

Instituto de Matemática e Estatística - IME/USP
Bacharelado em Ciência da Computação - BCC

Projeto de Pesquisa de Iniciação Científica

Orientadora: Profa. Dra. Kelly Rosa Braghetto

Aluna: Marília Takaguti Dicezare

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 protótipo de uma ferramenta web de linguagem gráfica para especificar simulações interativas para o ensino de computação no ensino fundamental. As simulações serão baseadas em conceitos de lógica de programação, como variáveis, operadores lógicos e aritméticos, condicionais, laços, entre outros, e devem permitir que os estudantes as explorem livremente. Ademais, cada simulação deve estar acompanhada do seu pseudocódigo correspondente, a fim de promover o contato com a estrutura real de um código. Assim, iremos desenvolver um arcabouço robusto que contenha pelo menos uma simulação de conceitos de lógica de programação, e que sirva como base para a implementação de novas simulações no futuro, juntamente com uma documentação. Este arcabouço será estruturado com base na separação da lógica de negócios de entidades externas através da arquitetura hexagonal e do padrão de projetos MVC (*Model-View-Controller*), promovendo um código modular e reutilizável. Para isso, utilizaremos um framework web e bibliotecas JavaScript, os quais irão facilitar a manipulação de elementos gráficos e o gerenciamento de eventos. Por fim, pretendemos avaliar a usabilidade do protótipo proposto realizando validações com educadores e estudantes através de questionários de usabilidade, como o SUS (*System Usability Scale*) e o emoti-SAM.

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. Algumas delas também estão envolvidas no conceito de pensamento computacional, o qual vai muito além de programar, mas requer múltiplos níveis de abstração, sendo a forma como humanos resolvem problemas, tendo como base a matemática (Wing, 2006).

Essas habilidades são cada vez mais relevantes para crianças e jovens de diversas faixas etárias. Nesse sentido, muito tem sido discutido acerca da aprendizagem de computação na educação básica nas últimas décadas. Escolas em vários países ao redor do mundo já introduziram o ensino de computação. Alguns estudos revelam os desafios e as expectativas desta implementação.

Hubwieser et al. (2015) mostraram a situação do ensino de computação em escolas *k-12* (ensino fundamental) em 14 casos de estudo em estados de 12 países: Alemanha, Estados Unidos, Índia, Nova Zelândia, França, Coreia, Suécia, Reino Unido, Finlândia, Israel, Rússia e Itália. Eles identificaram diferenças em diversos aspectos nas abordagens, encontrando cerca de 40 termos diferentes para descrever áreas relacionadas à ciência da computação, 24 categorias de objetivos de ensino e 19 de conteúdos de computação, 4 subcategorias de linguagens de programação (sistemas baseados em hardware, ambientes de programação educacional com linguagem própria e baseados em outras linguagens, e linguagens de programação profissionais), e diferentes níveis de educação exigidos dos professores. Os autores acreditam que o mapeamento realizado pode contribuir no processo de planejamento do currículo de ciência da computação nas escolas de ensino fundamental e que a computação poderá se tornar uma matéria regular na educação básica em todos os países do mundo.

Outro trabalho, conduzido por Falkner et al. (2019), analisou a implementação do ensino de ciência da computação no *k-12*, comparando os requisitos curriculares definidos por padrões (pretendido) com o que realmente é implementado nas salas de aula pelos docentes. Foram analisados currículos em 7 países: Austrália, Inglaterra, Irlanda, Itália, Malta, Escócia e Estados Unidos, com foco apenas em conceitos de ciência da computação e linguagens de programação. Eles observaram que os tópicos mais ensinados são algoritmos, programação, pensamento computacional e representação de dados. Alguns tópicos, como inteligência artificial e robótica, constam no currículo pretendido de apenas um país cada, mas são abordados em sala de aula em diversos países. Ademais, eles descobriram que os educadores tendem a escolher as linguagens de programação baseados em fatores motivados pelos alunos ao invés de se basearem no currículo pretendido. O estudo mostrou que os professores utilizam tanto programação visual quanto baseada em texto no ensino fundamental, além de identificarem o uso comum de atividades desplugadas, apesar destas não estarem explicitamente definidas no

currículo padrão. Os autores também identificaram diferenças no alinhamento curricular entre os países, as quais acreditam que possam ajudar a guiar uma futura reforma curricular.

No Brasil, pesquisadores, instituições de ensino e organizações têm juntado esforços para implementar o ensino de computação na educação básica. Segundo relatório do Conselho Nacional de Educação (CNE) (Buchweitz et al. 2021), com a implantação da Base Nacional Comum Curricular (BNCC) em 2017, o CNE ficou responsável por elaborar as normas específicas sobre computação. 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 publicados entre os anos de 2009 e 2012. Eles observaram um crescente interesse dos pesquisadores brasileiros no assunto, com destaque para as instituições de pesquisa das regiões Nordeste e Sul do país. Em outro estudo, Bordini et al. (2016) apresentaram um levantamento de trabalhos relacionados ao pensamento computacional no ensino fundamental e médio realizados entre 2010 e 2015, com o intuito de mostrar o que já foi alcançado nessa área no Brasil. Eles identificaram diferenças nas metodologias de ensino, bem como nas ferramentas utilizadas e no público alvo dos estudos, mostrando conformidade com os trabalhos internacionais.

Ainda, Santos et al. (2018) analisaram a literatura sobre pensamento computacional e programação na educação básica brasileira no período de 2001 a 2016. Eles consideraram fatores como metodologia, ferramentas, público alvo e outros, mostrando novamente o crescente interesse dos pesquisadores no tema. Ademais, os autores apontaram algumas tendências e lacunas nessa área. Dos Santos et al. (2021) realizaram uma revisão sistemática sobre as diferentes 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: plugadas, desplugadas e híbridas, apontando tendências e lacunas observadas nos trabalhos, como por exemplo, equidade e inclusão.

Diante desse cenário nacional e mundial, 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, como mostra a Figura 1:

- **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.

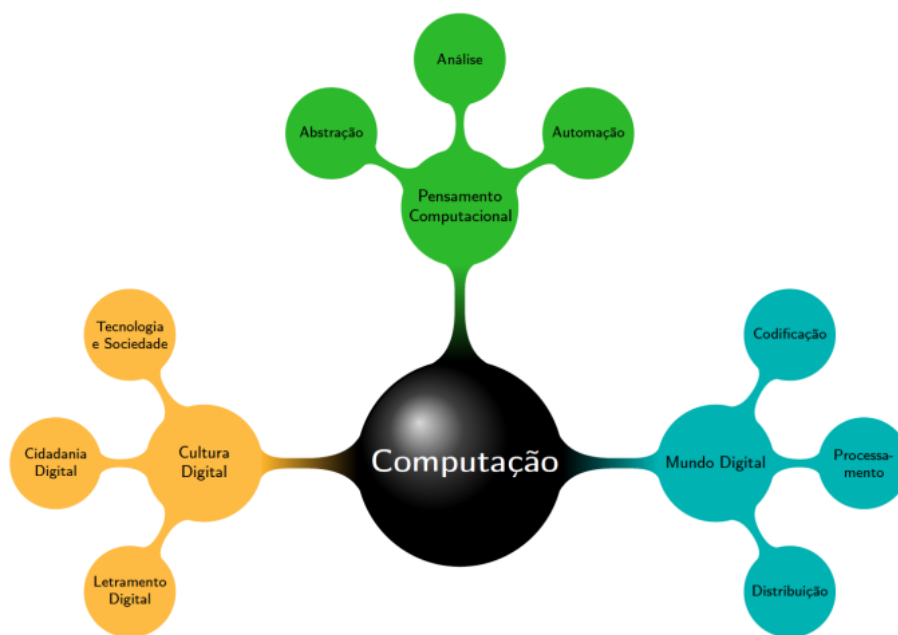


Figura 1 - Eixos da Computação (Ribeiro et al., 2019).

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 tocante, 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 & 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. Nesse sentido, o uso de simulações interativas facilita a visualização de conceitos abstratos, utilizando exemplos concretos para representá-los. Fernandes et al. (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.

Outros trabalhos mostram a efetividade das simulações no ensino em outras áreas. 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., 2014). Criado em 2002 pelo ganhador do prêmio Nobel, Carl Wieman, o projeto da Universidade de Colorado Boulder apresenta atualmente 159 simulações interativas distribuídas nas áreas de física, química, matemática, ciências da terra e biologia, destinadas tanto a alunos da educação básica como de ensino superior.

Entre 2012 e 2013, a plataforma realizou uma pesquisa com docentes que utilizavam a ferramenta, recebendo cerca de 2000 respostas de escolas nos Estados Unidos (Price et al., 2018). O estudo mostrou três aspectos que contribuem para a escolha da ferramenta na sala de aula. Primeiro, devido a sua flexibilidade, as simulações são utilizadas de diversas maneiras e com diferentes objetivos de aprendizagem, como entender conceitos, processos científicos e aumentar a motivação dos estudantes. Além disso, os docentes preferem que os estudantes tenham controle da simulação. Por fim, algumas propriedades percebidas das simulações foram a visualização, manipulabilidade e a capacidade de realizar demonstrações que não poderiam ser feitas em sala.

No Brasil, diversos estudos mostram pontos positivos da utilização dessa ferramenta como forma de agregar ao ensino tradicional. Uma pesquisa bibliográfica de publicações de autores brasileiros entre os anos de 2010 e 2020, realizada por Cardoso et al. (2021), mostrou que a utilização do simulador PhET, junto a uma metodologia de ensino, potencializou o aprendizado dos estudantes, além de torná-los participantes ativos desse processo.

Outro trabalho, conduzido por Cravo e Espartosa (2021), reforça esta conclusão. Eles avaliaram simulações de ciências e biologia, através de um protocolo de avaliação próprio e da realização de oficinas com futuros docentes, mostrando os desafios e potencialidade da ferramenta. Os autores concluíram que ela contribui positivamente no processo de ensino-aprendizagem, tornando-o mais didático e dinâmico, e que a atuação de professores como mediadores do ensino pode ajudar a contornar possíveis deficiências na sua utilização. Ainda, os trabalhos de Araújo et al. (2021) e Santos e da Silva (2020) apresentam conclusões similares em relação ao uso de simulações no ensino de física e química, respectivamente.

Assim, esta 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

¹ <https://phet.colorado.edu/>

até onde sabemos, não existem simulações de conceitos de lógica de programação iguais às encontradas no PhET.

Objetivo

Este projeto de pesquisa tem como objetivo a criação de um protótipo de uma ferramenta de linguagem gráfica para especificar simulações interativas para o ensino de computação no ensino fundamental. Para isso, iremos desenvolver um arcabouço para a criação de simulações de conceitos fundamentais de lógica de programação tais como variáveis, operadores lógicos e aritméticos, condicionais, laços, entre outros. Além das simulações, queremos que os usuários tenham contato com a estrutura real do código correspondente a cada conceito simulado. Assim, pretendemos integrar às simulações uma visualização do pseudocódigo associado a cada uma.

Utilizaremos um framework web e bibliotecas JavaScript para a construção deste protótipo. Avaliaremos a sua usabilidade com a ajuda de educadores e estudantes, através de questionários de usabilidade. Ademais, queremos realizar uma análise exploratória para descobrir as melhores formas de representar tais conceitos, através da ajuda de profissionais da área de educação na informática.

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, funções, vetores, matrizes, entre outros. Dessa forma, o projeto proposto deve criar um arcabouço para o desenvolvimento das simulações interativas desses conceitos. As simulações devem encorajar os estudantes a explorá-las livremente, com controles intuitivos e uma interface que possibilite boa usabilidade. A Figura 2 mostra um exemplo que apresenta tais aspectos.

Para obter tais funcionalidades, é necessário utilizar as ferramentas adequadas. O escopo do projeto inclui a criação de pelo menos uma simulação no arcabouço proposto para validação, com o intuito de fornecer a base para a ferramenta web de linguagem gráfica, junto a uma documentação, facilitando o desenvolvimento de novas simulações no futuro. A seguir, analisamos e definimos os requisitos e a arquitetura do sistema a ser desenvolvido.

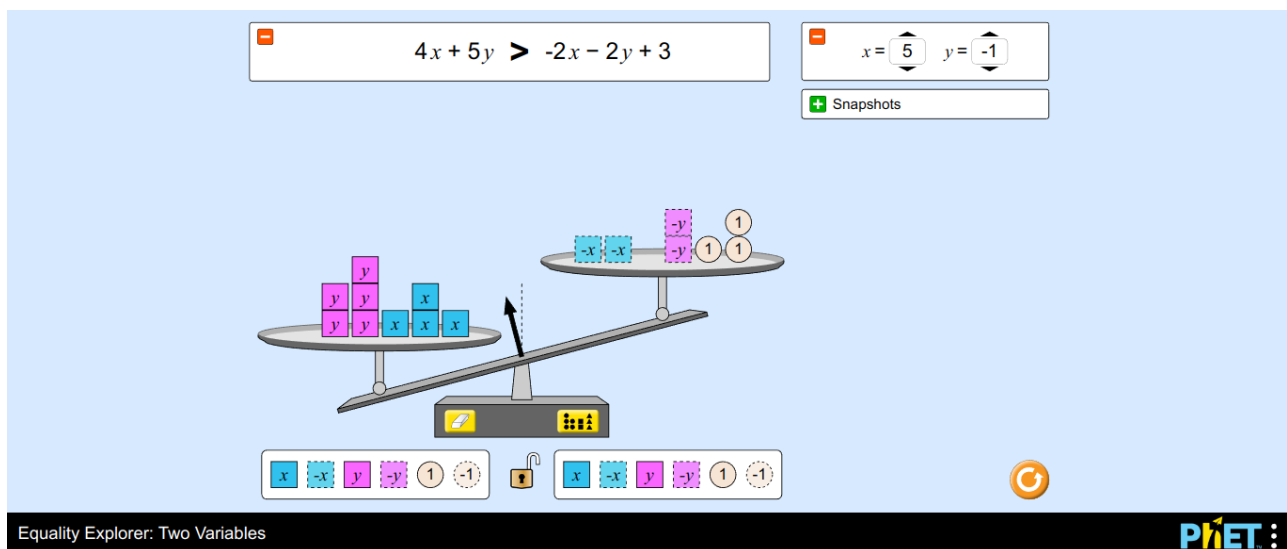


Figura 2 - Layout do simulador interativo do núcleo de Matemática do PhET: “Explorador de Igualdade: Duas Variáveis”. A interface apresenta elementos que podem ser arrastados pelo espaço da simulação, além de um botão de “reset” no canto inferior direito da tela, o qual leva às configurações iniciais. (PhET, 2022).

Requisitos Funcionais

A ferramenta de linguagem gráfica para especificar simulações interativas deverá permitir que o usuário escolha uma simulação relacionada a um conceito de programação específico. Em cada simulação, haverá um painel onde será possível a interação direta do usuário com a interface. A interação com a simulação será dada ao clicar, arrastar e adicionar elementos pelo espaço demarcado, ao digitar texto pelo teclado e através de recursos de áudio. Ademais, conforme o usuário interage com a simulação de um determinado conceito de programação, será possível visualizar o pseudocódigo referente aos conceitos utilizados em um outro painel. Esse outro painel será opcional, o usuário poderá habilitá-lo ou não, e será posicionado ao lado do espaço reservado para simulação.

A página de cada simulação também irá conter um menu com abas que disponibilizam informações sobre uma determinada simulação, recursos para educadores e um painel de ajuda. Deste modo, a ferramenta proposta deverá ter dois perfis de usuários possíveis, um destinado a educadores e outro a estudantes. O docente poderá explorar quais atividades deseja apresentar em sala para os alunos. Todavia, como dito anteriormente, as simulações serão pensadas para que os estudantes possam explorá-las livremente. A Figura 3 ilustra um protótipo da página da simulação de variáveis.

Variáveis

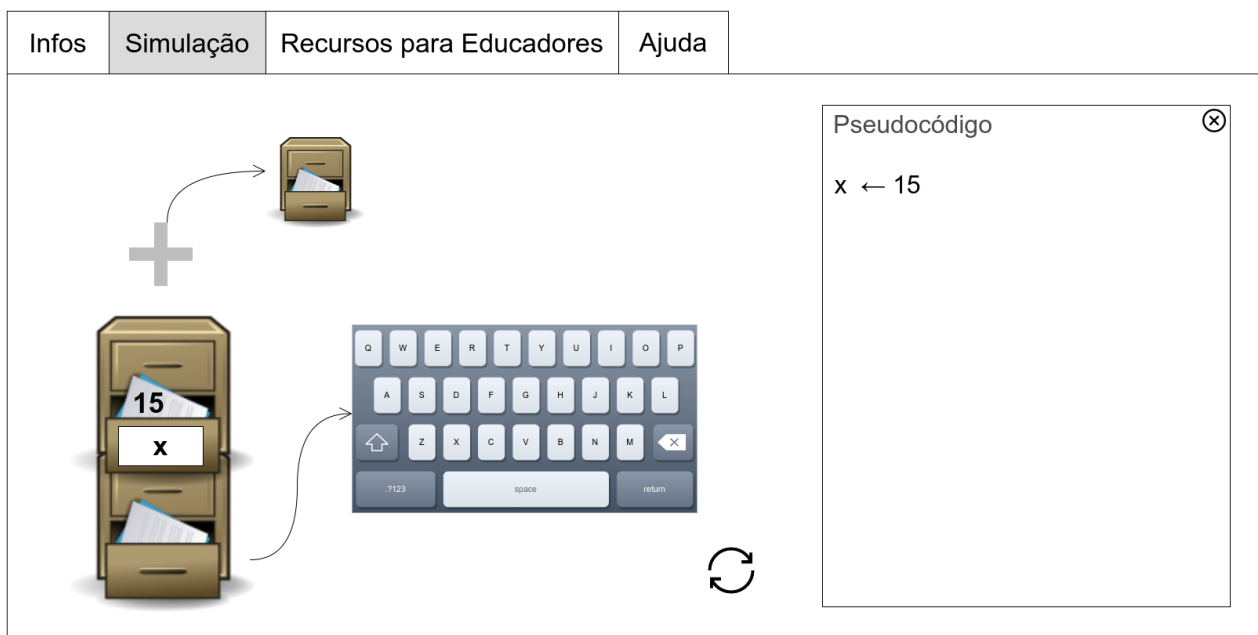


Figura 3 - Protótipo da página de simulação de variáveis. Com a aba de Simulação selecionada é possível observar o painel da simulação de variáveis e interagir com os elementos gráficos contidos nele, além de observar o painel de pseudocódigo a direita.

A ferramenta web ainda deve permitir o acesso pelo navegador de dispositivos de diversos tamanhos, desktops ou notebooks, tablets e smartphones. Dessa forma, as páginas devem ser responsivas, para que a usabilidade não seja afetada devido ao tamanho da tela. Além de possibilitar a interação tanto pelo mouse como pelo toque.

Por fim, o sistema deve apresentar aspectos como: textos descritivos para conteúdos que não apresentam texto, como imagens; distinção de elementos visuais e sonoros entre planos, incluindo cores, contraste, tamanho da fonte e áudio; tempo suficiente para que os usuários possam ler e usar o conteúdo; páginas navegáveis, promovendo maneiras de ajudar os usuários a navegar, encontrar o conteúdo e se localizar; entre outros (Caldwell et al., 2008). Estes aspectos promovem maior acessibilidade na ferramenta web para pessoas com algum tipo de deficiência.

Requisitos Não-Funcionais

O desenvolvimento desta ferramenta de simulações interativas será focado na criação de um arcabouço robusto com código e arquitetura modular, que ajude a guiar a implementação de novas simulações e a sua manutenção no futuro, promovendo um código aberto que seja reutilizável, testável, seguro, extensível e adaptável. Um arcabouço (ou framework) é uma estrutura de base sólida para a criação de software, no qual posteriormente são adicionadas

funcionalidades de mais alto-nível específicas de uma aplicação. Normalmente, é utilizado em sistemas onde há funcionalidades em comum a várias aplicações. Os frameworks de software podem incluir programas de suporte, bibliotecas, APIs (*Application Programming Interfaces*) e outros componentes que facilitam o desenvolvimento de um projeto ou sistema, promovendo um código reutilizável, testável, seguro, extensível e adaptável.

Além disso, a ferramenta deverá apresentar boa usabilidade, que será avaliada a partir de um questionário de usabilidade, uma vez que o sistema deve permitir que os usuários interajam livremente com a interface e aprendam a usá-la facilmente. Alguns critérios de usabilidade importantes são o tempo que o usuário interage com uma simulação, número de interações para realizar algum comando válido, quantidade de elementos disponíveis não utilizados, frequência do uso de recursos de ajuda, entre outros.

Ademais, a ferramenta web deve apresentar um bom desempenho, o qual também afeta a usabilidade. O sistema deve ser leve, de forma que o tempo de resposta seja rápido, e não exigir muito espaço em disco para armazenamento de informações, para que seja possível acessá-lo através de dispositivos com diferentes capacidades de processamento e com condições variadas de internet.

Arquitetura

Iremos arquitetar o sistema pensando na separação da lógica de negócios de entidades externas, como dispositivos de tempo de execução e bancos de dados. O estilo arquitetural que será utilizado é a Arquitetura Hexagonal ou Portas e Adaptadores (Hexagonal Architecture ou Ports and Adapters), na qual a comunicação entre drivers e a aplicação é feita através de adaptadores específicos em portas da aplicação (Cockburn, 2005). A Figura 4 apresenta o esquema da arquitetura hexagonal.

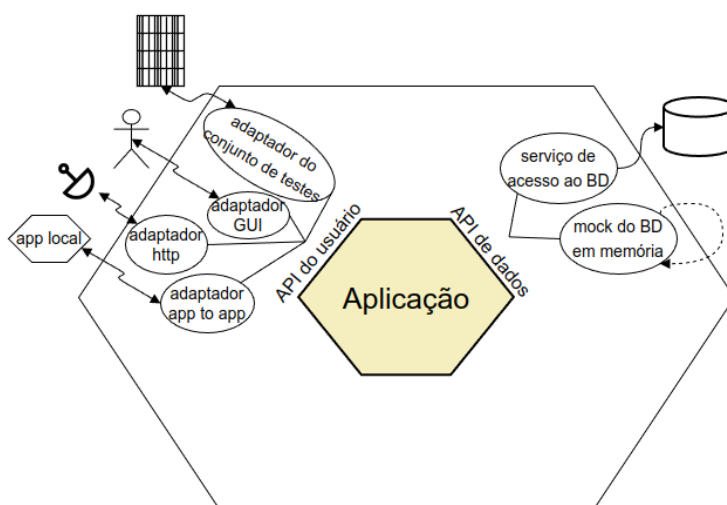


Figura 4 - Esquema da arquitetura hexagonal com duas portas ativas e diversos adaptadores em cada uma. (adaptado de Cockburn, 2005)

A forma hexagonal desta arquitetura proporciona um efeito visual que não a restringe a um desenho unidimensional, permitindo que as pessoas adicionem quantas portas e adaptadores acharem necessários. Assim, ela explora a assimetria existente entre dentro e fora da aplicação, seguindo uma regra de dependência, na qual o código pertencente à parte interna não deve ultrapassar para a região externa. Dessa forma, a aplicação fica completamente desacoplada da tecnologia do driver, facilitando o seu desenvolvimento e a testabilidade do todo.

Em nível de implementação as portas e adaptadores ainda podem se apresentar em dois níveis, primário e secundário. O primário diz respeito a portas e adaptadores que direcionam a aplicação e o secundário aos que são direcionados pela aplicação, seja para obter respostas ou para realizar notificações. Na implementação do arcabouço proposto, teremos basicamente três portas, uma relacionada ao usuário, outra aos dados e uma ligada a notificações, e vários adaptadores em cada uma. Então, a aplicação será direcionada pela interface gráfica do usuário (GUI - *Graphical User Interface*), pelas requisições http e por um conjunto de testes automatizados. Ademais, o sistema irá acessar um banco de dados através de um adaptador de serviço de acesso aos dados, e um *mailer*, por meio de um adaptador de email para a porta de notificações. O esquema da arquitetura da aplicação pode ser observado na Figura 5.

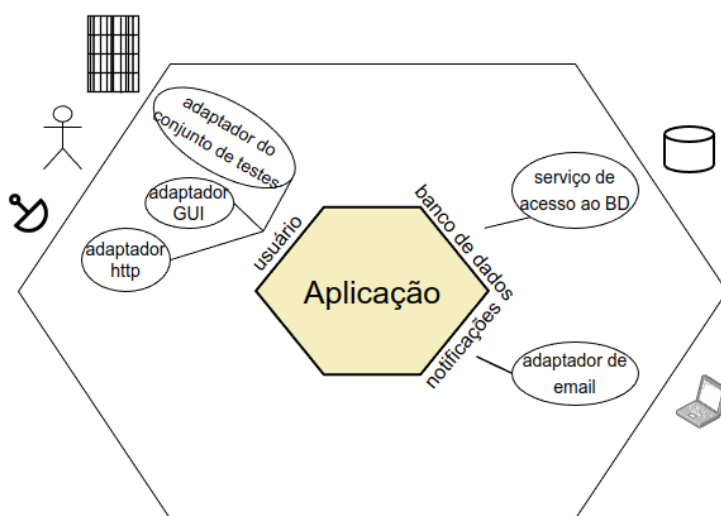


Figura 5 - Esquema da arquitetura hexagonal para a ferramenta proposta neste projeto, com três portas referentes a parte do usuário, do banco de dados e de notificações. Cada porta pode conter diferentes adaptadores.

Além disso, o arcabouço será desenvolvido com base no padrão arquitetural MVC (*Model-View-Controller*), o qual se apresenta em camadas (modelo, visão e controlador) e proporciona o isolamento das regras de negócio da interface gráfica do usuário. Este padrão implementa a ideia de portas e adaptadores no nível primário (Cockburn, 2005). No MVC, o modelo gerencia os dados e a lógica de negócios, a visão lida com leiaute e exibição das informações para o usuário e o controlador encaminha os comandos para o modelo e a visão.

Dessa forma, qualquer lógica do comportamento da aplicação, como a visibilidade e estado dos componentes da simulação fazem parte do modelo, a apresentação dessa informação, da visão, e o controle da entrada e eventos, do controlador.

Para implementar a arquitetura descrita, usaremos um framework e bibliotecas JavaScript que irão facilitar a manipulação de imagens e elementos gráficos. Alguns frameworks de código aberto para aplicações web bastante conhecidos e utilizados são React, AngularJS e Vue. O React é utilizado principalmente em interfaces interativas e utiliza componentes que facilitam o encapsulamento do código e uma sintaxe chamada JSX, a qual combina JavaScript e HTML. O AngularJS apresenta um ambiente que estende atributos HTML com diretivas e vincula dados ao HTML com expressões. O Vue é muito utilizado para criar aplicações de página única (*single-page applications*) e facilita a integração com outros projetos, pois ele pode ser adotado de forma progressiva. Ademais, algumas possíveis bibliotecas de código aberto que possuem integração com alguns dos arcabouços citados são Snap.svg, Konva.js, entre outras. Elas são bibliotecas ricas em animações e que facilitam o gerenciamento de eventos.

Forma de Análise dos Resultados

Para analisar a usabilidade do protótipo da ferramenta de linguagem gráfica proposto, pretendemos realizar validações com especialistas da área de educação e utilizar questionários de usabilidade aplicados aos dois perfis de usuários, educadores e estudantes, envolvidos na avaliação das simulações. Em particular, usaremos os questionários SUS (*System Usability Scale*) e emoti-SAM.

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 usabilidade em sistemas (Brooke, 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 deste questionário varia de 0 a 100 e determinados itens têm contribuições diferentes para a pontuação final.

Lourenço et al. (2022) realizaram uma tradução e adaptação cultural do SUS para o português do Brasil. O título proposto foi “Escala de Usabilidade de Sistema” e a versão pré-final desta tradução corresponde aos seguintes itens:

1. Eu acho que gostaria de usar esse sistema frequentemente.
2. Eu achei esse sistema desnecessariamente complexo.
3. Eu achei esse sistema fácil de usar.

4. Eu achei que precisaria de ajuda de uma pessoa técnica para ser capaz de usar esse sistema.
5. Eu achei que as várias funções desse sistema foram bem integradas.
6. Eu acho que o sistema apresenta muita inconsistência.
7. Eu imagino que a maioria das pessoas pode aprender a usar esses sistema rapidamente.
8. Eu achei esse sistema muito pesado para usar.
9. Eu me senti muito seguro usando o sistema.
10. Eu precisei aprender muitas coisas antes que pudesse utilizar esse sistema.

Também utilizaremos o emoti-SAM, uma ferramenta de avaliação direcionada a crianças, a qual utiliza emoticons para extrair respostas afetivas (Hayashi et al., 2016). Ele é uma adaptação do SAM (*Self-Assessment Manikin*), uma técnica não verbal de avaliação associada à reação afetiva de uma pessoa a diversos estímulos em três níveis: prazer, entusiasmo e dominância (Bradley & Lang, 1994). A Figura 6 apresenta os dois tipos de avaliação.

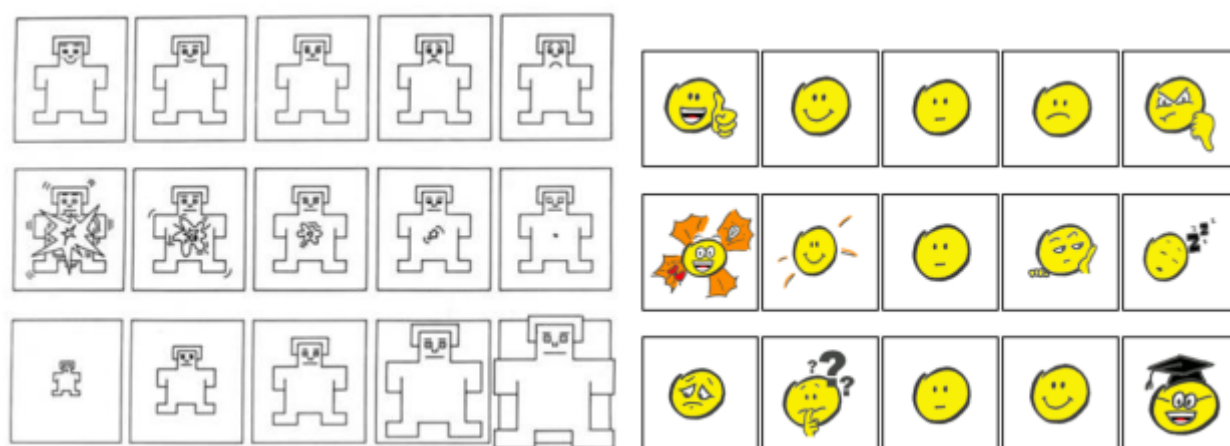


Figura 6 - A esquerda, o formulário SAM e, a direita, o Emoti-SAM (Hayashi et al., 2016).

O emoti-SAM é intuitivo e as crianças gostam de sua aparência, em comparação com o SAM original (Hayashi et al., 2016). Ademais, Brennand & Baranauskas (2018) mostraram que o emoti-SAM avalia aspectos emocionais, sendo relevante para a experiência de usuário (UX).

Com isso, esperamos obter *feedback* de educadores e estudantes para melhorar a usabilidade da ferramenta proposta.

Plano de Trabalho e Cronograma

As atividades que serão desenvolvidas neste projeto e o seu cronograma se encontram na Tabela 1. Adicionalmente, em cada etapa produziremos a documentação do arcabouço conforme seja necessário reportar registros relevantes.

Tabela 1 - Atividades planejadas e cronograma.

Atividade	Nov-Dez/ 22	Jan-Fev/ 23	Mar-Abr/ 23	Mai-Jun/ 23	Jul-Ago/ 23	Set-Out/ 23
1. Estudar os frameworks e bibliotecas JavaScript	x					
2. Projetar as classes do sistema	x	x				
3. Desenvolver o arcabouço de simulações		x	x	x		
4. Desenvolver simulações com ajuda de profissionais na área de educação em informática				x	x	
5. Avaliar a ferramenta junto a educadores e estudantes						x
6. Elaborar relatório e artigo para apresentar resultados						x

Referências Bibliográficas

Araújo, E. S., do Nascimento, J. L. B., Silva, J. C., & Bim, C. F. A. (2021). O uso de simuladores virtuais educacionais e as possibilidades do PhET para a aprendizagem de Física no Ensino Fundamental. *Revista de Ensino de Ciências e Matemática*, 12(3), 1-25.

Blatt, L., Becker, V., & Ferreira, A. (2017, October). Mapeamento Sistemático sobre Metodologias e Ferramentas de apoio para o Ensino de Programação. In *Anais do workshop de informática na escola* (Vol. 23, No. 1, pp. 815-824).

Bordini, A., Avila, C. M. O., Weissshahn, Y., da Cunha, M. M., da Costa Cavalheiro, S. A., Foss, L., ... e Reiser, R. H. S. (2016). Computação na educação básica no brasil: o estado da arte. *Revista de Informática Teórica e Aplicada*, 23(2), 210-238.

Bradley, M. M., e Lang, P. J. (1994). Measuring emotion: the self-assessment manikin and the semantic differential. *Journal of behavior therapy and experimental psychiatry*, 25(1), 49-59.

Brennand, C. V. L. T., e Baranauskas, M. C. C. (2018). Evaluating UX-case studies in socio-enactive scenarios. *Nuevas Ideas en Informática Educativa*, 14, 260-271.

Brezolin, C. V. S., e Silveira, M. S. (2021, July). Panorama Brasileiro de Uso de Ferramentas para Desenvolvimento do Pensamento Computacional e Ensino de Programação. In *Anais do XXIX Workshop sobre Educação em Computação* (pp. 398-407). SBC.

Brooke, J. (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., e Cunha, W. (2021). Normas sobre Computação na Educação Básica - Complemento à BNCC. Ministério da Educação. Conselho Nacional de Educação.

Caldwell, B., Cooper, M., Reid, L. G., Vanderheiden, G., Chisholm, W., Slatin, J., & White, J. (2008). Web content accessibility guidelines (WCAG) 2.0. *WWW Consortium (W3C)*, 290, 1-34.

Cardoso, K. T. D. S. N., de Sousa Rodrigues, J. C., Torres, R. L., Brito, F. W. P., Ramos, M. C., dos Santos Lins, W., e Carvalho, M. D. C. S. (2021). O ensino de ciências com o uso da ferramenta digital simulador phet por meio da estratégia investigativa nos anos finais do ensino fundamental

Science teaching with the use of the phet simulator digital tool through the investigative strategy in the final years of fundamental education.

Cravo, A. R., & Espartosa, K. D. (2021). Avaliação de simulações interativas em Ciências da plataforma on-line “PHET” por meio de parâmetros de avaliação e de oficinas com futuros docentes. *Revista de Ensino de Biologia da SBEnBio*, 658-679.

Cockburn, A. (2005). Hexagonal architecture. Disponível em: <https://alistair.cockburn.us/hexagonal-architecture/>. Último acesso em: 26/09/2022.

Falkner, K., Sentance, S., Vivian, R., Barksdale, S., Busuttil, L., Cole, E., ... & Quille, K. (2019, July). An international benchmark study of k-12 computer science education in schools. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 257-258).

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

Fernandes, A., Espíndola, A. L., & Oliveira, M. P. (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, October). Ensino de programação para crianças e adolescentes: um estudo exploratório. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação* (Vol. 6, No. 1, p. 490).

Hayashi, E. C., Posada, J. E. G., Maike, V. R., & Baranauskas, M. C. C. (2016, October). Exploring new formats of the Self-Assessment Manikin in the design with children. In *Proceedings of the 15th Brazilian Symposium on Human Factors in Computing Systems* (pp. 1-10).

Hubwieser, P., Giannakos, M. N., Berges, M., Brinda, T., Diethelm, I., Magenheimer, J., ... & Jasute, E. (2015). A global snapshot of computer science education in K-12 schools. In *Proceedings of the 2015 ITiCSE on working group reports* (pp. 65-83).

Khatri, R., Henderson, C., Cole, R., & Froyd, J. (2014). Over one hundred million simulations delivered: A case study of the PhET interactive simulations. In *Proceedings of the Physics Education Research Conference* (pp. 205-208).

Lourenço, D. F., Carmona, E. V., & de Moraes Lopes, M. H. B. (2022). Tradução e adaptação transcultural da System Usability Scale para o português do Brasil. *Aquichan*, 22(2), 4.

PhET. *Physics Educational Technology*. Disponível em: <https://phet.colorado.edu/>. Último acesso em: 10/09/2022

Price, A. M., Perkins, K. K., Holmes, N. G., & Wieman, C. E. (2018). How and why do high school teachers use PhET interactive simulations. *Learning*, 33, 37.

Ribeiro, L., Castro, A., Fröhlich, A. A., Ferraz, C. A. G., Ferreira, C. E., Serey, D., ... & Cavalheiro, S. (2019). Diretrizes da Sociedade Brasileira de Computação para o Ensino de Computação na Educação Básica. Sociedade Brasileira de Computação.

Santos, P. S., Araujo, L. G. J., & Bittencourt, R. A. (2018, October). A mapping study of computational thinking and programming in brazilian k-12 education. In 2018 IEEE Frontiers in Education Conference (FIE) (pp. 1-8). IEEE.

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

Santos, E. O., & da Silva, I. P. (2020). Revisão acerca do tema Simulações Computacionais no ensino de Química (2008–2017). *Debates em Educação*, 12(27), 841-855.

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.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.