



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE QUIXADÁ
CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

EDUARDO DA SILVA GOMES

**CODEPLAY: UM JOGO DIGITAL EDUCATIVO QUE ESTIMULA O PENSAMENTO
COMPUTACIONAL NA ELABORAÇÃO DE SOLUÇÕES DE PROBLEMAS**

QUIXADÁ
2023

EDUARDO DA SILVA GOMES

CODEPLAY: UM JOGO DIGITAL EDUCATIVO QUE ESTIMULA O PENSAMENTO
COMPUTACIONAL NA ELABORAÇÃO DE SOLUÇÕES DE PROBLEMAS

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Sistemas de Informação
do Campus de Quixadá da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Jefferson de Carva-
lho Silva

QUIXADÁ

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)

G613c Gomes, Eduardo da Silva.

CODEPLAY: um jogo digital educativo que estimula o pensamento computacional na elaboração de soluções de problemas / Eduardo da Silva Gomes. – 2023.

49 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Sistemas de Informação, Quixadá, 2023.

Orientação: Prof. Dr. Jefferson de Carvalho Silva.

1. Pensamento Computacional. 2. Jogos Educativos. 3. Evasão Universitária. I. Título.

CDD 005

EDUARDO DA SILVA GOMES

CODEPLAY: UM JOGO DIGITAL EDUCATIVO QUE ESTIMULA O PENSAMENTO
COMPUTACIONAL NA ELABORAÇÃO DE SOLUÇÕES DE PROBLEMAS

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Sistemas de Informação
do Campus de Quixadá da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Sistemas de Informação.

Aprovada em: ____/____/____

BANCA EXAMINADORA

Prof. Dr. Jefferson de Carvalho Silva (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Wladimir Araujo Tavares
Universidade Federal do Ceará (UFC)

Prof. Dr. Antonio Rafael Braga
Universidade Federal do Ceará (UFC)

AGRADECIMENTOS

A Deus, por me conceder forças durante toda esta trajetória, por me fazer superar as dificuldades e me fazer amadurecer.

Aos meus pais, por estarem sempre acreditando no meu crescimento pessoal e acadêmico.

À minha namorada Suele, por todo seu apoio e companheirismo, obrigado por acreditar no meu potencial.

Ao Victor Cavalcante, por ter me ajudado, convidando os discentes à participarem do experimento. E à todos os discistes que se colocaram à disposição para participar.

Ao Prof. Dr. Jefferson de Carvalho Silva, por me orientar na minha monografia, por dedicar o seu tempo e todo conhecimento.

A todos os professores do campus da UFC Quixadá, pela disposição de sempre buscar oferecer um ensino superior de qualidade.

RESUMO

Nos últimos anos, as instituições de ensino superior têm enfrentado sérios problemas ocasionados pela evasão acadêmica. Este trabalho tem o objetivo de estimular o pensamento computacional por meio de um jogo digital educativo, com intuito de motivar os discentes a praticar mais sobre programação e propor uma ferramenta capaz de diminuir os índices de evasão e retenção acadêmica. Portanto, foi realizado um levantamento dos recursos de interface, mecânica e interação dos jogos semelhantes para servir de base para o desenvolvimento do CodePlay. Para obter os resultados foi necessário aplicar o jogo e um questionário em um grupo de discentes do primeiro ano de graduação, afim de coletar os dados para justificar o trabalho.

Palavras-chave: pensamento computacional; jogos educativos; evasão acadêmica.

ABSTRACT

In recent years higher education institutions have faced serious problems caused by academic dropout rates. This work aims to stimulate computational thinking through an educational digital game, with the aim of motivating students to practice more about programming and proposing a tool capable of reducing dropout rates and academic retention. Therefore, a survey of the interface, mechanics and interaction features of similar games was carried out to serve as a basis for the development of CodePlay. To obtain the results, it was necessary to apply the game and a questionnaire to a group of first-year undergraduate students, in order to collect data to justify the work.

Keywords: computational thinking; educational games; academic dropout.

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1 – Os Quatros Pilares do Pensamento Computacional | 15 |
| Figura 2 – Jogo The Foos | 18 |
| Figura 3 – Jogo Progster | 20 |
| Figura 4 – Jogo WAlgor | 21 |
| Figura 5 – Metodologia da pesquisa | 23 |
| Figura 6 – Estrutura da aplicação | 26 |
| Figura 7 – Questão 2 | 29 |
| Figura 8 – Questão 3 | 29 |
| Figura 9 – Questão 4 | 30 |
| Figura 10 – Questão 5 | 30 |
| Figura 11 – Questão 6 | 31 |
| Figura 12 – Questão 7 | 31 |
| Figura 13 – Questão 8 | 32 |
| Figura 14 – Questão 9 | 32 |
| Figura 15 – Questão 11 | 33 |
| Figura 16 – Questão 12 | 34 |
| Figura 17 – Questão 13 | 35 |

LISTA DE QUADROS

| | |
|--|----|
| Quadro 1 – Comparativo entre os jogos semelhantes e o CodePlay | 22 |
|--|----|

LISTA DE ABREVIATURAS E SIGLAS

| | |
|-----|---------------------------------|
| IES | Instituições de Ensino Superior |
| PC | Pensamento Computacional |
| JDE | Jogos Digitais Educativos |
| IHC | Interação Humano-Computador |
| GDD | Game Design Document |

SUMÁRIO

| | | |
|----------------|--|----|
| 1 | INTRODUÇÃO | 12 |
| 1.1 | Objetivos | 13 |
| <i>1.1.1</i> | <i>Objetivo Geral</i> | 13 |
| <i>1.1.2</i> | <i>Objetivo(s) Específico(s)</i> | 13 |
| 1.2 | Organização do Trabalho | 13 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 14 |
| 2.1 | Pensamento Computacional | 14 |
| <i>2.1.1</i> | <i>Os Quatros Pilares do Pensamento Computacional</i> | 14 |
| <i>2.1.1.1</i> | <i>Decomposição</i> | 15 |
| <i>2.1.1.2</i> | <i>Abstração</i> | 15 |
| <i>2.1.1.3</i> | <i>Reconhecimento de Padrões</i> | 16 |
| <i>2.1.1.4</i> | <i>Algoritmos</i> | 16 |
| 2.2 | Jogos Digitais Educativos | 16 |
| 3 | TRABALHOS RELACIONADOS | 18 |
| 3.1 | O pensamento computacional através de jogos infantis: uma análise de elementos de interação | 18 |
| 3.2 | Progster - um jogo digital infantil para o ensino de pensamento computacional | 19 |
| 3.3 | WAlgor: um jogo de tower defense para o desenvolvimento do Pensamento Computacional e apresentação de algoritmos computacionais | 20 |
| 4 | PROCEDIMENTOS METODOLÓGICOS | 23 |
| 4.1 | Revisão Bibliográfica | 23 |
| 4.2 | Análise dos Trabalhos Relacionados | 23 |
| 4.3 | Criação do Game Design | 24 |
| 4.4 | Desenvolvimento do Jogo | 24 |
| <i>4.4.1</i> | <i>O jogo CodePlay</i> | 26 |
| 4.5 | Aplicação do Jogo em um Grupo de Discentes | 27 |
| 4.6 | Aplicação do Questionário Sobre o Jogo | 27 |
| 4.7 | Análise dos Resultados Obtidos no Questionário | 27 |
| 5 | RESULTADOS | 28 |

| | | |
|------------|--|-----------|
| 5.1 | Aplicação do Questionário Sobre o Jogo | 28 |
| 6 | CONCLUSÃO | 36 |
| | REFERÊNCIAS | 38 |
| | APÊNDICES | 40 |
| | APÊNDICE A–GAM DESIGN DOCUMENT | 40 |
| | APÊNDICE B–ALGORITMO PARA SOLUÇÃO DO PROBLEMA . . | 48 |
| | APÊNDICE C–QUESTIONÁRIO SOBRE O JOGO | 49 |

1 INTRODUÇÃO

Nos últimos anos, alunos recém-chegados aos cursos de computação trouxeram consigo deficiências em sua educação básica, referentes à interpretação textual e à solução de problemas. Estas habilidades são consideradas importantes para as disciplinas introdutórias de programação, normalmente ofertadas no primeiro ano de graduação (Lima et al., 2019). Tais deficiências são vistas como fatores determinísticos para a retenção acadêmica ou abandono da disciplina (Lima et al., 2019).

Segundo Moraes (2015), a retenção acadêmica é o termo empregado em função do processo que resulta no retardamento da conclusão de curso pelos alunos do ensino superior, induzindo-os a uma permanência prolongada nas Instituições de Ensino Superior (IES). Espíndola (2016) enfatiza que a sociedade sofre com o aparecimento da retenção acadêmica, visto que ela causa um atraso na disponibilidade de profissionais qualificados no mercado de trabalho.

Para Mourão (2017), as disciplinas introdutórias de programação, especificamente nos cursos de Computação, possuem grandes índices de reprovação e muitas das vezes acarretam na evasão do curso. Segundo Freire *et al.* (2019), tais disciplinas requerem conhecimentos sobre lógica, cálculos matemáticos e abstração. No entanto, para Lima *et al.* (2019), uma grande parcela dos alunos apresenta dificuldades em desenvolver estas habilidades, impossibilitando assim, o entendimento no decorrer destas disciplinas.

Segundo Hoed (2016), a evasão acadêmica de discentes nos cursos de graduação em Computação é uma preocupação antiga. Martins (2007), acredita que a evasão acadêmica é um termo que indica a saída de um discente de uma IES, de maneira definitiva ou temporária e por um motivo qualquer, exceto a diplomação. Segundo Moraes (2015), um elevado número de reprovações ou trancamentos podem acarretar em um desinteresse do aluno em permanecer no curso, levando-o a evasão acadêmica. Hoed (2016) também ressalta que, além de a evasão acadêmica ocasionar problemas à vida pessoal e profissional, também afeta as instituições, pois elas não atingem os índices esperados em relação à colação de grau de seus alunos.

Uma possibilidade plausível de mitigar essas dificuldades enfrentadas pelos discentes é estimular o aprendizado baseado na lógica e no uso do Pensamento Computacional (PC).

Para Stefanello *et al.* (2013), o PC não se restringe às ferramentas e recursos totalmente vinculados às tecnologias, mas também está voltado para um pensamento analítico diretamente relacionado com algumas características do pensamento matemático. Portanto, trata-se de uma forma específica de pensar e de analisar um problema, sendo influenciado por

todas as áreas do conhecimento, sendo assim, ele ressalta que o PC é crucialmente importante e que deve ser uma habilidade a ser desenvolvida nos alunos em disciplinas relacionadas a lógica e a programação.

Segundo Alves (2020), para estimular a aprendizagem do PC nos alunos, há um método bastante aplicado e trata-se dos jogos digitais educativos, que têm por objetivo transmitir conhecimento ao mesmo tempo que proporciona entretenimento.

Pensando nesta iniciativa de desenvolver o PC por meio dos jogos digitais, este trabalho tomou como foco o desenvolvimento de um jogo chamado CodePlay, que tem como finalidade motivar os discentes a praticar mais sobre programação e as suas aplicações. Portanto, no jogo será levado em consideração o pensamento algorítmico para abstrair e resolver problemas simples de programação através de uma metodologia lúdica.

Assim, com base no que foi apresentado anteriormente, na próxima Seção deste Capítulo são apresentados os objetivos deste trabalho.

1.1 Objetivos

1.1.1 Objetivo Geral

Desenvolver um jogo digital acadêmico que estimule o Pensamento Computacional na elaboração de soluções de problemas, inibindo os índices de evação e retenção acadêmica.

1.1.2 Objetivo(s) Específico(s)

1. Utilizar o ReactJS e o Java com Spring Boot no desenvolvimento do jogo;
2. Publicar o site no servidor externo;
3. Aplicar o jogo em um grupo de discentes do 1º ano de graduação;
4. Aplicar um questionário neste grupo referente ao jogo;
5. Justificar o trabalho perante os resultados obtidos do questionário.

1.2 Organização do Trabalho

O restante deste trabalho encontra-se estruturado da seguinte forma: o Capítulo 2 apresenta a fundamentação teórica; Capítulo 3, os trabalhos relacionados; no Capítulo 4, os procedimentos metodológicos; no Capítulo 5, os resultados e por fim, o Capítulo 6, a conclusão.

2 FUNDAMENTAÇÃO TEÓRICA

Apresenta-se neste Capítulo os conceitos fundamentais para o entendimento dos termos teóricos abordados neste trabalho.

Na Seção 2.1 será mostrado o conceito de Pensamento Computacional, abordando seus quatro pilares. Na Seção 2.2 será abordado o conceito de Jogos Digitais Educativos, onde iremos fazer uso neste trabalho.

2.1 Pensamento Computacional

Para Wing (2006), o PC é uma competência cujo foco é utilizar-se dos fundamentos da computação, baseado no raciocínio analítico [podendo fazer uma análise lógica e sintática] para obter uma solução de um problema complexo. No mesmo artigo, Wing (2006), acredita que o PC não é uma competência que deve ser desenvolvida apenas por entusiastas da computação, mas sim, uma habilidade que deveria ser também trabalhada no ensino fundamental de todos, em prol de uma sociedade mais qualificada em termos de intelectualidade.

Para SBC (2018), o PC está ganhando bastante respaldo nos últimos anos, pois vem sendo entendido como um dos princípios básicos do conhecimento, juntamente com a leitura e escrita; pois projeta, explica e apoia a compreensão do universo e de seus sistemas complexos. Ainda para SBC (2018), o PC trata-se de uma habilidade capaz de identificar, avaliar, entender problemas e implementar possíveis soluções de forma cautelosa e eficaz, através da elaboração de um conjunto de passos, que constitui um algoritmo.

Deste modo, podemos compreender o PC como uma forma eficaz de organizar ideias, reutilizando padrões e fundamentos da computação para identificar, analisar e resolver problemas. Portanto, o PC contribui na elaboração dos passos que constituem o algoritmo, de modo que o ser humano ou uma máquina consiga entendê-lo e executá-lo de forma inteligente. Dessa forma, o PC é baseado em quatro pilares, conforme apresentado na Seção a seguir (BBC, 2019).

2.1.1 *Os Quatros Pilares do Pensamento Computacional*

Nesta Seção, será mostrado os quatros pilares que correspondem à composição fundamental do PC. Seu papel é servir de molde para reconhecer e solucionar problemas complexos. Os tais pilares são: decomposição, reconhecimento de padrões, abstração e algoritmo.

Figura 1 – Os Quatros Pilares do Pensamento Computacional



Fonte: Shoop e Flot (2016)

2.1.1.1 *Decomposição*

Ao se deparar com um problema grande e complexo, tentar solucionar ele por completo, pode ser um trabalho difícil e às vezes cansativo. Desse modo, entendemos que um problema complexo certamente é composto por diversos subproblemas, complicando assim, sua compreensão. Neste sentido, uma saída inteligente de resolver isto é fazendo o uso do mecanismo de decomposição, que se resume em resolver problemas complexos, fragmentando-os em partes menores, tornando-os assim, problemas mais simples de serem resolvidos (BBC, 2019).

2.1.1.2 *Abstração*

Segundo Liukas (2015), este é o pilar responsável pela filtragem dos dados relevantes, desconsiderando todas as informações irrelevantes em função de manter seu foco apenas nos detalhes que realmente importa para formular uma resolução coerente de um determinado problema.

2.1.1.3 Reconhecimento de Padrões

Em um processo extenso de resolução de um problema, é possível detectar algumas particularidades homogêneas entre os subproblemas, que podem ser reutilizadas em outras resoluções. Conforme BBC (2019), o reconhecimento de padrões é o pilar que consiste em uma forma de lidar com problemas de modo ágil, aplicando o reuso de soluções em problemas similares.

Dito isso, suponha que um determinado indivíduo tenha que solucionar um grande problema, sabendo que o mesmo já aplicou o método de Decomposição, ele pode agora analisar os problemas menores de modo a encontrar padrões entre eles e que ao resolvê-los, seja possível reutilizar a mesma solução para os problemas com padrões semelhantes.

2.1.1.4 Algoritmos

O algoritmo é um conjunto de instruções cujo a finalidade é resolver um problema. Supondo que haja um problema já modelado, de acordo com o uso dos três métodos já citados anteriormente, o momento agora é de idealizar o algoritmo que resolva tal problema. Para BBC (2019), um algoritmo é um mesclado de passo a passo que irá instruir a resolução de um certo problema.

Desse modo, as instruções devem ser coerentes e concisas para que o problema seja resolvido da melhor maneira. Portanto, tais instruções podem ser descritas em formato de pseudocódigo (língua facilmente compreendida pelo ser humano), a fim de que posteriormente possa ser convertida em código, por meio de uma linguagem de programação.

O Codeplay por sua vez, busca promover situações interessantes e desafiadoras para a resolução de problemas, com o intuito de estimular o uso do pensamento computacional.

2.2 Jogos Digitais Educativos

Os jogos digitais são desenvolvidos com o objetivo de entreter seus jogadores e intensificar o desenvolvimento de novas habilidades analíticas e estratégicas. Segundo Gros (2003), com o aparecimento de novas maneiras de propagar e abstrair conhecimento, impulsionadas pelo surgimento de novas tecnologias, os jogos digitais passaram a ser vistos também como uma ferramenta de cunho educacional, abandonando aquela ideia de serem utilizados somente para diversão.

Para Decian (2010), os Jogos Digitais Educativos (JDE), em específico, são jogos que possuem uma proposta pedagógica implantada em seu escopo e conteúdo, com o intuito de promover o aprendizado e o desenvolvimento de novas habilidades, contribuindo assim, para que o processo de ensino-aprendizagem aconteça de forma mais natural e dinâmica.

Souza e Dias (2012) afirmam que a busca por JDE adequados para auxiliar no desenvolvimento do PC tem sido largamente explorada. Neste contexto, compreende-se também por PC:

uma abordagem para a resolução de problemas de forma que pode ser implementada com um computador, usando um conjunto de conceitos, tais como abstração, recursão e iteração, processamento, análise de dados e criação de artefatos reais e virtuais (WANGENHEIM *et al.*, 2014).

A seguir, Moratori (2003) cita alguns benefícios que os JDE pode proporcionar ao jogador, mesmo que de forma indireta:

1. Memória (visual e auditiva);
2. Orientação temporal e espacial;
3. Coordenação motora visomanual (ampla e fina);
4. Raciocínio lógico-matemático;
5. Expressão lingüística (oral e escrita);
6. Planejamento e organização.

Desta forma, os JDE devem propagar algum tipo de conhecimento, mas por outro lado, também não devem ser vistos apenas como uma ferramenta pedagógica, de modo que percam seu entretenimento, por isso, é necessário que haja uma harmonia entre o papel pedagógico e o papel divertido do jogo.

Este conceito é bastante relevante para o presente trabalho, pois o CodePlay é um jogo digital educativo e portanto, foi necessário este estudo sobre as características básicas para o seu desenvolvimento.

3 TRABALHOS RELACIONADOS

Neste Capítulo, serão apresentados os trabalhos relacionados, bem como seus aspectos fundamentais, semelhanças e diferenças com o presente trabalho.

O PC juntamente com os JDE vem sendo um tema bastante relevante e abordado em outros trabalhos. Por exemplo, Falcão *et al.* (2015), Alves (2020) e o Michel *et al.* (2019), apresentam características de pesquisa similares ao que é abordado no presente trabalho.

3.1 O pensamento computacional através de jogos infantis: uma análise de elementos de interação

No trabalho de Falcão *et al.* (2015), foi realizada uma análise a respeito da interação do jogo digital "The Foos", que tem como finalidade propagar o ensino a respeito das concepções fundamentais de lógica de programação. Com tudo, foi feita uma avaliação do entendimento e do uso dos elementos de interface, bem como, eles interagem com a dinâmica do jogo e também na formação dos conceitos associados ao PC.

Figura 2 – Jogo The Foos



Fonte: CodeSpark Academy

O *The Foos* é um JDE desenvolvido com foco em contribuir de forma didática para com as crianças acima de 5 anos de idade e estimular o raciocínio lógico, em especial o pensamento computacional. No jogo, o usuário conduz o personagem (de cor verde conforme mostrado na Figura 2) a realizar os desafios, por meio de peças que correspondem as seguintes

ações: andar, coletar, pular e também instruções mais elaboradas, como condição e repetição.

Como pode ser observado na Figura 2, o personagem precisa se locomover até a rosca (localizada no topo direito), mas durante o percurso ele tem a missão de coletar as moedas verdes. Dito isso, é fundamental que o jogador arraste para a área de execução, as peças referentes às ações necessárias para realizar esta missão, como por exemplo: andar, pular e coletar. A cada fase do jogo, novos mecanismos são inseridos e para que o usuário consiga avançar nestas fases, ele precisa arrastar os blocos correspondentes para a área de execução e depois clicar no personagem.

De acordo com o Quadro 1, o presente estudo se assemelha ao trabalho de Falcão *et al.* (2015) porque ambos os jogos têm como objetivo específico a solução de problemas. A principal diferença entre os trabalhos citados anteriormente, é que no jogo avaliado por Falcão *et al.* (2015), seu público alvo foi crianças de 5 a 7 anos, enquanto que neste trabalho, o principal público alvo é os alunos de primeiro ano de graduação.

3.2 Progster - um jogo digital infantil para o ensino de pensamento computacional

O trabalho de Alves (2020), tem como objetivo o desenvolvimento de um jogo digital baseado em um tabuleiro eletrônico, levando em consideração os aspectos visuais e conceituais originais. O tabuleiro que serviu de inspiração para o desenvolvimento deste jogo digital foi proposto por Lemos *et al.* (2017), que inclusive, ganhou 1º lugar em Simpósio de Interação Humano-Computador (IHC), neste momento, o Progster havia sido contemplado e apresentado apenas como um protótipo inaugural.

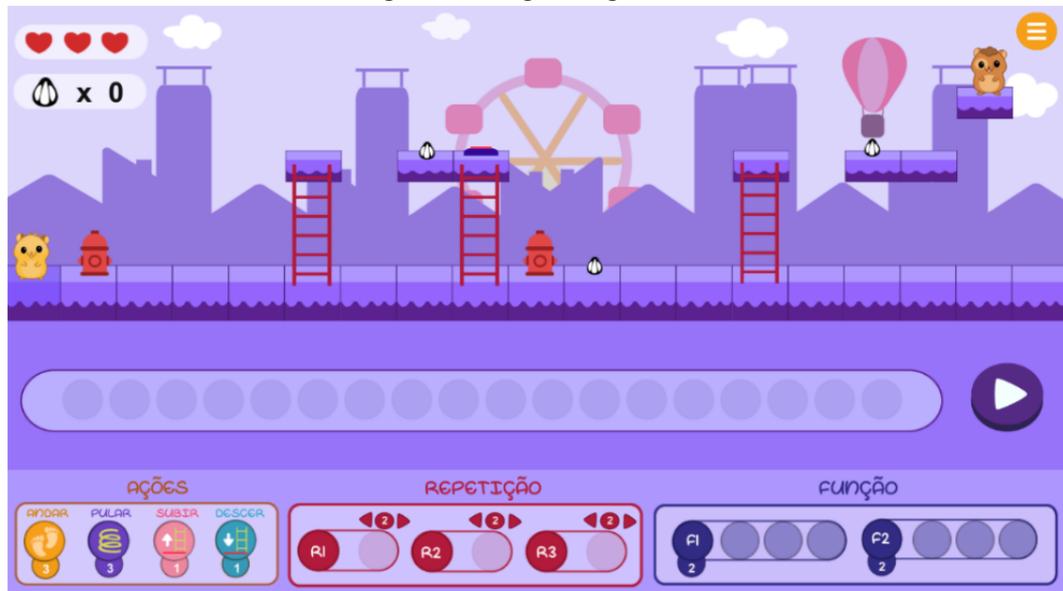
A proposta de Alves (2020) foi atualizar a mecânica do jogo, interface e interação em relação ao jogo original (tabuleiro eletrônico). No entanto, o intuito principal do brinquedo Progster foi sustentado, com o ensino de PC baseado na abordagem dos quatros pilares, já descritos anteriormente na Seção 2.1.1.

O Progster é um jogo digital que foi desenvolvido com o objetivo de ensinar o PC, utilizando os fundamentos da computação para estimular o desenvolvimento do pensamento analítico, formulando passos de instruções, onde as peças são encaixadas de forma sequencial e às vezes limitadas para que as crianças consigam utilizar tais conceitos e assim, possam chegar à etapa final do jogo.

Na Figura 3, apresenta a última fase do jogo, no mundo "cidade", nela o personagem precisa chegar até o seu amigo (lado superior direito), mas terá alguns desafios, por exemplo,

subir as estradas. Dito isso, é possível observar que a quantidade de peças é insuficiente para chegar até o Luke (nome do seu amigo), posto que tem apenas três peças que condiz com as ações de andar e pular e somente uma que condiz com as ações de subir e descer.

Figura 3 – Jogo Progster



Fonte: Alves (2020)

A princípio, uma possível saída para este caso, é fazer o uso da área de repetição, mas, nota-se que a peça que condiz com a ação de subir não pode se repetir, fazendo com que o jogador pense em outra solução. Desta forma, entende-se que o jogador deve aplicar o método citado em 2.1.1.3, para o reconhecimento de um padrão de ações, de modo que reutilize algumas peças e que faça o uso de função. Como pode ser observado na Figura 3, a área de função permite reutilizar três peças até duas vezes, portanto, essa reutilização menciona o conceito de função.

As principais semelhanças entre o presente trabalho e o trabalho de Alves (2020), é que os dois buscam ensinar e/ou motivar o uso dos conceitos do PC integrado aos JDE. A principal diferença entre esses dois trabalhos se dá em função de que o jogo que será desenvolvido no presente trabalho não tem uma dinâmica baseada em fases, já o jogo desenvolvido no trabalho de Alves (2020) conta com três seções chamadas "mundos", cada uma delas com três fases.

3.3 WAlgor: um jogo de tower defense para o desenvolvimento do Pensamento Computacional e apresentação de algoritmos computacionais

No trabalho de Michel *et al.* (2019), a proposta é o desenvolvimento de um JDE chamado WAlgor, baseado no Pensamento Computacional e nos seus pilares, já citados anteri-

ormente, na Seção 2.1.1. A finalidade do jogo é abordar algoritmos computacionais de forma implícita para a resolução de problemas. O WAlgor também se trata de uma implementação categorizada como *serious game* (foco para fins educacionais e menos entretenimento).

O WAlgor é um JDE desenvolvido para dispositivos móveis, que contém uma estratégia com base no gênero *tower defense* (jogos em que o objetivo é defender uma base de ataques inimigos) e ideal para quem gosta de ação. O nome do jogo teve origem de uma abreviação de *Warriors of Algorithms*, que neste contexto, pode ser entendido como Guerreiros Algorítmicos. A mecânica do WAlgor é semelhante a jogos já renomados, como *Plants vs Zombies* e *Trolls vs Vikings*, onde o propósito é defender um reino utilizando estratégias lógicas e no caso do WAlgor, isso se dar por meio dos cavaleiros.

Para combater os ataques, o jogador precisa utilizar-se do PC para formular algoritmos que posicionem os cavaleiros em batalha, de modo que defenda o reino, como pode ser observado na figura 4.

Figura 4 – Jogo WAlgor



Fonte: Michel *et al.* (2019)

No Quadro 1, é possível observar que o presente estudo se assemelha ao trabalho de Michel *et al.* (2019) porque ambos os jogos têm mecânicas de deslocamento do personagem. A principal diferença entre os trabalhos citados anteriormente, é que no trabalho de Michel *et al.* (2019) utilizou-se tecnologias para o desenvolvimento de uma plataforma para dispositivos móveis, enquanto que no presente estudo, foram utilizadas tecnologias para o desenvolvimento de uma aplicação *Web*.

Apresenta-se no Quadro 1, uma comparação do jogo que o presente trabalho se propôs a desenvolver com os demais jogos dos trabalhos similares. Para isso, foi definido algumas métricas para realizar tal comparação, seguem:

- **Abordagem:** analisa-se qual o contexto do PC sob o qual o trabalho foi desenvolvido;
- **Módulos:** verifica-se os conteúdos que buscam ser ensinados no jogo;
- **Mecânica:** analisa-se como se dar a interação com o jogo, se é feita mediante ao arraste

- das peças ou deslocamento do personagem;
- **Interface:** verifica-se os elementos visuais que integram o jogo;
 - **Público-alvo:** analisa-se qual o público-alvo do jogo.

Quadro 1 – Comparativo entre os jogos semelhantes e o CodePlay

| Jogos/Métricas | The Fools | Progster | WAlgor | Codeplay |
|----------------|---|--|--|--|
| Abordagem | Ensino dos conceitos básicos de lógica de programação | Ensino dos conceitos da ciência da computação para desenvolver o raciocínio lógico | Formular algoritmos computacionais para a resolução de problemas | Desenvolver o pensamento computacional para solução de problemas |
| Módulos | Lógica, sequenciamento, reconhecimento de padrões, e solução de problemas | Condição, repetição e função | Condição, repetição, ordenação e busca | Condição, repetição, listas e operadores matemáticos |
| Mecânica | Arraste de Peças | Arraste de Peças | Deslocamento do personagem | Deslocamento do personagem |
| Interface | Personagem, moeda, estrela | Bloco, semente e bandeirinha | Inimigos, cavaleiros e reino | Personagem, casas, árvores e inimigos |
| Público-alvo | Crianças de 5 a 7 anos de idade | Crianças de 7 a 9 anos | Usuários a partir de 10 anos | Alunos de primeiro ano de graduação |

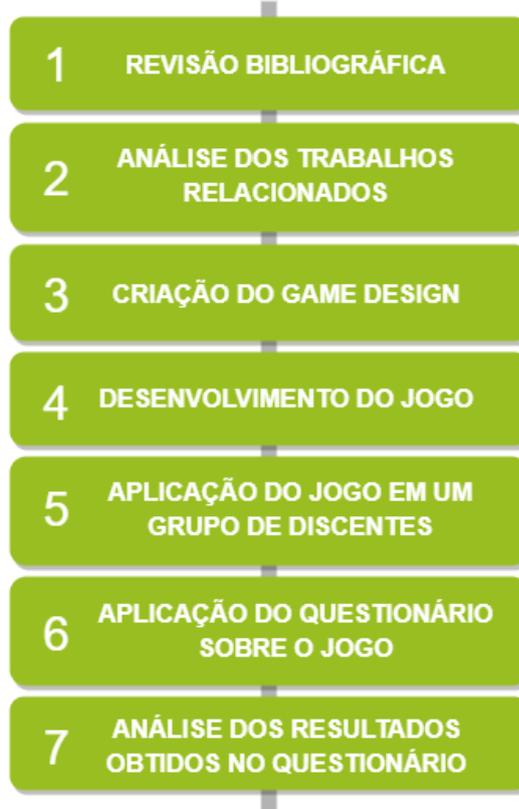
Fonte: elaborado pelo autor (2023).

A partir dos trabalhos apresentados, é possível perceber as principais abordagens que vêm sendo utilizadas em pesquisas envolvendo PC em jogos digitais. Assim, tais abordagens empregadas nestes trabalhos foram utilizadas como base de pesquisa para o presente trabalho.

4 PROCEDIMENTOS METODOLÓGICOS

Para atingir o objetivo proposto, foi realizado a execução dos seguintes passos, descritos na Figura 5.

Figura 5 – Metodologia da pesquisa



Fonte: elaborado pelo autor (2023).

4.1 Revisão Bibliográfica

Nesta etapa inicial, foi realizado um levantamento bibliográfico do estado da arte sobre o assunto abordado neste trabalho, buscou-se compreender sobre o uso dos jogos digitais educativos como um método para o desenvolvimento do PC.

4.2 Análise dos Trabalhos Relacionados

Segundo Alves (2020), no ato de desenvolver um jogo se faz necessário coletar e avaliar as características dos jogos semelhantes, para que possa ser observado os aspectos de interface e interação, bem como, os pontos que devem ou não constar no jogo que será desenvolvido.

Desse modo, foi realizado um estudo referente aos seguintes trabalhos: o de Falcão *et al.* (2015), intitulado como "O pensamento computacional através de jogos infantis: uma análise de elementos de interação", o de Alves (2020), intitulado de "Progster - Um jogo digital infantil para o ensino de pensamento Computacional" e por último, mas não menos importante, o trabalho de Michel *et al.* (2019), intitulado de "WAlgor: um jogo de tower defense para o desenvolvimento do Pensamento Computacional e apresentação de algoritmos computacionais".

Portanto, o intuito desse estudo foi coletar as principais semelhanças e aspectos desses jogos em relação à interface e interação, para que pudesse servir de base para a criação do *Game Design* e para o desenvolvimento do CodePlay.

4.3 Criação do Game Design

O *Game Design* é um documento que planeja e descreve as características de um determinado jogo. Segundo Machado *et al.* (2009), durante o desenvolvimento de um jogo, é recomendado que nenhuma implementação seja feita, caso o *Game Design Document (GDD)* não esteja pronto. No entanto, o GDD pode ser adaptado ao jogo e mesmo estando "pronto", pode também sofrer algumas alterações.

Portanto, nesta etapa, foi realizada a definição da história, mecânicas, interface, personagem e dinâmica do jogo CodePlay, levando em consideração o que foi levantado na etapa anterior (Seção 4.2). Tais definições estão contidas no GDD, localizado no Apêndice A.

4.4 Desenvolvimento do Jogo

Nesta etapa, foi desenvolvido uma versão beta do CodePlay, uma versão para testes que ainda há possibilidades de se encontrar alguns *bugs*, mas que já pode ser apresentada ao público alvo. Com base no GDD que foi formulado na etapa anterior, foi utilizado como principal tecnologia para a implementação, o *ReactJS*, que segundo Batista (2020) é compreendido como:

uma biblioteca de código aberto do *JavaScript*, desenvolvida pelo *Facebook*, focada em criar interfaces de usuário para aplicações *web*; atualmente é mantida por milhares de desenvolvedores de forma voluntária. O *ReactJS* também pode ser classificado como a camada *View* no padrão *Model-View-Controller (MVC)*.

O jogo foi desenvolvido na versão *Web* (para rodar em navegadores), essencialmente em computadores. Isto porque, observou-se a necessidade de haver uma boa visualização dos elementos de interface, para que o jogo seja o mais claro e intuitivo possível. No desenvolvimento

do *front-end*, principal linguagem de programação que foi utilizada é o *JavaScript*, por ser a linguagem mais comumente utilizada no desenvolvimento de projetos implementados em *ReactJS*.

Para complementar o desenvolvimento do CodePlay também foi utilizado o *Java*, juntamente com o *Spring Boot*, no que diz respeito à implementação do *back-end*. Para Arnold *et al.* (2005), o *Java* é a linguagem de programação orientada a objetos, cujo os programas são implementados por meio de classes. Desse modo, a partir de uma definição de classe, pode ser criado vários objetos que são chamados como instâncias, dessa classe. Já o *Spring Boot*, segundo Wang *et al.* (2020), é um *framework* que conta com uma estrutura de configuração que auxilia no processo de desenvolvimento e publicação de aplicações, melhorando assim, a performance de desenvolvimento.

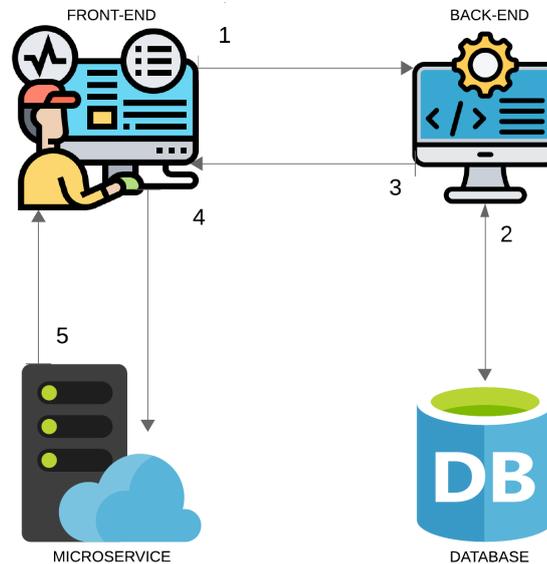
O *Swagger* foi outro recurso utilizado no desenvolvimento do CodePlay. A ferramenta serviu para documentar a aplicação Backend do jogo, auxiliando o desenvolvedor no gerenciamento e padronização da *API REST*, visando um melhor desempenho.

O jogo ainda conta com um *microservice* que interpreta e faz o devido julgamento do algoritmo elaborado pelo jogador. Este *microservice* é um reuso de uma aplicação implementada em *Node.js*, uma espécie de ambiente de execução de códigos *JavaScript* orientado a eventos. Segundo Barsoti e Gibertoni (2020), isso o torna muito eficiente, pois intensifica o tráfego de dados para aplicações em *real-time* (tempo real), sendo um ponto muito significativo no âmbito das aplicações *Web*.

O Banco de Dados que foi utilizado para o armazenamento de dados, é o *MongoDB*, que segundo Marchioni (2015), trata-se de um *Database* (banco de dados), orientado a documentos, de código aberto e *NoSQL*, tipo de armazenamento que não segue o modelo relacional. Por fim, foi usado o *GitHub* para o armazenamento e controle de versões de código.

Pode-se observar abaixo na Figura 6, uma ilustração estrutural da aplicação do jogo.

Figura 6 – Estrutura da aplicação



Fonte: elaborado pelo autor (2023).

Na primeira imagem da ilustração mostra o jogador interagindo com o jogo (*front-end*). No passo 1, indica que em algum momento o jogador fez requisições ao *back-end*, que por sua vez, está conectado ao banco de dados (passo 2), onde estar armazenado todos os desafios. Dessa forma, o *back-end* retorna a resposta da requisição, alimentando a aplicação (passo 3).

Desse modo, o jogador terá acesso aos desafios, podendo submeter seus algoritmos para resolvê-los (passo 4). E por fim, o *microservice* executa o julgamento do algoritmo elaborado pelo jogador e transmite para o *front-end*, que se encarrega de exibir para o jogador (passo 5).

4.4.1 O jogo CodePlay

O CodePlay tem como embasamento o projeto proposto pelo PAIP (Programa de Acolhimento e Incentivo a Permanência), intitulado "O uso de metodologias ativas como estratégia para desenvolver o pensamento computacional nas disciplinas introdutórias de programação", cujo o autor do presente trabalho exerceu a função de voluntário do projeto no período de 03/05/2021 a 14/02/2022. O objetivo deste projeto foi desenvolver um protótipo de um jogo, denominado como Código Ventura, com a finalidade de identificar as dificuldades dos alunos nas disciplinas introdutórias de programação.

Diferentemente do protótipo desenvolvido no projeto citado, o CodePlay é composto

por duas partes, na primeira parte, tem um teste de conhecimento. Por meio de um questionário que contém 10 perguntas; se o jogador acertar 7 ou mais respostas, ele passa para a próxima parte. Caso contrário, entende-se que ele ainda não está preparado.

A segunda parte do jogo consiste em um mapa onde o jogador poderá percorrê-lo, com o intuito de evitar os inimigos e resolver os problemas de programação, abordando os conteúdos que podem ser visualizados no Quadro 1. Portanto, nesta segunda parte, o jogador tem como objetivo resolver pelo menos 3 desafios para finalizar o jogo.

A seguir é possível observar um desafio que pode ser encontrada no CodePlay:

– "Eduardo precisa identificar quantas letras uma palavra têm. Para isso, desenvolva um programa onde ele possa inserir uma palavra e receber a quantidade de letras que ela tem". No Apêndice B, pode ser observado o algoritmo que resolve este problema.

4.5 Aplicação do Jogo em um Grupo de Discentes

Nesta etapa, foi realizado a aplicação do jogo em um grupo de discentes, totalizando 5. Este experimento foi possível graças à contribuição de Victor Cavalcante, aluno do 4º semestre do curso de Sistemas de Informação. O mesmo convidou os discentes do 1º ano de graduação a participarem do teste, jogando o CodePlay.

Foi criado um grupo no aplicativo WhatsApp com os participantes, para transmitir todas as informações necessárias, incluindo uma breve introdução sobre o PC, como ele está relacionado com o trabalho, o link do jogo e um suporte, caso surgisse alguma dúvida.

4.6 Aplicação do Questionário Sobre o Jogo

Nesta etapa, foi formulado um questionário a respeito do jogo, que posteriormente foi aplicado no grupo de participantes. Foi avaliado os seguintes aspectos: interface, conteúdos abordados, atenção, relevância, e satisfação dos jogadores. Além disso, outro ponto relevante que foi levantado nesta etapa, foi a aceitação do jogo.

No Apêndice C, pode-se observar as perguntas selecionadas para esta avaliação.

4.7 Análise dos Resultados Obtidos no Questionário

Etapa de coleta e análise das respostas dos participantes para formular os devidos resultados.

5 RESULTADOS

Neste Capítulo, serão apresentados os resultados obtidos correspondentes à Seção 4.6.

Na Seção 5.1, pode-se observar toda a avaliação realizada sobre os resultados coletados no questionário. O público alvo que participou do questionário foram os alunos do 1º ano de graduação, em especial, os alunos que apresentaram ter uma certa dificuldade em desenvolver a lógica de programação.

5.1 Aplicação do Questionário Sobre o Jogo

Aplicação do Questionário Sobre o Jogo se deu por meio de um formulário aplicado no grupo de discentes, que pode ser melhor visualizado no apêndice C. O mesmo foi composto por perguntas que abrange os seguintes aspectos: interface, conteúdos abordados, atenção, relevância e satisfação dos jogadores.

Na questão 1, foi feita a seguinte pergunta: "Algo interessante chamou sua atenção durante o jogo? Se sim, conte o que foi". Portanto, foram coletadas as seguintes respostas:

- Participante Nº 1: "Gostei bastante da interface, além da música. Também apreciei a dinâmica das questões, algo bem dinâmico e bem estruturado".
- Participante Nº 2: "Sim, as artes estão muito boas e o nível das questões estão de acordo com o público alvo".
- Participante Nº 3: "A parte de ter que passar por um teste primeiro para poder jogar o jogo. Assim você vai ter que provar que tem algum entendimento na linguagem utilizada no mesmo".
- Participante Nº 4: "O gráfico do mapa é muito bonito, a lógica de programação utilizando python também é uma ótima escolha".
- Participante Nº 5: "A jogabilidade em si, a interface gráfica e os desafios".

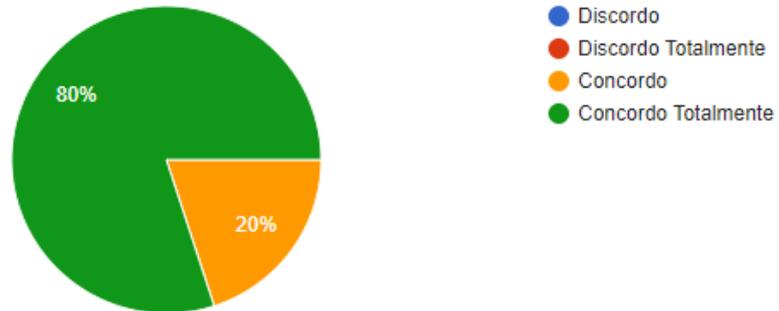
Nesse contexto, pode-se considerar que as respostas no critério de atenção foram muito positivas, indicando que o jogo conseguiu capturar a atenção dos jogadores.

Na Figura 7, é possível observar a avaliação feita com base no quesito de satisfação, toda boa experiência satisfaz muito o ego pessoal e potencialmente, o aprendizado. Portanto, nota-se que todos tiveram uma experiência muito boa, isso implica que, os resultados obtidos foram bastante positivos.

Figura 7 – Questão 2

Minha experiência foi muito boa ao jogar o CodePlay:

5 respostas



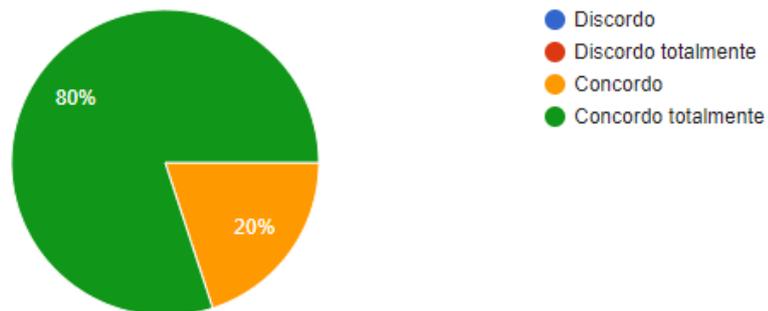
Fonte: elaborado pelo autor (2023).

Na Figura 8, nota-se a avaliação realizada com base no quesito de interface, em suma, os jogos que têm um design de interface atraente costumam prender mais a atenção de seus jogadores. Nesse sentido, pode-se considerar que as respostas neste critério foram muito positivas, indicando mais uma vez, que o jogo conseguiu capturar a atenção dos jogadores através de sua boa interface.

Figura 8 – Questão 3

Achei o design atraente:

5 respostas



Fonte: elaborado pelo autor (2023).

O propósito da pergunta apresentada na Figura 9 foi avaliar o quanto o jogo é desafiador. Os resultados obtidos revelaram que 40% concordaram e 60% concordaram totalmente que o jogo é desafiador. Portanto, considera-se que o CodePlay é um jogo desafiador e que as respostas foram muito satisfatórias neste quesito.

Figura 9 – Questão 4

Considero o jogo desafiador:

5 respostas



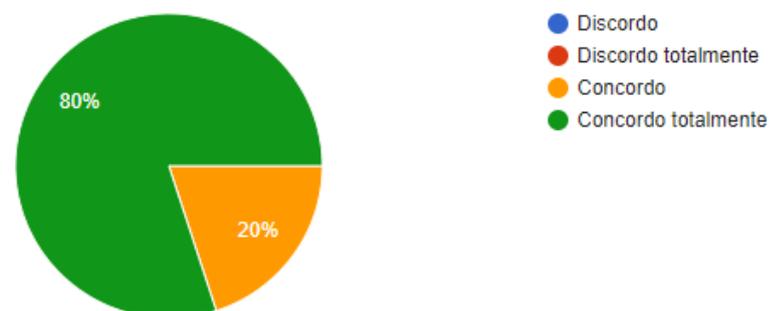
Fonte: elaborado pelo autor (2023).

Na Figura 10, é possível observar a avaliação realizada com base no quesito de relevância, visto que, todos os participantes consideraram que o jogo pode ser utilizado como uma ferramenta de ensino para auxiliar o processo de aprendizagem, nas disciplinas introdutórias de programação. Nesse contexto, percebe-se que as respostas foram muito positivas neste critério, apontando sua relevância.

Figura 10 – Questão 5

Considero o jogo uma ferramenta de auxílio para o ensino, em Disciplinas Introdutórias de Programação:

5 respostas



Fonte: elaborado pelo autor (2023).

Na Figura 11, é possível observar a avaliação realizada com base também no quesito de relevância, posto que, a pergunta faz um levantamento sobre o incentivo do pensamento computacional por meio do CodePlay. Desta forma, nota-se que 100% dos participantes concordaram totalmente que o jogo estimula o pensamento computacional, isso implica que, os resultados foram ótimos e que o objetivo principal foi alcançado.

Figura 11 – Questão 6

Concordo que o jogo estimula de alguma forma o pensamento computacional.
(Entende-se como pensamento computacional, uma forma inteligente de resolver um problema, estimulando o pensamento crítico e analítico):

5 respostas



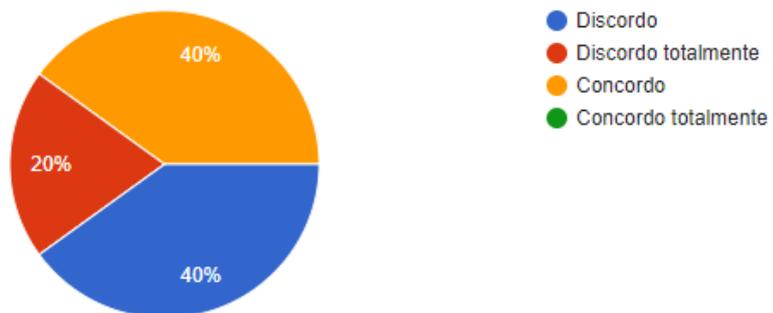
Fonte: elaborado pelo autor (2023).

O propósito da pergunta apresentada na Figura 12 foi avaliar as expectativas dos participantes em relação ao quanto o jogo foi difícil. Os resultados coletados indicam que 20% discordaram totalmente, 40% discordaram e 40% concordaram que o jogo foi difícil mais do que imaginavam. Tendo em vista que, o jogo não tem a finalidade de ser difícil, considera-se que as respostas foram satisfatórias.

Figura 12 – Questão 7

O jogo foi mais difícil do que eu imaginava:

5 respostas



Fonte: elaborado pelo autor (2023).

Na Figura 13, nota-se a avaliação realizada com base no quesito de satisfação, pois ao finalizar o jogo é natural que provoque um sentimento de realização. Nesse cenário, percebe-se que 100% dos participantes concordaram totalmente, isso mostra que as respostas foram positivas e que o jogo foi capaz de transmitir emoção aos jogadores.

Figura 13 – Questão 8

Atingir o objetivo do jogo me deu um sentimento de realização:

5 respostas



Fonte: elaborado pelo autor (2023).

Na Figura 14, pode-se observar a análise realizada com base nos conteúdos abordados, pois ao finalizar o jogo é esperado que se tenha compreendido melhor os conteúdos que foram abordados. Nesse sentido, percebe-se que 60% dos participantes concordaram totalmente e 40% também concordaram, isso indica que os resultados foram muito positivos e que o jogo conseguiu exercer uma boa função.

Figura 14 – Questão 9

Após o jogo sinto que consigo compreender melhor os conteúdos que foram abordados:

5 respostas



Fonte: elaborado pelo autor (2023).

A análise realizada sobre a questão 10, teve como fundamento os conteúdos abordados. A pergunta dirigida para os participantes foi: "Os conteúdos abordados no jogo contribuíram de alguma forma para a sua aprendizagem na programação? Se sim, como?". Portanto, foram obtidas as seguintes respostas:

- Participante Nº 1: "Sim, ao resolver as questões, consegui aprender a escrever o código em si. Isso me permitiu resolver não apenas essas questões, mas também outras que seguem a mesma lógica".
- Participante Nº 2: "O jogo ele força o usuário a buscar e entender melhor o python e a lógica de programação".
- Participante Nº 3: "Sim. Pois como nunca tive um contato tão direto com a linguagem utilizada no jogo, pude aprender como ela funciona e por se tratar de uma linguagem envolvendo mais a parte da lógica o jogo ajuda bastante o desenvolvimento nesse conceito".
- Participante Nº 4: "Sim, pois coloquei em prática a lógica que tenho em python, mesmo não sabendo o suficiente, foi extremamente gratificante participar".
- Participante Nº 5: "Sim, o uso de loops para percorrer vetores, por exemplo".

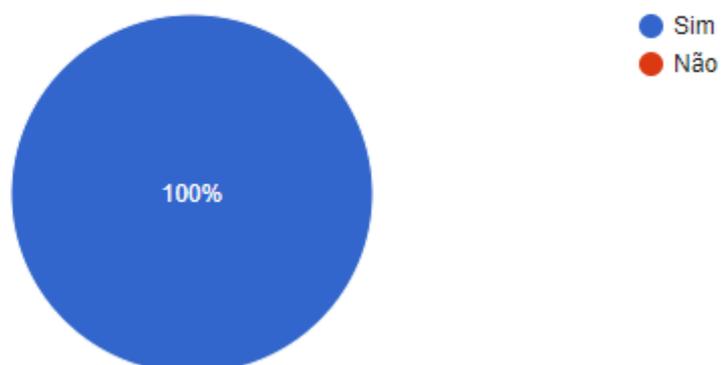
Desse modo, pode-se considerar que as respostas foram bastante satisfatórias, indicando que o jogo aborda conteúdos significativos, contribuindo assim, para a aprendizagem voltada à programação.

Na Figura 15, pode-se observar a avaliação realizada com base no critério de satisfação, dado que, caso o jogador tenha se sentido bem e satisfeito com sua experiência, o mesmo jogará novamente o jogo, facilmente. Nesse contexto, nota-se que todos os participantes concordaram totalmente com o questionamento, portanto, tais respostas foram bastante favoráveis para o presente trabalho.

Figura 15 – Questão 11

Apesar de ser um jogo educacional, você jogaria novamente?

5 respostas



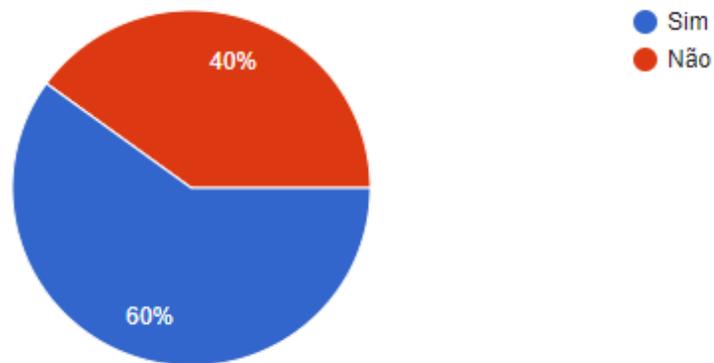
Fonte: elaborado pelo autor (2023).

Já na Figura 16, a análise realizada foi considerada com base no critério de interface, pois um tutorial bem elaborado contribui de forma suficiente para que o jogador não tenha nenhum impasse durante o jogo, além de colaborar com uma interface mais completa. Portanto, os resultados apontam que 60% dos participantes concordaram que o tutorial foi suficiente, enquanto 40% relataram o contrário, isso indica que, os resultados foram bons, entretanto, o tutorial poderia ter sido melhor implementado em função de se obter uma jogabilidade mais adequada.

Figura 16 – Questão 12

O tutorial foi suficiente para você conseguir jogar?

5 respostas



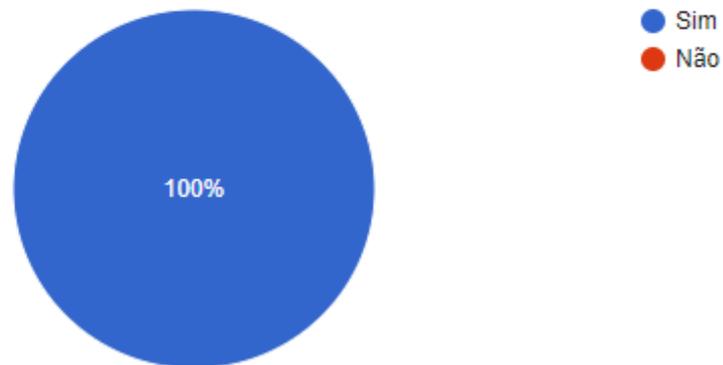
Fonte: elaborado pelo autor (2023).

A análise realizada na questão 13, apresentada na Figura 17 teve como propósito avaliar a relevância do jogo ao ponto do participante recomendá-lo para um colega. Nesse contexto, nota-se que todos concordaram que recomendariam o jogo para um colega de turma, isso mostra que, as respostas obtidas foram muito positivas, indicando que o jogo ainda pode contribuir bastante.

Figura 17 – Questão 13

Você recomendaria o jogo para um colega de turma?

5 respostas



Fonte: elaborado pelo autor (2023).

Para fechar essa Seção, na questão 14 pode-se observar a última avaliação realizada. Esta análise não teve um embasamento em um critério específico, pois a proposta foi coletar sugestões para um possível aprimoramento do jogo. A pergunta dirigida para os participantes foi: "Qual sua sugestão de melhoria?". Portanto, foram obtidas as seguintes respostas:

- Participante N° 1: "A implementação de novas linguagens de programação, como C, Java e C++, para a resolução de questões, e o aumento do banco de questões".
- Participante N° 2: "1 - O botão de tutorial está muito escondido, penso eu que se o tutorial for obrigatório o botão poderia continuar pequeno. 2 - Uma possível seleção de nível de dificuldade (fácil, médio e difícil) para aqueles alunos que querem um possível desafio".
- Participante N° 3: "O jogo poderia ter mais inimigos adicionados no mapa e em vez do personagem ter que cumprir três desafios, ele vai ter o objetivo de chegar ao fim do mapa respondendo as perguntas de lógica dos inimigos e fazendo os programas sugeridos pelos moradores das casas durante o caminho".
- Participante N° 4: "Nenhuma sugestão".
- Participante N° 5: "Melhorar o tutorial".

6 CONCLUSÃO

Com o objetivo de desenvolver o pensamento computacional e com isso motivar os discentes a praticar mais sobre programação na elaboração de soluções de problemas, o presente trabalho desenvolvido demonstrou ser uma ferramenta adequada e com um potencial satisfatório para auxiliar no ensino em Disciplinas Introdutórias de Programação, visando também diminuir os índices de evasão e retenção acadêmica.

O desenvolvimento do CodePlay não foi uma tarefa simples de ser executada, em função de ser um projeto que engloba vários recursos. Foi crucial a utilização do pensamento computacional não só para a resolução dos desafios do jogo, mas também para sua implementação, principalmente quando se refere a decomposição. Por exemplo, no Front-end a tela principal é composta por três componentes, e cada componente é composto por funções, variáveis, estruturas JSON e métodos condicionais. Cada recurso tem uma finalidade específica em função de seu objetivo e portanto, decompor estes componentes otimiza bastante o processo de desenvolvimento.

Durante o jogo, os pilares do pensamento computacional podem ser utilizados nas possíveis ocasiões:

- **DECOMPOSIÇÃO**: quando o jogador identifica um problema complexo e divide em partes menores para facilitar na resolução.
- **RECONHECIMENTO DE PADRÕES**: quando o jogador identifica um padrão específico que possa ser reutilizado para elaborar seu algoritmo de forma eficiente.
- **ABSTRAÇÃO**: quando o jogador consegue filtrar somente os dados relevantes, podendo assim, otimizar seu tempo e trabalho.
- **ALGORITMO**: quando o jogador consegue elaborar o passo a passo das instruções para resolver o problema.

Os dados mostram que o CodePlay foi capaz de chamar bastante a atenção do grupo de participantes, fazendo com que eles tivessem uma experiência muito boa, tanto pela sua interface gráfica como também pela sua dinâmica de desafios e de jogabilidade.

De acordo com os resultados obtidos, o jogo também demonstra ser uma plataforma de utilidade de cunho educacional, pois auxilia no ensino em disciplinas introdutórias de programação. Desse modo, o jogo estimula o pensamento computacional e ainda, consegue ser desafiador (fator bastante positivo para um jogo digital).

É importante enfatizar as contribuições deste trabalho, visto que, ele cooperou

diretamente no desenvolvimento da lógica de programação e para alguns participantes, colaborou no aprendizado de uma nova linguagem (o Python). Além disso, pode-se enfatizar também que o trabalho conseguiu contribuir no entendimento de alguns conteúdos abordados nas disciplinas do 1º ano de graduação. Portanto, isso implica que os resultados foram bons e que o objetivo principal foi alcançado.

Entretanto, também é importante deixar registrado as limitações deste trabalho, posto que, o jogo apenas interpreta algoritmos escritos em Python e qualquer código escrito em outra linguagem de programação não será aceito. Outro ponto é a ausência de alguns conteúdos que não foram abordados, tais como: função e recursividade.

O principal desafio deste trabalho foi a etapa de aplicação do jogo, onde os participantes testaram o CodePlay, jogando. Para isso, foi criado um grupo no WhatsApp com todos os participantes, com o intuito de informá-los como seria o teste e qual o seu objetivo. O grupo também serviu como ferramenta de suporte e tira-dúvidas.

Projetando trabalhos futuros, algumas sugestões podem ser implantadas para potencializar o presente trabalho, tais como:

1. Implementar um tutorial mais completo;
2. Implementar um botão para pular o teste de conhecimento;
3. Implementar um botão para silenciar o jogo;
4. Implementar um indicador de erro e/ou acerto, na tela de teste de conhecimento;
5. Implementar novas fases, aumentando o nível dos desafios;
6. Implementar um mecanismo de identificação, para sinalizar a casa que já teve seu desafio resolvido.

REFERÊNCIAS

- ALVES, M. A. B. Progster-um jogo digital infantil para o ensino de pensamento computacional. **Trabalho de Conclusão de Curso (Graduação em Design Digital)**, Universidade Federal do Ceará, Quixadá, 2020.
- ARNOLD, K.; GOSLING, J.; HOLMES, D. **The Java programming language**. [S. l.]: Addison Wesley Professional, 2005.
- BARSOTI, N.; GIBERTONI, D. Impacto que o sequelize traz para o desenvolvimento de uma api construída em node. js com express. js. **Revista Interface Tecnológica**, v. 17, n. 2, p. 231–243, 2020.
- BATISTA, M. B. d. A. Avaliação da interface do sistema web allocate por meio de teste de usabilidade. **Trabalho de Conclusão de Curso (Graduação em Ciência da Computação)**, Universidade Federal do Ceará, Quixadá, 2020.
- BBC. **Introduction to computational thinking**. Bitesize: [S. n.], 2019.
- DECIAN, M. Jogos digitais educacionais enquanto recurso para o ensino-aprendizagem da língua portuguesa. **SiB-UFSM**, Universidade Federal de Santa Maria, 2010.
- ESPÍNDOLA, Q. C. Retenção acadêmica nos cursos de licenciatura em ciências da natureza da unipampa. **Trabalho de Conclusão (Graduação) – Curso de Licenciatura em Ciências da Natureza**, Universidade Federal do Pampa, Dom Pedrito,RS, 2016.
- FALCÃO, T. P.; GOMES, T. C. S.; ALBUQUERQUE, I. R. O pensamento computacional através de jogos infantis: uma análise de elementos de interação. **Anais do XVI IHC-Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais-IHC**, 2015.
- FREIRE, L.; COUTINHO, J.; LIMA, V.; LIMA, N. Uma proposta de encontros de tutoria baseada em metodologias ativas para disciplinas de programação introdutória. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. [S. l.: s. n.], 2019. v. 8, n. 1, p. 298.
- GROS, B. The impact of digital games in education. **First Monday**, v. 8, n. 7, p. 6–26, 2003.
- HOED, R. M. Análise da evasão em cursos superiores: o caso da evasão em cursos superiores da área de computação. **UnB**, 2016.
- LEMOS, R.; BRAZ, M. A.; BARROSO, G.; MONTEIRO, I. Progster: Aprendendo lógica de programação com um tabuleiro eletrônico. **XVI Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais-IHC**, 2017.
- LIMA, Á.; DINIZ, M.; ELIASQUEVICI, M. Metodologia 7cs: Uma nova proposta de aprendizagem para a disciplina algoritmos. In: SBC. **Anais do XXVII Workshop sobre Educação em Computação**. [S. l.], 2019. p. 429–443.
- LIUKAS, L. **Hello Ruby: adventures in coding**. [S. l.: s. n.], 2015. v. 1.
- MACHADO, L. S.; MORAES, R. M.; NUNES, F. Serious games para saúde e treinamento imersivo. **Abordagens práticas de realidade virtual e aumentada**, SBC Porto Alegre, v. 1, p. 31–60, 2009.

- MARCHIONI, F. **MongoDB for Java developers**. [S. l.]: Packt Publishing Ltd, 2015.
- MARTINS, C. B. N. Evasão de alunos nos cursos de graduação em uma instituição de ensino superior. **Fundação Pedro Leopoldo**, FPL educacional, 2007.
- MICHEL, F.; PIRES, F.; PESSOA, M. Walgor: um jogo de tower defense para o desenvolvimento do pensamento computacional e apresentação de algoritmos computacionais. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. [S. l.: s. n.], 2019. v. 8, n. 1, p. 514.
- MORAES, J. d. P. B. Retenção discente no centro de ciências humanas e naturais da ufes: o caso dos cursos letras-português e geografia. **Dissertação (Pós-Graduação em Gestão Pública)**, Universidade Federal do Espírito Santo, 2015.
- MORATORI, P. B. Por que utilizar jogos educativos no processo de ensino aprendizagem. **UFRJ. Rio de Janeiro**, p. 04, 2003.
- MOURÃO, A. Uma proposta da eficiência do uso da metodologia ativa baseada em problemas, utilizando dojo de programação, aplicada na disciplina de lógica de programação. In: **Anais do Workshop de Informática na Escola**. [S. l.: s. n.], 2017. v. 23, n. 1, p. 667–676.
- SBC. Ensino de computação na educação básica. **Diretrizes para ensino de Computação na Educação Básica**, Sociedade Brasileira de Computação, 2018.
- SHOOP, R.; FLOT, J. Helping students build conceptual models. **Carnegie Mellon Robotics Academy**, 2016.
- SOUZA, P. R. de A.; DIAS, L. R. Kodu game labs: Estimulando o raciocínio lógico através de jogos. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. [S. l.: s. n.], 2012. v. 23, n. 1.
- STEFANELLO, D. R.; FILHO, J. A. L.; GUIMARÃES, I. A.; MARTINS, S. G.; ALMEIDA, W. R. A contribuição cognitiva da robótica educacional como ferramenta interdisciplinar no contexto do ensino superior. **II Seminário Diálogos em Educação a Distância e XIII Encontro para ações em EaD na FURG**, p. 66, 2013.
- WANG, Z.; TANG, F.; YU, Z. L. Design and implementation of a health status reporting system based on spring boot. In: IEEE. **2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE)**. [S. l.], 2020. p. 453–457.
- WANGENHEIM, C. G. V.; NUNES, V. R.; SANTOS, G. D.; ALVES, N. C.; COAN, E. S.; MANSUR, C. Resumo de objetivos de aprendizagem de computação no ensino fundamental. **Currículo de Referência CSTA/ACM K-12**, 2014.
- WING, J. M. Computational thinking. **Communications of the ACM**, ACM New York, NY, USA, v. 49, n. 3, p. 33–35, 2006.

APÊNDICE A – GAM DESIGN DOCUMENT

CodePlay

AUTOR: EDUARDO DA SILVA GOMES

VERSÃO: 2.0

QUIXADÁ

2023

SUMÁRIO

| | | |
|---|--|---|
| 1 | INTRODUÇÃO | 3 |
| 2 | HISTÓRIA | 3 |
| 3 | GAMEPLAY | 3 |
| | 3.1 Mecânica e controles | 3 |
| | 3.2 Desafio(s) | 4 |
| | 3.3 Como ganhar | 4 |
| | 3.4 Como perder | 4 |
| 4 | DINÂMICA DA RESOLUÇÃO DO DESAFIO | 4 |
| 5 | PERSONAGEM | 4 |
| 6 | INIMIGOS | 5 |
| 7 | INTERFACE | 5 |
| 8 | EFEITOS SONOROS | 8 |

3. INTRODUÇÃO

Este documento tem o intuito de apresentar a história, mecânica, interface e os elementos de interação do jogo CodePlay, bem como, seu objetivo que é desenvolver o pensamento computacional para elaborar soluções de pequenos problemas.

4. HISTÓRIA

O jogo traz a história de um personagem chamado Dio que se encontra no mapa, ele terá que percorrer o mapa e chegar em uma determinada casa, onde encontrará os desafios. Durante o jogo, o usuário deverá identificar, analisar e pensar como solucionar os desafios (problemas de programação envolvendo os conteúdos citados no Quadro 1, tais soluções deverão ser escritas em Python). Dito isso, ele aprenderá a utilizar os fundamentos do pensamento computacional, para auxiliar na elaboração e compreensão dos passos para gerar o algoritmo responsável pela resolução dos desafios.

O jogo foi desenvolvido para computadores, essencialmente para rodar em navegadores. O design do CodePlay foi planejado de modo que o torne intuitivo o bastante para que os discentes tenham uma experiência lúcida, pois o seu objetivo é motivar o uso dos conceitos do PC integrado aos JDE, para a resolução de problemas.

5. GAMEPLAY

O jogador deve locomover o personagem pelo mapa e localizar uma determinada casa, como já dito anteriormente. No entanto, no decorrer de sua trajetória deve-se evitar os inimigos, pois a colisão com eles, pode ocasionar o surgimento de um teste lógico ou até mesmo Game Over, caso seja um teste, o jogador só terá acesso ao restante do caminho pelo mapa se acertá-lo, caso contrário, pode ocasionar o anulamento de seus pontos.

3.1 Mecânica e Controles

O personagem poderá se movimentar de acordo com a escolha do jogador. Para isso, deve-se utilizar as setas do teclado, que irá permitir os seguintes direcionamentos:

1.  : Ao utilizar essa tecla, Dio executa a ação de subir.
2.  : Ao utilizar essa tecla, Dio executa a ação de se mover para a esquerda.
3.  : Ao utilizar essa tecla, Dio executa a ação de se mover para a direita.
4.  : Ao utilizar essa tecla, Dio executa a ação de descer.

3.2 Desafio(s)

O CodePlay aborda o(s) seguinte(s) desafio(s):

- **Desafio Inicial:** Trata-se de um teste, onde o jogador terá que responder um questionário que contém 10 questões. Caso o jogador acerte 7 ou mais questões, ele passará para próxima etapa.
- **Desafio Principal:** Consiste na resolução dos problemas de programação, onde cada casa terá um problema e se por acaso o personagem passe ou volte à mesma casa mais de uma vez, a mecânica do jogo deve devolver outro problema a ele. Nesse sentido, isso funciona através de um mecanismo que gera valores aleatórios por meio de uma função *Random*, e com isso, possibilita chamar arbitrariamente uma questão a cada interação do personagem com uma determinada casa.

3.3 Como ganhar

O jogador só vencerá o jogo, caso ele consiga resolver 3 desafios. Para isso ele terá que passar pelo teste de conhecimento e ao chegar no mapa, possivelmente terá que desviar dos inimigos para conseguir alcançar este objetivo de forma mais eficaz. Neste caso, irá aparecer a tela de Parabéns, ela pode-se observada na Figura 7.

3.4 Como perder

O jogador somente irá perder no jogo quando colidir com um determinado inimigo. Neste caso, aparecerá a tela de *game over* (tradução literal seria “fim de jogo”, mas neste caso, indica derrota). Na Figura 8, pode-se observar a tela de *Game Over*.

6. DINÂMICA DA RESOLUÇÃO DO DESAFIO

No mapa do jogo tem cinco casas e cada uma terá um desafio, que pode ser relacionado a condição, repetição ou operadores matemáticos. A resolução dos desafios deve se dar em uma área específica que é melhor apresentada na Figura 6, onde o jogador deve inserir seu algoritmo e submeter a resolução no ato do click no botão “Enviar”.

7. PERSONAGEM

O jogo tem apenas um personagem, conforme a Figura 1:

Figura 1 - Personagem Dio



Fonte: <https://pipoya.itch.io/>

8. INIMIGOS

Na Figura 2, é possível observar alguns inimigos do jogo.

Figura 2 - Inimigos



Fonte: <https://pipoya.itch.io/>

9. INTERFACE

O design do jogo foi pensado de forma que o torne claro e intuitivo o bastante para gerar no jogador uma experiência lúcida e que mantenha o seu objetivo. O jogo foi desenvolvido com as seguintes telas:

1. **Tela inicial:** Na Figura 3, pode-se observar a primeira tela que aparece ao executar o jogo. O intuito dela é dar as boas vindas e descrever um pouco sobre como será o jogo.

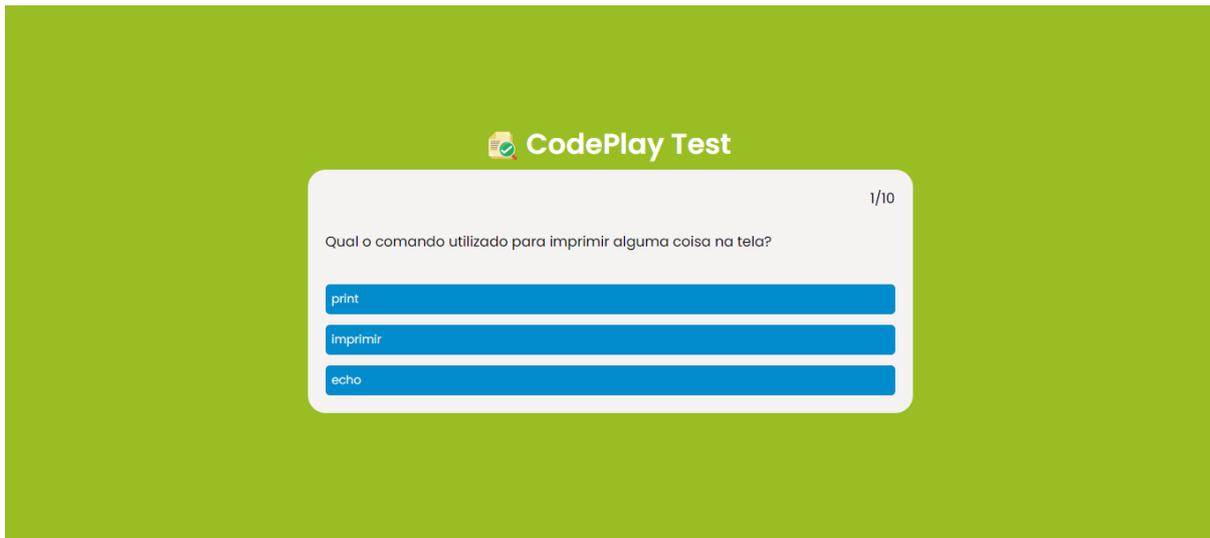
Figura 3 - Tela Inicial



Fonte: elaborado pelo autor (2023).

2. **Tela de teste de conhecimento:** Na Figura 4, pode-se observar a segunda tela do jogo. Trata-se de um teste de conhecimento sobre lógica de programação. O teste é composto por 10 perguntas, caso o jogador acerte 7 ou mais, ele estará apto para a próxima etapa.

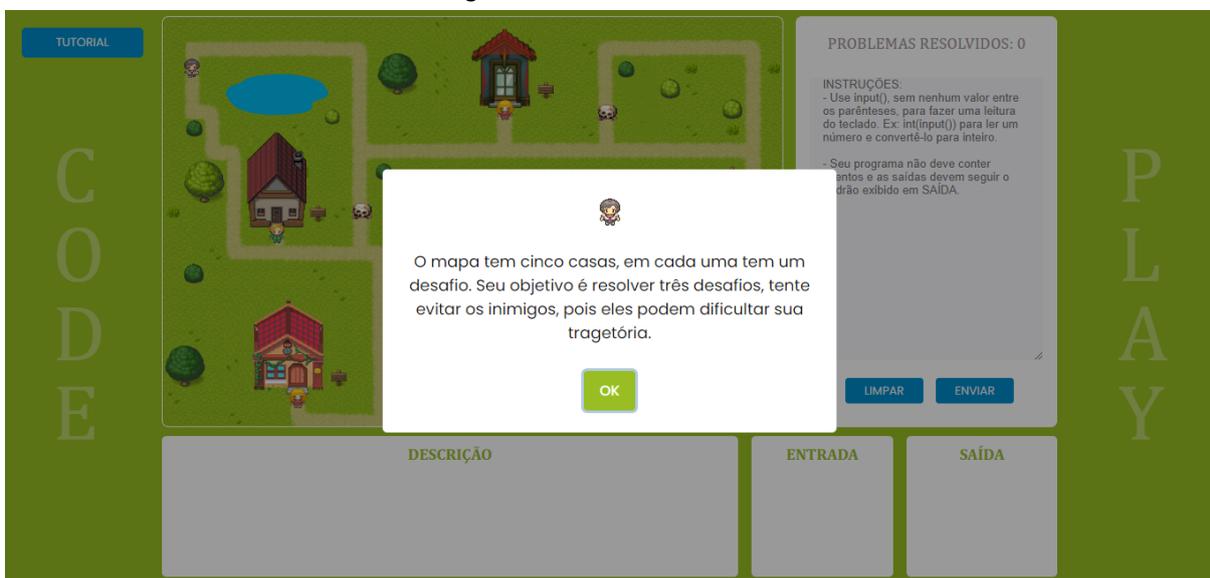
Figura 4 - Tela de teste de conhecimento



Fonte: elaborado pelo autor (2023).

3. **Tela de Tutorial:** Na Figura 5, pode-se observar a tela de tutorial do jogo. Nesta tela, aparece um modal onde o personagem passa as orientações necessárias para jogar.

Figura 5 - Tela de Tutorial



Fonte: elaborado pelo autor (2023).

4. **Tela principal:** Na Figura 6, pode-se observar a terceira e principal tela do jogo. Nesta tela, são apresentados os recursos fundamentais para jogar o CodePlay, como o mapa para o deslocamento do personagem, os *cards* com as informações necessárias para o entendimento dos desafios e o formulário para o envio do código. Além disso, a tela ainda conta com um tutorial na parte superior esquerda.

Figura 6 - Tela principal



Fonte: elaborado pelo autor (2023).

5. **Tela de Parabéns:** Na Figura 7, pode-se observar a tela de parabéns do jogo. O surgimento desta tela ocorre quando o jogador alcança o objetivo do jogo, resolvendo os 3 desafios.

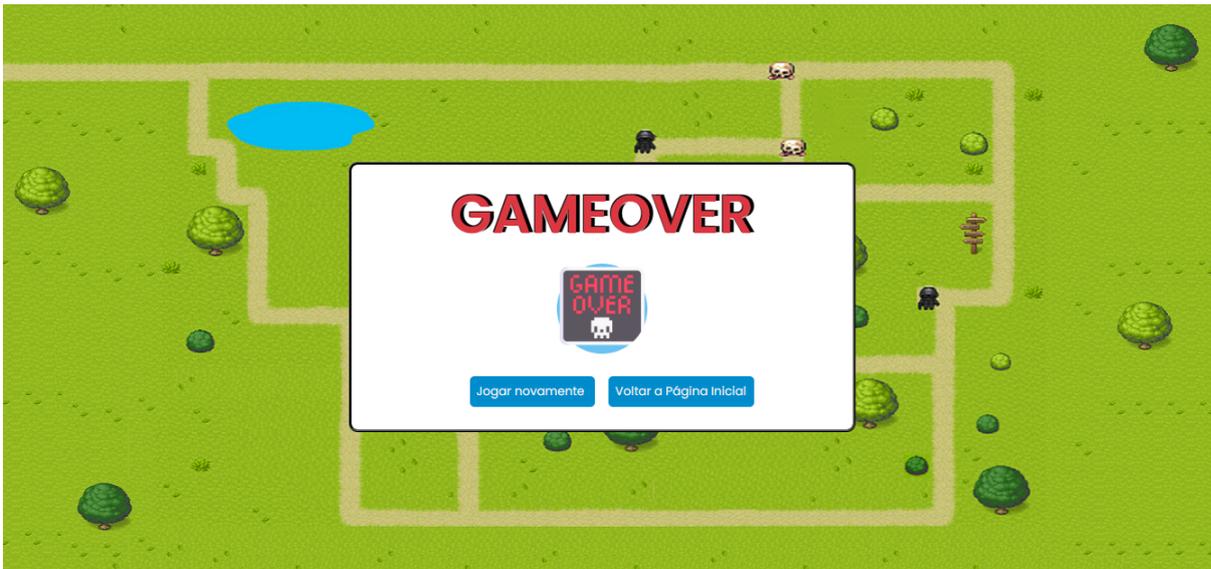
Figura 7 - Tela de Parabéns



Fonte: elaborado pelo autor (2023).

6. **Tela de GAME OVER:** Na Figura 8, pode-se observar a tela de *game over* do jogo. O surgimento desta tela ocorre quando o personagem cruza um de seus inimigos, falhando na missão. Nesta tela, o jogador tem a opção de jogar novamente ou voltar para a tela inicial do jogo.

Figura 8 - Tela de Game Over



Fonte: elaborado pelo autor (2023).

Os demais elementos de interface foram retirados dos seguintes sites:

1. <https://opengameart.org/>
2. <https://www.artstation.com/>
3. <https://forum.gdevelop.io/>
4. <https://img.itch.zone/>
5. <https://www.flaticon.com/>

8. EFEITOS SONOROS

Os sons utilizados no desenvolvimento do jogo foi retirados do site a seguir:

1. <https://pixabay.com/>

APÊNDICE B - ALGORITMO PARA SOLUÇÃO DO PROBLEMA

```
1 word=input()  
2 count=0  
3 for x in word:  
4     count=count+1  
5 print(count)
```

APÊNDICE C – QUESTIONÁRIO SOBRE O JOGO

Questão 1. Algo interessante chamou sua atenção durante o jogo? Se sim, conte o que foi.
(*Questão subjetiva*).

Questão 2. Minha experiência foi muito boa ao jogar o CodePlay:

- (a) Discordo
- (b) Discordo Totalmente
- (c) Concordo
- (d) Concordo Totalmente

Questão 3. Achei o design atraente:

- (a) Discordo
- (b) Discordo Totalmente
- (c) Concordo
- (d) Concordo Totalmente

Questão 4. Considero o jogo desafiador:

- (a) Discordo
- (b) Discordo Totalmente
- (c) Concordo
- (d) Concordo Totalmente

Questão 5. Considero o jogo uma ferramenta de auxílio para o ensino, em Disciplinas Introdutórias de Programação:

- (a) Discordo
- (b) Discordo Totalmente
- (c) Concordo
- (d) Concordo Totalmente

Questão 6. Concordo que o jogo estimula de alguma forma o pensamento computacional. (Entende-se como pensamento computacional, uma forma inteligente de resolver um problema, estimulando o pensamento crítico e analítico):

- (a) Sim
- (b) Não

Questão 7. O jogo foi mais difícil do que eu imaginava:

- (a) Discordo

- (b) Discordo Totalmente
- (c) Concordo
- (d) Concordo Totalmente

Questão 8. Atingir o objetivo do jogo me deu um sentimento de realização:

- (a) Discordo
- (b) Discordo Totalmente
- (c) Concordo
- (d) Concordo Totalmente

Questão 9. Após o jogo sinto que consigo compreender melhor os conteúdos que foram abordados:

- (a) Discordo
- (b) Discordo Totalmente
- (c) Concordo
- (d) Concordo Totalmente

Questão 10. Os conteúdos abordados no jogo contribuíram de alguma forma para a sua aprendizagem na programação? Se sim, como? (*Questão subjetiva*).

Questão 11. Apesar de ser um jogo educacional, você jogaria novamente?

- (a) Sim
- (b) Não

Questão 12. O tutorial foi suficiente para você conseguir jogar?

- (a) Sim
- (b) Não

Questão 13. Você recomendaria o jogo para um colega de turma?

- (a) Sim
- (b) Não

Questão 14. Qual sua sugestão de melhoria? (*Questão subjetiva*).