

UNIVERSIDADE FEDERAL DE SANTA MARIA  
CENTRO DE TECNOLOGIA  
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Diego Viana dos Santos

**PLATAFORMA ONLINE PARA CRIAÇÃO DE JOGOS  
PERSONALIZADOS COM LABIRINTOS PARA INTRODUÇÃO À  
LÓGICA DE PROGRAMAÇÃO**

Santa Maria, RS  
2022

**Diego Viana dos Santos**

**PLATAFORMA ONLINE PARA CRIAÇÃO DE JOGOS PERSONALIZADOS  
COM LABIRINTOS PARA INTRODUÇÃO À LÓGICA DE PROGRAMAÇÃO**

Trabalho Final de Graduação apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Sistemas de Informação**.

ORIENTADORA: Profa. Andrea Schwertner Charão

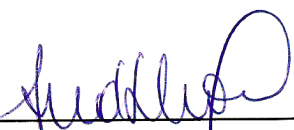
Santa Maria, RS  
2022

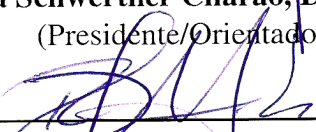
**Diego Viana dos Santos**

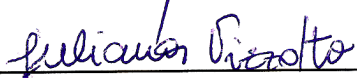
**PLATAFORMA ONLINE PARA CRIAÇÃO DE JOGOS PERSONALIZADOS  
COM LABIRINTOS PARA INTRODUÇÃO À LÓGICA DE PROGRAMAÇÃO**

Trabalho Final de Graduação apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Sistemas de Informação**.

**Aprovado em 16 de agosto de 2022:**

  
\_\_\_\_\_  
**Andrea Schwertner Charão, Dra. (UFSM)**  
(Presidente/Orientadora)

  
\_\_\_\_\_  
**Roseclea Duarte Medina, Dra. (UFSM)**

  
\_\_\_\_\_  
**Juliana Kaizer Vizzotto, Dra. (UFSM)**

Santa Maria, RS  
2022

# PLATAFORMA ONLINE PARA CRIAÇÃO DE JOGOS PERSONALIZADOS COM LABIRINTOS PARA INTRODUÇÃO À LÓGICA DE PROGRAMAÇÃO

## ONLINE PLATFORM FOR CREATING CUSTOM GAMES WITH MAZES AS AN INTRODUCTION TO PROGRAMMING LOGIC

Diego dos Santos Viana<sup>1</sup>, Andrea Schwertner Charão<sup>2</sup>

### RESUMO

Aprender e ensinar linguagem de programação na maioria das vezes é um grande desafio. O uso de linguagem baseada em blocos tem se mostrado promissor neste cenário. Usualmente, o ensino de programação com blocos é aplicado em forma de jogos ou tutoriais, tornando o ambiente de aprendizagem mais amigável. Essa abordagem permite que os blocos possam ser encaixados uns aos outros para criar uma forma de código semelhante a um quebra-cabeça, em vez de comandos de texto. Em um trabalho anterior, foi desenvolvido um gerador de jogos personalizados com labirintos para Introdução à Lógica de Programação utilizando linguagem baseada em blocos (Bastos, 2019). No presente trabalho, é apresentada a plataforma online My Blockly Maze, uma aplicação que adiciona funcionalidades no gerador desenvolvido anteriormente, transformando-o em uma plataforma onde o usuário consegue criar, jogar, gerenciar e compartilhar os seus jogos. Por meio dos testes realizados observou-se que a plataforma facilita o compartilhamento dos jogos e é um bom local para o primeiro contato e exercício de lógica de programação.

**Palavras-chave:** Programação com blocos. Linguagem baseada em blocos. Introdução à Lógica de Programação.

### ABSTRACT

Learning and teaching programming language are most of the time a great challenge. The use of a block-based language has shown promise in this scenario. Usually, the teaching of programming with blocks is applied in the form of games or tutorials, making the learning environment more friendly. This approach allows blocks to be snapped together to create a puzzle-like form of code, rather than text commands. In previous work, a custom game generator with mazes was developed for the Introduction to Programming Logic using a block-based language (Bastos, 2019). In the present work, the online platform My Blockly Maze is presented, an application that adds functionality to the previously developed generator, transforming it into a platform where the user can create, play, manage and share their games. Through the tests carried out, it was observed that the platform facilitates the sharing of games and is a good place for the first contact and programming logic exercise.

**Keywords:** Programming with blocks. Block-based language. Introduction to Programming Logic.

---

<sup>1</sup> Autor: Graduando em Bacharelado em Sistemas de Informação pela Universidade Federal de Santa Maria.

<sup>2</sup> Orientadora: Professora do Departamento de Linguagens e Sistemas de Computação da Universidade Federal de Santa Maria.

## LISTA DE FIGURAS

Figura 1 – Interface do jogo de labirinto. . . . .	11
Figura 2 – Exemplo de código textual gerado após a conclusão do nível. . . . .	12
Figura 3 – Editor da plataforma Kodetu. . . . .	13
Figura 4 – Interface do gerador de jogos proposto por Bastos. . . . .	13
Figura 5 – Interface do gerador de jogos. . . . .	14
Figura 6 – Diagrama de casos de uso de login e cadastro. . . . .	15
Figura 7 – Protótipo da tela de criação de conta. . . . .	16
Figura 8 – Protótipo da tela de fazer login. . . . .	16
Figura 9 – Diagrama de casos de uso de interação com os jogos . . . . .	17
Figura 10 – Protótipo da tela de visualização dos jogos da comunidade. . . . .	17
Figura 11 – Protótipo da tela de visualização de detalhes dos jogos. . . . .	18
Figura 12 – Protótipo da tela de pesquisa por um jogo. . . . .	18
Figura 13 – Protótipo da tela de redirecionamento para criação do jogo. . . . .	19
Figura 14 – Protótipo da tela de gerenciamento dos jogos. . . . .	19
Figura 15 – Arquitetura da plataforma. . . . .	20
Figura 16 – Diagrama Entidade-Relacionamento. . . . .	20
Figura 17 – Protótipo da tela de criação dos jogos. . . . .	22
Figura 18 – Modal com instruções para criação dos jogos. . . . .	23
Figura 19 – Modal de confirmação da exclusão do último nível criado. . . . .	24
Figura 20 – Modal de notificação de erro. . . . .	24
Figura 21 – Execução de um jogo personalizado. . . . .	25
Figura 22 – Modal com instruções para jogar. . . . .	25

## LISTA DE TABELAS

Tabela 1 – Atores do sistema . . . . .	15
Tabela 2 – Perguntas do formulário . . . . .	27

## SUMÁRIO

1	Introdução . . . . .	8
2	Fundamentação e trabalhos relacionados . . . . .	9
	2.1 Programação visual com blocos . . . . .	10
	2.2 Blockly e Blockly Games . . . . .	10
	2.3 Desafios de programação com labirintos . . . . .	11
3	Visão geral do projeto . . . . .	13
4	Definição dos requisitos . . . . .	14
	4.1 Casos de uso do acesso do usuário . . . . .	15
	4.2 Casos de uso de interação com os jogos . . . . .	15
	4.2.1 Visualizar jogos da comunidade . . . . .	16
	4.2.2 Visualizar detalhes do jogo . . . . .	16
	4.2.3 Pesquisar por um jogo . . . . .	17
	4.2.4 Criar um jogo . . . . .	18
	4.2.5 Gerenciar jogos criados . . . . .	18
5	Desenvolvimento da plataforma . . . . .	19
	5.1 Banco de dados . . . . .	19
	5.2 Back-end . . . . .	21
	5.3 Front-end . . . . .	21
	5.4 Implementação da criação de jogos com labirintos . . . . .	22
	5.5 Implementação da execução dos jogos . . . . .	23
6	Testes com usuários . . . . .	25
	6.1 Aplicação do teste de usabilidade . . . . .	26
	6.2 Resultados . . . . .	28
7	Conclusão . . . . .	28
	<b>Referências . . . . .</b>	<b>29</b>

# 1 INTRODUÇÃO

A programação de computadores é frequentemente discutida como uma tarefa complicada e exigente, sendo particularmente difícil para estudantes iniciantes (MEDEIROS; RAMALHO; FALCÃO, 2019; LAHTINEN; ALA-MUTKA; JÄRVINEN, 2005). Sabe-se que a programação se relaciona com conhecimentos, habilidades e competências que vão muito além da codificação de um programa, passando pelo raciocínio lógico e abstrato, por estratégias de resolução de problemas, e pela construção de algoritmos reutilizáveis.

Frequentemente, a programação aparece associada às dimensões do Pensamento Computacional (PC), que “*é considerado uma competência importante para o século XXI e está presente em diversas experiências de Educação em Computação apresentadas na literatura*” (SANTANA; CHAVEZ; BITTENCOURT, 2021). Com raízes históricas no trabalho de Papert (1980) e impulsionado pela visão de Wing (2006), o Pensamento Computacional pode ser descrito como um processo de resolução de problemas por meio de habilidades cognitivas, constituindo um tema atual de estudos, discussões e ações. Segundo Guarda e Pinto (2020), a programação pode ser vista como uma habilidade que ora contribui para o PC, ora é impulsionada pelo PC.

Em uma pesquisa recente Silva e Meneghetti (2019) é feita uma análise entre PC e as competências das áreas de linguagens, matemática, ciências da natureza e ciências humanas da Base Nacional Comum Curricular (BNCC). O estudo mostra que o Pensamento Computacional pode auxiliar na resolução de problemas e na forma de pensar em diferentes áreas do conhecimento.

Xavier et al. (2021) relaciona os conceitos de PC com as habilidades propostas na BNCC para a temática da Matemática, apresentando uma lista de estudos sobre a viabilidade de fomentar o PC nas subáreas da Matemática.

Neste cenário, nota-se que muitos estudos e intervenções têm abordado a programação para iniciantes, beneficiando-se de uma grande variedade de ambientes e plataformas concebidos para popularizar fundamentos de Computação e desenvolver aspectos do PC (TIKVA; TAMBOURIS, 2021). Uma abordagem que tem se consolidado neste nicho é a **programação visual baseada em blocos**. Nesses ambientes, os blocos podem ser encaixados uns aos outros para criar uma forma de código semelhante a um quebra-cabeça, em vez de comandos de texto (ALRUBAYE; LUDI; MKAOUER, 2019; DOUKAKIS, 2019).

O uso de programação baseada em blocos visuais em ambientes educacionais tem como premissa familiarizar, motivar e facilitar o início da aprendizagem em programação (LADIAS; KARVOUNIDIS; LADIAS, 2020; MOORS; LUXTON-REILLY; DENNY, 2018), permitindo que os iniciantes se concentrem nos conceitos, deixando de lado a complexidade sintática de uma linguagem baseada em texto (GOMEZ; MORESI; BENOTTI, 2019). Os ganhos de aprendizagem desenvolvidos pelos estudantes ao utilizarem linguagem baseada em blocos poderão ser aplicados em linguagens de programação baseadas em texto (WEINTROP; WILENSKY, 2019).

A programação visual baseada em blocos é suportada em muitos ambientes e linguagens (SILVA; FERREIRA, 2022). Alguns exemplos proeminentes são Scratch<sup>3</sup>, App Inventor<sup>4</sup> e Blockly<sup>5</sup>, que alcançaram uma grande quantidade de usuários e têm

<sup>3</sup> <https://scratch.mit.edu/>

<sup>4</sup> <https://appinventor.mit.edu/>

<sup>5</sup> <https://developers.google.com/blockly>



sido usados em muitos estudos. Estas e outras ferramentas estão presentes na plataforma Code.org<sup>6</sup>, que é uma plataforma de alcance mundial projetada para desmistificar os “códigos”, demonstrando que qualquer pessoa pode aprender o básico e expandir sua participação na área de Ciência da Computação. Nela existe uma grande variedade de materiais em formato de jogos e tutoriais, que podem apoiar o desenvolvimento do PC e da lógica de programação.

Um desafio comumente proposto em ambientes de programação visual tem a forma de um labirinto ou caminho, que deve ser percorrido por um personagem programável por meio de blocos encaixáveis, até atingir um determinado objetivo. Este tipo de desafio está presente em diversos tutoriais na plataforma Code.org, assim como em atividades com Scratch<sup>7</sup> e em jogos como LightBot<sup>8</sup>. Para resolver um desafio deste tipo, que pode ter níveis crescentes de dificuldade, utilizam-se conceitos básicos de programação imperativa e estruturada, como sequências, condicionais, repetições e até mesmo subprogramas (procedimentos ou funções).

Apesar de ser um desafio recorrente, nota-se que os labirintos propostos são geralmente pré-estabelecidos, ou seja, as plataformas online usualmente não oferecem meios para criação de labirintos com características personalizáveis. Estima-se que esta possibilidade poderia potencializar os benefícios da programação visual, envolvendo sujeitos (estudantes, instrutores, professores ou público em geral) também na concepção dos desafios.

No trabalho de Bastos (2019), foi desenvolvida uma ferramenta capaz de personalizar o jogo Maze do Blockly Games<sup>9</sup>, com a inserção de imagens de fundo e com um editor de caminhos para traçado de novos labirintos. A ferramenta é uma aplicação web que executa totalmente do lado do cliente, sem necessidade de servidor web, podendo funcionar offline. Os jogos criados com a ferramenta ficam salvos no computador do usuário, que pode manualmente hospedar o jogo em um servidor web ou usá-lo em demonstrações sem necessidade de conexão à Internet (caso de uso adaptado a eventos em locais com baixa conectividade).

No presente trabalho, tem-se como objetivo desenvolver uma plataforma online para iniciantes em programação e pessoas em geral que permita criar, gerenciar, compartilhar, visualizar e jogar jogos com labirintos, estendendo o trabalho de Bastos (2019). Esta plataforma poderá ser acessada a partir de qualquer navegador web, facilitando o processo de criação e compartilhamento de desafios personalizados baseados em labirintos.

O restante deste texto está organizado da seguinte forma: na seção 2, apresenta-se uma fundamentação para o trabalho e discute-se trabalhos relacionados; na seção 3, apresenta-se uma visão geral do projeto da plataforma, seguida da definição de requisitos (seção 4) e de aspectos do seu desenvolvimento (seção 5); na seção 6, apresenta-se os testes realizados; ao final, a seção 7 apresenta a conclusão e trabalhos futuros.

## 2 FUNDAMENTAÇÃO E TRABALHOS RELACIONADOS

Este trabalho se insere em uma temática recorrente, que articula pesquisa e desenvolvimento de artefatos computacionais de apoio à iniciação à programação. A produção

---

<sup>6</sup> <https://code.org>

<sup>7</sup> <https://www.create-learn.us/blog/how-to-make-a-maze-in-scratch/>

<sup>8</sup> <https://lightbot.com>

<sup>9</sup> <https://blockly.games/>

bibliográfica nesta temática é profícua e tem sido alvo de revisões sob diferentes ângulos e delimitações. Por exemplo, [Combéfis, Beresnevicus e Dagiené \(2016\)](#) analisam diversos trabalhos sobre aprendizagem de programação apoiada em jogos e desafios, sejam com linguagens textuais como visuais. Em [Bastos \(2019\)](#), tem-se uma síntese de alguns ambientes proeminentes na temática em questão (Logo, Scratch, MIT App Inventor, LightBot, RoboMind e CodeCombat). Mais recentemente, [Silva e Ferreira \(2022\)](#) trazem uma revisão de literatura concentrada na programação visual.

Diante disso, decidiu-se focar esta seção no referencial que está o mais diretamente relacionado ao tema deste trabalho, a saber: a programação visual com blocos, a biblioteca Blockly e jogos/desafios de programação baseados em labirintos.

## 2.1 Programação visual com blocos

A programação de computadores tem crescido em popularidade nos últimos anos, impulsionada por iniciativas educacionais, por ações em escala mundial e por avanços tecnológicos. Esse interesse pela programação floresceu particularmente nos mais jovens. Devido a essa transformação, várias abordagens e técnicas de programação precisam ser adotadas para ajudar iniciantes a se acostumarem com os princípios básicos de programação de maneira eficaz. Essa preocupação levou ao design e desenvolvimento de diversos ambientes de programação baseados em blocos.

Alguns dos ambientes de programação baseados em blocos mais populares para educação são Scratch<sup>10</sup> e Blockly. Esses ambientes usam blocos que podem ser anexados uns aos outros e criam uma forma de código semelhante a um quebra-cabeça, em vez de comandos de texto.

Os benefícios da utilização deste tipo de ambientes de programação, de acordo com pesquisas recentes, podem ser resumidos no seguinte:

- os ambientes de programação baseados em blocos podem facilitar o processo de aprendizagem de programação e minimizar os equívocos dos iniciantes sobre os princípios de programação ([MLADENOVIC; BOLJAT; ŽANKO, 2018](#));
- através do uso de ambientes de programação baseados em blocos, os iniciantes podem dominar conceitos fundamentais de codificação ([KANDEMIR; KALELIOGLU; GULBAHAR, 2021](#));
- através do uso de ambientes de programação baseados em blocos, os iniciantes se concentram nos conceitos de programação, deixando de lado a complexidade da sintaxe de uma linguagem de programação baseada em texto ([GOMEZ; MORESI; BENOTTI, 2019](#)).

## 2.2 Blockly e Blockly Games

Blockly<sup>11</sup> é uma biblioteca implementada em JavaScript que permite adicionar um editor de código visual a aplicações web. O editor Blockly usa blocos gráficos interligados para representar conceitos de programação como variáveis, expressões lógicas, laços, condicionais, entre outros. Os blocos podem ser rotulados com textos (por exemplo

---

<sup>10</sup> <https://scratch.mit.edu/>

<sup>11</sup> <https://developers.google.com/blockly>

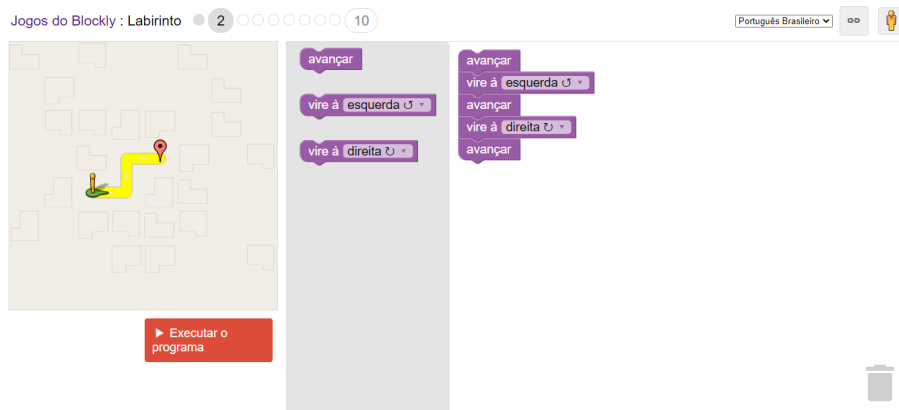
repeat ou repita), sendo que a biblioteca tem suporte a diferentes idiomas, incluindo português. Mais do que um editor, Blockly provê também a funcionalidade de conversão do código baseado em blocos para linguagens textuais (JavaScript ou Python, por exemplo).

O código da biblioteca Blockly é gratuito e aberto<sup>12</sup>. Segundo informações em sua documentação oficial, Blockly é usado em centenas de projetos, dentre os quais destaca-se MIT App Inventor e muitos tutoriais da plataforma Code.org. Blockly também é uma das bibliotecas usadas para criar blocos da linguagem Scratch 3.0<sup>13</sup> (versão da linguagem lançada em 2019). Blockly, no entanto, não é por si só uma linguagem visual, mas sim um recurso para desenvolvedores criarem aplicações e até mesmo novas linguagens baseadas em blocos (PASTERNAK; FENICHEL; MARSHALL, 2017).

O site Blockly Games<sup>14</sup> reúne jogos educativos criados com Blockly, formando uma trilha de desafios a serem solucionados por meio de programação visual com blocos. Além de terem propósito educacional, os jogos também servem de exemplo para desenvolvedores com Blockly, uma vez que todos os desafios têm código aberto<sup>15</sup>.

Dentre os jogos do Blockly Games, tem-se o jogo Maze (ou Labirinto, em português), que apresenta uma sequência de desafios em que um personagem deve ser programado para percorrer um caminho. Para resolver cada desafio proposto pelo jogo, é necessário mover e encaixar blocos para que o personagem consiga chegar ao objetivo final. Nesse jogo, que começa de forma simples e vai ficando mais desafiador a cada nível, o usuário se familiariza com sequências, condicionais e laços. A interface do jogo de labirinto pode ser vista na Figura 1.

Figura 1 – Interface do jogo de labirinto.



Fonte: Adaptado de <https://blockly.games/>

Na conclusão do nível é apresentado o código de programação textual gerado pela programação em bloco, como mostrado na Figura 2.

### 2.3 Desafios de programação com labirintos

O jogo Maze do Blockly Games é apenas um dentre diversos exemplos de desafios de programação baseados em labirintos. Na plataforma Code.org, por exemplo, este tema

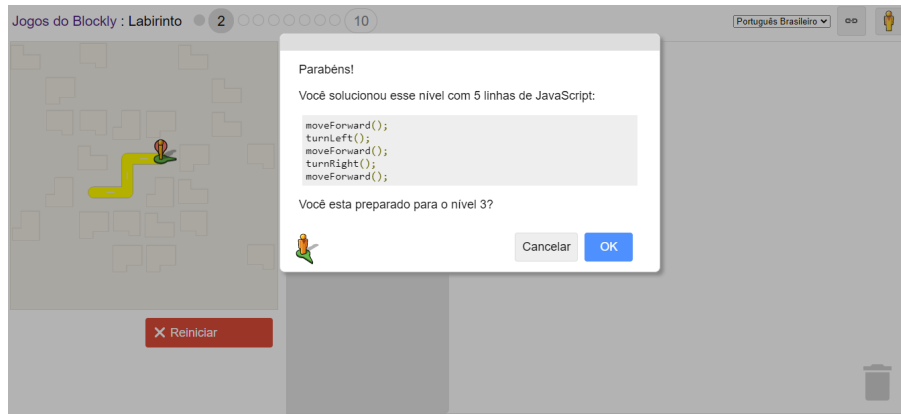
<sup>12</sup> <https://github.com/google/blockly>

<sup>13</sup> <https://en.scratch-wiki.info/wiki/Blockly>

<sup>14</sup> <https://blockly.games/>

<sup>15</sup> <https://github.com/google/blockly-games/wiki>

Figura 2 – Exemplo de código textual gerado após a conclusão do nível.



Fonte: Adaptado de <https://blockly.games/>

aparece em tutoriais curtos (Hour of Code<sup>16</sup>) e também em trilhas de maior duração<sup>17</sup>. Labirintos também são encontrados como tema de atividades de programação em robótica educacional (EVRIPIDOU et al., 2020), que também se beneficia da programação visual baseada em blocos.

Na literatura, há trabalhos que empregam desafios de programação com labirintos em contextos educacionais. Por exemplo, Martins, Reis e Marques (2016) analisa sua utilização com estudantes do Ensino Fundamental, em uma escola no Brasil. Já Guenaga et al. (2021) utilizam o desafio em uma plataforma concebida para estudar em detalhes o desenvolvimento do Pensamento Computacional junto a um grande público de estudantes na Espanha.

Em sua grande maioria, os desafios com labirintos são pré-definidos, ou seja, as plataformas oferecem ao usuário a possibilidade de resolução, não de criação de desafios. Uma exceção é a plataforma Kodetu<sup>18</sup>, que oferece um editor, embora o foco de seus desenvolvedores fosse a coleta de dados sobre a resolução dos desafios por parte dos estudantes (GUENAGA et al., 2021).

Kodetu é uma plataforma que permite desenvolver o Pensamento Computacional. Nela, os desafios são apresentados na forma de labirintos semelhantes ao modelo do jogo Maze do Blockly Games. Um ponto que difere está na forma em que os desafios são expostos – por exemplo, em alguns níveis, o usuário é desafiado a elaborar um código com blocos que solucione o trajeto de dois labirintos. Ao concluir, não é exibido o código em programação textual relativo à solução adotada em programação com blocos, mas sim uma mensagem de parabenização ou uma escala de satisfação do desafio proposto.

No editor disponível na plataforma Kodetu (Figura 3), é possível elaborar os caminhos dos labirintos, escolher a aparência do personagem e configurar alguns ajustes no funcionamento do jogo. Observou-se que o processo de criação do labirinto possui um maior nível de dificuldade e não disponibiliza ao usuário a possibilidade de adicionar uma imagem de fundo personalizada.

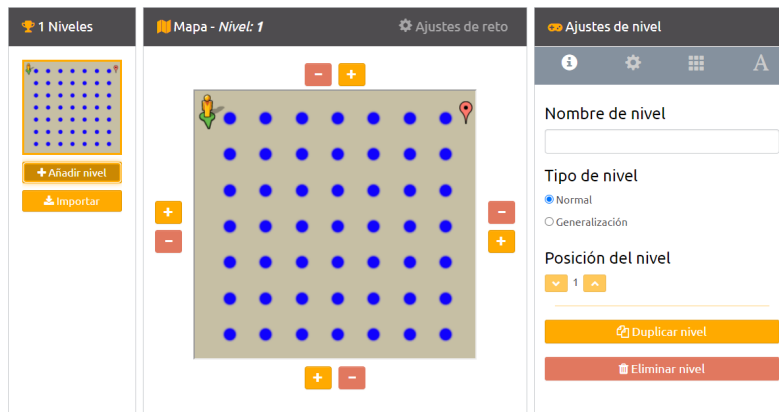
O trabalho de Bastos (2019), que serviu de ponto de partida para o presente trabalho, avançou no sentido da criação de desafios com labirintos, fornecendo ao usuário

<sup>16</sup> <https://code.org/hourofcode>

<sup>17</sup> <https://studio.code.org/courses>

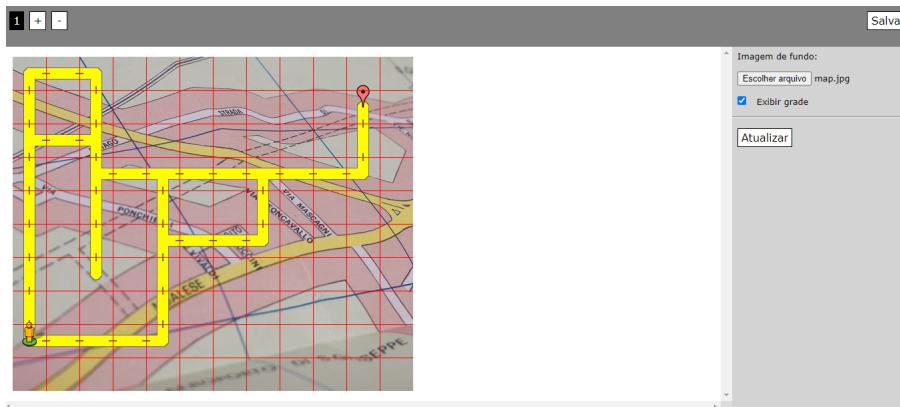
<sup>18</sup> <http://kodetu.org>

Figura 3 – Editor da plataforma Kodetu.



uma ferramenta de fácil manipulação e personalização. Ao criar um novo jogo, o usuário pode escolher uma imagem de fundo personalizada e adicionar ou remover níveis. Conforme Bastos (2019) explica, com base na Figura 4, o editor de labirintos possui uma grade que auxilia o usuário na criação dos mesmos. Ao realizar uma sequência de cliques nos espaços da grade, a aplicação alterna entre um espaço sem caminho, um espaço com caminho, um ponto de partida e um ponto de chegada. Desta maneira, fica à critério do usuário criar uma estratégia para preencher estes espaços, criando um labirinto com as características necessárias para explorar os conceitos de programação desejados.

Figura 4 – Interface do gerador de jogos proposto por Bastos.



### 3 VISÃO GERAL DO PROJETO

O presente trabalho propõe uma plataforma web de criação de jogos personalizados com labirintos, voltada para Introdução à Lógica de Programação, utilizando linguagem baseada em blocos.

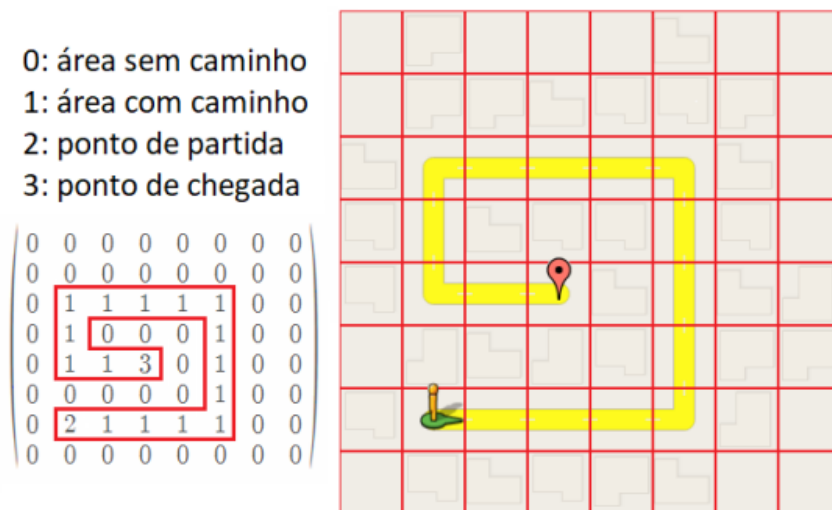
Usuários poderão acessar o sistema por meio de qualquer navegador de Internet. Em seu primeiro acesso, serão direcionados a página inicial da plataforma, onde será possível visualizar os jogos criados pela comunidade, buscar por um jogo específico e posteriormente executar o jogo. Caso deseje criar um novo jogo, será necessário criar uma

conta na plataforma para ter permissão de acesso às páginas de criação e gerenciamento dos jogos.

No processo de criação do jogo, além da elaboração do caminho e dos níveis, será solicitado que o usuário nomeie o jogo e selecione uma imagem de fundo para o labirinto.

Com base no estudo do gerador desenvolvido por (BASTOS, 2019), foi possível identificar que os labirintos do jogo são armazenados em uma lista de matrizes numéricas, sendo que cada matriz equivale a um nível que compõem o jogo. Na Figura 5 é apresentada a forma que a área sem caminho, ponto de partida, ponto de chegada e área com caminho são armazenados na matriz. O ponto de partida é marcado como (2), já o ponto de chegada é marcado como (3), caminho vazio e onde há caminhos são indicados como (0) e (1), respectivamente.

Figura 5 – Interface do gerador de jogos.



Fonte: Adaptado de Bastos (2019)

Diferentemente da versão anterior, onde é gerado um novo arquivo HTML com as informações necessárias para a execução do jogo de labirinto personalizado, a nova versão realizará o armazenamento de tais dados em um banco de dados, que será descrito nas próximas seções. Com isso, a manipulação dos jogos para posterior consumo na aplicação web será feita por meio de requisições HTTP ao servidor do sistema.

#### 4 DEFINIÇÃO DOS REQUISITOS

Os requisitos de um sistema consistem nas descrições do que a aplicação deve fazer, incluindo os serviços que ela fornece e as restrições de sua operação. Esses requisitos refletem as necessidades que os usuários do sistema possuem, como por exemplo, controlar um dispositivo, fazer um pedido ou encontrar informações (SOMMERVILLE, 2011).

Casos de uso são uma técnica de descoberta de requisitos. Um conjunto de casos de uso representam todas as interações possíveis que deverão ser realizadas pelo sistema. Os atores envolvidos em uma iteração são representados como bonecos de palito. Cada caso de uso é representado por uma elipse nomeada e o relacionamento entre eles é representado com uma linha.

A Tabela 1 mostra os atores com iterações na aplicação proposta.

Ator	Descrição
Não cadastrado	Usuário que não criou uma conta na plataforma. Possui acesso a visualização dos jogos da comunidade e a possibilidade de executar um jogo.
Cadastrado	Usuário cadastrado na plataforma e que está logado. Possui acesso total para criar e gerenciar seus jogos.

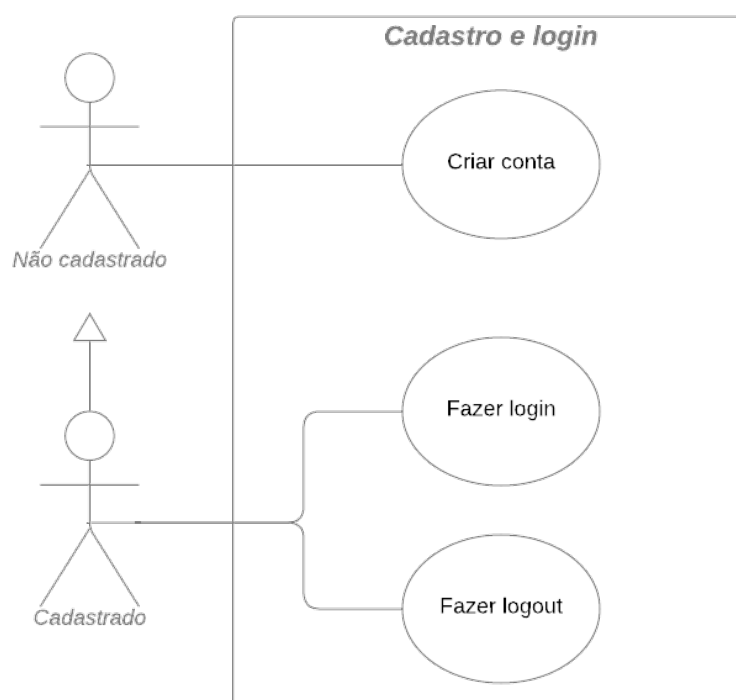
Tabela 1 – Atores do sistema

As subseções a seguir apresentam o caso de uso envolvendo acesso do usuário e o caso de uso de interação com os jogos e seus respectivos protótipos de telas.

#### 4.1 Casos de uso do acesso do usuário

A Figura 6 mostra as interações referentes à criação de conta (Figura 7), login (Figura 8) e logout dos usuários.

Figura 6 – Diagrama de casos de uso de login e cadastro.



Fonte: Produção do próprio autor

#### 4.2 Casos de uso de interação com os jogos

Os casos de uso descritos na Figura 9 apresentam as principais interações dos usuários com a plataforma. A partir deles, é possível visualizar os jogos, seus detalhes e jogá-los. Se o usuário estiver autenticado, terá acesso a criação e gerenciamento dos jogos.

Figura 7 – Protótipo da tela de criação de conta.

My BLOCKLY Maze

Nome ou código do jogo... **Pesquisar** Home Entrar **Cadastrar** Sobre

**Cadastre-se para criar um jogo**  
Crie seu usuário

**Nome:**  
Nome do usuário

**E-mail:**  
E-mail do usuário

**Senha:**  
Insira sua senha

**Confirmação de senha:**  
Confirme a sua senha

**Cadastrar**

Fonte: Produção do próprio autor

Figura 8 – Protótipo da tela de fazer login.

My BLOCKLY Maze

Nome ou código do jogo... **Pesquisar** Home **Entrar** Cadastrar Sobre

**Entrar**

Faça o login para poder utilizar o sistema

**E-mail:**  
E-mail do usuário

**Senha:**  
Insira sua senha

**Entrar**

Fonte: Produção do próprio autor

#### 4.2.1 Visualizar jogos da comunidade

Ao acessar a aplicação o usuário terá acesso a página principal do sistema (Figura 10). Nesta página, também são apresentados os jogos criados recentemente na plataforma e o campo de busca por um jogo, este localizado na barra de navegação e visível em todas as páginas da aplicação.

#### 4.2.2 Visualizar detalhes do jogo

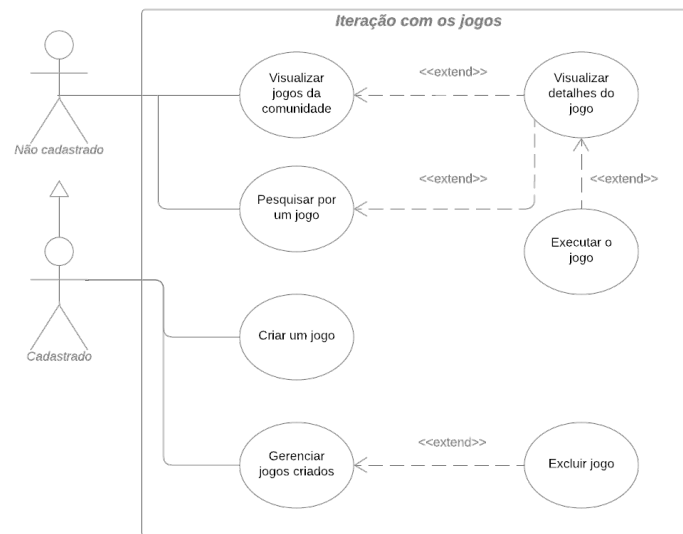
Ao clicar no botão “Detalhes” em um dos jogos da página inicial (Figura 10), o usuário é direcionado para a página de detalhes do jogo (Figura 11). Nela constam mais informações do jogo como nome, código, imagem de fundo, nome do criador, quantidade de níveis e de execuções. Também constam botões de compartilhamento e direcionamento para a execução do jogo.

O botão de compartilhamento é um fator importante da plataforma, uma vez que ela busca facilitar este processo. Ao clicar neste botão, é realizada a cópia da URL da página, assim fica a cargo do usuário fazer o compartilhamento do modo que achar conveniente.

As estatísticas presentes auxiliam os usuários na tomada de decisão em relação



Figura 9 – Diagrama de casos de uso de interação com os jogos



Fonte: Produção do próprio autor

Figura 10 – Protótipo da tela de visualização dos jogos da comunidade.



Fonte: Produção do próprio autor

à escolha do jogo. Com elas, é possível verificar o engajamento da comunidade de acordo com a quantidade de execuções e estimar a duração e complexidade com base na quantidade de níveis.

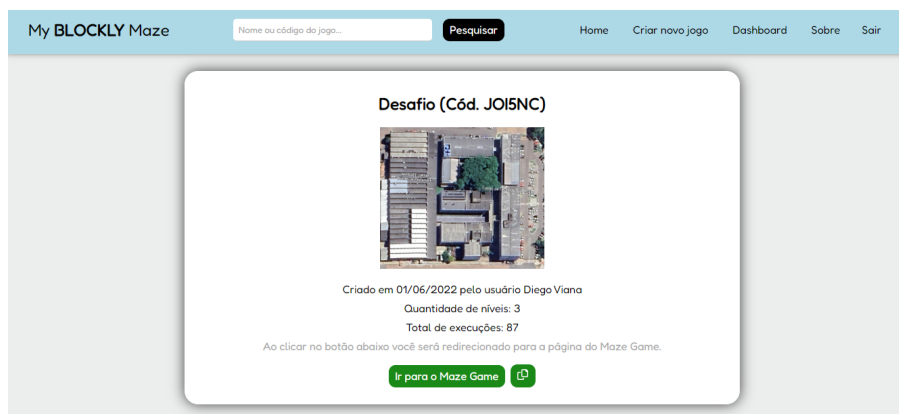
### 4.2.3 Pesquisar por um jogo

Ao digitar o nome ou código de algum jogo no campo de busca, o usuário é direcionado para a página de resultado da pesquisa (Figura 12).

Optou-se por desenvolver mais de um método de pesquisa, a fim de evitar o possível retorno de jogos com o mesmo nome ao realizar pesquisas. Uma abordagem amplamente utilizada em aplicações web é a definição de um código de identificação único, com isso, o mecanismo de busca se torna mais eficaz.

O sistema não realiza distinção entre letras maiúsculas e minúsculas, ou seja,

Figura 11 – Protótipo da tela de visualização de detalhes dos jogos.



Fonte: Produção do próprio autor

Figura 12 – Protótipo da tela de pesquisa por um jogo.



Fonte: Produção do próprio autor

independentemente da maneira que o usuário digitar a aplicação vai retornar com o jogo ou jogos com o nome ou trecho pesquisado.

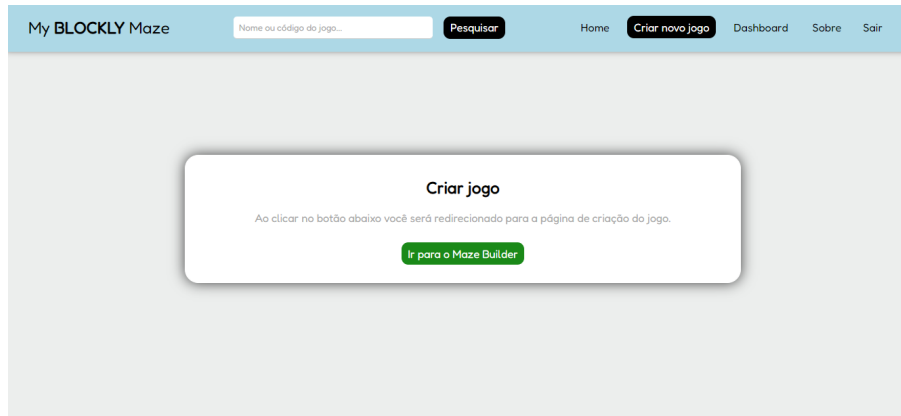
#### 4.2.4 Criar um jogo

Nesta página, o usuário é informado que será redirecionado para o Maze Builder (Figura 13), nome dado à nova versão do gerador de jogos personalizados com labirintos apresentado na subseção 5.4.

#### 4.2.5 Gerenciar jogos criados

O usuário autenticado tem permissão para acessar a página de gerenciamento dos jogos criados por ele na plataforma. Os jogos são apresentados em ordem decrescente em relação à data de criação do jogo no sistema. Cada elemento possui duas ações com botões nomeados como "Detalhes" e "Excluir". Ao clicar no botão "Detalhes", o usuário será direcionado para a página descrita na subseção 4.2.2. Ao clicar no botão "Excluir" o sistema fará a exclusão do jogo na base de dados. Após a conclusão da ação, a listagem

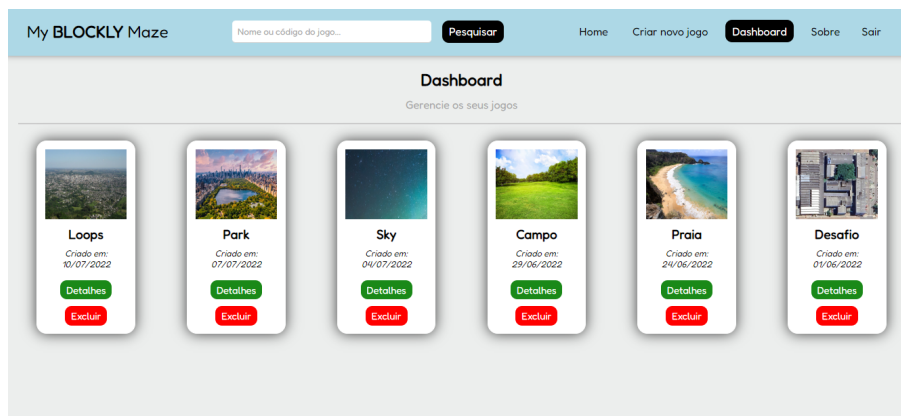
Figura 13 – Protótipo da tela de redirecionamento para criação do jogo.



Fonte: Produção do próprio autor

dos jogos é atualizada mostrando apenas os jogos que o usuário ainda possui. A Figura 14 apresenta a página de gerenciamento de jogos.

Figura 14 – Protótipo da tela de gerenciamento dos jogos.



Fonte: Produção do próprio autor

## 5 DESENVOLVIMENTO DA PLATAFORMA

A arquitetura da plataforma (Figura 15) é constituída por uma aplicação back-end, uma aplicação front-end e um banco de dados.

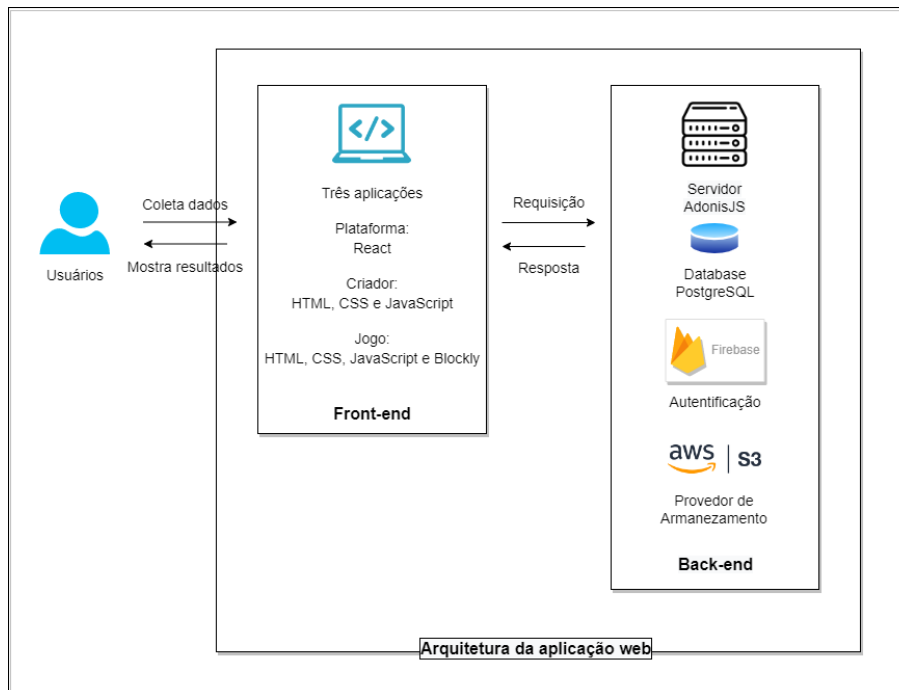
### 5.1 Banco de dados

A base de dados da aplicação foi definida para atender os requisitos definidos na seção anterior. A Figura 16 apresenta o Diagrama Entidade-Relacionamento desenvolvido para o sistema.

A seguir é feita a descrição de cada tabela:

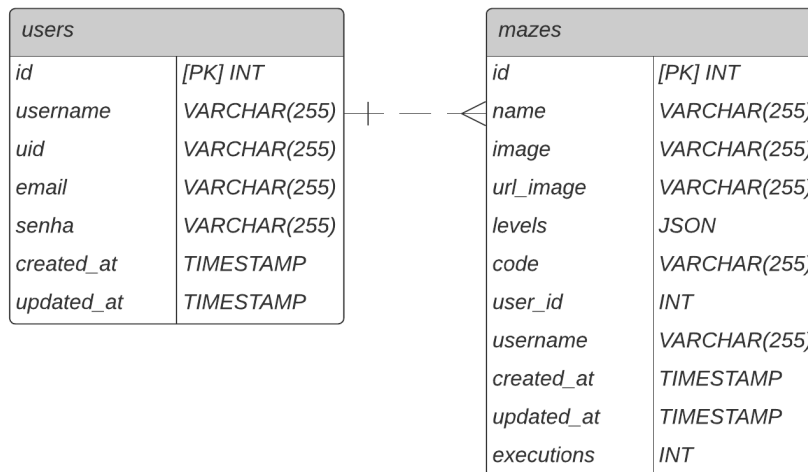
- `users`: armazena os dados dos usuários.
- `mazes`: armazena os dados dos jogos. Um jogo está vinculado a um usuário.

Figura 15 – Arquitetura da plataforma.



Fonte: Produção do próprio autor

Figura 16 – Diagrama Entidade-Relacionamento.



Fonte: Produção do próprio autor

Optou-se por usar na aplicação o SGBD PostgreSQL<sup>19</sup>, um sistema de banco de dados relacional avançado e de código aberto. O PostgreSQL suporta consultas SQL.

Utilizou-se o ElephantSQL<sup>20</sup>, um serviço que oferece hospedagem online de uma base de dados PostgreSQL. Todo o processo de configuração do serviço é simples, uma vez

<sup>19</sup> <https://www.postgresql.org/>

<sup>20</sup> <https://www.elephantsql.com/>

que ele automatiza todas as partes da configuração e execução dos clusters PostgreSQL, fazendo com que a base de dados já esteja pronta para uso em menos de dois minutos, como o próprio site informa.

## 5.2 Back-end

Back-end é o lado do servidor de uma aplicação web. Ele armazena e organiza os dados que são repassados pelo front-end e também garante o bom funcionamento da aplicação no lado do cliente retornado com a resposta do que lhe foi passado. Os usuários interagem indiretamente com o back-end, pois o contato acontece por meio dos componentes da interface gráfica do front-end. A comunicação entre o front-end, back-end e base de dados é realizada por meio de rotas que seguem o protocolo HTTP e utilizam o padrão de arquitetura REST, retornando ao cliente um objeto no formato de JSON.

Para desenvolvimento do back-end, foi utilizado AdonisJS<sup>21</sup>, um framework que possui uma estrutura robusta para estruturação da API. Outro ponto positivo em seu uso é o suporte nativo ao Lucid ORM. Um ORM que fornece suporte para bancos de dados SQL, com mecanismos de consultas sem o uso de consultas SQL e um sistema de migrações das tabelas criadas no framework para a base de dados.

Nesta etapa, também foi feita a ativação do CORS (Cross Origin Resource Sharing)<sup>22</sup>, um protocolo que permite que os servidores recebam solicitações de diferentes domínios.

No processo de upload das imagens de fundo dos jogos, foi utilizado o serviço de armazenamento na nuvem do Amazon S3<sup>23</sup>. Para lidar com isso, o AdonisJS possui uma abstração nativa chamada Drive<sup>24</sup>, com o objetivo de fornecer métodos que auxiliem na manipulação de arquivos em diferentes provedores de armazenamento.

Em relação às rotas de requisição e respostas, foram implementadas aquelas com funcionalidade de criação, listagem total, listagem de um item e exclusão para as tabelas `users` e `mazes`. Nesta última, foi desenvolvida a funcionalidade de atualização, visando a necessidade de atualizar a quantidade de execuções a cada nova execução de um jogo.

Para controlar o acesso à aplicação por meio de autenticação dos usuários com email e senha cadastrados no sistema, foram utilizados os serviços disponibilizados pelo Firebase Authentication<sup>25</sup>.

## 5.3 Front-end

Front-End é a parte do sistema com a qual o usuário interage diretamente, também conhecido como “lado do cliente” da aplicação. Isso inclui tudo o que os usuários experimentam diretamente, como, páginas, cores, estilos de texto, imagens, gráficos, tabelas, botões e menu de navegação. É responsável por receber as informações inseridas pelos usuários, processá-las, repassá-las ao back-end, se necessário, e exibir os resultados de maneira organizada na interface. HTML<sup>26</sup>, CSS<sup>27</sup> e JavaScript são as linguagens usadas

---

<sup>21</sup> <https://adonisjs.com/>

<sup>22</sup> <https://docs.adonisjs.com/guides/security/cors>

<sup>23</sup> <https://aws.amazon.com/pt/s3/>

<sup>24</sup> <https://docs.adonisjs.com/guides/drive>

<sup>25</sup> <https://firebase.google.com/>

<sup>26</sup> <https://developer.mozilla.org/pt-BR/docs/web/HTML>

<sup>27</sup> <https://developer.mozilla.org/pt-BR/docs/web/CSS>

no desenvolvimento front-end.

Para desenvolvimento do front-end foi utilizado React.js<sup>28</sup>, uma biblioteca de código aberto, baseada em componentes, eficiente e flexível, responsável apenas pela camada de visualização da aplicação. É mantida pelo Facebook<sup>29</sup>.

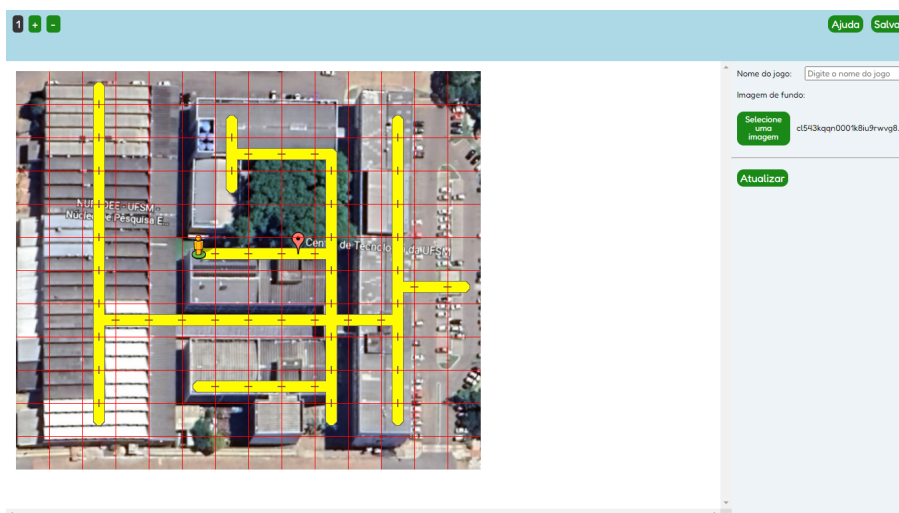
A comunicação do front-end com o back-end foi implementada utilizando a API Fetch<sup>30</sup>, uma interface que permite fazer requisições HTTP assíncronas para servidores com o uso de uma Promise.

#### 5.4 Implementação da criação de jogos com labirintos

Nesta etapa, tomou-se como ponto de partida o gerador que foi explicado na subseção 2.3. Após o estudo do código fonte foi observado que ele foi desenvolvido somente utilizando HTML, CSS e JavaScript. Tendo isso em vista, decidiu-se manter o desenvolvimento da versão atualizada com essas mesmas linguagens, com o objetivo de eliminar possíveis problemas que surgiriam ao tentar integrar o gerador ao React.JS.

A interface passou por algumas adaptações, como alteração nas cores e a adição de um campo para o inserção do nome do jogo. A Figura 17 mostra a interface.

Figura 17 – Protótipo da tela de criação dos jogos.



Fonte: Produção do próprio autor

Durante o desenvolvimento, foi observado que a versão anterior carecia de algumas funcionalidades de auxílio aos usuários e checagem em relação aos labirintos construídos. Outro ponto analisado foram as realizações de notificações de sistema por meio de alertas nativos do navegador web.

A imagem de fundo poderia ser considerada um ponto crítico da ferramenta, pois o sistema estava limitado a aceitar imagens até uma determinada dimensão. Este cenário tinha potencial para desmotivar o usuário durante o processo de criação, uma vez que ele teria que encontrar uma imagem que fosse compatível com o sistema.

<sup>28</sup> <https://reactjs.org/>

<sup>29</sup> <https://www.facebook.com/>

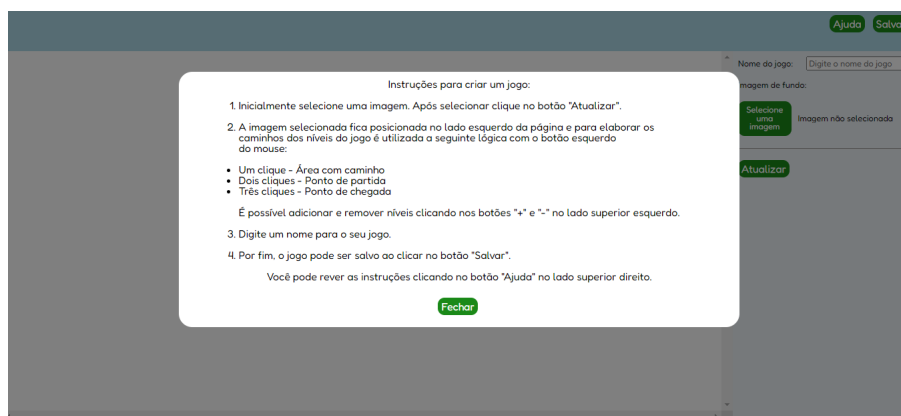
<sup>30</sup> <https://developer.mozilla.org/pt-BR/docs/web/API/fetch>

Buscando fortalecer a ferramenta e melhorar a interação do usuário, decidiu-se implementar as funcionalidades não existentes descritas anteriormente, além do salvamento das informações do jogo personalizado com labirintos no banco de dados.

O processo de redimensionamento da imagem foi feito por meio da tag HTML canvas<sup>31</sup>. Essa tag é usada para desenhar vários tipos de elementos, entre eles imagens. Na manipulação da tag via JavaScript é possível determinar a dimensão que o elemento desenhado deve ter. É importante ressaltar que existe a possibilidade da imagem ficar distorcida ao final deste processo.

As instruções para criar um jogo são apresentadas ao usuário através de um modal (Figura 18), assim que ele realiza o acesso a esta página. Nele estão presentes instruções de como alternar entre um ponto de partida, ponto de chegada e área com caminho, entre outras. O modal pode ser reaberto a qualquer momento ao clicar no botão “Ajuda”.

Figura 18 – Modal com instruções para criação dos jogos.



Fonte: Produção do próprio autor

O sistema de notificações de sistema foi alterado para interagir com os usuários usando modais. Entre eles estão o pedido de confirmação da exclusão do último nível criado (Figura 19) e o aviso de impossibilidade de excluir o primeiro nível do labirinto.

No concernente às checagens, foram elaboradas verificações sobre o preenchimento do campo de nome do jogo e se a imagem de fundo foi selecionada. Outra verificação está relacionada ao labirinto elaborado ter mais de um ponto de partida ou de fim ou não possuir alguma área com caminho. Caso o sistema detecte uma destas situações, a aplicação retornará o seguinte modal da Figura 20).

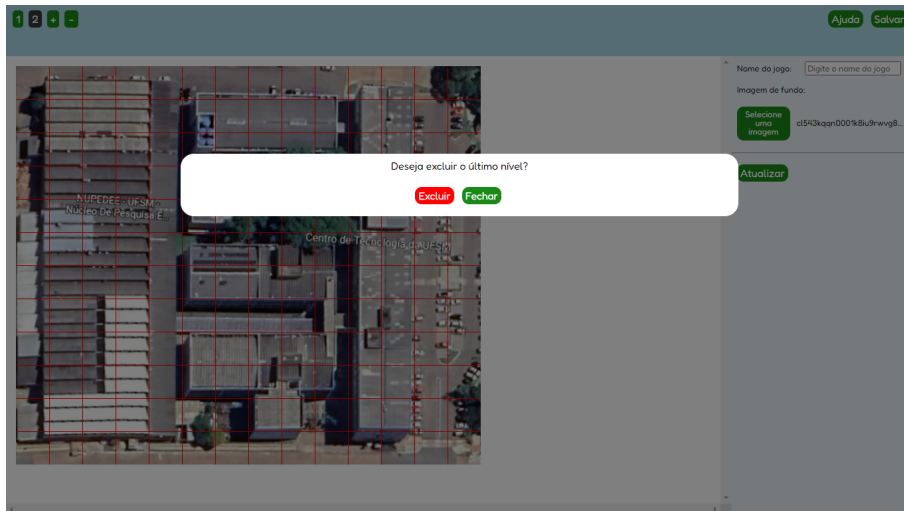
Ao clicar no botão “Salvar”, visível na Figura 17, é feita a requisição HTTP utilizando o método POST para inserir na tabela `mazes` as informações referentes ao id do jogo, nome do jogo, id do criador, nome do criador, código do jogo (criado randomicamente pelo back-end), arquivo de imagem, data de criação e os níveis do jogo criado em formato JSON. Após o usuário receber um aviso informando que o salvamento foi realizado com sucesso, o jogo ficará visível nas telas apresentadas na subseção 4.2.

## 5.5 Implementação da execução dos jogos

Para realizar a execução dos jogos personalizados, usou-se como base o jogo Maze do Blockly Games. Seu funcionamento padrão consiste em executar os níveis de

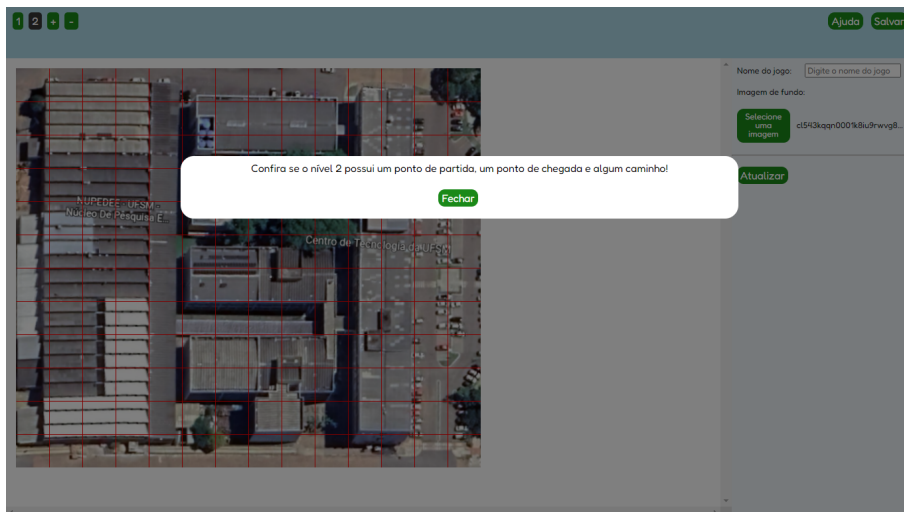
<sup>31</sup> [https://developer.mozilla.org/pt-BR/docs/Web/API/Canvas\\_API](https://developer.mozilla.org/pt-BR/docs/Web/API/Canvas_API)

Figura 19 – Modal de confirmação da exclusão do último nível criado.



Fonte: Produção do próprio autor

Figura 20 – Modal de notificação de erro.



Fonte: Produção do próprio autor

somente um jogo que foi definido estaticamente. Também foi decidido usar a sua versão em português.

No desenvolvimento desta etapa, as maiores alterações foram no código-fonte, onde foi realizada a adaptação para que se tornasse possível a execução do jogo personalizado. No aspecto visual, foram feitos alguns ajustes na estilização e a remoção de alguns itens e botões da página.

Relativo a adaptação, foi adotada a estratégia de gerar uma URL dinâmica contendo o endereço base da nova versão do Maze Game, os níveis e o endereço da imagem de fundo armazenada no Amazon S3, com isso, é feita a manipulação da nova URL via JavaScript para que a aplicação consiga consumir tais informações no processo de reprodução do jogo. A Figura 21 mostra a execução de um jogo personalizado.

Por conta do feedback, recebido durante o teste de usabilidade descrito na próxima



Figura 21 – Execução de um jogo personalizado.



Fonte: Produção do próprio autor

seção, implementou-se também um modal com um vídeo de instruções. O vídeo foi desenvolvido no formato de tutorial, mostrando a mecânica de execução do jogo e as funções dos blocos, este último realizado através de textos adicionados por meio de edição. O usuário pode visualizar o modal clicando no botão “Ajuda” presente na página de execução do jogo (Figura 21). A Figura 22 mostra o modal relatado.

Figura 22 – Modal com instruções para jogar.



Fonte: Produção do próprio autor

## 6 TESTES COM USUÁRIOS

Com o objetivo de atestar a usabilidade da plataforma e identificar se a mesma facilita o compartilhamento dos jogos personalizados com labirintos, aplicou-se um teste de usabilidade.

A escala de usabilidade do sistema (SUS (BROOKE, 1995)) é um instrumento comumente utilizado em testes de usabilidade de produtos comerciais, considerado um método rápido de avaliação e de baixo custo a níveis industriais.

O SUS consiste em uma escala simples de dez perguntas que oferecem uma visão global das avaliações relacionadas à usabilidade. A ISO 9241-11 define que a medida da usabilidade deve abranger:

- Eficácia (capacidade de conclusão das tarefas pelos usuários do sistema).
- Eficiência (nível de recursos consumidos nas execuções das tarefas).
- Satisfação (reações subjetivas dos usuários ao usar o sistema).

Cada pergunta pode ser respondida através de uma escala que varia de 1 a 5, onde 5 indica “concordo totalmente” e 1 indica “discordo totalmente”. Essas questões são geralmente respondidas depois que os usuários realizam um conjunto de tarefas no sistema a ser avaliado.

A pontuação SUS é dada pelo seguinte cálculo:

- Para as perguntas 1,3,5,7 e 9 é feita a subtração da pontuação atribuída - 1.
- Para as perguntas 2,4,6,8 e 10 é feita a subtração da pontuação atribuída - 5.
- Soma-se os resultados das subtrações e multiplica-se por 2,5.

O resultado do cálculo será um valor entre 0 e 100, onde o sistema é classificado da seguinte maneira (BROOKE, 2013):

- Melhor imaginável (+85)
- Excelente (+73)
- Bom (+52)
- Ok (+39)
- Pobre (+25)
- Pior imaginável (+0)

As próximas subseções relatam a aplicação do teste e os resultados obtidos.

## 6.1 Aplicação do teste de usabilidade

Os testes foram realizados com 8 pessoas de diferentes grupos: 1 professor de graduação de computação, 3 com conhecimento aprofundado em programação, 2 iniciantes em programação e 2 sem conhecimento com programação. Somente o teste com o professor de graduação foi realizado presencialmente.

Os participantes do teste remoto receberam uma breve explicação sobre o funcionamento da plataforma, criação dos jogos e de como jogá-los. Ao término, os usuários realizaram as seguintes tarefas presentes no roteiro do teste:

- Crie uma conta na plataforma.
- Crie um novo jogo.

- Verifique se o jogo criado está na dashboard e na home da plataforma.
- Acesse os detalhes do jogo.
- Teste o jogo.
- Compartilhe o jogo. (Botão de cópia da URL na página de detalhes do jogo)

Após realizar as tarefas, os usuários responderam um formulário com 10 perguntas disponibilizado no Google Forms<sup>32</sup>.

A Tabela 2 mostra as perguntas e a quantidade de marcações de cada alternativa.

	1 pt	2 pts	3 pts	4 pts	5 pts
Eu achei a aplicação fácil de usar.			2	3	3
Eu achei a aplicação atrapalhada de usar.	4	2	1	1	
Eu acho que precisaria de ajuda para utilizar a plataforma.	2		1	3	2
Eu achei fácil o processo de criação do jogo.				1	7
Eu acho que o sistema facilita o compartilhamento de jogos.			1		7
Eu acho que usaria a plataforma com frequência.		1	2	5	
Eu acredito que outras pessoas teriam interesse em conhecer a plataforma.				3	5
Eu acho que a plataforma é um bom lugar para ter o primeiro contato com lógica de programação.				3	5
Acredito que os jogos podem ajudar a exercitar a lógica de programação.					8
Eu acho que a plataforma não apresenta benefícios relacionados a programação.	7	1			

Tabela 2 – Perguntas do formulário

Com base nas respostas obtidas, foi realizado o cálculo de SUS, onde o somatório da pontuação de cada usuário é 522,5. Para se ter o valor médio de SUS é necessário dividir o somatório por 8 (quantidade de pessoas que participaram dos testes), sendo assim, a aplicação é classificada como boa, pois o SUS médio resultante é 65,31 pontos.

Durante o teste presencial, iniciado antes do usuário ser instruído sobre o funcionamento da aplicação, foi observado que a maior dificuldade estava relacionada a entender o objetivo do jogo e o que cada bloco representa. Após uma breve explicação verbal do instrutor, o usuário testador conseguiu concluir o desafio proposto no jogo.

Analisando as respostas do formulário, pode-se afirmar que o processo de criação do jogo é classificado como fácil. Acredita-se que o modal com instruções tenha sido um dos fatores que favoreceram esta classificação. Muitos usuários indicaram a necessidade de ajuda para utilizar a plataforma, circunstância causada por ser a primeira experiência de uso de um ambiente com blocos.

<sup>32</sup> <https://workspace.google.com/intl/pt-BR/products/forms/>

## 6.2 Resultados

A partir de algumas observações realizadas pelos usuários, foi adicionada na página de detalhes do jogo a quantidade de níveis que ele possui. Também foi adotado o uso da versão em português do jogo e o desenvolvimento de um modal com instruções sobre seu objetivo e funções dos blocos.

Uma sugestão recebida foi a criação de um novo bloco de repetição sem estar relacionado com o ponto de chegada. Segundo o usuário, seria interessante que o novo bloco seguisse a lógica de “se caminho à frente repita”, com o campo “à frente” podendo ser alterado entre direita e esquerda.

Um dos usuários relatou problemas para executar a plataforma na versão mobile. Cabe destacar que a plataforma no atual estágio atende somente desktops, pois, mesmo sendo possível visualizar as telas e arrastar os blocos em um dispositivo móvel, a área de programação e o labirinto ocupam um espaço considerável.

Todos os usuários conseguiram concluir as tarefas propostas no roteiro de teste. Os jogos criados pelos testadores estão disponíveis na plataforma para consulta e execução. Ao analisar o resultado das criações, é possível notar que grande parte dos usuários optou por criar desafios com poucos níveis, com exceção de um participante que elaborou oito níveis. As imagens de fundo também apresentam algumas diferenças entre os usuários. Entre elas está a escolha de uma imagem que se integra perfeitamente com o caminho do labirinto personalizado, por exemplo, uma vista aérea de algum local.

O nível de complexidade seguiu a tendência de demandar um conjunto de combinações de blocos condicionais para que o personagem atingisse o objetivo final. Em determinados níveis, é necessário mais esforço mental durante o processo de solução do desafio personalizado.

Pontos positivos relacionados à programação apontados pelos usuários através das respostas no formulário são que a plataforma é um bom local para o primeiro contato com lógica de programação e os jogos podem ajudar a exercitar a lógica de programação.

O sistema de compartilhamento foi um dos quesitos com melhor avaliação, logo a aplicação atende o objetivo estabelecido anteriormente.

## 7 CONCLUSÃO

O presente trabalho teve como objetivo implementar o My Blockly Maze, uma plataforma online para criação de jogos personalizados com labirintos para a Introdução a Lógica de Programação. Estes labirintos foram desenvolvidos de maneira semelhante ao jogo Maze presente na coletânea de jogos educacionais voltados à aprendizagem de programação, Blockly Games. O desafio proposto em um tal jogo é instigar o jogador a movimentar o personagem até o ponto final do labirinto. Esta movimentação é realizada utilizando programação baseada em blocos visuais, que representam comandos existentes em linguagens de programação textuais. No processo de criação dos jogos de labirinto, foi realizada a expansão de uma ferramenta apresentada em um trabalho anterior. A plataforma atua na integralização entre a criação, execução e compartilhamento dos jogos personalizados.

As principais funcionalidades previstas no projeto foram implementadas, assim ao acessar a plataforma o usuário pode visualizar os jogos elaborados pela comunidade, acessar os detalhes dos jogos, executá-los e compartilhá-los. Outra possibilidade, caso

o usuário possua uma conta na plataforma ou crie uma, é a criação de um novo jogo e o acesso ao painel de gerenciamento de jogos criados.

A plataforma My Blockly Maze, com todas as funcionalidades descritas neste documento, está disponível em [<https://myblocklymaze.vercel.app/>](https://myblocklymaze.vercel.app/).

Para trabalhos futuros, espera-se desenvolver mais funcionalidades na plataforma, como a inclusão de mais estatísticas, além das já presentes (quantidade de níveis e de execuções dos jogos). Também, espera-se implementar o restante das sugestões realizadas pelos usuários durante a fase de testes, assim como a versão mobile.

Destaca-se que a elaboração de um sistema de ranqueamento de jogadores e criadores poderia motivar a participação dos usuários. Outra possibilidade de melhoria está na adição de uma seção de comentários na página de detalhes do jogo, com isso o jogador poderia interagir com outros usuários sobre a solução adotada na resolução do labirinto.

De modo geral, a aplicação foi desenvolvida buscando disponibilizar aos usuários um ambiente que possibilite a apresentação dos conceitos introdutórios de Lógica de Programação e o desenvolvimento do Pensamento Computacional. Paralelamente a isso, também buscou-se desenvolver um sistema que facilite o compartilhamento dos novos jogos com mais usuários. Devido a isso, acredita-se que a plataforma tem potencial para ser uma ferramenta de auxílio a iniciantes em programação e público em geral.

## REFERÊNCIAS

ALRUBAYE, H.; LUDI, S.; MKAOUER, M. W. Comparison of block-based and hybrid-based environments in transferring programming skills to text-based environments. In: *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering*. USA: IBM Corp., 2019. (CASCON '19), p. 100–109. Citado na página 8.

BASTOS, G. P. *Gerador de jogos personalizados com labirintos para introdução à lógica de programação*. Trabalho de Conclusão de Curso (Ciência da Computação) — Universidade Federal de Santa Maria, Santa Maria, 2019. Disponível em: [<https://repositorio.ufsm.br/handle/1/22515>](https://repositorio.ufsm.br/handle/1/22515). Citado 5 vezes nas páginas 9, 10, 12, 13 e 14.

BROOKE, J. Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, v. 189, 11 1995. Citado na página 25.

BROOKE, J. Sus: a retrospective. *Journal of Usability Studies*, v. 8, p. 29–40, 01 2013. Citado na página 26.

COMBÉFIS, S.; BERESNEVICIUS, G.; DAGIENÉ, V. Learning programming through games and contests: Overview, characterisation and discussion. *Olympiads in Informatics*, v. 10, p. 39–60, 2016. Disponível em: [<https://doi.org/10.1145/1118178.1118215>](https://doi.org/10.1145/1118178.1118215). Citado na página 10.

DOUKAKIS, S. Exploring brain activity and transforming knowledge in visual and textual programming using neuroeducation approaches. *AIMS Neuroscience*, v. 6, p. 175–190, 09 2019. Citado na página 8.

EVRIPIDOU, S. et al. Educational robotics: Platforms, competitions and expected learning outcomes. *IEEE Access*, v. 8, p. 219534–219562, 2020. Citado na página 12.

GOMEZ, M. J.; MORESI, M.; BENOTTI, L. Text-based programming in elementary school: A comparative study of programming abilities in children with and without block-based experience. In: *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2019. (ITiCSE '19), p. 402–408. ISBN 9781450368957. Disponível em: <<https://doi.org/10.1145/3304221.3319734>>. Citado 2 vezes nas páginas 8 e 10.

GUARDA, G.; PINTO, S. Dimensões do pensamento computacional: conceitos, práticas e novas perspectivas. In: *Anais do XXXI Simpósio Brasileiro de Informática na Educação*. Porto Alegre, RS, Brasil: SBC, 2020. p. 1463–1472. Disponível em: <<https://sol.sbc.org.br/index.php/sbie/article/view/12902>>. Citado na página 8.

GUENAGA, M. et al. How do students develop computational thinking? assessing early programmers in a maze-based online game. *Computer Science Education*, Routledge, v. 31, n. 2, p. 259–289, 2021. Disponível em: <<https://doi.org/10.1080/08993408.2021.1903248>>. Citado na página 12.

KANDEMIR, C.; KALELIOGLU, F.; GULBAHAR, Y. Pedagogy of teaching introductory text-based programming in terms of computational thinking concepts and practices. *Computer Applications in Engineering Education*, v. 29, p. 29–45, 01 2021. Citado na página 10.

LADIAS, D.; KARVOUNIDIS, T.; LADIAS, A. Composition of programming structures into scenarios and their matching with the solo levels. In: *2020 5th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*. [S.l.: s.n.], 2020. p. 1–6. Citado na página 8.

LAHTINEN, E.; ALA-MUTKA, K.; JÄRVINEN, H.-M. A study of the difficulties of novice programmers. In: *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2005. (ITiCSE '05), p. 14–18. ISBN 1595930248. Disponível em: <<https://doi.org/10.1145/1067445.1067453>>. Citado na página 8.

MARTINS, R.; REIS, R.; MARQUES, A. Inserção da programação no ensino fundamental: Uma análise do jogo labirinto clássico da code.org através de um modelo de avaliação de jogos educacionais. In: *Anais do XXII Workshop de Informática na Escola*. SBC, 2016. p. 121–130. Disponível em: <<https://sol.sbc.org.br/index.php/wie/article/view/16408>>. Citado na página 12.

MEDEIROS, R. P.; RAMALHO, G. L.; FALCÃO, T. P. A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education*, v. 62, n. 2, p. 77–90, 2019. Citado na página 8.

MLADENOVIC, M.; BOLJAT, I.; ŽANKO, Comparing loops misconceptions in block-based and text-based programming languages at the k-12 level. *Education and Information Technologies*, v. 23, p. 1483–1500, 07 2018. Citado na página 10.

MOORS, L.; LUXTON-REILLY, A.; DENNY, P. Transitioning from block-based to text-based programming languages. In: *2018 International Conference on Learning and Teaching in Computing and Engineering (LaTICE)*. [S.l.: s.n.], 2018. p. 57–64. Citado na página 8.

PAPERT, S. *Mindstorms: Children, Computers, and Powerful Ideas*. USA: Basic Books, Inc., 1980. ISBN 0465046274. Citado na página 8.

PASTERNAK, E.; FENICHEL, R.; MARSHALL, A. N. Tips for creating a block language with blockly. In: *2017 IEEE Blocks and Beyond Workshop (B&B)*. [S.l.: s.n.], 2017. p. 21–24. Citado na página 11.

SANTANA, B.; CHAVEZ, C.; BITTENCOURT, R. Uma definição operacional para pensamento computacional. In: *Anais do Simpósio Brasileiro de Educação em Computação*. Porto Alegre, RS, Brasil: SBC, 2021. p. 93–103. Disponível em: <<https://sol.sbc.org.br/index.php/educomp/article/view/14475>>. Citado na página 8.

SILVA, F. M. d.; MENEGHETTI, R. C. G. Pensamento computacional e a relação com a base nacional comum curricular. In: *Congresso Brasileiro de Educação - CBE*. UNESP - Faculdade de Ciências, 2019. Disponível em: <<https://repositorio.usp.br/item/002989340>>. Citado na página 8.

SILVA, M.; FERREIRA, A. P. Linguagens visuais para o ensino de programação: uma revisão da literatura com foco em paradigmas de programação. In: *Anais do II Simpósio Brasileiro de Educação em Computação*. Porto Alegre, RS, Brasil: SBC, 2022. p. 18–28. Disponível em: <<https://sol.sbc.org.br/index.php/educomp/article/view/19195>>. Citado 2 vezes nas páginas 8 e 10.

SOMMERVILLE, I. *Software engineering*. v. 9, 2011. Citado na página 14.

TIKVA, C.; TAMBOURIS, E. Mapping computational thinking through programming in k-12 education: A conceptual model based on a systematic literature review. *Computers & Education*, v. 162, p. 104083, 2021. ISSN 0360-1315. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0360131520302815>>. Citado na página 8.

WEINTROP, D.; WILENSKY, U. Transitioning from introductory block-based and text-based environments to professional programming languages in high school computer science classrooms. *Computers & Education*, v. 142, p. 103646, 2019. ISSN 0360-1315. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S036013151930199X>>. Citado na página 8.

WING, J. M. Computational thinking. *Commun. ACM*, Association for Computing Machinery, New York, NY, USA, v. 49, n. 3, p. 33–35, mar 2006. ISSN 0001-0782. Disponível em: <<https://doi.org/10.1145/1118178.1118215>>. Citado na página 8.

XAVIER, E. et al. Pensamento computacional integrado à matemática na bncc: proposta para o primeiro ano do ensino fundamental. In: *Anais do XXXII Simpósio Brasileiro de Informática na Educação*. Porto Alegre, RS, Brasil: SBC, 2021. p. 989–1001. ISSN 0000-0000. Disponível em: <<https://sol.sbc.org.br/index.php/sbie/article/view/18124>>. Citado na página 8.