



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
ESCOLA DE INFORMÁTICA APLICADA

PROGRAMSE: UM JOGO PARA APRENDIZAGEM DE CONCEITOS DE LÓGICA
DE PROGRAMAÇÃO

RODRIGO RIBEIRO SILVA

Orientador

RODRIGO PEREIRA DOS SANTOS

Coorientador

LUIS JORGE ENRIQUE RIVERO CABREJOS

RIO DE JANEIRO, RJ – BRASIL

JULHO DE 2019

Catálogo informatizada pelo(a) autor(a)

R586 Ribeiro Silva, Rodrigo
ProgramSE: Um Jogo para Aprendizagem de
Conceitos de Lógica de Programação / Rodrigo Ribeiro
Silva. -- Rio de Janeiro, 2019.
82

Orientador: Rodrigo Pereira dos Santos.
Coorientador: Luis Jorge Enrique Rivero Cabrejos.
Trabalho de Conclusão de Curso (Graduação) -
Universidade Federal do Estado do Rio de Janeiro,
Graduação em Sistemas de Informação, 2019.

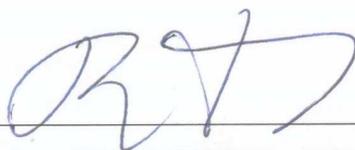
1. Ensino de Programação. 2. Jogos Digitais. 3.
Jogos Educacionais. I. Pereira dos Santos, Rodrigo,
orient. II. Jorge Enrique Rivero Cabrejos, Luis,
coorient. III. Título.

PROGRAMSE: UM JOGO PARA APRENDIZAGEM DE CONCEITOS DE LÓGICA
DE PROGRAMAÇÃO

RODRIGO RIBEIRO SILVA

Projeto de Graduação apresentado à Escola de
Informática Aplicada da Universidade Federal do
Estado do Rio de Janeiro (UNIRIO) para obtenção do
título de Bacharel em Sistemas de Informação.

Aprovada por:



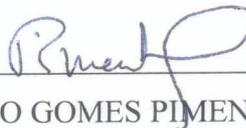
RODRIGO PEREIRA DOS SANTOS (UNIRIO)



LUIS JORGE ENRIQUE RIVERO CABREJOS (UFMA)



GEIZA MARIA HAMAZAKI DA SILVA (UNIRIO)



MARIANO GOMES PIMENTEL (UNIRIO)

RIO DE JANEIRO, RJ – BRASIL.

JULHO DE 2019

Agradecimentos

Meus queridos pais, Katia e Sergio, vocês nunca desistiram de mim. Mesmo com todas as adversidades, vocês sempre me amaram e me apoiaram. Hoje sou quem sou por vocês. Minha irmã, Fernanda, todas as nossas brincadeiras e risadas bobas com certeza me ajudaram bastante. Nossas conversas também foram fundamentais para nosso crescimento. Aos meus avós, Regina, Lina, Justino e Brás, vocês foram a base que permitiram essa família se formar. Minhas irmãs de vida, Malu e Ju, todos os memes trocados, risadas compartilhadas e desabafos. À toda minha família, brasiliense e carioca, todo o zelo e torcida de vocês é fundamental. Agradeço todos os dias por ter uma família incrível, que com seus altos e baixos, se ama e se apoia.

A todos os meus amigos, obrigado por todo o carinho. Aos amigos de faculdade, em especial Danielle, Jordana, Igor, Paula, Renata e Thais, me aturaram por todos esses anos e fizeram a vida acadêmica suportável. Mesmo seguindo caminhos diferentes, tenho a certeza da amizade de vocês. Aos grupos de amigos que a vida me deu, Buracos e Migos, a vida ficaria muito sem graça sem vocês. Caio, obrigado por acreditar em mim, me apoiar e por toda a compreensão. Aos meus queridos amigos da medicina, obrigado pela pureza e recepção de vocês.

Aos amigos acadêmicos, professores e orientadores; o caminho é longo e duro, mas a sabedoria de todos vocês o torna possível. Ao meu orientador, Rodrigo Santos, obrigado pelos ensinamentos. Thais Ferreira e Juliana Fernandes, obrigado por me ajudarem a desenvolver essa pesquisa com todo o apoio de vocês.

Por último, não menos importante, Deus, agradeço todos os dias por me permitir viver e encerrar mais esse ciclo. Que sua proteção esteja sobre mim em meus próximos passos. A jornada foi incrível, e tenho certeza que com sua benção aproveitarei muito mais cada dia e momento perto de todas as pessoas incríveis que me rodeiam.

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. Além disso, a falta de compreensão do raciocínio lógico tem sido uma das principais razões pelo alto índice de reprovação e pela 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, a utilização de jogos digitais. Este trabalho teve como objetivo projetar e desenvolver um jogo digital para auxiliar o processo de ensino-aprendizagem de programação para iniciantes no ensino superior. A proposta do jogo é auxiliar o estudante para que possa conhecer novos conceitos de lógica de programação e fixar conhecimentos prévios por meio de analogias a ações rotineiras. Como primeira etapa da metodologia adotada, foi realizada uma revisão sistemática da literatura sobre a utilização de jogos no domínio de ensino de programação a fim de identificar quais ferramentas e teorias pedagógicas são aplicadas para o ensino-aprendizagem de programação. Após essa etapa, o desenvolvimento do jogo foi realizado por meio do motor de jogo *Construct 3*, se baseou na programação em blocos para a construção de desafios e o seu desenvolvimento foi embasado pelas metodologias pedagógicas do Construtivismo e da Taxionomia de Bloom. A aceitação, eficácia e efetividade do jogo foram avaliadas em duas turmas de iniciantes em programação da UNIRIO por meio do questionário MEEGA+. Os resultados apontaram que o jogo proposto pode auxiliar os estudantes no processo de ensino-aprendizagem de programação.

Palavras-chave: ensino de programação, jogos digitais, jogos educacionais.

ABSTRACT

The teaching of programming is part of the basic academic training in Computer Science and related areas. Previous research reports that students have 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; for example, the use of digital games. In this context, this work aimed to design and develop a digital game to support the teaching and learning process of programming for beginners in higher education. The purpose of the game is to help students to know new concepts of programming logic and to establish previous knowledge through analogies to daily actions. As a first step in our methodology, a systematic literature review on the use of games in the field of programming teaching was carried out in order to identify which pedagogical tools and theories are applied to teaching and learning of programming. Next, the development of the game was carried out using the game engine called Construct 3, based on block programming to build challenges, and its development was based on the pedagogical methodologies of Bloom's Taxonomy and Constructivism. The acceptance, effectiveness and effectiveness of the game were evaluated in two classes of beginners in programming at the UNIRIO based on the MEEGA+ questionnaire. Results pointed out that the proposed game can help students in the teaching and learning process of programming.

Keywords: teaching of programming, digital games, educational games.

Índice

1	Introdução	1
1.1	Contexto	1
1.2	Problema	2
1.3	Motivação	2
1.4	Objetivo	4
1.5	Metodologia	5
1.6	Organização do texto	5
2	Fundamentação Teórica	7
2.1	Cenário para Ensino de Programação	7
2.2	Jogos Digitais Educacionais	9
2.3	Elementos de Jogos	10
2.4	Motores de Jogos	10
2.5	Trabalhos Relacionados	12
3	Revisão Sistemática da Literatura	14
3.1	O Processo da Revisão Sistemática	14
3.2	Resultados	18
3.3	Discussão dos Resultados	22
4	ProgramSE	25
4.1	Desenvolvimento do Jogo	25
4.2	Descrição do Jogo	26
4.3	Nível de Aprendizagem	28
5	Avaliação Experimental	41
5.1	Planejamento	41
5.2	Execução	42
5.3	Análise dos Dados	44
5.4	Discussão dos Resultados	54
6	Conclusão	59
6.1	Limitações	59
6.2	Trabalhos Futuros	60
6.3	Considerações Finais	61
	Anexo 1 - Planejamento do Jogo	64
	Anexo 2 - Questionário MEEGA+	68

Índice de Tabelas

Tabela 1. Justificativa para o uso de jogos. Fonte: Uzunca e Jansen (2016).	4
Tabela 2. Elementos de Jogos.	11
Tabela 3. Comparação entre as Principais Game Engines. Fonte: Silva et al. (2018b)	12
Tabela 4. Uso dos critérios da estrutura PIO.	15
Tabela 5. Critérios de seleção dos artigos.	17
Tabela 6. Sumarização do processo de busca.	18
Tabela 7. Artigos selecionados para a RSL.	19
Tabela 8. Relação Nível x Fase x Conceitos do jogo ProgramSE	28

Índice de Figuras

Figura 1. Metodologia de pesquisa	6
Figura 2. Níveis de Desenvolvimento. Fonte: Santos et al. (2008)	8
Figura 3. Arquitetura do Ambiente.	9
Figura 4. Contexto dos casos reais.	20
Figura 5. Quantidade de artigos por instituição e Percentual de artigos por região.	22
Figura 6. Tela de menu do jogo ProgramSE	27
Figura 7. (a) Quarto; (b) Corredor; (c) Banheiro; e (d) Sótão.	29
Figura 8. (a) Escada; (b) Cozinha; e (c) Sala de Estar.	30
Figura 9. Entrada.	30
Figura 10. Exemplo de Sequência.	32
Figura 11. Exemplo de Se.	33
Figura 12. Exemplo de Se/Senão.	33
Figura 13. Exemplo de Caso.	34
Figura 14. Exemplo de Enquanto.	35
Figura 15. Exemplo de Para/Passo.	35
Figura 16. Exemplo de Função.	36
Figura 17. Exemplo de Procedimento.	37
Figura 18. (a) Livro - Vetores; (b) Exemplo de Vetor.	38
Figura 19. (a) Livro - Matrizes; (b) Exemplo de Matriz.	39
Figura 20. Livro - Ordenação.	39
Figura 21. Exemplo de Recursividade.	40
Figura 22. (a) Avaliação em TP1; (b) Avaliação em ICC.	43
Figura 23. Informações demográficas (TP1).	45
Figura 24. Usabilidade (TP1).	46
Figura 25. Experiência do Jogador (TP1).	47

Figura 26. Questões Discursivas (TP1).	48
Figura 27. Informações Demográficas (TP1).	50
Figura 28. Usabilidade (ICC).	51
Figura 29. Experiência do Jogador (ICC).	52
Figura 30. Questões Discursivas (ICC).	53

1 Introdução

1.1 Contexto

Segundo as Diretrizes Curriculares Nacionais¹, a matéria de programação faz parte da formação básica em Computação e, à medida que a tecnologia avança, a demanda por profissionais nessa área também cresce. No entanto, um em cada três alunos ingressantes em Sistemas de Informação (SI) terminam o curso, enquanto para o curso de Ciência da Computação (CC), os números são de um para cada quatro alunos². Trabalhos anteriores apontam a existência de muitas dificuldades dos estudantes, sendo a falta de compreensão do raciocínio lógico uma das principais razões pelo alto índice de reprovação e pela desistência de cursos de graduação da área [Santos et al. 2008].

Muitas técnicas têm sido desenvolvidas com o intuito de diminuir a evasão dos alunos. Uma delas é a gamificação, uma metodologia representativa do uso de jogos na educação que significa a utilização de jogos digitais em contextos não relacionados a jogos [Kapp 2012]. Os jogos digitais na educação têm o poder de estimular o aluno, despertar sua curiosidade e criatividade, desenvolver a capacidade de concentração e raciocínio e proporcionar uma maneira de aprender mais prazerosa. Eles são territórios para a experimentação, onde simulam um ambiente passível de tentativas e erros [Klapztein 2014]. A lógica de funcionamento dos jogos digitais apresenta grande capacidade de promover o engajamento de seus participantes, a curiosidade, a colaboração e o aprendizado pelos erros³. Neste sentido, momentos lúdicos podem ser proporcionados como iterações para estimular processo cognitivo de estudantes [Sá et al. 2007].

¹http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=52101-rces005-16-pdf&category_slug=novembro-2016-pdf&Itemid=30192

²<http://www.semesp.org.br/imprensa/indice-de-evasao-de-alunos-e-maior-na-area-de-tecnologia-da-informacao-2/>

³<http://ois.sebrae.com.br/boaspraticas/gamefication-aplicacao-da-logica-de-jogos-na-educacao/>

1.2 Problema

As disciplinas introdutórias dos cursos da área de Computação são importantes para o desenvolvimento do estudante [Netto et al. 2017]. Elas devem estimular habilidades multidisciplinares (como a criatividade e o pensamento crítico) para fortalecer o pensamento lógico dos iniciantes, fazendo-os pensar além da limitação computacional, raciocinando e argumentando frente a desafios. A construção dessa base fortalece a motivação dos alunos, podendo ser um fator para diminuir os riscos de reprovação e evasão nas disciplinas.

Alguns métodos utilizados no modelo de ensino convencional podem não ser atrativos e engajadores para alguns estudantes. No eixo da programação, a metodologia de ensino utilizada, geralmente, envolve a criação de sistemas com linguagens de programação complexas. No entanto, o processo de ensino-aprendizagem de lógica é árduo e inovações no ensino facilitam ou auxiliam a aquisição de conhecimento. Métodos educacionais lúdicos podem ser um recurso facilitador do processo de aquisição de conhecimento.

Ferramentas populares como o *Scratch*⁴ e o *Blockly*⁵ fazem o usuário abstrair da linguagem de programação e desenvolver códigos baseado em blocos, ou seja, focando apenas na lógica sintaticamente correta. Neste caso, estas ferramentas são uma maneira intuitiva e visual de construir um código. Da perspectiva de um desenvolvedor, são uma interface pronta para criar uma linguagem visual que emite um código gerado pelo usuário. Essas ferramentas, entretanto, têm como objetivo a construção de novos aplicativos. Para este trabalho, a proposta é auxiliar o ensino de programação apresentando o pensamento lógico, desafiando os estudantes a aprenderem ou fixarem o conteúdo, relacionando-o com ações rotineiras. Há softwares com a proposta similar, porém não estão disponíveis em português e focados no público do ensino superior [Netto et al. 2017] [Silva et al. 2017].

1.3 Motivação

Dentre as metodologias de ensino utilizadas no contexto educacional, o modelo instrucionista é bastante aplicado, sendo que nele o estudante assume um papel passivo

⁴ <https://scratch.mit.edu/>

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

na aquisição do conhecimento. Como a falta de compreensão do raciocínio lógico é apontado com uma das razões para o índice de reprovação e desistência de estudantes em cursos de Computação, buscou-se nesta pesquisa propor uma ferramenta que seja potencialmente atrativa e engajadora para estudantes iniciantes no estudo de programação.

O desenvolvimento da proposta do ProgramSE 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, tendo o professor como um agente facilitador do desenvolvimento cognitivo [Neto et al. 2012]. 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, segundo 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 importantes ferramentas 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) ‘aprender fazendo’, envolvendo-se, motivando-se e entretendo-se; (3) possibilitar a capacidade de encontrar e usar a informação sem a necessidade de memorização; e (4) simular ambientes complexos/caros e situações perigosas/críticas. A Tabela 1 sumariza algumas justificativas para o uso de jogos como forma de aprendizado.

Sendo assim, a principal motivação para o desenvolvimento da proposta do ProgramSE foi fornecer uma ferramenta que auxilie o estudante a desempenhar um papel ativo no processo de aquisição do seu conhecimento e a ser o construtor da sua própria sabedoria.

1.4 Objetivo

Com o intuito de auxiliar o processo de ensino-aprendizagem de programação, os métodos para lecionar novos conteúdos vêm se difundindo. Os jogos digitais voltados ao ensino têm se tornado elementos de motivação para os alunos, uma vez que possibilitam o aprendizado de maneira lúdica. Segundo os resultados da revisão sistemática realizada por Scaico e Scaico (2016), o uso de jogos no ensino introdutório de programação traz resultados positivos para o aprendizado.

Tabela 1. Justificativa para o uso de jogos. Fonte: Uzunca e Jansen (2016).

Justificativa	Descrição
Importância crescente	Design, engenharia e produção de jogos ⁶ ainda estão em sua infância
Alvo em movimento	Negócios emergentes com múltiplos usos nos setores da indústria, educação, saúde e administração pública
Potencial de crescimento	2,35 bilhões de euros no mercado global (estimativa do IDATE 2014 ⁷)
	Fragmentado e precisa de massa crítica para competir globalmente

Levando em conta os esforços de proporcionar um ensino engajador e construtivo para o estudante, o objetivo geral deste trabalho é auxiliar o processo de ensino-aprendizagem de programação para iniciantes no ensino superior por meio de um jogo digital. Este jogo deve permitir que o estudante possa conhecer novos conceitos de lógica de programação e fixar conhecimentos prévios por meio de analogias a ações rotineiras.

Como objetivos específicos deste trabalho, destacam-se:

- Analisar jogos digitais utilizados no ensino de programação: revisar a literatura buscando entender a utilização de jogos digitais no ensino de programação no ensino superior dos últimos dez anos;
- Procurar ambientes de desenvolvimento de jogos digitais: procurar os principais motores de jogos do mercado e listar os prós e contras de sua utilização;
- Identificar elementos de jogos e a importância de cada um deles para o engajamento de usuários em jogos digitais: catalogar os elementos de jogos e

⁶Para Protopsaltis et al. (2011), são jogos que envolvem alcance de metas baseadas no mundo real.

⁷ http://www.idate.org/en/News/Serious-Games_643.html

agrupá-los em categorias, para entender como utilizá-los para o engajamento do jogo;

- Desenvolver um jogo para auxiliar o ensino de programação: realizar a especificação de requisitos, modelagem e desenvolvimento do jogo; e
- Avaliar a eficiência e a usabilidade do jogo produzido: avaliar a efetividade, a eficiência e a usabilidade do jogo desenvolvido a partir de estudos experimentais.

1.5 Metodologia

O projeto foi dividido em três etapas, sendo a primeira, a concepção do jogo; a segunda, sua construção; e a terceira, a avaliação. Para a realização da primeira etapa, o primeiro passo consistiu em uma revisão sistemática da literatura (RSL) acerca da utilização de jogos digitais para o ensino de programação da última década. O objetivo da RSL foi identificar o panorama da utilização de jogos digitais para a aprendizagem de certos conceitos do tema no ensino superior. Este passo gerou uma publicação de artigo completo no Simpósio Brasileiro de Informática na Educação 2018 [Silva et al. 2018a].

Com base nos resultados obtidos, o segundo passo consistiu no planejamento para o desenvolvimento do jogo ProgramSE. Foi realizada uma investigação de elementos e motores de jogos para suporte a um ambiente para ensino de programação baseado na colaboração de desenvolvedores externos ao projeto. Este passo gerou uma publicação no Simpósio Brasileiro de Jogos e Entretenimento Digital 2018 [Silva et al. 2018b]. O motor de jogo *Construct 3*⁸ foi escolhido para o desenvolvimento do jogo ProgramSE. No terceiro passo, a avaliação foi realizada com alunos ingressantes nas disciplinas de programação utilizando um questionário baseado no MEEGA⁹ (Petri et al. 2018). A Figura 1 exhibe as atividades realizadas na pesquisa.

1.6 Organização do texto

O presente trabalho está estruturado em capítulos e, além desta introdução, o Capítulo 2 apresenta a fundamentação teórica que embasou a proposta apresentada; o Capítulo 3 detalha o processo de RSL que permitiu esboçar um panorama da utilização

⁸ <https://www.construct.net/>

⁹ <http://www.gqs.ufsc.br/tag/meega/>

de jogos digitais no ensino de programação no ensino superior; o Capítulo 4 apresenta o jogo ProgramSE e os detalhes do processo de desenvolvimento; o Capítulo 5 descreve o estudo experimental realizado para a avaliação do jogo; e, por fim, o Capítulo 6 traz a conclusão do estudo.

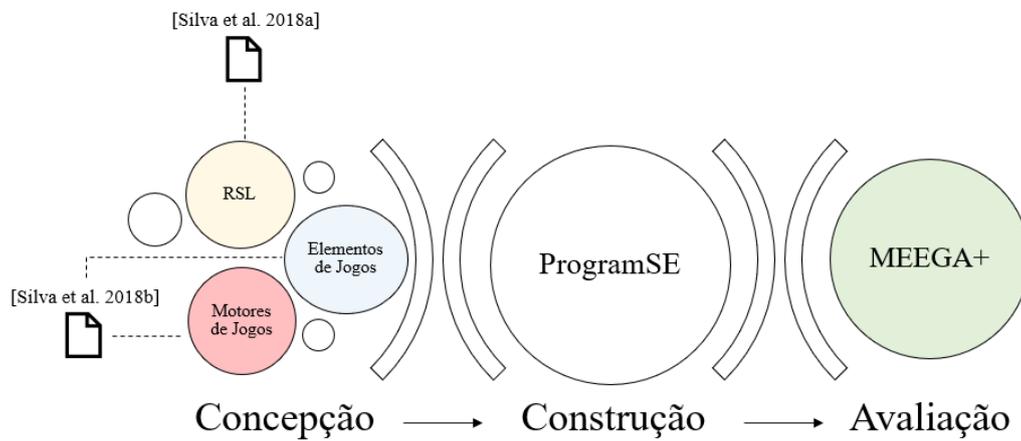


Figura 1. Metodologia de pesquisa

2 Fundamentação Teórica

Este capítulo apresenta a fundamentação teórica necessária para o entendimento do trabalho. O capítulo está organizado da seguinte forma: a Seção 2.1 apresenta o cenário para o ensino de programação; a Seção 2.2 explica sobre utilização de jogos digitais educacionais; nas Seções 2.3 e 2.4, os elementos de jogos e motores de jogos, respectivamente, são apresentados; e a Seção 2.5 apresenta os trabalhos relacionados.

2.1 Cenário para Ensino de Programação

A falta de compreensão do raciocínio lógico representa uma das principais razões do alto índice de reprovação nas disciplinas de algoritmos [Santos et al. 2008]. Devido a isto, é importante pensar em um cenário que o processo de ensino-aprendizado aconteça de forma eficaz a ponto de estimular o estudante a prosseguir mesmo diante de desafios e reprovações. Nesse caso, é importante identificar o principal papel do professor a fim de incentivar esse crescimento.

Para Vygotsky (1980), existem três níveis de desenvolvimento e aprendizagem (Figura 2): (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 e este 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. Se estiverem além da zona ZPD (i.e., NDP), o aluno pode ficar desmotivado.

Para conseguir atingir a zona adequada, é importante entender e esboçar o cenário ideal para o processo de ensino-aprendizagem. A partir dessa análise, Santos et al. (2008) identificaram o tutor, o monitor e o aprendiz como os principais atores nesse contexto. Seguindo a linha de raciocínio de Santos et al. (2008), este trabalho de conclusão de curso também é baseado na utilização de três personagens fundamentais

para o quadro proposto, sendo eles: (1) o educador; (2) o monitor; e (3) o aprendiz. Todos têm papéis distintos no processo e possuem características próprias.

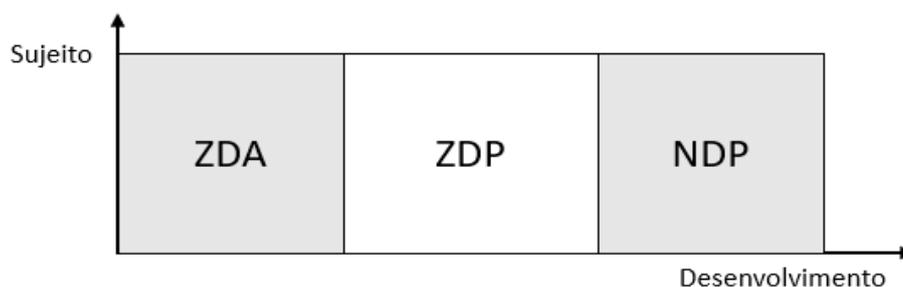


Figura 2. Níveis de Desenvolvimento. Fonte: Santos et al. (2008)

- O educador - possui a função de mediar e facilitar o processo de ensino-aprendizagem, e.g. professor;
- O monitor - atua como colaborador do processo, complementando o educador, e.g. alunos mais experientes;
- O aprendiz - um conjunto de pessoas que têm em comum a assimilação do conteúdo como objetivo principal, e.g. alunos da disciplina em questão.

A arquitetura para o ambiente de ensino é apresentada na Figura 3 e, para compreender o cenário, as relações entre os elementos são fundamentais e estão descritas a seguir:

- Jogo Digital / Aprendiz - a linha de comunicação do jogo digital com o aprendiz é unidirecional, onde a ferramenta em questão apresenta o conteúdo de forma lúdica ao aluno. O canal se dá através de alguma plataforma, e.g. internet;
- Aprendiz / Canal de Comunicação / Educador e Monitor - essa relação é extremamente importante para o cenário, pois por meio dela o aprendiz consegue expor seus conhecimentos e dúvidas ao educador/monitor. Por se tratar de um caminho bidirecional, também possibilita que o aluno receba um *feedback* dos tutores. A comunicação se dá a partir de um canal, que pode ser um sistema de gerenciamento, e.g. Moodle ou ambientes não digitais;
- Educador e Monitor / Jogo Digital - esse trecho fica restrito à escolha do educador de como o jogo digital será utilizado para auxiliar o processo de ensino-aprendizagem.

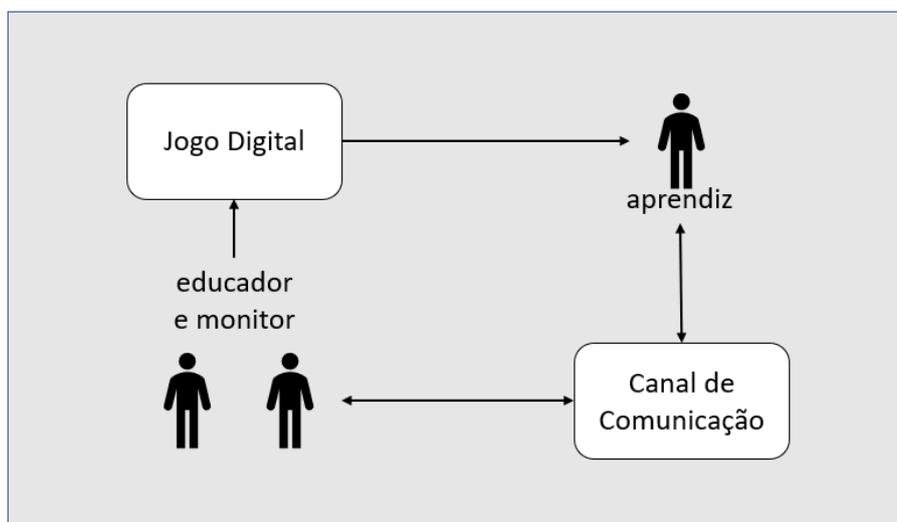


Figura 3. Arquitetura do Ambiente de Ensino. Fonte: Silva et al. (2018b)

2.2 Jogos Digitais Educacionais

Atualmente, os jogos digitais têm sido um recurso utilizado para auxiliar o processo de ensino-aprendizado, visto que são uma forma lúdica para abordar um problema qualquer. Em um estudo realizado por Hoelfmann (2016), os jogos digitais têm grande potencial para auxiliar a educação, podendo ser um importante instrumento para o processo. A lógica de funcionamento dos jogos digitais apresenta grande capacidade de promover o engajamento de seus participantes, a curiosidade, a colaboração e o aprendizado pelos erros¹⁰.

Os jogos digitais educacionais não se destinam apenas ao entretenimento do jogador, mas são desenvolvidos para o ensino de algum conteúdo, ou seja, para expandir os conhecimentos de seus jogadores, simulando atividades práticas em um ambiente não real [Moreira 2018]. O diferencial é a maior interatividade entre o estudante e o conteúdo a ser ensinado, tornando o aprendizado dinâmico e enriquecido de experiências que podem envolver emoções, capturar a atenção dos alunos, promover sua diversão, gerar um sentimento de realização ao completar as tarefas do jogo e oferecer novos desafios adequados ao contexto.

No cenário de ensino de programação, por exemplo, os jogos digitais podem ajudar os estudantes a sanarem suas dúvidas com representações visuais dos conceitos aprendidos em sala. Com o aumento do uso das tecnologias digitais, diversas mudanças ocorreram nas relações e na maneira de pensar [Sánchez 2014], influenciando

¹⁰ <http://ois.sebrae.com.br/boaspraticas/gamefication-aplicacao-da-logica-de-jogos-na-educacao/>

diretamente no modo com que os estudantes estudam e aprendem e gerando resultados positivos a partir de experiências com jogos educacionais [Sheldon 2012].

2.3 Elementos de Jogos

Para que um jogo digital possa compor um cenário de ensino, faz-se necessário que haja uma combinação adequada de elementos de jogos para o engajamento do jogador no contexto proposto. Nesse sentido, pode-se definir de forma mais consistente como sendo uma estratégia apoiada na aplicação de elementos de jogos para atividades *non-game*¹¹ que é 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 são classificados de três tipos: (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. As três categorias são definidas na Tabela 2. Kapp (2012) também comenta sobre outro importante item: a 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 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 através do cenário e, a partir desse movimento, concluir as etapas propostas para atingir o objetivo final.

2.4 Motores de Jogos

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 de jogos. Do inglês *game engines* (ou apenas *engine*), tais motores consistem em programas de computador capazes de simplificar o processo de criação de jogos digitais.

¹¹ Forma de entretenimento que não tem um vencedor ou conclusão [Iwata 2005].

Tabela 2. 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).
Dinâmico	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.
Dinâmico	Progressão	Ideia de dar aos jogadores a sensação de avançar dentro do jogo.
Dinâmico	Relacionamentos	Refere-se à interação entre os jogadores, seja entre amigos, companheiros ou adversários.
Dinâmico	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.
Mecânico	Feedback	A avaliação permite que os jogadores vejam como estão progredindo no jogo.
Mecânico	Chance	Os resultados de ação do jogador são aleatórios para criar uma sensação de surpresa e incerteza.
Mecânico	Cooperação e competição	Cria-se um sentimento de vitória e derrota.
Mecânico	Desafios	Os objetivos que o jogo define para o jogador.
Mecânico	Recompensas	O benefício que o jogador pode ganhar a partir de uma conquista no jogo.
Mecânico	Transações	Significa compra, venda ou troca de algo com outros jogadores no jogo.
Mecânico	Turnos	Cada jogador no jogo tem seu próprio tempo e oportunidade para jogar.
Mecânico	Vitória	O “estado” que define ganhar o jogo.
Componente	Avatar	Representação visual do personagem do jogador.
Componente	Bens virtuais	Itens dentro do jogo que os jogadores podem coletar e usar de forma virtual e não real, mas que ainda tem valor para o jogador.
Componente	Boss	Um desafio geralmente difícil no final de um nível que tem de ser derrotado, a fim de avançar no jogo.
Componente	Coleções	Formadas por itens acumulados dentro do jogo. Emblemas e medalhas são frequentemente parte de coleções.
Componente	Combate	Disputa que ocorre em que o jogador possa derrotar oponentes em uma luta.
Componente	Conquistas	Recompensa que o jogador recebe por fazer um conjunto de atividades específicas.
Componente	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.
Componente	Medalhas	Representação visual de realizações dentro do jogo.
Componente	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.
Componente	Missão	Similar a “conquistas”. É uma noção de jogo de que o jogador deve fazer executar algumas atividades que são especificamente definidas dentro da estrutura do jogo.
Componente	Níveis	Representação numérica da evolução do jogador. O nível do jogador aumenta à medida que o jogador se torna melhor no jogo.
Componente	Pontos	Ações no jogo que atribuem pontos. São muitas vezes ligadas a níveis.
Componente	Presentes	A possibilidade distribuir ao jogador coisas como itens ou moeda virtual para outros jogadores.
Componente	Ranking	Lista jogadores que apresentam as maiores pontuações/conquistas/itens em um jogo.
Componente	Times	Possibilidade de jogar com outras pessoas com mesmo objetivo.

Os motores de jogos servem como um conjunto de bibliotecas que facilitam o desenvolvimento, fazendo com que o programador não precise fazer tudo do zero. Geralmente, dão suporte para animações, sons, colisões, parte gráfica, entre outros.

Cada motor de jogo possui um ambiente de desenvolvimento e características próprios. A Tabela 3 apresenta alguns motores de jogos [Silva et al. 2018b]. Além disso, a Tabela 3 também exibe a comparação entre as principais tecnologias utilizadas para se desenvolver jogos. O comparativo foi realizado para auxiliar na escolha da *engine* mais adequada para desenvolver o jogo digital proposto.

Tabela 3. Comparação entre os Principais Motores de Jogos. Fonte: Silva et al. (2018b)

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++	<i>Drag-n-drop</i> , detecta colisão e noções de eventos	Só pode executar em HTML no <i>Free</i>	Scirra
GameMaker	Delphi	Tutoriais, versão "lite", linguagem simples	Linguagem própria	Yoyo
Scratch	Visual	Fácil "programação"	Comandos limitados	Livre
Stencyl	Haxe, C++ e Java	<i>Drag-n-drop</i> , multiplataforma, possibilita extensões	-	Stencyl, LLC
Unity 3D	C++ e C#	3D, multiplataforma, fácil de usar, melhor condição de licenciamento	Ferramentas limitadas	Unity Technologies

Muitas ferramentas podem ser encontradas no mercado. No entanto, a escolha da tecnologia para a construção da proposta considerou a combinação de diversos elementos, como o preço/orçamento, a existência de uma comunidade para suporte, a fácil aprendizagem e as atualizações disponibilizadas pela mantenedora.

2.5 Trabalhos Relacionados

Alguns trabalhos relacionados à proposta do ProgramSE foram identificados. Silva et al. (2017), por exemplo, construíram uma plataforma de programação virtual para ensinar crianças e adolescentes a programar. Os autores basearam 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, contemplando o ensino fundamental e médio. Segundo a avaliação dos jogadores, o software obteve um retorno positivo e otimista. Porém, é concluído que são necessárias novas versões para melhorar a captação de *feedback*. Como no trabalho anterior, a metodologia para ensinar escolhida foi baseada na programação em blocos.

Tendo em vista os trabalhos relacionados 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 software voltado para esse público-alvo. A ferramenta proposta foi projetada em português e teve como objetivo apoiar o aluno no aprendizado de conceitos de programação. Como consequência, espera-se que o jogo possa ser uma alternativa utilizada para estimular estudantes iniciantes no curso superior de Computação e auxiliar na diminuição das taxas de evasão nos cursos introdutórios de programação.

3 Revisão Sistemática da Literatura

Uma RSL foi realizada a fim de conhecer o cenário brasileiro da utilização de jogos digitais no ensino de programação do ensino superior, buscando aplicar na proposta do ProgramSE elementos relevantes reportados nos estudos. Portanto, este capítulo apresenta os resultados dessa revisão.

Foi identificado que o estudo de Medeiros et al. (2013), referente aos anos 2008 a 2012, se propunha a discutir o cenário da utilização de jogos. Portanto, este estudo foi replicado a fim de sedimentar as novas metodologias e ferramentas educacionais voltadas para o ensino-aprendizado de programação no cenário brasileiro. Dessa forma, os artigos relacionados ao uso de jogos digitais para o ensino de programação para iniciantes em Computação no Brasil no nível superior foram analisados, abrangendo os anos 2013 a 2017. Foram comparados os resultados obtidos ao longo da última década e identificadas quais ferramentas e teorias pedagógicas são aplicadas para o ensino-aprendizagem de programação. Com este resultado, o ProgramSE pode se basear nas evidências encontradas nos últimos dez anos sobre a utilização de jogos digitais no ensino de programação do ensino superior.

Os resultados obtidos foram publicados em um artigo completo no Simpósio Brasileiro de Informática na Educação (SBIE) [Silva et al. 2018a] e o restante do capítulo está organizado da seguinte forma: a Seção 3.1 apresenta o protocolo utilizado na RSL realizada; e, nas Seções 3.2 e 3.3, os resultados são apresentados e discutidos, respectivamente.

3.1 O Processo da Revisão Sistemática

A revisão conduzida foi baseada no processo de Kitchenham e Charters (2007), em que uma abordagem sistemática é pré-definida por meio de um protocolo e métodos para identificar, avaliar e interpretar as evidências disponíveis, relacionadas a uma ou mais questões de pesquisa. O processo desta RSL é descrito nas próximas subseções.

3.1.2 Questões de Pesquisa

A definição das questões de pesquisa foi feita a partir dos critérios especificados pela estrutura PICO (população, intervenção, comparação e resultados) de Kitchenham e Charters (2007). Entretanto, uma parte dos atributos de PICO é considerada neste trabalho, como mostra a Tabela 4: população, intervenção e resultado (PIO). O motivo para esta decisão está no fato de que o objetivo é identificar e discutir o uso de jogos digitais no ensino de programação aos iniciantes em Computação no nível superior, e não compará-los em termos de algum atributo.

Tabela 4. Uso dos critérios da estrutura PIO.

População	Iniciantes em Computação no nível superior
Intervenção	O uso de jogos digitais no ensino de programação
Resultados	Panorama da última década

Seguindo a abordagem GQM (*Goal-Question-Metric*) [Basili, 1992], o objetivo deste estudo foi formalizado: **analisar** o uso de jogos digitais no ensino de programação para iniciantes em Computação no nível superior **com o propósito de** caracterizar **com respeito ao** panorama da última década **do ponto de vista de** pesquisadores **no contexto do** cenário brasileiro. A partir desse objetivo, a questão de pesquisa (QP) desta RSL é: (QP) *Qual o panorama do uso de jogos digitais no ensino de programação para iniciantes em Computação no nível superior na última década?*. Para respondê-la, foram utilizadas subquestões de pesquisa (Sub-Q) propostas em Medeiros et al. (2013):

- Sub-Q1: Como os jogos digitais têm sido aplicados no ensino de programação?
- Sub-Q2: Que ferramentas têm sido empregadas?
- Sub-Q3: Quais os efeitos observados nos alunos com a introdução de atividades de programação baseadas em jogos digitais?
- Sub-Q4: Que teorias pedagógicas estão sendo utilizadas?
- Sub-Q5: Quais são as instituições de pesquisas envolvidas na área e como elas estão distribuídas no Brasil?

A SubQ de Medeiros et al. (2013) identificada como ‘QP2’ foi desconsiderada porque diz respeito ao público-alvo (nível médio e superior) e, em nosso estudo, o público-alvo é restringido aos iniciantes no nível superior. Além disso, a agregação dos

resultados da RSL de Medeiros et al. (2013) e desta RSL foi feita para responder a uma nova Sub-Q visando evidenciar o que é consenso na literatura e o que é novidade:

- Sub-Q6: Quais os avanços obtidos no uso de jogos no ensino de programação para iniciantes em Computação no nível superior na última década?

3.1.3 Estratégia de Pesquisa e Fontes de Busca

A estratégia de pesquisa utilizada consistiu em buscas manuais nos anais dos últimos cinco anos (2013 a 2017) de quatro eventos e três revistas na área de Informática na Educação do Brasil. Além das fontes escolhidas por Medeiros et al. (2013) para 2008 a 2012, foram investigadas novas fontes que surgiram a partir de 2012, o que resultou na inclusão do Workshop de Desafios da Computação Aplicada à Educação. Dessa forma, as buscas foram realizadas nas bases descritas a seguir: (1) Simpósio Brasileiro de Informática na Educação (SBIE); (2) Workshop de Desafios da Computação Aplicada à Educação (DesafIE); (3) Workshop de Informática na Escola (WIE); (4) Workshop sobre Educação em Computação (WEI); (5) Revista Brasileira de Informática na Educação (RBIE); (6) Revista de Informática Teórica e Aplicada (RITA); e (7) Revista Novas Tecnologias na Educação (RENOTE).

3.1.4 Critérios de Seleção

Todos os artigos escolhidos para esta RSL foram submetidos a um conjunto de critérios de seleção, conforme mostra a Tabela 5. Tais critérios foram adaptados de Medeiros et al. (2013) a fim de contemplarem somente os estudos referentes ao ensino no nível superior. A robótica educacional, sendo discutida como uma ferramenta para auxiliar as novas metodologias de ensino-aprendizagem de programação, é relevante pelo seu caráter multidisciplinar [Attrot e Ayrosa 2002]. Devido a isso, o seu uso foi incluído na análise desta RSL. O procedimento de seleção dos artigos foi feito em quatro passos, listados a seguir: (1) Leitura do título e do resumo; (2) Leitura da introdução e da conclusão; (3) Leitura do artigo completo; e (4) Revisão da atividade de seleção.

3.1.5 Critérios de Qualidade

Para avaliar os aspectos metodológicos dos estudos selecionados, foram utilizados os critérios de qualidade (CQ) levantados por Medeiros et al. (2013):

- CQ1: Foi utilizada alguma teoria pedagógica para auxiliar no processo de aprendizagem de programação?

- CQ2: Foi utilizada alguma ferramenta existente no mercado para auxiliar no processo de ensino-aprendizagem de programação?
- CQ3: Foi realizado algum tipo de experimento controlado ou estudo de caso real para avaliação da metodologia proposta?
- CQ4: O estudo avaliado apresenta uma breve comparação com outras metodologias relacionadas existentes?
- CQ5: O estudo avaliado apresenta o endereço (URL) na Internet onde o jogo digital utilizado esteja disponível para uso ou para cópia (*download*)?
- CQ6: O estudo avaliado foi aplicado em alguma universidade?

Tabela 5. Critérios de seleção dos artigos.

Tipo	ID	Critério
Inclusão	CI1	O tema do artigo aborda o ensino de programação com jogos digitais e/ou robótica para iniciantes em Computação no nível superior no Brasil.
	CI2	O tema do artigo é sobre o ensino de programação com jogos digitais e/ou robótica para iniciantes em Computação no nível superior no Brasil.
	CI3	A ferramenta utilizada no trabalho foi desenvolvida e está disponível.
Exclusão	CE1	O tema do artigo aborda o ensino de programação sem o uso de jogos digitais (e/ou robótica) para iniciantes em Computação no nível superior no Brasil.
	CE2	O tema do artigo não é sobre o ensino de programação com jogos digitais (e/ou robótica) para iniciantes em Computação no nível superior no Brasil.
	CE3	A ferramenta utilizada no trabalho não foi desenvolvida.
	CE4	Artigos duplicados.
	CE5	Artigos resumidos.

A escala de pontuação dos critérios de qualidade utilizada foi proposta por Kitchenham e Charters (2007), baseando-se nas respostas: (1) Sim, 1 ponto; (2) Parcial, 0,5 ponto; e (3) Não, 0 ponto. Esta avaliação visa classificar a qualidade dos artigos, de modo que os estudos que obtiveram nota zero foram retirados da RSL. Nessa etapa, dois pesquisadores (P1 e P2) leram individualmente os artigos selecionados e os pontuaram

de acordo com a escala citada acima. Ao final, os resultados obtidos foram comparados por um terceiro pesquisador (P3), que julgou e resolveu divergências entre P1 e P2.

3.1.6 Formulário de Extração de Dados

Após a seleção, os artigos foram submetidos a uma extração sistemática com uso de um formulário definido visando responder às Sub-Qs. Isso favoreceu a síntese, pois permitiu a retirada uniforme de dados dos artigos. Os campos que compunham esse formulário são: (1) estudo de caso real; (2) metodologia pedagógica; (3) ferramenta utilizada; (4) em que universidade foi desenvolvido o projeto; (5) resultado com os alunos; e (6) trabalhos futuros.

3.2 Resultados

A busca manual resultou em 34 artigos, dos quais 12 foram selecionados. Como esta revisão é uma replicação, foram incluídos ainda oito estudos de Medeiros et al. (2013), totalizando 20 para análise final. Os artigos reutilizados de 2013 atendem aos critérios exibidos nas Subseções 4.3 e 4.4. A Tabela 6 apresenta uma compilação do processo de busca realizado nesta RSL referente aos anos 2008-2012 e 2013-2017 (atualização). Na Tabela 7, são apresentados os artigos selecionados e seus identificadores próprios (ID).

Tabela 6. Sumarização do processo de busca.

Evento/ Revista	Quantidade de Artigos		Busca Manual		Selecionados	
	2008-2012	2013-2017	2008-2012	2013-2017	2008-2012	2013-2017
SBIE	434	721	18	14	5	2
DesafIE	-	69	-	0	-	0
WIE	200	427	8	5	3	2
WEI	122	215	13	10	0	5
RBIE	96	156	4	1	0	1
RITA	78	101	0	0	0	0
RENTE	584	570	3	4	0	2

Tabela 7. Artigos selecionados para a RSL.

ID	Título	Ano	Instituição	Fonte
A1	Ensino de Algoritmos Apoiado pelo Uso de Jogos Digitais Educativos	2013	UDESC	RENOTE
A2	Jogo Baralho das Variáveis	2013	UFPA	WEI
A3	Teddy Racer: Lógica de Programação e Ludicidade	2015	IFES	RENOTE
A4	DEG4Trees: Um Jogo Educacional Digital de Apoio ao Ensino de Estruturas de Dados	2015	UFG	WEI
A5	Testando a Diversão em um Jogo Sério para o Aprendizado Introdutório de Programação	2015	UDESC	WEI
A6	Robótica Educativa na aprendizagem de Lógica de Programação: Aplicação e análise	2015	IFBA	WIE
A7	O Desafio da Serpente - Usando gamification para motivar alunos em uma disciplina introdutória de programação	2016	UFPB	SBIE
A8	Um Modelo Lúdico para o Ensino de Conceitos de Programação de Computadores	2016	UNIFACS	SBIE
A9	POOGame: Um Jogo Sério para o Ensino de Programação Orientada a Objetos	2016	UFC	WEI
A10	Um Modelo Lúdico para o Ensino de Conceitos de Programação de Computadores	2017	UNIFACS	RBIE
A11	Game Logic: Um jogo para auxiliar na aprendizagem de lógica de programação	2017	IFMS	WEI
A12	Análise do Laboratório Remoto de Robótica Educacional (LERO) como Ferramenta de Ensino e Aprendizagem de Introdução à Programação	2017	IFBA	WIE
A13	O uso do Lego Mindstorms no apoio ao Ensino de Programação de Computadores	2009	FURB	WIE
A14	Aprendendo a Ensinar Programação Combinando Jogos e Python	2010	UFPB	SBIE
A15	Avaliação Empírica da Utilização de um Jogo para Auxiliar a Aprendizagem de Programação	2010	UNIVALI	SBIE
A16	ProGame: um jogo para o ensino de algoritmos e programação	2010	IESP, UFPB	SBIE
A17	Uma abordagem lúdica para a aprendizagem de programação de computadores	2010	PUC-PR	WIE
A18	A Robótica como Ferramenta de Apoio ao Ensino de Disciplinas de Programação em Cursos de Computação e Engenharia	2011	UFF	WIE
A19	Limitações da Utilização do Alice no Ensino de Programação para Alunos de Graduação	2012	PUC-PR	SBIE
A20	Uma visão do cenário Nacional do Ensino de Algoritmos e Programação: uma proposta baseada no Paradigma de Programação Visual	2012	USP	SBIE

3.2.1 Como os jogos digitais têm sido aplicados no ensino de programação?

Para responder a Sub-Q1, compreender o nível de apoio (estudo de caso real) realizado por cada pesquisa é necessário para validar o processo de ensino-aprendizagem. Os artigos selecionados apresentam formas diferentes de avaliação: seis apresentaram oficinas como caso real (A1-2, A5, A9, A12, A19); dois recorreram a avaliações por especialistas (A3-4); sete foram realizados ao longo do período letivo dos cursos de Computação (A6-7, A13-16, A18); três utilizaram estudos experimentais com alunos (A8, A10-11); e dois ofereceram curso fora das disciplinas (A17, A20). A Figura 4 mostra um gráfico com a contagem dos casos reais. Pode-se perceber que a validação durante o período letivo é o método mais utilizado entre os estudos, enquanto as oficinas (validação do estudo em aula específica) foram o segundo.

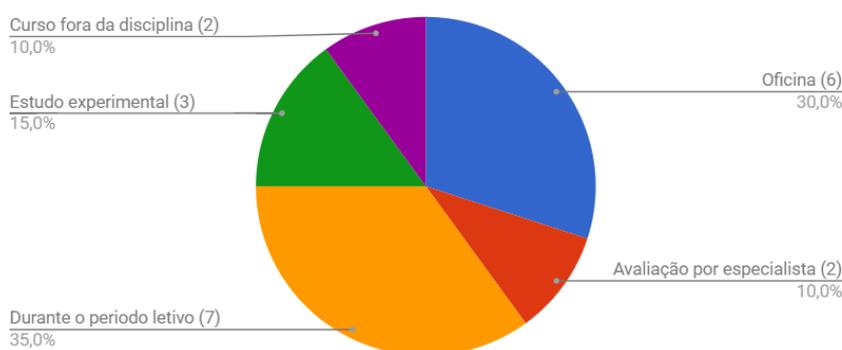


Figura 4. Contexto dos casos reais.

3.2.2 Que ferramentas têm sido empregadas?

Para responder a Sub-Q2, foram encontradas 15 ferramentas para apoio à programação: HTML, CSS, JavaScript, Adobe Flash Professional CS5, Blockly, Unity 3D, Construct 2, GameMaker, Lego Mindstorm, RoboMind, Stencyl, PyGame, Escracho, Alice e iVprog. HTML, CSS e Javascript são fundamentais para gerar documentos com padrões lidos por *browsers* (A1). Adobe Flash Professional CS5 é usado para criar conteúdo interativo e permitir manuseio de objetos 3D (A2). Blockly é uma biblioteca JavaScript da Google para construir editores de programação visual e se assemelha ao Scratch, usado como *framework* junto com HTML e Java (A5). Unity 3D é um motor de jogo multiplataforma que suporta animações 3D, tem licença proprietária e usa as linguagens C++ e C# (A3, A9). Construct 2 também é um motor que não requer codificação (o usuário cria o enredo do jogo e o resto da aplicação é feito pela ação de arrastar e soltar os objetos no cenário), com licença proprietária e tem suporte a Java e C++ (A8, A10).

Outro motor, Game Maker, é proprietário, possui linguagem simples baseada em Delphi e conta com recursos próprios como editores de imagens, sons, *scripts* e fases (A16).

Por outro lado, Lego Mindstorms é uma linha do brinquedo LEGO que explora introdução à robótica, estimulando a criatividade e a solução de problemas (A6, A18). RoboMind é uma linguagem de programação educacional para ensinar noções básicas de lógica, robótica e inteligência artificial em um ambiente simulador que, a partir de comandos do usuário, faz com que um robô explore um local (A13). Já Stencyl é uma ferramenta comercial para desenvolver jogos 2D, cujos produtos podem ser exportados para Web via Adobe Flash Player (A4). PyGame é uma biblioteca de jogos com Python, com API em C e funcionalidades de programação gráfica, além de ter licença do tipo LGPL (A14). Escracho é um programa cuja linguagem contém elementos gráficos, que são arrastados e encaixados, desde que respeitem restrições sintáticas (A15). Alice é um ambiente de programação tridimensional com licença livre que possibilita o uso de animações 3D e arrastar instruções para criar jogos (A19). Por fim, o iVprog permite interação com elementos visuais, sendo usado para ensinar algoritmos em aplicações Web (A20). Alguns trabalhos não informaram a ferramenta usada (A7, A11-12, A17).

3.2.3 Quais os efeitos observados nos alunos com a introdução de atividades de programação baseadas em jogos digitais?

A maioria dos estudos afirma que os jogos digitais utilizados no ensino de programação para iniciantes em Computação contribuíram para a aprendizagem. Um artigo não havia reportado avaliação (A7) e outro relatou que o uso de jogos digitais desenvolvidos com Alice não contribuiu para melhor rendimento do aluno em disciplinas de programação (A19). De modo geral, o uso de jogos digitais e/ou métodos lúdicos para o ensino de programação têm uma alta aceitação dos alunos, promovendo o seu engajamento.

3.2.4 Que teorias pedagógicas estão sendo utilizadas?

As principais teorias pedagógicas utilizadas pelos estudos são a Teoria Pedagógica do Construtivismo (CON) e a Taxionomia 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 realize escolhas durante a interação (A1, A6, A13, A18). Por sua vez, a TX (ou taxonomia dos objetivos educacionais) deve auxiliar a identificação da hierarquia do propósito do sistema ligado ao desenvolvimento cognitivo do aprendiz. Segundo Henrique e Silva (2017), os conteúdos apresentados devem ser divididos em vários níveis de conhecimento, iniciando sempre do menor nível de complexidade (A7, A15-16). Treze artigos não explicitaram uso de teorias pedagógicas para desenvolver a metodologia de ensino de programação aos iniciantes em Computação no nível superior (A2-5, A8-12, A14, A17, A19, A20).

3.2.5 Quais são as instituições de pesquisas envolvidas na área e como elas estão distribuídas no Brasil?

Esta revisão encontrou 15 instituições realizando pesquisas na área. A Figura 5.a exibe a relação entre as instituições identificadas e a quantidade de trabalhos. Vale ressaltar que um dos estudos (A16) foi realizado pela IESP em conjunto com a UFPB. Além disso, pode-se perceber que a UFPB conduziu três estudos; UDESC, IFBA, UNIFACS e PUC-PR conduziram dois; e as demais, apenas um. Na Figura 5.b, observa-se a distribuição das pesquisas em função das regiões. As regiões Nordeste, Sul e Sudeste são as que apresentam maior engajamento nas pesquisas na área, com 28,6% de autoria cada uma.

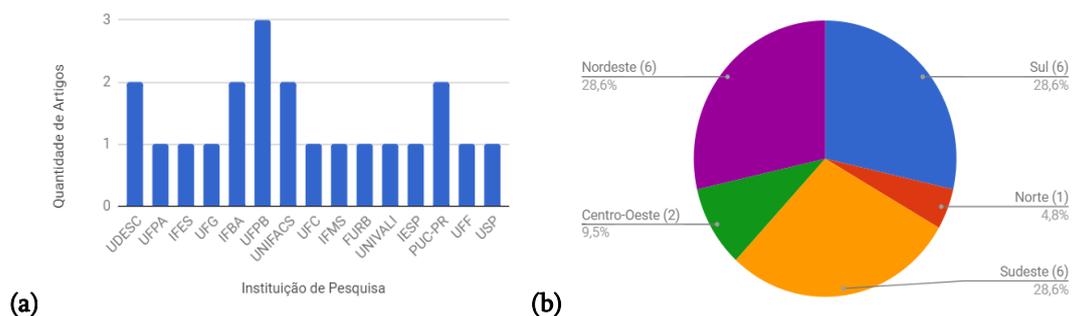


Figura 5. (a) Quantidade de artigos por instituição e (b) Percentual de artigos por região.

3.3 Discussão dos Resultados

Visando responder a Sub-Q6, algumas observações são apontadas. A primeira se trata dos casos reais dos estudos. Nos últimos anos, as oficinas foram utilizadas de forma crescente, partindo de uma ocorrência no período 2008 a 2012 para cinco entre 2013 e 2017. Mesmo a quantidade de casos “durante o ano letivo” sendo maior, o seu uso

decreceu com o passar dos anos, estando mais concentrado no primeiro quinquênio. Novas formas de avaliar os trabalhos surgiram com o passar dos anos, por exemplo, a avaliação por especialista e os estudos experimentais.

Dentre as ferramentas, Unity 3D, Construct 2 e Lego Mindstorm foram as mais utilizadas, sendo empregadas em dois estudos cada. A programação de desafios nesses ambientes é simples, o que pode ser uma das razões para o uso. A robótica educacional figurou em apenas três artigos, mas contribuiu para um aprendizado lúdico construtivo e motivador ao aprendiz, tornando-se um potencial instrumento de interesse no processo. Outro aspecto é o engajamento das regiões em pesquisas sobre jogos educacionais. As instituições no Nordeste apresentaram mais trabalhos em 2013-2017 em relação a 2008-2012. Há também trabalhos das regiões Centro-Oeste e Norte, que não constam entre os estudos do primeiro quinquênio. A região Sul apareceu com menos trabalhos em 2013-2017 e a região Sudeste se manteve ativa. Em relação aos anos das publicações, vale ressaltar que não foram encontrados trabalhos aderentes aos requisitos desta RSL nos anos 2008 e 2014 e que os anos 2010 e 2015 tiveram o maior número (quatro em cada).

Uma limitação enfrentada é a quantidade limitada de artigos no tema desta RSL (20), pois o trabalho se restringiu ao uso de jogos digitais para ensino de programação aos iniciantes em Computação no nível superior na última década. No entanto, os artigos analisados mencionaram esforços dos grupos para liberar novas versões dos jogos (com novas missões e mais aprendizes) bem como no acompanhamento dos alunos em novos experimentos. Observa-se pouco uso de metodologias pedagógicas, demonstrado pela quantidade de trabalhos que não explicitam quais técnicas utilizaram.

Conforme os resultados, essa nova forma de se buscar aprendizado é eficaz quando empregada da maneira certa. Em torno de 90% dos artigos analisados alcançaram o seu propósito, ratificando a afirmação de Victal et al. (2015) e Hoelfmann (2016) sobre a sua importância como instrumento de apoio ao ensino. No entanto, o uso de jogos digitais ainda não é totalmente eficaz. Para Scaico e Scaico (2016), é preciso desenvolver estudos com maior rigor científico e metodológico. Devido à quantidade reduzida de trabalhos na área, percebe-se que o emprego de jogos digitais para o ensino requer mais pesquisas para agregar qualidades técnicas e reforçar a sua utilidade. Acredita-se que jogos digitais são ferramentas para o envolvimento e o aprendizado do aluno e o seu uso deve ser bem pensado, pois os jogos devem auxiliar o tutor no processo de ensino-aprendizagem (e não o substituir).

Neste caso, o desenvolvimento do ProgramSE considerou algumas características das teorias pedagógicas (CON e Tx) que auxiliaram os jogos estudados durante a RSL para embasar a primeira versão de seus desafios e a forma como o conteúdo é apresentado aos estudantes, e o estudo de casos reais para escolher a dinâmica para apresentar o jogo aos alunos da UNIRIO. Além da preocupação pelo conteúdo apresentado, o ProgramSE também foi baseado em elementos de usabilidade para melhorar a experiência dos jogadores e tornar o processo agradável.

4 ProgramSE

Tendo em vista os problemas relacionados à alta evasão dos estudantes do ensino superior em Computação, os elementos catalogados da literatura sobre o engajamento em jogos digitais e seu desenvolvimento e o panorama da utilização de jogos digitais no domínio da educação, o jogo digital para auxiliar o ensino de programação *ProgramSE*¹² foi desenvolvido. Sua construção teve o propósito de cativar estudantes e exercitar o pensamento diante da lógica e conceitos de programação. Neste capítulo, a Seção 4.1 e 4.2 apresentam o desenvolvimento e descrição do jogo, respectivamente; a Seção 4.3 apresenta os níveis de aprendizagem; e a Seção 4.4, os desafios do jogo.

4.1 Desenvolvimento do Jogo

O jogo digital 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 contempla conceitos iniciais a intermediários de lógica de programação. O público foi definido no objetivo da RSL, considerando-se artigos para entender o cenário do ensino de programação no nível superior. A mecânica do jogo foi desenvolvida utilizando o motor de jogos *Construct 3*, que foi escolhido por ser uma plataforma de fácil utilização (conforme visto na Tabela 3). Para os cenários do jogo, utilizou-se o programa para projeção de casas e design de interior *Sweet Home 3D*¹³. O software livre *Inkscape*¹⁴ também foi utilizado 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 desse princípio, o cenário para o uso do *ProgramSE* tem como principal objetivo a Teoria Pedagógica do Construtivismo, onde o aprendiz é autônomo e participa ativamente do próprio aprendizado [Henrique e Silva 2017]. A Taxionomia de *Bloom* também é importante, uma vez que o conteúdo é

¹² <https://bit.ly/2IHRtjZ>

¹³ <http://www.sweethome3d.com/>

¹⁴ <https://inkscape.org/>

exibido de uma forma hierárquica, iniciando do menor nível de complexidade [Henrique e Silva 2017].

Para o desenvolvimento do jogo, foi adotada uma estratégia baseada no modelo ágil, onde eram definidas entregas semanais e tal demanda era influenciada pela elaboração do documento de planejamento do jogo (Anexo 1). Após o fim desta etapa, foi realizada uma avaliação experimental com os alunos das disciplinas Técnicas de Programação 1 (TP1), do curso de Bacharelado em Sistemas de Informação, e Introdução à Ciência da Computação (ICC), do curso de Licenciatura em Matemática, por meio de um questionário, conforme apresentada no próximo capítulo.

4.2 Descrição do Jogo

O ProgramSE é um jogo no estilo *point & click*, ou seja, sua interação ocorre através do ponteiro e do clique do mouse e é baseado em jogos *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, um transtorno de ansiedade que a impede, por medo, de sair de casa. Ao decorrer da trama, a personagem precisa revisitar o passado, através da terapia, e se lembrar do acidente que a fez perder sua família (e a fez desenvolver essa fobia). No jogo, a história conta um enredo no qual o personagem principal sofre de perda de memória e se encontra preso em uma casa que ele não reconhece. O jogador precisa completar as fases para ter o desfecho da história do jogo, sendo similar à do livro. Assim que os desafios são cumpridos com sucesso, a causa da amnésia do jogador vem à tona. O enredo foi definido apenas como um elemento de jogo, não tendo influência educacional para o jogador (apenas para o seu engajamento).

A tela inicial do ProgramSE (o menu) apresenta a opção “Jogar” que, ao ser clicada, dá início ao jogo. A Figura 6 apresenta o menu. A finalidade 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 daquele 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 evolui de fases, novos comandos de programação são apresentados em níveis de dificuldade mais elevados.



Figura 6. Tela de menu do jogo ProgramSE

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. Eles 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 Nível de Aprendizagem

O jogo foi desenvolvido visando englobar o conteúdo programático das disciplinas introdutórias de programação TP1 e ICC, sendo eles: (1) Sequência; (2) Seleção; (3) Repetição; (4) Função e Procedimento; (5) Recursividade; (6) Vetor e Matriz; e (7) Ordenação. Sua exposição é dividida em três níveis, tendo cada um suas próprias fases. A relação dos conceitos apresentados em cada parte do ProgramSE é apresentada na Tabela 8.

Tabela 8. Relação Nível x Fase x Conceitos do jogo 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	Entrada	Recursividade

Nível 1: O objetivo dessa fase é iniciar o aprendiz no pensamento lógico, começando o estudo pelo conceito principal da programação; a estruturação do algoritmo (e.g. Sequência) que, segundo Niklaus Wirth, é a arte ou técnica de construir e formular algoritmos de uma forma sistemática. 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. Sua importância 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 no nível servem para o aperfeiçoamento do algoritmo, sendo eles: (1) comandos de controle (e.g. Seleção), i.e. estruturas condicionais que realizam testes lógicos antes de executar ações; e (2) comandos de repetição (e.g. Repetição), i.e. estrutura que executa repetidamente um bloco de ações. Na Figura 7, as fases que compõem o Nível 1 são apresentadas.

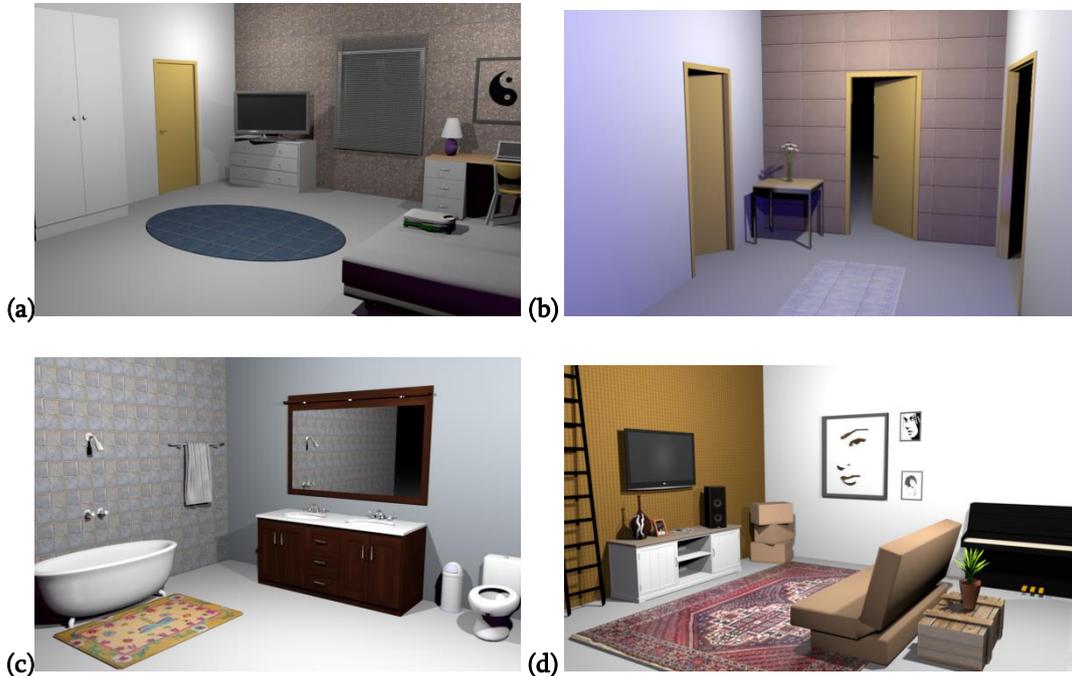


Figura 7. (a) Quarto; (b) Corredor; (c) Banheiro; e (d) Sótão.

Nível 2: Esse nível tem a responsabilidade de apresentar ao estudante estruturas de programação mais complexas. Por entender que após o Nível 1 o estudante aprendeu 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), i.e. blocos de comandos que, ao serem chamados, serão executados e retornarão algum resultado; (3) estruturas de dados (e.g. Vetor), i.e. uma estrutura de dados unidimensional para armazenar elementos do mesmo tipo; e (4) algoritmos de ordenação, i.e. 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 dos mesmos. O principal objetivo dessa fase é capacitar o estudante a, além de saber como programar tais problemas, entender ludicamente como é sua representação. A Figura 8 apresenta os cenários das fases do Nível 2.

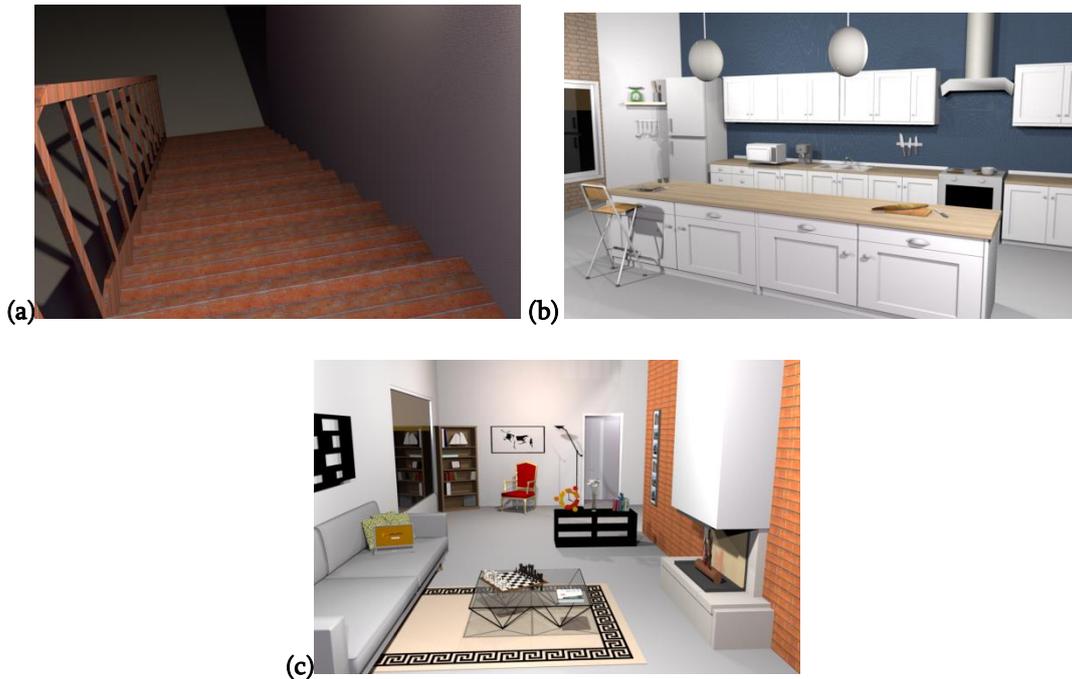


Figura 8. (a) Escada; (b) Cozinha; e (c) Sala de Estar.

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 9 mostra a única fase do Nível 3.



Figura 9. Entrada.

4.4 Desafios

Os desafios apresentados no jogo são baseados na programação em blocos, como o *Scratch* e o *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 aquele desafio. Em algumas interações, o jogador pode receber itens colecionáveis (que são armazenados em um inventário) que serão necessários para objetos futuros. Na Figura 10, 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 a definição apresentada naquele momento. Como os conceitos são repetidos no percorrer do jogo e 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.

Os desafios são progressivos, ou seja, 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 da disciplina lhe é apresentado. Estas dicas foram retiradas do material de aulas de ICC (no ano de 2017, ministradas pelo professor Rodrigo Santos) e validadas por profissionais na área.

Os conceitos apresentados no jogo (Sequência, Seleção, Repetição, Modularização, Estrutura de Dados, Ordenação e Recursividade), a forma como são apresentados no jogo e suas respectivas dicas são:

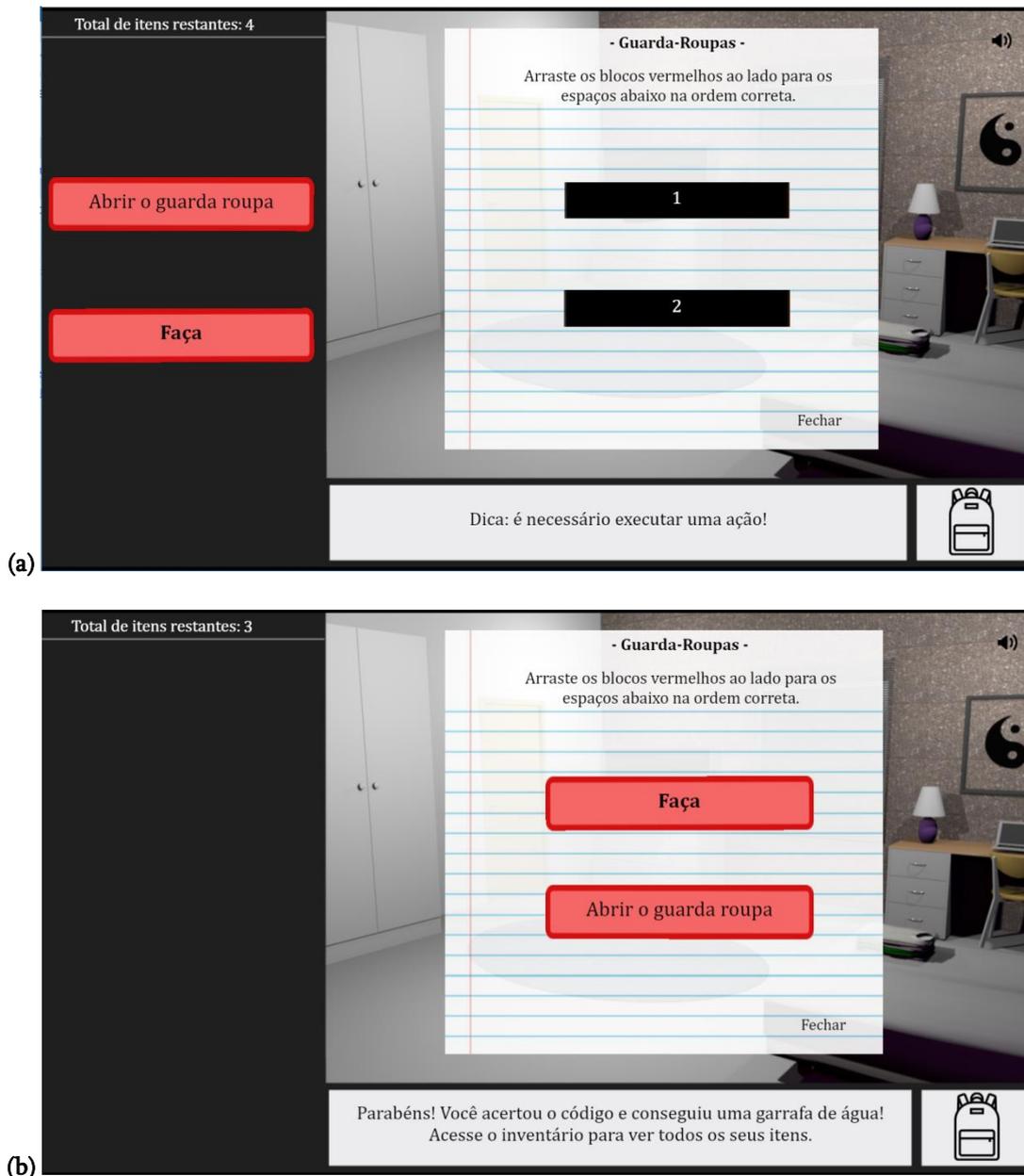


Figura 10. Exemplo de Sequência.

Sequência: Uma sequência simples é um conjunto de ações a serem executadas de forma estruturada e sequencial. No jogo, é o primeiro conceito apresentado ao estudante. A dica disponibilizada é “*é necessário executar uma ação!*” e a Figura 10 (b) apresenta um exemplo de como o conceito é apresentado.

Seleção:

- **Se:** Quando a ação a ser executada depender de uma inspeção, haverá uma estrutura condicional, simples ou composta, que, caso seja verdadeira, a ação

será executada. A dica disponível no jogo é “é necessário checar uma condição para que a ação seja executada!” e a Figura 11 mostra um exemplo do conceito.

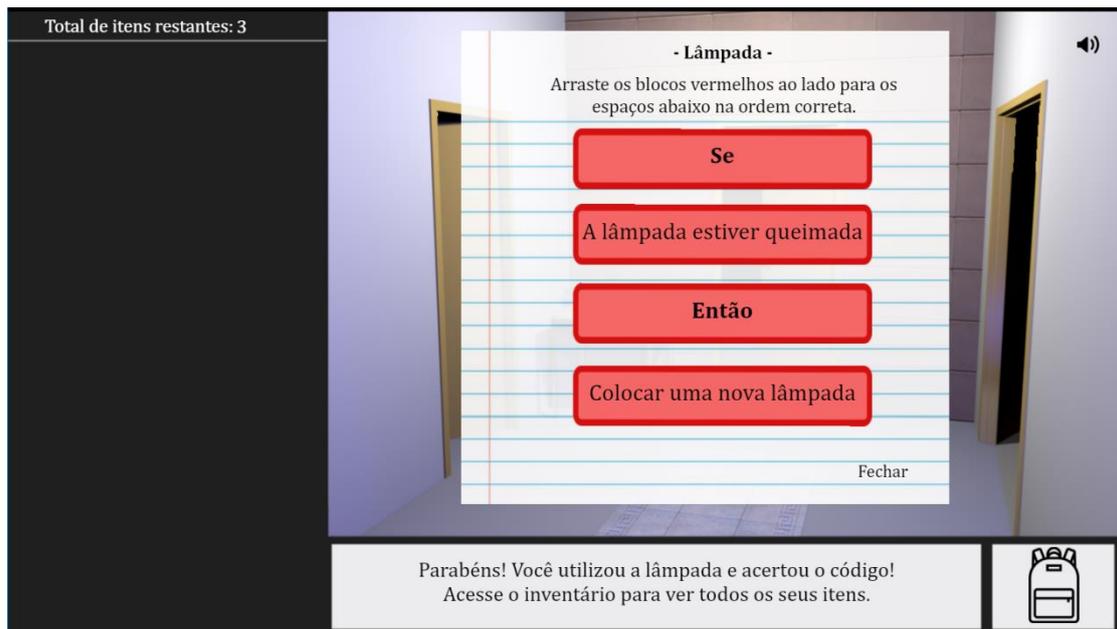


Figura 11. Exemplo de Se.

- **Se/Senão:** A estrutura é a mesma apresentada no 'Se'. Porém, a condição testada não for válida, o programador poderá determinar outra ação a ser executada em contrapartida. Sua dica é “é necessário checar uma condição para que a ação seja executada, caso contrário, deve-se executar outra ação!” e a Figura 12 exibe um exemplo.

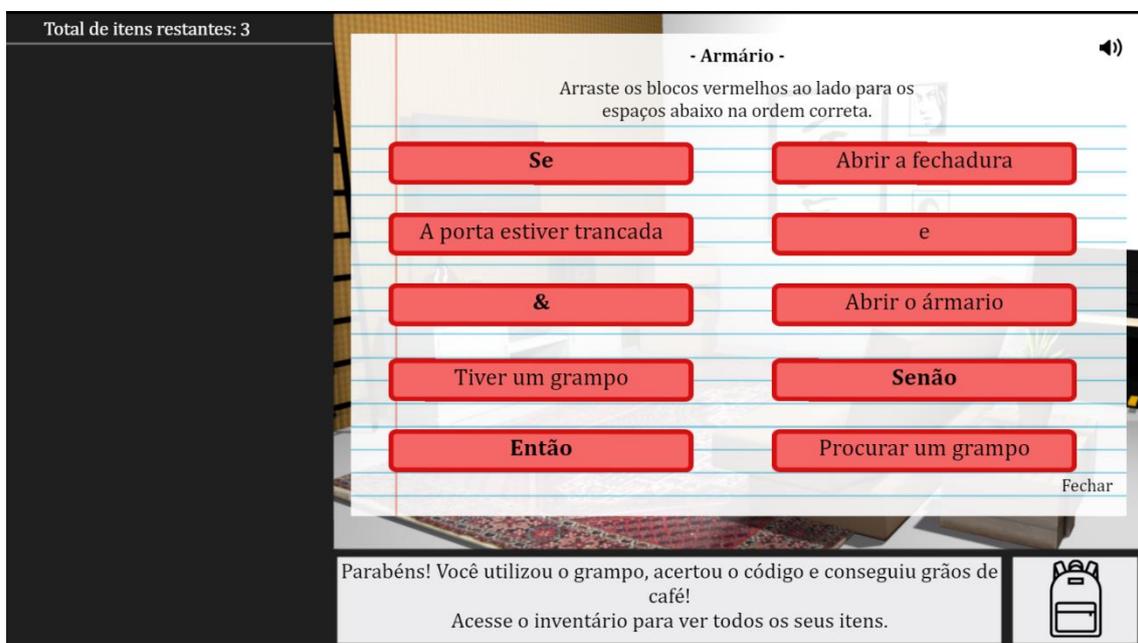


Figura 12. Exemplo de Se/Senão.

- **Caso:** Essa estrutura é similar à semântica do Se/Senão, porém é particularmente interessante para a construção de algoritmos que executem várias funções e o usuário precise determinar qual operação deva ser executada. A dica apresentada nessa interação é “*é necessário executar diferentes ações dependendo de um estado inicial!*” e sua representação é apresentada na Figura 13.

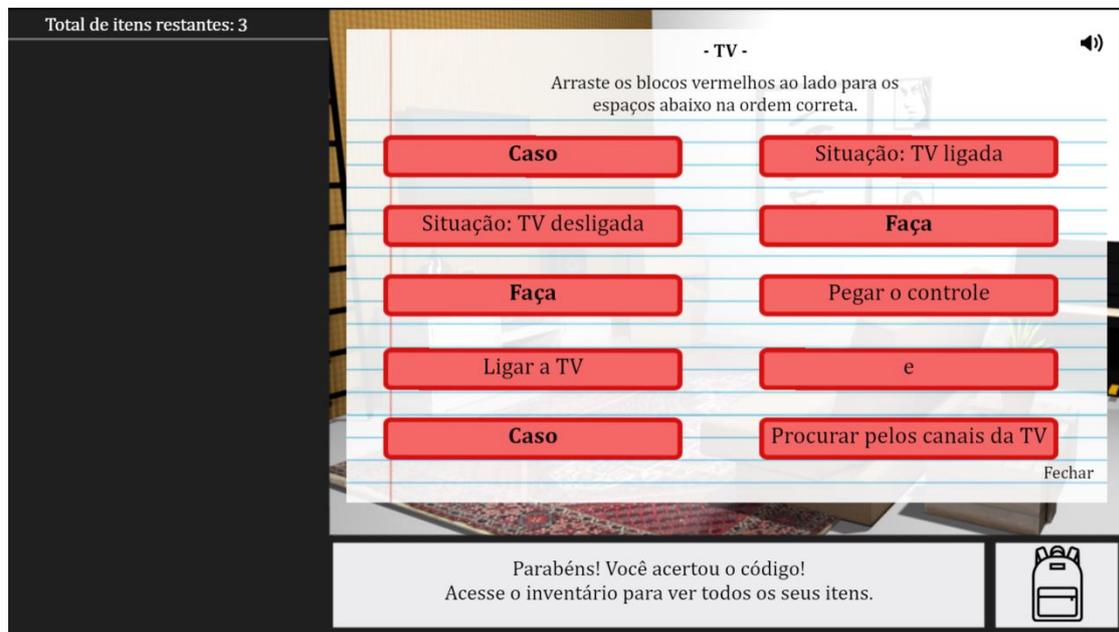


Figura 13. Exemplo de Caso.

Repetição:

- **Enquanto:** Uma estrutura de repetição acontece quando um conjunto de ações é executado repetidamente enquanto uma determinada condição permanece válida. A dica disponível no jogo é “*é necessário realizar uma ação repetidamente enquanto determinada condição permanece válida!*” e a Figura 14 apresenta um exemplo do conceito.
- **Para/Passo:** A estrutura de repetição *para* difere da estrutura *enquanto* por utilizar uma variável de controle que atua como um contador de repetições. Ela possui um incremento próprio. No entanto, no jogo, o contador é apresentado em momentos diferentes (para facilitar o entendimento do conceito da estrutura). Sua dica é “*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!” e a Figura 15 traz um exemplo.

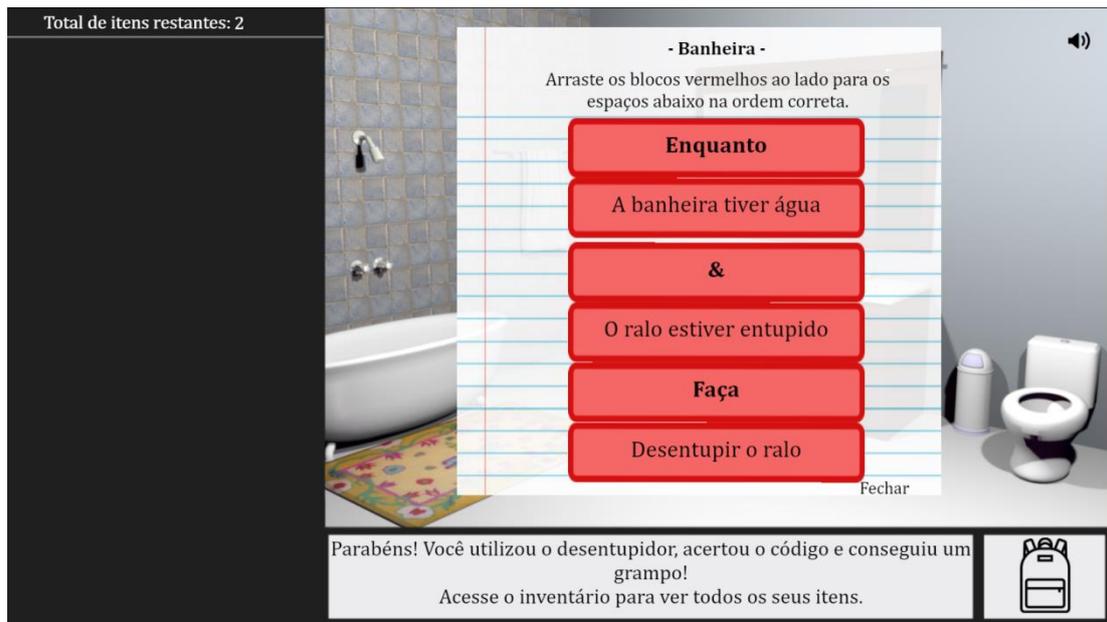


Figura 14. Exemplo de Enquanto.

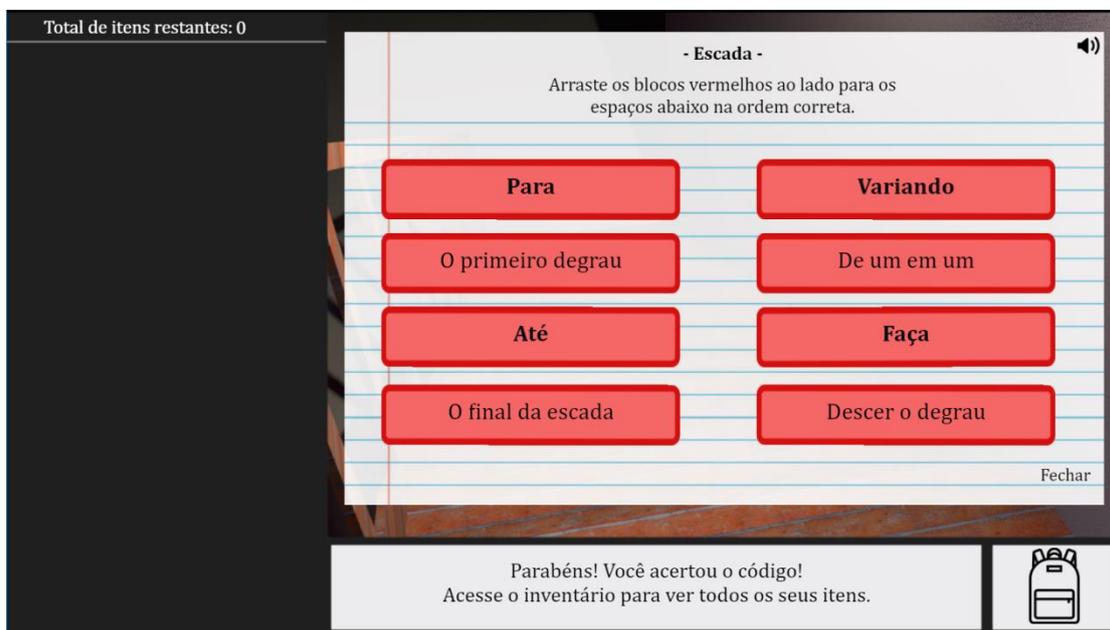


Figura 15. Exemplo de Para/Passo.

Modularização:

- **Função:** As *funções*, também conhecidas como sub-rotinas, são utilizadas na programação para criar um bloco de comandos que serão executados diversas vezes (e em diversos lugares diferentes). Um dos grandes benefícios é não precisar copiar o código todas as vezes que precisar executar aquela operação, além de deixar a leitura do código mais intuitiva. A dica exibida para o conceito é “é necessário montar um bloco de comandos que, ao serem chamados, serão

executados e retornarão algum resultado!”. Por ser um conceito mais complexo de ser entendido, além da dica também é exibido um *obs.*, i.e. observação, para facilitar o entendimento do código - “*Os valores recebidos pela função são chamados de Parâmetros; e o valor retornado, Retorno*”. A Figura 16 apresenta um exemplo de sua utilização.

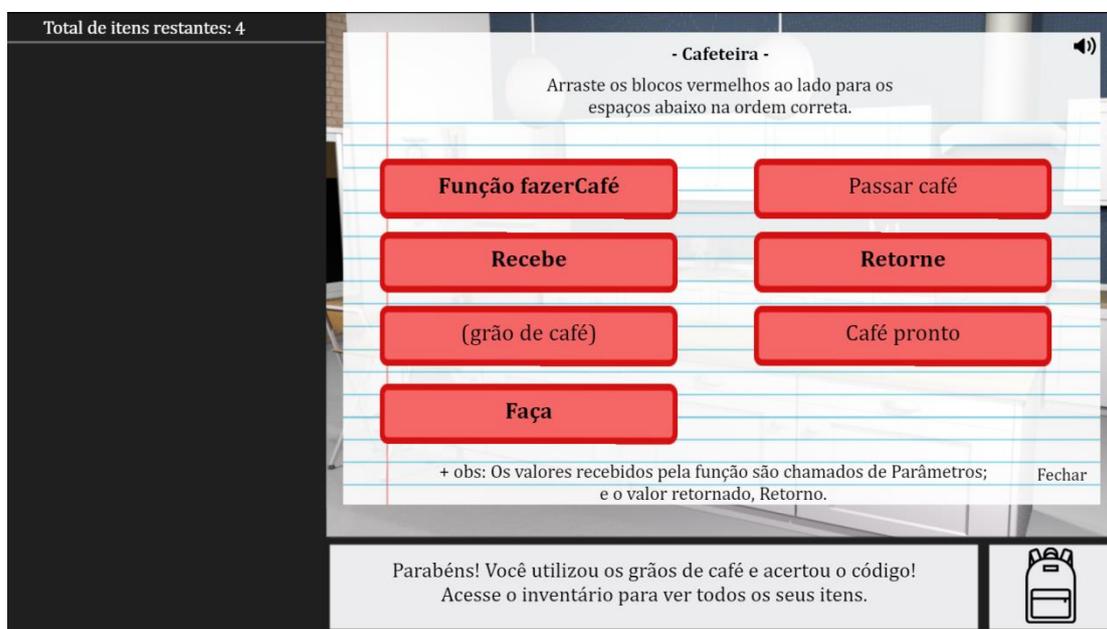


Figura 16. Exemplo de Função.

- **Procedimento:** Os *procedimentos* diferem das *funções* apenas por não retornarem resultado. Sua dica e seu *obs.* são, respectivamente, “*é necessário montar um bloco de comandos que, ao serem chamados, serão executados e não retornarão nada!*” e “*Os valores recebidos pelo procedimento são chamados de Parâmetros; e diferente da Função, o Procedimento não possui retorno*”. A Figura 17 apresenta um exemplo de *procedimento*.

Estrutura de Dados:

- **Vetor:** O *vetor* é uma variável que armazena vários valores do mesmo tipo. Sua dica para o jogador é “*é necessário criar uma estrutura de dados unidimensional, e percorrê-la, para armazenar itens do mesmo tipo!*”. Da mesma forma que a modularização, também apresenta um *obs.*, “*os elementos armazenados no vetor são acessados através de um índice, que é representado ao percorrer cada coluna do mesmo*”. Por serem estruturas ludicamente complicadas de explicar, os objetos referentes a tal tópico só poderiam ser

acessados após o jogador clicar no objeto *livro*, que contém uma explicação textual. Na Figura 18, estão dispostos o *livro* referente a *vetor* e um exemplo de sua utilização.

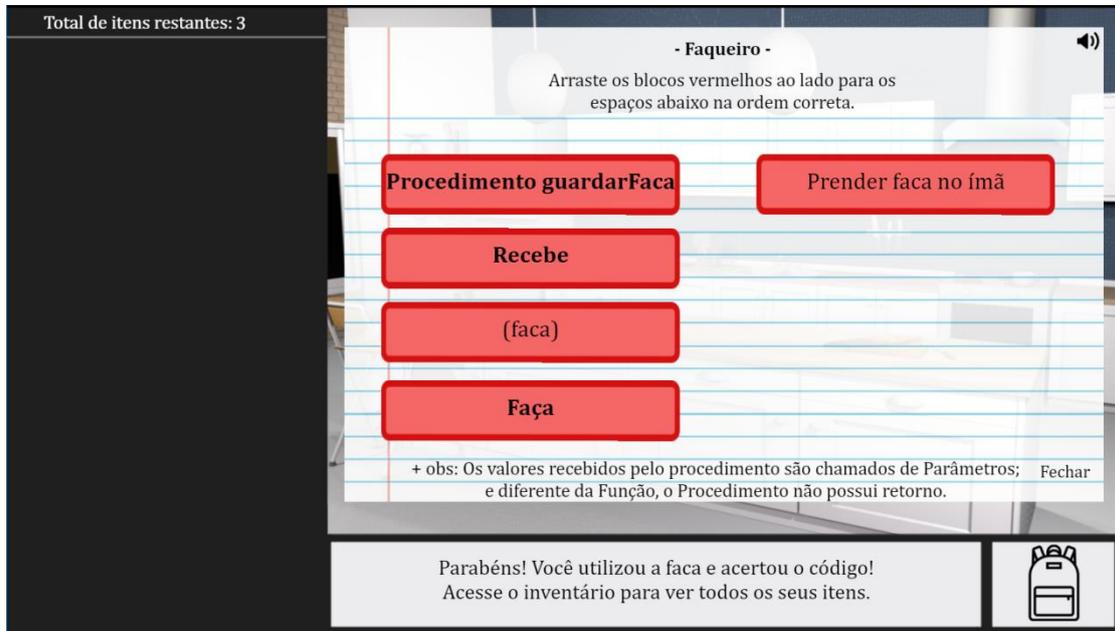
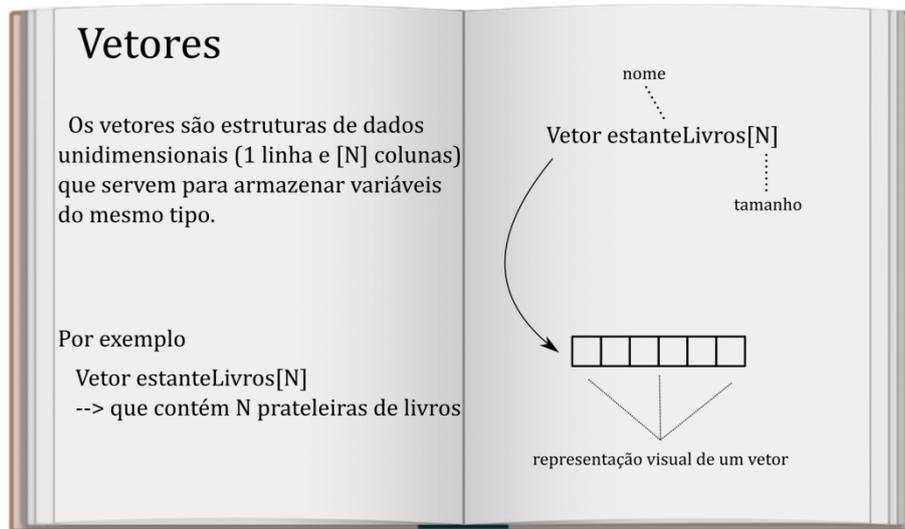


Figura 17. Exemplo de Procedimento.

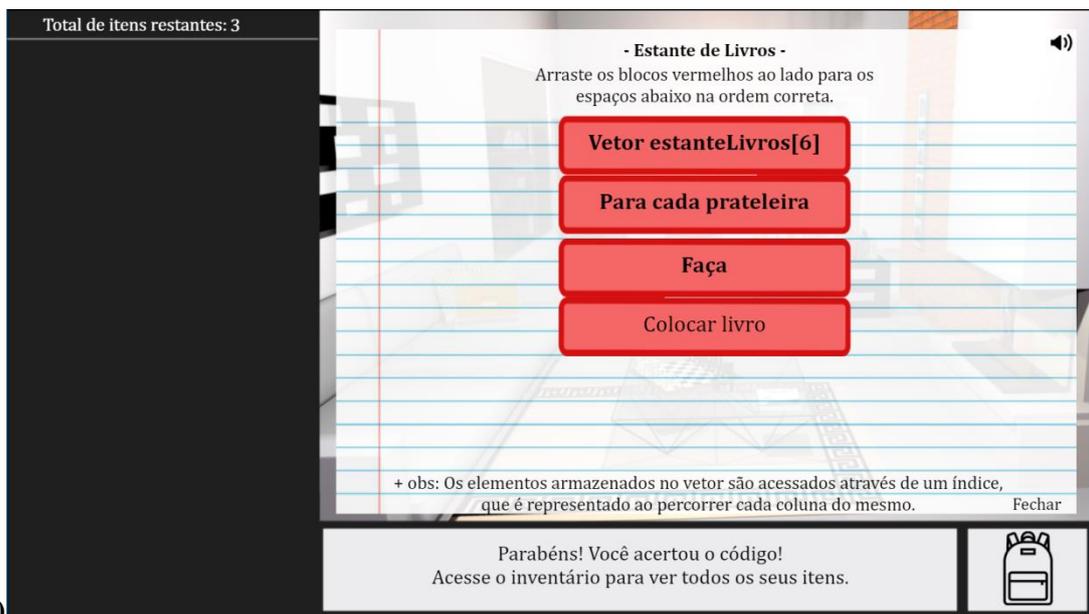
- **Matriz:** A *matriz* é um *vetor* que armazena outros *vetores*. Sua dica para o jogador é “é necessário criar uma estrutura de dados bidimensional, e percorrê-la, para armazenar itens do mesmo tipo!”. Da mesma forma que a modularização, também apresentam um *obs.*, “os elementos armazenados na matriz são acessados através de um índice, que é representado ao percorrer cada linha e coluna do mesmo”. Assim como os *vetores*, os objetos referentes a tal tópico só poderiam ser acessados após o jogador clicar no objeto *livro*, que contém uma explicação textual. Na Figura 19, estão dispostos o *livro* referente à *matriz* e um exemplo de sua utilização.

Ordenação: A atividade de *ordenação* é o processo de rearranjo de certo conjunto de objetos de acordo com um critério (ordem) específico. A ordenação pode ser feita de diversos métodos: InsertionSort, BubbleSort, QuickSort, SelectionSort etc.

Como, para esse conceito, o principal é o entendimento do fluxo e não o algoritmo em si, sua apresentação é como o conteúdo do *livro*, apenas explicando teoricamente. A Figura 20 demonstra o terceiro capítulo do *livro*, sobre *ordenação*.

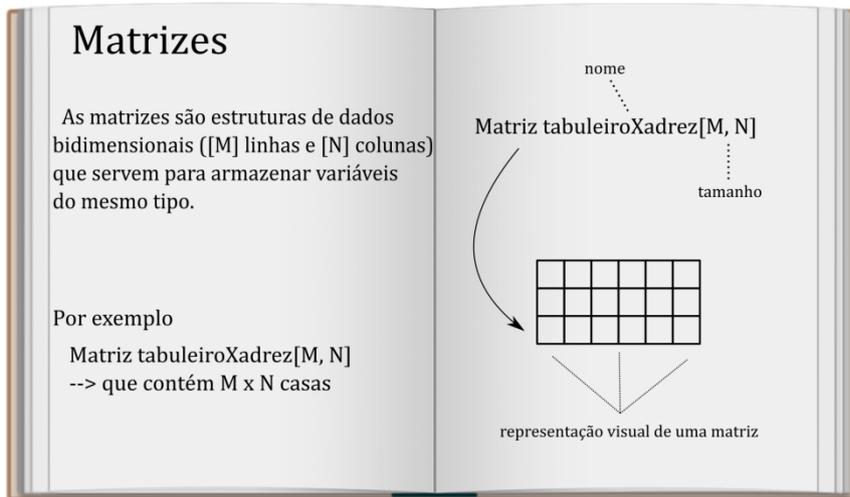


(a)

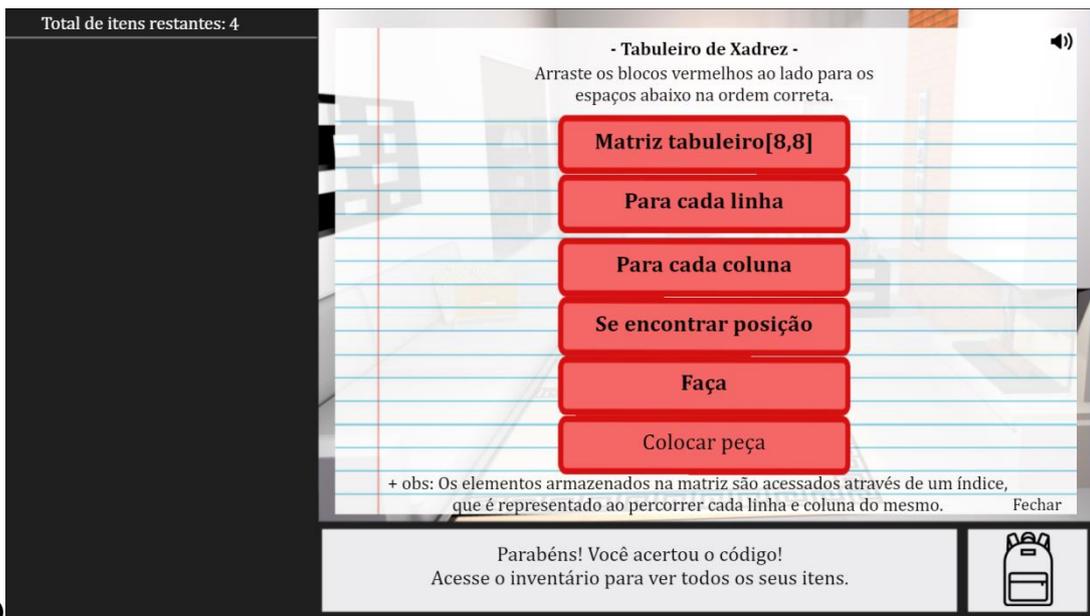


(b)

Figura 18. (a) Livro - Vetores; (b) Exemplo de Vetor.



(a)



(b)

Figura 19. (a) Livro - Matrizes; (b) Exemplo de Matriz.

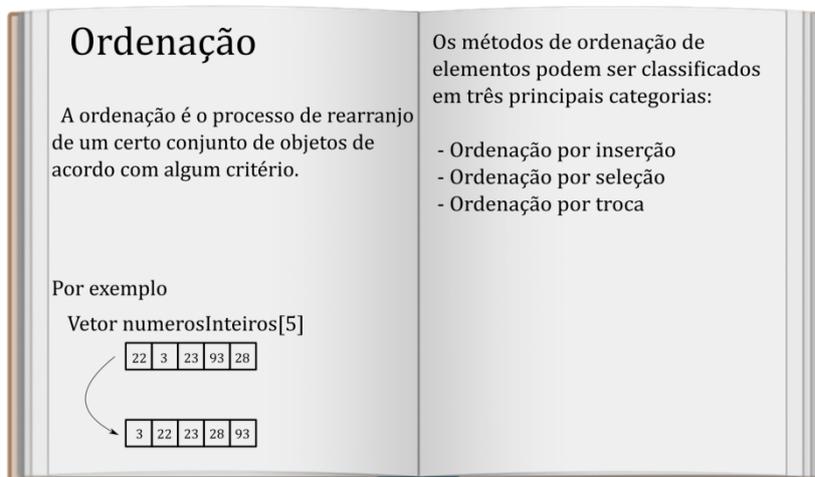


Figura 20. Livro - Ordenação.

Recursividade: A *recursividade* é o último conceito apresentado no jogo. Uma *função* é recursiva quando, no corpo dessa *função*, existe uma chamada a si própria, podendo utilizar os mesmos parâmetros de entrada (correndo riscos de provocar um ciclo infinito) ou outros. A dica apresentada para o jogador é “*é necessário criar uma função que contenha uma chamada a si próprio!*”.

Por ser um conceito bem abstrato, foi criada uma nova dinâmica no jogo para esta estrutura. O jogador está em um ambiente que contém apenas uma porta e, sempre que selecionada, o jogador “volta” para o mesmo ambiente. Em determinado ponto, aparece um elemento na tela (condição de parada da recursividade) que, quando selecionado, apresenta o desafio da fase. Ao concluí-lo, o jogador pode sair do cômodo. A Figura 21 mostra como a estrutura do código é apresentada.

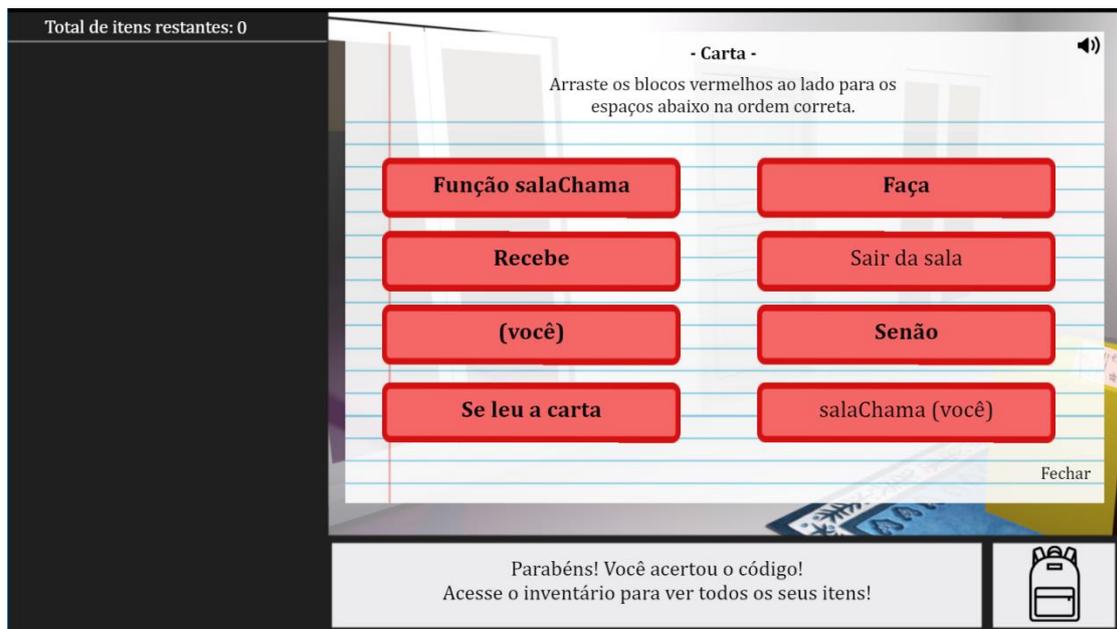


Figura 21. Exemplo de Recursividade.

Os demais exemplos sobre a forma em que os conceitos foram introduzidos no jogo são apresentados no Anexo 1.

5 Avaliação Experimental

Após a construção do jogo *ProgramSE*, o processo de avaliação foi realizado por meio de um experimento a fim de verificar se os objetivos foram atingidos com o seu uso no contexto de sala de aula. Com a aplicação e uso do jogo, foi possível avaliar a sua usabilidade e eficácia; além de ser possível verificar se as decisões tomadas durante o planejamento acerca dos elementos do jogo foram satisfatórias sob o ponto de vista dos estudantes. Os resultados obtidos são apresentados neste capítulo, a Seção 5.1 apresenta o planejamento; a Seção 5.2, a execução do processo; e, nas Seções 5.3 e 5.4, os resultados são apresentados e discutidos, respectivamente.

5.1 Planejamento

Após o desenvolvimento do jogo *ProgramSE*, o questionário modelo do MEEGA+ para jogos educacionais digitais foi customizado. Foi criada uma versão digital do questionário utilizando o Google Forms¹⁵ e acrescentado um termo de consentimento livre e esclarecido. O Anexo 2 mostra o questionário utilizado do MEEGA+ especificado para avaliar o *ProgramSE* na etapa exploratória. Por ser um jogo *single player*, o jogo não atende aos requisitos de interação social no modelo do questionário. Portanto, não foram incluídas na análise as perguntas associadas à interação dos participantes com os outros jogadores. O questionário final foi composto por vinte e nove (29) questões.

O questionário MEEGA+ foi escolhido para esta etapa por ser um modelo sistemático para avaliar a qualidade de jogos educacionais utilizados para o ensino de conceitos de programação. Suas respostas são padronizadas, variando de -2 (discordo fortemente) até +2 (concordo fortemente). Dessa forma, quando a maioria das respostas de determinada característica avaliada se aproximar de +2, significa que o tópico obteve uma avaliação positiva; caso contrário, uma avaliação negativa, caso se aproxime de -2.

¹⁵ <https://www.google.com/forms/about/>

Foi realizada uma avaliação piloto do jogo a fim de avaliar o ProgramSE por uma especialista na área de informática antes do contato com os estudantes. O piloto fomentou discussões sobre os conceitos escolhidos para a ferramenta e sobre alguns elementos de jogos (e.g. enredo e *feedbacks*). 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 (Figura 13); e (5) mudar o termo técnico "Passo" para "Variando" nos exemplos do comando de repetição Para. Com as recomendações obtidas no piloto, o ProgramSE pode ser refinado para o experimento em si.

O estudo com os estudantes ocorreu em maio de 2019 com o *ProgramSE* sendo adotado na turma de TP1 (UNIRIO – Bacharelado em Sistemas de Informação) e ICC (UNIRIO – Licenciatura em Matemática). Ambas as disciplinas abordam ensino de programação para os iniciantes de seus respectivos cursos. No entanto, os estudantes possuem diferentes perfis, visto que a turma de ICC é formada por alunos que não são da área de Computação. A escolha dessa turma para a aplicação do experimento foi importante para avaliar a utilidade e a eficácia da aprendizagem com uso do ProgramSE por estudantes fora do contexto da programação, ou seja, alunos provenientes de outras áreas de conhecimento. A avaliação foi mediada pelo autor deste trabalho e acompanhada por duas instrutoras da área de informática.

5.2 Execução

Após o planejamento, a avaliação do jogo *ProgramSE* ocorreu no dia 09 de maio de 2019 e contou com a participação de 37 alunos, sendo 23 da disciplina de TP1 e 14 de ICC. Cada avaliação durou aproximadamente 45 minutos, reservando 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 TP1 e 14 de ICC.

No início da avaliação, o objetivo geral do trabalho foi apresentado, uma explicação sobre o jogo *ProgramSE* foi realizada e os estudantes foram orientados sobre o processo. Em seguida, foi disponibilizado o *link* para o formulário online, que é 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 parte do questionário consiste em perguntas sobre informações demográficas, i.e. faixa etária e o sexo. A frequência com que o estudante costuma jogar também é importante e questionada nesse passo. A terceira etapa da avaliação foi realizada após a utilização do *ProgramSE*, pois consiste no 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 o aprimoramento do jogo. A Figura 22 é uma foto tirada no momento da execução da avaliação experimental.



Figura 22. (a) Avaliação em TP1; (b) Avaliação em ICC.

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 TP1 ocorreu das 17h27min até às 18h08min, durando 41 minutos. Enquanto isso, na aula de ICC, a avaliação durou 49 minutos, ocorrendo das 20h47min às 21h36min. Ambas as avaliações foram feitas nos laboratórios de aula das disciplinas no prédio de Centro de Ciências Exatas e Tecnologia (CCET) da UNIRIO.

5.3 Análise dos Dados

Com todos os dados coletados, iniciou-se a etapa de análise. As respostas foram processadas e alguns gráficos foram gerados para evidenciar e identificar acertos e pontos de melhorias do jogo. Como as avaliações ocorreram em turmas diferentes, os gráficos são exibidos separados nesta seção.

Técnicas de Programação 1 (TP1): Na disciplina TP1, 18 alunos participaram da pesquisa e responderam ao questionário. Como a primeira etapa do estudo envolveu perguntas para entender o cenário demográfico da turma, a Figura 23 exibe os gráficos obtidos através do processamento de tais dados. 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 uma pessoa, de 29 a 39 anos. Não tiveram participantes da pesquisa acima dos 40 anos em TP1.

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 sobre a utilização de 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 informaram que os utilizam diariamente; 4 pessoas, semanalmente; 2 pessoas responderam que utilizam raramente; uma pessoa não utiliza jogos digitais e nenhuma pessoa selecionou a opção mensalmente. Em relação ao uso de jogos não-digitais, os resultados foram bem diferentes dos primeiros. Nenhuma pessoa da amostra os utiliza diariamente; uma pessoa faz uso semanal; 6 pessoas utilizam jogos não-digitais mensalmente; 8 pessoas raramente os jogam; e 3 pessoas nunca utilizam esses tipos de jogos.

A próxima etapa do questionário corresponde a questões sobre a usabilidade do jogo. Ou seja, aspectos visuais e princípios percebidos sobre a eficiência do ProgramSE. No questionário, as afirmações foram agrupadas nos seguintes aspectos: Estética, Aprendizabilidade, Operabilidade e Acessibilidade. A Figura 24 informa o gráfico obtido das respostas sobre o tema.

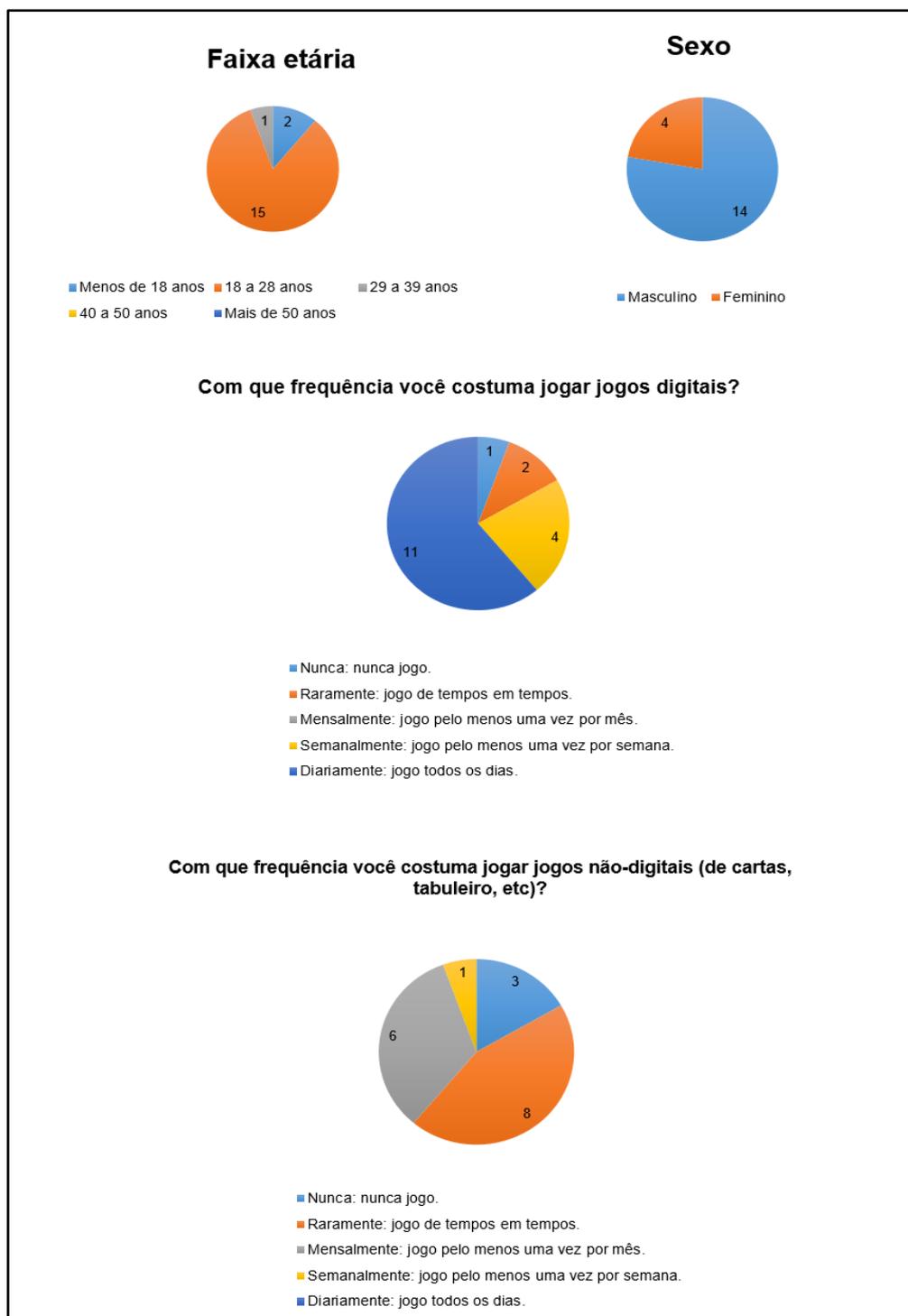


Figura 23. Informações demográficas (TP1).

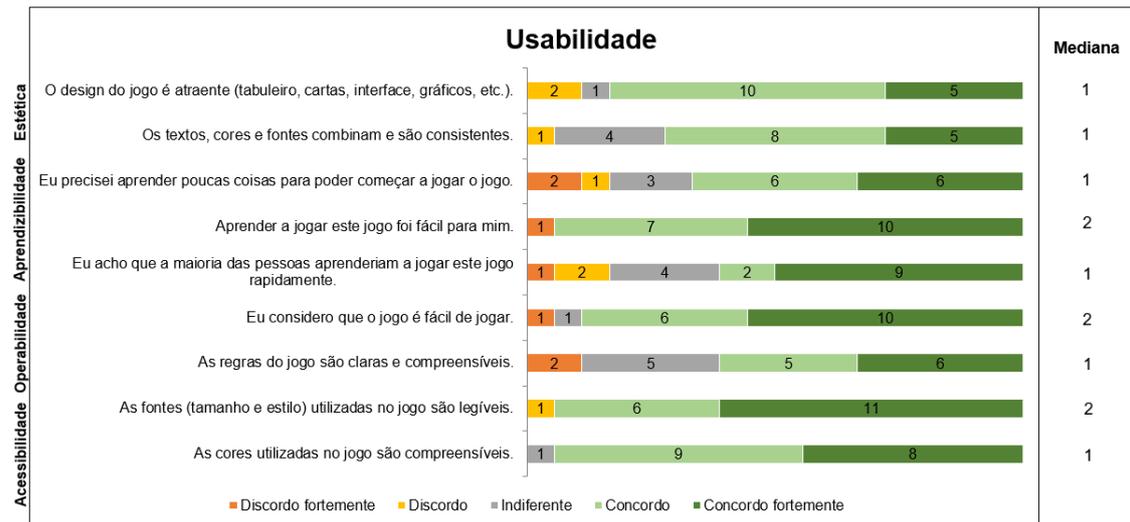


Figura 24. Usabilidade (TP1).

Analisando o gráfico, pode-se inferir que todos os aspectos avaliados obtiveram um bom resultado (estética, aprendizibilidade, operabilidade, acessibilidade). Como todas as respostas tendem a algum nível de concordância mais do que discordância, em termos de usabilidade o jogo ProgramSE obteve retorno positivo dos alunos de TP1. Isto revela que os participantes se sentiram confortáveis em utilizar o programa.

O aspecto de Acessibilidade recebeu a avaliação mais alta, obtendo a maior parte das avaliações 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. Em contra partida, o aspecto que obteve a menor nível de concordância foi a Aprendizabilidade. Ou seja, os tópicos referentes à facilidade em aprender a jogar o ProgramSE obtiveram menos avaliações positivas que as demais. Em relação aos aspectos restantes, a Estética obteve 50% de nível de concordância (+1) entre as respostas e 28% de concordância fortemente (+2). O aspecto Operabilidade obteve 30% de nível de concordância (+1) e 44% de concordância fortemente (+2). A Estética do jogo diz respeito ao seu design e a Operabilidade é sobre a clareza quanto às regras e funcionalidades do jogo.

Outros aspectos avaliados foram referentes à experiência do jogador (confiança, desafio, satisfação, diversão, atenção focada, relevância, percepção de aprendizagem), ilustrados na Figura 25.

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 concordância fortemente (+2). Porém, também não possuem avaliações negativas (-1 ou -2). No geral,

as avaliações positivas das experiências dos jogadores foram menores que as apresentadas na usabilidade (Figura 24).



Figura 25. Experiência do Jogador (TP1).

O aspecto com a pontuação mais alta é a Relevância do jogo, com 37% de concordância (+1) e 22% de concordância fortemente (+2). O tópico é referente à relevância do conteúdo apresentado no ProgramSE para os interesses do aluno e da disciplina. Os aspectos com menores 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), são eles: Confiança, Satisfação, Percepção de Aprendizagem e Diversão. Vale ressaltar 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 concordância fortemente (+2).

Os participantes também apresentaram sugestões e/ou melhorias que eles identificaram para o progresso do jogo. Essa etapa consistiu no quarto passo do questionário, as questões discursivas. Essas informações foram processadas e consolidadas no gráfico da Figura 26.

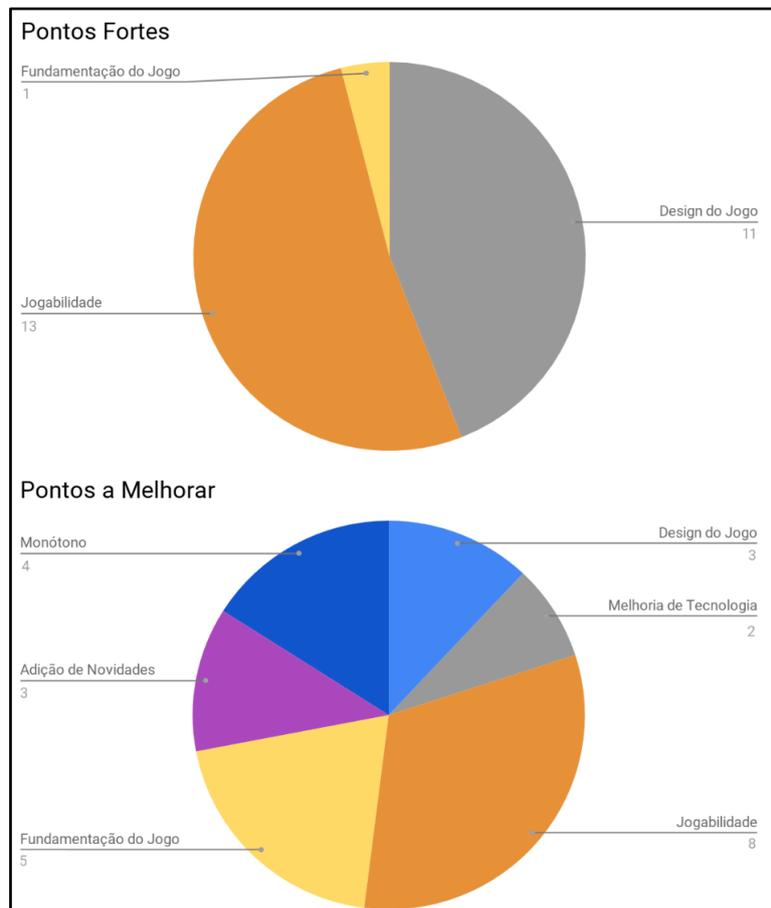


Figura 26. Questões Discursivas (TP1).

Entre os pontos fortes, o mais citado entre os participantes foi em relação à Jogabilidade do ProgramSE, i.e. o fluxo de progressão do jogo, sendo 13 comentários feitos. Os outros tópicos que surgiram dentre os pontos fortes foram o Design do Jogo, com 11, e a Fundamentação do Jogo, i.e. a formatação e a forma como os conceitos de programação foram ensinados (e.g. os comandos e as 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; Adição de Novidades, i.e. novas fases e novas dinâmicas no jogo, com 3; e, por último, Melhoria de Tecnologia, com 2 comentários.

Introdução à Ciência da Computação (ICC): Como a avaliação do jogo ProgramSE foi feita em duas disciplinas diferentes, a segunda parte da análise de dados é referente aos 14 alunos participantes de ICC. O processo do estudo ocorreu da mesma forma que em TP1 e a Figura 27 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. Apenas uma pessoa dentre os menores de 18 anos, uma de 29 a 39 anos e uma maior de 50 anos. O restante, 2 pessoas com idade entre 40 e 50 anos.

A segunda análise feita é referente ao sexo dos participantes. Os participantes do gênero masculino totalizam 13 pessoas. Houve apenas uma pessoa do gênero feminino.

Os dois últimos gráficos, assim como em TP1, 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 jogar jogos digitais, 6 pessoas informaram que os utilizam diariamente; 3 pessoas, semanalmente; 4 pessoas responderam que raramente utilizam; uma pessoa não utiliza jogos digitais e nenhuma pessoa selecionou a opção mensalmente. Em relação ao uso de jogos não-digitais, os resultados foram bem diferentes dos primeiros. Nenhuma pessoa da amostra os utiliza diariamente e ninguém selecionou a opção nunca; uma pessoa faz uso semanal; 6 pessoas utilizam jogos não-digitais mensalmente; e 7 pessoas raramente utilizam esses tipos de jogos.

A próxima etapa do questionário corresponde a questões sobre a usabilidade do jogo. Ou seja, aspectos visuais e princípios percebidos sobre a eficiência do ProgramSE. No questionário, as afirmações foram agrupadas nos seguintes aspectos: Estética, Aprendizabilidade, Operabilidade e Acessibilidade. A Figura 28 informa o gráfico obtido das respostas sobre o tema.

Analisando o gráfico, pode-se inferir que todos os aspectos avaliados também obtiveram um bom resultado (estética, aprendizagem, operabilidade, acessibilidade). 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 ICC. Isto revela que os participantes se sentiram confortáveis em utilizar o programa.



Figura 27. Informações Demográficas (TP1).

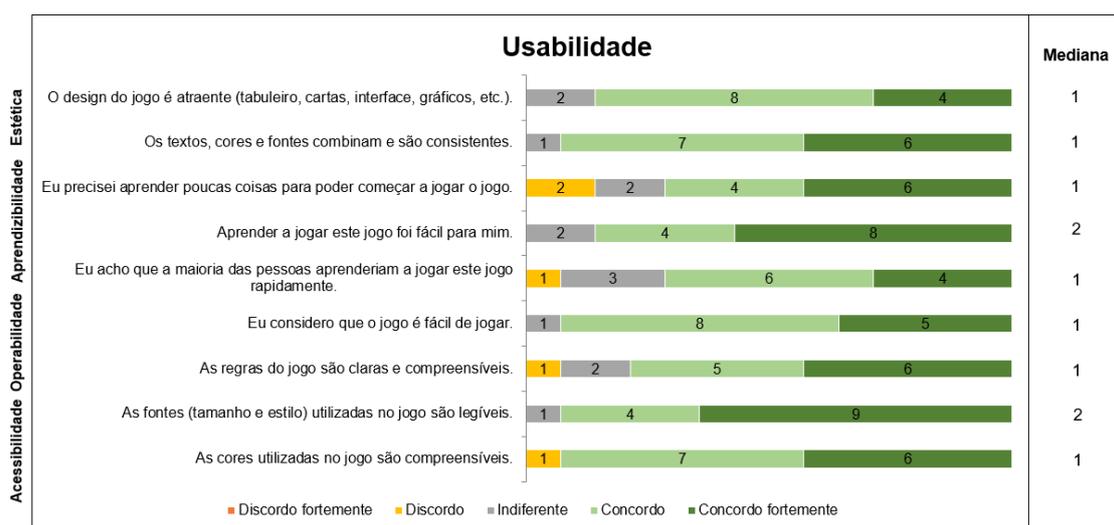


Figura 28. Usabilidade (ICC).

O aspecto de Acessibilidade também recebeu a avaliação mais alta para essa turma; obteve a maior parte das avaliações 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. Em contrapartida, o aspecto que obteve o menor nível de concordância nos itens avaliados foi a Aprendizabilidade. Ou seja, os tópicos referentes à facilidade em aprender a jogar o ProgramSE obtiveram menos avaliações positivas que as demais. Em relação aos aspectos restantes, a Estética obteve 42% de nível de concordância (+1) entre as respostas e 28% de concordância fortemente (+2). O aspecto Operabilidade obteve 36% de nível de concordância (+1) e 30% de concordância fortemente (+2). A Estética do jogo diz respeito ao seu design e a Operabilidade é sobre a clareza quanto às regras e funcionalidades do jogo. Em comparação com TP1, a quantidade de avaliações positivas sobre a Usabilidade do ProgramSE foi maior em ICC.

Outros aspectos avaliados foram referentes à experiência do jogador (confiança, desafio, satisfação, diversão, atenção focada, relevância, percepção de aprendizagem), ilustrados na Figura 29.

A avaliação sobre a experiência dos jogadores em ICC obteve, no geral, algum nível de concordância (+1) e, em algumas afirmativas, prevaleceu a neutralidade dos estudantes (0). Nesse gráfico, foram encontradas respostas tendendo a algum nível de concordância (+1 e +2) e também não possuem avaliações negativas (-1 ou -2). No geral, as avaliações positivas das experiências dos jogadores foram menores que as apresentadas na usabilidade (Figura 28).

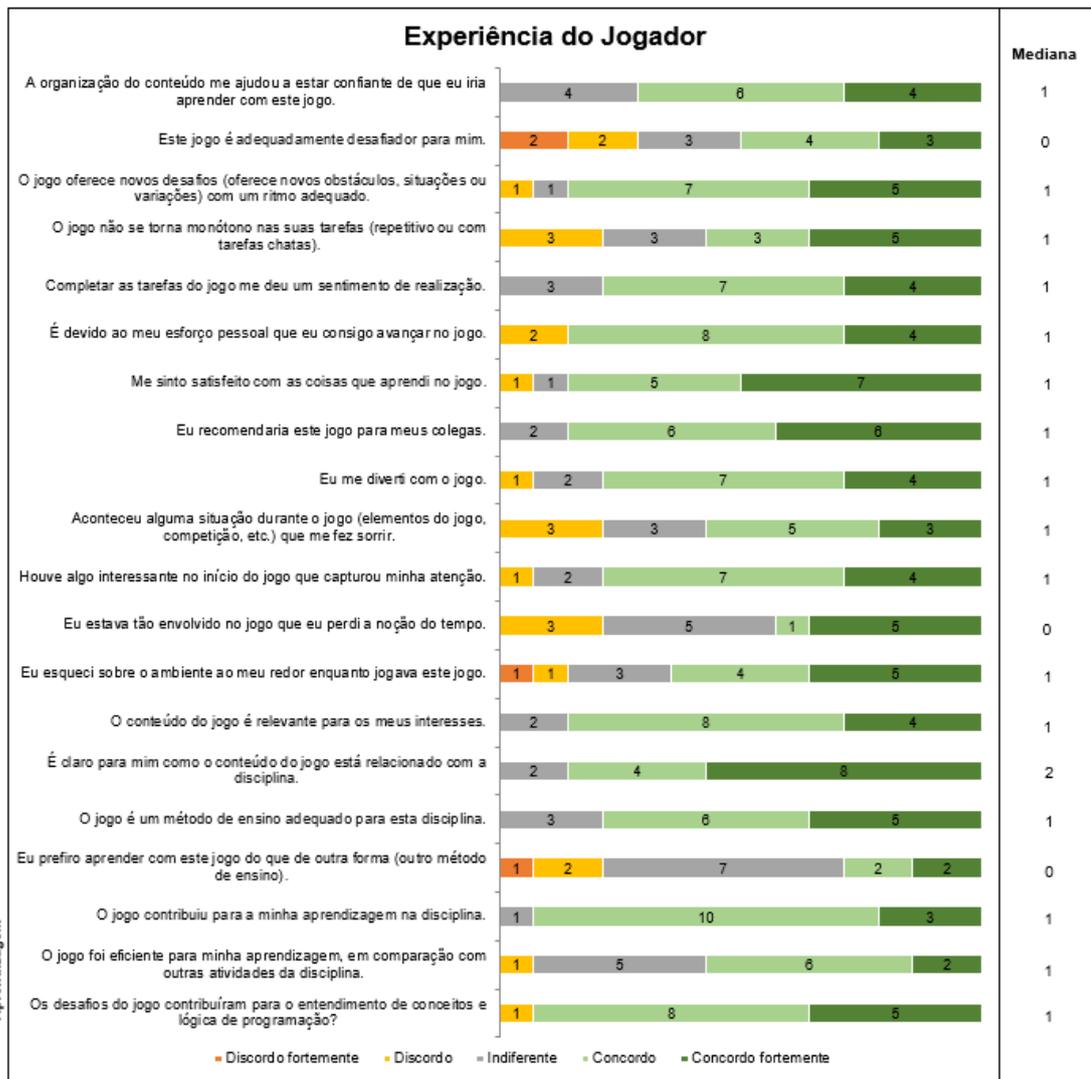


Figura 29. Experiência do Jogador (ICC).

O aspecto com a pontuação mais alta é a Satisfação do jogo, com 36% de concordância (+1) e 29% de concordância fortemente (+2). O tópico é referente à satisfação em relação ao conteúdo ensinado no ProgramSE. O aspecto com as menores notas positivas foi Atenção Focada. A maior parte dos participantes não se sentiu tão engajada no enredo do jogo. Os demais aspectos tiveram um maior nível de concordância (+1) e indiferença (0), são eles: 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*” está incluída no aspecto Percepção de Aprendizagem, obtendo mediana 1, sendo 45% de concordância (+1) e 28% de concordância fortemente (+2).

Os participantes também apresentaram sugestões e/ou melhorias que eles identificaram para o progresso do jogo. Essa etapa consistiu no quarto passo do

questionário (as questões discursivas). Essas informações foram processadas e consolidadas no gráfico da Figura 30.

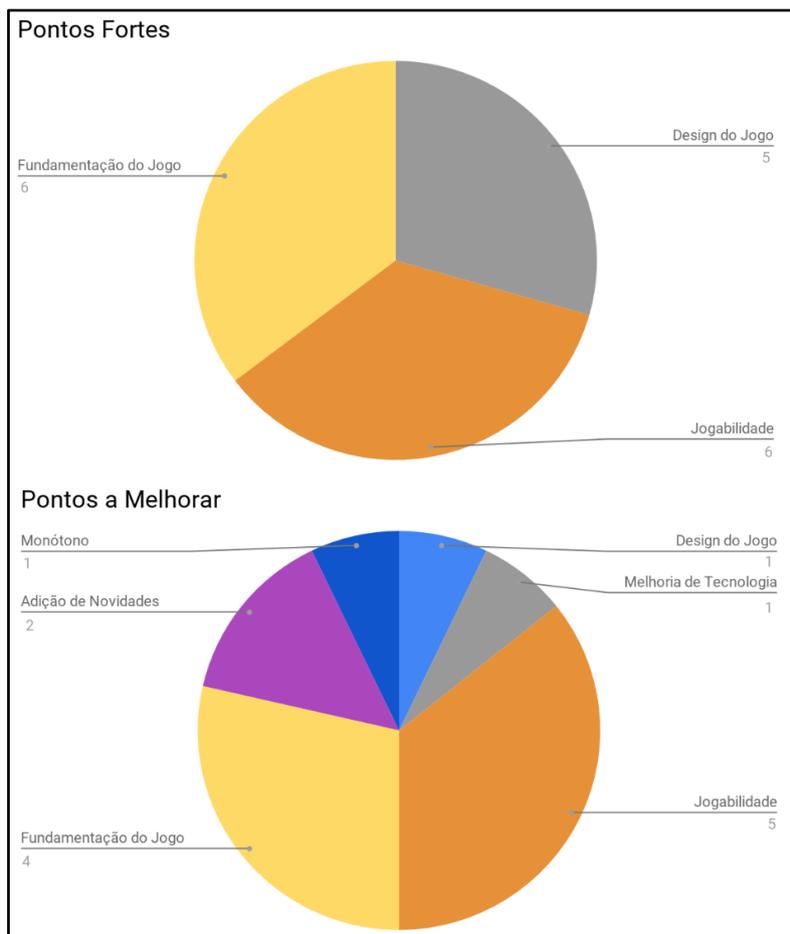


Figura 30. Questões Discursivas (ICC).

Entre os pontos fortes, o mais citado entre os participantes foi em relação à Jogabilidade do ProgramSE, sendo 6 comentários feitos, e à Fundamentação do Jogo, com a mesma quantidade. 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 aparecem nessa parte, com aproximadamente 1 e 4 comentários, respectivamente. Vale ressaltar que, entre os comentários sobre a melhora da fundamentação do jogo, foram críticas feitas às dicas dispostas (por estarem vagas). Os novos aspectos comentados foram Monótono, i.e. o jogo se tornou monótono em determinado momento, com um comentário; Adição de Novidades, i.e. novas fases e novas dinâmicas no jogo, com 2 comentários; e, por último, Melhoria de Tecnologia, com um comentário.

5.4 Discussão dos Resultados

A partir dos resultados apresentados com as avaliações do *ProgramSE* e as percepções obtidas durante o estudo exploratório, foi possível discutir pontos importantes que estão relacionados tanto ao objetivo do jogo quanto a discussões vigentes relacionadas ao cenário do ensino em geral. A discussão dos resultados é apresentada nesta subseção.

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, pode-se entender que ainda faltam estímulos para mulheres ingressarem e investirem na área. Segundo Adriana Tonini¹⁶ (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. 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 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 Matemática, são adultos jovens. Comparando os alunos entre as turmas em questão, as idades em ICC são mais diversificadas que em TP1, tendo esta turma um perfil mais jovial (menores de 28 anos). Os participantes de ICC também são em sua maioria jovens, porém existem alunos acima dos 40 anos (que não há em TP1). Marques 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 Matemática é noturno, pode ser esse o motivo de encontrarmos alunos mais velhos na disciplina ICC, pela possibilidade de conciliar o trabalho com os estudos.

Nenhum dos participantes da avaliação utiliza jogos não-digitais diariamente e, em TP1, houve alunos que selecionaram que nunca os jogam. Em contrapartida, 53% do total de estudantes utilizam jogos digitais todos os dias e 21% os utiliza semanalmente. Baseado no modelo de Cruz (2012), existem relações entre jovens e o envolvimento em jogos e, como a faixa etária é 75% adulto jovem, explica-se o alto uso de jogos digitais.

Em ambas as disciplinas, houve uma pessoa que selecionou que nunca utiliza jogos digitais, sendo aproximadamente 6% da amostragem de estudantes. O aluno de

¹⁶ <https://epocanegocios.globo.com/Carreira/noticia/2018/05/precisamos-incentivar-meninas-para-carreiras-de-exatas-desde-base-diz-diretora-do-cnpq.html>

TP1 possui de 29 a 39 anos e o de ICC, de 40 a 50 anos. Os dois responderam que o ProgramSE foi eficiente 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 o ProgramSE foi desenvolvido com base nas métricas de usabilidade (e.g. aprendizagem e memorização) e sua eficiência é validada quando pessoas que não tem 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, i.e. +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, fez-se importante para a concretização do objetivo principal do ProgramSE.

Outra premissa que obteve resposta alta foi “*Eu considero que o jogo é fácil de jogar*”; porém, a mediana de TP1 foi maior que a de ICC, respectivamente, +2 e +1. O fato da primeira turma ter uma quantidade alunos que utilizam jogos digitais diariamente maior que a segunda pode justificar a diferença entre as avaliações (mesmo que sutil). Em contrapartida, as avaliações referentes ao desafio proposto tiveram mais respostas positivas em ICC que em TP1. Elas avaliam se o jogo é desafiador e, como em TP1 os alunos julgaram o jogo como fácil de ser jogado, pode justificar a falta de momento desafiadores no jogo.

As avaliações sobre a experiência dos jogadores são as mais importantes de serem interpretadas, pois determinam a eficiência 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 o ProgramSE. No geral, os aspectos definidos nessa etapa tiveram 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). 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 em relação ao ensino. Neste caso, o jogo, que teve como objetivo auxiliar estudantes iniciantes no processo de ensino-aprendizagem de 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 disciplinas. Possivelmente, pela grande quantidade de alunos que jogam diariamente, as expectativas para o jogo não foram correspondidas, o que pode ter acarretado no não-sucesso dos aspectos citados. Ou seja, o conteúdo foi apresentado de uma maneira lúdica e interessante, mas, em termos de diversão, imersão no jogo e o desafio proposto, o *ProgramSE* não satisfez os alunos totalmente. Existem, entretanto, diferenças entre jogos de entretenimento e jogos educacionais. O engajamento de *gamers*, i.e. termo que se refere a pessoas que jogam, se torna um empecilho para a utilização de jogos pedagógicos, uma vez que estão acostumados com o meio e sua alta produção.

Aspectos relacionados a Desafio, Satisfação, Atenção Focada e Percepção de Aprendizagem tiveram mediana 0 apenas em TP1. O aspecto da Percepção de Aprendizagem vale ressaltar pela diferença entre as disciplinas e seu público (pois, em ICC, a sua mediana foi +1). A disciplina TP1 é ofertada em um curso de informática, ou seja, as pessoas estão inseridas no meio da programação e suas tecnologias. Outro fato, é que a avaliação foi feita no meio do período. Dessa forma, os estudantes de TP1 já tiveram seu primeiro contato com lógica de programação, o que torna o conteúdo do *ProgramSE* uma repetição dos conteúdos já visitados durante a disciplina, justificando a nota. Em ICC, o curso de Matemática oferece essa disciplina, mas não é o objetivo geral. O público dessa turma não tem muito contato com o assunto, tendo os conceitos vistos no *ProgramSE* como um aprofundamento de seu conhecimento. No entanto, como também foi avaliado no meio do período, pode ter influenciado negativamente as avaliações (visto que alguns conceitos já foram 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 para os estudantes de TP1 (tanto fortes como a melhorar) foram relacionados à jogabilidade, i.e. os elementos apresentados no jogo e sua progressão. Em ICC, o ponto forte mais comentado foi em

relação à Fundamentação do Jogo, o conteúdo apresentado e a forma de ensinar os conceitos. Como a primeira turma teve mais contato com programação (e apresenta muitos *gamers*), focou sua atenção nos elementos de jogos em si. Outro aspecto que obteve um bom retorno 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, o jogo *ProgramSE*.

O questionário MEEGA+ tem campos de questões discursivas, possibilitando ao estudante escrever pontos fortes e a melhorar de acordo com a sua percepção do jogo. Entre os pontos a melhorar (de ambas as disciplinas), algumas observações feitas pelos alunos ajudaram para o aperfeiçoamento do jogo educacional proposto. 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, listando a adição de novidades como algo a ser melhorado.

“O jogo poderia ser mais demorado, com mais "fases"”
Estudante 7 de TP1

“Poderia ser mais longo”
Estudante 3 de ICC

Outro fator que pode ter colaborado para a baixa aceitação da jogabilidade foi o fato de alguns alunos acharem o ProgramSE monótono. Esse aspecto foi mais comentado na turma de TP1, o que pode ser justificado, pois a turma, no geral, não achou o jogo desafiador. O jogo se baseia na programação em blocos, ou seja, sua mecânica é bastante repetida e utilizada ao longo do jogo. Por último, a melhoria de tecnologia apareceu, pois algumas versões do browser utilizados na avaliação eram diferentes da utilizada ao desenvolver o jogo, o que causou alguns transtornos (principalmente na parte do *drag`n`drop*¹⁷ dos blocos de programação).

“Mecânicas novas, menos repetitivas e mais mecânicas fora o mouse.”
Estudante 8 de TP1

¹⁷ A ação de clicar em um objeto virtual e "arrastá-lo" a uma posição diferente.

*“Tentar fazê-lo ficar menos monótono”
Estudante 5 de ICC*

*“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 de TP1*

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.

*“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 de TP1*

*“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 de TP1*

*“A forma como ele aborda o conteúdo, é de uma forma mais interessante que o normal.”
Estudante 4 de ICC*

A avaliação realizada pelos alunos evidenciou alguns problemas de jogabilidade que são importantes e cruciais para o engajamento deles no jogo. Porém, o lado educacional funcionou, sendo possível ensinar os conceitos de programação com uma nova metodologia. Por ser no campo dos jogos digitais, é necessário haver, além da base conceitual, um estudo aprofundado sobre a forma de apresentar os desafios do jogo. No geral, o ProgramSE obteve um bom desempenho, agregando qualidade técnica sobre a utilização de jogos digitais no domínio da educação.

6 Conclusão

Este trabalho apresentou o *ProgramSE*, um jogo do gênero *Escape the Room* que utiliza desafios para ensinar conceitos de lógica de programação. Sua proposta é auxiliar o ensino para iniciantes na graduação, baseando-se em aspectos da teoria pedagógica do Construtivismo e da Taxionomia de Bloom para proporcionar ao estudante um ambiente engajador e motivador para o ensino.

Estudos na área apontam o ensino de lógica como um dos fatores para as altas evasões nos cursos de informática. Levando em consideração a aplicação da programação em blocos (e.g. Scratch e Blockly) e a metodologia lúdica no domínio da educação, foi proposto um jogo educacional para agregar conhecimentos técnicos à área. A primeira etapa consistiu em revisar os trabalhos similares realizados nos últimos dez anos e catalogar os principais requisitos e elementos de jogos digitais, fundamentando a construção da plataforma. A segunda etapa começou com o desenvolvimento do jogo, utilizando o motor de jogo *Construct 3*, o software de design de interiores *Sweet Home 3D* e o software livre *Inkscape* para as demais artes.

Após o seu desenvolvimento, a eficácia do *ProgramSE* foi avaliada por meio do questionário MEEGA+ em duas turmas de programação para iniciantes na UNIRIO (Sistemas de Informação - Técnicas de Programação 1 e Matemática - Introdução à Ciência da Computação). Com o processamento dos dados, seus resultados e a efetividade do jogo foram discutidos. A percepção dos alunos em relação aos conceitos ensinados no *ProgramSE* foi positiva, tendo um retorno otimista dos mesmos.

Como contribuições, este trabalho permitiu a publicação de resultados relacionados direta ou indiretamente com a pesquisa realizada como artigos completos em duas conferências: SBIE 2018 [Silva et al. 2018a] e SBGAMES 2018 [Silva et al. 2018b].

6.1 Limitações

O estudo contou com algumas limitações tanto em relação à construção do jogo quanto em relação ao estudo realizado para a sua avaliação. Durante a execução da avaliação,

foi observado que alguns estudantes utilizaram um *browser* desatualizado em relação ao utilizado no desenvolvimento do ProgramSE. Devido a isto, durante o estudo, observou-se que algumas funcionalidades do jogo foram afetadas. Em determinados momentos, colisões nos blocos de programação com os guias de programação não ocorriam como deveria. Isto causou um pouco de desconforto e irritabilidade em alguns participantes do estudo. Embora este problema tenha sido identificado, o fluxo do jogo não foi alterado, mantendo-se fiel a proposta.

Além disso, não foi disponibilizado para os estudantes fone de ouvido. Logo, quem não tinha o acessório, não pode jogar com 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 para o desenrolar do enredo. O ProgramSE também possuía, além da música de fundo, *feedbacks* sonoros em algumas ações do aluno. Como muitos alunos não tinham/utilizaram fones de ouvido, não puderam desfrutar dessa total imersão no ambiente do jogo.

Por mais que tenha sido escolhido um motor de jogo para auxiliar o processo de criação do jogo, a falta de experiência sobre o universo dos jogos prejudicou o planejamento do ProgramSE, uma vez que é importante entender como os jogos digitais funcionam e quais diretrizes seguir para construir um jogo 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.

6.2 Trabalhos Futuros

Como trabalhos futuros, pretende-se evoluir o jogo e englobar o ensino de novos conceitos de programação. Mesmo o jogo visando ensinar lógica para iniciantes, existem outros comandos mais avançados que os estudantes podem ter seu primeiro contato através do jogo educacional. Isso ajudaria os alunos a terem uma visão mais completa sobre conceitos de lógica e programação de uma forma lúdica e engajada.

A partir das avaliações realizadas, pretende-se levar em consideração os comentários feitos e resultados discutidos para agregar valor ao jogo e tornar sua jogabilidade um fator que contribua para o processo de ensino-aprendizagem. Pretende-

se ainda incluir novos desafios a fim de torná-lo menos monótono, além de novas fases para torná-lo mais extenso.

6.3 Considerações Finais

Pode se afirmar que o *ProgramSE* pode auxiliar estudantes ingressantes no processo de ensino-aprendizagem de programação. Muitos comentários parabenizando o projeto e informando que conseguiriam assimilar o conteúdo foram feitos. Percebe-se que o ensino carece, de fato, de investimento em novas metodologias dinâmicas para auxiliar o crescimento cognitivo dos estudantes. Em contrapartida, o jogo obteve retornos negativos em relação aos desafios propostos, por serem monótonos. Isso reforça a análise de que, no campo dos jogos digitais, os aspectos avaliados pelos alunos influenciam no sucesso do projeto.

Os jogos educacionais são uma boa alternativa para auxiliar o tutor no ensino de programação para iniciantes na área. No entanto, apenas os conceitos e estruturas não são suficientes para o total sucesso de um projeto: a jogabilidade tem um peso grande para determinar o engajamento dos alunos. Desta forma, deve haver um estudo aprofundado sobre jogos digitais e quais são os principais elementos para a imersão e o engajamento dos participantes.

Referências Bibliográficas

- Attrot, W. e Ayrosa, P. (2002). Aplicações da Robótica no Ensino de Ciência da Computação. Anais do Workshop sobre Educação em Computação, pp. 1-8.
- Basili, V. (1992). “Software Modeling and Measurement: The Goal/Question/Metric Paradigm”. Technical Report CS-TR-2956, University of Maryland, Maryland, EUA.
- BUNCHBALL INC. (2010), “Gamification 101: an introduction to the use of game dynamics to influence behavior”.
- 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, v. 6, n. 2, pp. 44-65.
- Cruz, D. V. A. (2012), “Juventude e Jogos Digitais: envolvimento e relações sociais através dos Massively Multiplayer Online Role-Play Games”, Universidade Federal do Rio Grande do Sul.
- Finn, A. J. (2018), “A Mulher na Janela”, William Morrow and Company, 1ª edição.
- Hoelfmann, C. (2016), “O uso dos jogos digitais educacionais no processo de ensino e aprendizagem”, Universidade Federal de Santa Catarina.
- Iwata, S. (2005), “GDC 2005: Iwata Keynote Transcript - IGN”. <http://www.ign.com/articles/2005/03/11/gdc-2005-iwata-keynote-transcript?page=3>, acessado em Jul/2018.
- Kapp, K. M. (2012), “The Gamification of learning and instruction: Game-based methods and strategies for training and education”, Pfeiffer, 1ª edição.
- Kitchenham, B. e Charters, S. (2007), “Guidelines for Performing Systematic Literature Reviews in Software Engineering”, Keele Univ./Durham University Joint Report.
- Klapztein, P. (2014). Jogos e Gamificação como Ferramentas de Auxílio à Educação. In Anais do 10º Seminário Nacional de Educação a Distância.
- Lima, T., Campos, B., Santos, R. e Werner, C. (2012). UbiRE: A game for teaching requirements in the context of ubiquitous systems. Anais do XXXVIII Conferencia Latinoamericana En Informática, pp. 1-10.
- Marji, M. (2014), “Aprenda a Programar com Scratch: Uma introdução visual à programação com jogos, arte, ciência e matemática”, Nonatec Editora Ltda.
- Marques, B. S. e Silva, M. A. C. (2017). TRABALHADORES-ALUNOS: motivações e desafios que configuram um cenário de luta. Anais do I Encontro Internacional de Gestão, Desenvolvimento e Inovação, v. 1, n. 1, pp. 1-16.
- Medeiros, T. et al. (2013). Ensino de programação utilizando jogos digitais: uma revisão sistemática da literatura. RENOTE v. 11, n. 3, pp. 1-10.
- Moreira, G. G. (2018), “Avaliação da Percepção de Jogadores sobre Jogos Educacionais para Ensino de Scrum”, Universidade Federal do Ceará.
- Neto, G. F. et al. (2012). Utilização do kit Lego Mindstorm NXT no Ensino de Controle de Processos. Anais do XL Congresso Brasileiro de Educação em Engenharia.

- Netto, D. et al. (2017). Game Logic: Um jogo para auxiliar na aprendizagem de lógica de programação. Anais do XXV Workshop sobre Educação em Computação, pp. 2297-2306.
- Piteira, M. e Haddad, S. (2011). Innovate in your program computer class: an approach based on a serious game. Proceedings of the Workshop on Open Source and Design of Communication, p. 49-54.
- Petri, G. et al. (2018). “MEEGA+: A Method for the Evaluation of Educational Games for Computing Education.”. Technical Report INCoD/GQS.06.2018.E, Universidade Federal de Santa Catarina, Santa Catarina, Brasil.
- Protopsaltis, A., Hetzner, S., Pappa, D. e Pannese, L. (2011), “Serious Game for Formal and Informal Learning”. E-Learning Papers, ISSN: 1887-1542, pp. 1-8.
- Sá, E. J. V. et al. (2007). Design de atividades de aprendizagem que usam Jogos como princípio para Cooperação. Anais do XVIII Simpósio Brasileiro de Informática na Educação, pp. 539-549.
- Sánchez, A. V. (2014), Tecnologias para a modalidade EAD: um estudo no cenário educacional atual. Revista E-Tech: Tecnologias para Competitividade Industrial, ISSN: 1983-1838, pp. 71-104.
- Santos, R. P. et al. (2008). Uma Proposta de Cenário para Ensino de Algoritmos e Programação com Contribuições de Cooperação, Colaboração e Coordenação. Anais do XVI Workshop sobre Educação em Computação, pp. 218-227.
- 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. Anais do XXVII Simpósio Brasileiro de Informática na Educação, pp. 549-558.
- Sheldon, L. (2012). The Multiplayer Classroom: Designing Coursework as a Game. Boston, MA: Cengage Learning.
- Silva, L. A. et al. (2017). Poredu: um ambiente de programação em blocos. Anais dos Workshops do VI Congresso Brasileiro de Informática na Educação, pp. 144-151.
- Silva, R. R. et al. (2018a). Panorama da Utilização de Jogos Digitais no Ensino de Programação no Nível Superior na Última Década: Uma Revisão Sistemática da Literatura. Anais do XXIX Simpósio Brasileiro de Informática na Educação.
- Silva, R. R. et al. (2018b). Arquitetura de uma Plataforma para um Ecossistema de Software no Domínio de Ensino de Programação. Anais do XXIX Simpósio Brasileiro de Jogos e Entretenimento Digital.
- Uzunca, B. e Jansen, S. (2016). How do Ecosystem Dynamics work in Serious Gaming Ecosystems? Challenges and Opportunities. Proceedings of the 36th Annual Conference – Strategic Management Society.
- Victal, E. R. N. et al. (2015). Aprendendo sobre o uso de Jogos Digitais na Educação. Anais do XXI Workshop de Informática na Escola, pp. 444-453.
- Vygotsky, L. S. (1980), “Mind in Society: The Development of Higher Psychological Processes”. Cambridge: Harvard University Press, pp. 52-91.
- Werbach, K. e Hunter, D. (2012), “For the win: how game thinking can revolutionize your business”. WHARTON DIGITAL PRESS.

Anexo 1 - Planejamento do Jogo

Nível 1 - Se - Senão, Enquanto, Para, Caso

Quarto - Introdução ao pensamento lógico

- Abajur: Faça > Pegar a lâmpada (+ lâmpada)
- Armário: Faça > Abrir o guarda roupa (+ garrafa d'água)
- Cama: Faça > Olhar debaixo da cama
- Cômoda: Faça > Mexer nas gavetas (+ chave)

Corredor - Se e Enquanto

- Banheiro: Se > A porta estiver destrancada > Faça > Entrar no cômodo (- chave)
- Lâmpada: Se > A lâmpada estiver queimada > Faça > Colocar uma nova lâmpada (- lâmpada)
- Tapete: Enquanto > O tapete estiver desarrumado > Faça > Esticar o tapete (+ faca)
- Vaso de planta: Enquanto > A garrafa tiver água > Faça > Regar a planta (- garrafa d'água)

Banheiro - Se e Enquanto com 2 condições e Para

- Armário: Se > A porta está destrancada > & > A porta estiver com o puxador > Faça > Abrir a porta (+ desentupidor)
- Banheira: Enquanto > O ralo estiver entupido > & > A banheira tiver água > Faça > Desentupir o ralo (- desentupidor, + grampo)
- 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 (+ tachinha)

Sótão - Condicional Se-Senão, Caso

- Armário: Se > A porta estiver trancada > & > Tiver um grampo > Faça > Abrir a fechadura > e > Abrir o armário > Senão > Procurar um grampo (+ grão de café, - grampo)
- Caixa: Se > A caixa estiver lacrada > & > Tiver algo com ponta > Faça > Rasgar o lacre > Senão > Procurar algo para rasgá-lo (- tachinha)
- 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 - Para - Passo, Função e Procedimento

Escada - Para - Passo

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

Cozinha - Função/Procedimento

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

Sala - Vetor/Matriz, Ordenação

- 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

Conteúdo do Livro

Vetores: Os vetores são estruturas de dados unidimensionais (1 linha e [N] colunas) que servem para armazenar variáveis do mesmo tipo.

Por exemplo, vetor estanteLivros [N] que contém N prateleiras de livros.

Matrizes: As matrizes são estruturas de dados bidimensionais ([M] linhas e [N] colunas), que servem para armazenar variáveis do mesmo tipo.

Por exemplo, matriz tabuleiro[M, N] que contém M x N casas.

Ordenação: A ordenação é o processo de rearranjo de um certo conjunto de objetos de acordo com algum critério.

Nível 3 - Recursividade

Entrada - Recursividade

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

Dinâmica recursividade

1- "Que lugar é esse? Preciso sair daqui"

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

2- "O quê? Como voltei para cá? Preciso sair logo."

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

3- "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

4- "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

5- "Tudo bem... a carta...deve ter algo ali..."

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

6- "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

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!"

Anexo 2 - Questionário MEEGA+

Avaliação do Jogo ProgramSE

Este questionário visa avaliar a qualidade do jogo Programse, que busca ensinar conceitos e lógica de programação.

*Obrigatório

Termo de Consentimento Livre Esclarecido

Procedimento

A avaliação ocorrerá de forma presencial, sendo dividida em três etapas. Na primeira, pedimos que responda ao questionário fornecendo informações demográficas.

Para a segunda etapa, pedimos que jogue o jogo ProgramSE por meio do link que será disponibilizado.

A terceira etapa ocorrerá logo após o término do jogo, onde será pedido que responda ao questionário. Nele, solicitamos a sua especial colaboração em: (1) responder sobre as suas impressões do jogo; e (2) permitir que os dados resultantes da sua participação sejam estudados.

Estima-se que para realizar a primeira etapa sejam necessários cerca de 5 (cinco) minutos, para a segunda etapa 20 (vinte) minutos e que para realizar a terceira etapa sejam necessários aproximadamente 10 (dez) minutos.

Confidencialidade

Eu estou ciente de que os dados obtidos por meio deste estudo serão mantidos sob confidencialidade e os resultados serão posteriormente apresentados de forma agregada, de modo que um participante não seja associado a um dado específico.

Estou ciente de que algumas fotografias poderão ser feitas como registro desta atividade, mas não serão publicadas em nenhum local sem autorização.

Da mesma forma, me comprometo a não comunicar meus resultados enquanto o estudo não for concluído, bem como manter sigilo das técnicas e documentos apresentados e que fazem parte do experimento.

Benefícios e Liberdade de Desistência

Eu entendo que, uma vez que o experimento tenha terminado, os trabalhos que desenvolvi serão estudados visando analisar a qualidade do jogo.

Entendo que sou livre para realizar perguntas a qualquer momento, solicitar que qualquer informação relacionada à minha pessoa não seja incluída no estudo ou comunicar minha desistência de participação, sem qualquer penalidade. Por fim, declaro que participo de livre e espontânea vontade com o único intuito de contribuir para a avaliação e posterior melhora do jogo ProgramSE.

Responsáveis

Pesquisador:
Rodrigo Ribeiro Silva (rodrigo.ribeiro@uniriotec.br)
Escola de Informática Aplicada - UNIRIO

Professores:
Prof. Rodrigo Pereira dos Santos (rps@uniriotec.br)
Departamento de Informática Aplicada - UNIRIO

Prof. Luis Rivero (luisrivero@nca.ufma.br)
Programa de Pós-graduação em Ciência da Computação - UFMA

1. *

Marque todas que se aplicam.

Eu concordo em participar da avaliação conduzida por Rodrigo Ribeiro Silva da UNIRIO, sob orientação do Prof. Rodrigo Pereira dos Santos e coorientação do Prof. Luis Rivero.

Ir para a pergunta 2.

Informações Demográficas

Gostaríamos que você respondesse as questões abaixo com as informações demográficas. Todos os dados são coletados anonimamente e somente serão utilizados no contexto desta pesquisa.

2. **Instituição ***

3. **Curso ***

4. **Disciplina ***

5. **Faixa Etária ***

Marcar apenas uma oval.

- Menos de 18 anos
 18 a 28 anos
 29 a 39 anos
 40 a 50 anos
 Mais de 50 anos

6. **Sexo ***

Marcar apenas uma oval.

- Masculino
 Feminino

7. **Com que frequência você costuma jogar: ***

Marcar apenas uma oval por linha.

	Nunca	Raramente	Mensalmente	Semanalmente	Diariamente
Jogos Digitais	<input type="radio"/>				
Jogos Não-Digitais (de cartas, tabuleiro, etc.)	<input type="radio"/>				

Ir para "Link e Descrição do Jogo".

Link e Descrição do Jogo

O ProgramSE é um jogo educativo em 2D do gênero escape the room que visa apresentar conceitos e lógica de programação.

O jogo está disponível no seguinte link: <https://bit.ly/2iHRtjZ>

Após finalizar o jogo, clique em Próxima para responder ao questionário.

Ir para a pergunta 8.

Questionário para a Avaliação da Qualidade do Jogo

Este questionário visa avaliar a qualidade do jogo ProgramSE, que busca apresentar conceitos e lógica de programação.

Gostaríamos que você respondesse as questões abaixo sobre a sua percepção da qualidade do jogo para nos ajudar a melhorá-lo.

Usabilidade

Por favor, marque uma opção de acordo com o quanto você concorda ou discorda de cada afirmação abaixo.

8. *

Marcar apenas uma oval por linha.

	Discordo Totalmente	Discordo	Nem discordo, Nem Concordo	Concordo	Concordo Totalmente
O design do jogo é atraente (tabuleiro, cartas, interfaces, gráficos, etc.).	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Os textos, cores e fontes combinam e são consistentes.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eu precisei aprender poucas coisas para poder começar a jogar o jogo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aprender a jogar este jogo foi fácil para mim.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eu acho que a maioria das pessoas aprenderiam a jogar este jogo rapidamente.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eu considero que o jogo é fácil de jogar.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
As regras do jogo são claras e compreensíveis.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
As fontes (tamanho e estilo) utilizadas no jogo são legíveis.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
As cores utilizadas no jogo são compreensíveis.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Experiência do Jogador

Por favor, marque uma opção de acordo com o quanto você concorda ou discorda de cada afirmação abaixo.

9. *

Marcar apenas uma oval por linha.

	Discordo Totalmente	Discordo	Nem Discordo, Nem Concordo	Concordo	Concordo Totalmente
A organização do conteúdo me ajudou a estar confiante de que eu iria aprender com este jogo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Este jogo é adequadamente desafiador para mim.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O jogo oferece novos desafios (oferece novos obstáculos, situações ou variações) com um ritmo adequado.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O jogo não se torna monótono nas suas tarefas (repetitivo ou com tarefas chatas).	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Completar as tarefas do jogo me deu um sentimento de realização.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
É devido ao meu esforço pessoal que eu consigo avançar no jogo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Me sinto satisfeito com as coisas que aprendi no jogo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eu recomendaria este jogo para meus colegas.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eu pude interagir com outras pessoas durante o jogo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O jogo promove momentos de cooperação e/ou competição entre os jogadores.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eu me senti bem interagindo com outras pessoas durante o jogo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eu me diverti com o jogo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aconteceu alguma situação durante o jogo (elementos do jogo, competição etc.) que me fez sorrir	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Houve algo interessante no início do jogo que capturou minha atenção.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eu estava tão envolvido no jogo que eu perdi a noção do tempo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

	Discordo Totalmente	Discordo	Nem Discordo, Nem Concordo	Concordo	Concordo Totalmente
Eu esqueci sobre o ambiente ao meu redor enquanto jogava este jogo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O conteúdo do jogo é relevante para os meus interesses.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
É claro para mim como o conteúdo do jogo está relacionado com a disciplina.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O jogo é um método de ensino adequado para esta disciplina.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eu prefiro aprender com este jogo do que de outra forma(outro método de ensino).	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O jogo contribuiu para a minha aprendizagem na disciplina.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O jogo foi eficiente para minha aprendizagem, em comparação com outras atividades da disciplina.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Os desafios do jogo contribuíram para o entendimento de conceitos e lógica de programação?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Questões Discursivas

10. O que você mais gostou no jogo? *

11. O que poderia ser melhorado no jogo? *

12. Gostaria de fazer mais algum comentário?
