MAC0499 - Trabalho de Formatura Supervisionado
Proposta de Trabalho

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

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

Abril 2024

Resumo

O ensino da computação para crianças e jovens é cada vez mais importante na Era Digital. O conhecimento de programação promove muitas competências fundamentais para o pensamento crítico e lógico desses jovens. Por isso, o ensino da computação ganhou cada vez mais espaço nas escolas de educação básica brasileiras, tornando-se parte dos currículos escolares. Ferramentas de apoio pedagógico são fundamentais nesse processo de aprendizagem e o uso de simulações interativas na educação tem se mostrado eficaz em outras ciências. Nesse sentido, este projeto de pesquisa tem como objetivo a criação de um produto mínimo viável (MVP - *Minimum Viable Product*) de uma ferramenta web de simulações interativas para o ensino de computação no ensino fundamental, juntamente com uma documentação. A simulação será baseada em conceitos de lógica de programação pré-determinados, como variáveis, entrada, condicionais e laços, e deve permitir que os estudantes a explore livremente. Ademais, ela deve estar acompanhada do seu pseudocódigo correspondente, a fim de promover o contato com a estrutura real de um código. Para realizar este trabalho, seguiremos a metodologia *Design Science Research* (DSR) em Engenharia de Software, na qual começamos por investigar o problema através de uma contextualização e, a partir disso, projetamos e validamos um artefato, o MVP. Além disso, realizaremos a implementação do artefato utilizando o *framework* Vue.js em Typescript. Por fim, pretendemos avaliar a usabilidade do MVP proposto, realizando avaliações com educadores e estudantes por meio de questionários de usabilidade, como o SUS (*System Usability Scale*).

1 Introdução

O ensino da computação estimula o desenvolvimento de importantes habilidades do mundo digital, como o raciocínio lógico, a análise e resolução de problemas, o pensamento crítico, a criatividade, a colaboração, entre outras. Essas habilidades são cada vez mais relevantes para crianças e adolescentes. Dessa forma, a implementação do ensino de computação na educação básica brasileira tem sido alvo de discussões nas últimas décadas.

Com a implantação da Base Nacional Comum Curricular (BNCC) em 2017, o Conselho Nacional de Educação (CNE) ficou responsável por elaborar as normas específicas sobre computação (Buchweitz et al., 2021). Entre 2019 e 2021, foram designados os membros da comissão para formular tais normas. Ademais, houve colaborações de pesquisadores da Sociedade Brasileira de Computação (SBC), do Centro de Inovação para a Educação Brasileira (CIEB), do Ministério da Educação (MEC), dentre outras organizações. Alguns estudos mostram o panorama desse processo de implementação da computação na educação básica brasileira.

de França e do Amaral, 2013 realizaram um mapeamento sistemático de artigos referentes ao ensino de computação na educação básica no Brasil, publicados entre os anos de 2009 e 2012. Os autores observaram um crescente interesse dos pesquisadores brasileiros no assunto no período analisado, com destaque para as instituições de pesquisa das regiões Nordeste e Sul do país.

Em outro estudo mais recente, dos Santos et al., 2021 realizaram uma revisão sistemática sobre as diversas abordagens de ensino de computação no ensino fundamental em estudos publicados entre 2009 e 2020, em bases nacionais e internacionais. Eles analisaram diferentes tipos de atividades voltadas ao ensino de computação, materiais didáticos, eixos e conceitos da computação e áreas de conhecimentos atreladas. Os autores apontaram como tendências a realização de pesquisas com o 5º e 6º anos, o uso de materiais "desplugados" e estudos sobre o eixo de pensamento computacional. Além disso, também observaram algumas lacunas referentes a equidade e inclusão.

Diante desse cenário, a SBC elaborou as Diretrizes de Ensino de Computação na Educação Básica (Ribeiro et al., 2019), visando facilitar a implementação do ensino de computação nos currículos das escolas brasileiras. Os conceitos de Computação são organizados em 3 eixos:

- Pensamento Computacional: refere-se à habilidade de analisar, compreender, modelar, comparar, solucionar e automatizar problemas e soluções de maneira sistemática, por meio de abstrações, análise de informações e da construção de algoritmos.
- Mundo Digital: nele ocorre a codificação ou representação de informações, o processamento dos dados codificados e a distribuição dessas informações de forma segura.
- Cultura Digital: envolve o conhecimento das tecnologias digitais, bem como as relações da computação com outras áreas do conhecimento e a participação crítica, ética e responsável na sociedade do Mundo Digital.

Dessa forma, é possível perceber que o ensino de computação não se limita à aprendizagem de uma linguagem de programação. Ela é apenas uma ferramenta utilizada para aplicar estes conceitos de forma a chegar na solução de problemas, a partir do conhecimento de teoria da computação e paradigmas de programação (Blatt et al., 2017). Assim, diferentes metodologias podem ser adotadas no ensino de programação para crianças e jovens e o uso de recursos didáticos apropriados, ferramentas ou aplicativos pedagógicos é indispensável no aprendizado desses conceitos.

Nesse sentido, há diversas maneiras de explorar a aprendizagem dos conceitos de computação: jogos, programação visual, programação com blocos, kits de robótica, simulações, *storytelling*, entre outras. Dentre as ferramentas utilizadas no ensino fundamental, nota-se uma preferência pelas linguagens visuais e plataformas focadas no ensino dos fundamentos e não no desenvolvimento (Gomes et al., 2017). A linguagem de programação em blocos é uma das mais citadas em estudos, apresentando diversas opções de ferramentas (Brezolin e Silveira, 2021; de Souza et al., 2021).

Deste modo, softwares de apoio ao ensino de programação devem facilitar a compreensão das abstrações envolvidas nos conceitos de computação, bem como, estimular o raciocínio lógico. Assim, o uso de simulações interativas facilita a visualização de ideias abstratas, utilizando exemplos concretos para representá-las. Fernandes, 2012 obtiveram resultados positivos na criação de objetos de aprendizagem na forma de animações e simulações em temas de lógica de programação, demonstrando o potencial dessa ferramenta. Ademais, o uso de simulações interativas na educação em outras áreas tem se mostrado eficaz. Em particular, a plataforma de simulações interativas PhET (Physics Education Technology)[1] é um recurso bastante utilizado em várias disciplinas de ciências (Khatri et al., 2013).

Assim, a simulação é uma abordagem interessante a ser explorada também no ensino da computação, pois da mesma forma que as ciências da natureza e das humanidades ajudam a explicar o mundo real, a ciência da computação ajuda a explicar o mundo digital (Ribeiro et al., 2019). Portanto, é natural que busquemos ferramentas análogas para o estudo dessas ciências, e até onde sabemos, não existem simulações de conceitos de lógica de programação semelhantes às encontradas na ferramenta PhET.

2 Objetivo

Este projeto de pesquisa tem como objetivo a criação de um produto mínimo viável (MVP - *Minimum Viable Product*) de uma ferramenta de simulações interativas para o ensino de conceitos de lógica

[1] https://phet.colorado.edu/

de programação para crianças no ensino fundamental. Para isso, iremos desenvolver uma versão da aplicação com um conjunto mínimo de requisitos, contendo uma simulação que envolva alguns conceitos de lógica de programação. Além disso, queremos que os usuários tenham contato com a estrutura real do código de programação correspondente a cada conceito simulado. Assim, pretendemos integrar à simulação uma visualização do pseudocódigo associado, gerado a partir das representações visuais.

Também queremos aplicar o MVP em sala de aula para obter *feedback* para um futuro desenvolvimento do sistema completo. Dessa forma, iremos avaliar a sua usabilidade com a ajuda de estudantes e professores, através de questionários de usabilidade. Ademais, queremos investigar como representar os conceitos visualmente de forma que facilite a compreensão da lógica de programação por trás deles.

3 Metodologia

No ensino da computação, alguns conceitos introdutórios são fundamentais para o aprendizado da lógica de programação, tais como variáveis, entrada e saída, operadores lógicos e aritméticos, condicionais, laços de repetição, funções, vetores, matrizes, entre outros. Desse modo, para que possamos testar a viabilidade do MVP proposto, a simulação desenvolvida deve conter um conjunto mínimo desses conceitos, a fim de verificar o entendimento deles pelos alunos.

Neste cenário, o projeto e a investigação do MVP podem ser conduzidos através do paradigma *Design Science Research* (DSR) em Engenharia de Software. A *Design Science* envolve as atividades de design e investigação de artefatos em um dado contexto por meio de etapas de conceitualização do problema, projeto da solução e validação empírica (Runeson et al., 2020; Wieringa, 2014).

Wieringa, 2014 apresentou diretrizes para realizar pesquisas em *Design Science* em Sistemas da Informação e Engenharia de Software. O autor mostra a atividade de design decomposta em três tarefas, designadas como ciclo de design. Este ciclo está inserido em um outro, de engenharia, no qual temos o resultado do ciclo de design sendo introduzido no contexto real e avaliado. As etapas de cada um dos ciclos estão descritas a seguir:

Ciclo de Engenharia:

- **Ciclo de Design:**
  - **Investigação do Problema:** qual problema deve ser investigado e por quê?
  - **Projeto do Tratamento (*Treatment Design*):** projeto de um ou mais artefatos para tratar o problema.
  - **Validação do Tratamento (*Treatment Validation*):** analisar se esse projeto contribui para o tratamento do problema, caso seja implementado.
- **Implementação do Tratamento (*Treatment Implementation*):** tratar o problema com um dos artefatos projetados.
- **Avaliação da Implementação (*Implementation Evaluation*):** verificar se o tratamento foi bem sucedido.

Ainda, o autor destaca que projetos de pesquisa em *Design Science* estão relacionados apenas às três etapas do ciclo de design. Runeson et al., 2020 define etapas similares a estas para este ciclo, envolvendo a conceitualização do problema, o projeto da solução e a validação empírica. Ademais, o autor explica que contribuições nesse paradigma de pesquisa devem ser avaliadas em relação a relevância, rigor e inovação.

Para realizar este projeto, seguiremos a metodologia proposta para pesquisas em *Design Science* em Engenharia de Software, além de realizarmos as demais etapas do ciclo de engenharia. Nas seções a seguir descrevemos as etapas dos Ciclos de Engenharia e de Design para esta pesquisa.

3.1 Investigação do Problema

Este projeto é uma continuação da pesquisa de Iniciação Científica (IC) intitulada "Uma Ferramenta de Simulações Interativas para Ensino de Computação para Crianças", realizada entre outubro de 2022 e outubro de 2023. Na IC, realizamos um estudo sobre o ensino de computação para crianças e também uma busca extensiva de ferramentas de apoio pedagógico existentes e similares a proposta neste trabalho. Os resultados obtidos na etapa de investigação do problema estão descritos resumidamente na Seção 1 deste documento.

3.2 Projeto do Artefato

O artefato que pretendemos projetar como tratamento do problema é um MVP de uma ferramenta de simulações interativas de conceitos de lógica de programação destinada a crianças do ensino fundamental. Um MVP (*Minimum Viable Product*) é um produto com funcionalidades suficientes para atrair um público inicial e validar uma ideia. Assim, para realizar o seu projeto, ou seja, construir um protótipo do MVP, primeiramente, definimos alguns requisitos mínimos, listados a seguir:

- O MVP deve apresentar uma tela com uma área de trabalho e uma opção de ajuda com uma documentação explicativa de uso;
- A área de trabalho deve apresentar um espaço fixo para a simulação e outro para o pseudocódigo correspondente;
- A simulação deve envolver os seguintes conceitos de lógica de programação: variáveis, entrada, condicionais e laços de repetição;
- A simulação deve apresentar um número limitado de interações possíveis com o usuário;
- A simulação deve encorajar os usuários a explorá-la livremente, com controles intuitivos e uma interface que possibilite boa usabilidade;
- O pseudocódigo deve ser gerado automaticamente conforme as interações com a simulação vão ocorrendo.

Além dos requisitos, é necessário também definir e descrever os conceitos de lógica de programação que serão simulados e a sintaxe do pseudocódigo que será gerado. Para isso, alguns protótipos iniciais de simulações a serem desenvolvidas foram feitos no Figma[2], uma plataforma web colaborativa para projetar interfaces. As Figuras 1 e 2 mostram dois desses protótipos.

Na Figura 1, temos um cenário de quarto de brinquedos. O protótipo da simulação apresenta elementos visuais como um relógio de parede, um baú para guardar objetos e brinquedos espalhados pelo chão. Além disso, temos dois painéis, um controle para regular a quantidade de brinquedos espalhados e um contador que mostra quantos brinquedos foram guardados. Ainda, o protótipo apresenta um pseudocódigo associado à simulação.

A princípio, a ideia desta simulação é apresentar duas interações possíveis: alterar a hora no relógio e guardar os brinquedos no baú de forma iterativa. Com isso, queremos abordar o conceito de variáveis, principalmente, com a ação de guardar os objetos em um contêiner; de entrada, com a informação recebida através da interação com o usuário ao alterar o horário; de condicionais, verificando se é o momento de organizar o quarto; e de laços de repetição, repetindo a ação de guardar cada objeto até que o quarto esteja organizado.

A Figura 2 apresenta um cenário de planejamento de uma festa. Nele há elementos visuais que representam itens que podem ser escolhidos para incorporar uma festa de aniversário. Ademais, temos um contador que mostra quantos itens referentes ao escolhido já estão preparados. Nesta simulação, o usuário poderá ter duas interações: escolher um item para a festa e "prepará-lo" (incrementar um determinado item) até estar completo. Deste modo, neste cenário, abordamos os conceitos de variáveis, por meio do total de objetos que devemos ter de acordo com a escolha do item; de entrada, a partir

[2]https://www.figma.com/

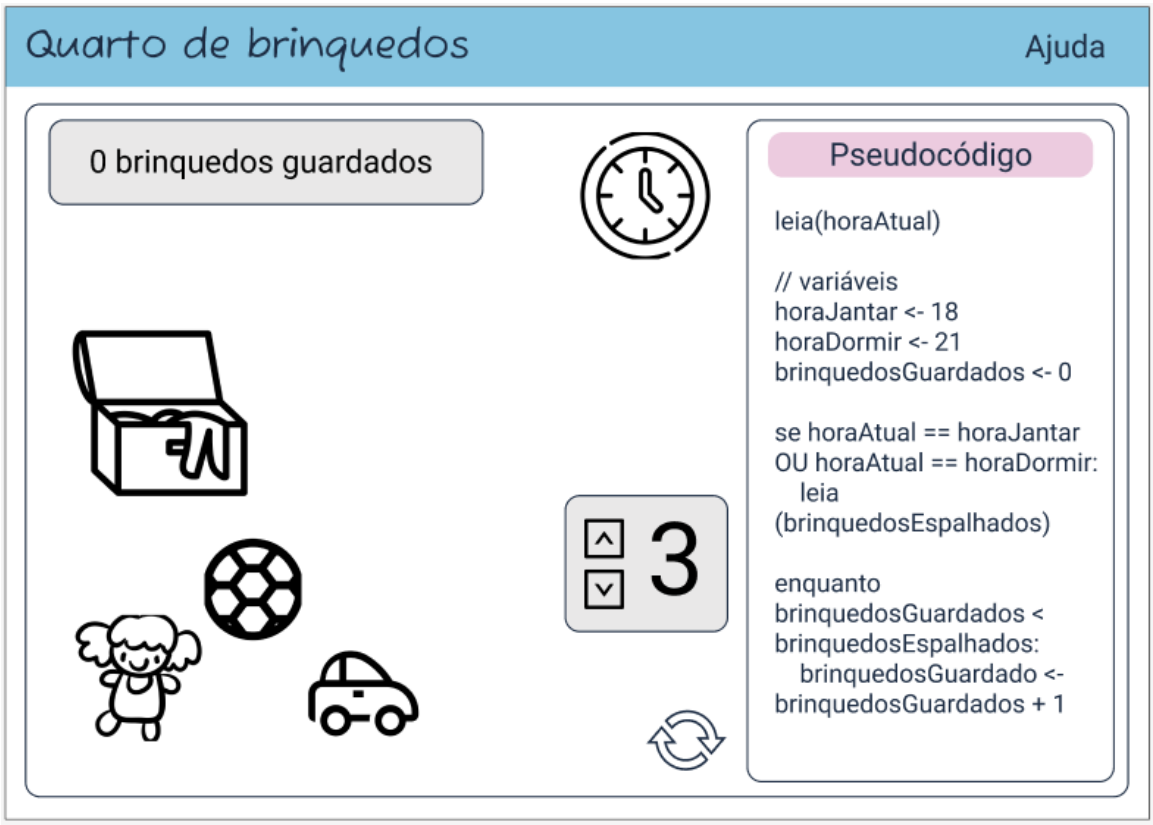

Figura 1: Protótipo da simulação Quarto de Brinquedos.

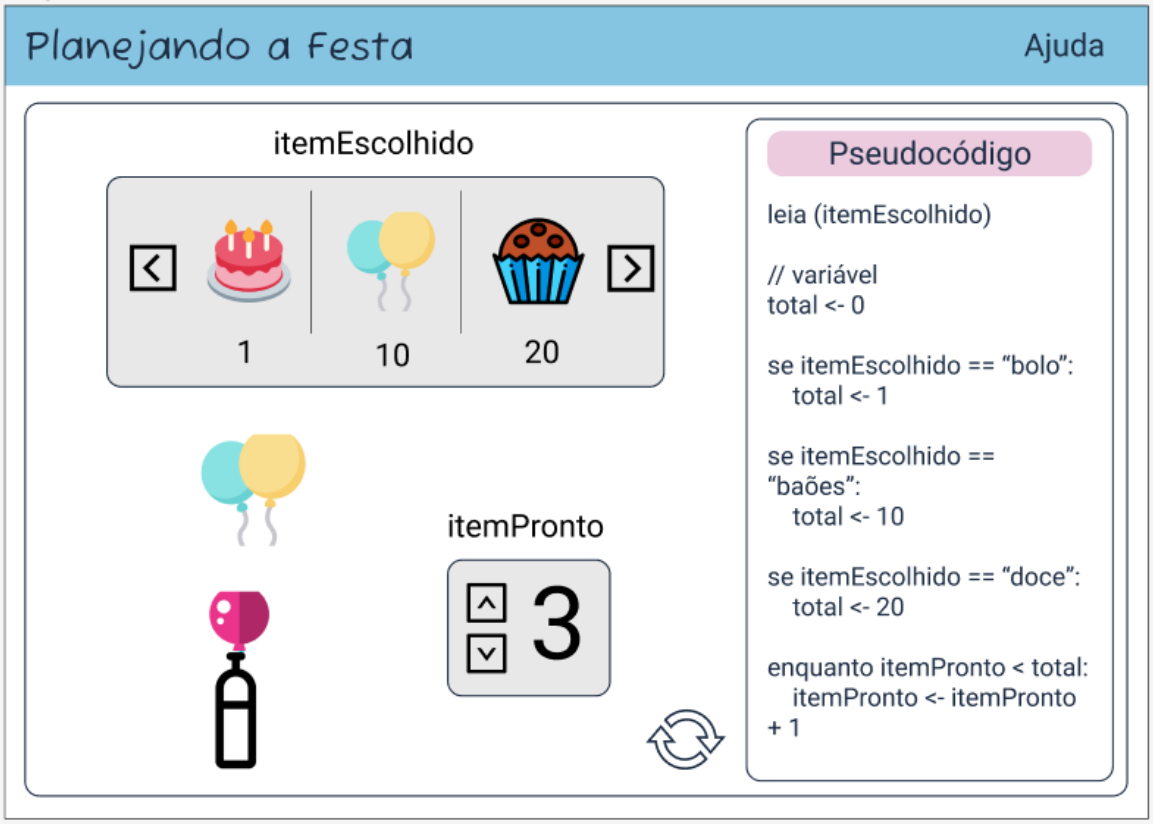

Figura 2: Protótipo da simulação Planejando a Festa.

da escolha de um item para a festa; de condicionais, verificando qual item foi escolhido; e de laços de repetição, conforme incremento dos objetos até atingirem o total de acordo com cada elemento.

A partir desses protótipos iniciais pretendemos estudar ideias similares e validar o artefato para analisar qual será possível implementar.

3.3 Validação do Artefato

A validação do protótipo do MVP será realizada com a ajuda do grupo de pesquisa do professor Dr. Leônidas Brandão do Departamento de Ciência da Computação do IME, que conduz projetos relacionados ao ensino de computação para crianças e adolescentes. Dessa forma, iremos verificar a usabilidade e utilidade do MVP a partir do protótipo projetado.

3.4 Implementação do MVP

O MVP será desenvolvido utilizando o *framework* de código aberto Vue.js[3], que é muito utilizado para criar aplicações de página única (*single-page applications*), com a intenção de agilizar o desenvolvimento, permitindo a utilização de soluções existente e reduzindo a necessidade de escrever códigos do zero. Juntamente, iremos utilizar a linguagem de programação Typescript[4].

3.5 Avaliação do MVP

O MVP desenvolvido será analisado em relação a sua usabilidade, considerando a facilidade de uso, de aprendizado e satisfação, por meio de questionários de usabilidade, métrica comum na avaliação de um protótipo ou sistema. Em particular, usaremos o questionário SUS (*System Usability Scale*), aplicado aos usuários finais.

Apesar de não haver medidas absolutas de usabilidade, é possível utilizar escalas gerais para comparar usabilidade em determinados contextos. O SUS representa uma escala de usabilidade com 10 itens que pode ser utilizada para avaliar sistemas (Brooke et al., 1996). É baseado na escala Likert, a qual contém afirmações e os avaliadores indicam o grau de acordo ou desacordo com cada uma, que varia de 0 a 5. Recomenda-se que as respostas para cada item sejam registradas de imediato, sem que os respondentes levem muito tempo pensando nelas. A pontuação do questionário SUS varia de 0 a 100 e determinados itens têm contribuições diferentes para a pontuação final.

O SUS é considerado um questionário robusto e confiável, tendo sido utilizado em diversos projetos de pesquisa e avaliações na indústria (Brooke et al., 1996). Embora não tenha sido projetado considerando necessidades específicas de compreensão por crianças, o questionário tem sido utilizado em vários estudos de testes e avaliações de ferramentas com crianças de diversas faixas etárias com sucesso (i.e. Dexheimer et al., 2017; Durand-Rivera, Martínez-González et al., 2020; Tasfia et al., 2023; Wrońska et al., 2015).

Ademais, Putnam et al., 2020 realizaram uma adaptação e teste do SUS com crianças na faixa etária de 7 a 11 anos. Os autores adaptaram o questionário, com o auxílio de professores da educação básica, em um contexto de aplicativos móveis de jogos focados no ensino de programação e pensamento computacional. As afirmações foram ainda modificadas pensando na separação de dois grupos de faixa etária, entre 7 e 8 anos e entre 9 e 11 anos. A Tabela 1 mostra os enunciados do SUS original e as adaptações propostas para os dois grupos mencionados, em inglês.

Além das simplificações das afirmações, os autores também utilizaram uma representação visual da escala de Likert (Figura 3), sugerida pelos docentes participantes do experimento.

Os resultados obtidos nos experimentos mostraram que o questionário modificado juntamente com a escala visual foi compreendido pelas crianças participantes, necessitando apenas de clarificações mínimas. Eles ainda sugeriram alterações nas afirmações 6, 8 e 10 para melhorar a compreensão e a confiabilidade delas.

Dessa forma, utilizando uma possível adaptação do questionário SUS, esperamos obter *feedback* de estudantes e educadores sobre o MVP para decidir em relação a sua usabilidade e a ideia de ensino de conceitos de programação proposta. A avaliação do MVP será realizada em sala de aula em uma

[3]https://vuejs.org/

[4]https://www.typescriptlang.org/

Tabela 1: Afirmativas em inglês do questionário de avaliação de usabilidade de sistemas SUS e adaptações propostas por Putnam et al., 2020 para crianças entre 7 e 11 anos.

Afirmativa	SUS original	Grupo 9-11 anos	Grupo 7-8 anos
1	I think that I would like to use this system frequently.	If I had this [app] on my iPad, I think that I would like to play it a lot.	I would like to play [app] a lot more.
2	I found the system unnecessary complex.	I was confused many times when I was playing [app].	[app] was hard to play.
3	I thought the system was easy to use.	I thought [app] was easy to use.	I thought [app] was easy to use.
4	I think that I would need the support of a technical person to be able to use this system.	I would need help from an adult to continue to play [app].	I would need help to play [app] more.
5	I found the various functions in this system were well integrated.	I always felt like I knew what to do next when I played [app].	I knew what to do next when I played [app].
6	I thought there was too much inconsistency in the system.	Some of the things I had to do when playing [app] did not make sense.	Some things in [app] made no sense.
7	I would imagine that most people would learn to use this system very quickly.	I think most of my friends could learn to play [app] very quickly.	[app] would be easy for my friends to learn.
8	I felt the system was cumbersome to use.	Some of the things I had to do to play [app] were kind of weird.	To play [app] I had to do some weird things.
9	I felt very confident using the system.	I was confident when I was playing [app].	I was proud of how I played [app].
10	I needed to learn a lot of things before I could get going with this system.	I had to learn a lot of things before playing [app] well.	There was a lot to learn to play [app].

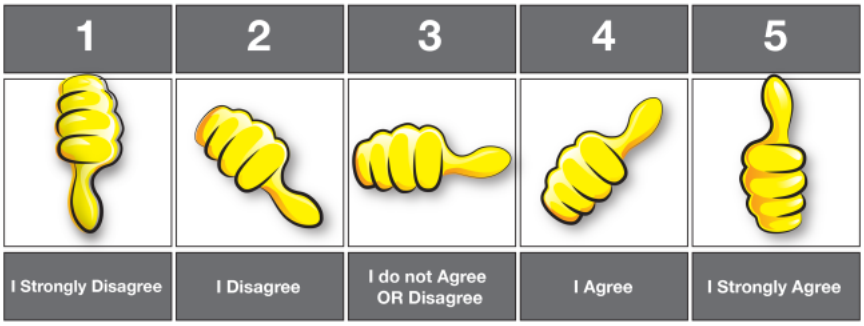

Figura 3: Representação visual da escala de Likert (Putnam et al., 2020).

possível parceria com o professor Leônidas em um projeto com minicursos de computação na Escola de Aplicação da FEUSP.

4 Plano de Trabalho e Cronograma

As atividades que serão desenvolvidas neste projeto durante o período do TCC estão descritas a seguir e o cronograma se encontra na Tabela 2.

1. Desenhar o protótipo da simulação;
2. Validar o protótipo com o grupo de pesquisa do prof. Leônidas Brandão do Departamento de Ciência da Computação do IME-USP;
3. Desenvolver o MVP com a simulação projetada;
4. Preparar formulários e plano de aulas;
5. Aplicar o MVP em sala de aula e recolher *feedbacks* de alunos e professores;
6. Analisar o *feedback* coletado;
7. Elaborar a monografia.

Tabela 2: Cronograma das atividades planejadas para o TCC.

Atividades	Mar.	Abr.	Mai.	Jun.	Jul.	Ago.	Set.	Out.	Nov.	Dez.
1										
2										
3										
4										
5										
6										
7										

5 Referências Bibliográficas

Blatt, L., Becker, V., & Ferreira, A. (2017). Mapeamento Sistemático sobre Metodologias e Ferramentas de apoio para o Ensino de Programação. *Anais do XXIII Workshop de Informática na Escola*, 815–824.

Brezolin, C. V. S., & Silveira, M. S. (2021). Panorama brasileiro de uso de ferramentas para desenvolvimento do pensamento computacional e ensino de programaçao. *Anais do XXIX Workshop sobre Educação em Computação*, 398–407.

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

Buchweitz, A., Siqueira, I. C. P., Capovilla, F. C., Braga, V., & Cunha, W. (2021). *Normas sobre Computação na Educação Básica - Complemento à BNCC*. Ministério da Educação. Conselho Nacional de Educação.

de França, R. S., & do Amaral, H. J. C. (2013). Ensino de computaçao na educaçao básica no brasil: Um mapeamento sistemático. *XXI Workshop sobre Educação em Computação*.

de Souza, F. A., Falcão, T. P., & Mello, R. F. (2021). O ensino de programação na Educação Básica: uma revisão da literatura. *Anais do XXXII Simpósio Brasileiro de Informática na Educação*, 1265–1275.

Dexheimer, J. W., Kurowski, B. G., Anders, S. H., McClanahan, N., Wade, S. L., & Babcock, L. (2017). Usability evaluation of the SMART application for youth with mTBI. *International journal of medical informatics*, *97*, 163–170.

dos Santos, A. d. S., Pereira, W. G., & de França, R. S. (2021). Como Ensinar Ciência da Computação para Crianças? Tendências e Lacunas de Pesquisa na Área. *Anais do XXIX Workshop sobre Educação em Computação*, 298–307.

Durand-Rivera, J., Martínez-González, C., et al. (2020). Usability evaluation of a tangible user interface and serious game for identification of cognitive deficiencies in preschool children. *International Journal of Advanced Computer Science and Applications*, *11*(6).

Fernandes, A. (2012). Animações e simulações para apoio ao ensino da lógica de programação. *Revista Técnico-Científica do IFSC*, 266–266.

Gomes, V., Pontes, R., Camelo, C., Cavalcanti, G., & Perkusich, M. (2017). Ensino de programação para crianças e adolescentes: um estudo exploratório. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, *6*(1), 490.

Khatri, R., Henderson, C., Cole, R., & Froyd, J. (2013). Over one hundred million simulations delivered: A case study of the PhET interactive simulations. *Physics Education Research Conference*, 205–208.

Putnam, C., Puthenmadom, M., Cuerdo, M. A., Wang, W., & Paul, N. (2020). Adaptation of the system usability scale for user testing with children. *Extended abstracts of the 2020 CHI conference on human factors in computing systems*, 1–7.

Ribeiro, L., Castro, A., Fröhlich, A. A., Ferraz, C. A. G., Ferreira, C. E., Serey, D., de Angelis Cordeiro, D., Aires, J., Bigolin, N., & Cavalheiro, S. (2019). Diretrizes da sociedade brasileira de computaçao para o ensino de computaçao na educaçao básica. *Sociedade Brasileira de Computação*.

Runeson, P., Engström, E., & Storey, M.-A. (2020). The design science paradigm as a frame for empirical software engineering. *Contemporary empirical methods in software engineering*, 127–147.

Tasfia, S., Islam, M. N., Nusrat, S. A., & Jahan, N. (2023). Evaluating usability of AR-based learning applications for children using SUS and heuristic evaluation. *Proceedings of the Fourth International Conference on Trends in Computational and Cognitive Engineering: TCCE 2022*, 87–98.

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

Wrońska, N., Garcia-Zapirain, B., & Mendez-Zorrilla, A. (2015). An iPad-based tool for improving the skills of children with attention deficit disorder. *International journal of environmental research and public health*, *12*(6), 6261–6280.