



FACULDADE DE TECNOLOGIA DA ZONA LESTE

RAQUEL MARTINS DO NASCIMENTO

ROBSON HENRIQUE FERREIRA

LOGIKÓS

Sistema lúdico de auxílio ao aprendizado de Lógica de Programação

SÃO PAULO

2021



FACULDADE DE TECNOLOGIA DA ZONA LESTE

Raquel Martins do Nascimento

Robson Henrique Ferreira

Projeto LOGIKÓS

Sistema lúdico de auxílio ao aprendizado de Lógica de Programação

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia da Zona Leste, sob a orientação do Professor Me. Fabio Pereira da Silva, como requisito parcial para a obtenção do diploma de Graduação no Curso de Tecnologia em Análise e Desenvolvimento de Sistemas.

São Paulo

2021

NASCIMENTO, Raquel Martins; FERREIRA, Robson Henrique

LOGIKÓS: Sistema lúdico de auxílio ao aprendizado de Lógica de Programação /

Raquel Martins Nascimento, Robson Henrique Ferreira – Faculdade de Tecnologia da Zona Leste,
São Paulo, 2021

54 p

Orientador: Me. Fabio Pereira da Silva

Trabalho de Conclusão de Curso – Faculdade de Tecnologia da Zona Leste

1. Lógica de Programação 2. Lúdico 3. Aprendizagem

Dedicamos esse trabalho às nossas famílias, amigos e professores que sempre
nos apoiaram e desejaram nossa vitória.

“Ensinar não é transferir conhecimento, mas criar as possibilidades para a sua própria produção ou a sua construção”.

(Paulo Freire)

AGRADECIMENTOS

Agradecemos ao professor Ricardo Satoshi Oyakawa por colaborar com essa pesquisa diretamente abrindo espaço em suas aulas para solicitar aos estudantes que participassem da pesquisa.

Agradecemos, também, aos professores Luciano Francisco de Oliveira e Fabio Pereira da Silva pelas ricas orientações e sugestões ao longo do caminho por nós percorrido durante a execução desse trabalho.

Agradecemos aos amigos e famílias que colaboraram, seja com dicas, sugestões, sendo cobaias de testes para o sistema, ou apenas com compreensão.

NASCIMENTO, Raquel Martins; FERREIRA, Robson Henrique. **LOGIKÓS: Sistema lúdico de auxílio ao aprendizado de Lógica de Programação**. 54p. Trabalho de conclusão de curso. Faculdade de Tecnologia da Zona Leste, São Paulo, 2021

RESUMO

Este trabalho tem como objetivo principal o desenvolvimento de uma aplicação lúdica para o auxílio no ensino-aprendizagem de lógica de programação para estudantes que estejam aprendendo sobre a disciplina, podendo estender seu público-alvo para estudantes no geral. Para tal, realizou-se uma série de levantamentos e análises bibliográficas sobre educação lúdica e ensino-aprendizagem de lógica de programação, bem como houve um levantamento das necessidades e dores dos alunos da Fatec Zona Leste no primeiro semestre de 2021 e a partir das maiores dificuldades encontradas no tema foram elicitados requisitos para o sistema, que teve seu protótipo desenvolvido utilizando a tecnologia da IDE Unity com linguagem C# e gravação de dados em arquivo local. Como resultados, além do protótipo, houve a formação de ampla base de dados a respeito das dificuldades dos estudantes na disciplina de lógica de programação, onde a maior dificuldade encontrada foi relacionada a Arrays (matrizes e vetores), seguida do tema de funções.

Palavras-Chaves: Lógica de Programação; Lúdico; Aprendizagem; Unity; C#.

NASCIMENTO, Raquel Martins; FERREIRA, Robson Henrique. **LOGIKÓS: Sistema lúdico de auxílio ao aprendizado de Lógica de Programação**. 54p. Trabalho de conclusão de curso. Faculdade de Tecnologia da Zona Leste, São Paulo, 2021

ABSTRACT

This work has as main objective the development of a ludic application to assist in the teaching-learning of programming logic for students who are learning about the subject, being able to extend its target audience to students in general. To this end, a series of surveys and bibliographic analyzes on ludic education and teaching-learning of programming logic were carried out, as well as a survey of the needs and pains of Fatec Zona Leste students in the first half of 2021 and, based on the most reported difficulties in the subject requirements were elicited for the system, which had its prototype developed using IDE Unity technology with C# language and data recording in a local file. As a result, in addition to the prototype, there was the formation of a large database about the difficulties of students in the discipline of programming logic, where the greatest difficulty was related to Arrays (matrices and vectors), followed by the theme of functions.

Keywords: Programming logic; Ludic; Learning; Unity; C#.

ÍNDICE DE IMAGENS

Figura 1 - Trilha de uma atividade lúdica	12
Figura 2 - Benefícios do uso de ludificação.....	13
Figura 3 - Pirâmide de desenvolvimento de jogos.....	14
Figura 4 - Diagramas da UML.....	18
Figura 5 - Questionário: Qual sua faixa etária?	20
Figura 6 - Questionário: Você tem formação em outra área?	21
Figura 7 - Questionário: Qual(is) sua(s) outra(s) área(s) de formação?	21
Figura 8 - Questionário: Você sente/sentiu alguma dificuldade ao aprender lógica de programação?.....	22
Figura 9 - Questionário: Qual foi sua maior dificuldade?	22
Figura 10 - Questionário: Você acredita que um jogo interativo desenvolvido exclusivamente para o ensino de Lógica de Programação iria ajudar no aprendizado dessa disciplina?	23
Figura 11 - Como você prefere aprender?	24
Figura 12 - Caso de uso: Estudante.....	29
Figura 13 - Arquitetura de diretórios e arquivos	30
Figura 14 - Tela Inicial	32
Figura 15 - Tela Sobre (Informativa)	33
Figura 16 - Tela Entrar	34
Figura 17 - Tela Seleção Fases	35
Figura 18 - Tela Aprendizagem.....	36
Figura 19 - Tela Fase 5.....	37
Figura 20 - Tela com Texto Dica	38
Figura 21 - Tela com Texto indicação do que fazer	39

SUMÁRIO

1. INTRODUÇÃO	10
2. LUDIFICAÇÃO	12
2.1. ELEMENTOS DO JOGO	14
3. METODOLOGIA	16
3.1. SCRUM	16
3.2. UML	18
4. PROPOSTA	19
4.1. PRIMEIRO ESTÁGIO	19
4.2. SEGUNDO ESTÁGIO	24
5. DOCUMENTAÇÃO DO SISTEMA	25
5.1. VISÃO GERAL	25
5.2. MONETIZAÇÃO	26
5.3. CONCORRÊNCIA	26
5.4. REGRAS DE NEGÓCIO	27
5.5. QUESTIONÁRIO	28
5.6. REQUISITOS FUNCIONAIS	28
5.7. REQUISITOS NÃO FUNCIONAIS	29
5.8. DIAGRAMA DE CASOS DE USO	29
5.9. CASOS DE USO TEXTUAIS	30
5.10. ARQUITETURA UNITY	30
5.10.1. TELAS DE PROTÓTIPO	32
6. DISCUSSÃO E CONSIDERAÇÕES FINAIS	40
7. REFERÊNCIAS	43
8. ANEXOS	46
8.1. TABELA DE RESPOSTAS DO QUESTIONÁRIO DA ETAPA 1	46

1. INTRODUÇÃO

Considerando o contexto atual, cada vez mais conectado, a qualificação e as competências individuais ou conjunto de características (conhecimentos, habilidades e atitudes) possuem um grande destaque e não são restritas às áreas de tecnologia, pois pode-se encontrar em notícias, artigos e sites de sobre trabalho e qualificação reportagens e artigos que indicam a necessidade cada vez maior de bases de conhecimento gerais que sustentam o conhecimento técnico específico. No livro “Competências, habilidades e currículos de educação profissional: crônicas e reflexões”, de Deisi Deffume e Léa Depresbiteris (2019), foram reunidos artigos sobre esse tópico e de forma cativante e reflexiva que reafirmam sua importância perante a forma como o mundo vêm se desenvolvendo.

Outro fator relacionado está na indústria 4.0 e no mercado de trabalho que englobam grandes desafios, dentre os quais estão: o desenvolvimento dos trabalhadores, o uso de novas tecnologias e o compartilhamento de conhecimento (AIRES, 2017). As verdades presentes no trecho “A verdadeira vantagem competitiva está na capacidade e na velocidade do aprendizado das pessoas nas organizações, logo, a gestão do capital intelectual faz-se necessária”, extraído do artigo “Indústria 4.0: desafios e tendências para a gestão do conhecimento” (AIRES, 2017) se mostram cada vez mais pertinentes nos dias atuais.

Assim, pode-se entender o elo entre o conhecimento que se precisa aprender e a forma com que irá ser utilizado no cotidiano, que é propício para a utilização de métodos lúdicos cujo entendimento pode ser captado de várias formas e por múltiplas inteligências, e essas por sua vez canalizadas para outros fins, saindo de um sentido unidimensional, assim como evidencia a pesquisa “As múltiplas faculdades humanas são independentes em graus significativos” (Gardner, 1995). O conhecimento e aprendizado permeiam por anos pesquisas cujos objetivos são de buscar novas formas de aprender, ensinar, absorver e compartilhar o saber com as outras pessoas no ambiente acadêmico ou profissional.

A abordagem utilizada nesse trabalho foi descrever e analisar os impedimentos e dificuldades dos alunos e como contorná-los nos cursos de tecnologia, principalmente ligados ao processo de ensino-aprendizagem de lógica de programação. Portanto, foi-se realizada análise mais aprofundada sobre artigos que dizem respeito a essa temática específica.

Ao analisar fatores mais individuais, tem-se que outros dos fatores que podem ocasionar dificuldades no curso e/ou evasão escolar referem-se ao desconhecimento da metodologia do curso, deficiência da educação básica, diferenças entre o ensino básico e o superior a nível de processos de ensino-aprendizagem – que no ensino básico, no Brasil, em geral é focado na memorização, enquanto no Ensino Superior há a necessidade de se utilizar o raciocínio lógico e crítico (PLATT NETO, 2008).

Para esses alunos que tenham dificuldade na interpretação dos conteúdos utilizados no Ensino Superior – como lógica de programação – a universidade deve propor atividades curriculares e extracurriculares, novas metodologias ou mesmo projetos que visem interação entre estudantes como meios de buscar fazer o estudante se sentir parte efetiva do processo de aprendizagem ocorrido na universidade (ANDRIOLA, 2006). Detecta-se aqui uma oportunidade de apresentar justamente novas tecnologias como parte importante de uma reforma nas metodologias de ensino tradicionais, como a tecnologia proposta nesse trabalho.

Uma pesquisa realizada por Mello (2013) mostrou que 50% dos alunos de uma universidade de tecnologia do Sul do Brasil que evadiram dos cursos o fizeram no primeiro ano efetivo. Ou seja, o dado corrobora com as análises apresentadas por outros autores que sustentam que parte dos alunos sentem dificuldades ao entrarem em contato pela primeira vez com as metodologias do Ensino Superior e os conteúdos mais exigentes de raciocínio crítico.

Como meio de minimizar os danos causados na trajetória acadêmica e profissional por conta da defasagem em Lógica de Programação nos primeiros anos de estudo em Lógica de Programação, pensou-se esse projeto, como forma de suprir um meio de estudo alternativo às aulas tradicionais.

2. LUDIFICAÇÃO

Lúdico – tem sua origem da palavra latina: “ludus” de “ludere” que contempla todo o terreno do jogo com uma única palavra, mas que possui as mais diversas variações, não limitado a uma atividade ou brincadeira e sim a noção de “jogo” como um fator distinto e fundamental. “Ludus” também abrange: os jogos infantis, a recreação, as competições, as representações litúrgicas e teatrais e os jogos de azar (Huizinga,2000).

Na indústria é usada como termo para designar o uso de elementos de jogos em um produto, serviço ou contexto que não seja um jogo eletrônico ou analógico. De acordo com os autores do site Geekie (site.geekie.com.br), apoiadores de desafios na educação básica no Brasil, a ludificação não é apenas usar jogos prontos ou pontuar atividades, mas sim utilizar elementos dos games para promover o aprendizado dos envolvidos como demonstrados nas figuras 8 e 9.

Figura 1 - Trilha de uma atividade lúdica



Fonte: LORENZONI, 2021.

Figura 2 - Benefícios do uso de ludificação



Fonte: LORENZONI, 2021.

O envolvimento num jogo é extremamente particular, porém igualmente poderoso para incentivar novas ideias, raciocínios, tentativas e lógica aprendizados – é uma tarefa, um tanto sofisticada e onerosa, que leva o jogador ao sentimento de que algo está “em jogo” dentro dos conceitos: desafio, perigo, competição, cooperação, aposta, etc. algo como a temeridade natural frente ao que você não pode fugir. Tudo isso está muito próximo num jogo e o domínio lúdico pode ser aplicado nos mais diversos contextos (Huizinga,2000).

“Os jogos oferecem a cada indivíduo a oportunidade de vivenciar experiências de aprendizagem personalizadas, em que se possa escolher revisitar o conteúdo ou adotar estratégias distintas em relação a outro colega de trabalho, mas, ainda assim, atingir o mesmo objetivo de aprendizagem. No âmbito mais simples da memorização, a repetição de conteúdos proporcionada pelos jogos pode nos ajudar a lembrar de informações-chave, como: dados sobre produtos, informações relacionadas ao setor, fases relativas a processos, etc.”.

(BOLLER; KAPP, 2018)

Trecho do livro Jogar para aprender de Sharon Boller e Karl Kapp (2018) que resume de forma muito objetiva os princípios desse projeto.

2.1. ELEMENTOS DO JOGO

Existem vários elementos e mecânicas de jogos que são utilizadas na criação e desenvolvimentos de jogos e também podem ser aplicados em outras áreas, a figura 10 - uma adaptação de pirâmide do “For the Win: How Game thinking can revolutionize your business” (Werbach & Hunter, 2012) demonstra componentes (objetos e ação), mecânicas (regras e objetivos) e dinâmicas (identificação e caracterização) como principais aspectos.

Figura 3 - Pirâmide de desenvolvimento de jogos



Fonte: OBERLIN, 2021.

Como ainda não é possível definir precisamente quais elementos estão contidos nos jogos, (Neto, 2017) é possível representar casos mais comuns. Existe uma infinidade de combinações que sofrem modificações baseando-se no objetivo principal e secundários do projeto, desejos do público-alvo, o tempo de jogo e temática escolhida.

3. METODOLOGIA

O termo “Metodologias Ágeis” refere-se a um conjunto de práticas realizadas em processos de desenvolvimento de softwares que exigem uma velocidade de desenvolvimento e um dinamismo de requisitos maior. Essas metodologias geralmente deixam em segundo plano ferramentas como uma documentação robusta e contratos com clientes, dando mais importância a interações com indivíduos, software executável em menor tempo, colaboração com o cliente e respostas rápidas às mudanças (SOARES, 2004).

Como esse trabalho foi desenvolvido por duas pessoas e houve a necessidade de muitas interações com orientadores e revisores – fatos que acabam gerando mudanças grandes e pequenas ao longo do tempo – percebeu-se uma das metodologias ágeis como um meio de dar vazão a essas mudanças em tempo de alterar e ajustar as ideias e tarefas aqui realizadas. Esse método foi o de nome SCRUM.

Também, visando maior agilidade na troca de arquivos, versões e implementações entre os componentes do grupo, utilizou-se o Git com Github.

Desenvolvido por Linus Torvalds (O também criador do Linux), o Git é um sistema de versionamento de arquivos que possibilita o controle total das modificações, exclusões e inserções desses arquivos em repositórios coletivos e/ou individuais em que se está trabalhando uma equipe de desenvolvimento de software ou de qualquer outro tipo de equipe (DIAS, 2016).

Como IDE e agrupamento de principais ferramentas, utilizou-se o Unity para o desenvolvimento do jogo.

Para a documentação, utilizou-se alguns componentes do sistema UML.

3.1. SCRUM

Na utilização do Scrum utilizou-se principalmente sua mecânica de *sprints*, reuniões de *planning*, *daily* e *review*.

Sprint: é um ciclo de trabalho em que um valor é criado para a equipe e/ou produto;

Planning: é uma reunião de planejamento em que os itens a serem desenvolvidos na *Sprint* são levantados e priorizados – ocorre no momento anterior à *Sprint*;

Daily: é uma reunião diária em que há o alinhamento entre os membros da equipe para que o desenvolvimento dos requisitos para a *Sprint* sejam atendidos da melhor maneira possível;

Review: é uma reunião de revisão em que há um feedback quanto ao que foi desenvolvido ao final da *Sprint*.

Deste modo, houve três *Sprints* devidamente planejadas, desenvolvidas e revisadas:

Sprint 1: Nessa *Sprint* planejou-se a criação da fase 1 do projeto, que foi o levantamento bibliográfico sobre Ludificação, Lógica de Programação e assuntos relacionados, criação do Questionário a ser passado para estudantes de Lógica de Programação e, por último, análise desses resultados e criação do documento inicial deste trabalho.

Sprint 2: Nessa *Sprint* planejou-se a criação da fase 2 do projeto, que foi a criação e o desenvolvimento propriamente dito do protótipo prático do jogo que auxilia o processo de ensino-aprendizagem de Lógica de Programação. Esse planejamento previu 5 fases + 1 fase de tutorial, que teriam como tema “Matrizes”, inspirado na mecânica do clássico jogo Campo Minado, além de telas de menus e relacionados. A *Review* dessa *Sprint* trouxe pontos importantes a serem melhorados na *Sprint* subsequente e mais testes no sistema.

Sprint 3: Nessa *Sprint* planejou-se a criação da documentação do protótipo criado na *Sprint* anterior, seguindo os padrões UML, desenvolvimento de mecânicas aprimoradas para o protótipo e testes de sistema. A *Review* dessa *Sprint* trouxe bons feedbacks relacionados à qualidade do sistema e aprovação da documentação pelos membros da equipe.

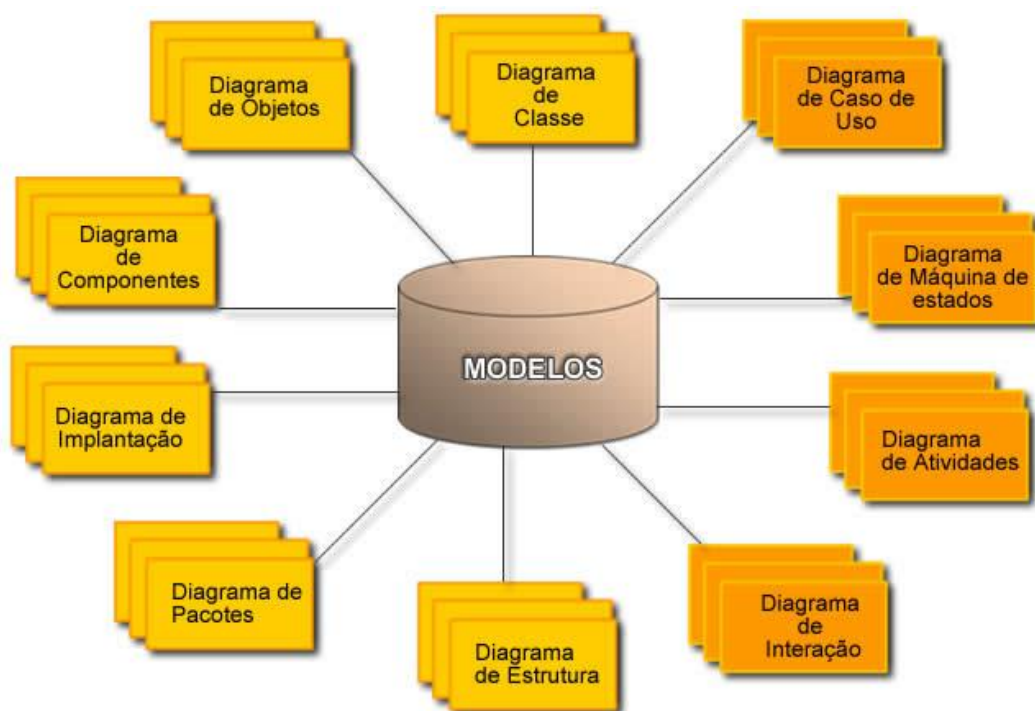
Durante todas as Sprints os membros se comunicaram a partir das reuniões diárias e rápidas para alinhamento de tudo que estava sendo produzido.

3.2. UML

A UML (Unified Modeling Language – Linguagem Unificada de Modelagem) é um sistema desenvolvido como uma base de padronização de modelagem de sistemas e documentação na engenharia de software. A UML oferece auxílio para a visualização clara e simples dos objetos desenvolvidos no sistema e do relacionamento entre eles a partir de diagramas personalizados (MACORATTI, 2021).

A imagem 4 apresenta bem a estrutura de diagramas visuais estabelecidos na UML para a representação do sistema a ser desenvolvido:

Figura 4 - Diagramas da UML



Fonte: MARTINEZ, 2021.

Para a melhor utilização da UML, há a necessidade de terem sido levantados os dados correspondentes aos requisitos não-funcionais, requisitos funcionais e regras de negócio do produto a ser desenvolvido.

4. PROPOSTA

Este projeto se dispõe ao desenvolvimento de um protótipo (MVP – Mínimo Produto Viável) com o intuito de auxiliar o processo de aprendizagem da matéria de lógica de programação para universitários, usando referências do conteúdo ministrado, preferencialmente, nas aulas do curso de Análise e Desenvolvimento de Sistemas da Fatec Zona Leste. Em suma, haverá exercícios lúdicos e técnicas de ludificação cujas abordagens são amigáveis e intuitivas. Esta é uma proposta defendida por muitos pesquisadores e especialistas nas áreas de inovação e metodologias ativas, mas pouco utilizada em ambientes acadêmicos como citado no livro “Aula em Jogo: Descomplicando a gamificação para educadores” do Tiago Eugenio (2020).

4.1. PRIMEIRO ESTÁGIO

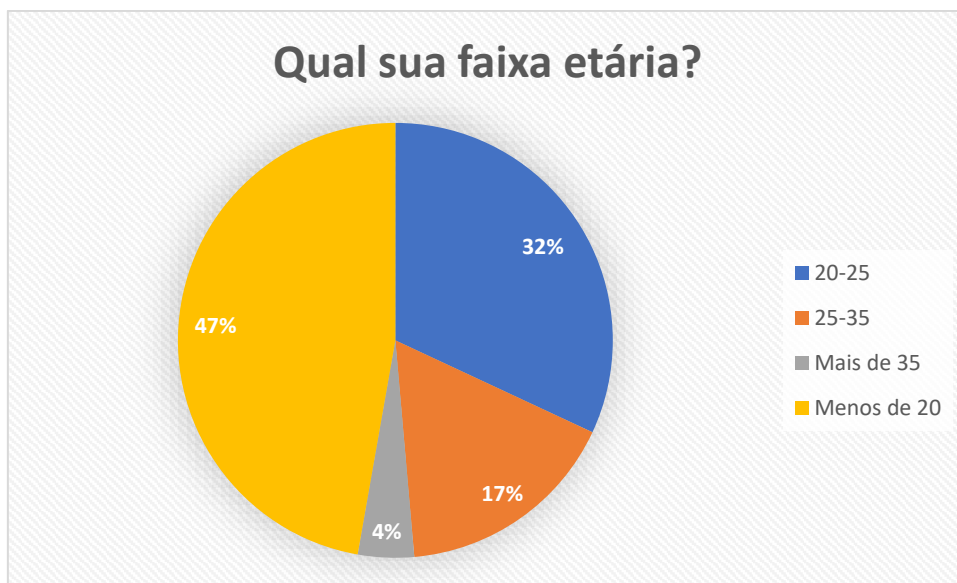
Para iniciar houve o levantamento bibliográfico para embasamento teórico sobre os tópicos que colaboram e enriquecem objetivo desse projeto, contendo referências mais consolidadas do valor entregue. Entre os assuntos tem-se:

- Aprendizado
- Taxinomia de Bloom
- Ludificação
- Múltiplas Inteligências
- Competências
- Lógica de Programação

Após a pesquisa, foi realizada uma revisão analítica dos trabalhos encontrados, extraíndo e discutindo os pontos de maior relevância para enfim apresentar coerência no desenvolvimento e execução desse trabalho. As imagens seguintes mostram os resultados de uma pesquisa de interesse realizada com os alunos do primeiro semestre do curso de Análise e Desenvolvimento de Sistemas das faculdades Fatec Zona Leste e Fatec São Paulo em Maio de 2021 (em anexo, segue também uma

tabela com maiores detalhes de todas as respostas da pesquisa). A mesma pesquisa foi realizada com estudantes de outros semestres para validação de público-alvo. Essas outras respostas estão compiladas em anexo ao final do documento.

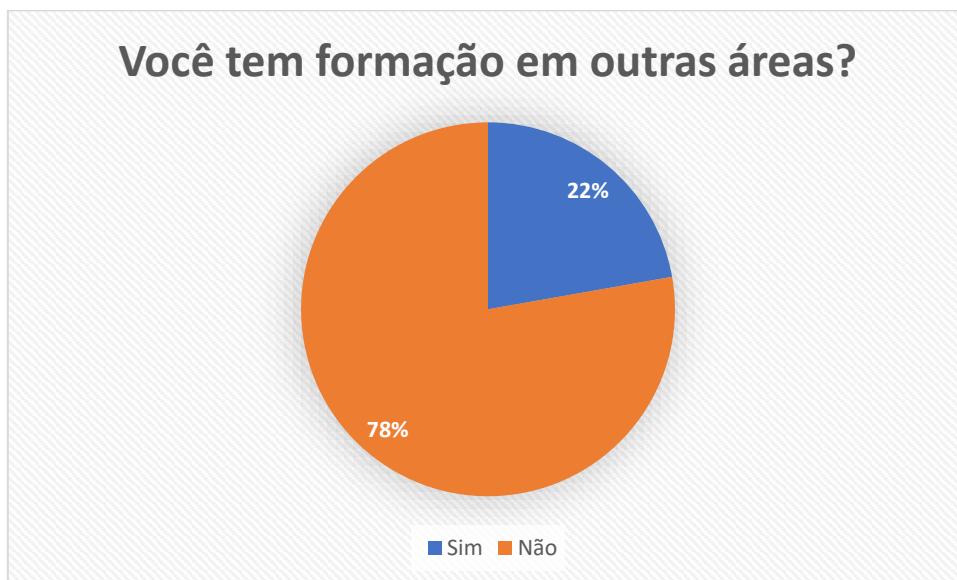
Figura 5 - Questionário: Qual sua faixa etária?



Fonte: Autoria Própria, 2021.

Na Figura 5 pode-se notar que a maior parte das pessoas pesquisadas estão nas faixas-etárias mais jovens, entre as faixas "Menos de 20" e "20-25" anos de idade.

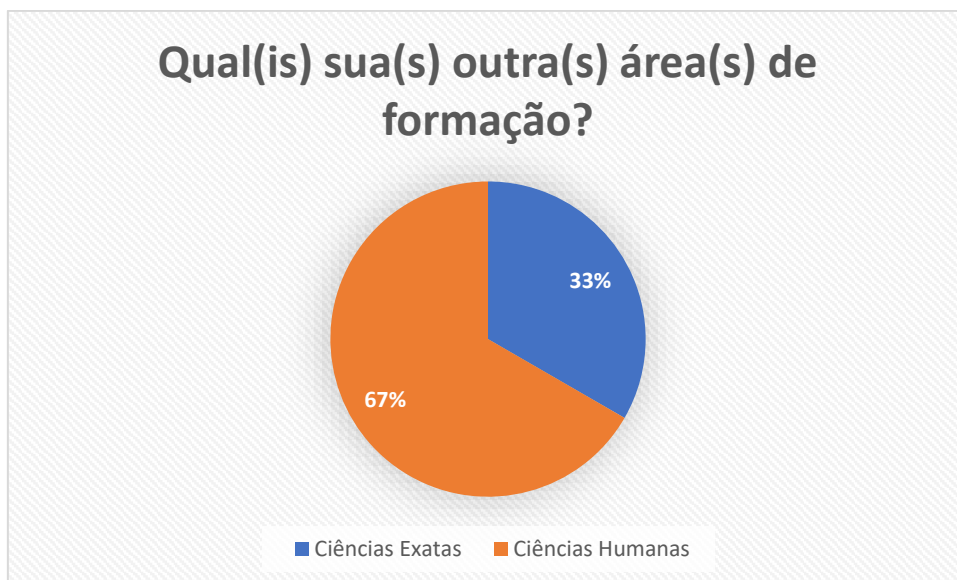
Figura 6 - Questionário: Você tem formação em outra área?



Fonte: Autoria Própria, 2021.

Na Figura 6 evidencia-se que a maior parte dos participantes nesse filtro não tem formação em outras áreas, podendo correlacionar com a Figura 5, já que a maior parte dos participantes são jovens.

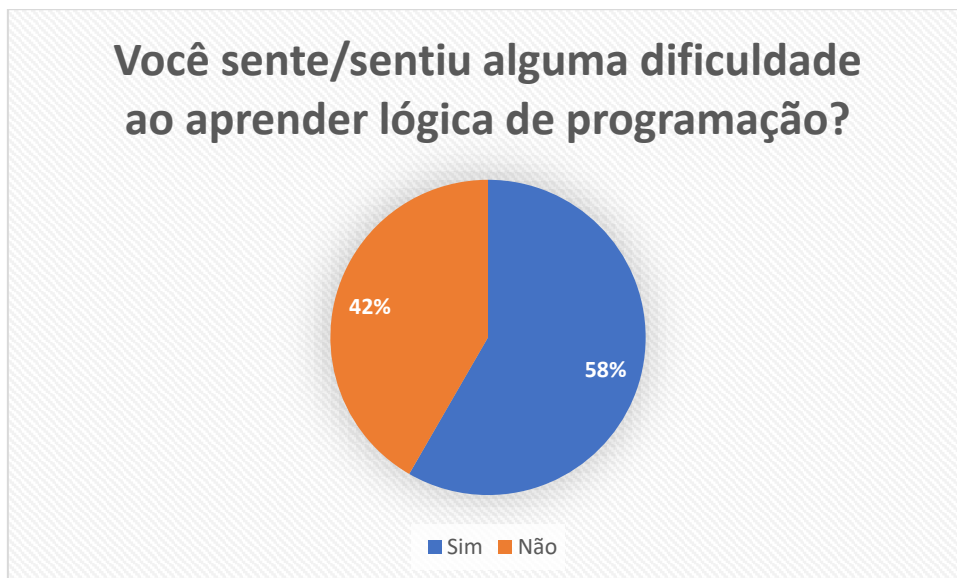
Figura 7 - Questionário: Qual(is) sua(s) outra(s) área(s) de formação?



Fonte: Autoria Própria, 2021.

Na Figura 7 pode-se levantar que a maior parte das pessoas que têm formação em outras áreas veio de áreas das Ciências Humanas.

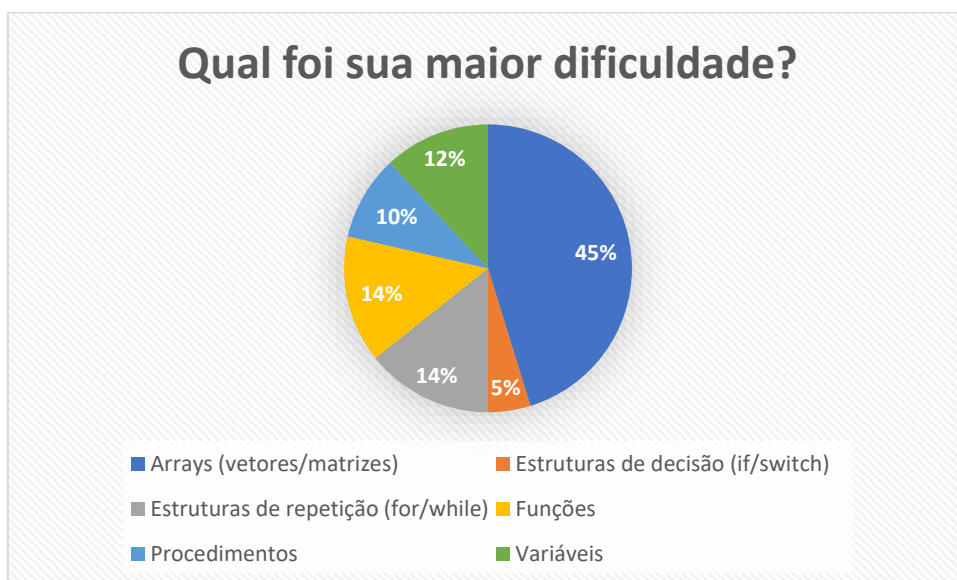
Figura 8 - Questionário: Você sente/sentiu alguma dificuldade ao aprender lógica de programação?



Fonte: Autoria Própria, 2021.

A Figura 8 mostra que a maior parte dos alunos sente/sentiu alguma dificuldade no aprendizado de Lógica de Programação.

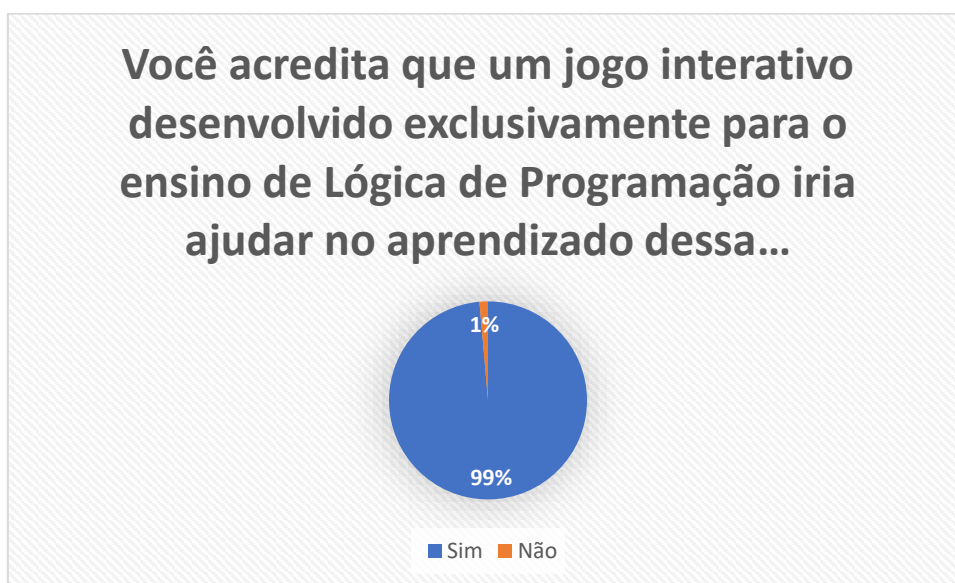
Figura 9 - Questionário: Qual foi sua maior dificuldade?



Fonte: Autoria Própria, 2021.

A Figura 9 elucida a necessidade de se trabalhar em cima do tema de Arrays (vetores e matrizes), já que a grande maioria dos estudantes que responderam o questionário tem dificuldade nesse assunto.

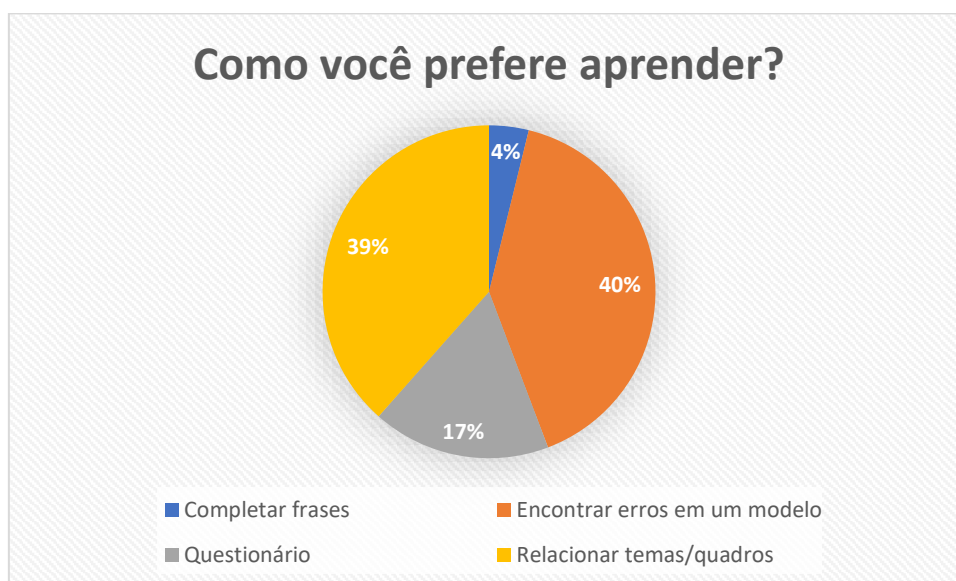
Figura 10 - Questionário: Você acredita que um jogo interativo desenvolvido exclusivamente para o ensino de Lógica de Programação iria ajudar no aprendizado dessa disciplina?



Fonte: Autoria Própria, 2021.

A Figura 9 traz resultados que comprovam a necessidade da criação de uma forma não tradicional, que é o jogo interativo, para auxiliar no aprendizado de Lógica de Programação, pois a maioria dos alunos concordam que o jogo ajudaria nesse processo.

Figura 11 - Como você prefere aprender?



Fonte: Autoria Própria, 2021.

A última pergunta aqui relacionada do questionário foi utilizada para que fosse possível pensar e elaborar o meio em que as fases do jogo seriam pensadas e elaboradas.

4.2. SEGUNDO ESTÁGIO

Desenvolvimento do MVP para computadores adotando a ideia do aprendizado lúdico. Será apresentado como um pequeno jogo juntamente com os artefatos de desenvolvimento para software como: requisitos, regras de negócio, protótipos e diagramas. Os exemplos de códigos utilizados nos exercícios seguem o padrão "JAVA Code Conventions" para fins de uma melhor compreensão visual.

Utilizando os dados do levantamento, encontra-se uma abordagem mais dinâmica de exercícios com múltiplas respostas para uma mesma pergunta, pois não há uma única maneira de chegar num resultado e para mesclar o aprendizado os exercícios são baseados nos exemplos práticos em sala.

Todo o detalhamento do segundo estágio está divulgado na sessão de Documentação do Sistema, incluindo: técnicas, mecânicas, controles, banco de dados

e as informações que serão recolhidas para medir os resultados do projeto mediante ao cadastro do usuário no sistema, não haverá uso de dados sensíveis.

5. DOCUMENTAÇÃO DO SISTEMA

A seguir, toda a documentação do desenvolvimento do sistema seguindo o padrão UML.

5.1. VISÃO GERAL

A ideia durante a execução desse projeto de TCC deverá auxiliar, a partir de técnicas de ludificação a melhora do desempenho do aprendizado em lógica de programação.

Para tanto, criou-se um sistema inspirado em *Escape Room*, um jogo onde você precisa resolver alguns desafios para sair da sala e avançar com as suas “chaves” para o enigma seguindo o conceito do *Campo Minado*, o clássico jogo nativo do Windows que conta com uma mecânica simples, porém que já demanda um certo exercício de lógica para jogá-lo, pois ele te dá pistas de como evitar uma bomba.

Em resumo, há questionamentos distintos em cada fase do jogo, na qual o estudante terá que pensar na resposta lógica e ir escolhendo os valores que estarão dispostos na tela, em que cada clique poderá: explodir uma bomba – para respostas erradas ou marcar com uma pontuação com a resposta correta gerando um score por fase para no final ir liberando mais fases e tópicos novos para se aprimorar.

5.2. MONETIZAÇÃO

Como meios de monetizar o aplicativo e garantir sua autonomia em recursos, foi-se pensado nos seguintes métodos:

- Anúncios com Google AdSense (barras de anúncios);
- Espaços para Patrocínio (para venda de cursos online, por exemplo);
- Apoio financeiro por meio de instituições como universidades públicas e particulares (principais instituições beneficiadas);

5.3. CONCORRÊNCIA

Após pesquisa nos principais buscadores de aplicativos, levantou-se alguns possíveis concorrentes indiretos, que estão representados na tabela 1, comparados ao sistema aqui proposto:

Tabela 1 – Comparativo entre Logikós e concorrentes indiretos

	É lúdico	Focado em Lógica de Programação	Língua Portuguesa	Explica conceitos e definições
Logikós	X	X	X	X
Lógica & Programação		X	X	
SmartCode		X	X	X
Curso Algorítmico			X	
Cidade do Algoritmo	X	X		
Meoweb	X		X	X

De forma mais explicativa:

- Lógica & Programação: não é lúdico. Se baseia em questionário escrito.
- SmartCode: tem boa base de apostilas, textos e imagens de códigos, possibilitando aprendizado por leitura, porém sem a ludificação.
- Curso Algorítmico: não é lúdico, apenas lista cursos online famosos do youtube.
- Cidade do Algoritmo: apenas em língua inglesa, não explica conceitos ou definições, apenas sequências de ordens a partir de ícones. É lúdico.
- Meoweb: com gráficos bons e boa jogabilidade, o jogo consegue reforçar sintaxe CSS, porém não ensina lógica de programação.

Como visto na listagem acima, os possíveis concorrentes do Logikós são concorrentes indiretos, já que apesar de possuírem a mesma temática – aprendizado de lógica de programação – ou temática similar, não possuem os mecanismos equivalentes aos da Logikós de ludificação na aprendizagem através de breve explicação de conceitos e aplicação de questões que exigem mecânicas e elementos de jogos.

5.4. REGRAS DE NEGÓCIO

- RN1: Não é possível acessar as fases do jogo sem realizar o login ou registro
- RN2: Será possível verificar a pontuação ao final de cada fase
- RN4: Haverá tempo máximo para concluir cada fase
- RN5: Possibilidade de responder um questionário sobre a experiência no final do jogo.
- RN7: O jogador só pode avançar no mesmo tópico se passar pela fase anterior, após liberada ele pode acessar a qualquer momento

5.5. QUESTIONÁRIO

O questionário apresentado na aplicação para que o estudante possa fornecer feedback está hospedado no url: <https://forms.gle/26gRRLXrJakHihFg8> e conta com as seguintes questões:

- 1) Obteve sucesso em quantas fases? – Resposta objetiva
- 2) Você acredita que o jogo ajuda no aprendizado de Lógica de Programação? – Resposta objetiva
- 3) Deixe sua opinião (críticas positivas, negativas ou elogios) – Resposta aberta.

Esse questionário auxiliará na análise posterior de resultados e levantamento de melhorias a serem realizadas na aplicação Logikós.

5.6. REQUISITOS FUNCIONAIS

- RF1: Processar visualização de estatísticas de pontuação do usuário;
- RF3: Configurar login com registro.
- RF3: Exibir fases de diferentes dificuldades;
- RF5: Ordenar aleatoriamente as respostas das questões, a cada vez que o usuário entrar no jogo;
- RF6: Guardar dados do usuário para caso ele saia e volte do jogo.

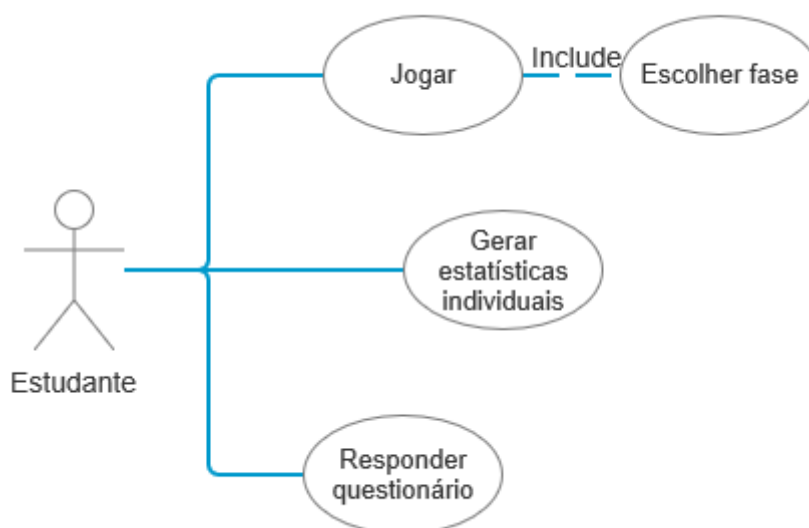
5.7. REQUISITOS NÃO FUNCIONAIS

- RNF1: A linguagem predominante a ser utilizada é o C#, com complementações em javascript;
- RNF2: Será gravado dados locais no computador do jogador;
- RNF3: Para a ambientação gráfica e configuração de jogabilidade será utilizado o Unity;
- RNF4: O jogo será responsivo a diferentes tamanhos de tela.

5.8. DIAGRAMA DE CASOS DE USO

Abaixo está diagrama de caso de uso para o Estudante.

Figura 12 - Caso de uso: Estudante



Fonte: Autoria Própria, 2021.

5.9. CASOS DE USO TEXTUAIS

Abaixo estão os casos de uso textuais.

Caso de uso: Jogar
Visão Geral: O estudante clica no botão para iniciar jogo, é obrigado a escolher a fase que deseja. Então, joga.

Caso de uso: Gerar estatísticas individuais
Visão Geral: O estudante clica para gerar estatísticas individuais, aguarda o carregamento das informações na tela e as visualiza.

Caso de uso: Responder Questionário
Visão Geral: Após testar os recursos do jogo, o estudante clica para responder questionário, responde e o envia a partir de um botão.

5.10. ARQUITETURA UNITY

Figura 13 - Arquitetura de diretórios e arquivos



Fonte: Autoria Própria, 2021.

A engine da Unity possui um diretório chamado Assets que armazena tudo que será utilizado para o desenvolvimento do jogo, para organizar todos os insumos

desenvolvidos, os arquivos foram separados em pastas que seguem o padrão MVC (Model-View-Controller) de arquitetura de software:

- **Scenes:** armazena cada fase ou tela utilizada no aplicativo correspondente a camada de interação com o usuário a VIEW do MVC.
- **Scripts** armazena scripts em c# utilizadas para cada evento ou ação representa os CONTROLLER no modelo MVC
- **Sounds** armazena todos os sons e audios do jogo para a trilha sonora, efeitos, notificações e respostas ao jogador durante a fase
- **Sprites** armazena todas as imagens, fundos, texturas e modelos utilizados nas fases semelhante ao MODEL do modelo MVC
- **TextMesh Pro** armazena fontes e configurações de texto para uma melhor leitura

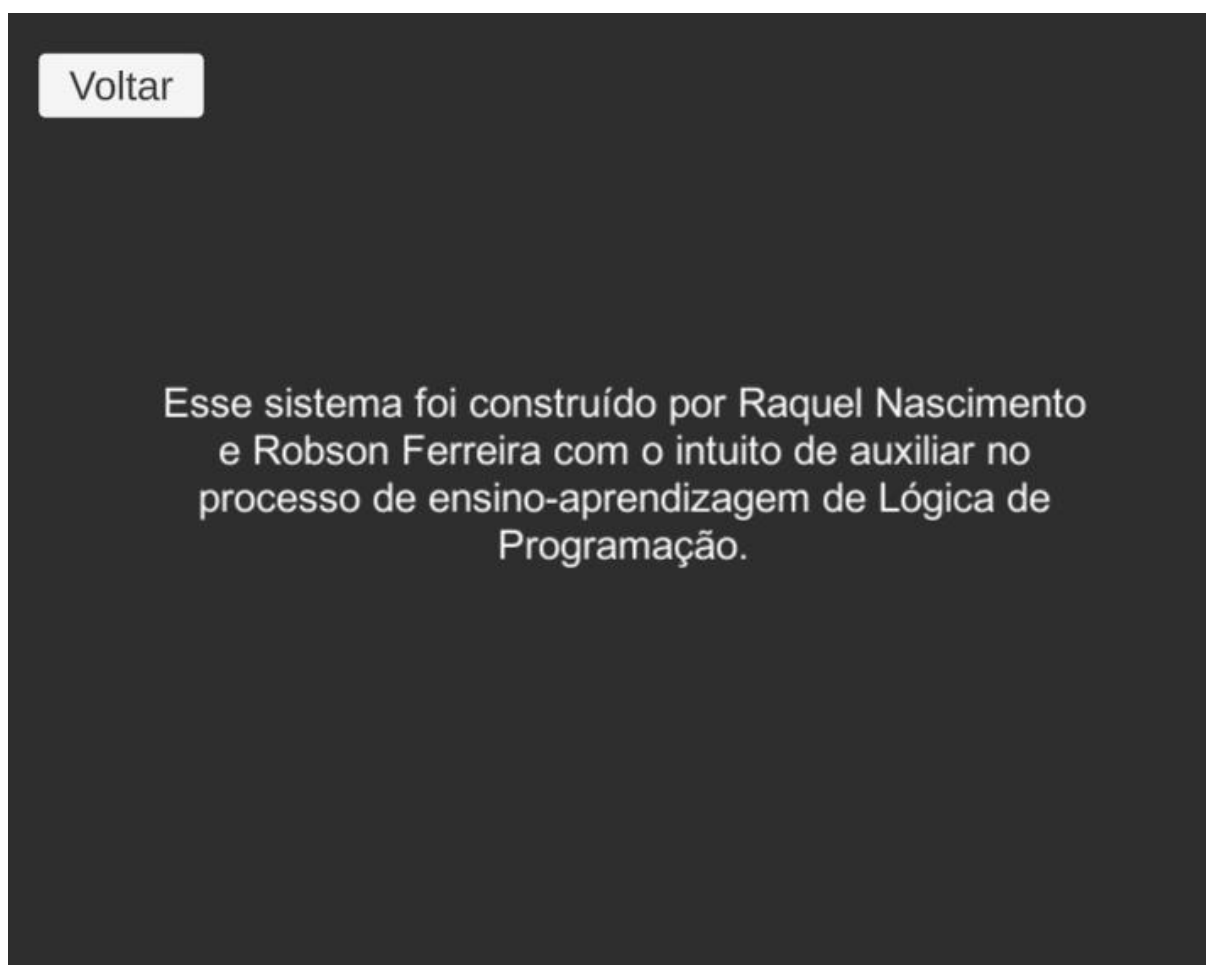
5.10.1. TELAS DE PROTÓTIPO

Figura 14 - Tela Inicial



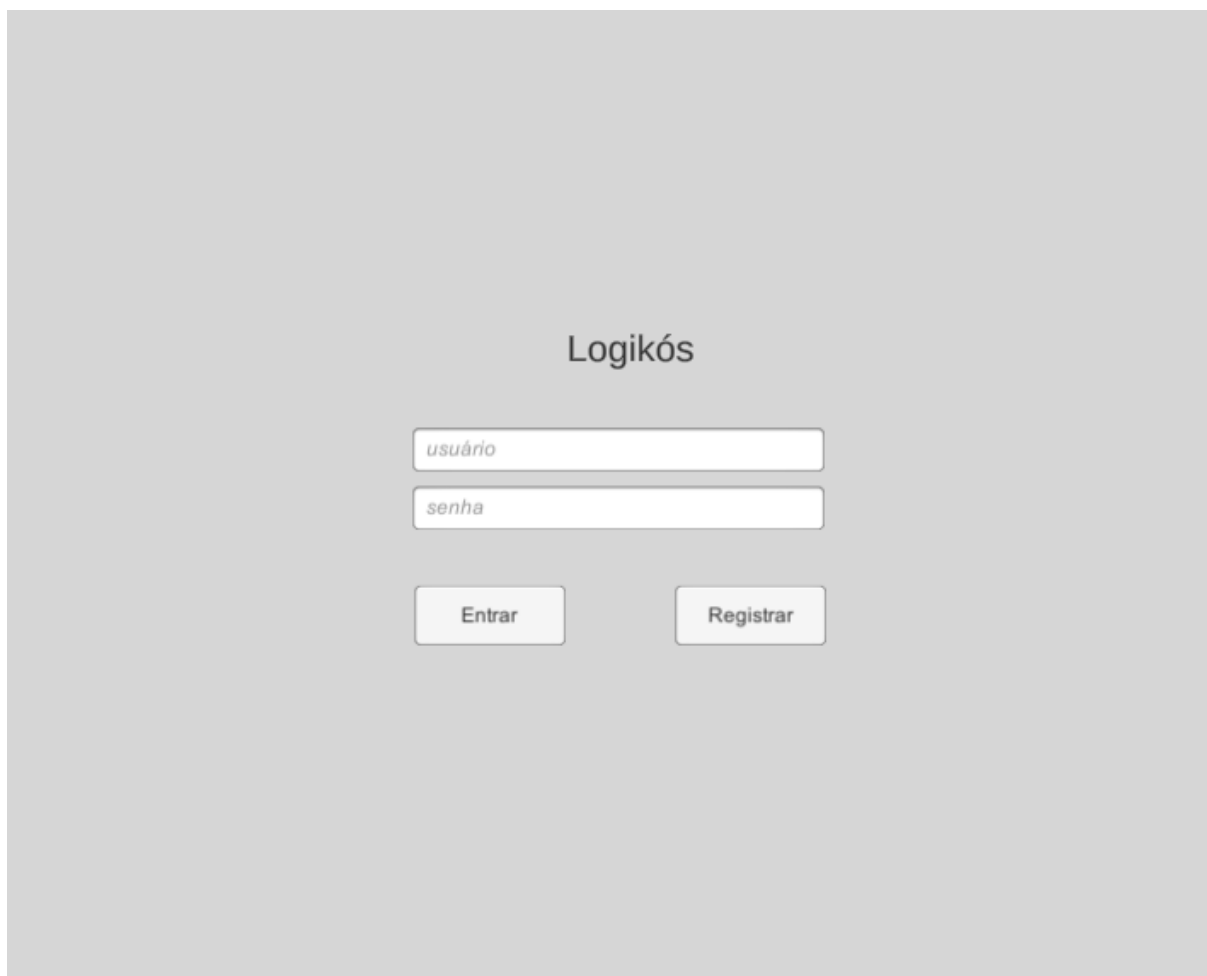
Fonte: Autoria Própria, 2021.

Figura 15 - Tela Sobre (Informativa)



Fonte: Autoria Própria, 2021.

Figura 16 - Tela Entrar



Logikós

usuário

senha

Entrar Registrar

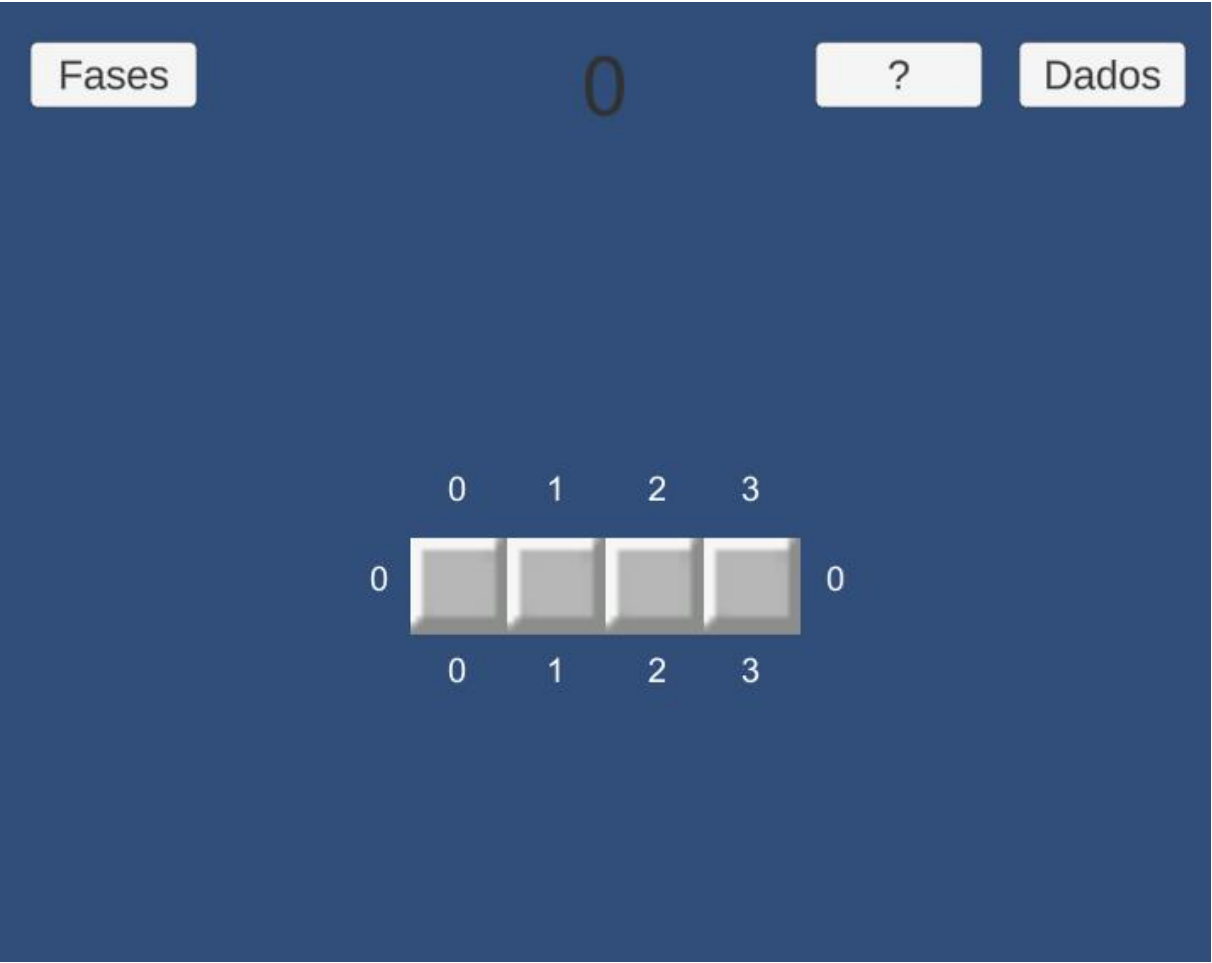
Fonte: Autoria Própria, 2021.

Figura 17 - Tela Seleção Fases



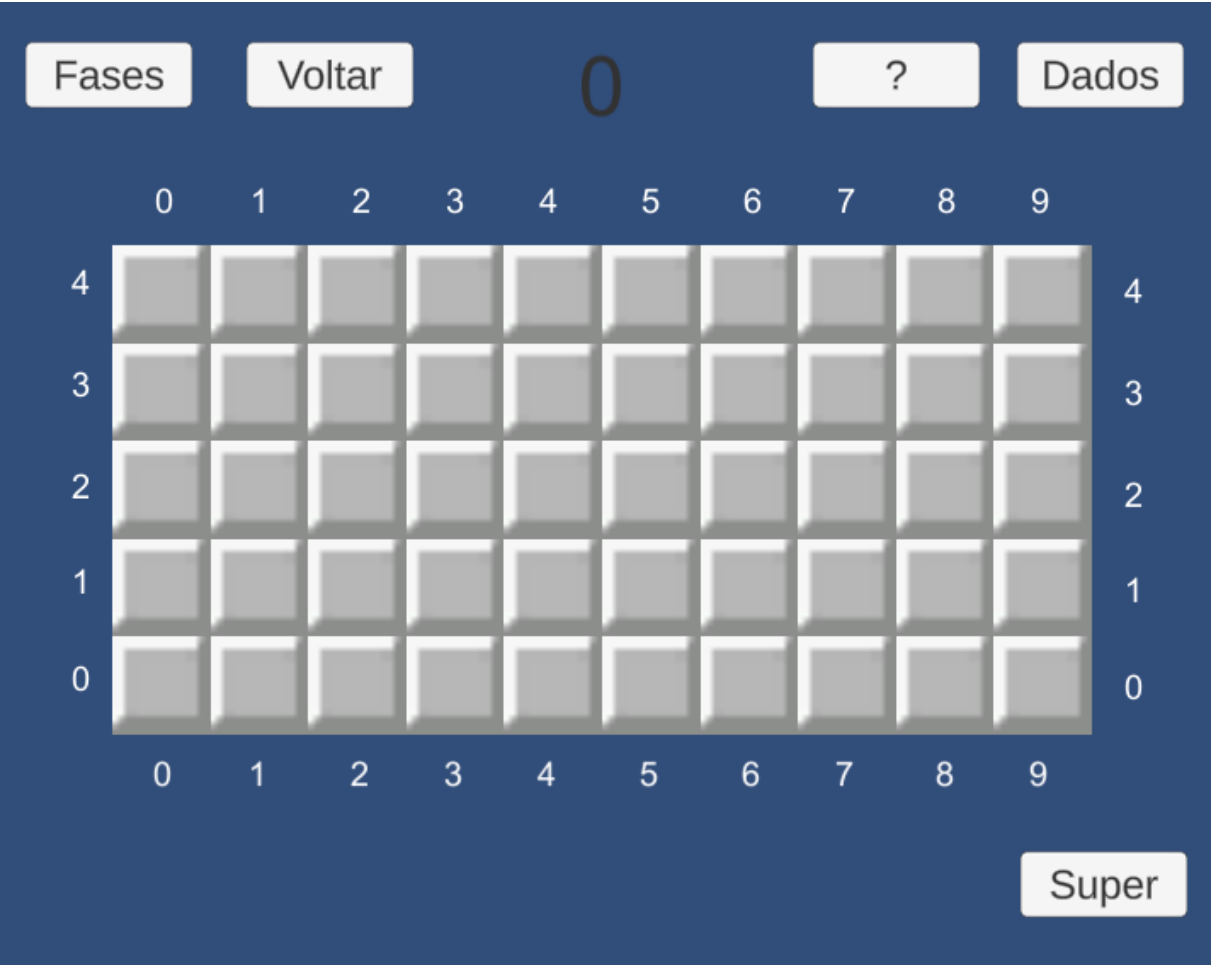
Fonte: Autoria Própria, 2021.

Figura 18 - Tela Aprendizagem



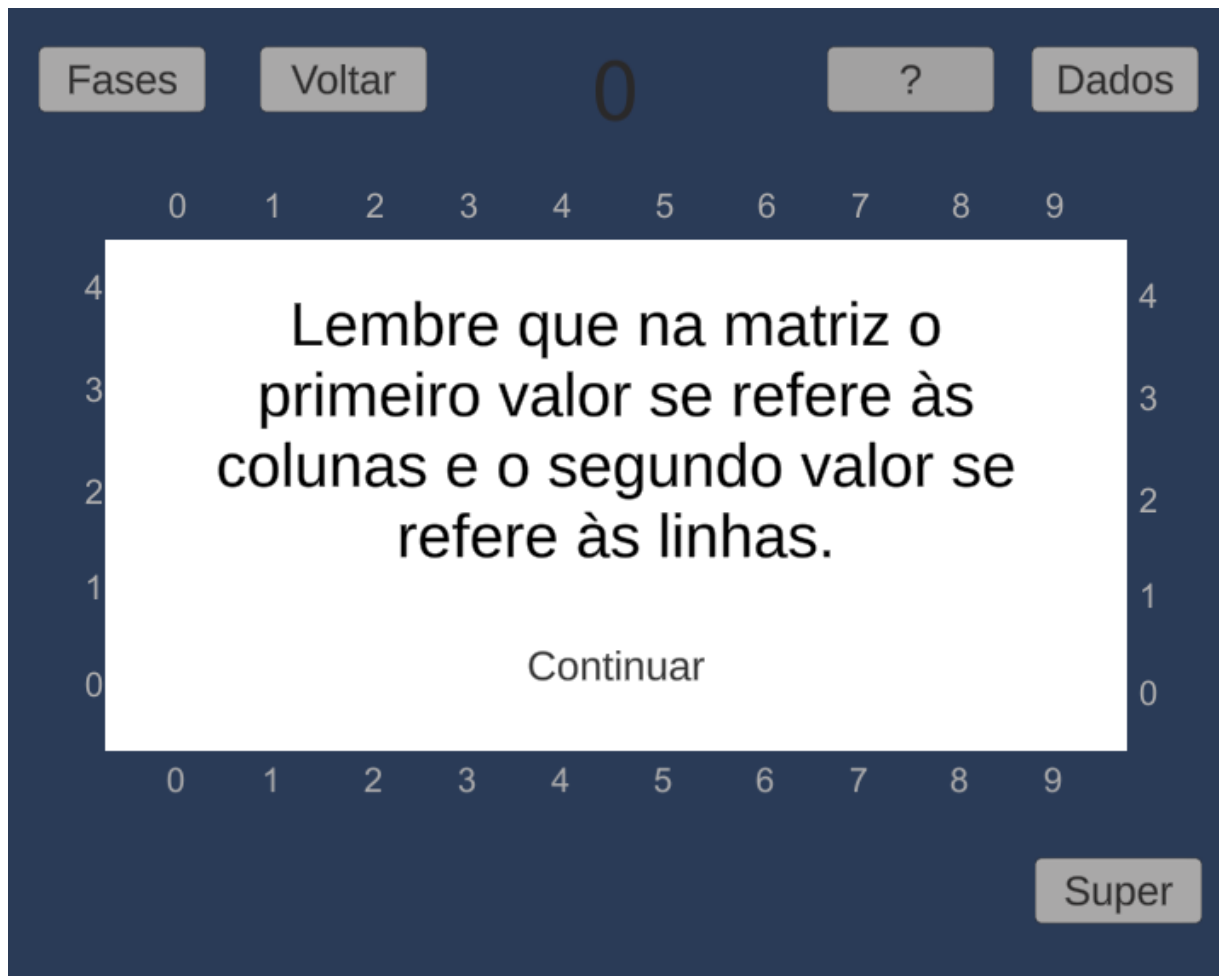
Fonte: Autoria Própria, 2021.

Figura 19 - Tela Fase 5



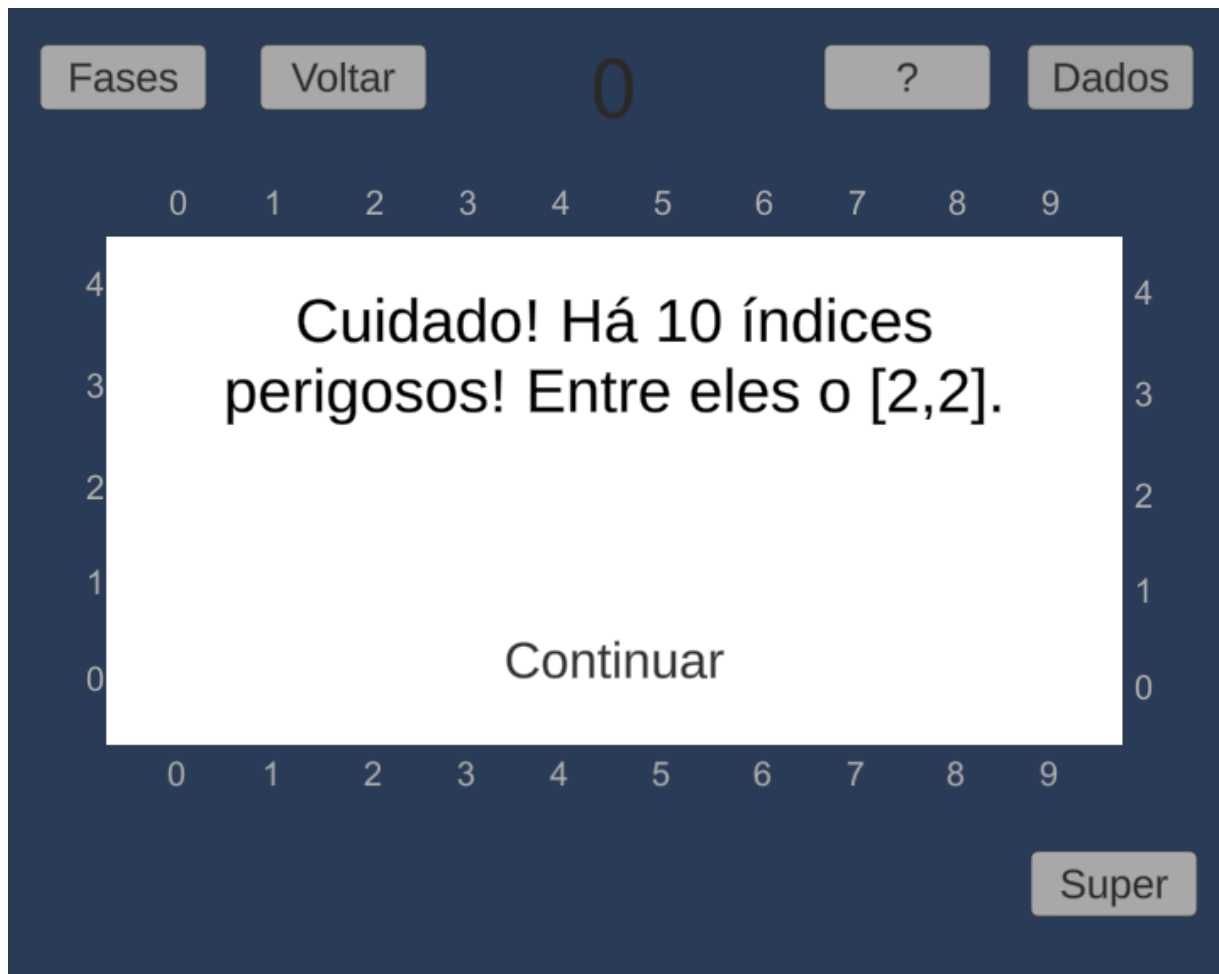
Fonte: Autoria Própria, 2021.

Figura 20 - Tela com Texto Dica



Fonte: Autoria Própria, 2021.

Figura 21 - Tela com Texto indicação do que fazer



Fonte: Autoria Própria, 2021.

6. DISCUSSÃO E CONSIDERAÇÕES FINAIS

O primeiro estágio trouxe resultados bem satisfatórios para a pesquisa, a partir de 206 respostas ao questionário que foi distribuído para os alunos de Tecnólogo em Análise e Desenvolvimento de Sistemas, das quais 72 foram de alunos do primeiro semestre e 22 foram do segundo semestre. A partir destas respostas pôde-se levantar que uma porcentagem substancial de alunos desse curso têm ou tiveram alguma dificuldade no aprendizado de lógica de programação, mostrando que, apesar de os métodos mais tradicionais de ensino proporcionarem bons resultados, é possível que melhorias sejam feitas no processo de ensino-aprendizagem visando atingir um grau ainda maior de aproveitamento.

Colocando em porcentagens e de acordo com o semestre, tem-se que os alunos que têm ou tiveram dificuldades representam: 58,5% do 1º semestre; 45,5% do 2º semestre; 44% do 3º semestre; 28% do 4º semestre; 24% do 5º semestre; e 48% do 6º semestre ou mais. Portanto, como era esperado no início da pesquisa, vê-se que o primeiro semestre é o que mais necessita de apoio no aprendizado de Lógica de Programação, até porque para muitos é o primeiro contato com essa disciplina.

Ainda a partir dos resultados do Primeiro Estágio, viu-se a necessidade de um foco maior para alguns temas de Lógica de Programação no desenvolvimento do jogo proposto, como Arrays (em que 45% dos alunos do primeiro semestre com dificuldade disseram ser o tema mais difícil) e Funções (em que 14% dos alunos com dificuldade disseram ser o tema mais difícil). Esses resultados podem ser verificados mais atentamente a partir de uma tabela disponível no Anexo I. Sendo assim, o planejamento para o desenvolvimento do jogo ganhou um direcionamento no que diz respeito a temas iniciais abordados e onde focar maior esforço de trabalho ao desenvolver o sistema LOGIKÓS, que ganhou nas primeiras fases do jogo o tema de Arrays com vetores e matrizes. Atacando o problema maior logo de início, poderá se trazer maior aderência ao sistema quando posto em uso pelos estudantes.

Um dado interessante é que era de se esperar que os estudantes que possuíam formação anterior em algum curso da área das Ciências Exatas são os que encontraram menos dificuldades na disciplina de Lógica de Programação, com o

percentual de 40%. Isso pode se explicar ao considerar que muitos desse grupo podem ter tido grande contato com disciplinas que usam a lógica como parte essencial, como a Matemática e Física. Em contrapartida, alunos com formação na área das Ciências Humanas tiveram um percentual de 50%, maior que o percentual de todos os alunos de modo geral (com e sem formação), que ficou em 45,5%. Talvez esse resultado para o grupo possa ser derivado do costume de tratar assuntos de modo não binário e mais relacional e incerto quanto aos parâmetros utilizados (como o ser humano e seus inúmeros comportamentos).

Para finalizar a análise do primeiro estágio, tem-se que 99% dos alunos do primeiro semestre e 93,3% do total dos entrevistados responderam que um jogo interativo desenvolvido exclusivamente para o ensino de Lógica de Programação iria ajudar no aprendizado dessa disciplina, viabilizando o projeto ao considerar sua necessidade para esse público-alvo (estudantes de Análise e Desenvolvimento de Sistemas, com foco nos estudantes de primeiro semestre).

O segundo estágio recebeu o desenvolvimento da aplicação propriamente dita, em que se tem uma estruturação dos códigos do jogo, ideias de fases e interações conforme proposto na documentação mostrada nos capítulos anteriores e no ambiente selecionado. Sendo assim, encontra-se pronta a aplicação-protótipo para que possa ser melhorada e aplicada na prática, enviando aos alunos que fazem parte do público-alvo. Por se tratar de um protótipo, a aplicação aqui desenvolvida não possui, ainda, todos os elementos necessários para seu uso total e ativo de modo a coletar todos os benefícios levantados no decorrer da pesquisa de que um sistema lúdico para o auxílio no aprendizado de Lógica de Programação pode oferecer.

Como recomendação de trabalho posterior, levantou-se a possibilidade de uma Terceira Fase em que poderá ser realizada a divulgação e posterior coleta de resultados do sistema desenvolvido. Desse modo, os alunos do primeiro semestre do curso de Tecnólogo em Análise e Desenvolvimento de Sistemas – público-alvo da pesquisa da Primeira Fase – poderão experimentar o sistema LOGIKÓS e praticar e construir seu conhecimento na disciplina de Lógica de Programação. Também, os professores relacionados às matérias na universidade poderão disponibilizar o sistema, bem como manter métricas de uso por aluno. Assim, poderá ser possível

analisar e estabelecer o impacto real da plataforma no processo de ensino-aprendizagem e na construção de conhecimento por parte dos estudantes.

7. REFERÊNCIAS

AIRES, Regina; MOREIRA, Fernanda; FREIRE Patrícia. INDÚSTRIA 4.0: DESAFIOS E TENDÊNCIAS PARA A GESTÃO DO CONHECIMENTO. In SUCEG (Seminário Universidades Corporativas e Escolas de Governo) Industria 4.0: Desafios e Tendências para a gestão do Conhecimento, 2017. Disponível em: <<https://anais.suceg.ufsc.br/index.php/suceg/article/view/49/17>> Acesso em: 25/04/2021.

ANDRIOLA, W.B.; ANDRIOLA, C. G.; MOURA, C. P. Opiniões de docentes e de coordenadores acerca do fenômeno da evasão discente dos cursos de graduação da Universidade Federal do Ceará (UFC). Ensaio: avaliação políticas públicas Educacionais. Rio de Janeiro, v 14, n 52, p 365-382. 2006.

BOLLER, Sharon; KAPP, Karl. Jogar para aprender. Editora DVS, 2018

DIAS, ANDRÉ. Conceitos Básicos de Controle de Versão de Software — Centralizado e D. Blog Pronus. 2016. Disponível em:<<https://blog.pronus.io/posts/conceitos-basicos-de-controle-de-versao-de-software-centralizado-e-distribuido/>>. Acesso em: 18 Nov. 2020.

EUGENIO, Tiago. Aula em Jogo: descomplicando a gamificação para educadores. Editora Évora, 2020.

FADEL, Aline Cristine; SILVEIRA, Henrique da Mota. Metodologias ágeis no contexto de desenvolvimento de software: XP, Scrum e Lean. Universidade Estadual de Campinas – UNICAMP, Limeira: 2010.

GARDNER, H. *Inteligências Múltiplas: a teoria na prática*. Porto Alegre: Artes Médicas, 1995.

HUIZINGA, Johan. *Homo Ludens*. Perspectiva, 2000. Disponível em <http://jnsilva.ludicum.org/Huizinga_HomoLudens.pdf> Acesso em: 25/04/2021.

LORENZONI, Marcela. Gamificação: o que é e como pode transformar a aprendizagem. Geekie. Disponível em: <<https://site.geekie.com.br/blog/gamificacao/>> Acesso em: 01/05/2021.

MACORATTI, José Carlos. UML – Unified Modeling Language e Visual Modeler (Visual Basic 6). Macoratti.net. Artigo online. Disponível em: <http://www.macoratti.net/uml_vb.htm> Acesso em: 27/06/2021.

MELLO, Simone Portella Teixeira. O fenômeno evasão nos cursos superiores de tecnologia: um estudo de caso em uma universidade pública no Sul do Brasil. XIII Coloquio de Gestión Universitaria em Américas: 2013. Disponível em: <<https://repositorio.ufsc.br/bitstream/handle/123456789/113096/2013129%20-%20O%20fen%c3%b4meno%20evas%c3%a3o%20nos%20cursos%20superiores.pdf?sequence=1&isAllowed=y>> Acesso em: 23/11/2020.

OBERLIN, Tim. Gamification. Behance. Disponível em: <<https://www.behance.net/gallery/35662235/Gamification>> Acesso em: 01/05/2021

PLATT NETO, O. A. da; CRUZ, F.; PFITSCHER, E. D. Utilização de metas de desempenho ligadas à taxa de evasão escolar nas universidades públicas. Revista de Educação e pesquisa em Contabilidade. Brasília, v. 2, p. 54-74. mai. -ago. 2008.

SOARES, M. DOS S. Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software. INFOCOMP Journal of Computer Science, v. 3, n. 2, p. 8-13, 1 Nov. 2004.

8. ANEXOS

Neste tópico estarão os anexos relacionados ao projeto.

8.1. TABELA DE RESPOSTAS DO QUESTIONÁRIO DA ETAPA 1

Em qual semestre você está?	Qual sua faixa etária?	Você tem formação em outra área?	Qual(is) sua(s) outra(s) área(s) de formação?	Você sente/sentiu alguma dificuldade ao aprender lógica de programação?	Qual foi sua maior dificuldade?	Você acredita que um jogo interativo desenvolvido exclusivamente para o ensino de Lógica de Programação iria ajudar no aprendizado dessa disciplina?
6 (ou mais)	25-35	Não		Não		Não
6 (ou mais)	20-25	Sim	Ciências Humanas	Sim	Arrays (vetores/matrices)	Sim
6 (ou mais)	20-25	Não		Não		Sim
3	25-35	Sim	Ciências Biológicas	Sim	Arrays (vetores/matrices)	Sim
6 (ou mais)	25-35	Não		Não		Sim
6 (ou mais)	20-25	Não		Não		Sim
4	25-35	Sim	Ciências Humanas	Sim	Arrays (vetores/matrices)	Sim

2	20-25	Não		Sim	Variáveis	Não
6 (ou mais)	20-25	Não		Não		Sim
6 (ou mais)	20-25	Não		Sim	Estruturas de decisão (if/switch)	Sim
6 (ou mais)	20-25	Não		Sim	Procedimentos	Sim
5	20-25	Não		Não		Sim
5	20-25	Não		Não		Sim
6 (ou mais)	20-25	Não		Sim	Estruturas de repetição (for/while)	Sim
4	20-25	Não		Não		Sim
5	Menos de 20	Não		Sim	Arrays (vetores/matrizes)	Sim
6 (ou mais)	20-25	Não		Sim	Estruturas de repetição (for/while)	Não
3	20-25	Não		Não		Sim
4	20-25	Não		Sim	Funções	Sim
6 (ou mais)	20-25	Não		Sim	Estruturas de repetição (for/while)	Sim
3	Menos de 20	Não		Sim	Arrays (vetores/matrizes)	Sim
5	20-25	Não		Não		Sim
3	Menos de 20	Não		Não		Sim

2	25-35	Sim	Ciências Exatas			
5	20-25	Não		Não		Sim
4	25-35	Sim	Ciências Humanas	Não		Sim
3	20-25	Não		Sim	Estruturas de decisão (if/switch)	Sim
5	Mais de 35	Sim	Ciências Biológicas	Sim	Funções	Sim
5	Menos de 20	Não		Não		Sim
3	20-25	Não		Sim	Variáveis	Sim
3	20-25	Não		Sim	Variáveis	Sim
6 (ou mais)	25-35	Sim	Ciências Exatas	Não		Sim
3	Menos de 20	Não		Não		Sim
6 (ou mais)	25-35	Não		Não		Sim
3	Menos de 20	Não		Não		Sim
3	Menos de 20	Não		Não		Sim
5	20-25	Não		Sim	Arrays (vetores/matrices)	Sim
5	25-35	Sim	Ciências Exatas	Sim	Funções	Sim

1	25-35	Sim	Ciências Humanas	Sim	Funções	Sim
5	20-25	Sim	Ciências Exatas	Não		Sim
5	20-25	Não		Não		Sim
5	25-35	Sim	Ciências Humanas	Não		Sim
4	Menos de 20	Não		Não		Sim
1	20-25	Sim	Ciências Humanas	Não		Sim
6 (ou mais)	Mais de 35	Sim	Ciências Biológicas	Sim	Arrays (vetores/matrices)	Sim
5	25-35	Sim	Ciências Exatas	Sim	Arrays (vetores/matrices)	Sim
4	Mais de 35	Sim	Ciências Exatas	Não		Não
6 (ou mais)	25-35	Não		Sim	Estruturas de repetição (for/while)	Sim
6 (ou mais)	Mais de 35	Não		Não		Sim
5	25-35	Não		Não		Sim
5	Menos de 20	Não		Não		Sim
2	Mais de 35	Sim	Ciências Humanas	Sim	Arrays (vetores/matrices)	Sim

2	20-25	Sim	Ciências Humanas	Sim	Arrays (vetores/matricizes)	Sim
5	25-35	Sim	Ciências Exatas	Não		Sim
2	Menos de 20	Não		Sim	Arrays (vetores/matricizes)	Sim
2	25-35	Sim	Ciências Exatas	Não		Sim
2	Menos de 20	Não		Não		Sim
2	Menos de 20	Não		Não		Sim
6 (ou mais)	25-35	Sim	Ciências Exatas	Sim	Arrays (vetores/matricizes)	Sim
4	20-25	Não		Não		Sim
1	20-25	Não		Não		Sim
5	25-35	Não		Não		Sim
6 (ou mais)	25-35	Não		Sim	Arrays (vetores/matricizes)	Sim
5	Mais de 35	Sim	Ciências Humanas	Não		Sim
6 (ou mais)	20-25	Não		Não		Sim
6 (ou mais)	25-35	Sim	Ciências Exatas	Não		Sim
1	25-35	Não		Sim	Variáveis	Sim

4	Menos de 20	Não		Não		Sim
6 (ou mais)	20-25	Não		Não		Não
6 (ou mais)	20-25	Não		Sim	Funções	Não
6 (ou mais)	Menos de 20	Não		Não		Sim
3	Menos de 20	Não		Não		Sim
3	20-25	Sim	Ciências Humanas	Não		Sim
4	25-35	Sim	Ciências Humanas	Não		Sim
1	25-35	Sim	Ciências Humanas	Não		Sim
4	25-35	Sim	Ciências Humanas	Sim	Funções	Sim
2	25-35	Sim	Ciências Humanas	Sim	Variáveis	Sim
1	Menos de 20	Não		Sim	Arrays (vetores/matrices)	Sim
6 (ou mais)	Mais de 35	Sim	Ciências Biológicas	Não		Sim
1	Menos de 20	Não		Não		Sim
5	25-35	Sim	Ciências Humanas	Sim	Procedimentos	Sim

6 (ou mais)	20-25	Não		Sim	Funções	Sim
6 (ou mais)	20-25	Não		Sim	Procedimentos	Sim
6 (ou mais)	20-25	Não		Não		Sim
1	Menos de 20	Não		Sim	Estruturas de decisão (if/switch)	Sim
3	Menos de 20	Não		Sim	Procedimentos	Sim
4	25-35	Não		Sim	Funções	Sim
5	25-35	Sim	Ciências Exatas	Não		Sim
6 (ou mais)	Mais de 35	Sim	Ciências Exatas	Não		Sim
5	25-35	Sim	Ciências Humanas	Não		Sim
4	20-25	Não		Não		Sim
5	Mais de 35	Sim	Ciências Exatas	Não		Sim
2	25-35	Não		Não		Sim
2	Menos de 20	Sim	Ciências Exatas	Não		Sim
5	25-35	Não		Não		Não
2	20-25	Sim	Ciências Exatas	Não		Sim
1	20-25	Não		Sim	Estruturas de decisão (if/switch)	Sim
2	20-25	Não		Não		Não

2	Menos de 20	Não		Sim	Arrays (vetores/matricizes)	Sim
6 (ou mais)	Mais de 35	Não		Não		Sim
5	20-25	Sim	Ciências Exatas	Sim	Funções	Sim
5	Mais de 35	Sim	Ciências Exatas	Não		Sim
1	20-25	Não		Sim	Arrays (vetores/matricizes)	Sim
1	20-25	Sim	Ciências Exatas	Sim	Funções	Sim
1	Menos de 20	Não		Não		Sim
4	25-35	Sim	Ciências Humanas	Não		Sim
5	20-25	Sim	Ciências Humanas	Não		Sim
3	20-25	Não		Não		Sim
5	20-25	Não		Não		Sim
4	Mais de 35	Sim	Ciências Humanas	Não		Não
5	Menos de 20	Não		Não		Sim
1	25-35	Sim	Ciências Humanas	Não		Sim
6 (ou mais)	25-35	Não		Sim	Procedimentos	Sim

5	Mais de 35	Sim	Ciências Biológicas	Não		Sim
1	25-35	Não		Sim	Procedimentos	Sim
4	25-35	Sim	Ciências Exatas	Não		Sim
4	25-35	Sim	Ciências Exatas	Não		Sim
4	25-35	Não		Sim	Arrays (vetores/matrices)	Sim
3	20-25	Não		Não		Sim
6 (ou mais)	20-25	Não		Sim	Funções	Sim
6 (ou mais)	Mais de 35	Sim	Ciências Humanas	Não		Sim
2	20-25	Sim	Ciências Exatas	Não		Não
6 (ou mais)	20-25	Não		Sim	Arrays (vetores/matrices)	Sim
3	Menos de 20	Não		Não		Sim
2	25-35	Sim	Ciências Humanas	Não		Sim
2	Menos de 20	Não		Não		Sim
6 (ou mais)	20-25	Não		Sim	Arrays (vetores/matrices)	Sim

1	Menos de 20	Não		Sim	Variáveis	Sim
4	25-35	Não		Não		Sim
1	Menos de 20	Sim	Ciências Exatas	Sim	Arrays (vetores/matrices)	Sim
6 (ou mais)	Mais de 35	Não		Não		Sim
2	20-25	Não		Não		Sim
1	Menos de 20	Não		Sim	Variáveis	Sim
4	20-25	Não		Não		Sim
1	Menos de 20	Não		Não		Sim
1	25-35	Sim	Ciências Biológicas	Sim	Estruturas de repetição (for/while)	Sim
2	20-25	Não		Não		Sim
3	25-35	Sim	Ciências Exatas	Não		Sim
3	20-25	Não		Não		Sim
3	25-35	Sim	Ciências Biológicas	Sim	Arrays (vetores/matrices)	Sim
3	Menos de 20	Não		Não		Sim
3	20-25	Não		Sim	Arrays (vetores/matrices)	Sim

3	25-35	Sim	Ciências Humanas	Sim	Arrays (vetores/matrices)	Sim
3	Menos de 20	Não		Sim	Estruturas de repetição (for/while)	Sim
4	25-35	Não		Não		Sim
4	Mais de 35	Sim	Ciências Humanas	Sim	Arrays (vetores/matrices)	Sim
3	Mais de 35	Sim	Ciências Exatas	Sim	Arrays (vetores/matrices)	Sim
6 (ou mais)	20-25	Não		Não		Sim
1	20-25	Sim	Ciências Exatas	Não		Sim
3	Menos de 20	Não		Não		Sim
6 (ou mais)	25-35	Sim	Ciências Humanas			
1	Mais de 35	Não		Não		Sim
1	Menos de 20	Não		Não		Sim
1	Menos de 20	Não		Sim	Arrays (vetores/matrices)	Sim
1	Menos de 20	Não		Sim	Arrays (vetores/matrices)	Sim

1	Menos de 20	Não		Não		Sim
1	Menos de 20	Não		Sim	Funções	Sim
1	Menos de 20	Não		Não		Sim
1	20-25	Não		Não		Sim
1	20-25	Não		Não		Sim
1	20-25	Não		Sim	Arrays (vetores/matrices)	Sim
1	Menos de 20	Não		Não		Sim
1	20-25	Sim	Ciências Humanas	Sim	Estruturas de repetição (for/while)	Sim
1	25-35	Não		Não		Sim
1	Menos de 20	Não		Sim	Funções	Sim
1	25-35	Não		Não		Sim
1	25-35	Não		Sim	Arrays (vetores/matrices)	Sim
1	20-25	Não		Sim	Arrays (vetores/matrices)	Sim
1	20-25	Não		Não		Sim

1	Menos de 20	Não		Não		Sim
1	Menos de 20	Não		Sim	Estruturas de repetição (for/while)	Sim
1	Menos de 20	Não		Não		Sim
1	Menos de 20	Não		Não		Sim
1	Mais de 35	Sim	Ciências Humanas	Sim	Variáveis	Sim
1	Menos de 20	Não		Sim	Arrays (vetores/matrices)	Sim
1	Menos de 20	Não		Sim	Arrays (vetores/matrices)	Sim
1	Menos de 20	Não		Sim	Arrays (vetores/matrices)	Sim
1	Menos de 20	Sim	Ciências Exatas	Não		Sim
1	Menos de 20	Sim	Ciências Humanas	Sim	Arrays (vetores/matrices)	Não
1	20-25	Não		Sim	Estruturas de repetição (for/while)	Sim

1	Menos de 20	Não		Sim	Procedimentos	Sim
1	Menos de 20	Não		Sim	Arrays (vetores/matricizes)	Sim
1	20-25	Sim	Ciências Humanas	Não		Sim
1	Menos de 20	Não		Não		Sim
1	25-35	Sim	Ciências Humanas	Sim	Arrays (vetores/matricizes)	Sim
1	Menos de 20	Não		Sim	Arrays (vetores/matricizes)	Sim
1	Menos de 20	Não		Não		Sim
2	Mais de 35	Sim	Ciências Exatas	Sim	Arrays (vetores/matricizes)	Sim
1	25-35	Não		Sim	Variáveis	Sim
1	20-25	Não		Sim	Estruturas de repetição (for/while)	Sim
1	20-25	Não		Não		Sim
2	25-35	Sim	Ciências Exatas	Sim	Arrays (vetores/matricizes)	Sim

1	Menos de 20	Não		Sim	Funções	Sim
1	20-25	Não		Sim	Arrays (vetores/matricizes)	Sim
1	20-25	Não		Não		Sim
1	20-25	Não		Sim	Estruturas de repetição (for/while)	Sim
1	20-25	Não		Não		Sim
2	20-25	Sim		Sim	Arrays (vetores/matricizes)	Sim
1	Mais de 35	Não		Sim	Arrays (vetores/matricizes)	Sim
1	20-25	Não		Sim	Arrays (vetores/matricizes)	Sim
1	20-25	Não		Sim	Arrays (vetores/matricizes)	Sim
1	20-25	Não		Sim	Procedimentos	Sim
1	Menos de 20	Não		Não		Sim
1	25-35	Sim	Ciências Exatas	Sim	Procedimentos	Sim
1	Menos de 20	Não		Não		Sim

	Meno					
	s de					
1	20	Não		Sim	Funções	Sim