

# ProgramSE: Um Jogo para Aprendizagem de Conceitos de Lógica de Programação

## *ProgramSE: A Game for Learning Programming Logic Concepts*

Rodrigo Ribeiro Silva  
Universidade Federal do Estado do  
Rio de Janeiro (UNIRIO)  
rodrigo.ribeiro@uniriotec.br

Luis Rivero  
Universidade Federal do  
Maranhão (UFMA)  
luisrivero@nca.ufma.br

Rodrigo Pereira dos Santos  
Universidade Federal do Estado do  
Rio de Janeiro (UNIRIO)  
rps@uniriotec.br

### Resumo

*O ensino de programação faz parte da formação acadêmica básica em Computação e áreas afins. Pesquisas anteriores reportam que estudantes têm muitas dificuldades relacionadas às disciplinas de programação, como a falta de compreensão do raciocínio lógico, que afeta o índice de reprovação e gera desistência de estudantes nos cursos de graduação da área. Com isso, técnicas e ferramentas são concebidas para apoiar o ensino-aprendizagem de disciplinas de programação, dentre elas, o uso de jogos digitais. Neste trabalho, o objetivo é apresentar um jogo digital educacional cuja utilização visa apoiar o processo de ensino-aprendizagem de programação para iniciantes no ensino superior. A proposta do ProgramSE é auxiliar o estudante para que possa conhecer novos conceitos de lógica de programação e fixar conhecimentos prévios, por meio de analogias a ações da rotina diária de um estudante em sua casa (e.g., ligar equipamentos, organizar objetos etc.). Partindo dos resultados de estudos prévios, o jogo foi construído e inspirado pela Teoria Pedagógica do Construtivismo e pela Taxonomia de Bloom. Adotou-se como base a programação em blocos para a construção de desafios a partir do Construct 3. A usabilidade e efetividade baseada na experiência do jogador foram avaliadas em duas turmas de iniciantes em programação de dois cursos de graduação por meio do modelo MEEGA+. A percepção dos alunos em relação à forma com que os conceitos de programação foram ensinados foi positiva. O jogo obteve notas medianas positivas para a maioria dos itens avaliados a partir do modelo MEEGA+. Além disso, na percepção dos estudantes, o jogo pode auxiliar no processo inicial de ensino-aprendizagem de programação. Os resultados indicaram a aceitação dos estudantes quanto aos jogos digitais voltados ao ensino, devendo-se considerar os aspectos de jogabilidade para o engajamento.*

**Palavras-Chave:** *educação em programação, jogos digitais, jogos educacionais.*

### Abstract

*Programming education is part of the basic academic training in Computer Science and related areas. Previous research reports that students face several difficulties related to programming disciplines. Moreover, the lack of the understanding of logical reasoning has been one of the main reasons for the high failure rate and the student evasion in the context of undergraduate courses in the field. As such, techniques and tools are designed to support the teaching and learning of programming disciplines, such as the use of digital games. In this context, this work aims to present an educational digital game to aid the teaching and learning process of programming for beginners in higher education. The purpose of ProgramSE is to help students know new concepts of programming logic and to establish previous knowledge through analogies to actions of a student's daily routine at home (e.g., turning an equipment on, organizing objects etc.). From the results of previous studies, the development of the game was inspired by Bloom's Taxonomy and Constructivism. In addition, it was based on block programming to build challenges from the game engine called Construct 3. The usability and effectiveness based on experience with the game were evaluated in two classes of beginners in programming at the university based on the MEEGA+ model. The students' perception of how programming concepts were taught was positive. Therefore, the proposed game had positive median scores for most items assessed with the MEEGA+ model. Moreover, given the students' perception, the game can assist in the initial teaching-learning process of programming concepts. Results also indicate the students' acceptance of digital games. However, it is important to consider the gameplay for enhancing the students' engagement during the use.*

**Keywords:** *programming education, digital games, educational games.*

Cite as: Silva, R. R., Rivero, L., & Santos, R. P. (2021). ProgramSE: A Game for Learning Programming Logic Concepts (ProgramSE: Um Jogo para Aprendizagem de Conceitos de Lógica de Programação). *Brazilian Journal of Computers in Education (Revista Brasileira de Informática na Educação - RBIE)*, 29, 301-330.  
DOI: 10.5753/RBIE.2021.29.0.301

## 1 Introdução

As disciplinas introdutórias dos cursos na área de Computação possuem grande importância para o desenvolvimento de habilidades relacionadas à solução de problemas com o raciocínio lógico (Krzyzanowski *et al.*, 2019). Elas devem estimular habilidades multidisciplinares (criatividade e pensamento crítico) para fortalecer o pensamento lógico dos iniciantes, fazendo-os pensar além da limitação computacional, raciocinando e argumentando diante de desafios. A construção dessa base fortalece a motivação dos alunos, podendo ser um fator para diminuir os riscos de reprovação e evasão nas disciplinas. No entanto, alguns métodos utilizados no modelo de ensino convencional podem não ser atrativos e engajadores para alguns estudantes (Backlund e Hendrix, 2013). Em programação, a metodologia de ensino utilizada, geralmente, envolve a criação de sistemas com linguagens de programação, o que pode não ser tão simples (Yildiz *et al.*, 2020).

No entanto, o processo de ensino-aprendizagem de lógica é árduo e inovações no ensino facilitam ou auxiliam a aquisição de conhecimento. Neste contexto, métodos educacionais lúdicos podem ser um recurso facilitador (Silva e Falcão, 2020). Da perspectiva do desenvolvedor, uma interface pronta para criar uma linguagem visual que emite um código gerado pelo usuário, porém com foco na construção de aplicativos. Além disso, vale destacar que, entre as metodologias de ensino utilizadas no contexto educacional, o modelo instrucionista é muito aplicado, no qual o estudante assume um papel passivo na aquisição do conhecimento. No mundo globalizado, as pessoas estão acostumadas a acessarem informações de maneira rápida, direta e prática, de modo que os métodos de ensino vigentes podem tornar as salas de aula espaços entediantes. Para Piteira e Haddad (2011), isso gera baixa motivação entre os alunos e, conforme Scaico e Scaico (2016), o interesse é um aspecto que influencia no seu engajamento para aprender. Nesse contexto, os jogos educacionais surgem como alternativa, pois possibilitam um processo de aprendizagem participativo, comunicativo e capaz de levar o aprendiz a resolver problemas de forma lúdica.

Segundo Victal *et al.* (2015), os jogos digitais são ferramentas úteis para o aprendizado, justificando o seu uso na educação. Nesse sentido, Hoelfmann (2016) realizou um estudo que indica como os jogos podem auxiliar os alunos em sua percepção, tendo um grande potencial para se tornarem importantes instrumentos no processo de ensino-aprendizagem. Para Uzunca e Jansen (2016), os jogos são uma nova forma de aprender, pois proporcionam: (1) o aprendizado em uma realidade virtual e/ou com papel simulado, o que encoraja o jogador, pois ele sempre pode recomeçar; (2) o ‘aprender fazendo’, envolvendo-se, motivando-se e entretendo-se; (3) a possibilidade de instigar a capacidade de encontrar e usar a informação sem a necessidade de memorização; e (4) a simulação de ambientes complexos/caros e de situações perigosas/críticas. Além disso, ferramentas populares como o *Scratch*<sup>1</sup> e o *Blockly*<sup>2</sup> ajudam o usuário a abstrair a linguagem de programação e desenvolver códigos baseado em blocos, ou seja, focando apenas na lógica sintaticamente correta (Seraj *et al.*, 2019). Tais ferramentas são uma forma intuitiva e visual de construir um código (Medeiros *et al.*, 2020).

Levando-se em conta os esforços de proporcionar um ensino engajador e construtivo para o estudante, o objetivo deste trabalho é apresentar a concepção e uma avaliação inicial de um jogo digital educacional cuja utilização visa apoiar o processo de ensino-aprendizagem de programação para iniciantes no ensino superior (i.e., primeiro semestre da graduação). Este jogo deve permitir que o estudante possa conhecer novos conceitos de lógica de programação e fixar conhecimentos prévios por meio de ações da rotina diária de um estudante em sua casa (e.g., ligar equipamentos, organizar objetos etc.). Como raciocínio lógico permanece uma das razões mais comuns para o índice de reprovação e desistência de estudantes em cursos de Computação (Santos e Costa, 2006; Medeiros *et al.*, 2020), buscou-se propor uma ferramenta que seja atrativa e engajadora para

<sup>1</sup> <https://scratch.mit.edu/>

<sup>2</sup> <https://developers.google.com/blockly/>

estudantes iniciantes. O desenvolvimento se deu com base no modelo construtivista que, por sua vez, foca na aprendizagem na qual o estudante se torna o construtor da própria sabedoria e o professor é agente facilitador do desenvolvimento cognitivo (Neto *et al.*, 2012). A meta é auxiliar o ensino de programação, apresentando o pensamento lógico, desafiando estudantes a aprenderem ou fixarem o conteúdo. Há softwares com proposta similar, porém não estão disponíveis em português e focados no público do ensino superior (Neto *et al.*, 2017; Silva *et al.*, 2017).

O artigo está estruturado nas seguintes seções, além desta introdução: a Seção 2 apresenta a fundamentação teórica; a Seção 3 apresenta a metodologia; a Seção 4 apresenta o jogo ProgramSE e os detalhes do seu desenvolvimento; a Seção 5 descreve o estudo experimental realizado para a avaliação do jogo; a Seção 6 apresenta as limitações e ameaças à validade desta pesquisa; por fim, a Seção 7 conclui o artigo com as considerações finais e trabalhos futuros.

## 2 Fundamentação Teórica

A falta de desenvolvimento e de maturidade do raciocínio lógico representa uma das principais razões do alto índice de reprovação nas disciplinas de algoritmos (Medeiros *et al.*, 2020). Assim, é importante pensar em um cenário no qual processo de ensino-aprendizado aconteça de forma eficaz a ponto de estimular o estudante a prosseguir mesmo diante de desafios e possíveis reprovações. Nesse caso, o principal papel do professor passa a ser incentivar esse crescimento.

Para Vygotsky (1980), existem três níveis de desenvolvimento e aprendizagem (Figura 1): (1) Zona de Desenvolvimento Atual (ZDA), em que o aprendiz é capaz de realizar sozinho certas tarefas; (2) Zona de Desenvolvimento Proximal (ZDP), considerada o ideal, na qual o aprendiz consegue realizar determinadas tarefas com a ajuda de terceiros; e (3) Nível de Desenvolvimento Potencial (NDP), em que as tarefas estão além do nível de desenvolvimento cognitivo do aprendiz, de modo que ele não consegue desempenhar nenhuma tarefa, mesmo com a ajuda de outros. Logo, é importante determinar a zona em que as atividades devem se encontrar, pois, se estiverem na ZDA, os estudantes conseguem executá-las sem avanços significativos e, se estiverem além da zona ZPD (i.e., NDP), o aluno pode ficar desmotivado.

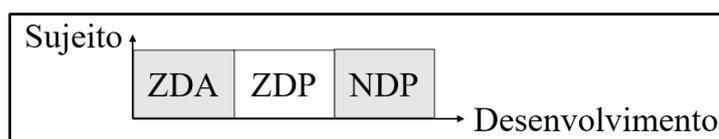


Figura 1: Níveis de desenvolvimento. Adaptado de (Vygotsky, 1980).

Para atingir a zona adequada, é importante entender e esboçar o cenário almejado para o processo de ensino-aprendizagem. A partir de uma análise, Santos *et al.* (2008) identificaram o tutor, o monitor e o aprendiz como os principais atores nesse contexto. Todos eles têm papéis distintos no processo e possuem características próprias: (i) *educador*: possui a função de mediar e facilitar o processo de ensino-aprendizagem (professor); (ii) *monitor*: atua como colaborador do processo, complementando o educador (alunos mais experientes); e (iii) *aprendizes*: conjunto de pessoas que têm em comum a assimilação de conteúdo de disciplina como objetivo principal.

Neste contexto, Henrique e Silva (2017) propõem um catálogo de requisitos pedagógicos para auxiliar o desenvolvimento de software educacional, destacando duas metodologias: a Teoria Pedagógica do Construtivismo (CON) e a Taxonomia de Bloom (TX). CON tem como principais objetivos a construção do conhecimento e a autonomia do aprendiz, ou seja, propõe-se que o aluno participe ativamente do próprio aprendizado. Para Henrique e Silva (2017), os requisitos principais para CON são: (1) exibir *feedback* construtivo e individualizado; (2) propor reflexões críticas; (3) propor ao aprendiz construir a maior quantidade de soluções possíveis; e (4) permitir que o aprendiz realize escolhas durante a interação. Por sua vez, TX (ou taxonomia dos objetivos

educacionais) busca auxiliar na identificação da hierarquia do propósito do sistema ligado ao desenvolvimento cognitivo do aprendiz. Ainda segundo os autores, os conteúdos apresentados devem ser divididos em vários níveis de conhecimento, iniciando sempre do menor nível de complexidade. Ambas as metodologias inspiraram o desenvolvimento do jogo proposto.

A seguir, é apresentado o estado da arte de jogos digitais educacionais e trabalhos realizados na área de jogos relacionados com ensino de programação. A partir do estado da arte levantado, é proposto o jogo ProgramSE, que tem por objetivo o apoio ao ensino de programação em turmas iniciais de cursos de graduação, introduzido na Seção 4. Mais detalhes sobre o contexto de ensino do conteúdo são apresentados na Seção 3 (Metodologia) e na Seção 5 (Avaliação).

## 2.1 Jogos Digitais Educacionais

Os jogos digitais têm sido um recurso utilizado para auxiliar o processo de ensino-aprendizagem, visto que são uma forma lúdica para abordar um problema (Hoelfmann, 2016). A lógica de funcionamento dos jogos digitais envolve a capacidade de promover engajamento de usuários, curiosidade, colaboração e aprendizado pelos erros (Sánchez, 2014). Por sua vez, os jogos digitais educacionais não se destinam apenas ao entretenimento do jogador, mas são desenvolvidos para o ensino de algum conteúdo (Sheldon, 2012), ou seja, para expandir os conhecimentos dos jogadores, simulando atividades práticas em um ambiente não real (Moreira, 2018). No cenário de ensino de programação, por exemplo, os jogos digitais podem ajudar os estudantes a sanarem suas dúvidas com representações visuais dos conceitos abordados em sala. Nesse sentido, para que um jogo digital possa compor um cenário de ensino, faz-se necessária uma combinação de elementos de jogos para engajar o jogador no contexto proposto. Pode-se então definir uma estratégia apoiada na aplicação de elementos de jogos para atividades *non-game* utilizada para influenciar e causar mudanças no comportamento de indivíduos e grupos (Bunchball Inc., 2010).

Segundo Werbach e Hunter (2012) e Costa e Marchiori (2015), os elementos de jogos podem ser classificados de três tipos (Tabela 1): (1) *dinâmicos*, que representam o mais alto nível de abstração dos elementos; (2) *mecânicos*, que se referem a ações mais específicas; e (3) *componentes*, que são aplicações específicas visualizadas e utilizadas na interface do jogo. Kapp (2012) também comenta sobre outro importante item: abstração. Nesse sentido, os jogos devem envolver a abstração da realidade, assim como um software. As histórias e signos utilizados no jogo podem carregar tais abstrações, concentrando-se apenas nos aspectos essenciais do contexto. Além destes elementos, a movimentação deve ser considerada um elemento de jogo, visto que a maneira como o personagem se desloca pelo ambiente proposto é um importante requisito a ser analisado. A representação do jogador no jogo digital deve ser capaz de se locomover pelo cenário e, a partir desse movimento, concluir as etapas propostas para atingir o objetivo final.

Além da necessidade de definir os elementos de jogos, a sua idealização e concepção deve contar com diversas etapas. Para facilitar esse processo, existem os motores (do inglês, *game engines*, ou apenas *engine*), que consistem em programas de computador capazes de simplificar o processo de criação de jogos digitais. Esses motores servem como um conjunto de bibliotecas que facilitam o desenvolvimento, fazendo com que o programador não precise criar tudo do zero. Geralmente, dão suporte para animações, sons, colisões, parte gráfica, entre outros. Cada motor de jogo possui um ambiente de desenvolvimento e suas características. A Tabela 2 apresenta alguns motores e também exibe a comparação entre as principais tecnologias utilizadas para desenvolvê-los. Vale ressaltar que uma investigação prévia acerca de elementos e motores foi realizada e publicada (Silva *et al.*, 2018a) e foi uma das bases para a construção do ProgramSE. O comparativo não é exaustivo e foi realizado para auxiliar na escolha do motor mais adequado para desenvolver o jogo digital educacional proposto neste trabalho. Estes requisitos foram selecionados pela equipe de desenvolvimento por conveniência. Portanto, dependendo dos objetivos da equipe e contexto de uso, outros atributos poderão ser considerados.

Tabela 1. Elementos de Jogos. Fonte: (Werbach e Hunter, 2012).

Categoria	Elemento	Descrição
Dinâmico	Emoções	Jogos podem criar diferentes tipos de emoções, especialmente a da diversão (reforço emocional que mantém as pessoas jogando).
	Narrativa	Estrutura que torna o jogo coerente. A narrativa não tem que ser explícita, como uma história em um jogo. Também pode ser implícita, na qual toda a experiência tem um propósito em si.
	Progressão	Ideia de dar aos jogadores a sensação de avançar dentro do jogo.
	Relacionamentos	Refere-se à interação entre os jogadores, seja entre amigos, companheiros ou adversários.
	Restrições	Refere-se à limitação da liberdade dos jogadores dentro do jogo.
Mecânico	Aquisição de recursos	O jogador pode coletar itens que o ajudam a atingir os objetivos.
	Feedback	A avaliação permite que os jogadores vejam como estão progredindo no jogo.
	Chance	Os resultados de ação do jogador são aleatórios para criar uma sensação de surpresa e incerteza.
	Cooperação e competição	Cria-se um sentimento de vitória e derrota.
	Desafios	Os objetivos que o jogo define para o jogador.
	Recompensas	O benefício que o jogador pode ganhar a partir de uma conquista no jogo.
	Transações	Significa compra, venda ou troca de algo com outros jogadores no jogo.
	Turnos	Cada jogador no jogo tem seu próprio tempo e oportunidade para jogar.
	Vitória	O “estado” que define ganhar o jogo.
Componente	Avatar	Representação visual do personagem do jogador.
	Bens virtuais	Itens no jogo que os jogadores podem coletar e usar de forma virtual e não real, mas que ainda tem valor para o jogador.
	Boss	Um desafio difícil no final de um nível que tem de ser derrotado, a fim de avançar no jogo.
	Coleções	Formadas por itens acumulados dentro do jogo. Emblemas e medalhas são parte de coleções.
	Combate	Disputa que ocorre no jogo, a partir da qual o jogador possa derrotar oponentes em uma luta.
	Conquistas	Recompensa que o jogador recebe por fazer um conjunto de atividades específicas.
	Conteúdos desbloqueáveis	A possibilidade de desbloquear e acessar certos conteúdos no jogo se os pré-requisitos forem preenchidos. O jogador precisa fazer algo específico para ser capaz de desbloquear o conteúdo.
	Medalhas	Representação visual de realizações no jogo.
	Gráfico social	Capacidade de ver amigos que também estão no jogo e ser capaz de interagir com eles. Um gráfico social torna o jogo uma extensão de sua experiência de rede social.
	Missão	Similar a “conquistas”. É uma noção de jogo de que o jogador deve fazer executar algumas atividades que são especificamente definidas na estrutura do jogo.
	Níveis	Representação numérica da evolução do jogador. O nível do jogador aumenta à medida que o jogador se torna melhor no jogo.
	Pontos	Ações no jogo que atribuem pontos. São muitas vezes ligadas a níveis.
	Presentes	A possibilidade distribuir ao jogador coisas como itens ou moeda virtual para outros jogadores.
	Ranking	Lista jogadores que apresentam as maiores pontuações/conquistas/itens em um jogo.
	Times	Possibilidade de jogar com outras pessoas com mesmo objetivo.

Tabela 2. Comparação entre os principais motores de jogos. Fonte: (Silva et al., 2018a).

Nome	Programação	Prós	Contras	Licença
Adobe Flash Professional CS5	ActionScript	Multiplataforma, formato vectorial	Linguagem própria	Adobe
Alice	Visual	3D, criação de métodos	Não possui muitos recursos	Livre
Construct 2/ Construct 3	Java e C++	Drag-n-drop, detecta colisão e noções de eventos	Só pode executar em HTML no Free	Scirra
GameMaker	Delphi	Tutoriais, versão "lite", linguagem simples	Linguagem própria	Yoyo
Scratch	Visual	Fácil "programação"	Comandos limitados	Livre
Stencyl	Haxe, C++ e Java	Drag-n-drop, multiplataforma, possibilita extensões	-	Stencyl, LLC
Unity 3D	C++ e C#	3D, multiplataforma, fácil de usar, melhor condição de licenciamento	Ferramentas limitadas	Unity Technologies

## 2.2 Trabalhos Relacionados

Em uma revisão sistemática previamente realizada e publicada (Silva et al., 2018b), os autores deste trabalho identificaram estudos que relatam propostas relacionadas com o jogo ProgramSE – a contextualização da revisão sistemática para este trabalho é feita na Seção 3. Dentre os trabalhos identificados, no que se refere ao enfoque na programação em blocos para a construção de desafios, Silva et al. (2017) construíram uma plataforma de programação virtual para ensinar crianças e adolescentes a programar. Os autores basearam o seu software na programação em blocos e utilizaram uma biblioteca do Blockly para facilitar o desenvolvimento. O texto não deixa claro quanto à eficácia do jogo, mas foram listados pontos a serem desenvolvidos sugeridos pelos estudantes. Em outro trabalho, Netto et al. (2017) apresentaram um jogo voltado para o ensino-aprendizagem de algoritmos no ensino fundamental e médio. Segundo a avaliação dos jogadores,

utilizando um questionário simples com 4 questões sobre a experiência do jogador, o software obteve um retorno positivo e otimista. Porém, é concluído que são necessárias novas versões para melhorar a captação de *feedback*, além da necessidade de uma avaliação mais detalhada.

Outros trabalhos têm focado no apoio ao desenvolvimento de jogos educacionais para o ensino de programação. Por exemplo, Monclar *et al.* (2018) apresentam uma revisão da literatura em que 26 jogos para o ensino de programação são identificados. Além de apresentar um resumo de cada jogo identificado, indicando a categoria do jogo, sua dinâmica e conteúdos apresentados, os autores apresentam uma análise comparativa. Por sua vez, Silva *et al.* (2017) apresentam um ambiente de programação em blocos. A plataforma, denominada Poderu, é voltada para crianças e adolescentes e conta com um personagem robô que faz ações programadas por blocos. O jogo foi avaliado aplicando heurísticas de usabilidade e obtendo a opinião de crianças de 10 a 12 anos através de um formulário *online*, porém sem apresentar muitos detalhes sobre o contexto da avaliação e os resultados. Finalmente, Raabe *et al.* (2015) apresentam a utilização de três jogos aplicados ao ensino de programação. Nessa pesquisa, os jogos foram avaliados do ponto de vista de alunos de graduação, respondendo questões relacionadas à sua preferência, engajamento e diversão, estilo do jogo e percepção sobre programação e conceitos envolvidos.

Tendo em vista os trabalhos relacionados, que versam sobre soluções para ensino de programação em blocos para a construção de desafios para iniciantes em Computação (o que inclui o perfil de adolescentes como potenciais ingressos em cursos de graduação), e a necessidade de desenvolver um método engajador e facilitador do ensino-aprendizagem de programação para iniciantes no nível superior, o presente trabalho apresenta um jogo digital educacional para este fim e público-alvo. A ferramenta proposta foi projetada em português e tem como objetivo apoiar o aluno no aprendizado de conceitos de programação. A motivação principal é que o jogo possa ser uma alternativa para estimular estudantes iniciantes em cursos superiores e auxiliar na diminuição das taxas de evasão nas disciplinas introdutórias de programação.

### 3 Metodologia

Esta pesquisa foi dividida em três etapas (Figura 2): a primeira se refere à concepção do jogo; a segunda se refere à sua construção; e a terceira se refere à sua avaliação. Para a realização da primeira etapa, o primeiro passo consistiu na condução de uma revisão sistemática da literatura (RSL). O objetivo da RSL foi identificar o panorama da utilização de jogos digitais para a aprendizagem de certos conceitos do tema no ensino superior no contexto do Brasil (Silva *et al.*, 2018b). Foram identificadas tendências observadas na década 2008-2017 no que se refere à utilização de jogos digitais no ensino de programação para iniciantes em Computação no nível superior no Brasil. Por exemplo, concluiu-se que, nos últimos anos, oficinas para ensino de programação com jogos digitais foram exploradas de maneira crescente, partindo de uma (2008-2012) para cinco (2013-2017) iniciativas. Além disso, novas formas de avaliar os trabalhos dos alunos apareceram com o passar do tempo, tais como avaliação por especialista e estudos experimentais apoiados por instrumentos semelhantes ao modelo MEEGA+ (Petri *et al.*, 2019).

Além da RSL, foi realizada uma investigação de elementos e motores para suporte ao desenvolvimento de jogos digitais (Silva *et al.*, 2018a). Com base nos resultados, a segunda etapa consistiu na construção do jogo ProgramSE. A partir das diretrizes pedagógicas encontradas na RSL e levando em consideração os principais requisitos para o engajamento nos jogos digitais, foi escolhido o motor Construct 3 para o seu desenvolvimento. Na terceira etapa, a avaliação foi realizada com 32 alunos ingressantes nas disciplinas de programação em dois cursos de graduação a partir de questionário baseado no modelo MEEGA+. Este modelo foi escolhido por ser um modelo sistemático para avaliar a qualidade de jogos educacionais utilizados para o ensino de conceitos de computação, já aplicado em diversos estudos (Petri *et al.*, 2019). Cabe destacar que,

para verificar a viabilidade desta avaliação, foi realizado inicialmente um piloto que permitiu identificar e acomodar sugestões de melhorias recebidas.

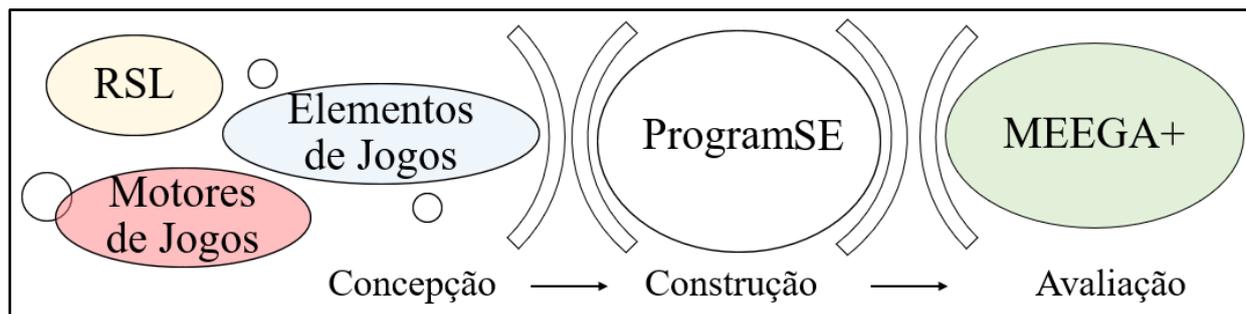


Figura 2: Etapas da pesquisa.

O desenvolvimento do ProgramSE considerou algumas teorias pedagógicas, como a Teoria Pedagógica do Construtivismo e a Taxonomia de Bloom. Tais teorias auxiliaram as soluções estudadas na RSL e, então, embasaram a primeira versão dos desafios do jogo e a forma como o conteúdo é apresentado aos estudantes. Além disso, a RSL permitiu a identificação de formas de avaliação em estudos experimentais apoiados por instrumentos semelhantes do modelo MEEGA+, que exploram a experiência de uso de jogos educacionais com estudantes, conforme adotado nesta pesquisa (detalhes na Seção 5). Adicionalmente à preocupação com o conteúdo, ProgramSE também foi norteado por elementos de usabilidade para melhorar a experiência dos jogadores e tornar o processo de jogar agradável.

## 4 ProgramSE

Tendo em vista os problemas da alta evasão no ensino superior em Computação, os elementos extraídos da literatura sobre o engajamento em jogos digitais e seu desenvolvimento, além do panorama da utilização de jogos digitais no domínio da educação, o jogo ProgramSE foi desenvolvido com o propósito de cativar estudantes e exercitar o pensamento de lógica e conceitos de programação. O principal objetivo educacional é melhorar a aquisição do conhecimento em relação à forma com que os conceitos de programação são ensinados para iniciantes na graduação.

### 4.1 Desenvolvimento do Jogo

ProgramSE é um software voltado para estudantes iniciantes em programação no ensino superior, voltado para o público adulto jovem (dos 18 aos 28 anos), em português e que contempla de conceitos iniciais a conceitos intermediários de lógica de programação. O público foi definido desde o começo, no objetivo da RSL, considerando-se estudos para entender o cenário do ensino de programação no nível superior no Brasil. A mecânica do jogo foi desenvolvida utilizando o motor de jogos *Construct 3*, que foi escolhido pelos seguintes aspectos: preço/orçamento (promoção), baixa complexidade, interface intuitiva, comunidade para suporte, exportação para diferentes plataformas e atualizações disponibilizadas pela instituição/empresa. Para os cenários do jogo, utilizou-se o programa para projeção de casas e *design* de interior *Sweet Home 3D*<sup>3</sup>. O software livre *Inkscape*<sup>4</sup> também foi usado como editor de imagens.

A missão do jogo visa estimular o aluno a perceber como a programação está presente em ações usuais, fazendo com que o estudante construa – com exemplos concretos – o raciocínio lógico. Partindo deste princípio, à luz do trabalho de Henrique e Silva (2017), o cenário para o uso do ProgramSE foi inspirado pelas metodologias pedagógicas do Construtivismo, em que o

<sup>3</sup> <http://www.sweethome3d.com/>

<sup>4</sup> <https://inkscape.org/>

aprendiz é autônomo e participa ativamente do próprio aprendizado, e da Taxonomia de Bloom, que apregoa que o conteúdo deve ser exibido de uma forma hierárquica, iniciando do menor nível de complexidade. Por sua vez, para o desenvolvimento, foi adotada uma estratégia baseada no modelo ágil, na qual foram definidas entregas semanais e tal demanda era influenciada pela elaboração do documento de planejamento do jogo. Após o fim desta etapa, foi realizada uma avaliação experimental com alunos de duas disciplinas, apresentada na Seção 5.

## 4.2 Descrição do Jogo

ProgramSE é um jogo no estilo *point & click*, ou seja, a sua interação ocorre através do ponteiro e do clique do *mouse* e é baseado em jogos do tipo *Escape the Room*, onde o objetivo principal é encontrar uma maneira de escapar de um lugar (e.g., um quarto, uma sala ou uma prisão) utilizando os objetos do cenário. Como o público alvo foi definido para adultos jovens, a motivação para desenvolver o enredo do jogo se deu pela leitura de uma obra literária intitulada "A Mulher na Janela" (Finn, 2018) – um suspense psicológico em que a protagonista sofre de agorafobia pós-traumática (i.e., um transtorno de ansiedade que a impede de sair de casa por medo). No decorrer da trama, a personagem precisa revisitar o passado, em sessões de terapia, e se lembrar do acidente que a fez perder sua família (e a fez desenvolver essa fobia). No ProgramSE, a história conta um enredo no qual o personagem sofre de perda de memória e se encontra preso em uma casa que ele não reconhece. O jogador precisa completar as fases para chegar ao desfecho da história do jogo, similar à história do livro. Assim que os desafios são cumpridos com sucesso, a causa da amnésia do jogador vem à tona. O enredo foi definido estritamente como um elemento de jogo, servindo apenas para o seu engajamento.

A tela inicial do ProgramSE (menu) apresenta a opção "Jogar" que, ao ser clicada, dá início ao jogo. Em cada fase, a meta é clicar e interagir com os objetos dispostos nos cenários, onde são apresentados comandos de programação embaralhados e o estudante deve colocá-los em uma ordem logicamente correta para ganhar o item correspondente àquele elemento. Ou seja, ao acertar o comando de programação correspondente ao objeto "Armário", o jogador consegue o item "Chave". Dessa forma, o jogo propõe o ensino da lógica de programação por meio da programação em blocos, fazendo com que a construção do conhecimento seja mais engajadora e lúdica. À medida que o jogador avança nas fases, novos comandos de programação são apresentados em níveis de dificuldade mais elevados.

O roteiro do jogo é apresentado junto com a evolução do jogador, com a finalidade de nortear e esclarecer o enredo. São colocados elementos-surpresa escondidos nas fases que, quando encontrados, modificam o rumo da história. Se localizados, é apresentada a história completa que explica a amnésia do personagem. São *feedbacks* sonoros sobre o que aconteceu com o personagem, ajudando a criar o ambiente de suspense envolvente para que o jogador consiga compreender o que aconteceu. Caso os elementos-surpresa não sejam encontrados, a história final apresentada é vaga e não explica o que ocorreu com o personagem. Os elementos-surpresa não influenciam no conteúdo a ser aprendido pelo estudante (nem no fluxo do jogo), mas são componentes dinâmicos relacionados à narrativa da história. O jogo acaba quando o jogador conseguir concluir os desafios e sair da casa.

## 4.3 Níveis de Aprendizagem

O jogo foi desenvolvido visando englobar o conteúdo programático das disciplinas introdutórias de programação, sendo eles: (1) Sequência; (2) Seleção; (3) Repetição; (4) Função e Procedimento; (5) Recursividade; (6) Vetor e Matriz; e (7) Ordenação. A exposição de conceitos é dividida em três níveis, tendo cada um delas as suas próprias fases. A relação dos conceitos apresentados em cada parte do ProgramSE é apresentada na Tabela 3.

Tabela 3. Relação Nível x Fase x Conceitos do ProgramSE.

Nível	Fase	Conceito
Nível 1	Quarto	Sequência
	Corredor	Seleção (Se), Repetição (Enquanto)
	Banheiro	Seleção (Se), Repetição (Enquanto - Para)
	Sótão	Seleção (Se/Senão - Caso)
Nível 2	Escada	Repetição (Para/Passo)
	Cozinha	Seleção (Se), Repetição (Enquanto - Para), Modularização (Função - Procedimento)
	Sala de Estar	Modularização (Função - Procedimento), Estrutura de Dados (Vetor - Matriz), Ordenação
Nível 3	Hall de Entrada	Recursividade

**Nível 1:** O objetivo é iniciar o aprendiz no pensamento lógico, começando o estudo pelo conceito principal da programação, i.e., a estruturação do algoritmo (i.e., Sequência). No desenrolar do nível, novos conceitos são apresentados. O Nível 1 é o mais longo do jogo, porém, por se basear em aspectos da linha de aprendizagem de Bloom, é o menos complexo. A meta é fixar os conceitos iniciais da programação, fazendo com que o estudante adquira o senso crítico sobre a diferença do ‘pensamento do dia-a-dia’ para o pensamento da lógica de programação. Os outros conceitos apresentados neste nível servem para o aperfeiçoamento do algoritmo, sendo eles: (1) comandos de controle (e.g., Seleção) – estruturas condicionais que realizam testes lógicos antes de executar ações; e (2) comandos de repetição (e.g., Repetição) – estrutura que executa um bloco de ações repetidamente. A Figura 3 mostra as fases que compõem o Nível 1.

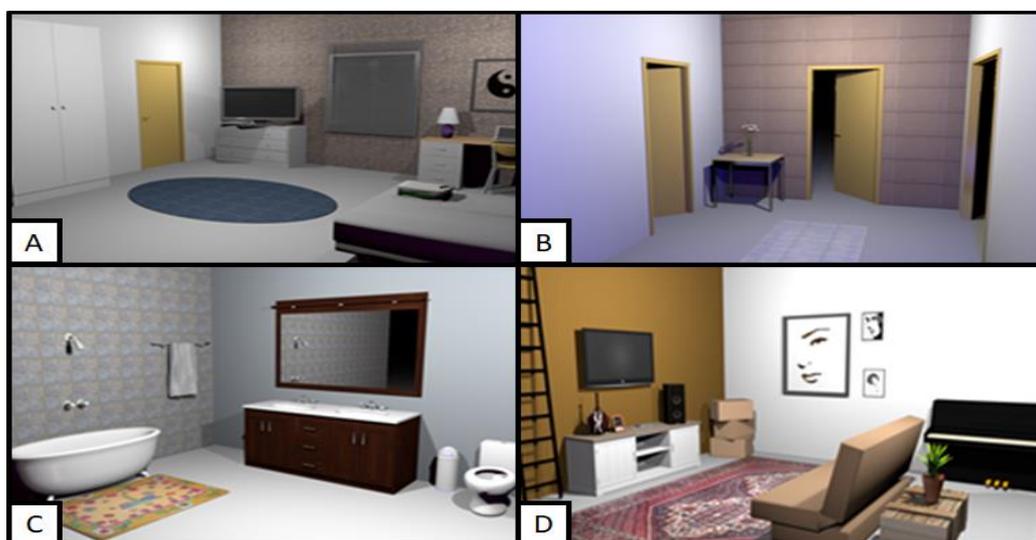


Figura 3: (a) Quarto; (b) Corredor; (c) Banheiro; e (d) Sótão.

**Nível 2:** Este nível visa apresentar ao estudante algumas estruturas de programação mais complexas. Por entender que, após o Nível 1, o estudante teve contato com a estruturação de alguns conceitos de programação, o Nível 2 foca em exibir novos comandos, sendo eles: (1) aperfeiçoamento de comandos de repetição; (2) modularização (e.g., Função) – blocos de comandos que, ao serem chamados, serão executados e retornarão algum resultado; (3) estruturas de dados (e.g., Vetor) – uma estrutura de dados unidimensional para armazenar elementos do mesmo tipo; e (4) algoritmos de ordenação – processo de rearranjo de um certo conjunto de objetos de acordo com um critério. Por serem comandos de complexidade moderada, o Nível 2 conta com novas mecânicas introduzidas ao jogo, i.e., a interação com os comandos de Vetor e Matriz é liberada após a leitura dos conceitos relacionados. O principal objetivo é capacitar o estudante a, além de saber como programar tais estruturas, entender ludicamente como é sua representação. A Figura 4 apresenta os cenários das fases do Nível 2 (A, B e C).

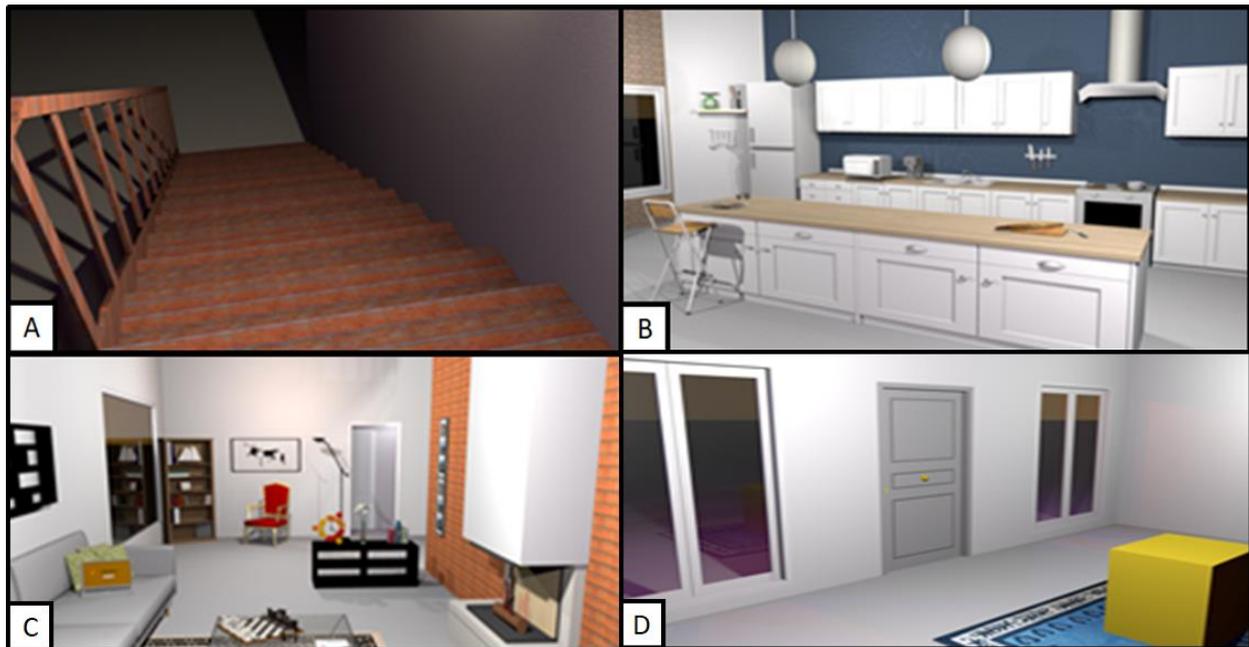


Figura 4: (a) Escada; (b) Cozinha; (c) Sala de Estar; e (d) Hall de Entrada.

**Nível 3:** Nesse nível, é apresentado um conceito novo: (1) recursividade, i.e., uma função que contenha uma chamada a si própria. Por ser o conceito mais complexo de ensinar, foi apresentada isolada no nível de maior dificuldade do jogo. Foi criada uma dinâmica diferente dos demais níveis, onde o estudante fica preso em uma sala e toda vez que clica para sair, volta para o mesmo cômodo. Dessa forma, precisa-se encontrar uma condição que pare a recursividade. Após a conclusão desta fase, o jogador é apresentado ao desfecho da narrativa e o jogo termina. A Figura 4 (D) mostra a única fase do Nível 3.

#### 4.4 Desafios

Os desafios apresentados no jogo são baseados na programação em blocos, como acontece no *Scratch* e *Blockly*. A motivação para apresentá-los desta forma foi buscar por um aprendizado divertido e de fácil compreensão dado que, ao invés de escrever um comando de programação, o usuário necessita apenas conectar blocos de maneira lógica (Majed, 2014). Como a proposta do jogo é interagir com o ambiente para poder avançar, os blocos de programação são apresentados ao jogador no momento em que algum objeto do cenário é clicado. Blocos com frases em português que simulam comandos de programação aparecem embaralhados na tela e o estudante deve ordená-los de forma logicamente correta para que conclua o desafio. Em algumas interações, o jogador pode receber itens colecionáveis, que são armazenados em um inventário e que serão necessários para objetos futuros.

Na Figura 5, um exemplo de uma interação no Nível 1 apresenta o conceito de Sequência. Os elementos dispostos no momento da interação são: os blocos de programação (à esquerda); os guias da programação, ou seja, os locais em que o jogador deve arrastar e encaixar os blocos (no centro); e a dica para ajudar no processo (embaixo). Além desses elementos, o cenário também apresenta um contador de itens restantes da fase (canto superior esquerdo) e uma mochila para acessar o inventário (canto inferior direito). Espera-se que, ao final do desafio, o estudante possa entender melhor a definição apresentada naquele momento. Como os conceitos são repetidos ao longo do jogo, o jogador pode aprender, memorizar e aplicar o seu conhecimento em outros objetos. Os blocos de programação são móveis, ou seja, o jogador deve arrastá-los até a posição correta do algoritmo (representado pelos guias da programação). Quando o estudante arrastar o bloco para o local certo, o mesmo fica imóvel e não pode mais ser arrastado. Caso o bloco seja

arrastado para o local errado do algoritmo, volta à posição inicial (à esquerda) e o jogador tem que repetir o processo até acertar a posição de todos os blocos.

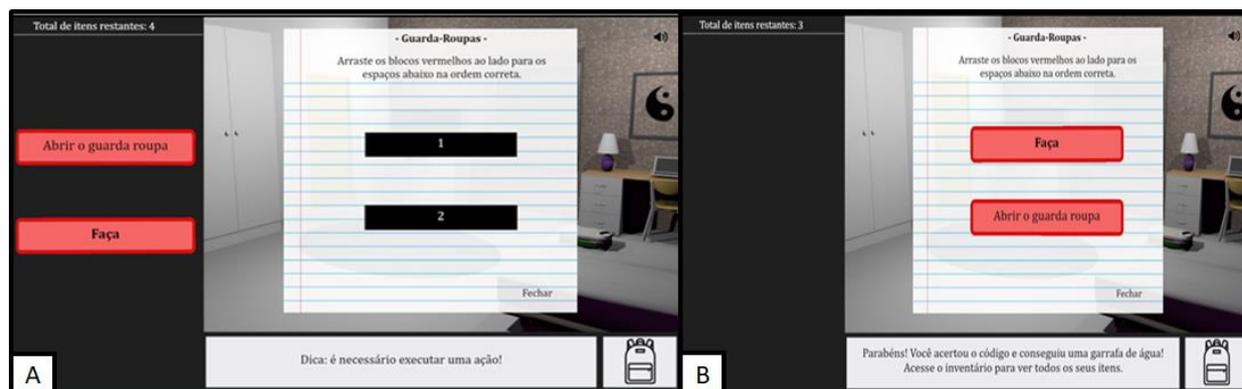


Figura 5: Exemplo de apresentação de conceito de Sequência, sendo (A) antes e (B) depois de mover comandos.

Os desafios são progressivos, ou seja, a sua complexidade vai aumentando ao longo do jogo. A decisão de tornar a complexidade progressiva foi tomada a fim de permitir que o estudante se exponha a maiores desafios aos poucos, para evitar frustrações. Para auxiliar na fixação dos conceitos de programação, o jogo permite que o estudante leia dicas conforme um assunto que é apresentado. Os conceitos apresentados no jogo (Sequência, Seleção, Repetição, Modularização, Estrutura de Dados, Ordenação e Recursividade), o *link* do jogo, a forma como são apresentados no jogo e suas respectivas dicas são apresentados no Apêndice I.

## 5 Avaliação Experimental

Após a construção do ProgramSE, a sua avaliação foi realizada por meio de um experimento a fim de verificar se os objetivos foram atingidos com o seu uso no contexto de sala de aula. O processo de avaliação seguiu as recomendações de Wohlin *et al.* (2012). Com a aplicação e uso do ProgramSE, foi possível avaliar a sua usabilidade e efetividade baseada na experiência do jogador, além de ser possível verificar se decisões de planejamento a respeito dos elementos do jogo foram satisfatórias sob o ponto de vista dos estudantes.

### 5.1 Planejamento

Após o desenvolvimento do ProgramSE, o questionário do modelo de avaliação MEEGA+ (Petri *et al.*, 2019), voltado para jogos educacionais digitais, foi customizado. O questionário final foi composto por vinte e nove (29) questões. Suas respostas são padronizadas, variando em uma escala de -2 (discordo fortemente) até +2 (concordo fortemente). Assim, quando a maioria das respostas de determinada característica avaliada se aproximar de +2, significa que o tópico obteve uma avaliação positiva; caso contrário, ao se aproximar de -2, uma avaliação negativa. Foi criada uma versão digital do questionário utilizando o *Google Forms* e acrescentado um termo de consentimento livre e esclarecido. Por ser um jogo *single player*, o jogo não atende aos requisitos de interação social do modelo do questionário. Portanto, não foram incluídas na análise as perguntas associadas à interação dos participantes com outros jogadores.

Foi realizado um piloto da avaliação com duas pessoas, uma graduanda e uma mestranda que é professora da área de informática, contemplando as duas visões do MEEGA+ (estudante e instrutor). Ambas responderam o questionário MEEGA+ após jogarem o ProgramSE e suas respostas fomentaram discussões sobre os conceitos escolhidos para a ferramenta e sobre alguns elementos de jogos (e.g., enredo e *feedbacks*). O objetivo de ambas preencherem o questionário MEEGA+ foi coletar opiniões e sugestões de melhoria do estudo experimental antes da execução. As principais mudanças no fluxo do jogo realizadas após as sugestões do piloto foram: (1) mudar

o *feedback* de objetos que não possuem itens ao acertar o desafio; (2) melhorar o *feedback* sobre itens utilizados nos desafios; (3) incluir textos informativos sobre qual objeto o jogador está realizando o desafio; (4) mudar a ação do objeto Sótão - TV; e (5) mudar o termo técnico "Passo" para "Variando" nos exemplos do comando de repetição Para. Com as recomendações do piloto, o jogo pode ser refinado para o experimento.

O estudo com os estudantes ocorreu em maio de 2019, sendo o ProgramSE adotado na turma da disciplina de Técnicas de Programação I do curso de Sistemas de Informação (SIS), integral, e na turma da disciplina de Introdução à Ciência da Computação do curso de Licenciatura em Matemática (MAT), noturno, da Universidade Federal do Estado do Rio de Janeiro. Ambas as disciplinas são lecionadas presencialmente, por professores diferentes com experiência de, pelo menos, 15 anos no ensino de programação para os iniciantes de seus respectivos cursos. Por sua vez, os estudantes estavam na metade do primeiro período dos respectivos cursos, sendo que as turmas tinham alguns alunos reprovados em semestres anteriores. Os estudantes possuem diferentes perfis, visto que a turma de MAT é formada por alunos que não focam em computação. A escolha dessa turma para a aplicação do experimento foi importante para avaliar a usabilidade e efetividade baseada na experiência do jogador com uso do ProgramSE por estudantes que não têm a computação como fim, ou seja, alunos provenientes de outras áreas de conhecimento. A avaliação foi mediada por um pesquisador e acompanhada por dois instrutores (professores) da área de informática, que focaram no acompanhamento dos estudantes durante a execução do estudo. Os estudantes realizaram as atividades no jogo (completa ou parcialmente, a depender do progresso) e, em seguida, responderam ao questionário MEEGA+.

## 5.2 Execução

Após o planejamento, a avaliação do ProgramSE ocorreu no dia 09 de maio de 2019 e contou com a participação de 37 alunos, sendo 23 da disciplina de SIS e 14 de MAT. Cada avaliação durou aproximadamente 45 minutos, sendo reservado no início um tempo para falar sobre o projeto, as regras do jogo e seus componentes. Ao término do período da avaliação, 32 estudantes responderam ao questionário. Desses, 18 são de SIS e 14 de MAT. Considerando que o preenchimento do questionário era de livre e espontânea vontade, ou seja, opcional, alguns discentes preferiram não responder, sem nenhuma penalização.

No início da avaliação, o objetivo geral do trabalho foi apresentado, uma explicação sobre o jogo foi realizada e os estudantes foram orientados sobre o processo. Em seguida, foi disponibilizado o *link* para o formulário *online*, dividido em quatro etapas. A primeira etapa é o Termo de Consentimento Livre e Esclarecido. Nele, o estudante concorda ou não em fazer parte da pesquisa. O termo explica todo o processo da avaliação, a confidencialidade dos dados obtidos e a liberdade de desistência dos participantes. A segunda etapa consiste em perguntas sobre informações demográficas, i.e., faixa etária e sexo. A frequência com que o estudante costuma jogar também é importante e foi perguntada nesse passo. A terceira etapa é realizada após a utilização do jogo, por se tratar do questionário para a avaliação da qualidade do jogo. As perguntas são agrupadas em Usabilidade e Experiência do Jogador. Por último, a quarta etapa possui questões discursivas para que o estudante possa dar sugestões e/ou críticas para aprimorar o jogo. A Figura 6 traz duas fotos tiradas nos momentos da execução da avaliação experimental.



Figura 6: (A) Avaliação em SIS; e (B) Avaliação em MAT.

O tempo da avaliação foi cronometrado a partir da disponibilização do *link* para o questionário *online* até que o último aluno enviasse suas respostas. Dessa forma, a avaliação em SIS ocorreu das 17h27min até às 18h08min, durando 41 minutos. Enquanto isso, para MAT, a avaliação durou 49 minutos, ocorrendo das 20h47min às 21h36min. Ambas as avaliações foram feitas nos laboratórios de programação da universidade.

### 5.3 Resultados

Com todos os dados coletados, iniciou-se a etapa de análise. As respostas foram processadas e alguns gráficos foram gerados para identificar acertos e pontos de melhorias do jogo. A análise dos dados foi feita de maneira separada para cada um dos grupos e utilizou a técnica de *design* de experimentos denominada bloco (Wöhlin *et al.*, 2012). O bloco é aplicado para sistematicamente eliminar o efeito indesejado na comparação entre tratamentos. Em um bloco, o efeito indesejado é o mesmo e pode-se estudar o efeito dos tratamentos naquele bloco. Além disso, essa técnica aumenta a precisão do experimento, considerando que, na avaliação do jogo, os participantes têm experiências diferentes e cursos com focos diferentes (computação como meio e como fim). Dessa forma, buscou-se minimizar estes efeitos, agrupando os participantes em dois grupos (blocos).

**Turma SIS:** Na disciplina de técnicas de programação, 18 alunos participaram da pesquisa e responderam ao questionário. Como a primeira etapa do estudo envolveu perguntas para entender o perfil demográfico da turma, a Figura 7 exibe os gráficos obtidos. Pode-se perceber que 15 participantes correspondem à faixa etária alvo deste trabalho, de 18 a 28 anos. Do restante, 2 pessoas são menores de 18 anos e 1 pessoa, de 29 a 39 anos. A segunda análise demográfica é em relação ao sexo dos participantes. Os participantes do gênero masculino totalizam 14 pessoas. Foram 4 pessoas do gênero feminino.

Os dois últimos gráficos são referentes à frequência do estudante em utilizar jogos digitais ou não-digitais. Essa frequência pode variar entre: (1) diariamente; (2) semanalmente; (3) mensalmente; (4) raramente; e (5) nunca. Sobre o costume de jogar jogos digitais, 11 pessoas relataram usar diariamente; 4 pessoas usam semanalmente; nenhuma pessoa selecionou a opção mensalmente; 2 pessoas usam raramente; e 1 pessoa não utiliza jogos digitais. Em relação ao uso de jogos não-digitais, os resultados foram diferentes. Nenhuma pessoa da amostra utiliza diariamente; 1 pessoa faz uso semanal; 6 pessoas utilizam jogos não-digitais mensalmente; 8 pessoas usam raramente; e 3 pessoas nunca utilizam esses tipos de jogos.

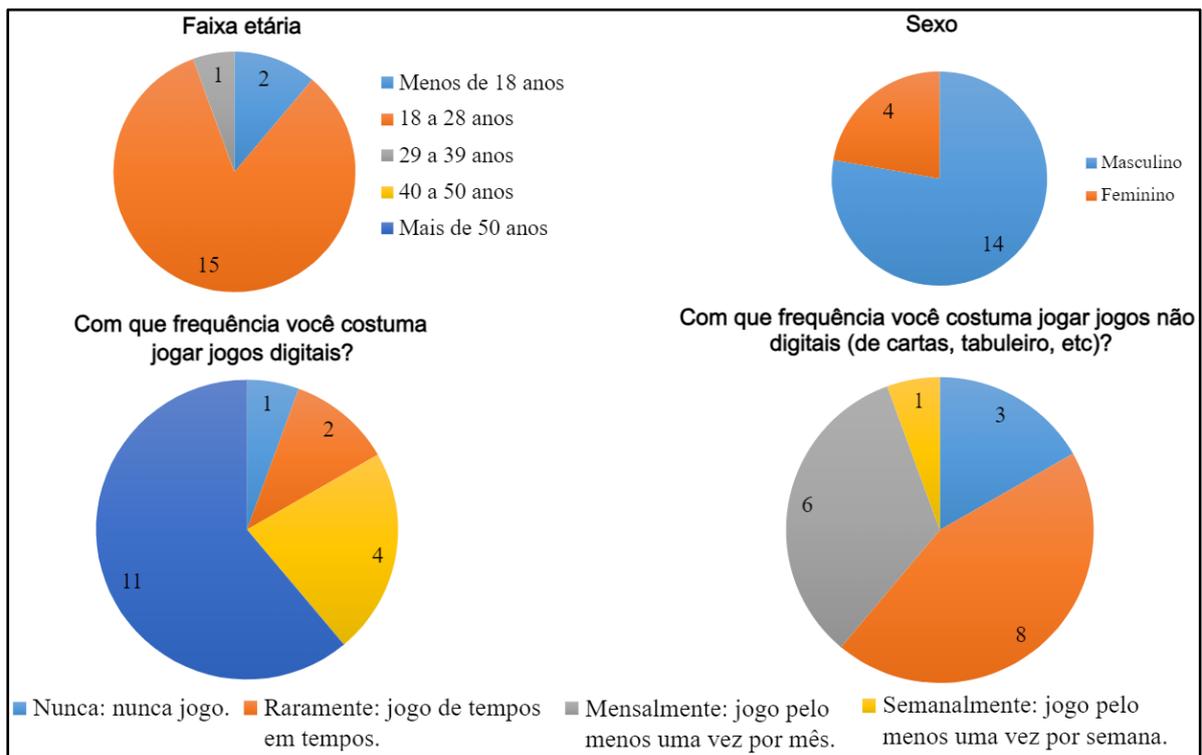


Figura 7: Informações demográficas (Turma SIS).

A próxima etapa do questionário corresponde a questões sobre a usabilidade do jogo. Ou seja, aspectos visuais e princípios percebidos na interação com ProgramSE. No questionário, as afirmações foram agrupadas nos seguintes aspectos: Estética, Aprendizabilidade, Operabilidade e Acessibilidade. A Figura 8 mostra o gráfico obtido das respostas sobre o tema. Uma análise do gráfico permite inferir que todos os aspectos avaliados obtiveram um resultado satisfatório. Como todas as respostas tendem a algum nível de concordância mais do que discordância, em termos de usabilidade, ProgramSE obteve retorno positivo dos alunos de SIS. Isto revela que os participantes se sentiram confortáveis em utilizar o jogo.

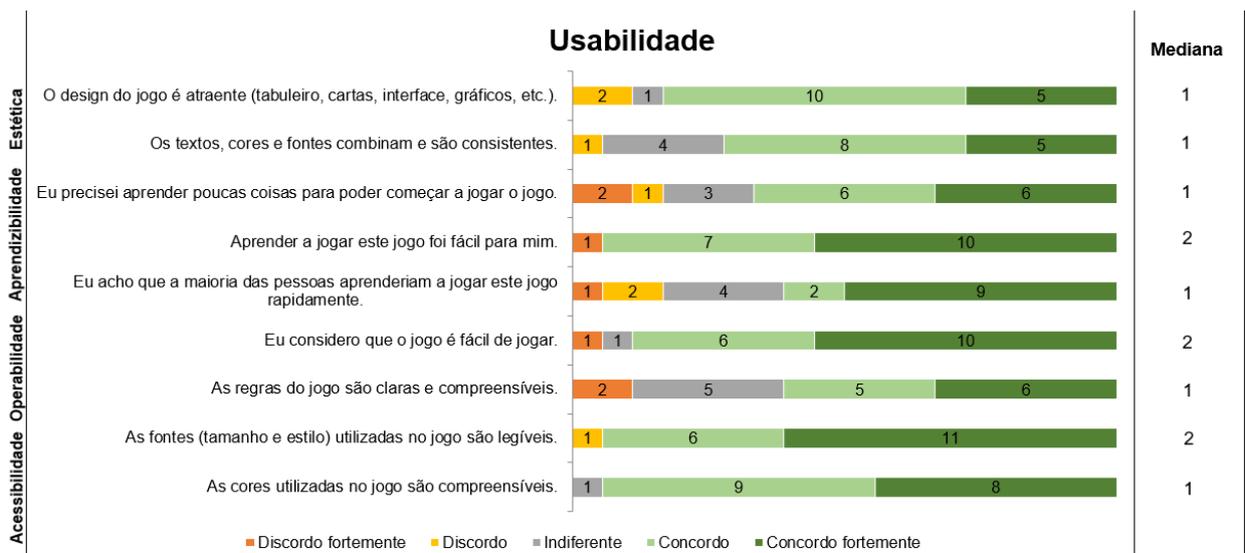


Figura 8: Usabilidade do ProgramSE (Turma SIS).

Para entender melhor a opinião dos participantes, também foi feita uma análise quantitativa por cada aspecto dentro de cada fator (usabilidade e experiência do jogador). No fator usabilidade – aspecto acessibilidade, por exemplo, foram considerados os itens relacionados a fontes e cores, analisando a mediana de cada item e, neste aspecto, contabilizando a quantidade de participantes

que indicaram algum nível de concordância (concordo ou concordo fortemente) e de discordância (discordo ou discordo fortemente), gerando o percentual de participantes. Com base nos resultados, foi possível ter uma ideia geral do grau de concordância dos participantes do estudo com relação aos itens avaliados.

O aspecto de Acessibilidade teve a avaliação mais alta, com maior parte das respostas com algum nível de concordância (+1 ou +2). Pode-se concluir que as fontes e as cores utilizadas contribuíram para melhorar a experiência do jogador. Vale destacar que não houve participantes com daltonismo ou outra deficiência visual nesta avaliação em particular. Além disso, o aspecto de Aprendizizibilidade, relacionado a aprender e jogar, também obteve uma das avaliações mais altas. No entanto, este aspecto obteve ainda algum nível de discordância ou indiferença maior em relação aos demais. Em relação aos aspectos restantes, a Estética obteve 50% de nível de concordância (+1) entre as respostas e 28% de forte concordância (+2). O aspecto Operabilidade obteve 30% de nível de concordância (+1) e 44% de forte concordância (+2). A Estética do jogo diz respeito ao *design* e a Operabilidade trata a clareza quanto às regras e funcionalidades do jogo.

Outros aspectos avaliados se referiam à experiência do jogador (i.e., confiança, desafio, satisfação, diversão, atenção focada, relevância e percepção de aprendizagem), como mostra a Figura 9. A avaliação sobre a experiência dos jogadores obteve, no geral, algum nível de concordância (+1) e, em algumas afirmativas, prevaleceu a neutralidade dos estudantes (0). Nesse gráfico, não foram encontradas respostas tendendo ao nível de forte concordância (+2). Porém, também não foram observadas avaliações negativas (-1 ou -2). No geral, as avaliações positivas das experiências dos jogadores foram menores que aquelas da usabilidade.

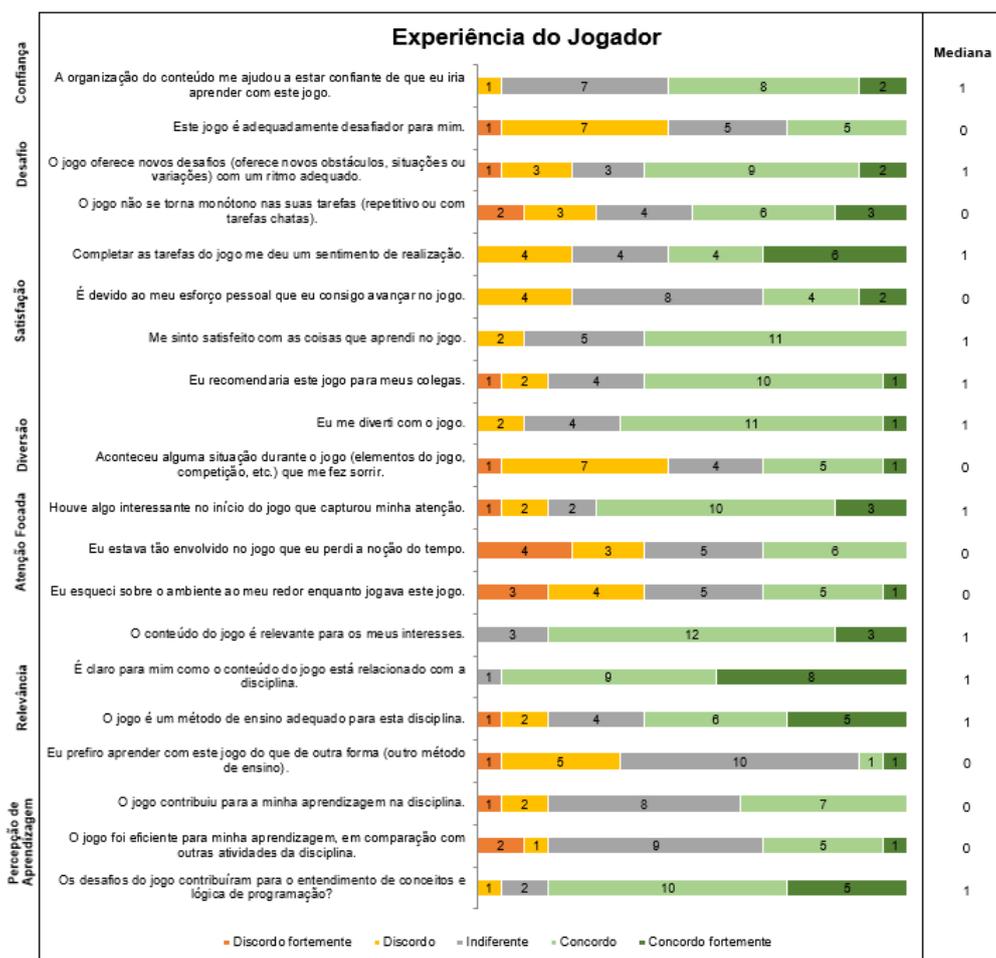


Figura 9: Experiência do jogador com ProgramSE (Turma SIS).

O aspecto com a pontuação mais alta é a Relevância do jogo, com 37% de concordância (+1) e 22% de forte concordância (+2). O tópico se refere à relevância do conteúdo apresentado no ProgramSE para os interesses do aluno e da disciplina. Os aspectos com menos notas positivas são o Desafio e a Atenção Focada. A maior parte dos participantes não achou o jogo desafiador e não se sentiu tão engajada no enredo do jogo. Os demais aspectos tiveram um maior nível de concordância (+1) e indiferença (0): Confiança, Satisfação, Percepção de Aprendizagem e Diversão. Ressalta-se que a afirmação “*Os desafios do jogo contribuíram para o entendimento de conceitos e lógica de programação*” está incluída no aspecto Percepção de Aprendizagem, obtendo mediana 1, sendo 55% de concordância (+1) e 28% de forte concordância (+2).

Os participantes também apresentaram sugestões e/ou melhorias que eles identificaram durante o avanço no jogo. Essa etapa consistiu na quarta etapa do questionário (questões discursivas). Essas informações foram processadas e consolidadas no gráfico da Figura 10. Entre os pontos fortes, o mais citado entre os participantes foi em relação à Jogabilidade, i.e., o fluxo de progressão do jogo, com 13 comentários feitos. Os outros tópicos que surgiram dentre os pontos fortes foram o *Design* do Jogo, com 11 comentários, e a Fundamentação do Jogo, i.e., a formatação e a forma como os conceitos de programação foram ensinados (e.g., comandos e dicas), com um comentário. Como pontos a melhorar, novos temas foram apresentados. O mais comentado foi a Jogabilidade, com 8 sugestões. O *Design* do Jogo e a Fundamentação do Jogo também aparecem nessa parte, com 3 e 5 comentários, respectivamente. Os novos aspectos comentados foram: Monótono (i.e., o jogo se tornou monótono em determinado momento), com 4 comentários; Adição de Novidades (i.e., novas fases e novas dinâmicas no jogo), com 3 comentários; e, por último, Melhoria de Tecnologia, com 2 comentários.

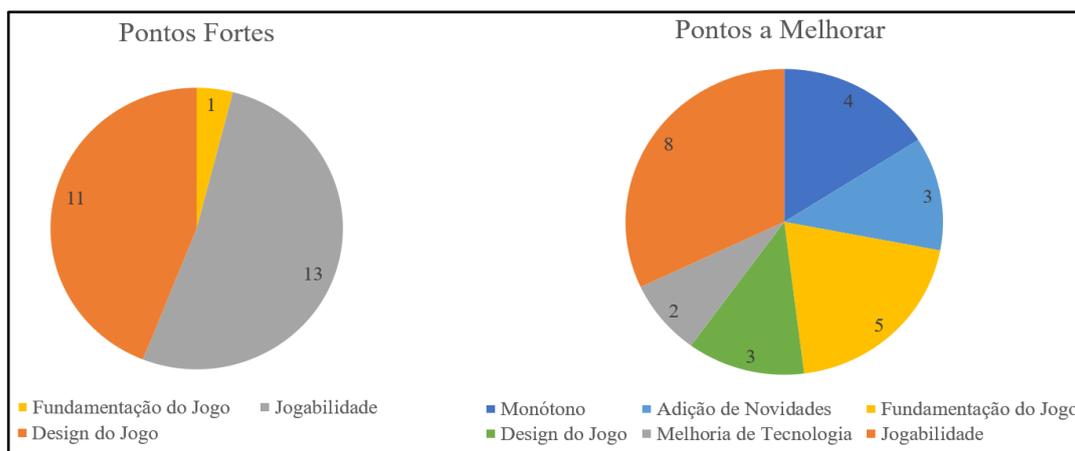


Figura 10: Questões discursivas sobre ProgramSE (Turma SIS).

**Grupo MAT:** Como a avaliação do ProgramSE foi feita em duas disciplinas diferentes, a segunda parte da análise de dados é referente aos 14 alunos de introdução à computação. O processo ocorreu da mesma forma que em SIS e a Figura 11 apresenta o gráfico gerado sobre as informações demográficas da turma. A faixa etária dos participantes se concentra entre os alunos de 18 a 28 anos, o público alvo do trabalho, sendo 9 pessoas do total. Havia apenas 1 pessoa dentre os menores de 18 anos, uma de 29 a 39 anos e 1 maior de 50 anos. Por fim, 2 pessoas tinham idade entre 40 e 50 anos. A segunda análise feita é referente ao sexo dos participantes. Os participantes do gênero masculino totalizam 13 pessoas e havia 1 pessoa do gênero feminino.

Os dois últimos gráficos, assim como em SIS, também são referentes à frequência do estudante sobre a utilização de jogos digitais ou não-digitais. Essa frequência varia entre: (1) diariamente; (2) semanalmente; (3) mensalmente; (4) raramente; e (5) nunca. Sobre o costume de usar jogos digitais, 6 pessoas relataram utilizar diariamente; 3 pessoas usam semanalmente; nenhuma pessoa selecionou a opção mensalmente; 4 pessoas raramente utilizam; e 1 pessoa não utiliza jogos digitais. Em relação ao uso de jogos não-digitais, os resultados foram diferentes.

Nenhuma pessoa da amostra utiliza diariamente; uma pessoa faz uso semanal; 6 pessoas utilizam jogos não-digitais mensalmente; 7 pessoas raramente utilizam esses tipos de jogos; e ninguém selecionou a opção nunca.

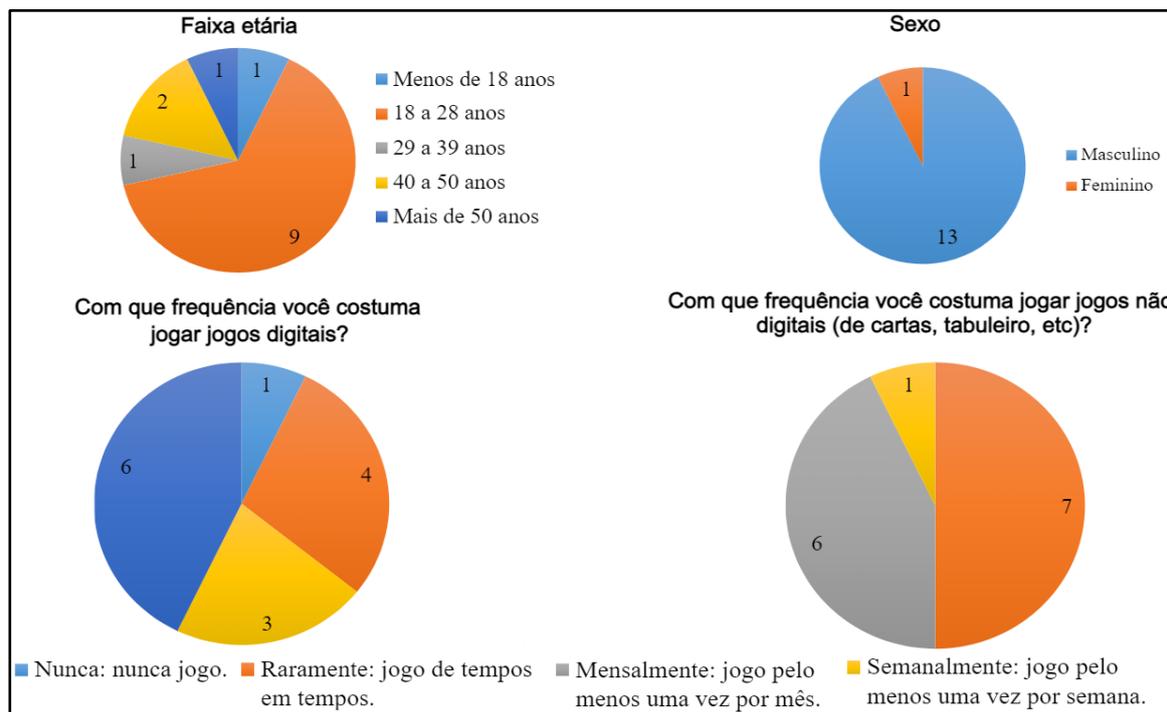


Figura 11: Informações demográficas (Turma MAT).

A próxima etapa do questionário corresponde a questões sobre a usabilidade do jogo. A Figura 12 mostra o gráfico das respostas relacionadas. Analisando-se o gráfico, pode-se inferir que todos os aspectos avaliados também obtiveram um bom resultado. Como todas as respostas tendem a algum nível de concordância mais do que discordância, em termos de usabilidade o jogo ProgramSE também obteve retorno positivo dos alunos de MAT. Isto revela que os participantes se sentiram confortáveis em utilizar o jogo.

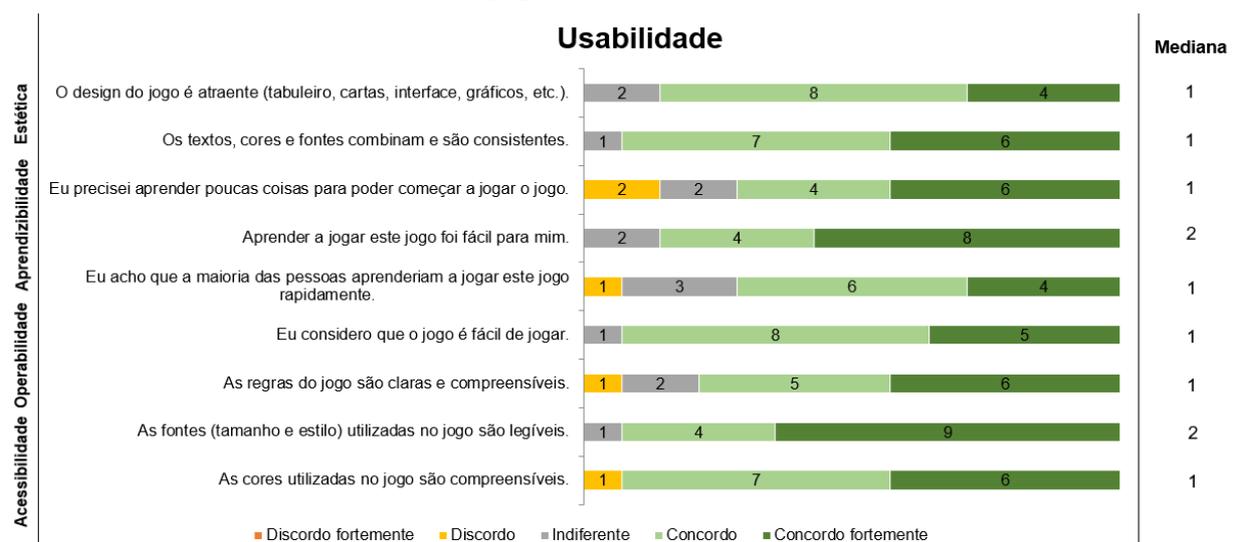


Figura 12: Usabilidade do ProgramSE (Turma MAT).

O aspecto de Acessibilidade também teve a avaliação mais alta para essa turma, com a maior parte das respostas com algum nível de concordância (+1 ou +2). Pode-se reforçar que as fontes e as cores utilizadas contribuíram para melhorar a experiência do jogador. O aspecto que obteve menor nível de concordância nos itens avaliados foi a Aprendizabilidade, referentes à

facilidade de aprender a jogar. Em relação aos aspectos restantes, a Estética obteve 42% de nível de concordância (+1) entre as respostas e 28% de forte concordância (+2). O aspecto Operabilidade obteve 36% de nível de concordância (+1) e 30% de forte concordância (+2). Em comparação com a turma SIS, a quantidade de avaliações positivas sobre a Usabilidade do ProgramSE foi maior na turma MAT. Outros aspectos avaliados contemplaram a experiência do jogador, como mostra a Figura 13. A avaliação sobre a experiência dos jogadores da turma MAT obteve, no geral, algum nível de concordância (+1) e, em algumas afirmativas, prevaleceu a neutralidade (0). Nesse gráfico, foram encontradas respostas tendendo a algum nível de concordância (+1 e +2) e também sem avaliações negativas (-1 ou -2). As avaliações positivas das experiências dos jogadores foram menores que as apresentadas na usabilidade.

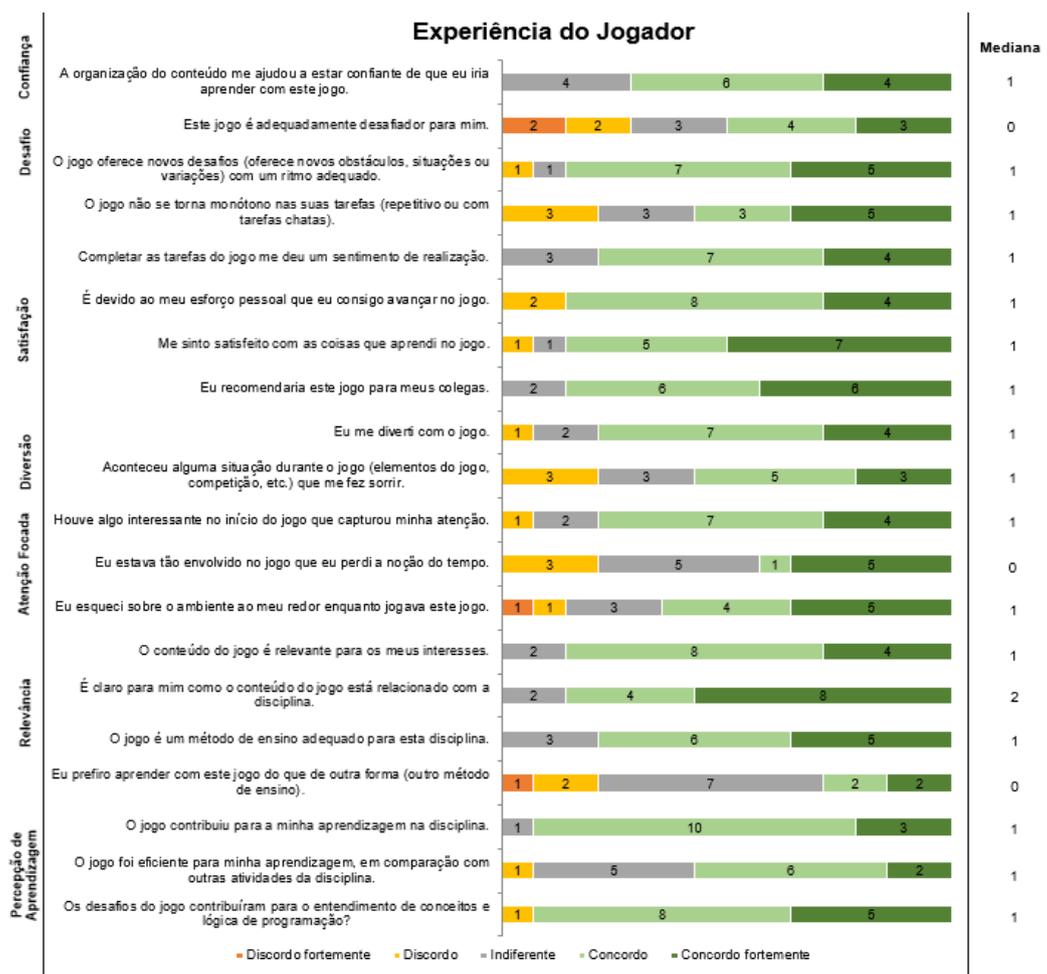


Figura 13: Experiência do jogador com o ProgramSE (Turma MAT).

O aspecto com a pontuação mais alta é a Satisfação do jogo, com 36% de concordância (+1) e 29% de forte concordância (+2). O aspecto com menos notas positivas foi a Atenção Focada. A maior parte dos participantes não se sentiram tão engajados no enredo do jogo. Os demais aspectos tiveram um maior nível de concordância (+1) e indiferença (0): Percepção de Aprendizagem, Confiança, Relevância, Diversão e Desafio. Vale ressaltar que a afirmação “Os desafios do jogo contribuíram para o entendimento de conceitos e lógica de programação”, incluída no aspecto Percepção de Aprendizagem, obteve mediana 1, sendo 45% de concordância (+1) e 28% de forte concordância (+2). Os participantes também apresentaram sugestões e/ou melhorias que eles identificaram para o progresso do jogo. Essa etapa consistiu na quarta etapa do questionário (questões discursivas). Essas informações são mostradas na Figura 14.

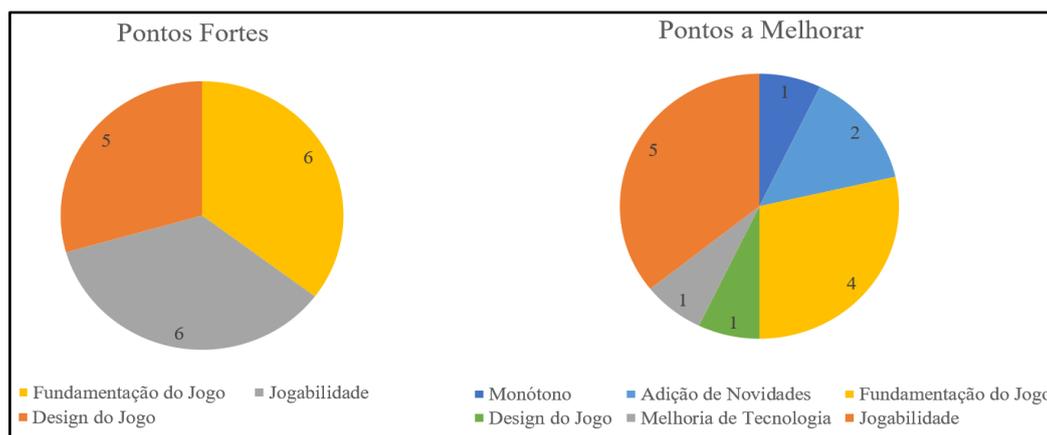


Figura 14: Questões discursivas sobre o ProgramSE (Turma MAT).

Entre os pontos fortes, o mais citado entre os participantes foi em relação à Jogabilidade, sendo 6 comentários feitos. O mesmo ocorreu para a Fundamentação do Jogo. Os outros tópicos que surgiram dentre os pontos fortes foram relativos ao *Design* do Jogo, com 5 comentários. Como pontos a melhorar, novos temas foram apresentados. O mais comentado também foi a Jogabilidade, com 5 sugestões. O *Design* do Jogo e a Fundamentação do Jogo também foram destacados, com aproximadamente 1 e 4 comentários, respectivamente. Vale ressaltar que, entre os comentários sobre a melhora na fundamentação, estão críticas às dicas dispostas, por estarem vagas. Os novos aspectos comentados foram: Monótono, com um comentário; Adição de Novidades, com 2 comentários; e, por último, Melhoria de Tecnologia, com um comentário.

#### 5.4 Discussão

A partir dos resultados e percepções obtidos na avaliação experimental do jogo, é possível discutir pontos importantes que estão relacionados tanto ao objetivo do jogo quanto a discussões relacionadas ao cenário do ensino em geral. A discussão dos resultados é apresentada nesta subseção. Inicialmente, em ambas as disciplinas, a presença do gênero masculino é majoritariamente maior que o feminino, que são apenas 15% do total de participantes. Por serem turmas de primeiro período dos cursos de graduação, pode-se entender que ainda faltam estímulos para as mulheres ingressarem e investirem na área. Segundo Adriana Tonini (diretora de Engenharias, Ciências Exatas, Humanas do CNPq em 2018), o Brasil carece de incentivos às meninas de 6 e 7 anos para estudarem Matemática e Exatas<sup>5</sup>. A afirmação pode ser confirmada por dados apresentados por Tonini, que informa que a participação das pesquisadoras mulheres é menor que 25% em publicações de áreas como Computação e Matemática.

Em relação à faixa etária dos avaliadores, 75% correspondem diretamente ao público alvo definido para este trabalho. Por serem turmas de iniciantes, pode-se aferir que a maioria dos ingressantes no nível superior destes cursos, Sistemas de Informação e Licenciatura em Matemática, são adultos jovens. Comparando os alunos entre as turmas em questão, as idades da turma MAT são mais diversificadas do que na turma SIS, tendo a última um perfil mais jovial (menores de 28 anos). Os participantes da turma MAT também são em sua maioria jovens, porém existem alunos acima dos 40 anos (que não há na turma SIS). Marquez e Silva (2017) observaram uma ascensão maior do público aluno-trabalhador ao ensino superior noturno público. Como o curso de Sistemas de Informação é integral e o de Licenciatura em Matemática é noturno, pode ser esse o motivo de haver alunos mais velhos na disciplina de introdução à computação da turma MAT, pela possibilidade de conciliar o trabalho com os estudos.

<sup>5</sup> <http://www.ufes.br/conteudo/cnpq-investe-em-projetos-que-estimulem-presenca-de-meninas-nas-ciencias-exatas>

Nenhum dos participantes da avaliação utilizam jogos não-digitais diariamente e, na turma SIS, houve alunos que selecionaram que nunca os jogam. Em contrapartida, 53% do total de estudantes utilizam jogos digitais todos os dias e 21% os utilizam semanalmente. Com base no modelo de Cruz (2012), existem relações entre jovens e o envolvimento em jogos e, como a faixa etária é de 75% adulto jovem, explica-se o grande uso de jogos digitais. Em cada uma das disciplinas, houve uma pessoa que selecionou que nunca utilizam jogos digitais. O aluno da turma SIS possui de 29 a 39 anos e o da turma MAT possui de 40 a 50 anos. Os dois responderam que ProgramSE foi eficaz e concordaram que os desafios do jogo contribuíram para o entendimento de conceitos e lógica de programação. Em relação à Aprendizibilidade do jogo, suas respostas indicaram que tiveram que aprender poucas coisas para jogar e que a jogabilidade foi fácil. Vale ressaltar que ProgramSE foi construído com base em aspectos de usabilidade (e.g., aprendizagem e memorização) e sua efetividade é indicada quando pessoas que não têm contato com jogos digitais conseguem se sentir seguros e confiantes para aprender e encarar os desafios propostos.

Os aspectos correspondentes à usabilidade do ProgramSE tiveram, em geral, muitas avaliações positivas. Pode-se aferir que o jogo agradou a maioria dos alunos em termos da interação humano-computador, sendo todas as avaliações entre +1 e +2. Vale ressaltar que as respostas mais altas (+2) em ambas as disciplinas foram em relação a Acessibilidade e Aprendizibilidade nas questões *“As fontes utilizadas no jogo são legíveis”* e *“Aprender a jogar este jogo foi fácil para mim”*. As questões referentes à usabilidade do jogo foram planejadas durante a discussão dos resultados da RSL realizada. Como o tema não foi muito abordado entre os artigos analisados, isso se fez importante para a concretização do objetivo principal do jogo. Outra premissa que obteve muitas respostas positivas foi *“Eu considero que o jogo é fácil de jogar”*. Porém, a mediana da turma SIS foi maior que a da turma MAT, respectivamente, +2 e +1. O fato da primeira turma ter uma quantidade de alunos que utilizam jogos digitais diariamente maior que a segunda pode justificar a diferença entre as avaliações, mesmo que sutil. Além disso, pode-se destacar ainda o efeito do conhecimento/interesse prévio em assuntos relacionados à computação e ao enfoque do curso (computação como meio ou como fim). Em contrapartida, as avaliações referentes ao desafio proposto tiveram mais respostas positivas na turma MAT do que na turma SIS. Elas avaliam se o jogo é desafiador e, como na turma SIS, os alunos julgaram o jogo como fácil de ser jogado, este fato pode justificar a falta de momentos desafiadores no jogo.

As avaliações sobre a experiência dos jogadores são as mais importantes de serem interpretadas, pois determinam a efetividade e relevância do jogo digital avaliado em relação ao conteúdo ensinado. Além disso, as reações dos estudantes enquanto jogadores durante a avaliação podem reforçar a percepção de quanto eles ficaram satisfeitos ao jogar ProgramSE. No geral, os aspectos definidos nessa etapa obtiveram um alto nível de concordância (+1 ou +2). Foram observadas muitas respostas indiferentes (0), o que diminui as medianas (porém, não podem ser consideradas negativas). Uma possível razão para isso pode ser o fato de que os participantes são iniciantes em programação e estavam no meio do primeiro período dos cursos. O aspecto *“Os desafios do jogo contribuíram para o entendimento de conceitos e lógica de programação?”* obteve 87,5% de avaliações positivas. Ou seja, o jogo obteve um retorno otimista por parte dos participantes em relação ao aprendizado. Neste caso, o jogo, que teve como objetivo auxiliar a aprendizagem de estudantes iniciantes em programação, pode ser uma ferramenta promissora.

Dentre as premissas, os questionamentos *“Este jogo é adequadamente desafiador para mim”*, *“Eu estava tão envolvido no jogo que eu perdi a noção do tempo”* e *“Eu prefiro aprender com este jogo a outra forma (outro método de ensino)”* tiveram mediana 0 em ambas as turmas. Possivelmente, isso se deve à grande quantidade de alunos que jogam diariamente, de modo que as expectativas para o jogo não foram correspondidas nesses aspectos citados. Ou seja, o conteúdo foi apresentado de uma maneira lúdica e interessante, mas, em termos de diversão, imersão no jogo e desafio proposto, ProgramSE não satisfaz os alunos totalmente. Existem, porém, diferenças entre jogos de entretenimento e jogos educacionais. O engajamento de *gamers* (termo que se

refere a pessoas que jogam) se torna um desafio para a utilização de jogos pedagógicos, uma vez que eles estão acostumados com o mercado e sua alta produção.

Aspectos relacionados a Desafio, Satisfação, Atenção Focada e Percepção de Aprendizagem tiveram mediana 0 apenas na turma SIS. Sobre o aspecto da Percepção de Aprendizagem, vale ressaltar a diferença entre as disciplinas e o seu público pois, na turma MAT, a mediana foi +1. A disciplina da turma SIS é ofertada em um curso de informática, ou seja, as pessoas estão inseridas no meio da programação e tecnologias. Outro fato é que a avaliação foi feita no meio do período. Dessa forma, os estudantes da turma SIS já tiveram seu primeiro contato com lógica de programação, o que torna o início do ProgramSE uma repetição dos conteúdos visitados durante a disciplina, justificando as respostas dos participantes. Por sua vez, o curso de Licenciatura em Matemática (MAT) oferece essa disciplina, porém não é a área fim do curso. O público dessa turma não tem muito contato com o assunto, sendo os conceitos vistos no ProgramSE um aprofundamento de seu conhecimento. No entanto, a avaliação feita no meio do período letivo também pode ter influenciado negativamente as avaliações de Aprendizabilidade, visto que alguns conceitos já tinham sido ensinados. A proposta do jogo é para iniciantes em programação e sua utilização deve ser um dos primeiros contatos do aluno com o tema.

Os gráficos apresentados para as questões discursivas podem reafirmar a diferença entre os públicos das disciplinas. Os pontos mais citados pelos estudantes da turma SIS (tanto fortes como a melhorar) foram relacionados à jogabilidade, i.e., os elementos apresentados no jogo e sua progressão. Na turma MAT, o ponto forte mais comentado foi em relação à Fundamentação do Jogo, i.e., o conteúdo apresentado e a forma de ensinar os conceitos. Como a turma SIS teve mais contato com programação (e tem muitos *gamers*), focou atenção nos elementos de jogos em si. Outro aspecto que teve um retorno positivo foi relacionado ao *Design* do Jogo. Como visto nos gráficos de usabilidade, uma boa interface e boas funcionalidades melhoram a percepção e a visão que os usuários têm de um software, nesse caso, ProgramSE.

O questionário do modelo MEEGA+ tem campos de questões discursivas, possibilitando ao estudante reportar pontos fortes e a melhorar, de acordo com a sua percepção do jogo. Entre os pontos a melhorar (de ambas as turmas), algumas observações feitas pelos alunos ajudaram no aperfeiçoamento do jogo educacional desenvolvido. Comentários em relação à jogabilidade foram os mais identificados. A baixa imersão dos estudantes no enredo do jogo se tornou uma dificuldade para o projeto, pois diminuiu a atenção focada ao ProgramSE e seus conteúdos. Como sugestão, alguns alunos acharam o jogo curto e sugeriram a adição de novidades:

*“O jogo poderia ser mais demorado, com mais “fases””* [Estudante 7 - SIS]

*“Poderia ser mais longo”* [Estudante 3 - MAT]

Outro fator que pode ter contribuído para a baixa aceitação da jogabilidade foi o fato de alguns alunos acharem o jogo monótono. Esse aspecto foi mais comentado na turma SIS, o que pode ser justificado pelo fato da turma (no geral) não ter achado o jogo desafiador. ProgramSE se baseia na programação em blocos, ou seja, sua mecânica se repete ao longo do jogo. Por último, atualizações nas versões dos navegadores utilizados na avaliação eram diferentes daquelas testadas no desenvolvimento do jogo e no piloto, o que causou alguns transtornos, sobretudo na parte do *drag-and-drop* dos blocos de programação. No entanto, os instrutores não observaram efeitos sobre a aprendizagem dos conceitos lógica de programação contemplados no ProgramSE:

*“Mecânicas novas, menos repetitivas e mais mecânicas fora o mouse”* [Estudante 8 - SIS]

*“Tentar fazê-lo ficar menos monótono”* [Estudante 5 - MAT]

*“Melhor detecção dos cliques por conta do hitbox dos blocos (tem que clicar bem no centro ao invés de ser apenas na área dos blocos)...”* [Estudante 2 - SIS]

Entre os pontos fortes listados, os estudantes reforçaram que o jogo contribuiu para o aprendizado de lógica e conceitos de programação, elogiando a forma lúdica em que o conteúdo é apresentado, conforme mostram alguns comentários a seguir:

*“Eu fiquei surpreso com como o jogo além de ser um estímulo e um método de ensino de programação ele também te faz pensar no sentido da vida”* [Estudante 12 - SIS]

*“Excelente jogo! Apesar de ser simples, todos os meus parabéns pelo tempo e dedicação de se preocupar com quem não está introduzido nesta área”* [Estudante 16 - SIS]

*“A forma como ele aborda o conteúdo, é de uma forma mais interessante que o normal”* [Estudante 4 - MAT]

A avaliação realizada pelos alunos evidenciou alguns problemas de jogabilidade que são importantes para o seu engajamento no jogo. Entretanto, a percepção dos alunos em relação à forma com que os conceitos de programação foram ensinados foi positiva, de modo que o jogo serviu como apoio para auxiliar o ensino de conceitos relacionados com a abordagem adotada. Por ser do campo dos jogos digitais, é necessário haver, além da base conceitual, um estudo aprofundado sobre a forma de apresentar os desafios do jogo, o que é indicado para investigação futura. Isso considera sobretudo o fato de que os resultados da avaliação também indicaram a aceitação dos estudantes quanto aos jogos digitais voltados ao ensino.

## 6 Limitações e Ameaças à Validade

O trabalho teve algumas limitações, tanto em relação à construção do jogo como sobre a avaliação experimental. Durante a execução da avaliação, foi observado que alguns estudantes utilizaram um navegador desatualizado em relação ao utilizado no desenvolvimento do ProgramSE e no piloto. Assim, observou-se que algumas funcionalidades foram afetadas. Em determinados momentos, colisões nos blocos de programação com os guias de programação não ocorriam como deveriam. Isto causou um pouco de desconforto e irritabilidade em alguns participantes. Embora este problema tenha sido identificado, o fluxo do jogo não foi alterado, mantendo-se a proposta, conforme relatado pelos instrutores que participaram da avaliação. Vale ressaltar que, embora este aspecto possa impactar na percepção dos discentes sobre o jogo proposto, é possível que, quando o jogo seja usado sobre outras condições, sejam usados navegadores com versões distintas das utilizadas nos testes antes da sua liberação. Portanto, este fator permite observar o que aconteceria em condições reais de uso. Adicionalmente, não foram coletados dados sobre os navegadores utilizados, o que deve ser considerado em futuras replicações desta avaliação.

Com relação ao impacto de fatores audiovisuais, fones de ouvido não foram disponibilizados para os estudantes durante a avaliação do ProgramSE. Logo, quem não tinha trazido o acessório, conforme solicitado, não pode perceber os elementos sonoros. Isso influencia diretamente na imersão do aluno no contexto do jogo, uma vez que a trilha sonora ajuda a caracterizar o ambiente no desenrolar do enredo. ProgramSE também possuía, além da música de fundo, *feedbacks* sonoros em algumas ações. Assim, alguns deles não puderam desfrutar da total imersão no ambiente do jogo. Além disso, o fato de que os discentes estavam em um laboratório durante a realização do estudo pode ter impactado na sua percepção e imersão na utilização do jogo, visto que é um ambiente compartilhado e com distrações. No entanto, foi sugerido que os discentes se concentrassem na tarefa de experimentar o jogo, para diminuir esta ameaça. Vale ressaltar ainda que não houve participantes com daltonismo ou outra deficiência visual nesta avaliação em particular, o que abre oportunidades para novas avaliações experimentais do jogo.

Por mais que tenha sido escolhido um motor para auxiliar o processo de criação do ProgramSE, a falta de experiência por parte da equipe de desenvolvimento prejudicou o planejamento do processo de desenvolvimento, uma vez que é importante entender como os jogos

digitais funcionam e quais diretrizes seguir para construir um sistema de sucesso (ou seja, que forneça ao jogador uma experiência agradável). Sobre a RSL, a quantidade reduzida de trabalhos na área se torna uma limitação. O emprego de jogos digitais requer mais qualidades técnicas para reforçar seus benefícios e como o utilizar de forma adequada para cumprir com o objetivo de auxiliar o processo de ensino-aprendizagem.

Por fim, considerando o escopo e contexto da pesquisa realizada ao longo de suas três etapas, pode-se afirmar o jogo é indicado para ser utilizado em sala de aula no primeiro período de cursos de graduação para ensino de programação, mais precisamente na primeira metade da disciplina, de modo ajude a ensinar conceitos antes de um contato mais intenso com uma linguagem de programação. Entretanto, novos estudos precisam ser conduzidos para evoluir e avaliar o ProgramSE em outros escopos e contextos.

## 7 Conclusão

Este trabalho apresentou um jogo do gênero *Escape the Room* que utiliza desafios para ensinar conceitos de lógica de programação, chamado ProgramSE. Sua proposta é auxiliar o ensino para iniciantes na graduação, inspirado nas metodologias pedagógicas do Construtivismo e da Taxonomia de Bloom visando proporcionar ao estudante um ambiente engajador e motivador para o ensino. Considerando a aplicação da programação em blocos (e.g., *Scratch* e *Blockly*) e a metodologia lúdica no domínio da educação, o jogo educacional foi especificado, desenvolvido e avaliado. A primeira etapa consistiu em revisar os trabalhos similares realizados na última década e catalogar os principais requisitos e elementos de jogos digitais como base para a construção do jogo. A segunda correspondeu ao desenvolvimento do jogo, utilizando o motor *Construct 3*, o software de *design* de interiores *Sweet Home 3D* e o software livre *Inkscape* para as demais artes.

Após o desenvolvimento, a usabilidade e efetividade do ProgramSE foram avaliadas com o uso do modelo MEEGA+ em duas turmas de programação com a participação de 32 alunos iniciantes: disciplina de técnicas de programação de um curso de Sistemas de Informação e disciplina de introdução à computação de um curso de Licenciatura em Matemática. Após a coleta e processamento dos dados, os resultados foram analisados e discutidos de forma quantitativa e qualitativa. A percepção dos alunos em relação aos conceitos ensinados no ProgramSE foi positiva do ponto de vista dos participantes, sobretudo para os aspectos de acessibilidade e facilidade de aprender a jogar. Nesse sentido, entende-se que o jogo pode auxiliar estudantes de graduação ingressantes no processo de ensino-aprendizagem de conceitos de programação. No que se refere ao *feedback* dos estudantes, muitos comentários parabenizaram o projeto do jogo e relataram que conseguiriam assimilar o conteúdo. Em contrapartida, foram listados pontos de melhoria sobre os desafios, em particular novas mecânicas e fases, por parecerem monótonos para jogadores frequentes. Isso reforça o fato que, no campo dos jogos digitais, os aspectos avaliados pelos alunos influenciam no sucesso do projeto. Além disso, apenas os conceitos e estruturas não são suficientes para o total sucesso de um projeto, pois a jogabilidade tem um peso grande para determinar o engajamento. Dessa forma, deve-se conduzir estudos mais aprofundados sobre jogos digitais e quais são os principais elementos para a imersão e o engajamento dos participantes.

Como trabalhos futuros, pretende-se evoluir o jogo e contemplar o ensino de novos conceitos de programação. Mesmo o jogo visando ensinar lógica de programação para iniciantes, existem outros comandos mais avançados para que os estudantes tenham seu primeiro contato por meio de um jogo educacional. Isso ajudaria os alunos a terem uma visão mais completa sobre conceitos de lógica de programação de uma forma lúdica e engajada. A partir das avaliações realizadas, pretende-se ainda levar em consideração os comentários feitos e os resultados discutidos para agregar valor ao ProgramSE e tornar sua jogabilidade um fator que contribua mais

amplamente para o processo de ensino-aprendizagem. Pretende-se ainda incluir novos desafios a fim de torná-lo menos monótono e novas fases para torná-lo mais extenso.

## Agradecimentos

Os autores agradecem pelo apoio financeiro parcial da UNIRIO (Edital PPQ/2020) e da FAPERJ (Proc. 211.583/2019) para a realização desta pesquisa.

## Referências

- Backlund, P. e Hendrix, M. (2013). Educational games - are they worth the effort? A literature survey of the effectiveness of serious games, *In 2013 5th International Conference on Games and Virtual Worlds for Serious Applications – VS-GAMES* (pp. 1-8). IEEE. DOI: [10.1109/VS-GAMES.2013.6624226](https://doi.org/10.1109/VS-GAMES.2013.6624226) [GS Search]
- Bunchball Inc. (2010). *Gamification 101: an introduction to the use of game dynamics to influence behavior*. Bunchball. [GS Search]
- Costa, A. C. S. e Marchiori, P. Z. (2015). Gamificação, elementos de jogos e estratégia: uma matriz de referência, *Revista de Ciência da Informação e Documentação – InCID*, vol. 6, n. 2, pp. 44-65. DOI: [10.11606/issn.2178-2075.v6i2p44-65](https://doi.org/10.11606/issn.2178-2075.v6i2p44-65) [GS Search]
- Cruz, D. V. A. (2012). *Juventude e Jogos Digitais: envolvimento e relações sociais através dos Massively Multiplayer Online Role-Play Games*. Tese, Universidade Federal do Rio Grande do Sul. [GS Search]
- Finn, A. J. (2018). *A Mulher na Janela*. William Morrow and Company, 1ª edição.
- Henrique, M. e Silva, C. (2017). Um catálogo de requisitos pedagógicos para auxiliar o desenvolvimento de Softwares Educacionais, *In Anais do XXVIII Simpósio Brasileiro de Informática na Educação – SBIE* (pp. 506-515). SBC. DOI: [10.5753/cbie.sbie.2017.506](https://doi.org/10.5753/cbie.sbie.2017.506) [GS Search]
- Hoelfmann, C. (2016). *O uso dos jogos digitais educacionais no processo de ensino e aprendizagem*. Monografia (especialização), Universidade Federal de Santa Catarina. [GS Search]
- Kapp, K. M. (2012). *The Gamification of learning and instruction: Game-based methods and strategies for training and education*. Pfeiffer, 1ª edição. [GS Search]
- Krzyzanowski, L., Beleti Jr, C., Santiago Jr, R. e Tostes, R. A. (2019). Ensino de programação: um estudo preliminar nos cursos de licenciatura em Computação no Brasil, *In Anais dos Workshops do Congresso Brasileiro de Informática na Educação – CBIE* (pp. 21-30). SBC. DOI: [10.5753/cbie.wcbie.2019.21](https://doi.org/10.5753/cbie.wcbie.2019.21) [GS Search]
- Majed, M. (2014). *Aprenda a Programar com Scratch: Uma introdução visual à programação com jogos, arte, ciência e matemática*. Nonatec Editora Ltda. [GS Search]
- Marquez, B. S. e Silva, M. A. C. (2017). TRABALHADORES-ALUNOS: motivações e desafios que configuram um cenário de luta, *In Anais do I Encontro Internacional de Gestão, Desenvolvimento e Inovação – EIGEDIN*, v. 1, n. 1, pp. 1-16. [GS Search]
- Medeiros, R. P., Falcão, T. P. e Ramalho, G. L. (2020). Ensino e Aprendizagem de Introdução à Programação no Ensino Superior Brasileiro: Revisão Sistemática da Literatura, *In Anais do XXVIII Workshop sobre Educação em Computação – WEI* (pp. 186-190). SBC. DOI: [10.5753/wei.2020.11155](https://doi.org/10.5753/wei.2020.11155) [GS Search]

- Monclar, R. S., Silva, M. A. e Xexéo, G. (2018). Jogos com Propósito para o Ensino de Programação, *In Anais do XVII Simpósio Brasileiro de Jogos e Entretenimento Digital – SBGames* (pp. 1132-1140). SBC. [[GS Search](#)]
- Moreira, G. G. (2018). *Avaliação da Percepção de Jogadores sobre Jogos Educacionais para Ensino de Scrum*. Monografia, Universidade Federal do Ceará. [[GS Search](#)]
- Neto, G. F., Silva, B. L. E., Pereira, R. R. e Araújo Júnior, L. O. (2012). Utilização do kit Lego Mindstorm NXT no Ensino de Controle de Processos, *In Anais do XL Congresso Brasileiro de Educação em Engenharia – COBENGE*. [[GS Search](#)]
- Netto, D., Medeiros, L. M., Pontes, D. e Morais, E. (2017). Game Logic: Um jogo para auxiliar na aprendizagem de lógica de programação, *In Anais do XXV Workshop sobre Educação em Computação – WEI* (pp. 2297-2306). SBC. DOI: [10.5753/wei.2017.3546](https://doi.org/10.5753/wei.2017.3546) [[GS Search](#)]
- Petri, G., von Wangenheim, C. G. e Borgatto, A. F. (2019). MEEGA+: Um Modelo para a Avaliação de Jogos Educacionais para o ensino de Computação, *Revista Brasileira de Informática na Educação – RBIE*, vol. 27, n. 3, pp. 52-81. DOI: [10.5753/rbie.2019.27.03.52](https://doi.org/10.5753/rbie.2019.27.03.52) [[GS Search](#)]
- Piteira, M. e Haddad, S. (2011). Innovate in your program computer class: an approach based on a serious game, *In Proceedings of the 2011 Workshop on Open Source and Design of Communication – OSDOC* (pp. 49-54). ACM. DOI: [10.1145/2016716.2016730](https://doi.org/10.1145/2016716.2016730) [[GS Search](#)]
- Raabe, A., Zanchett, G. e Vahldick, A. (2015). Jogos de Programar como uma Abordagem para os Primeiros Contatos dos Estudantes com a Programação, *In Anais dos Workshops do Congresso Brasileiro de Informática na Educação – CBIE* (pp. 1485-1494). SBC. DOI: [10.5753/cbie.wcbie.2015.1485](https://doi.org/10.5753/cbie.wcbie.2015.1485) [[GS Search](#)]
- Sánchez, A. V. (2014). Tecnologias para a modalidade EAD: um estudo no cenário educacional atual, *Revista E-Tech: Tecnologias para Competitividade Industrial*, 2014/2, pp. 71-104. DOI: [10.18624/e-tech.v0i0.444](https://doi.org/10.18624/e-tech.v0i0.444) [[GS Search](#)]
- Santos, R. P. e Costa, H. A. X. (2006). Análise de Metodologias e Ambientes de Ensino para Algoritmos, Estruturas de Dados e Programação aos Iniciantes em Computação e Informática, *INFOCOMP Journal of Computer Science*, v. 5, n. 1, pp. 41-50. [[GS Search](#)]
- Santos, R. P., Vivacqua, A. S., Souza, J. M. e Costa, H. A. X. (2008). Uma Proposta de Cenário para Ensino de Algoritmos e Programação com Contribuições de Cooperação, Colaboração e Coordenação, *In Anais do XVI Workshop sobre Educação em Computação – WEI* (pp. 218-227). SBC. [[GS Search](#)]
- Scaico, A. e Scaico, P. D. (2016). Uso de Jogos em Cursos Introdutórios de Programação no Ensino Superior na Área de Computação: Uma Revisão Sistemática, *In Anais do XXVII Simpósio Brasileiro de Informática na Educação – SBIE* (pp. 549-558). SBC. DOI: [10.5753/cbie.sbie.2016.549](https://doi.org/10.5753/cbie.sbie.2016.549) [[GS Search](#)]
- Seraj, M., Katterfeldt, E. S., Bub, K., Autexier, S. e Drechsler, R. (2019). Scratch and Google Blockly: How Girls' Programming Skills and Attitudes are Influenced, *In Proceedings of the 19th Koli Calling International Conference on Computing Education Research* (pp. 1-10). ACM. DOI: [10.1145/3364510.3364515](https://doi.org/10.1145/3364510.3364515) [[GS Search](#)]
- Sheldon, L. (2012). *The Multiplayer Classroom: Designing Coursework as a Game*. Cengage Learning. [[GS Search](#)]
- Silva, E. O. e Falcão, T. P. (2020). O Pensamento Computacional no Ensino Superior e seu Impacto na Aprendizagem de Programação, *In Anais do XXVIII Workshop sobre Educação em Computação – WEI* (pp. 171-175). SBC. DOI: [10.5753/wei.2020.11152](https://doi.org/10.5753/wei.2020.11152) [[GS Search](#)]

- Silva, L. A., Batista, E. J., Silva, C. L. e Lima, A. C. (2017). Poredu: um ambiente de programação em blocos, *In Anais dos Workshops do VI Congresso Brasileiro de Informática na Educação – WCBIE* (pp. 144-151). SBC. DOI: [10.5753/cbie.wcbie.2017.144](https://doi.org/10.5753/cbie.wcbie.2017.144) [GS Search]
- Silva, R. R., Fernandes, J. C. e Santos, R. P. (2018a). Arquitetura de uma Plataforma para um Ecossistema de Software no Domínio de Ensino de Programação, *In Anais do XVII Simpósio Brasileiro de Jogos e Entretenimento Digital – SBGames* (pp. 1028-1037). SBC. [GS Search]
- Silva, R. R., Fernandes, J. C. e Santos, R. P. (2018b). Panorama da Utilização de Jogos Digitais no Ensino de Programação no Nível Superior na Última Década: Uma Revisão Sistemática da Literatura, *In Anais do XXIX Simpósio Brasileiro de Informática na Educação – SBIE* (pp. 535-544). SBC. DOI: [10.5753/cbie.sbie.2018.535](https://doi.org/10.5753/cbie.sbie.2018.535) [GS Search]
- Uzunca, B. e Jansen, S. (2016). How do Ecosystem Dynamics work in Serious Gaming Ecosystems? Challenges and Opportunities, *In Proceedings of the 36th Annual Conference on Strategic Management Society*. [GS Search]
- Victal, E. R. N., Pereira Junior, H. A., Rios, P. T. G. e Menezes, C. S. (2015). Aprendendo sobre o uso de Jogos Digitais na Educação, *In Anais do XXI Workshop de Informática na Escola – WIE* (pp. 444-453). SBC. DOI: [10.5753/cbie.wie.2015.444](https://doi.org/10.5753/cbie.wie.2015.444) [GS Search]
- Vygotsky, L. S. (1980). *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, pp. 52-91. [GS Search]
- Werbach, K. e Hunter, D. (2012). *For the win: how game thinking can revolutionize your business*. Wharton Digital Press. [GS Search]
- Wöhlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B. e Wesslén, A. (2012). *Experimentation in Software Engineering*. Springer Science & Business Media. DOI: [10.1007/978-3-642-29044-2](https://doi.org/10.1007/978-3-642-29044-2) [GS Search]
- Yildiz, S. N., Cobanoglu, A. A. e Kislá, T. (2020). Perceived Acceptance and Use of Scratch Software for Teaching Programming: A Scale Development Study, *International Journal of Computer Science Education in Schools*, vol. 4, n. 1, pp. 53-71. DOI: [10.21585/ijcses.v4i1.59](https://doi.org/10.21585/ijcses.v4i1.59) [GS Search]

## Apêndice I – Documentação do ProgramSE (*Game Document Desing*)

### Link do Jogo

<https://bit.ly/2IHRtjZ>

### História

O ProgramSE é baseado em jogos *Escape the Room*, onde o objetivo principal é encontrar uma maneira de escapar de um lugar. No enredo, o personagem principal sofre de perda de memória e se encontra preso em uma casa que não reconhece. O jogador precisa completar os desafios para prosseguir às fases seguintes (cada fase é um ambiente diferente da casa) tendo, no final do jogo, o desfecho da história.

### Gameplay

O jogo é dividido em três níveis, conforme descrito a seguir: Nível 1 (possui as fases “Quarto”, “Corredor”, “Banheiro” e “Sótão”), Nível 2 (possui as fases “Corredor”, “Cozinha” e “Sala de Estar”) e o Nível 3 (possui a fase “Hall de Entrada”).

#### Nível 1

Neste nível, o jogador será apresentado ao pensamento lógico e aos conceitos Se-Senão, Enquanto, Para e Caso. Os conceitos são apresentados sequencialmente em quatro fases: Quarto; Corredor; Banheiro; e Sótão.

A primeira fase, Quarto, apresenta os desafios Abajur, Armário, Cama e Cômoda. Nesta fase, o jogador é apresentado ao pensamento lógico. A ordem correta esperada pelo jogo para cada desafio é:

- Abajur: Faça -> Pegar a lâmpada
- Armário: Faça -> Abrir o guarda roupa
- Cama: Faça -> Olhar debaixo da cama
- Cômoda: Faça -> Mexer nas gavetas

A segunda fase, Corredor, apresenta os desafios Porta do Banheiro, Lâmpada, Tapete e Vaso de Planta. Nesta fase, o jogador é apresentado aos conceitos Se e Enquanto. A ordem correta esperada pelo jogo para cada desafio é:

- Porta do Banheiro: Se -> A porta estiver destrancada -> Faça -> Entrar no cômodo
- Lâmpada: Se -> A lâmpada estiver queimada -> Faça -> Colocar uma nova lâmpada
- Tapete: Enquanto -> O tapete estiver desarrumado -> Faça -> Esticar o tapete
- Vaso de planta: Enquanto -> A garrafa tiver água -> Faça -> Regar a planta

A terceira fase, Banheiro, apresenta os desafios Armário, Banheira, Lixeira e Toalha. Nesta fase, o jogador é apresentado aos conceitos Se, Enquanto e Para. A ordem correta esperada pelo jogo para cada desafio é:

- Armário: Se -> A porta está destrancada -> & -> A porta estiver com o puxador -> Faça -> Abrir a porta
- Banheira: Enquanto -> O ralo estiver entupido -> & -> A banheira tiver água -> Faça -> Desentupir o ralo
- Lixeira: Para -> O primeiro papel -> Até -> O último -> Faça -> Retirar o papel da lixeira
- Toalha: Para -> A primeira toalha -> Até -> A última -> Faça -> Sacudi-la

A quarta fase, Sótão, apresenta os desafios Armário, Caixa, Piano e TV. Nesta fase, o jogador é apresentado aos conceitos Se-Senão e Caso. A ordem correta esperada pelo jogo para cada desafio é:

- Armário: Se -> A porta estiver trancada -> & -> Tiver um grampo -> Faça -> Abrir a fechadura -> e -> Abrir o armário -> Senão -> Procurar um grampo
- Caixa: Se -> A caixa estiver lacrada -> & -> Tiver algo com ponta -> Faça -> Rasgar o lacre -> Senão -> Procurar algo para rasgá-lo
- Piano: Caso -> A tampa esteja fechada -> Faça -> Olhar por trás do piano -> Caso -> A tampa esteja aberta -> Faça -> Olhar pelas cordas
- TV: Caso -> A TV esteja desligada -> Faça -> Ligar a TV -> Caso -> A TV esteja ligada -> Faça -> Pegar o controle -> e -> Procurar pelos canais da TV

### *Nível 2*

Neste nível, o jogador será apresentado aos conceitos Para-Passo, Enquanto, Função, Procedimento, Vetor, Matriz e Ordenação. Os conceitos são apresentados sequencialmente em três fases: Escada; Cozinha e; Sala de Estar.

A primeira fase, Escada, apresenta o desafio Escada. Nesta fase, o jogador é apresentado ao conceito Para-Passo. A ordem correta esperada pelo jogo para cada desafio é:

- Escada: Para -> O primeiro degrau -> Até -> O final da escada -> Passo -> De um em um -> Faça -> Descer o degrau

A segunda fase, Cozinha, apresenta os desafios Armário, Cafeteira, Faqueiro, Fogão e Geladeira. Nessa fase, o jogador é apresentado aos conceitos Para-Passo, Enquanto, Função e Procedimento. A ordem correta esperada pelo jogo para cada desafio é:

- Armário: Para -> O primeiro utensílio -> Até -> O último utensílio -> Passo -> De um em um -> Faça -> Retirar objeto do armário
- Cafeteira: Função “fazerCafé” -> Recebe -> (objeto grão de café) -> Faça -> Passar café -> Retorne -> Café pronto
- Faqueiro: Procedimento “guardarFaca” -> Recebe -> (objeto faca) -> Faça -> Prender faca no ímã
- Fogão: Função “aquecer” -> Recebe -> (objeto panela, objeto item congelado) -> Faça -> Esquentar a panela -> Retorne -> Item descongelado
- Geladeira: Enquanto -> Não achar item congelado -> Faça -> Procurar no congelador -> Se -> Encontrar -> Faça -> Guardar no inventário

A terceira fase, Sala, apresenta os desafios Estande de Livros, Quadro de Fotos, Tabuleiro de Xadrez, Computador e Lareira. Nesta fase, o jogador é apresentado aos conceitos Vetor, Matriz e Procedimento. A ordem correta esperada pelo jogo para cada desafio é:

- Estante de Livros: Vetor estanteLivros[6] -> Para cada prateleira -> Faça -> Colocar livro
- Quadro de fotos: Vetor quadroFotos[4] -> Para cada espaço vazio -> Faça -> Colocar foto
- Xadrez: Matriz tabuleiro[8,8] -> Para cada linha -> Percorra todas as colunas -> Se encontrar posição -> Faça -> Colocar peça
- Computador: Matriz planilhaExcel[7,7] -> Para cada linha -> Percorra todas as colunas -> Se encontrar célula -> Faça -> Inserir valor
- Lareira: Procedimento esquentarAmbiente -> Recebe -> ( ) -> Faça -> Esquentar ambiente

### *Nível 3*

Neste nível, o jogador será apresentado ao conceito da recursividade. Para tal, é realizada uma dinâmica diferente das demais apresentadas no jogo. Toda vez que o jogador tenta sair do ambiente, retorna ao mesmo lugar até que se encontra uma carta que o faz parar o ciclo. O nível

possui uma fase única, o *Hall* de Entrada. A ordem correta esperada pelo jogo para completar o desafio, apresentado ao clicar na carta encontrada é:

- Carta: Função “salaChama” -> Recebe -> (você) -> Se leu diário -> Faça -> Sair da sala -> Senão -> “salaChama” (você)

Em cada iteração do ciclo de recursividade da fase, é apresentada uma narrativa para o jogador compreender que a sua ação de clicar na porta de saída tem uma reação. A dinâmica narrativa da recursividade é apresentada abaixo:

Narrativa: "Que lugar é esse? Preciso sair daqui"

- O jogador vai clicar na porta -> jogador volta para o mesmo cômodo

Narrativa: "O quê? Como voltei para cá? Preciso sair logo."

- O jogador vai clicar na porta -> jogador volta para o mesmo cômodo

Narrativa: "O que está acontecendo comigo? Por que não consigo sair daqui? O que está havendo?"

- O jogador vai clicar na porta -> jogador volta para o mesmo cômodo

Narrativa: "Isso não está acontecendo...não está acontecendo...acho...acho... que esse lugar continuará me chamando...preciso encontrar uma condição para sair daqui..."

- O jogador vai clicar na porta -> jogador volta para o mesmo cômodo

Narrativa: "Tudo bem...a carta...deve ter algo ali..."

- O jogador vai clicar na carta -> o desafio é apresentado-> ao acertar o código, o jogo exibirá a carta

Narrativa: “Parece que uma criança morava aqui...o que está tentando me dizer? Estou ficando louco? Será que agora eu consigo sair daqui?”

- O jogador vai clicar na porta -> fim do jogo

## Dicas

Ao decorrer dos desafios do jogo, dicas são apresentadas ao jogador com o intuito de auxiliar o processo. A relação dos conceitos e suas respectivas dicas é apresentada abaixo:

- Faça: "Dica: é necessário executar uma ação!"
- Se: "Dica: é necessário checar uma condição para que a ação seja executada!"
- Enquanto: "Dica: é necessário realizar uma ação repetidamente enquanto determinada condição permanece válida!"
- Para: "Dica: é necessário realizar uma ação repetidamente enquanto determinada condição permanece válida, desde um estado inicial até um estado final!"
- Se-Senão: "Dica: é necessário checar uma condição para que a ação seja executada, caso contrário, deve-se executar outra ação!"
- Caso: "Dica: é necessário executar diferentes ações dependendo de um estado inicial!"
- Para-Passo: "Dica: é necessário realizar uma ação repetidamente enquanto determinada condição permanece válida, desde um estado inicial até um estado final, sabendo que sua variação será de um em um!"
- Função: "Dica: é necessário montar um bloco de comandos que, ao serem chamados, serão executados e retornarão algum resultado!"; "+ obs.: Os valores recebidos pela função são chamados de Parâmetros; e o valor retornado, Retorno."
- Procedimento: "Dica: é necessário montar um bloco de comandos que, ao serem chamados, serão executados e não retornarão nada!"; "+ obs.: Os valores recebidos pelo

procedimento são chamados de Parâmetros; e diferente da Função, o Procedimento não possui retorno."

- Enquanto-Se: "Dica: é necessário realizar uma ação repetidamente e, se uma condição for verdadeira, executar outra ação!"
- Vetor: "Dica: é necessário criar uma estrutura de dados unidimensional, e percorrê-la, para armazenar itens do mesmo tipo!"; "+ obs.: Os elementos armazenados no vetor são acessados através de um índice, que é representado ao percorrer cada coluna do mesmo."
- Matriz: "Dica: é necessário criar uma estrutura de dados bidimensional, e percorrê-la, para armazenar itens do mesmo tipo!"; "+ obs.: Os elementos armazenados na matriz são acessados através de um índice, que é representado ao percorrer cada linha e coluna do mesmo."
- Recursividade: "Dica: é necessário criar uma função que contenha uma chamada a si próprio!"