NÚCLEO INTERDISCIPLINAR DE PESQUISA

UNIDESC
Centro Universitário de Desenvolvimento do Centro-Oeste

Curso de Sistemas de Informação Artigo Original

ANÁLISE DA VIABILIDADE TÉCNICA DE UM JOGO EDUCACIONAL PARA AUXILIAR CRIANÇAS NO ENSINO DE LÓGICA DE PROGRAMAÇÃO

ANALYSIS OF THE TECHNICAL FEASIBILITY OF AN EDUCATIONAL GAME TO HELP CHILDREN IN THE TEACHING OF PROGRAMMING LOGIC

Leandro Junio Oliveira Alves[1], Joao Victor Lopo do Nascimento Alves Pereira[1], Ana Carolina Cintra Faria[2]
1 Alunos do Curso de Sistemas de Informação
2 Coordenadora do Curso de Sistemas de Informação

RESUMO

Este trabalho constitui-se na análise da viabilidade técnica para o desenvolvimento de um MVP de jogo educacional para auxiliar crianças no ensino de lógica de programação. A análise realizada permeia as etapas de desenvolvimento de um jogo educacional capaz de auxiliar crianças a compreenderem lógica de programação de forma simplificada e interativa. A utilização da metodologia Lean Startup no desenvolvimento do jogo resultou na criação da primeira versão de um produto mínimo viável para coleta de métricas da primeira versão, ajustes e validação da ferramenta. O resultado foi obtido após a execução completa do ciclo de feedback Construir-Medir-Aprender, onde o feedback gerado por uma amostra de pessoas indicou pontos de ajuste para melhorias na primeira versão do jogo. Foi constatada a viabilidade técnica da construção de um MVP de jogo educacional capaz de auxiliar no ensino de lógica de programação para crianças. A implementação do Lean Startup durante o processo de desenvolvimento do MVP foi fundamental para alcançar os objetivos específicos e o geral, resultando na criação do jogo.

Palavras-Chave: jogo educacional; lógica de programação; análise de viabilidade; MVP.

ABSTRACT

This work is constituted in the analysis of the technical viability for the development of an MVP of educational game to help children in the teaching of programming logic. The analysis carried out permeates the development stages of an educational game capable of helping children to understand programming logic in a simplified and interactive way. The use of the Lean Startup methodology in the development of the game resulted in the creation of the first version of a minimum viable product for the collection of metrics from the first version, adjustments and validation of the tool. The result was obtained after the complete execution of the Build-Measure-Learn feedback loop, where the feedback generated by a sample of people indicated adjustment points for improvements in the first version of the game. The technical viability of building an educational game MVP capable of assisting in teaching logic programming for children was verified. The implementation of Lean Startup during the MVP development process was essential to achieve the specific and general objectives, resulting in the creation of the game.

Keywords: educational game; programming logic; analysis of the technical feasibility; MVP.

Contato: leandrojoapi@gmail.com, jvvlopo@gmail.com, coord.ti@unidesc.edu.br.

INTRODUÇÃO

A evolução constante da tecnologia vem criando e moldando novas formas de trabalho e assim gerando uma grande demanda crescente de profissionais cada vez mais qualificados, logo, influencia diretamente na formação desses profissionais, ou seja, torna-se básico que os sistemas de ensino de um país se adéquem também a essas novas necessidades de mercado.

Além de proporcionar melhor desempenho e fortalecer a saúde mental, desenvolver o raciocínio lógico tem grande impacto na atividade profissional. Portanto, o pensamento crítico e o raciocínio são habilidades que variam muito, independentemente da especialização e do campo de atuação (BLOG PORTAL PÓS, 2021).

Segundo Girard, Ecalle e Magnan (2013, tradução nossa) em nove estudos analisados, tendo como maior público alvo crianças e jovens em idade escolar, a utilização de jogos digitais educativos demonstraram-se ser poderosas ferramentas para auxiliar no processo de aprendizagem dos indivíduos.

Assim como a leitura, escrita e aritmética, o pensamento computacional deveria estar incluso nas habilidades analíticas de qualquer criança. Pensamento computacional é uma habilidade fundamental para todos, não apenas para cientistas da computação, envolve resolver problemas, projetar sistemas e entender o comportamento humano. Pensar computacionalmente é pensar em termos de prevenção, proteção e recuperação nos piores cenários, contenção de danos e correção de erros, planejando e aprendendo na presença da incerteza (WING 2006, tradução nossa).

O emprego de jogos educacionais como ferramenta para o ensino de lógica de programação tem por finalidade a assimilação funcional num exercício de ações individuais já aprendidas. Os jogos são uma forma de imprimir sentimento de prazer e domínio das ações por parte de quem aprende (MEDEIROS, 2014).

Os jogos digitais educacionais podem trazer diversos benefícios ao processo de ensino e aprendizagem como, por exemplo, ser um facilitador do aprendizado, do desenvolvimento (MEDEIROS, 2014).

Considerando este cenário, uma questão é levantada: Como realizar a introdução de crianças a conceitos como lógica de programação, que assim como a leitura, escrita e aritmética, vem se tornando cada vez mais necessária? E mais especificamente: Qual a viabilidade técnica de um jogo educacional para auxiliar crianças no ensino de lógica de programação? Uma resposta para essa pergunta seria que, além de apresentar a lógica de programação para crianças, podemos também ensiná-las de forma mais simples e satisfatória, por meio de jogos digitais educativos, pois, segundo Vygotsky (2007) jogos além de serem motivadores, contribuem para facilitar a aprendizagem, estimulam a mente e intelecto do jogador, e podem aumentar a eficiência em armazenar o que foi aprendido. Tendo em vista estes benefícios, podem ser consideradas ferramentas eficientes para o processo de aprendizagem.

Sendo assim, o objetivo geral deste artigo é analisar a viabilidade técnica para o desenvolvimento de um MVP de jogo educacional para auxiliar crianças no ensino de

lógica de programação. E mais especificamente pretende-se: i) analisar a fundamentação teórica referente a pensamento computacional e jogos didáticos; ii) desenvolver um MVP voltado para o desenvolvimento do pensamento computacional em crianças, através de jogos didáticos; iii) empregar a metodologia Lean Startup nas fases de desenvolvimento do MVP; iv) realizar uma revisão de literatura relacionado a utilização de lógica de programação num contexto educacional; v) projetar os módulos a serem desenvolvidos e modelar o MVP; vi) validar a fase do jogo com profissionais da educação.

PROCEDIMENTOS METODOLÓGICOS

A metodologia de pesquisa empregada para caracterizar o recorte metodológico deste trabalho foi baseada em Turrioni e Mello (2012):

Natureza: A pesquisa de natureza aplicada pode ser caracterizada por seus objetivos mais comerciais, seguindo a necessidade do mercado durante o desenvolvimento de seus produtos ou serviços, onde seus resultados são aplicados de imediato no mercado, visando solucionar problemas já existentes.

Dessa forma, podemos caracterizar a seguinte pesquisa como de natureza aplicada, pois a intenção é conduzir um desenvolvimento prático, visando a entrega de um produto mínimo viável.

Objetivos: Pesquisas do tipo exploratória costumam envolver levantamentos bibliográficos, buscando entrevistas com pessoas já familiarizadas com o problema de pesquisa, e as utilizando como exemplo para uma análise mais aprofundada. Esse tipo de pesquisa visa construir hipóteses a partir do problema, tornando-os mais explícitos e aumentando sua familiaridade.

Portanto, podemos classificar esta pesquisa como exploratória, pois serão levantados materiais bibliográficos relacionados ao tema, assim como análise de mercado de modo a desenvolver uma ferramenta voltada para o ramo da educação.

Abordagem: O foco da abordagem qualitativa está no processo e no seu significado, através da interpretação dos fenômenos e na atribuição de seus resultados, considerando uma relação dinâmica entre o mundo real e o sujeito em questão.

Por possuir como fonte de dados o ambiente natural, esta abordagem não requer que sejam utilizados métodos ou técnicas estatísticas, tendo como núcleo principal o próprio pesquisador, e se tratando de uma pesquisa descritiva, onde os seus dados são analisados indutivamente.

Assim, definimos esta pesquisa como qualitativa, já que exploraremos o

comportamento dos consumidores, ao invés de somente mensurá-los.

Método: Emprego da metodologia Lean Startup no desenvolvimento do MVP.

O Lean Startup é baseado em 5 princípios sendo eles, o primeiro os empreendedores estão por toda parte sendo este conceito relacionado a qualquer instituição humana projetada para criar produtos e serviços sob condições de incerteza extrema, o segundo empreender é administrar, pois uma startup é uma instituição e não um produto desse modo um novo tipo de gestão deverá ser implementado devido ao contexto de incerteza, o terceiro é o aprendizado validado na busca para aprender a desenvolver um negócio sustentável, a aprendizagem pode ser validada por meio de experimentos frequentes, o que permite ao empreendedor testar diversos elementos de sua visão de negócio, o quarto é construir-medir-aprender startups devem construir produtos a partir de ideias, medir a reação de seus clientes e aprender se devem pivotar ou preserva-se, o quinto e último princípio é a contabilidade de inovação, medir progressos realizar a definição de marcos e a priorização do trabalho são assuntos que devem ser tratados com bastante foco (RIES, 2011).

Segundo Ries (2011) startups são catalisadoras que transformam ideias em produtos, e conforme os clientes vão interagindo com os produtos, acabam por gerar dados e feedback que podem ser tanto quantitativos quanto qualitativos. Esta informação é essencial para o processo de aprendizagem em como tornar uma startup em uma empresa sustentável, existem três etapas para neste processo, as quais podemos visualizar através do diagrama abaixo:

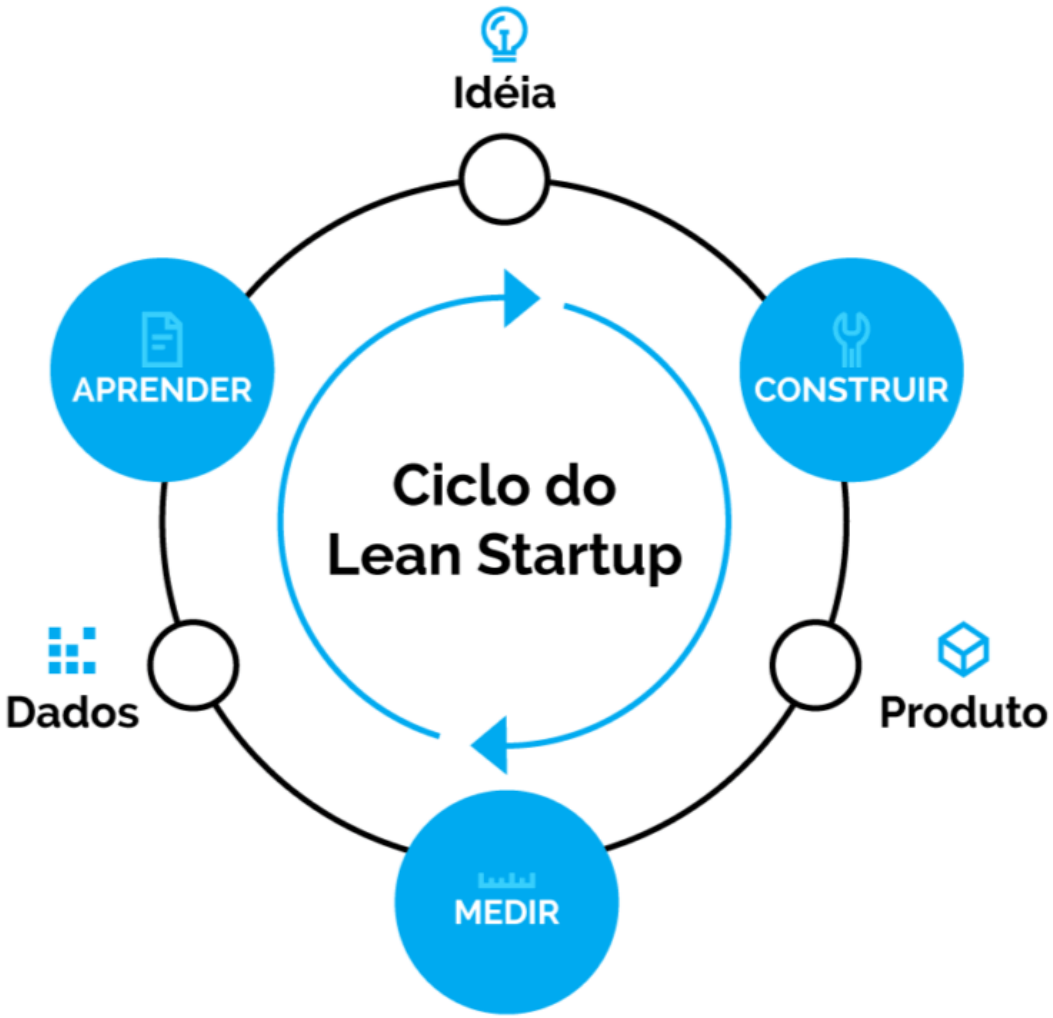

Figura 1 - Ciclo de feedback Construir-Medir-Aprender.
Fonte: Eric Ries (2011).

Etapa 1 (Construir): Será Inicialmente realizado o levantamento de requisitos e modelagem do processo sistêmico em sua totalidade. Já na etapa de construção, conforme o Lean Startup, o primeiro passo foi iniciar o mais rápido possível a fase de construção, a partir da ideia já estabelecida para o desenvolvimento do MVP, permitindo uma volta completa no fluxo de Construir-Medir-Aprender e mantendo uma quantidade mínima de esforço, com o menor tempo de desenvolvimento possível (RIES, 2011).

Etapa 2 (Medir): Na fase de produto aplicada ao nosso projeto, faremos a validação do MVP como amostra de um jogo digital educativo. Na etapa de medição a startup deve medir onde se encontra no momento tendo como referência seu plano de

negócios, análise dos números obtidos após a medição ajuda na criação de experiências para que os números reais se aproximem aos idealizados no plano de negócios (RIES, 2011).

Etapa 3 (Aprender): Conforme a interação dos clientes com o MVP, os dados e feedback que estes irão gerar serão analisados para contribuir no processo de aprendizagem sobre que rumos o produto precisa tomar ou se manter. Na etapa de aprendizagem a equipe deve demonstrar, baseado em experiências, que descobriu verdades valiosas referentes ao ponto de vista dos negócios, sejam elas presentes ou futuras (RIES, 2011).

Por meio do ciclo de feedback é possível realizar ajustes constantes, permitindo o aprendizado na identificação do momento em que devemos pivotar ou preservar-se no caminho atual (RIES, 2011).

O primeiro passo no Lean Startup é iniciar o mais rápido possível a fase de construção, onde será desenvolvido um MVP, permitindo uma volta completa no fluxo de Construir-Medir-Aprender, mantendo uma quantidade mínima de esforço, com o menor tempo de desenvolvimento possível.

FUNDAMENTAÇÃO TEÓRICA

MVP

O MVP surge a partir de uma metodologia chamada Lean Startup, desenvolvida por Eric Ries. Sua ideia era reduzir o processo de desenvolvimento de novos produtos, através da junção de experimentação de hipóteses orientada a negócios e lançamentos de produtos interativos.

O *Minimum Viable Product* (Mínimo Produto Viável) é o aspecto fundamental do conceito de *Build-Measure-Learn* (Construir-Medir-Aprender), um dos 5 núcleos da metodologia Lean Startup, onde o empreendedor deve transformar suas ideias em produtos, e a partir da resposta do cliente, decidir se dará continuidade ou não a sua ideia (RIES, 2011, tradução nossa).

Segundo Ries (2011, tradução nossa) o MVP se trata de uma versão com uma quantidade mínima de esforço e tempo de desenvolvimento do produto final, que apesar de carecer de muitos recursos essenciais, consegue ajudar empresários a iniciar o processo de aprendizagem o mais rápido possível. Um MVP é normalmente desenvolvido não apenas para responder a questões técnicas ou de design do produto, mas sim para testar hipóteses fundamentais de negócios, tendo como principal característica iniciar o processo de desenvolvimento, e não o terminar.

Apesar de se tratar de um mínimo produto viável, o MVP pode abranger alguns níveis de complexidade, podendo ser extremamente simples, servindo basicamente como publicidade, ou como protótipos mais completos, contando com diversas funcionalidades implementadas. Qual exatamente é a complexidade ideal irá variar de produto para produto e do julgamento do idealizador (RIES, 2011, tradução nossa).

Uma das principais lições deixadas pelo MVP é a de que, na dúvida, simplifique. Não é incomum encontrar times de desenvolvimento superestimando a quantidade de recursos necessários para a construção do MVP. Por isso, qualquer trabalho adicional, além do necessário para começar o aprendizado, mesmo que visto como importante, pode ser considerado desperdício de tempo e recursos (RIES, 2011, tradução nossa).

GERENCIAMENTO ÁGIL E MODELAGEM DE PROCESSOS

Métodos ágeis são construídos a partir dos 12 princípios do manifesto ágil, estes complementam os valores ágeis e tornam-se pilares necessários para elaboração de qualquer método ágil (GOMES; WILLI; REHEM, 2014).

Segundo Gomes, Willi e Rehem (2014):

1. A principal prioridade é satisfazer ao cliente com entregas contínua e adiantada de software com valor agregado. Este princípio ressalta que o maior objetivo é a realização de entregas de softwares com qualidade, com iterações rápidas e contínuas, devendo agregar valor ao negócio do cliente.

2. Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Os processos ágeis tiram vantagem das mudanças, visando à vantagem competitiva para o cliente. Aqui o cliente pode mudar uma decisão tomada no início do projeto que talvez tenha sido precipitada, beneficiando-se ao se adaptar a novos cenários de mercado e assim tornando-se mais competitivo.

3. Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo. A entrega contínua de um software funcional, capaz de responder rapidamente a mudanças e agregue valor ao negócio do cliente, somente é possível por conta de ciclos curtos que possibilitam também a análise de alguns fatores como, por exemplo, a velocidade de entrega do time, auxiliando no processo de melhoria contínua do projeto.

4. Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto. Deve haver colaboração entre a equipe de desenvolvimento e

o cliente, trabalhando prioritariamente juntos no mesmo ambiente, isto possibilitará discussões, um fluxo contínuo de apresentação e feedbacks.

5. Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessários e confie neles para realizar o trabalho. O ambiente deve ser de comunicação direta e constante, com feedbacks frequentes e o comprometimento em relação ao projeto deve ser de todos os envolvidos nele, com cada indivíduo realizando seu papel.

6. O método mais eficiente e eficaz de transmitir informação para a equipe e entre a equipe de desenvolvimento é a conversa frente a frente. Este princípio diz que conversas frente a frente são mais eficazes do que qualquer meio de comunicação indireta, quanto menos comunicação indireta houver menores serão os riscos de má interpretação.

7. Software funcional é a medida primária de progresso. Primordialmente, o bom andamento do desenvolvimento de um software deve ser mensurado através da quantidade de software funcional entregue e não pelo volume de documentação.

8. Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem conseguir manter um ritmo constante sempre. Ambientes que funcionem sem atingir seus limites operacionais, mas sim em níveis nos quais sua sustentação seja viável por ilimitados períodos devem ser criados e mantidos, tornando-se também um ambiente mais produtivo.

9. A contínua atenção à excelência técnica e bom design aumenta a agilidade. O código construído deve ser bem feito e o projeto ter qualidade, pois, desse modo, a necessidade de uma documentação cansativa é eliminada, a possibilidade de retrabalho é reduzida e a tomada de decisões rápidas é facilitada.

10. Simplicidade, a arte de maximizar a quantidade de trabalho não realizado, é essencial. Um projeto simples facilita o processo de manutenção, garantindo para o time mais tempo e energia para aprimoramento ou para implementação de alguma nova funcionalidade que venha agregar valor ao negócio.

11. As melhores arquiteturas, requisitos e design emergem de times auto-organizáveis. O processo de desenvolvimento de software é muito complexo, tornando qualquer tentativa de sistematizá-lo falha, logo não é viável investir muito tempo para detalhá-lo, pois a probabilidade de mudança do plano original é grande.

12. Em intervalos regulares, o time reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo. Este princípio defende que cada time deve encontrar por mérito próprio sua dinâmica de trabalho para atuar em um projeto. Desse

modo, a cada ciclo de trabalho deve ser identificado o que não está funcionando e potencializar aquilo que estiver dando certo, contribuindo assim para a melhoria contínua do projeto.

O gerenciamento ágil de projetos é o conjunto de ações que conduzem os trabalhos para complementar o objetivo de um projeto, de forma que todas as atividades necessárias para alcançar o objetivo definido concordem com os valores e os princípios ágeis (CRUZ, 2016).

Em uma gestão ágil é responsabilidade do time a condução dos trabalhos do projeto em direção ao objetivo principal, desse modo cada membro tem um papel específico para exercer, o que contribui para um gerenciamento distribuído. De modo geral, a gestão ágil favorece especialmente o relacionamento entre pessoas e a colaboração com o cliente, contribuindo para construção de serviços e produtos de qualidade, respondendo mais rapidamente a mudanças (CRUZ, 2016).

Existem vários frameworks e metodologias que são exemplos da gestão ágil de projetos e processos. O Scrum é um framework para gerenciamento de projetos ágeis, podendo ser utilizado não somente não para o desenvolvimento de software como também para gerenciamento, planejamento e desenvolvimento de qualquer produto, isso é possível trata-se de uma ferramenta iterativa e incremental. O controle de processos a partir da experiência e aplicação é a ideia principal do Scrum, para gerar conhecimento adquirido por percepções e pela relação de causa e efeito, além de manter o foco na entrega de valor ao negócio no menor prazo possível, sendo assim no Scrum os projetos são divididos em ciclos repetitivos e curtos que possibilitam uma correção o mais rápida possível em pedaços do produto que sofram algum desvio, estes ciclos repetitivos recebem o nome de “Sprints” podem ter uma duração que varia entre duas ou quatro semanas (CRUZ, 2016).

Segundo Cruz (2016) o Scrum possui três pilares de sustentação sendo, o primeiro pilar a transparência, este garante que qualquer aspecto do projeto que afete o resultado seja conhecido e visível aos que controlam o resultado, o segundo é a inspeção onde frequentemente os processos devem ser inspecionados visando detectar alguma variação, o terceiro pilar é a adaptação buscando minimizar desvios futuros em caso de alguma variação que torne o produto resultante inaceitável, o processo ou material produzido devem ser ajustados o mais breve possível.

Extreme Programming (XP) é uma metodologia de desenvolvimento de software que visa atender as necessidades do cliente combinando de forma simples, produtividade, rapidez e qualidade. Um dos objetivos é gerar o máximo de valor possível para o cliente

em um curto período, durante este tempo o cliente poderá realizar análises do produto entregue verificando se o mesmo lhe atende de forma satisfatória, isso é possível, pois, o XP é uma metodologia orientada para o desenvolvimento onde os requisitos podem sofrer constantes mudanças (PONTES; ARTHAUD, 2018).

O Lean é uma metodologia de produção focada em aumentar a velocidade de processos e a qualidade do produto final, eliminando desperdícios de recursos com objetivos que não seja a geração de valor para o cliente final. Dentre os princípios do Lean, o primeiro diz que qualquer coisa que não traga valor ao cliente deve ser eliminada, pois identificar a necessidade do cliente e entregá-la no menor prazo de tempo possível e com qualidade é essencial. Outro princípio do Lean é a amplificação de aprendizado gerada através da criação de conhecimento resultando na melhora do ambiente de desenvolvimento, pois, o conhecimento adquirido pelo desenvolvedor contribuirá para software cada vez melhor (PONTES; ARTHAUD, 2018).

O Kanban é baseado na ideia de que o total de atividades em andamento (WIP) deve ser limitado e uma atividade somente deve ser iniciada caso de fato seja possível iniciá-la, desse modo é necessário que uma atividade em andamento seja finalizada ou que o limite definido possibilita iniciar uma nova atividade, o acompanhamento do fluxo dessas atividades é realizado através de um mecanismo visual (PONTES; ARTHAUD, 2018).

Desenvolvida como uma tecnologia, a modelagem de processo é utilizada para descrever processos de modo que possam ser entendidos e desenvolvidos com maior visibilidade organizacional. Existem diversos métodos e anotações que podem ser utilizados para descrever um processo de forma mais detalhada, estes variando entre notações formais rigorosas, até notações gráficas, as duas melhores técnicas utilizadas para modelagem de processos em relação a notações gráficas e formais, são as nomeadas de BPMN e SPEM (BEZERRA, 2009).

VIABILIDADE TÉCNICA

Determinar a viabilidade técnica de um produto é um passo essencial para um bom planejamento e funcionamento do projeto, calcular os esforços que serão gastos em cada etapa do processo e os recursos necessários para se alcançar a versão final do produto desejado pode ser o diferencial para chegar em uma abordagem bem sucedida. Esses são conceitos que podemos observar na Primeira Lei da Termodinâmica e na Lei de Conservação das Massas, e são visões consideradas e utilizadas durante uma análise de viabilidade técnica (PROPEQ, 2020).

A análise de viabilidade é uma das etapas mais importantes no desenvolvimento de projetos, podendo ser utilizada tanto na análise de produtos que já estão no mercado, quanto no estudo de uma nova proposta de empreendimento (VALE, 2020).

Trata-se de um estudo técnico baseado principalmente em cálculos, estimativas e estatísticas, buscando evitar riscos e reduzir gastos desnecessários, podendo servir de base para a implantação de diversos processos produtivos (PROPEQ, 2020).

Para se dar início a uma análise de viabilidade, é necessário antes compreender muito bem o mercado onde o seu produto estará inserido, tendo ciência do que você pode ou não oferecer, e o que os seus concorrentes já estão oferecendo. Com essa informação em mãos, e tendo uma boa ideia do volume de mercado, potencial de clientes e o preço que será cobrado pelo produto final, já é possível começar o processo de projeção de mercado, verificando o potencial de captação de dinheiro do seu futuro negócio. Além disso, também é importante ter uma boa noção dos custos que o seu produto irá gerar, e qual a estimativa de fluxo de caixa que ele estará gerando.

Outra etapa importante é a de análise de indicadores, pois é através dela que você poderá descobrir o tempo necessário para se recuperar o investimento feito, além do quão efetivo o seu negócio pode ser (VALE, 2020).

Todas essas etapas citadas podem ser consideradas técnicas que ajudam o projeto a obter sucesso, buscando entregar um resultado que seja satisfatório e que possa ser adaptado às exigências dos clientes, enquanto evita falhas e problemas durante o planejamento. Por conta disso, são processos que devem ser utilizados durante a gestão de projetos, auxiliando na estruturação do projeto, em seu planejamento e em como será realizada a execução e monitoramento (CARVALHO, 2020).

RESULTADOS: EXECUÇÃO E DESENVOLVIMENTO DO MVP SEGUNDO A METODOLOGIA LEAN STARTUP

A seguir estão expressos os resultados do emprego da metodologia utilizada para execução do desenvolvimento do MVP.

FASE 1: IDEIAS - REQUISITOS

Seguindo o fluxo do Ciclo do Lean Startup, iniciamos pela fase de ideias onde foi decidido a temática do jogo e suas mecânicas, durante essa fase os requisitos funcionais, que compreendem as necessidades que devem ser atendidas pelo software por meio de funções ou serviços, foram levantados e estão listados no Quadro 1:

Quadro 1 – Lista dos Requisitos Funcionais

RF001	O Jogador deve possuir controle do personagem principal.
RF002	Deve existir um personagem secundário para fornecer dicas ao jogador.
RF003	Deve existir blocos contendo palavras.
RF004	O jogador deve conseguir empurrar os blocos de palavras com o personagem.
RF005	O jogador deve ter a capacidade de influenciar no ambiente do jogo através da lógica aplicada pelos blocos de palavras.
RF006	O jogador deve possuir um objetivo a ser alcançado.
RF007	Ao conquistar o objetivo uma tela contendo a lógica de programação em pseudo-código deve ser apresentada ao jogador.
RF008	O jogo deve ser disponibilizado para navegadores web.

Fonte: Próprio autor

E a seguir, no quadro 2, estão os requisitos não funcionais que compreendem todos aqueles requisitos que definem o como o sistema fará, não estão relacionados diretamente às funcionalidades de um sistema.

Quadro 2 – Lista dos Requisitos Não Funcionais

RNF001	O jogo deve possuir visão de cima para baixo com câmera fixa.
RNF002	O jogo deve possuir gráficos simples em 2D.
RNF003	O jogo deve ser desenvolvido utilizando o motor *GameMaker*.
RNF004	O jogo deve ser disponibilizado para navegadores web.
RNF005	O jogo deve possuir requisitos mínimos de hardware para ser jogado.
RNF006	O jogo deve oferecer liberdade para o jogador alcançar o objetivo principal.

Fonte: Próprio autor.

O quadro 3 apresenta um dos requisitos funcionais especificados. A especificação garante que os requisitos apresentarão todas as características principais para a delimitação de escopo e configuração das funcionalidades do jogo.

Quadro 3 - Requisitos Funcionais Especificados.

Identificador	**RF005**		
Nome	O jogador deve ter a capacidade de influenciar no ambiente do jogo através da lógica aplicada pelos blocos de palavras.		
Data de criação	18/07/2022	**Autor**	Leandro Alves
Data da última alteração	18/07/2022	**Autor**	Leandro Alves
Versão	1.0	**Prioridade**	Essencial
Descrição	• O jogador não poderá atravessar as paredes que cercam a bandeira enquanto os blocos estejam alinhados e formando a afirmação “parede te para". • O jogador tem total liberdade para desfazer essa afirmação e conseguir atravessar as paredes.		

Fonte: Próprio autor.

E o quadro 4 apresenta um exemplo de requisito não funcional especificado.

Quadro 4 - Requisitos Não Funcionais Especificados.

Identificador	**RNF001**		
Nome	O jogo deve possuir visão de cima para baixo com câmera fixa.		
Data de criação	16/07/2022	**Autor**	João Victor Lopo
Data da última alteração	16/07/2022	**Autor**	João Victor Lopo
Versão	1.0	**Prioridade**	Essencial
Descrição	• O jogador deve conseguir visualizar seu personagem e de todos os elementos pertencentes à fase a partir de uma vista de cima para baixo e fixa.		

Fonte: Próprio autor.

Após o levantamento e especificação dos requisitos iniciou-se a etapa 1 do ciclo do Lean Startup, a etapa de construção do MVP.

ETAPA 1: CONSTRUIR - TECNOLOGIAS UTILIZADAS

As tecnologias utilizadas na execução do MVP constituem a segunda fase do ciclo do Lean, a de construção do MVP.

GAMEMAKER STUDIO 2

Segundo Cossu (2019, tradução nossa), o *GameMaker Studio* 2 é um motor de jogos completo e fácil de usar, adotado por muitas empresas pequenas para criar alguns dos jogos 2D e 3D mais vendidos dos últimos anos. Por suportar o desenvolvimento de jogos com codificação *Drag and Drop* ou com a linguagem GML, é uma ferramenta perfeita para iniciantes e profissionais.

Drag and Drop é um sistema que permite estruturar algoritmos arrastando e soltando blocos que representam pedaços de código.

Já o *GML Code* é a linguagem de programação proprietária do *GameMaker*. É muito fácil de usar e aprender, e tem tudo o que você pode precisar para desenvolver a lógica do seu jogo.

O *GameMaker Studio* 2 consegue proporcionar todas as ferramentas necessárias para seguir no processo de desenvolvimento, dando acesso direto a um navegador de arquivos para gerenciar seus recursos, um editor de texto para escrever seu código GML, um editor gráfico 2D para criar, editar e animar a parte visual do jogo, intérprete para rodar e depurar seus jogos usando a Máquina virtual *GameMaker*, e um compilador para exportar o projeto final.

GML CODE

GML é uma linguagem de script baseada em C++ e JavaScript, criada pelo *GameMaker* especificamente para o *GameMaker Studio* 2, permitindo que você desenvolva toda a lógica por trás do seu jogo.

Apesar de bem semelhante com outras ferramentas do mercado, o GML permite que você crie jogos completos do zero com muito pouco esforço, da mesma forma que outras linguagens mais complexas usadas na indústria de jogos. Além disso, por possuir uma documentação completa e um grande acervo de tutoriais, é uma ótima opção para programadores iniciantes (COSSU, 2019, tradução nossa).

GIT

Git, uma ideia de Linus Torvalds, teve seu início em 2005, como um sistema de

gerenciamento de revisões, usado para coordenar o desenvolvimento do kernel Linux.

Ao longo dos anos, sua funcionalidade, portabilidade, eficiência e adoção de terceiros evoluíram, tornando-o líder de sua categoria.

O seu foco estava em identificar, controlar e divulgar as alterações e melhorias realizadas no código de um software, evitando, principalmente, que conflitos ocorram durante o desenvolvimento de software, em outras palavras, fornecendo um valioso controle (SPINELLIS, 2012, tradução nossa).

GITHUB

O GitHub simplifica muitas tarefas de gerenciamento de repositório por meio de uma interface de usuário baseada na Web. Além disso, promove a cooperação em projetos de código aberto, hospedados gratuitamente, facilitando para os desenvolvedores clonar projetos existentes e enviar suas contribuições como um *pull request,* fornecendo funcionalidades como a de configurar um repositório público, manter seus servidores e conectividade, mantê-lo seguro e atualizado, configurar contas de usuário e oferecer suporte a seus usuários. O GitHub também fornece um sistema de rastreamento de problemas, uma área de download e Gollum, um wiki baseado em git. Por meio do Gollum, você pode editar uma página na Web e registrar a alteração como um git *commit*, mas também pode realizar alterações manuais ou automatizadas nos arquivos de um clone wiki local e depois enviá-los para um repositório (SPINELLIS, 2012, tradução nossa).

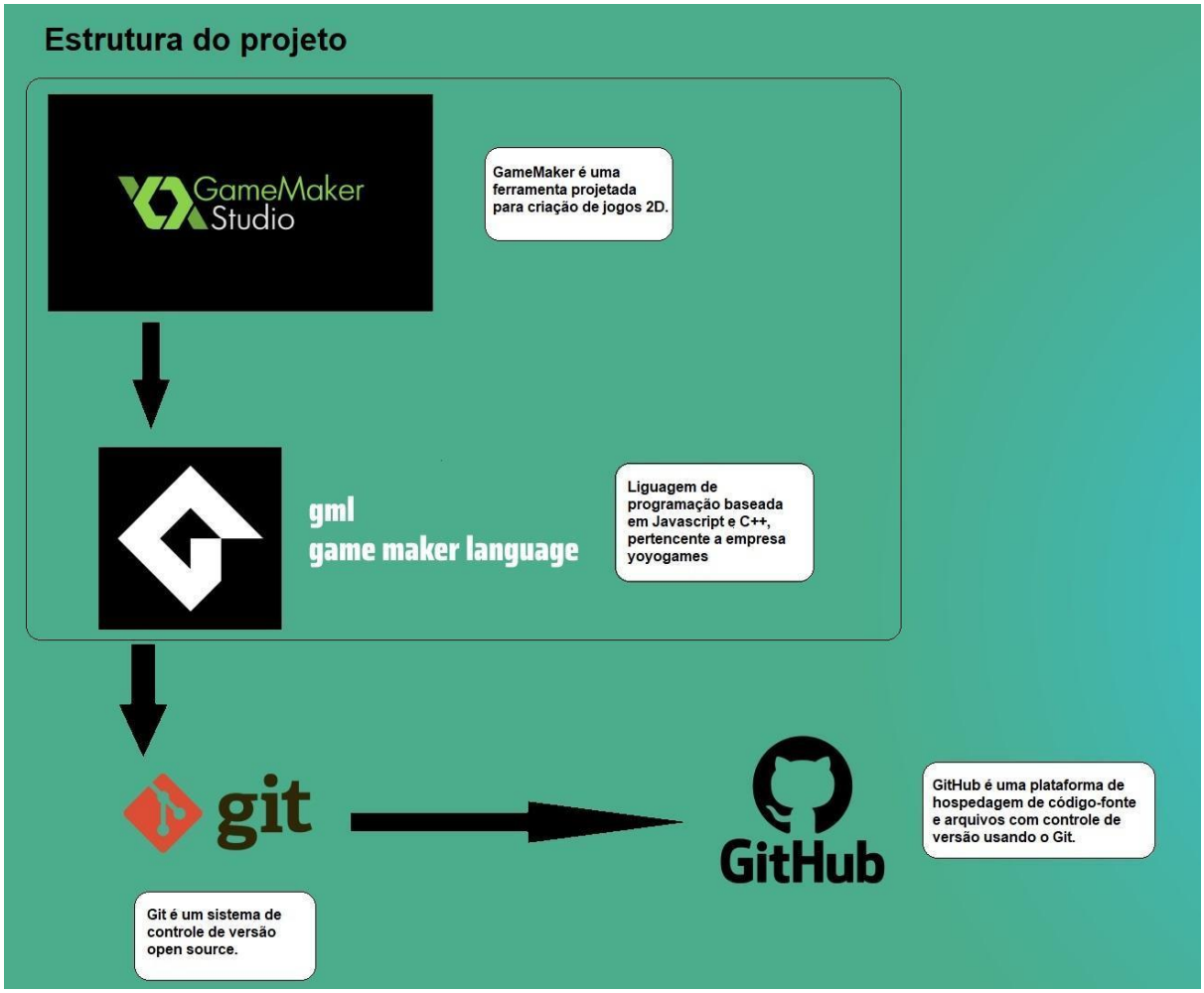

Figura 1 - Integração das tecnologias utilizadas
Fonte: próprio autor

A figura 1 apresenta a integração das tecnologias utilizadas para o desenvolvimento do MVP. Durante a etapa de construção a ferramenta chamada *GameMaker Studio* 2 foi escolhida para auxiliar no desenvolvimento do jogo, essa ferramenta é voltada para a construção de jogos 2D e fornece um ambiente de desenvolvimento integrado, onde foi realizado todo o trabalho de codificação através da utilização da linguagem de programação *GameMaker Language*, momento ao qual foi construído a interface e funcionamento do jogo, onde pelo controle de persistência ser feito pela própria *engine* (*GameMaker Studio* 2), não foi necessário o desenvolvimento e implementação de um *backend* ou banco de dados. Novas versões foram geradas a cada evolução ocorrida no desenvolvimento do jogo e a ferramenta escolhida para o controle destas foi o Git, e o GitHub foi a tecnologia utilizada para geração de um repositório remoto, ao qual recebe e armazena todas as versões do jogo.

Na fase de produto foi obtido a primeira versão do MVP e durante a etapa de medição a primeira versão do MVP foi disponibilizado para profissionais da área da

educação para que pudessem jogar e avaliar a importância que jogo tem no auxílio do ensino de lógica de programação para crianças, a opinião desses educadores foi analisada com base em um formulário enviado a eles.

FASE 2: PRODUTO – DIAGRAMAS

CASO DE USO

Diagramas de caso de uso são utilizados para descrever funções de alto nível e o escopo de um sistema, identificando as interações entre o sistema e seus atores. O relacionamento entre os atores e os casos de uso em um diagrama de caso de uso descrevem o que o sistema faz e como os atores o utilizam, não abordando como o sistema opera internamente (IBM, 2021).

A figura abaixo remete ao diagrama de caso de uso do MVP do jogo educacional para auxiliar no ensino de lógica de programação para crianças:

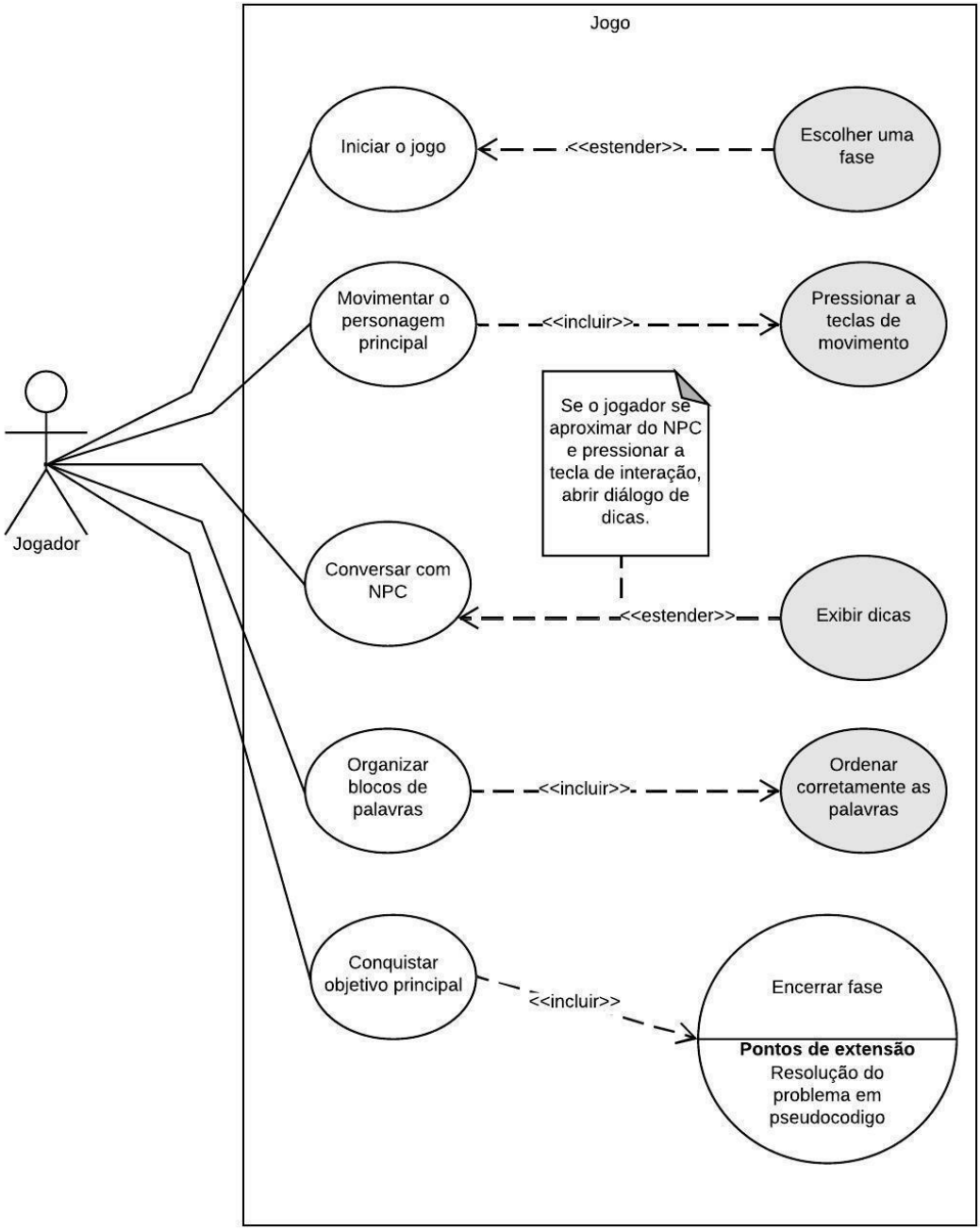

Figura 2 - Diagrama de caso de uso

Fonte: próprio autor

De acordo com a figura 2, o sistema representado pelo retângulo, é jogo desenvolvido, o ator que irá interagir com o sistema é representado pelo boneco palito, sendo este o jogador, os casos de uso estão representados pelas formas ovais contendo a descrição da ação a ser realizada, os casos de uso estão organizados em uma ordem lógica, sendo a primeira ação associada ao jogador estando localizada no topo e as demais em sequência respectivamente.

Ao iniciar o jogo, o jogador poderá decidir entre escolher uma fase específica ou apenas iniciar na fase 1, após a fase escolhida ser iniciada o jogador poderá movimentar seu personagem pelo ambiente e para isso é necessário que ele pressione as teclas de movimento para indicar a qual direção deseja se deslocar, dentro de cada fase haverá um personagem secundário em que caso o personagem principal se aproxime o suficiente e pressione a tecla de interação uma caixa de diálogo deverá surgir, com dicas contendo conceitos básicos de programação de como solucionar o problema atual para alcançar o objetivo principal, blocos contendo palavras estarão distribuídos em cada fase, o jogador deverá organizá-los e devem estar em uma ordem que formule uma frase corretamente, deste modo eliminando os obstáculos que impedem a conquista do objetivo principal que é capturar a bandeira, após este ao ser alcançado a fase é finalizada e uma tela contendo pseudocódigo elucidando o problema e sua resolução através de lógica de programação.

UML

UML é um padrão para a análise e design de software. Um dos componentes mais importantes da UML são os diagramas de classes, que modelam as informações sobre o domínio de interesse em termos de objetos organizados em classes e relacionamentos entre eles (BERARDI, CALVANESE, DE GIACOMO, 2005, tradução nossa).

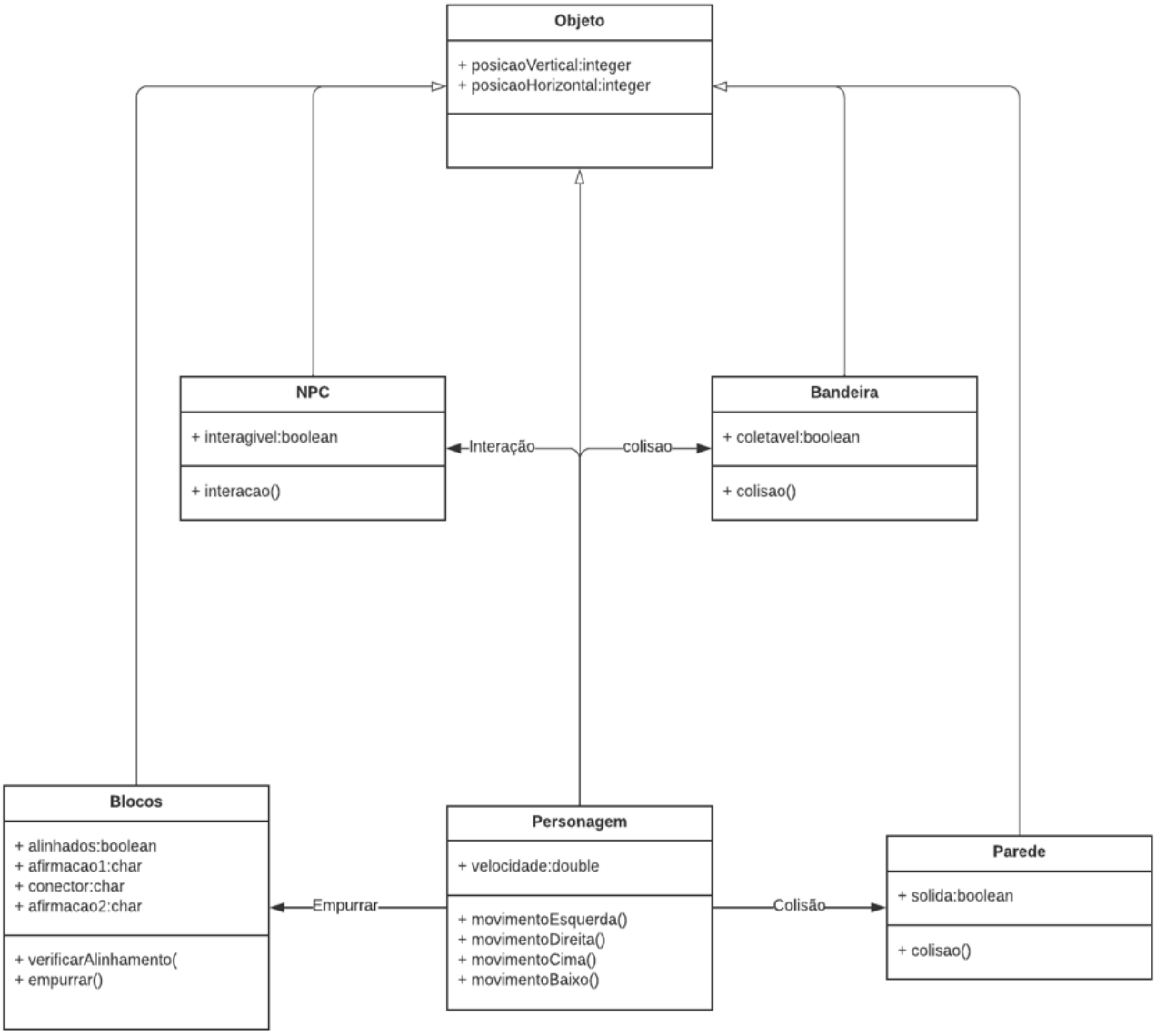

Figura 3 - Diagrama de classes do MVP
Fonte: próprio autor

Neste contexto, todas as entidades herdam da classe objeto, cada uma possuindo uma posição horizontal e vertical, definidos por X e Y em um plano cartesiano.

O personagem controlado pelo jogador possui uma variável de controle de velocidade, que será alterado conforme este tiver seu movimento interrompido por algum obstáculo. Além disso, possui métodos responsáveis por validar seu movimento quando o jogador pressiona as teclas correspondentes.

Personagem possui um evento de “colisão” com a entidade parede, onde será validado se o personagem deve ou não ser bloqueado pela parede.

Possui também um evento de “empurrar” com a entidade blocos, podendo movê-los e organizá-los conforme preferir, porém, apenas uma posição específica irá alterar o estado do atributo “alinhados”.

Os blocos sempre serão constituídos por uma afirmação referente a uma entidade, um verbo conector e uma segunda afirmação que se refere a um estado.

Personagem também possui um evento de interação com NPC, onde ao se aproximar o suficiente será validado se o jogador pressionou uma tecla pré-estabelecida, responsável por dar início a interação.

Por fim, é possível também iniciar um evento de colisão com a entidade bandeira, responsável por simbolizar o objetivo final do jogador.

ETAPA 2: MEDIR - MVP

A figura 4 a seguir demonstra todo o cenário da primeira fase do jogo, onde podemos observar o personagem principal sendo o coelho na cor branca, um personagem secundário sendo o gato na cor violeta responsável por fornecer dicas ao jogador, o conjunto de blocos alinhados formando a afirmação “Parede te para” e o objetivo final sendo este a bandeira:

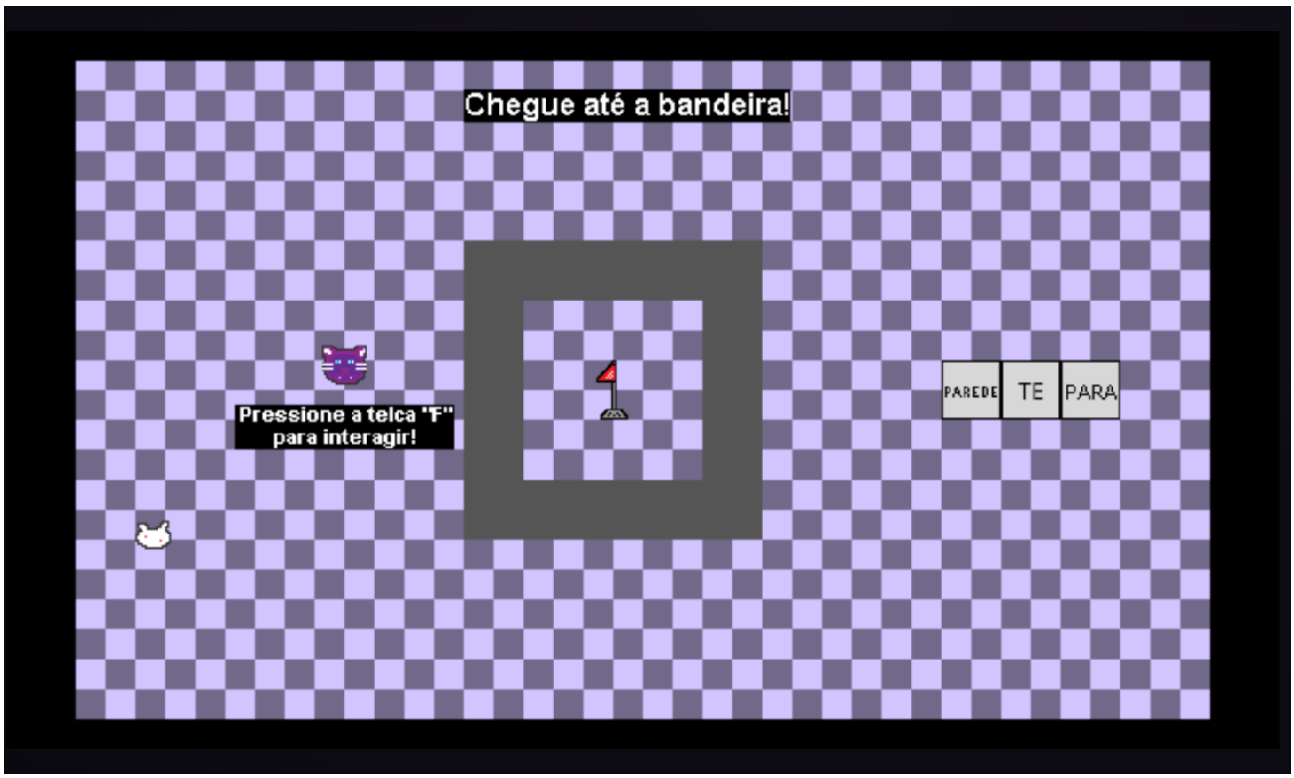

Figura 4 - Tela inicial

Fonte: próprio autor

Os blocos estão alinhados formando a afirmação “Parede te para”, deste modo a parede deve bloquear o caminho do jogador até o objetivo final:

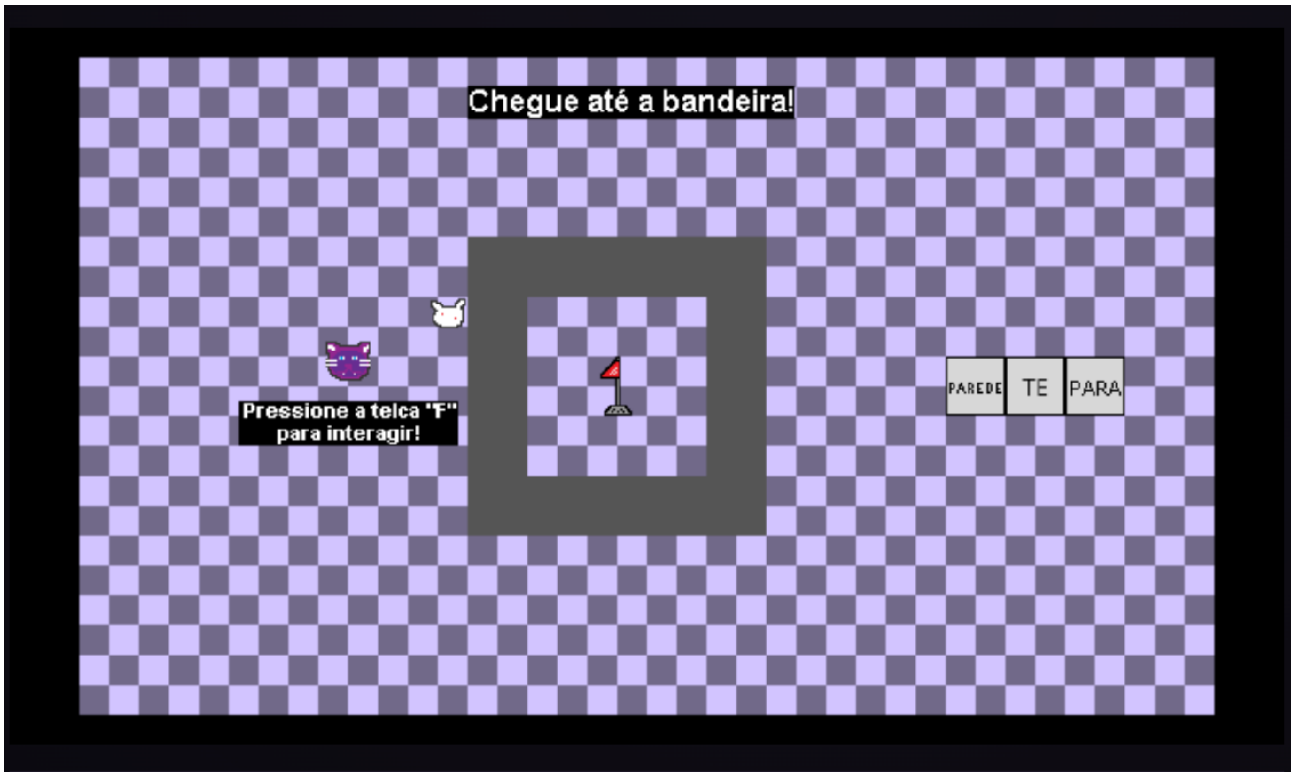

Figura 5 - Tela bloqueio da parede
Fonte: próprio autor

O jogador poderá interagir com o personagem secundário se aproximando deste e clicando na tecla F, ao realizar essa ação uma caixa de diálogo entre os personagens é aberta oferecendo dicas ao jogador. Como mostrado nas figuras 5 e 6:

Figura 6 - Tela início do diálogo

Fonte: próprio autor

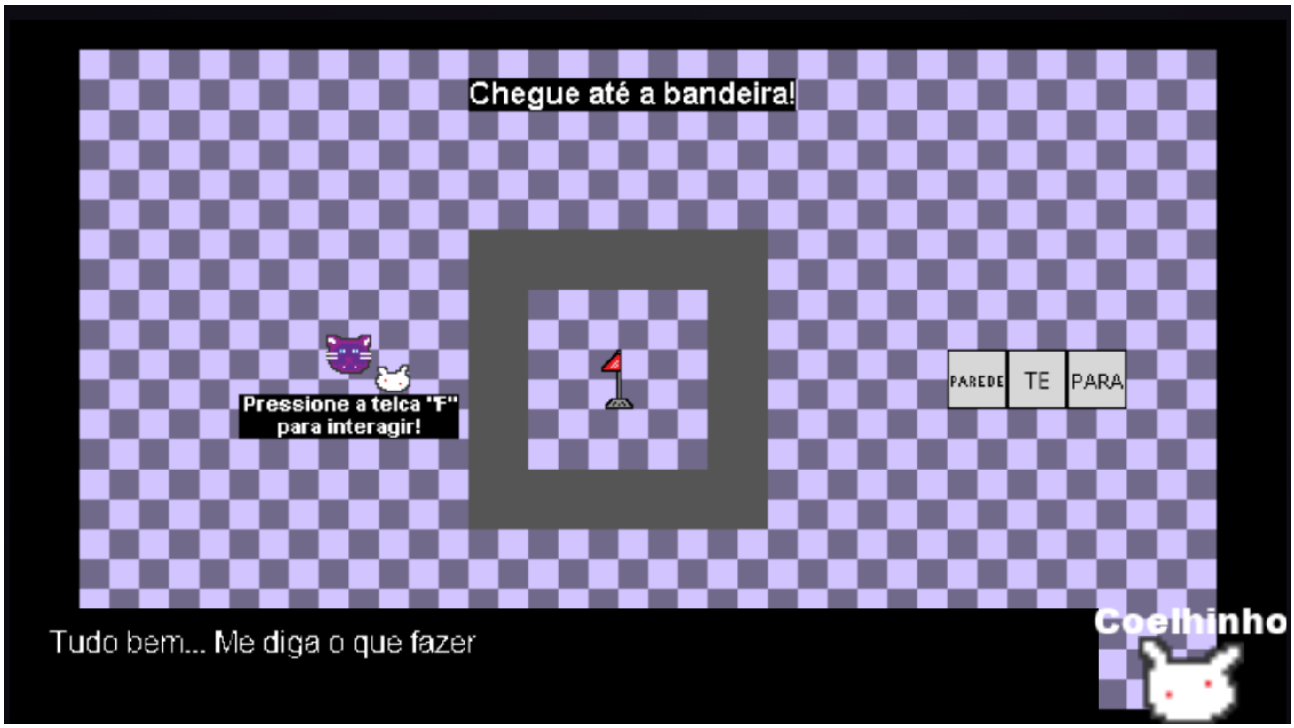

Figura 7 - Tela fim do diálogo

Fonte: próprio autor

Ao desfazer a afirmação “Parede te para” a parede deixa de bloquear a passagem do jogar até o objetivo final, deste modo o jogador estará livre para conquistar o objetivo. Conforme a figura 8:

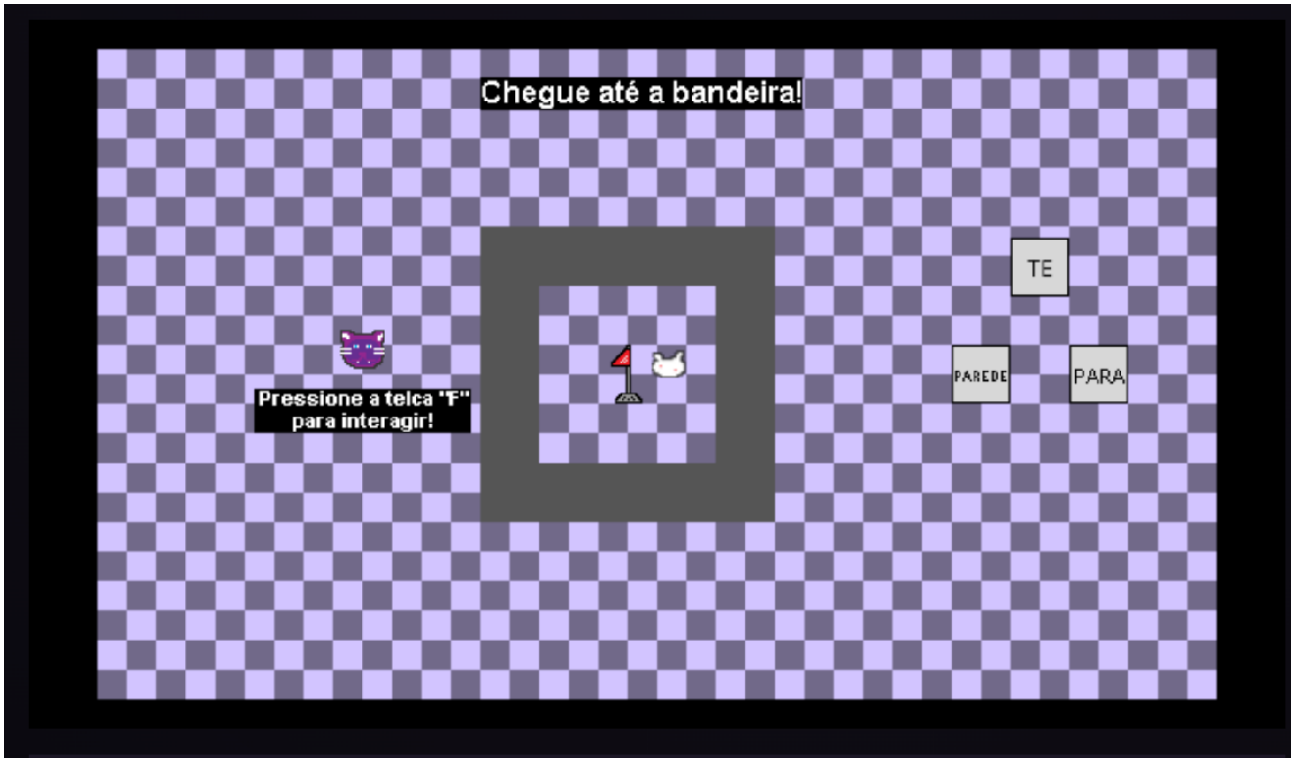

Figura 8 - Tela movimentando os blocos e desfazendo a afirmação
Fonte: próprio autor

Ao conquistar o objetivo final o jogador precisa tocar na bandeira para ser redirecionado para uma tela contendo a exemplificação do cenário da fase em pseudocódigo:

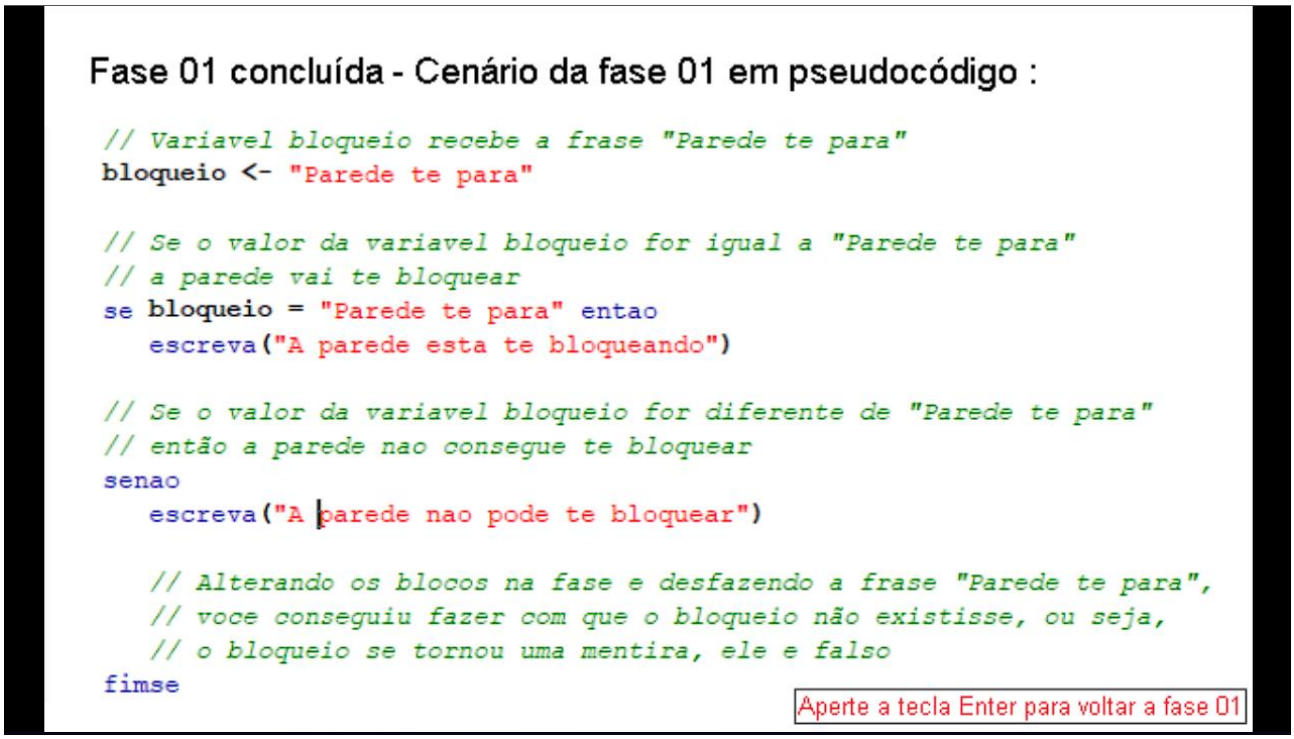

Figura 9 - Tela cenário da fase 01 em pseudocódigo
Fonte: próprio autor

Ao concluir a fase e atingir o objetivo, o jogador visualiza a figura 9, contendo o pseudocódigo e a lógica envolvida na execução da conclusão do objetivo que terminou de alcançar. Caso o jogador pressione a tecla "*Enter*", será redirecionado para o início da fase 01 novamente.

A literatura que discute o emprego de jogos educacionais para o ensino da lógica de programação, explica que em experimentos realizados entre dois grupos de crianças, onde o grupo 1 utilizou jogos educativos para auxiliar no processo de aprendizagem enquanto o grupo dois não, os resultados obtidos mostraram que os alunos do grupo 1 consideraram que a abordagem baseada em jogos educacionais era mais atraente e educacionalmente mais eficaz, pois conseguiram adquirir mais conhecimentos que os alunos do grupo 2, alguns dos motivos que contribuíram para este resultado estão relacionados a um maior engajamento que os jogadores tiveram durante o processo de aprendizagem e por consequência absorvendo mais conteúdo se comparado abordagem métodos tradicionais de ensino (GIRARD; ECALLE; MAGNAN, 2013, tradução nossa).

As crianças que experimentaram o *AutoThinking* (um jogo adaptativo para ensinar conceitos e habilidades de pensamento computacional) relataram três principais emoções. A primeira sendo a de se sentirem desafiadas, o que poderia estar relacionado a como elas precisaram se adaptar de modo a superar os desafios proporcionados pelo jogo. Para cada desafio superado, elas se sentiam cada vez mais confiantes, o que gerava um sentimento de felicidade. As crianças também relataram sentir um ganho de aprendizado e raciocínio durante o jogo, alegando terem gostado da dificuldade e dos desafios cognitivos que o jogo proporcionou e aquelas que já tinham alguma experiência com desenvolvimento conseguiram fazer conexões entre o jogo e programação (MALVA et al., 2020, tradução nossa).

Outra pesquisa realizada com 49 alunos durante as aulas de informática de uma escola particular, onde o objetivo foi verificar as principais dificuldades enfrentadas pelas crianças ao interagir com os jogos que envolvam lógica de programação, apresentou resultados positivos, como a compreensão satisfatória de conceitos como sequências e condicionais. As crianças apresentaram dificuldade para encarar sequências longas de instruções e para realizar combinações destas, a princípio por não conseguirem memorizar 5 ou mais instruções, os jogos utilizados na pesquisa não fornecem feedback visual que auxiliasse as crianças a identificar sua evolução durante a execução da

sequência de instruções necessárias para solucionar o problema (GOMES; TEDESCO, 2017).

Desse modo, acreditamos que o MVP obtido possa auxiliar crianças no ensino de lógica de programação através da sua abordagem simples, da possibilidade do fornecimento de dicas para que o jogador conquiste o objetivo, e da exemplificação da resolução do desafio por meio do pseudocódigo exibido após o término da fase. Um exemplo está no cenário explorado durante a primeira fase, onde é apresentado conceitos de programação com operadores lógicos e estruturas de validação.

FASE 3: DADOS - VALIDAÇÃO

O MVP foi disponibilizado para que alguns profissionais da área da educação realizassem testes de modo a avaliarem a eficiência e valor do produto referente a sua capacidade de auxiliar no ensino de lógica de programação para crianças. O feedback recebido foi analisado e inserido em tabelas, elencando os pontos considerados positivos e negativos.

Quadro 5 - Aspectos positivos e negativos levantados na validação

VALIDAÇÃO	**Pontos Positivos**	**Pontos negativos**
5 Professores da rede pública de ensino que trabalham na educação básica e ensino médio testaram o jogo e relataram os seguintes pontos positivos e negativos	- Abordagem de ensino não presa ao aspecto teórico, partindo da assimilação prática; - Capacidade de instigar a criança a explorar o cenário, gerando insights no decorrer do jogo; - Introdução de conceitos de lógica de programação e incentivo desde cedo para que crianças entendam sobre lógica de programação - Possibilidade de o aluno aprender a programar a partir da apresentação de pequenos trechos de programas	- Falta de uma interface mais atrativa e dinâmica; - Desenvolvimento de mais fases - Detalhamento de uma prévia sobre lógica de programação (conceitual e teórica)

Fonte: próprio autor

Foi possível observar que os educadores que validaram o jogo reforçam a contribuição cognitiva e desenvolvimento do raciocínio que a lógica de programação pode

proporcionar às crianças. Tanto a literatura quanto a validação evidenciam que o emprego da lógica de programação no ensino básico pode facilitar o processo de análise dos enunciados, diminuindo as dificuldades de interpretação.

Além disso, é possível identificar que os pontos positivos reforçam que a aprendizagem de estratégias de programação e retenção de dicas adquiridas com base nas estratégias aprendidas, podem instigar e proporcionar ao aluno o desenvolvimento de habilidades referentes à programação. Não é esperado que a criança aprenda a programar, mas a apresentação de pequenos trechos de programas, sem a exigência de implementação, pode aguçar a apreensão do raciocínio lógico nas fases iniciais de desenvolvimento da criança.

Os padrões empregados na fase do jogo e o fato de ser um MVP ocasionaram fragilidades apontadas pelos educadores na validação. Uma das barreiras encontradas está na interface, uma vez que não foi contemplada usabilidade e interface neste trabalho, o foco se deu na execução da fase. Os educadores ressaltaram ainda a ausência de mais fases e de um conteúdo que pudesse introduzir conceitos de lógica de programação no próprio jogo, além do expresso ao final da fase do jogo.

ETAPA 3: APRENDER - MELHORIA

O feedback gerado pelos profissionais da educação que testaram o MVP é fundamental e resulta em pontos de melhoria no jogo, pois esses serão analisados, em sequência, implementados e, como consequência, tornando o produto mais eficiente.

CONSIDERAÇÕES FINAIS

No início do trabalho foi constatado que o desenvolvimento do pensamento computacional em uma pessoa colabora para o bem-estar em diversos aspectos de sua vida, e que é importante iniciar esse processo na infância, visto que essa é uma habilidade fundamental que contribui na capacidade analítica e de planejamento do indivíduo. No decorrer da análise de viabilidade técnica para o desenvolvimento de um MVP de jogo educacional para auxiliar crianças no ensino de lógica de programação foi verificado os benefícios que os jogos educacionais oferecem para o desenvolvimento do pensamento computacional.

Tanto os objetivos específicos, quanto o objetivo geral do trabalho de pesquisa, foram alcançados. Inicialmente foram analisados artigos relevantes cujo tema estivesse associado a importância da lógica de programação e os benefícios que os jogos educacionais podem oferecer na aprendizagem das crianças, tendo como referência esse

conhecimento, os módulos do MVP foram projetados e em sequência construídos junto implementação do Lean Startup em cada fase do desenvolvimento. Por fim, realizou-se a disponibilização do produto para realização de testes por profissionais da educação e para auxiliá-los foi criado um roteiro explicando o funcionamento do jogo.

Através do uso combinado das tecnologias *GameMaker Studio* 2, *GameMaker* Language, Git e GitHub, em conjunto da implantação da metodologia Lean Startup, onde por meio do ciclo Construir-Medir-Aprender foi possível abstrair insumos para conclusão de todas as etapas, possibilitando a construção do MVP de maneira ágil, além de gerar também um fluxo eficiente de aprendizagem sobre possíveis melhorias no produto e seu valor para o mercado. É possível identificar que os pontos positivos identificados na validação da ferramenta, reforçam que a aprendizagem de estratégias de programação e retenção de dicas adquiridas com base nas estratégias aprendidas, podem instigar e proporcionar ao aluno o desenvolvimento de habilidades referentes à programação. Dessa forma, foi constatada a viabilidade técnica da construção de um MVP de jogo educacional capaz de auxiliar no ensino de lógica de programação para crianças.

Foram identificadas dificuldades no decorrer da pesquisa, no sentido de realizar a construção de outras fases no jogo, devido à complexidade envolvida na elaboração da lógica de ensino, e na obtenção do feedback de profissionais da educação, por conta da especificidade do tema em questão.

Para estudos futuros espera-se que seja possível implementar melhorias e funcionalidades ao jogo com base no feedback recebido na fase 3 de validação, tornando a sua interface mais atrativa e dinâmica, além desenvolver novas fases, cada uma abordando um tópico diferente envolvendo lógica de programação, enriquecendo o potencial de ensino do produto.

REFERÊNCIAS

BERARDI, Daniela; CALVANESE, Diego; DE GIACOMO, **Giuseppe. Reasoning on UML class diagrams. Artificial intelligence**

BEZERRA, André Luis Rodovalho. **Modelagem de Processos**. 2009.

Carvalho, Técia. O que é gestão de projetos? Aprenda como ela te auxilia nas atividades do negócio, **Voitto**, 2020. Disponível em: <https://www.voitto.com.br/blog/artigo/gestao-de-projeto>. Acesso em: 20 de abr. de 2022.

CAUCHICK MIGUEL, Paulo Augusto et al. Metodologia de pesquisa em engenharia de produção e gestão de operações. **Rio de Janeiro: Elzevir**, 2010.

Como saber a viabilidade técnica de um projeto?. **Propeq**, 2020. Disponível em: <https://tecnoblog.net/responde/referencia-site-abnt-artigos/>. Acesso em: 20 de abr. de 2022.

COSSU, Sebastiano M. **Game Development With Game Maker Studio 2**. Apress, 2019.

CRUZ, Fábio. **PMO Ágil: Escritório ágil de gerenciamento de projetos**. Brasport, 2016.

Diagramas de Caso de Uso. **IBM**, 2021. Disponível em: <https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=diagrams-use-case>. Acesso em: 11 de out. de 2022.

GIRARD, Coralie; ECALLE, Jean; MAGNAN, Annie. Serious games as new educational tools: how effective are they? A meta-analysis of recent studies. **Journal of computer assisted learning**, v. 29, n. 3, p. 207-219, 2013.

GOMES, Alexandre; WILLI, Renato; REHEM, Serge. O manifesto ágil. **R. Prikladnicki, R. Willi, F. Milani (col.). Métodos ágeis para desenvolvimento de software**, p. 311, 2014.

GOMES, Tancicleide; TEDESCO, Patrícia. **Exploring an approach based on digital games for teaching programming concepts to young children**, 2017.

MALVA, Liina et al. Engaging Estonian primary school children in computational thinking through adaptive educational games: A qualitative study. In: **2020 IEEE 20th international conference on advanced learning technologies (ICALT)**. IEEE, 2020. p. 188-190.

MEDEIROS, Tainá Jesus. **Um framework para criação de jogos voltados para o ensino de lógica de programação**. 2014. Dissertação de Mestrado. Universidade Federal do Rio Grande do Norte.

PONTES, Thiago Bessa; ARTHAUD, Daniel Dias Branco. Metodologias ágeis para o desenvolvimento de softwares. **Ciência E Sustentabilidade**, v. 4, n. 2, p. 173-213, 2018.

Raciocínio lógico: por que desenvolver essa habilidade? **Blog Portal Pós**, 2021. Disponível em: <https://blog.portalpos.com.br/raciocinio-logico-habilidade/>. Acesso em: 1 de jun. de 2022.

REIS, Eric. The lean startup. **New York: Crown Business**, v. 27, p. 2016-2020, 2011.

SPINELLIS, Diomidis. Git. **IEEE software**, v. 29, n. 3, p. 100-101, 2012.

VALE, Sávio. Descubra como fazer a análise de viabilidade: a ferramenta para validação de negócios. **Voitto**, 2020. Disponível em: <https://tecnoblog.net/responde/referencia-site-abnt-artigos/>. Acesso em: 20 de abr. de 2022.

VYGOTSKY, Lev Semenovich. A formação social da mente. brasileira. **São Paulo, Martins**, 1988.

WING, Jeannette M. Computational thinking. **Communications of the ACM**, v. 49, n. 3, p. 33-35, 2006.