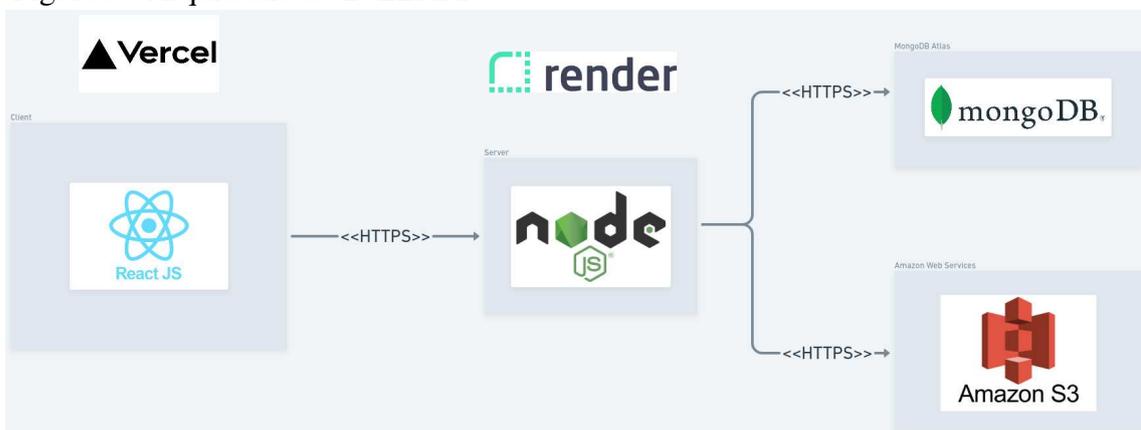
O D-LEARN precisava ser uma aplicação com boa portabilidade, usabilidade e performance tanto no cliente quanto no servidor. Além disso, o projeto necessitava de armazenamento persistente de dados e arquivos estáticos, logo, serviços externos também deveriam ser utilizados para este fim.

Por não ter um escopo tão grande, o D-LEARN não precisava de uma arquitetura tão robusta, no entanto, precisava de uma arquitetura escalável. Por isso, algumas boas práticas de desenvolvimento foram implementadas e algumas tecnologias confiáveis e amplamente aceitas na comunidade de desenvolvimento foram escolhidas.

Além dos requisitos do projeto, outro fator foi importante para decisão da arquitetura escolhida: o tempo para entrega do projeto. Por ser um projeto extenso e complexo, o desenvolvimento do D-LEARN precisava de agilidade na construção e familiaridade do desenvolvedor com as tecnologias.

Sabendo disso, foi escolhido a arquitetura cliente-servidor como base e a utilização de alguns recursos da arquitetura em nuvem para o projeto, sendo cliente o “nó” que irá conter toda a interface e interações com o usuário (cliente), e o servidor como “nó” que irá manter os dados e realizar operações (servidor). No cliente, foi utilizado o *Framework React JS* e no servidor, o *Node.js*. Além disso, alguns serviços externos foram consumidos. Na Figura 5 é possível visualizar a macro arquitetura da aplicação.

Figura 5: Arquitetura do D-LEARN



Fonte: elaborado pelo autor.

Como pode ser observado, foi utilizado *React JS* para construção da interface da aplicação e essa interface foi hospedada na *Vercel*. No servidor, foi utilizado *Node.js* para construção da API e os serviços *MongoDB Atlas* e *AWS S3* como banco de dados e armazenamento estático, respectivamente.

### 6.2.1 Definição do ambiente

O ambiente escolhido para o D-LEARN foi o ambiente *Web*, para maior portabilidade de acesso, menor custo e melhor manutenibilidade. Além disso, o D-LEARN foi desenvolvido para ser utilizado principalmente em dispositivos de tela maior, como computadores, *tablets* e *notebooks*, no entanto, ele também se adapta parcialmente a telas menores.

Por meio disso, os usuários irão precisar somente de acesso a *internet* para utilização do jogo. Ademais, o ambiente *Web* possui vários buscadores como *Google*, que podem indexar o site do D-LEARN em seus resultados de busca, fazendo com que o site seja visto por pessoas fora da universidade.

### 6.2.2 Definição de tecnologias

Como já falado anteriormente, as tecnologias utilizadas foram da *stack JavaScript*, sendo para o cliente o *framework React JS*. Na parte do servidor da aplicação, as tecnologias utilizadas foram:

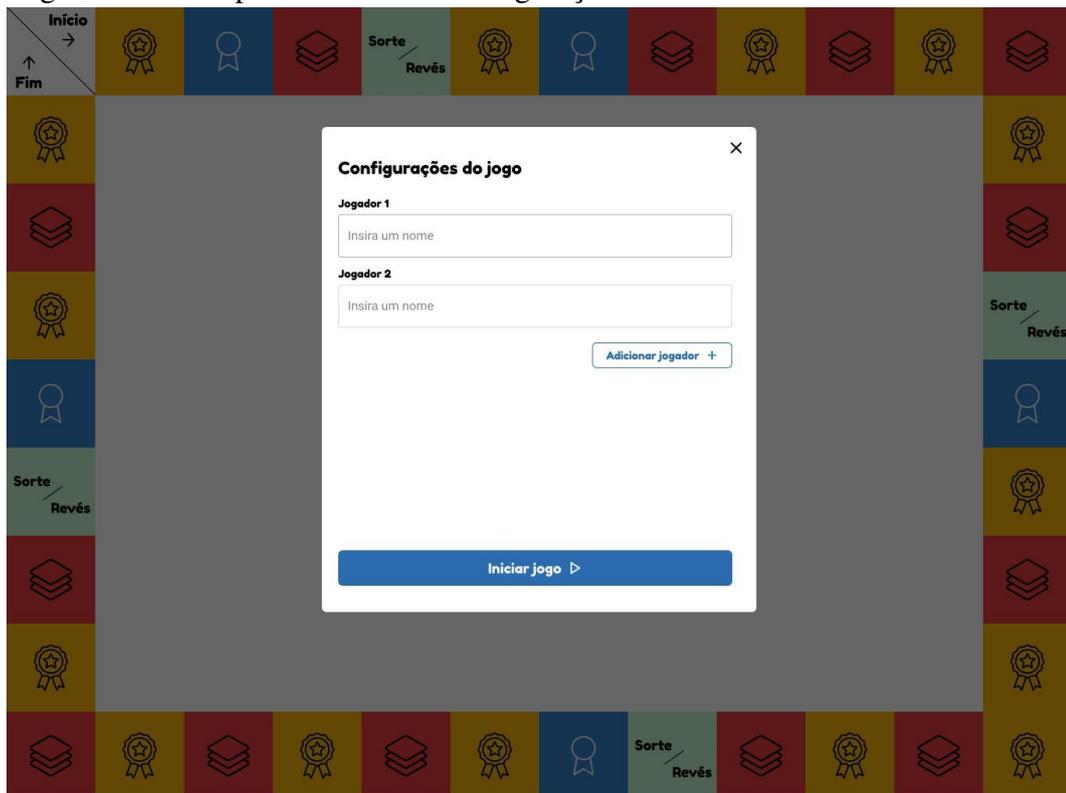
- *Node.js*;
- *Express*;
- *AWS S3*;
- *MongoDB* e *MongoDB Atlas*.

A tecnologia *TypeScript* foi utilizada em ambos (cliente e servidor).

## 6.3 Design da interface do usuário

Nesta etapa foi desenvolvido um protótipo para validação junto a professora da disciplina, o *Design* e possíveis fluxos da aplicação. Foi utilizada a ferramenta *Figma* para construção do protótipo. Nas figuras a seguir, é possível ver algumas telas do protótipo. O protótipo completo pode ser acessado através deste *link*: D-LEARN - Protótipo Figma .

Figura 6: Protótipo da Modal de Configuração do D-LEARN



Fonte: elaborado pelo autor.

Figura 7: Protótipo do tabuleiro e pilha de cartas do D-LEARN



Fonte: elaborado pelo autor.

## 6.4 Desenvolvimento

Foi utilizada a plataforma *Notion* para controle das tarefas de desenvolvimento do projeto. A priori, todas as funcionalidades provenientes dos requisitos foram transformadas em “*to-do*” no *Notion*. A partir disso, o processo de desenvolvimento foi seguindo completando item a item (em alguns casos também subitens). A cada progresso significativo, um encontro era marcado com a com a professora de arquitetura, Profa. Dra. Anna Beatriz dos Santos Marques e orientadora desta pesquisa, para validação do que foi feito. Além disso, o projeto foi hospedado desde o início, o que facilitava os testes da professora mesmo em momentos fora dos encontros de validação.

As primeiras funcionalidades desenvolvidas foram as funcionalidades do *Frontend*, onde toda a interface está contida. O jogo todo foi feito utilizando dados estáticos, ou seja, sem requisitar dados de algum servidor. As principais dificuldades encontradas no decorrer do desenvolvimento da interface foram aspectos relacionados à transição do jogo não-digital para o digital, onde foi necessário utilizar recursos da *Web* para atualizações automáticas da tela, efeitos sonoros e visuais, entre outros. Importante ressaltar que nessa fase do desenvolvimento, apenas o jogo em si foi criado, ou seja, o painel administrativo ficaria pendente até que o servidor ficasse pronto.

Após conclusão da interface do jogo da aplicação e validação junto à professora, era necessário construir o servidor da aplicação. Como ambos os projetos (interface e servidor) utilizaram *TypeScript*, foi fácil saber quais dados o *Frontend* iria necessitar e quais *Models* (modelos) precisariam ser criados no servidor. Com isso, as principais dificuldades encontradas foram somente na esfera de conexão com serviços de nuvem externos, como *MongoDB Atlas* e *AWS S3*, que são serviços de banco de dados não relacionais e serviços de armazenamento em nuvem, respectivamente. Por fim, o servidor foi hospedado em uma hospedagem gratuita para projetos *Node.js*, que tem algumas limitações no plano gratuito, mas que era suficiente para o servidor do D-LEARN. Tal hospedagem se chama *Render*, e pode ser acessada por meio deste *link*: <https://render.com>.

No decorrer do desenvolvimento do cadastro de cartas, um pequeno impasse foi encontrado nas cartas de sorte ou revés. Como há cartas que diminuem ou aumentam a quantidade de estrelas da próxima jogada de um jogador, era necessário fazer com que essa informação fosse “anexada” ao jogador, para que na próxima jogada ele utilizasse esse “anexo” para calcular quantas estrelas a carta atual (em caso de acerto) irá valer. Para isso, foi utilizado uma função

atribuída a cada carta de sorte ou revés que contém tal comportamento. Como funções *JavaScript* não podem ser armazenadas em banco de dados, essas cartas de sorte ou revés tiveram que ser cadastradas localmente (por meio de uma lista de objetos *JavaScript*) e depois mescladas com as demais cartas que virão do servidor. Como a proporção de cartas de sorte ou revés é bem menor comparando com os demais tipos de carta, e o tempo para adição de uma carta de sorte ou revés localmente ser por volta de no máximo 5m, esse desvio de fluxo ficará quase não usado.

Após conclusão do servidor, o painel administrativo foi criado, com intuito de manter as cartas e visualizar histórico de jogos. Uma página de login foi criada para ser possível entrar no painel. As figuras 8 e 9 apresentam algumas telas do painel.

Figura 8: Página de Login do painel do D-LEARN

Fonte: elaborado pelo autor.

Figura 9: Página de CRUD de cartas do painel do D-LEARN

#	Imagem	Tipo	Título	Estrelas	Pergunta	Solução	Ações
1		2	Repositório	4	[V ou F] Em repositório, os dados estão centr...	F	
2		0	Binding Time	2	[V ou F] Ao tomar uma decisão binding time é...	V	
3		1	Disponibilidade	4	[V ou F] Não se pode mensurar a disponibil...	F	
4		0	Modelo de coordenação	4	[V ou F] As decisões de modelo de coordenaç...	F	
5		0	Alocação de responsabilidades	4	[V ou F] A alocação de responsabilidades det...	F	
6		2	Map-Reduce	2	[V ou F] O Map-Reduce facilita tanto a leitura...	V	
7		0	Modelo de dados	4	[V ou F] O modelo de dados determina todos...	F	

Fonte: elaborado pelo autor.

Com interface do jogo, servidor e painel administrativo concluídos, era preciso criar a conexão entre a interface do jogo e o servidor, para que fosse possível consumir essas cartas cadastradas via painel no jogo. Além disso, foi preciso cadastrar todas as cartas por meio do painel - cartas que até então estavam estáticas no *Frontend* do jogo - no servidor para que estas ficassem disponíveis quando a interface do jogo requisita os recursos do jogo via API.

Por fim, o projeto foi concluído em toda sua interface para o jogo, interface para o painel e o servidor da aplicação. No entanto, foi notado que era necessário uma *Landing Page* (página de apresentação) para o D-LEARN. Logo, foi desenvolvida uma página para apresentar o jogo, apresentar os contribuidores do projeto e um *Ranking global* do jogo. Tanto essa página de apresentação, como o jogo, podem ser acessados por meio deste *link*: <https://d-learn.vercel.app>.

## 7 APRESENTANDO O D-LEARN

Este capítulo mostra o artefato gerado ao fim dessa pesquisa: o jogo digital D-LEARN. Serão apresentadas as principais funcionalidades e regras do jogo, a fim de deixar claro o que foi replicado do jogo não-digital e o que foi criado de novo.

### 7.1 Sobre o jogo

O D-LEARN foi concebido com intuito de melhorar o atual jogo não-digital LEARN, por meio de uma abordagem digital. Na versão digital do jogo, algumas alterações foram feitas tanto para melhorar o jogo atual quanto para evitar fluxos que não faziam mais sentido na dinâmica do jogo digital.

A primeira mudança foi em relação aos papéis. No D-LEARN, o papel de maestro pode ser opcional ou até mesmo nem existir, tendo em vista que o próprio jogador que tem a vez de jogar pode marcar “verdadeiro” ou “falso” nas perguntas e já receber o *feedback* instantâneo, deixando assim um mistério tanto para o jogador da vez quanto para o jogador que seria o maestro no jogo não-digital.

Outra mudança foi em relação à quantidade de jogadores. No jogo não-digital, cinco (ou até mais) jogadores conseguiam participar do jogo, mesmo com as limitações físicas como posicionar mais de cinco pinos em uma casa do tabuleiro. Além disso, a quantidade de jogadores era maior no jogo não-digital devido ao custo para confeccionar mais réplicas do jogo. Como o D-LEARN é um jogo digital, precisa somente de internet para funcionar, as limitações físicas são bem menores e não há necessidade de trabalho para confeccionar o jogo a cada experiência. A quantidade de jogadores no D-LEARN foi configurada para receber até quatro jogadores. Ademais, se um estudante/jogador deseja jogar o jogo sozinho, também é possível. Isso foi habilitado para que fosse possível que estudantes revisem o conteúdo por meio do jogo sem necessariamente precisar de um grupo.

Ainda sobre as mudanças, o tabuleiro do D-LEARN também sofreu alterações. No jogo não-digital, a quantidade de casas na horizontal era de 13 (treze) cartas, enquanto no jogo digital, a quantidade de casas é de 14 (quatorze). Essa alteração foi feita devido à melhor distribuição e uso de tela na plataforma web, uma vez que 1 (uma) casa foi reduzida nas casas verticais, tornando o tamanho de cada casa suficiente e igual em todas as casas do tabuleiro.

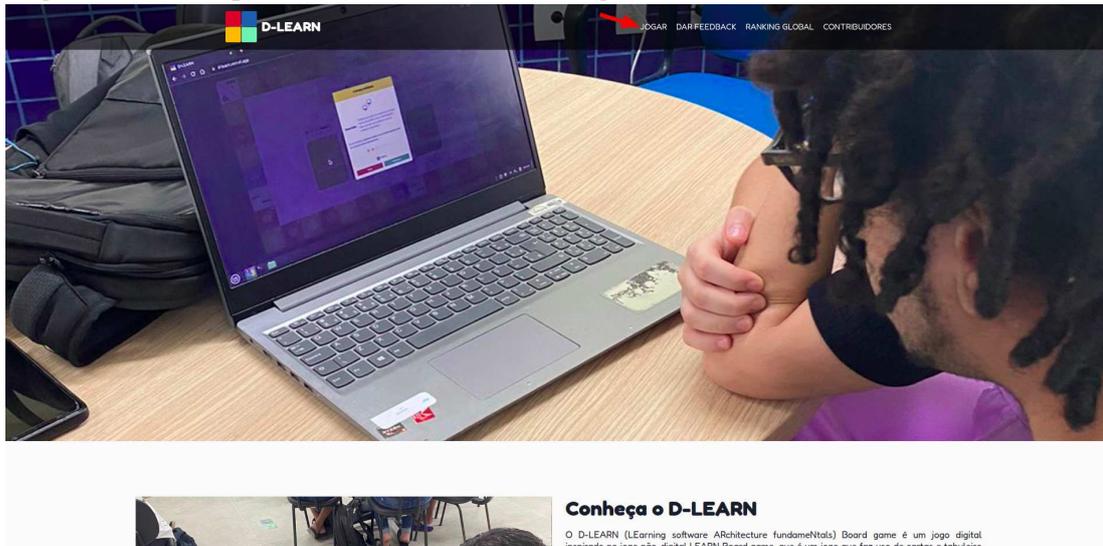
Portanto, algumas alterações foram feitas na versão digital do LEARN e algumas

foram bem notórias, como as que foram mencionadas acima. Algumas outras mudanças foram feitas, mas não tem o mesmo impacto no tocante a diferenças do jogo não-digital, pois foram adaptações relacionadas à plataforma web e visando atingir uma melhor experiência de usuário.

## 7.2 Iniciando e entendendo o jogo

Para iniciar o jogo, é necessário acessar o link e clicar em “Jogar” (Figura 10).

Figura 10: Link para o D-LEARN na Landing



Fonte: elaborado pelo autor.

Após clicar em jogar, uma Modal para escolher jogar com um usuário autenticado ou não, é mostrada. Caso a opção “Jogar como Guest” for selecionada, o jogo irá iniciar imediatamente (Figura 11). Caso contrário, o sistema irá solicitar as credenciais do jogador para realizar o login e oferecer a opção de criar a conta também.

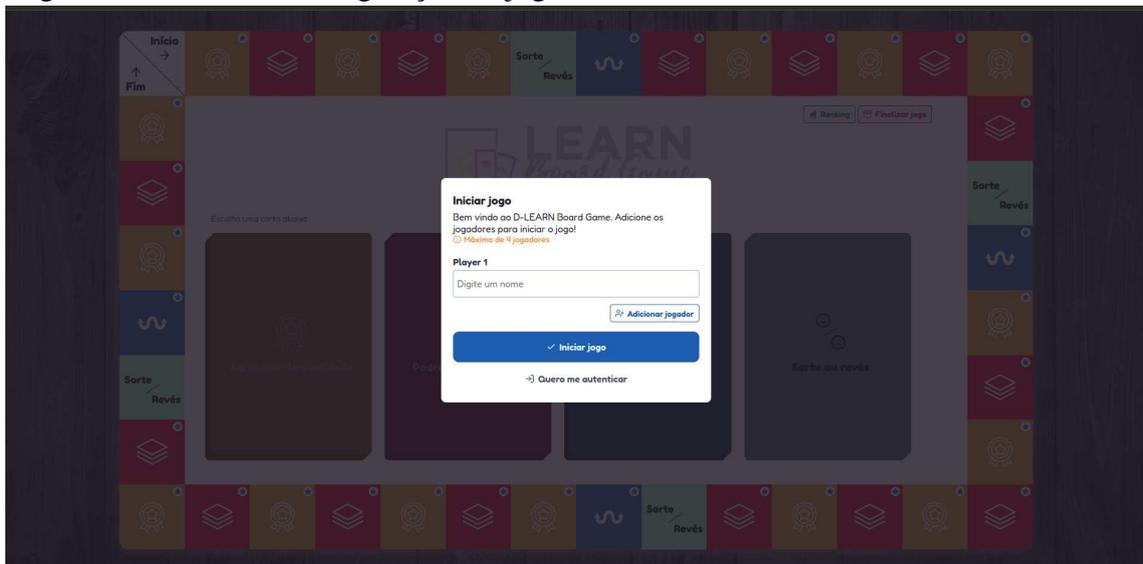
Figura 11: Modal de escolha do modo do jogo



Fonte: elaborado pelo autor.

Após essa fase, o jogo solicitará os nomes dos jogadores que irão jogar a partida. Em uma Modal, será possível adicionar jogadores até um máximo de 4 (quatro) jogadores. Além disso, um botão para autenticar-se é mostrado, a fim de oferecer esse fluxo ao usuário (Figura 12).

Figura 12: Modal de configuração do jogo



Fonte: elaborado pelo autor.

Ao informar os jogadores, o jogo é iniciado e os pinos dos jogadores são adicionados na casa inicial do jogo. Além disso, é informado quem será o primeiro jogador a puxar uma carta (Figura 13)

O jogador que possui a vez pode puxar uma carta de qualquer pilha de cartas (já que ele está na casa inicial do jogo). Ao abrir uma carta, é mostrado primeiramente o tipo da carta e com um efeito de transição é mostrado o conteúdo da mesma. Na Figura 14 é possível ver a disposição dos elementos da carta.

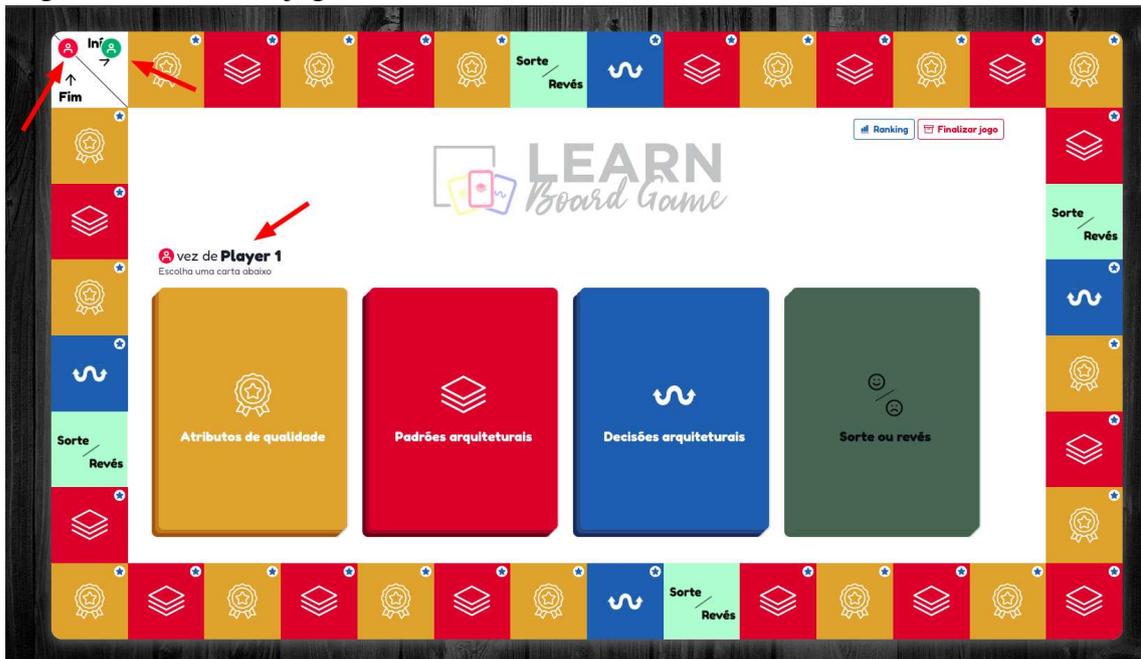
Ao responder errado, um feedback é mostrado ao jogador e um botão para encerrar a jogada e passar para o próximo é mostrado (Figura 15).

Ao responder corretamente, um feedback informando o acerto e um botão para avançar a quantidade de casas correspondente à quantidade de estrelas contida na carta é mostrado (Figura 16).

Após encerrar a jogada, o próximo jogador irá jogar e assim por diante. Caso um dos jogadores “caia” em uma casa de sorte ou revés, a carta é aberta automaticamente evidenciando se o jogador obteve sorte ou revés (Figura 17).

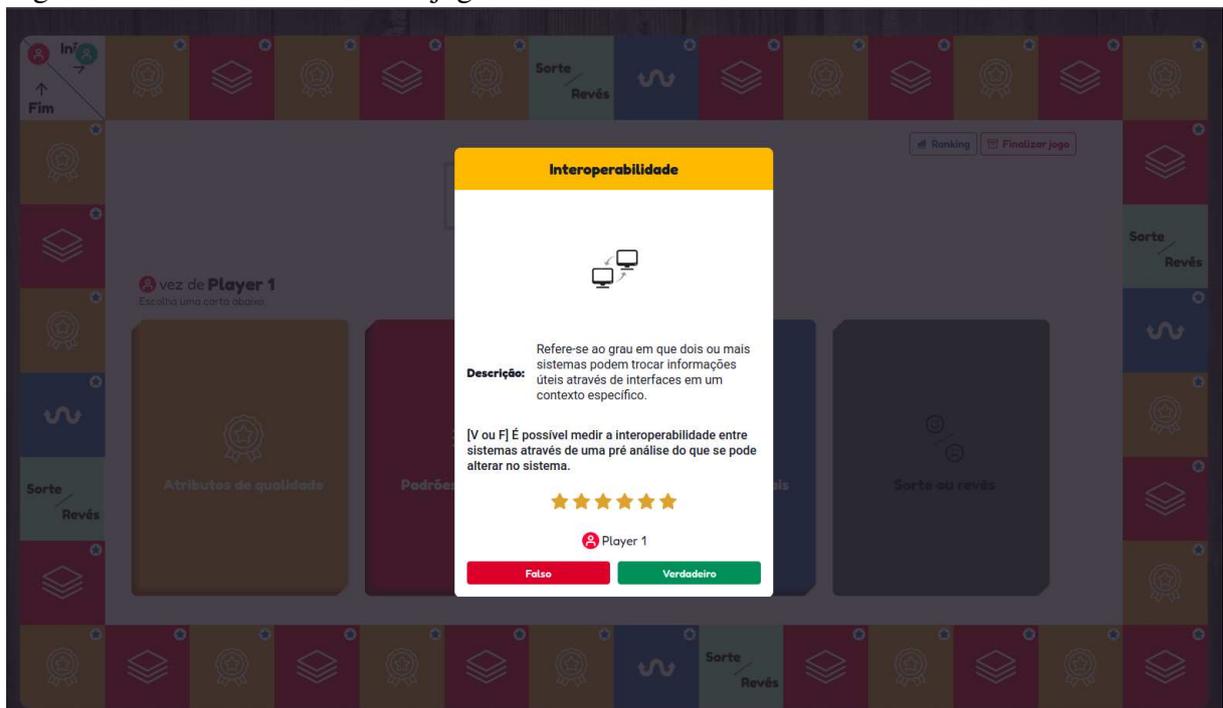
O jogo segue até que um jogador consiga dar uma volta no tabuleiro ou que a opção

Figura 13: Início do jogo



Fonte: elaborado pelo autor.

Figura 14: Carta selecionada no jogo

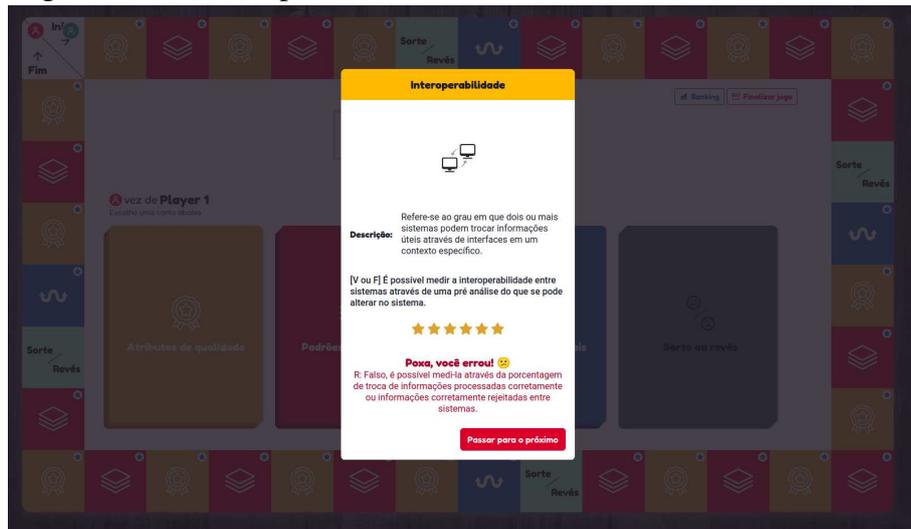


Fonte: elaborado pelo autor.

de encerrar o jogo seja utilizada. Em ambos os casos, uma Modal mostrando o Ranking do jogo será mostrada, como mostra a Figura 18.

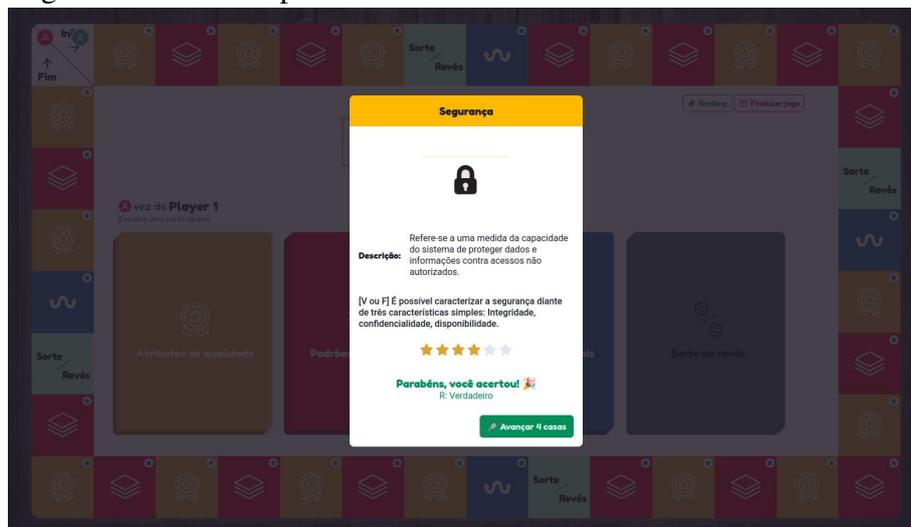
Nessa Modal, é possível visualizar o Ranking dos jogadores na partida por meio de suas pontuações. Além disso, um link para preencher uma pesquisa de *feedback* e um botão para reiniciar o jogo são mostrados.

Figura 15: Carta respondida errada



Fonte: elaborado pelo autor.

Figura 16: Carta respondida certa



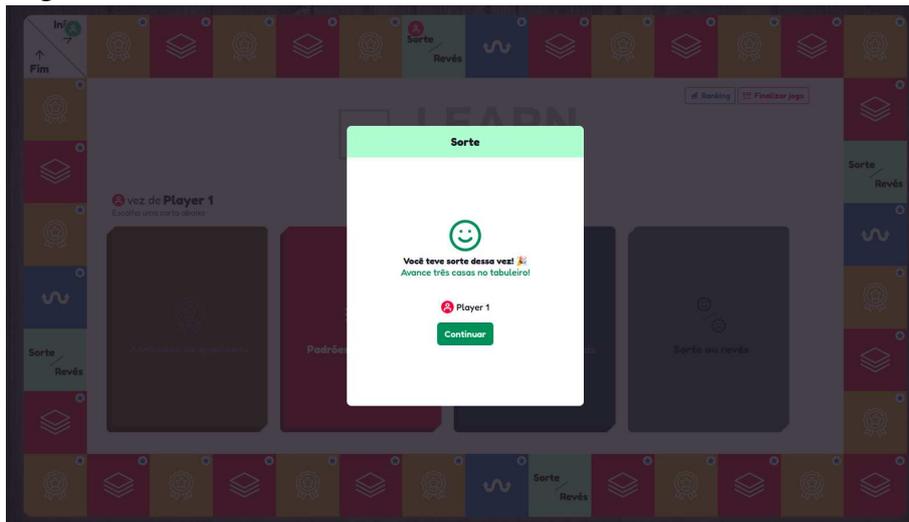
Fonte: elaborado pelo autor.

### 7.3 Painel administrativo do D-LEARN

Um dos principais requisitos para este projeto, era a possibilidade de criar, editar ou remover cartas de maneira simples. Para isso, como falado anteriormente no Capítulo 5, foi criado um painel administrativo para manter essas cartas. Tal painel é acessado somente por meio de um login. Na seção 5.2.4 do capítulo 5 pode ser visualizado a página de login. No entanto, ela pode ser acessada através do link: <https://d-learn.vercel.app/admin/login>.

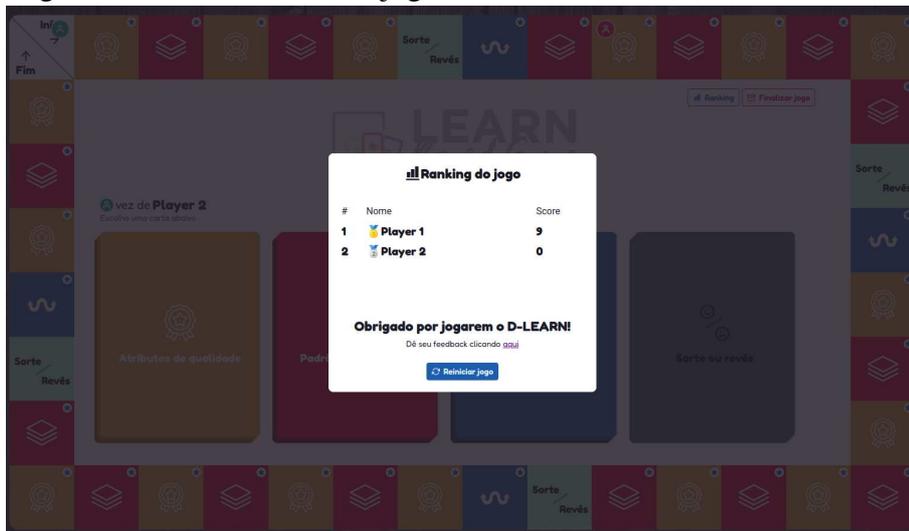
Para o painel do D-LEARN, um simples design foi utilizado. Um menu lateral contendo os menus de “Cartas” e “Jogos”. Nas figuras 19 e 20 podem ser visualizadas essas telas.

Figura 17: Carta de sorte ou revés



Fonte: elaborado pelo autor.

Figura 18: Modal de fim de jogo



Fonte: elaborado pelo autor.

Na Figura 21, pode ser visualizado a Modal para cadastrar uma nova carta, onde os campos Tipo, Nome, Descrição, Pergunta, Resposta, Explicação da resposta e quantidade de estrelas são solicitados. Além desses campos, é possível anexar uma imagem para a carta.

As demais ações dessa tela são para editar e visualizar uma carta. Como citado anteriormente na seção 5.2.4 do capítulo 5, as cartas de sorte ou revés são cadastradas localmente, no entanto, elas podem ser visualizadas normalmente no painel, contudo, sem a opção de editar e excluir.

Figura 19: Página de CRUD de cartas do painel do D-LEARN

#	Imagem	Tipo	Título	Estrelas	Pergunta	Solução	Ações
1		2	Repositório	4	[V ou F] Em repositório, os dados estão centr...	F	
2		0	Binding Time	2	[V ou F] Ao tomar uma decisão binding time é...	V	
3		1	Disponibilidade	4	[V ou F] Não se pode mensurar a disponibilid...	F	
4		0	Modelo de coordenação	4	[V ou F] As decisões de modelo de coordenaç...	F	
5		0	Alocação de responsabilidades	4	[V ou F] A alocação de responsabilidades det...	F	
6		2	Map-Reduce	2	[V ou F] O Map-Reduce facilita tanto a leitura...	V	
7		0	Modelo de dados	4	[V ou F] O modelo de dados determina todos...	F	

Fonte: elaborado pelo autor.

Figura 20: Página de listagem de jogos

#	Vencedor	Score do vencedor	Início	Fim	Tempo de jogo	Host do jogo
1	Luis Otávio	120	18/05/2023 às 15:00	18/05/2023 às 15:30	30m	luisotavio756
2	Luis Otávio	100	18/05/2023 às 15:00	18/05/2023 às 15:30	30m	luisotavio756
3	Luis Otávio	100	18/05/2023 às 15:00	18/05/2023 às 15:30	30m	luisotavio756

Fonte: elaborado pelo autor.

Figura 21: Modal de criação de carta

**Nova carta**

**Tipo**  
Selecione um tipo

**Título**  
Digite um título

**Descrição**  
Digite uma descrição

**Pergunta**  
Crie uma pergunta de V ou F

**Resposta**  
Selecione uma resposta

**Explicação de Resposta**  
Descreva a resposta

**Estrelas**  
Selecione uma quantidade de estrelas

Fonte: elaborado pelo autor.

## 8 AVALIAÇÃO DO D-LEARN

Este capítulo tem o intuito de mostrar alguns dos resultados do presente trabalho.

### 8.1 Planejamento e condução da avaliação

Para a aplicação do D-LEARN com os estudantes, foi necessário a preparação de uma dinâmica de jogo, que foi conduzida pelo autor deste trabalho e pela professora da disciplina de Arquitetura de *Software*. A turma do semestre 2023.1 de Arquitetura de *Software* foi a escolhida para a primeira dinâmica envolvendo o jogo. Para esta dinâmica, foi solicitado que a turma de aproximadamente 36 alunos se dividissem em grupos de 4 pessoas e que um participante de cada equipe levasse um *notebook* para realização da atividade em grupo. Toda a dinâmica foi realizada em uma sala de metodologias ativas na Universidade Federal do Ceará (UFC) Campus Russas.

Figura 22: Aplicação do D-LEARN em sala de aula



Fonte: elaborado pelo autor.

Primeiramente, foi realizada uma apresentação com intuito de explicar o D-LEARN para os participantes. Nessa apresentação, foi explicado como o jogo funciona, seus papéis e regras. Após a apresentação, o site do jogo foi acessado pelas equipes que estavam presentes na sala e configurado com os nomes dos participantes de cada grupo.

Após isso, a dinâmica do jogo se iniciou. O autor deste trabalho e a professora da disciplina ficaram de prontidão para sanar possíveis dúvidas e ajudar em eventuais problemas. No geral, o andamento do jogo foi satisfatório e todas as equipes conseguiram completar o jogo, ou seja, houve um entendimento de como o jogo funcionava por parte dos estudantes. Além disso, todas as equipes conseguiram dar uma volta no tabuleiro (algo que não havia acontecido ainda com o LEARN) e a dinâmica do jogo se deu de maneira fluida e mais rápida quando comparada à aplicação do jogo em formato não-digital.

Após finalização do jogo, um *feedback* foi solicitado aos estudantes, a fim de coletar informações sobre a experiência e qualidade do jogo. Foi utilizado o questionário MEEGA+ para realização da pesquisa com os estudantes da turma de Arquitetura de Software da UFC Campus Russas, no semestre letivo de 2023.1. O questionário foi respondido por 33 estudantes.

No MEEGA+ são utilizados dois fatores de qualidade: usabilidade e experiência do jogo. Cada fator contém seus próprios atributos de análise de qualidade, como pode ser observado na Tabela 5. Por meio desses dois fatores, as análises foram realizadas sobre as perguntas feitas no questionário, que serão melhor descritas a seguir.

Tabela 5: Fatores de qualidade do MEEGA+.

Usabilidade	<ol style="list-style-type: none"> <li>1. Estética</li> <li>2. Aprendizibilidade</li> <li>3. Operabilidade</li> <li>4. Acessibilidade</li> </ol>
Experiência do jogo	<ol style="list-style-type: none"> <li>1. Atenção focada</li> <li>2. Diversão</li> <li>3. Desafio</li> <li>4. Interação social</li> <li>5. Confiança</li> <li>6. Relevância</li> <li>7. Satisfação</li> <li>8. Percepção de Aprendizagem</li> </ol>

Fonte: elaborado pelo autor.

As respostas do questionário podem ser acessadas por meio do link: MEEGA+ RESPOSTA. A planilha também pode ser encontrada no repositório do D-LEARN, por meio do

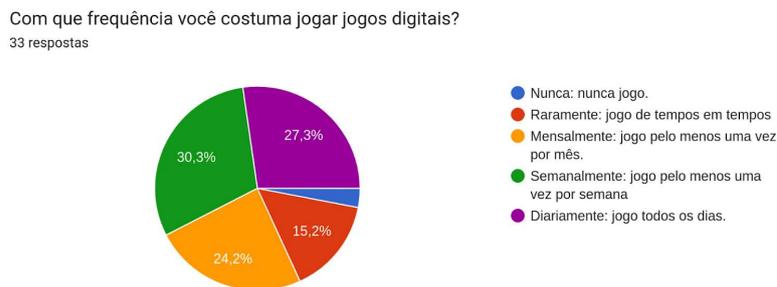
link: Github — meegaplus.

## 8.2 Informações demográficas dos estudantes

Com a aplicação do questionário finalizada, pode-se capturar algumas informações demográficas dos participantes. Em relação ao perfil dos estudantes, 97% responderam que têm entre 18 e 28 anos, enquanto 3% (1 respondente) respondeu que tem mais de 28 anos e 78,8% responderam que são do gênero masculino, enquanto 21,2% responderam que são do gênero feminino.

Quando questionados sobre a assiduidade em relação ao contato com jogos digitais, 57,6% dos estudantes responderam que jogam diariamente ou pelo menos uma vez por semana algum jogo digital, enquanto 15,2% responderam que raramente jogam e 3% (1 respondente) respondeu que nunca joga (Figura 23).

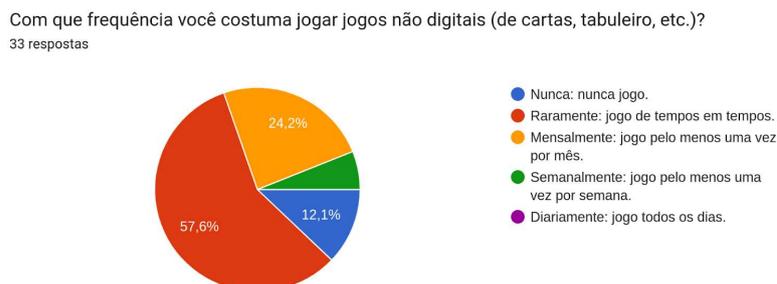
Figura 23: Pergunta sobre frequência em jogos digitais



Fonte: elaborado pelo autor.

Quando questionados sobre a assiduidade em relação a jogos não-digitais (cartas, tabuleiros e etc.) 69,7% dos estudantes responderam que jogam raramente ou nunca jogam e 30,3% responderam que jogam mensalmente ou semanalmente (Figura 24).

Figura 24: Respostas sobre frequência de uso de jogos não-digitais



Fonte: elaborado pelo autor.

Quando perguntados sobre a experiência dos estudantes em relação a outros jogos de tabuleiro voltados para o ensino, 60,6% dos estudantes responderam que já haviam jogado (Figura 25).

Figura 25: Respostas sobre experiência em jogos de tabuleiro voltados para o ensino



Fonte: elaborado pelo autor.

### 8.3 Informações sobre a usabilidade do D-LEARN

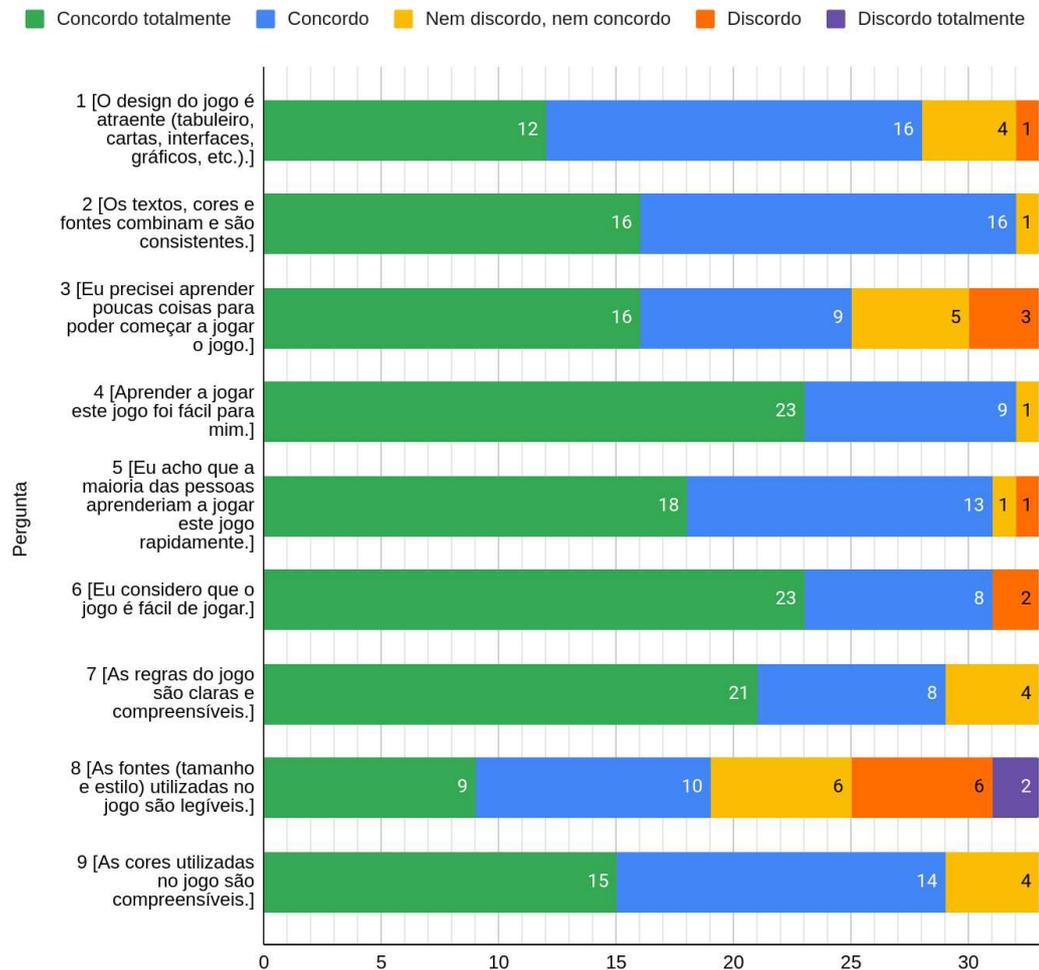
Para avaliação da usabilidade do D-LEARN, foram utilizados os atributos de usabilidade definidos pelo MEEGA+, como foi descrito na seção 5.3 do capítulo 5. Com isso, algumas perguntas foram realizadas a fim de coletar informações acerca da usabilidade do jogo, sendo elas:

1. O design do jogo é atraente (tabuleiro, cartas, interfaces, gráficos, etc.).
2. Os textos, cores e fontes combinam e são consistentes.
3. Eu precisei aprender poucas coisas para poder começar a jogar o jogo.
4. Aprender a jogar este jogo foi fácil para mim.
5. Eu acho que a maioria das pessoas aprenderiam a jogar este jogo rapidamente.
6. Eu considero que o jogo é fácil de jogar.
7. As regras do jogo são claras e compreensíveis.
8. As fontes (tamanho e estilo) utilizadas no jogo são legíveis.
9. As cores utilizadas no jogo são compreensíveis.

Na Figura 26 é possível analisar a quantidade de respostas para cada ponto acima.

Figura 26: Respostas sobre experiência em jogos de tabuleiro voltados para o ensino

## Usabilidade do jogo D-LEARN Board Game



Fonte: elaborado pelo autor.

Quanto à **estética** (1 e 2), a maioria dos participantes (28) responderam que o jogo é atraente. Mas cinco discordam que as fontes e as cores do texto são consistentes.

Em relação à **aprendizabilidade** (3 à 5), a maioria (25) concordou que foi preciso aprender poucas coisas para dar início ao jogo, enquanto 3 respondentes sinalizaram que não concordam. Quando perguntados se o D-LEARN foi um jogo fácil de aprender, a grande maioria (32) concordou que o jogo foi fácil de aprender, enquanto 1 respondente não concordou. Quando perguntados sobre a facilidade de outra pessoa aprender a jogar o jogo, novamente a grande maioria (31) concordou, enquanto 2 respondentes responderam que não concordaram.

No tocante a **operabilidade** do jogo (6 e 7), a grande maioria (31) concordou que o jogo é fácil de jogar, enquanto 2 responderam que não concordam. E quando perguntados

se as regras do jogo são fáceis e compreensíveis, a maioria (29) concordou, enquanto 4 nem discordaram e nem concordaram.

Quanto à **acessibilidade** do jogo (8 e 9), a grande parte (19) concordou que as fontes e tamanhos são legíveis, enquanto 8 responderam que não concordam. Quando perguntados sobre as cores do jogo, a grande maioria (29) respondeu que concorda.

#### **8.4 Informações sobre experiência de usuário do D-LEARN**

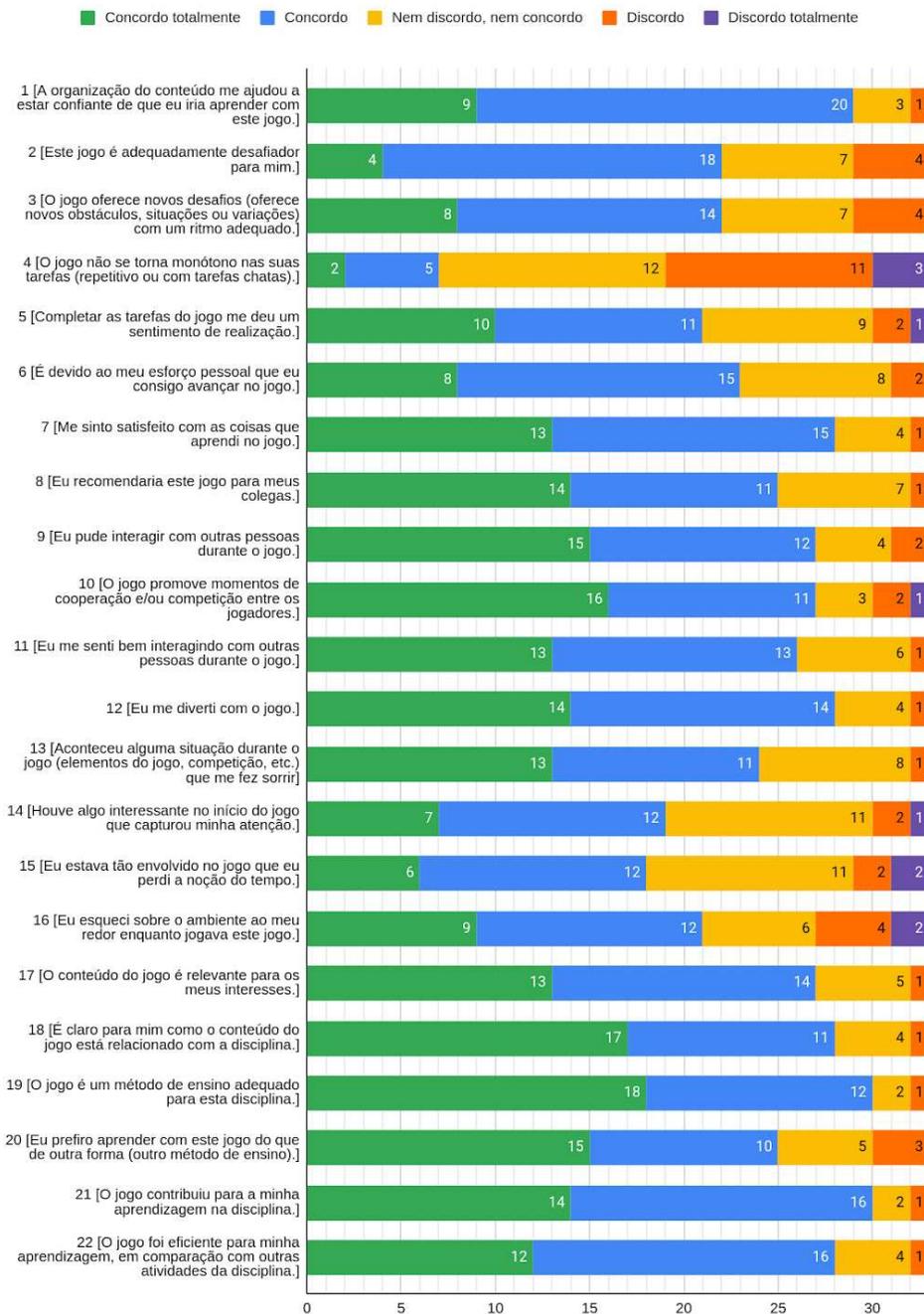
Para avaliação da experiência de jogo do D-LEARN, foram utilizados os atributos de experiência de jogo definidos pelo MEEGA+, como foi descrito na seção 5.3 do capítulo 5. Com isso, algumas perguntas foram realizadas a fim de coletar informações acerca da experiência do jogo, sendo elas:

1. A organização do conteúdo me ajudou a estar confiante de que eu iria aprender com este jogo.
2. Este jogo é adequadamente desafiador para mim.
3. O jogo oferece novos desafios (oferece novos obstáculos, situações ou variações) com um ritmo adequado.
4. O jogo não se torna monótono nas suas tarefas (repetitivo ou com tarefas chatas).
5. Completar as tarefas do jogo me deu um sentimento de realização.
6. É devido ao meu esforço pessoal que eu consigo avançar no jogo.
7. Me sinto satisfeito com as coisas que aprendi no jogo.
8. Eu recomendaria este jogo para meus colegas.
9. Eu pude interagir com outras pessoas durante o jogo.
10. O jogo promove momentos de cooperação e/ou competição entre os jogadores.
11. Eu me senti bem interagindo com outras pessoas durante o jogo.
12. Eu me diverti com o jogo.
13. Aconteceu alguma situação durante o jogo (elementos do jogo, competição, etc.) que me fez sorrir.
14. Houve algo interessante no início do jogo que capturou minha atenção.
15. Eu estava tão envolvido no jogo que eu perdi a noção do tempo.
16. Eu esqueci sobre o ambiente ao meu redor enquanto jogava este jogo.
17. O conteúdo do jogo é relevante para os meus interesses.
18. É claro para mim como o conteúdo do jogo está relacionado com a disciplina.

19. O jogo é um método de ensino adequado para esta disciplina.
20. Eu prefiro aprender com este jogo do que de outra forma (outro método de ensino).
21. O jogo contribuiu para a minha aprendizagem na disciplina.
22. O jogo foi eficiente para minha aprendizagem, em comparação com outras atividades da disciplina.

Figura 27: Dados sobre a experiência de usuário do D-LEARN

EXPERIÊNCIA DO JOGADOR com o jogo D-LEARN Board Game



Fonte: elaborado pelo autor.

Na Figura 27 é possível analisar a quantidade de respostas para cada ponto acima.

Em relação à **confiança** (1), a maioria dos participantes (29) concordaram que a organização do jogo os trouxe confiança, enquanto 4 responderam que não concordam.

Em relação ao **desafio** (2 à 4), a grande parte (22) concordou que o jogo é desafiador, enquanto 11 não concordaram. Quando perguntado sobre a dinamicidade dos desafios no jogo, a grande parte (22) concordou, enquanto 11 não concordaram. Quando perguntados se o jogo não se torna monótono no decorrer da dinâmica, 14 responderam que acham que o jogo se torna monótono, enquanto 7 responderam que o jogo não se torna.

Em relação à **satisfação** (5 a 8), 21 indicaram que a conclusão de tarefas proporcionou sensação satisfatória de realização, enquanto 9 se mostraram neutros. Na afirmativa “É devido ao meu esforço pessoal que consegui avançar no jogo”, oito respondentes se mostraram neutros enquanto 2 não concordaram. Quando questionados sobre a possibilidade de indicar o jogo a outro colega, a maioria (25) respondeu que concorda, enquanto 7 se mostraram neutros.

Em relação à **interação social** (9 a 11), a afirmação "Consegui interagir com outras pessoas durante o jogo" teve 27 que concordaram, enquanto 2 pessoas discordaram. Quando perguntados se o jogo promove a cooperação e/ou competição entre os jogadores, 27 responderam que concordam, enquanto 3 responderam que não concordam. Por fim, quando perguntados sobre a sensação de prazer ao interagir com outros jogadores durante o jogo, 26 responderam que concordam, enquanto 6 se mostraram neutros.

No que diz respeito à **diversão** (perguntas 12 a 13), 28 participantes concordaram que se divertiram durante o jogo e 14 responderam que houve momentos durante o jogo que os fizeram sorrir, enquanto 8 se mostraram neutros.

Em relação à **concentração** (perguntas 14 a 16), três pessoas discordaram que algo no início do jogo tenha chamado a atenção, e 18 afirmaram que perderam a noção do tempo devido ao alto envolvimento com o jogo. Por outro lado, 21 participantes concordaram que se esqueceram do ambiente ao redor enquanto jogavam.

Quanto à **relevância** (perguntas 17 a 20), a afirmação "O conteúdo do jogo desperta meu interesse" recebeu o maior número de respostas neutras (5 respostas). No entanto, 27 concordaram que o conteúdo do jogo está relacionado com a disciplina estudada. 30 estudantes concordaram que o jogo é um método apropriado para o ensino, e 25 preferiram aprender utilizando o LEARN em comparação com outros métodos, enquanto 3 não concordaram.

No tocante à **percepção de aprendizagem** (perguntas 21 e 22), 30 participantes

afirmaram que o jogo contribuiu para o aprendizado na disciplina, e a maioria deles (28) considerou que o jogo proporcionou uma forma eficiente de aprendizado em comparação com outras atividades práticas relacionadas à disciplina.

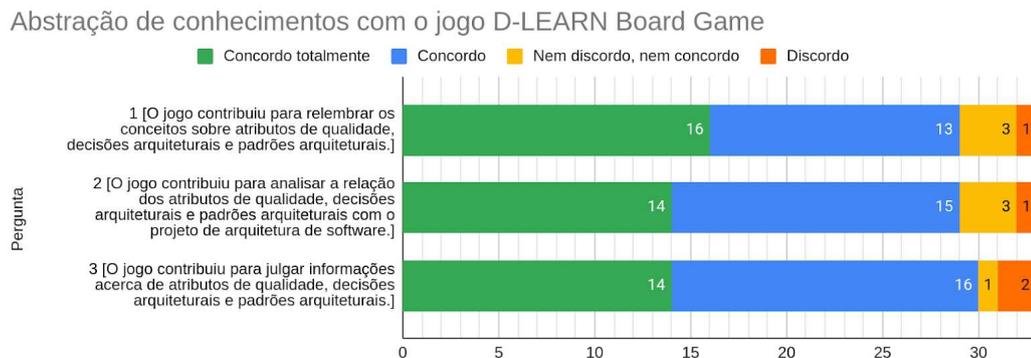
## 8.5 Percepção de aprendizagem sobre Arquitetura de Software

No MEEGA+, é indicado que se adicione até 3 perguntas sobre o conteúdo abordado no jogo com intuito também de analisar a percepção de aprendizado dos respondentes após a dinâmica do jogo. Essas perguntas foram:

1. O jogo contribuiu para lembrar os conceitos sobre atributos de qualidade, decisões arquiteturais e padrões arquiteturais.
2. O jogo contribuiu para analisar a relação dos atributos de qualidade, decisões arquiteturais e padrões arquiteturais com o projeto de arquitetura de software.
3. O jogo contribuiu para julgar informações acerca de atributos de qualidade, decisões arquiteturais e padrões arquiteturais.

Na Figura 28 é possível observar os resultados obtidos.

Figura 28: Dados sobre o aprendizado de arquitetura de software após jogar o jogo



Fonte: elaborado pelo autor.

No que diz respeito ao **aprendizado sobre atributos de qualidade, padrões arquiteturais e decisões arquiteturais** (perguntas 1 a 3), a grande maioria respondeu que o jogo contribuiu para lembrar, analisar e julgar conteúdos relacionados a esses atributos.

## 8.6 Respostas abertas do questionário

No MEEGA+, também há espaços para informar de maneira subjetiva o que o respondente mais gostou no jogo e o que poderia ser melhorado, ou seja, os pontos fortes e a

serem melhorados no jogo. Algumas dessas respostas foram sobre o mesmo assunto, logo, foram agrupadas e selecionadas. Na Tabela 6 é possível visualizar os principais comentários feitos pelos estudantes.

Tabela 6: Fatores analisados no formulário MEEGA+

Pontos fortes	
Comentário 1	“A forma de relacionar os conceitos e a teoria aprendida em sala de uma forma lúdica e competitiva” - P17
Comentário 2	“A automatização dos elementos. Além de que é possível aprender de forma dinâmica com o jogo” - P30
Comentário 3	“A junção do conteúdo visto em sala com o jogo, foi bom para revisar e atender melhor os conceitos.” - P13
Comentário 4	“Fácil jogabilidade, intuitivo e instigante.” - P7
Comentário 5	“A forma em que o conteúdo da disciplina é destrinchado, facilitando o aprendizado com uma forma divertida e dinâmica.” - P32
Sugestões de melhorias	
Comentário 6	“O tamanho e as fontes das cartas, além da tela do tabuleiro” - P6
Comentário 7	“A organização das casas/pontuação para evitar cair sempre nas casas de mesma cor” - P15
Comentário 8	“Acho que adicionar mais cartas e talvez adicionar um cronômetro para cada pergunta seria interessante” - P20
Comentário 9	“Ter mais cartas e ter a ação de jogar online em dispositivos diferentes” - P27
Comentário 10	“Mais perguntas por categoria” - P24

Em relação aos pontos fortes do jogo (comentários 1 a 5), foi possível notar a aceitação da dinâmica do jogo, bem como perceber que o jogo contribuiu para o aprendizado do conteúdo visto em aula. Além disso, comentários sobre a automatização e design do jogo foram citados.

No tocante às sugestões de melhoria (comentários 6 a 10), pontos sobre o tamanho e fontes do jogo (comentário 6) foram citados e pontos relacionados à organização do tabuleiro e da pontuação de cada carta para evitar a repetição de cartas e casas também foram citados (comentário 7). Além disso, novas *features* foram sugeridas como um adicionar um cronômetro para cada pergunta (comentário 8) e possibilidade de jogar em dispositivos diferentes de maneira remota (comentário 9) foram mencionados.

## 8.7 Nota de qualidade do D-LEARN segundo o cálculo do MEEGA+

Conforme indicado por Petri (PETRI et al., 2018), após a coleta dos resultados é necessário rodar um script para gerar uma nota de qualidade. Após preparar o ambiente e rodar esse script, o D-LEARN obteve uma nota de 84,7, o que representa um resultado excelente de acordo com as avaliações do MEEGA+. Os arquivos utilizados para execução do script, bem como a planilha de análise de dados estão disponíveis no link: Github MEEGA+

## 8.8 Validação com a professora da disciplina

Além da validação junto aos estudantes, uma validação foi feita com a professora da disciplina de arquitetura. Para essa validação, o jogo foi deixado um pouco de lado, tendo em vista que os estudantes foram os *stakeholders* principais desta parte. Na validação com a professora da disciplina, foi analisado um painel administrativo criado para manter as cartas do D-LEARN, sendo possível visualizar, criar, editar e excluir. A validação foi feita e aprovada pela professora, logo, no tocante ao painel administrativo do D-LEARN, não houveram mudanças vindas de sugestões da professora.

## 8.9 Desenvolvimento das melhorias

Após coleta do *feedback* dos participantes da dinâmica do jogo, uma análise foi feita dos pontos positivos e sugestões de melhorias do D-LEARN. Pontos como: “fontes do jogo pequenas”, “repetição de cartas e casas do tabuleiro”, “pinos pequenos” e “problemas de responsividade” foram alguns dos pontos mencionados pelos estudantes no questionário MEEGA+ respondido por eles ao fim da dinâmica. Alguns outros pontos de melhoria e positivos são descritos no capítulo 7.

Após análise das respostas, foi notado alguns pontos de melhoria relacionados à melhoria de *design* do jogo, como fontes, tamanhos, cores entre outros. Com isso, foi desenvolvido um *patch* de correções que gerou uma segunda versão do D-LEARN com as sugestões de melhorias que eram viáveis no momento implementadas. No entanto, vale salientar que algumas sugestões de melhorias não serão aplicadas neste momento, devido sua necessidade de uma pesquisa mais aprofundada acerca do problema, como por exemplo a melhorias da distribuição de casas no tabuleiro e de estrelas nas cartas a fim de evitar muitas repetições de casas no tabuleiro do jogo.

Portanto, o D-LEARN obteve uma nova versão após aplicação em sala de aula, de acordo com as sugestões de melhorias que eram viáveis serem desenvolvidas no momento, que foram elencadas de acordo com uma análise feita junto com a professora da disciplina de arquitetura.

## 9 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho teve como objetivo construir uma versão digital do D-LEARN, para auxiliar docentes e discentes no processo de ensino de arquitetura de software. Para isso, foi adotada a metodologia DSR e conduzido as etapas de investigação do problema, desenvolvimento e validação. Na etapa de investigação, foram realizadas atividades de pesquisa, com intuito de entender os principais pontos de melhoria do LEARN e obter insights de trabalhos relacionados. No desenvolvimento, foi construída toda parte de análise e projeto do sistema como requisitos e design da arquitetura, seguida da construção em si do projeto. Por fim, na validação foi realizado uma aplicação do jogo em sala de aula e uma coleta de dados por meio do questionário MEEGA+ para obter uma nota de qualidade.

Um dos objetivos específicos deste trabalho era melhorar o grau de satisfação e aprendizado do LEARN, por meio de uma versão digital. Para medir o grau de qualidade do LEARN, foi utilizado um cálculo sugerido por Petri *et al.* (2018) e o resultado foi uma nota que classificou o jogo como bom, de 45,6. Para que o D-LEARN pudesse ser comparado ao LEARN, o mesmo questionário foi utilizado como já mencionado, e O D-LEARN obteve um resultado satisfatório em sua primeira dinâmica e uma nota de qualidade considerada excelente de acordo com o cálculo realizado com os dados provenientes do questionário MEEGA+. Portanto, o objetivo de atingir uma qualidade maior que o jogo não-digital foi atingido.

Mesmo com uma nota considerada excelente, alguns pontos de melhoria precisam ser implementados para atingir um nível ainda maior de satisfação e aprendizado por parte dos estudantes. Os principais pontos mencionados foram de melhoria na distribuição das casas no tabuleiro e cartas do jogo, novas funcionalidades como cronômetro para cada pergunta e possibilidade de jogar remotamente. Além disso, alguns outros pontos da esfera de *design* foram citados, como aumento do tamanho das fontes, adaptação melhor para diferentes dispositivos e aumento no tamanho de alguns componentes do jogo para uma melhor visualização. Esses pontos relacionados ao *design* da aplicação foram implementados em um patch de correções logo após à aplicação do jogo, enquanto os pontos relacionados a novas funcionalidades irão ficar como melhorias futuras.

O jogo ficará disponível na web mesmo depois deste artigo e será adotado como padrão na disciplina de arquitetura de software, logo, o LEARN (jogo não-digital) deixará de ser utilizado nas turmas futuras. O LEARN não deixará de existir, uma vez que o D-LEARN é uma versão digital do LEARN, no entanto, para futuras melhorias, o D-LEARN se encaixa

melhor para realização de testes e provas de conceito, uma vez que o tempo de atualização do D-LEARN é somente o tempo de alteração no código e implantação no servidor, enquanto no jogo não-digital é necessário uma nova confecção de cartas e tabuleiros.

Além disso, o jogo também poderá ser acessado por pessoas fora da universidade, uma vez que o jogo está disponível na web, qualquer pessoa pode acessá-lo. Na *Landing Page* do D-LEARN é possível visualizar o *Ranking global*, que deixará registrado as partidas jogadas pelos usuários que optaram por entrar no jogo. Esses dados ficarão armazenados e poderão servir para estudos e análises futuras para os próximos pesquisadores que atuarem em melhorias no D-LEARN.

No tocante ao código, será disponibilizado de forma *Open Source* (Código livre) na plataforma Github, sendo possível assim que qualquer pessoa faça uma cópia do jogo ou contribua em melhorias. Por meio disso, pode ser possível que outras pessoas desejem adaptar a estrutura do D-LEARN para jogos de outras disciplinas, como fundamentos de programação, banco de dados entre outras, uma vez que seria necessário somente alterar quais cartas o jogo disponibilizaria e fazer algumas alterações para que essa cópia do jogo se adequasse ao novo contexto e não mais ao de Arquitetura de Software. O link do repositório do *Frontend* do D-LEARN pode ser acessado em: link do *frontend*, enquanto o link do repositório do *Backend* do D-LEARN, acessado em link do *backend*.

Em relação aos trabalhos futuros, as sugestões de melhorias que não foram aplicadas no presente trabalho, podem ser desenvolvidas como trabalhos futuros, tais como melhoria na distribuição de casas no tabuleiro e estrelas das cartas - a fim de evitar repetições desnecessárias nas mesmas casas, possibilidade de jogar remotamente com outros jogadores e adaptação completa para dispositivos celulares. Dentre essas sugestões, o mais citado pelos respondentes do questionário foi a melhoria na configuração do tabuleiro e das cartas, logo, seria um bom ponto a ser estudado e implementado por outro pesquisador.

Além das sugestões de melhoria feitas pelos alunos, algumas outras funcionalidades poderiam ser implementadas para trazer um senso de pertencimento por parte dos colaboradores, como por exemplo a possibilidade de um estudante/usuário submeter uma carta para ser adicionada ao jogo, onde tal submissão seria analisada por um usuário administrador para uma eventual aprovação ou rejeição. Isso poderia trazer uma maior escalabilidade na disponibilidade de cartas, maior engajamento dos usuários - uma vez que seriam dados créditos a esse criador, e aumento das possibilidades de jogo no D-LEARN. Outros pontos relacionados ao contexto de

jogos na Web também poderiam ser implementados, como efeitos de transição entre elementos, cronômetro de tempo para cada pergunta, efeitos sonoros e/ou visuais, entre outros.

## REFERÊNCIAS

- BASS, L. The software architect and devops. **IEEE Software**, IEEE, v. 35, n. 1, p. 8–10, 2018.
- BOER, R. C. D.; LAGO, P.; VERDECCHIA, R.; KRUCHTEN, P. Decidarch v2: An improved game to teach architecture design decision making. In: IEEE. **2019 IEEE International Conference on Software Architecture Companion (ICSA-C)**. [S. l.], 2019. p. 153–157.
- CERVANTES, H.; HAZIYEV, S.; HRYTSAY, O.; KAZMAN, R. Smart decisions: an architectural design game. In: **Proceedings of the 38th International Conference on Software Engineering Companion**. [S. l.: s. n.], 2016. p. 327–335.
- FORMIGHIERI, G. **Pirâmide de Aprendizagem de William Glasser**. 2022. Disponível em: <https://keeps.com.br/piramide-de-aprendizagem-de-william-glasser-conceito-e-estrutura>.
- GARCIA, I.; PACHECO, C.; MÉNDEZ, F.; CALVO-MANZANO, J. A. The effects of game-based learning in the acquisition of “soft skills” on undergraduate software engineering courses: A systematic literature review. **Computer Applications in Engineering Education**, Wiley Online Library, v. 28, n. 5, p. 1327–1354, 2020.
- GEERTS, G. L. A design science research methodology and its application to accounting information systems research. **International journal of accounting Information Systems**, Elsevier, v. 12, n. 2, p. 142–151, 2011.
- GRAPHO GAME. **GraphoGame**. 2023. Acesso em: 13 de fev. de 2023. Disponível em: <https://graphogame.com>.
- HEMMERT, F.; MEYER, K.; BRANDIES, T.; EBELING, L.; FIEBIG, M.; HORST, J.; KATSNELSON, A.; KUPFERSCHMIDT, L.; LUDWIG, D.; MARTINS, C. *et al.* Perspectives in play: Printable board games that teach about foreign policy. In: **Extended Abstracts of the 2021 Annual Symposium on Computer-Human Interaction in Play**. [S. l.: s. n.], 2021. p. 287–293.
- HEVNER, A.; CHATTERJEE, S. **Design research in information systems: theory and practice**. [S. l.]: Springer Science & Business Media, 2010. v. 22.
- HEVNER, A. R. A three cycle view of design science research. **Scandinavian journal of information systems**, v. 19, n. 2, p. 4, 2007.
- JABBAR, A. I. A.; FELICIA, P. Gameplay engagement and learning in game-based learning: A systematic review. **Review of educational research**, Sage Publications Sage CA: Los Angeles, CA, v. 85, n. 4, p. 740–779, 2015.
- LUND, A. M. Measuring usability with the use questionnaire<sup>12</sup>. **Usability interface**, v. 8, n. 2, p. 3–6, 2001.
- MEDVIDOVIC, N.; TAYLOR, R. N. Software architecture: foundations, theory, and practice. In: **Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2**. [S. l.: s. n.], 2010. p. 471–472.
- MINISTÉRIO DA EDUCAÇÃO DO BRASIL. **GraphoGame**. 2023. Acesso em: 13 de fev. de 2023. Disponível em: <https://alfabetizacao.mec.gov.br/grapho-game>.

NAVARRO, E. O.; BAKER, A.; HOEK, A. V. D. Teaching software engineering using simulation games. In: **Proceedings of the International Western Simulation Multiconference**. [S. l.: s. n.], 2004.

OLIVEIRA, B. R.; GARCÉS, L.; LYRA, K. T.; SANTOS, D. S.; ISOTANI, S.; NAKAGAWA, E. Y. An overview of software architecture education. In: SBC. **Anais do XXV Congresso Ibero-Americano em Engenharia de Software**. [S. l.], 2022. p. 76–90.

PETRI, G.; WANGENHEIM, C. Gresse von; BORGATTO, A. F. Meega+, systematic model to evaluate educational games. In: \_\_\_\_\_. **Encyclopedia of Computer Graphics and Games**. Cham: Springer International Publishing, 2018. p. 1–7. ISBN 978-3-319-08234-9. Disponível em: [https://doi.org/10.1007/978-3-319-08234-9\\_214-1](https://doi.org/10.1007/978-3-319-08234-9_214-1).

PINTO, F. de S.; SILVA, P. C. Gamification applied for software engineering teaching-learning process. In: **Proceedings of the XXXI Brazilian Symposium on Software Engineering**. [S. l.: s. n.], 2017. p. 299–307.

Presidência da República. **Lei nº 13.709**. 2018. [https://www.planalto.gov.br/ccivil\\_03/\\_ato2015-2018/2018/lei/l13709.htm](https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm). Lei Geral de Proteção de Dados (LGPD).

SALUTES, C. **O que é e como usar o duolingo - Guia completo das funções**. 2022. Disponível em: <https://canaltech.com.br/internet/o-que-e-e-como-usar-o-duolingo/>.

SOUSA, T. A.; MARQUES, A. B. Learn board game: A game for teaching software architecture created through design science research. In: **Proceedings of the XXXIV Brazilian Symposium on Software Engineering**. [S. l.: s. n.], 2020. p. 834–843.

STANDARDIZATION, I. O. for. **ISO/IEC 25010: 2011: Systems and Software Engineering-Systems and Software Quality Requirements and Evaluation (SQuaRE)-System and Software Quality Models**. [S. l.]: ISO/IEC, 2011.

WIERINGA, R. Design science methodology: principles and practice. In: **Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2**. [S. l.: s. n.], 2010. p. 493–494.

**APÊNDICE A – PERGUNTAS DO QUESTIONÁRIO MEEGA+**

Perguntas	
<b>Pergunta 1</b>	Qual sua Faixa etária?
<b>Pergunta 2</b>	Qual seu gênero?
<b>Pergunta 3</b>	Com que frequência você costuma jogar jogos digitais?
<b>Pergunta 4</b>	Com que frequência você costuma jogar jogos não digitais (de cartas, tabuleiro, etc.)?
<b>Pergunta 5</b>	Você já havia jogado um jogo de tabuleiro voltado ao ensino?
<b>Pergunta 6</b>	O design do jogo é atraente (tabuleiro, cartas, interfaces, gráficos, etc.)
<b>Pergunta 7</b>	Os textos, cores e fontes combinam e são consistentes.
<b>Pergunta 8</b>	Eu precisei aprender poucas coisas para poder começar a jogar o jogo.
<b>Pergunta 9</b>	Aprender a jogar este jogo foi fácil para mim.
<b>Pergunta 10</b>	Eu acho que a maioria das pessoas aprenderiam a jogar este jogo rapidamente.
<b>Pergunta 11</b>	Eu considero que o jogo é fácil de jogar.
<b>Pergunta 12</b>	As regras do jogo são claras e compreensíveis.
<b>Pergunta 13</b>	As fontes (tamanho e estilo) utilizadas no jogo são legíveis.
<b>Pergunta 14</b>	As cores utilizadas no jogo são compreensíveis.
<b>Pergunta 15</b>	A organização do conteúdo me ajudou a estar confiante de que eu iria aprender com este jogo.
<b>Pergunta 16</b>	Este jogo é adequadamente desafiador para mim.
<b>Pergunta 17</b>	O jogo oferece novos desafios (oferece novos obstáculos, situações ou variações) com um ritmo adequado.
<b>Pergunta 18</b>	O jogo não se torna monótono nas suas tarefas (repetitivo ou com tarefas chatas).
<b>Pergunta 19</b>	Completar as tarefas do jogo me deu um sentimento de realização.

<b>Pergunta 20</b>	É devido ao meu esforço pessoal que eu consigo avançar no jogo.
<b>Pergunta 21</b>	Me sinto satisfeito com as coisas que aprendi no jogo.
<b>Pergunta 22</b>	Eu recomendaria este jogo para meus colegas.
<b>Pergunta 23</b>	Eu pude interagir com outras pessoas durante o jogo.
<b>Pergunta 24</b>	O jogo promove momentos de cooperação e/ou competição entre os jogadores.
<b>Pergunta 25</b>	Eu me senti bem interagindo com outras pessoas durante o jogo.
<b>Pergunta 26</b>	Eu me diverti com o jogo.
<b>Pergunta 27</b>	Aconteceu alguma situação durante o jogo (elementos do jogo, competição, etc.) que me fez sorrir.
<b>Pergunta 28</b>	Houve algo interessante no início do jogo que capturou minha atenção.
<b>Pergunta 29</b>	Eu estava tão envolvido no jogo que eu perdi a noção do tempo.
<b>Pergunta 30</b>	Eu esqueci sobre o ambiente ao meu redor enquanto jogava este jogo.
<b>Pergunta 31</b>	O conteúdo do jogo é relevante para os meus interesses.
<b>Pergunta 32</b>	É claro para mim como o conteúdo do jogo está relacionado com a disciplina.
<b>Pergunta 33</b>	O jogo é um método de ensino adequado para esta disciplina.
<b>Pergunta 34</b>	Eu prefiro aprender com este jogo do que de outra forma (outro método de ensino).
<b>Pergunta 35</b>	O jogo contribuiu para a minha aprendizagem na disciplina.
<b>Pergunta 36</b>	O jogo foi eficiente para minha aprendizagem, em comparação com outras atividades da disciplina.
<b>Pergunta 37</b>	O jogo contribuiu para lembrar os conceitos sobre atributos de qualidade, decisões arquiteturais e padrões arquiteturais.

<b>Pergunta 38</b>	O jogo contribuiu para analisar a relação dos atributos de qualidade, decisões arquitetônicas e padrões arquitetônicos com o projeto de arquitetura de software.
<b>Pergunta 39</b>	O jogo contribuiu para julgar informações acerca de atributos de qualidade, decisões arquiteturais e padrões arquiteturais.
<b>Pergunta 40</b>	O que você mais gostou no jogo?
<b>Pergunta 41</b>	O que poderia ser melhorado no jogo?
<b>Pergunta 42</b>	Gostaria de fazer mais algum comentário?