

Uma Ferramenta de Simulações Interativas para o Ensino de Conceitos de Programação para Crianças

Trabalho de Formatura Supervisionado
MAC0499
Dezembro 2024

Marília Takaguti Dicezare
Supervisora: Profa. Dra. Kelly Rosa Braghetto



IME

USP

Contexto e Motivação

- O ensino da computação para crianças e adolescentes promove competências fundamentais para o pensamento crítico e lógico desses jovens.
- Em 2022, o Conselho Nacional de Educação (CNE) homologou as Normas sobre Computação na Educação Básica - Complemento à Base Nacional Comum Curricular (BNCC).
- Ferramentas de apoio pedagógico são fundamentais no processo de aprendizagem.

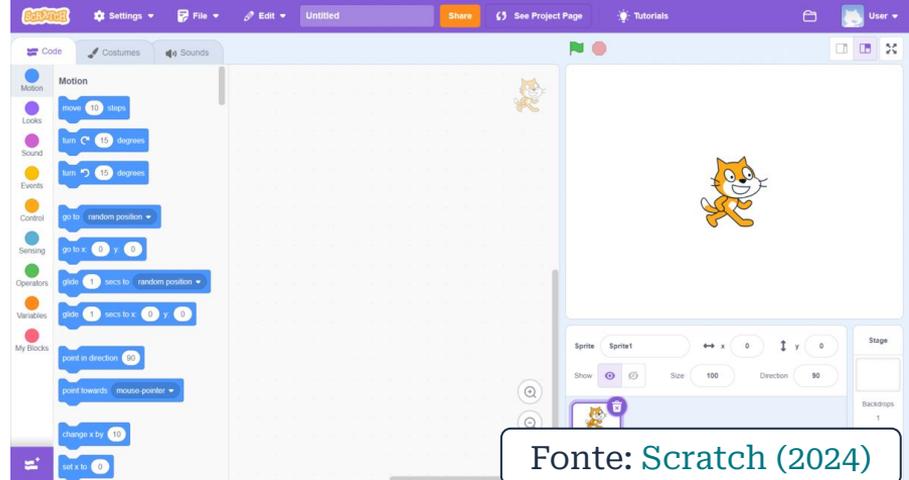


```

1 hero = game.spawnXY("goliath")
2 boss = game.spawnXY("cow", 46, 40)
3 coin = game.spawnXY("gold", 0, 0)
4 boss.missileType = coin
5
6 game.score = 0
7
8 def onCollect(event):
9     if event.item.type == "gold":
10        event.target.say("Gold!")
11        game.score += item.value
12        if game.score > 100:
13            game.showVictory()
14
15 hero.on("collect", onCollect)

```

Fonte: CodeCombat (2024)



Fonte: Scratch (2024)

LEGO MINDSTORMS EV3

CREATE & COMMAND YOUR OWN ROBOTS

EV3 PROGRAMMER APP
Programming at your fingertips - smart, fast, fun on your tablet!

PC/MAC SOFTWARE
Advanced programming - make your robots do anything you can imagine.

ROBOT COMMANDER APP
Remote control your robots with your smart device - instant action!

AND THERE'S MORE:
Visit LEGO.com/mindstorms for even more robot fun

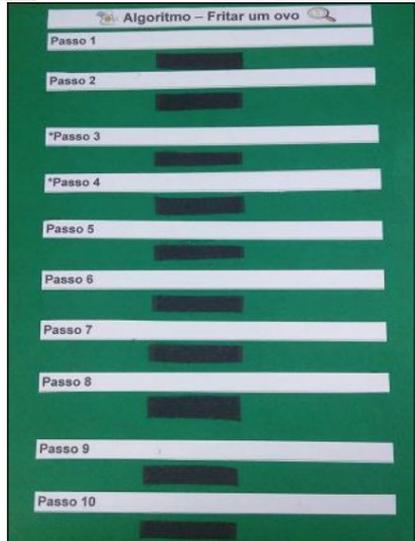
- Building instructions for 10 additional robots
- Learn-to-program tutorials
- Online community
- Inspiring videos
- Robot gallery
- Online game

PACKED WITH COOL FUNCTIONS

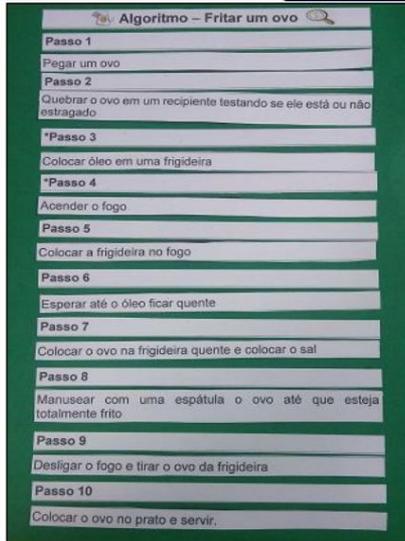
- Programmable brick with intuitive user interface and sound
- Infrared remote receiver - makes your robot really hear
- WiFi access via the USB port - connects your robot to the internet
- Mind Brick sensor - lets your robot see
- Colour Sensor - makes your robot recognise colours
- Remote Control - steers your robot from a distance
- Infrared Sensor - makes your robot see to track and find the remote control
- Touch Sensor - makes your robot respond to touch
- Motor ports - enable your robot to move in multiple ways and directions

LEGO.com/mindstorms

Fonte: Lego Mindstorms (2024)



Fonte: Crema et al. (2020)



Simulações Interativas

- As simulações facilitam a visualização de ideias abstratas, utilizando exemplos concretos para representá-las.
- O uso de simulações interativas na educação tem se mostrado eficaz em outras ciências, com a aplicação da ferramenta PhET (Khatri et al., 2013).

Equality Explorer: Two Variables

The simulation displays the inequality $4x + 5y > -2x - 2y + 3$. The current values are $x = 5$ and $y = -1$. The interface includes a 'Snapshots' button and a toolbar with algebra tiles and a lock icon.

Fonte: PhET (2024)

Objetivos

- Criação de um produto mínimo viável (**MVP**) de uma ferramenta de **simulações interativas** para o ensino de conceitos de lógica de programação para crianças do ensino fundamental.
 - ◆ Investigar como representar os conceitos visualmente.
- Aplicação do MVP em sala de aula e avaliação da sua **usabilidade**.

Design Science Research (DSR) - Wieringa (2014)

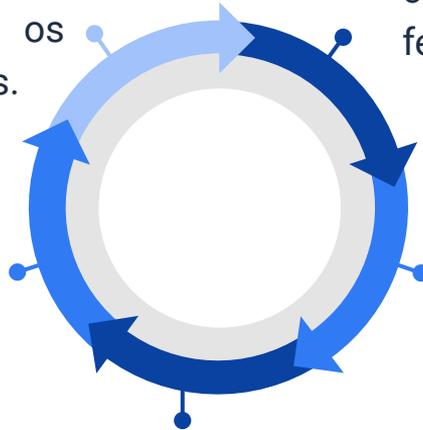
5. Avaliação do MVP

Formulário de usabilidade SUS (*System Usability Scale*).

Atividade de aprendizado sobre os conceitos de programação abordados.

4. Implementação do MVP

Implementação da simulação interativa do protótipo escolhido.



1. Investigação do problema

Pesquisa sobre o ensino de computação nas escolas brasileiras e ferramentas pedagógicas existentes.

2. Design do artefato

Projeto de protótipos para tratar o problema.

3. Validação do artefato

Feedback de profissionais sobre a viabilidade e usabilidade dos protótipos.

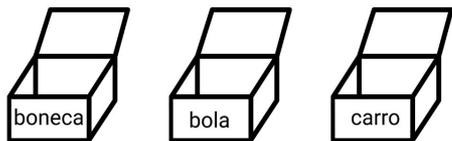
Protótipos

Guardando os brinquedos

Ajuda

BrinquedosParaGuardar

3



Pseudocódigo

leia (brinquedosParaGuardar)

enquanto brinquedosParaGuardar != 0 faça

 leia (brinquedoEscolhido)
 leia (caixaEscolhida)

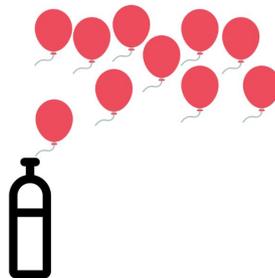
 se brinquedoEscolhido =
 caixaEscolhida então
 brinquedosParaGuardar <-
 brinquedosParaGuardar - 1

 senão
 escreva ("O brinquedo
 escolhido não pertence a caixa
 escolhida :(")

Planejando a festa

Ajuda

itemEscolhido



totalPronto

+ 10



Pseudocódigo

// variável
total <- 0

leia (itemEscolhido)

se itemEscolhido = "bolo" então
 total <- 1

senão

 se itemEscolhido = "balão" então
 total <- 10
 senão
 total <- 20

enquanto totalPronto < total faça
 totalPronto <- totalPronto + 1

MVP

(<https://mariliatd.github.io/logic-sims-mvp/>)

Planejando a Festa

Conceitos de programação

Logic Sims é uma ferramenta educacional para o ensino de conceitos de lógica de programação, baseada em simulações.

A simulação *Planejando a Festa* apresenta elementos visuais que simulam um pseudocódigo, e permite interações com ambos.

Você pode explorar a simulação livremente, observando com atenção as mudanças que as interações causam, para aprender os conceitos de programação simulados.

itensProntos

0

Pseudocódigo

```
itensProntos ← 0
enquanto itensProntos < 3 faça
  quantidade ← 0
  leia (itemEscolhido )
  se itemEscolhido = "bolo" então
    quantidade ← 1
    escreva ("Prepare 1 bolo!")
  senão
    se itemEscolhido = "balão" então
      quantidade ← 10
    senão
      se itemEscolhido = "bolo" então
        quantidade ← 10
      senão
        se itemEscolhido = "balão" então
          quantidade ← 20
        escreva ("Prepare 20 doces!")
```

Planejando a Festa

Conceitos de programação

itemEscolhido



quantidade

1 10 20

itensProntos

0

quantidade

1

itemEscolhido

bolo

totalPronto

0



Pseudocódigo

```
itensProntos ← 0
enquanto itensProntos < 3 faça
  quantidade ← 0
  leia (itemEscolhido )
  se itemEscolhido = "bolo" então
    quantidade ← 1
    escreva ("Prepare 1 bolo!")
  senão
    se itemEscolhido = "balão" então
      quantidade ← 10
    escreva ("Prepare 10 balões!")
    senão
      quantidade ← 20
      escreva ("Prepare 20 doces!")
  totalPronto ← 0
  enquanto totalPronto < quantidade faça
    prepare 1 (itemEscolhido)
    totalPronto ← totalPronto + 1
  itensProntos ← itensProntos + 1
  escreva ("O item (itemEscolhido) está pronto!")
escreva ("A festa está pronta!!!!")
```

Planejando a Festa

Conceitos de programação

itemEscolhido



quantidade

1 10 20

itensProntos

1

quantidade

1

totalPronto

1

O item bolo está pronto!

FECHAR



Pseudocódigo

```
itensProntos ← 0
enquanto itensProntos < 3 faça
  quantidade ← 0
  leia (itemEscolhido )
  se itemEscolhido = "bolo" então
    quantidade ← 1
    escreva ("Prepare 1 bolo!")
  senão
    se itemEscolhido = "balão" então
      quantidade ← 10
    escreva ("Prepare 10 balões!")
    senão
      quantidade ← 20
      escreva ("Prepare 20 doces!")
  totalPronto ← 0
  enquanto totalPronto < quantidade faça
    prepare 1 (itemEscolhido)
    totalPronto ← totalPronto + 1
  itensProntos ← itensProntos + 1
  escreva ("O item (itemEscolhido) está pronto!")
escreva ("A festa está pronta!!!!")
```

MVP

Pseudocódigo

itensProntos ← 0

enquanto itensProntos < 3 **faça**

quantidade ← 0

entrada **leia** (itemEscolhido) variável

se itemEscolhido = "bolo" **então**

quantidade ← 1

escreva ("Prepare 1 bolo!")

senão

se itemEscolhido = "balão" **então**

quantidade ← 10

escreva ("Prepare 10 balões!")

senão

quantidade ← 20

escreva ("Prepare 20 doces!")

totalPronto ← 0

enquanto totalPronto < quantidade **faça**

prepare 1 {itemEscolhido}

totalPronto ← totalPronto + 1

itensProntos ← itensProntos + 1

escreva ("O item {itemEscolhido} está pronto!")

escreva ("A festa está pronta!!!")

O que é um pseudocódigo?

Variáveis

Entrada e Saída

Operadores

Condicional

Laço de repetição

O que é um pseudocódigo?

É uma forma de escrever um algoritmo com uma linguagem simples, sem necessidade de conhecer uma linguagem de programação específica.

Um algoritmo é qualquer sequência finita de passos para resolver um problema ou completar tarefa. É como dizemos para um computador o que queremos que ele faça.

Receita de Bolo

1. Separar os ingredientes: açúcar, margarina, ovos, etc
2. Bater o açúcar, a margarina e os ovos
3. Acrescentar a farinha, o leite e o fermento em pó
4. Untar a forma com farinha
5. Assar o bolo a 180°C em forno pré-aquecido por 40 minutos

Pseudocódigo

```
1. positivos ← 0
2. i ← 0
3. enquanto i < 5 faça
4.     escreva("Digite um número")
5.     leia( numero )
6.     se numero > 0 então
7.         positivos ← positivos + 1
8.     i ← i + 1
9. escreva( "O número de números positivos é {positivos} ." )
```

Pseudocódigo

itensProntos ← 0

enquanto itensProntos < 3 **faça**

quantidade ← 0

leia (itemEscolhido)

se itemEscolhido = "bolo" **então**

quantidade ← 1

escreva ("Prepare 1 bolo!")

senão

se itemEscolhido = "balão" **então**

quantidade ← 10

escreva ("Prepare 10 balões!")

senão

quantidade ← 20

escreva ("Prepare 20 doces!")

totalPronto **laço de repetição**

enquanto totalPronto < quantidade **faça**

prepare 1 {itemEscolhido} **operadores**

totalPronto ← totalPronto + 1

itensProntos ← itensProntos + 1

escreva ("O item {itemEscolhido} está pronto!")

escreva ("A festa está pronta!!!")

Aplicação e Avaliação do MVP

- Aplicação em sala de aula para uma turma de 26 alunos do 8º ano do Ensino Fundamental II da Escola de Aplicação da FEUSP.
- Alunos exploraram a simulação de forma livre.
- Avaliação do MVP:
 - ◆ Formulário de usabilidade (SUS): 26 respostas
 - ◆ Atividade de aprendizado: 13 respostas

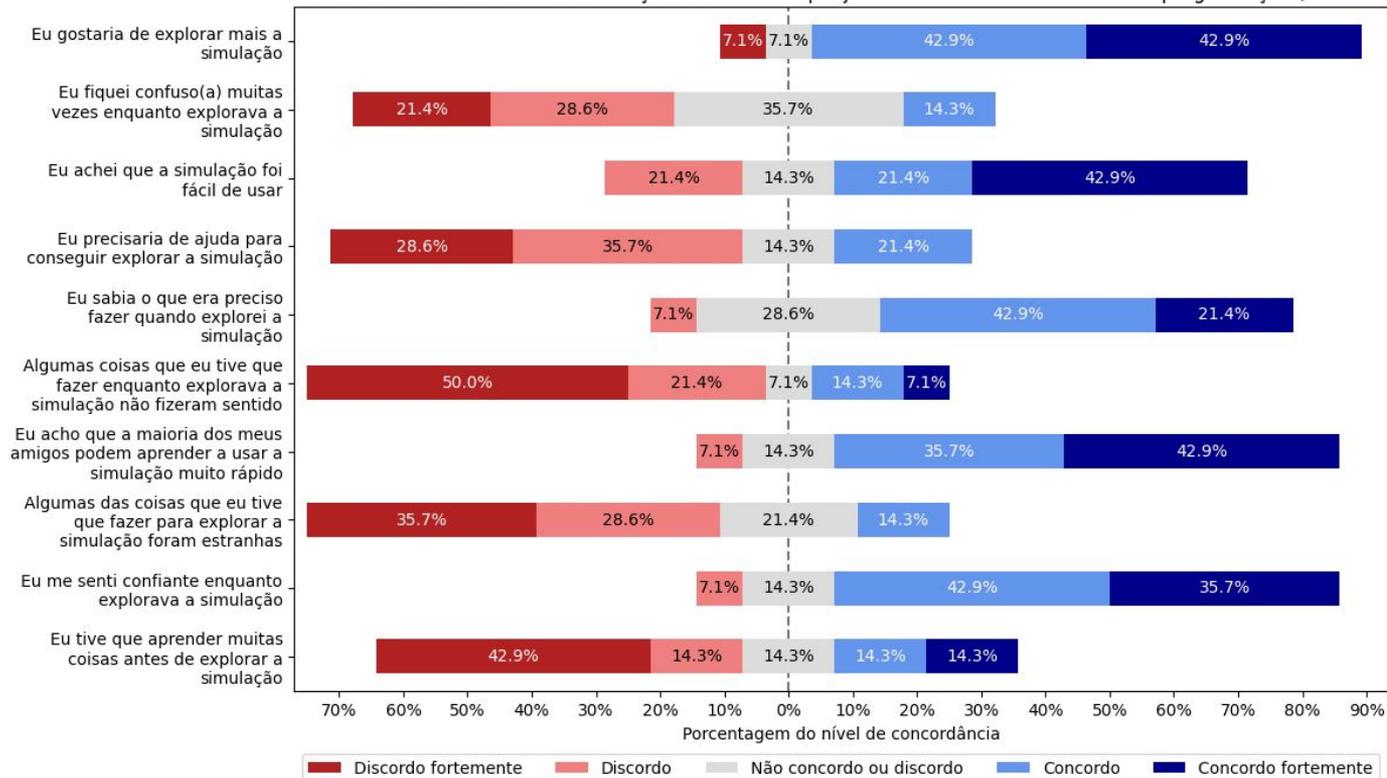
SUS adaptado para crianças (Putnam et al., 2020)



1. Eu gostaria de explorar mais a simulação.
2. Eu fiquei confuso(a) muitas vezes enquanto explorava a simulação.
3. Eu achei que a simulação foi fácil de usar.
4. Eu precisaria de ajuda para conseguir explorar a simulação.
5. Eu sabia o que era preciso fazer quando explorei a simulação.
6. Algumas coisas que eu tive que fazer enquanto explorava a simulação não fizeram sentido.
7. Eu acho que a maioria dos meus amigos podem aprender a usar a simulação muito rápido.
8. Algumas das coisas que eu tive que fazer para explorar a simulação foram estranhas.
9. Eu me senti confiante enquanto explorava a simulação.
10. Eu tive que aprender muitas coisas antes de explorar a simulação.

Resultados do Formulário de usabilidade

Nível de concordância com as afirmações dos alunos que já tiveram contato anterior com programação (14 alunos)



SUS score

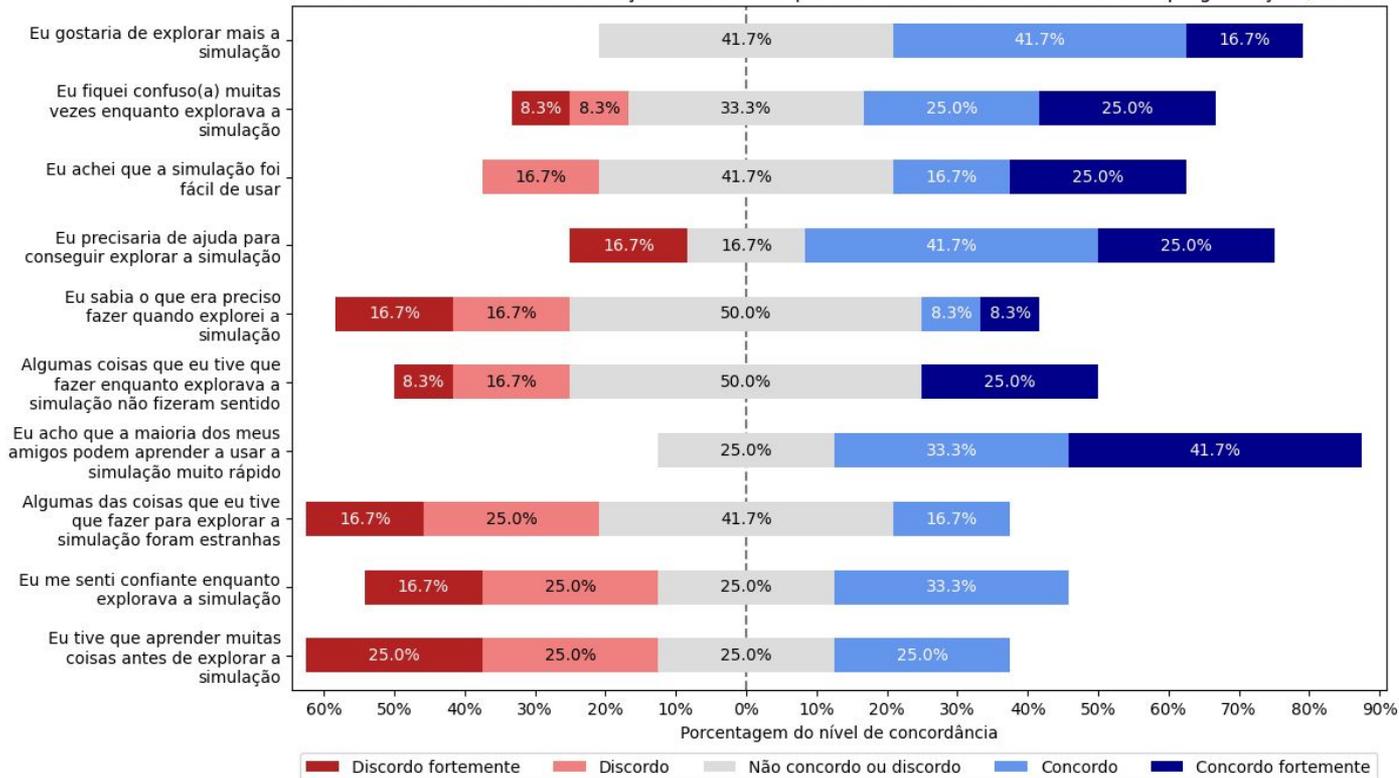
Média: 71.61

→ 7 acima

→ 7 abaixo

Resultados do Formulário de usabilidade

Nível de concordância com as afirmações dos alunos que não tiveram contato anterior com programação (12 alunos)



SUS score

Média: 53.96

→ 5 acima

→ 7 abaixo

Atividade de aprendizado

- Cada questão apresenta um pedaço do pseudocódigo com um ou mais conceitos de lógica de programação.
- Para cada questão, você deve selecionar apenas o conceito que melhor representa o pedaço de pseudocódigo correspondente.

1. Qual o conceito principal que aparece no trecho de pseudocódigo a seguir?

```
se itemEscolhido = "bolo" então  
  quantidade ← 1  
  escreva ("Prepare 1 bolo!")  
senão  
  se itemEscolhido = "balão" então  
    quantidade ← 10  
    escreva ("Prepare 10 balões!")  
  senão  
    quantidade ← 20  
    escreva ("Prepare 20 doces!")
```

- Variáveis
- Entrada
- Saída
- Operadores
- Condicional
- Laço de repetição

Resultados da Atividade de aprendizado

Acertaram todas ou quase todas as questões:

Simulação	Planejando a Festa	Guardando os Brinquedos
Alunos com conhecimento (7 alunos)	42.9%	14.3%
Alunos sem conhecimento (6 alunos)	33.3%	33.3%

Confusão entre os conceitos de programação:

- Trecho de pseudocódigo incluía muitos conceitos
- Falta de compreensão

Conclusões

- O uso de simulações interativas para ensinar conceitos de lógica de programação demonstrou potencial, principalmente, para crianças que já tiveram um primeiro contato com programação.
- No geral, os alunos mostraram interesse em explorar mais a simulação.
- Necessidade de melhorar a maneira de introduzir os conceitos para crianças que nunca tiveram contato com computação.
- Necessidade de diversas reformulações para o MVP, implicando em uma nova iteração nos ciclos de design e engenharia de DSR.

Sugestões para trabalhos futuros

- Introduzir os conceitos de programação na forma de tutorial antes de iniciar a interação com a simulação.
- Melhorar a indicação dos conceitos no pseudocódigo.
- Melhorar a atividade de aprendizado tentando remover a ambiguidade.
- Apresentar cada conceito em simulações separadas.

Referências Bibliográficas

[Brooke 1996] Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability evaluation in industry*, 189(194), 4-7.

[CodeCombat 2024] CodeCombat. URL: <https://codecombat.com/> (acesso em 27/11/2024).

[Crema et al. 2020] Crema, C., Gasparini, I. & Kemczinski, A. (2020). Cartilha com Atividades Desplugadas para o Ensino Médio. *Universidade do Estado de Santa Catarina*. URL: <http://educapes.capes.gov.br/handle/capes/585927> (acesso em 27/11/2024).

[Khatri et al. 2013] Khatri, R., Henderson, C., Cole, R., & Froyd, J. (2013, July). Over one hundred million simulations delivered: A case study of the PhET interactive simulations. In *Physics Education Research Conference* (pp. 205-208).

[Lego Mindstorms 2024] Lego Mindstorms EV3. URL: <https://kids.lego.com/pt-pt/sets/mindstorms/ev3-0729302d5ae04b5d88a30c2dd6d7afba> (acesso em 27/11/2024).

[PhET 2024] PhET. Physics Educational Technology. URL: <https://phet.colorado.edu/> (acesso em 27/11/2024).

[Putnam et al. 2020] Putnam, C., Puthenmadom, M., Cuervo, M. A., Wang, W., & Paul, N. (2020, April). Adaptation of the system usability scale for user testing with children. In *Extended abstracts of the 2020 CHI conference on human factors in computing systems* (pp. 1-7).

[Scratch 2024] Scratch. URL: <https://scratch.mit.edu/> (acesso em 27/11/2024).

[Wieringa 2014] Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering*. Springer.