

# Proposta de jogo digital para ensino de lógica de programação básica com realidade aumentada

Ian F. Nicácio<sup>1</sup>, Carlos A. S. Junior<sup>1</sup>

<sup>1</sup> Instituto Federal de Minas Gerais - Campus Sabará  
Caixa Postal 34.590-390 - Sabara - MG - Brasil

ianfontesn@gmail.com, carlos.junior@ifmg.edu.br

**Abstract.** *The increasing demand for new teaching methodologies has led to the exploration of innovative technologies to enhance the learning of complex concepts, such as programming logic. This work proposes a prototype of an educational application based on Augmented Reality (AR), aiming to teach fundamental programming concepts such as conditional expressions, variable declarations, and logic circuits. Using the Vuforia platform for image tracking, the prototype allows users to interact with virtual objects overlaid onto the physical world, creating a more immersive and dynamic experience. The application was developed in Unity, providing a user-friendly and functional interface for individuals with varying levels of programming knowledge. This study discusses the implementation of the application, the technologies used, and proposes future improvements for its enhancement.*

**Resumo.** *A crescente demanda por novas metodologias de ensino tem levado à exploração de tecnologias inovadoras para aprimorar a aprendizagem de conceitos complexos, como a lógica de programação. Este trabalho propõe um protótipo de aplicativo educacional baseado em realidade aumentada (RA), com o objetivo de ensinar conceitos fundamentais de programação, como expressões condicionais, declarações de variáveis e circuitos lógicos. Utilizando a plataforma Vuforia para rastreamento de imagens, o protótipo permite que os usuários interajam com objetos virtuais sobrepostos à realidade física, criando uma experiência mais imersiva e dinâmica. O aplicativo foi desenvolvido na Unity, proporcionando uma interface amigável e funcional para usuários de diferentes níveis de conhecimento em programação. Este estudo discute a implementação da aplicação, as tecnologias utilizadas e propõe melhorias futuras para seu aprimoramento.*

**Palavras-chave:** Realidade Aumentada; Educação; Programação.

## 1. Introdução

No cenário atual, o uso de dispositivos informatizados está cada vez mais presente no cotidiano das pessoas. Com a digitalização de serviços essenciais, como operações bancárias, emissão de documentos pessoais e até atendimentos médicos, a dependência da tecnologia se tornou inevitável. Na educação, esse cenário não é diferente, onde a tecnologia tem sido amplamente utilizada para expandir as capacidades de aprendizado dos alunos. De acordo com [Martins and Guimarães 2012], a redução dos custos de aquisição

de computadores e o surgimento de novas ferramentas digitais têm contribuído para a maior inserção desses recursos nas escolas.

Apesar dos avanços tecnológicos que transformaram o cotidiano e o ambiente educacional, o ensino de lógica de programação ainda enfrenta desafios significativos. Esta habilidade, cada vez mais essencial para os estudantes no contexto atual, onde a tecnologia está presente desde os primeiros anos de vida, nem sempre recebe a devida atenção. No Brasil, o ensino de computação, embora tenha avançado em algumas áreas, ainda está em estágios iniciais em muitas instituições de ensino. Frequentemente, as aulas de computação são erroneamente associadas ao ensino de informática básica, focando no uso de softwares de escritório, ferramentas de edição gráfica e plataformas de gestão de conteúdo web [Scaico et al. 2012]. Embora seja importante capacitar os alunos no uso dessas ferramentas, esse modelo educativo não promove um entendimento mais profundo dos princípios da programação e do funcionamento dos sistemas computacionais. Mesmo em instituições com laboratórios modernos e bem equipados, o enfoque predominante continua sendo a formação de usuários de tecnologia, em vez de futuros desenvolvedores ou pensadores computacionais. Assim, persiste uma lacuna entre o acesso à tecnologia e o desenvolvimento de competências essenciais para a compreensão e criação de soluções tecnológicas.

Considerando esses desafios, novas tecnologias como a Realidade Aumentada (RA) surgem como soluções promissoras para aprimorar o ensino de lógica de programação. A RA, que integra elementos virtuais ao ambiente físico, tem se mostrado uma ferramenta poderosa em diversas áreas da educação, proporcionando uma experiência imersiva e interativa para os alunos. No ensino de programação, essa tecnologia pode transformar conceitos abstratos em experiências visuais concretas, permitindo que os alunos interajam diretamente com algoritmos e estruturas de dados de maneira visual e prática.

Diferentemente dos métodos tradicionais, onde o ensino de programação é frequentemente teórico, a RA possibilita que os alunos manipulem e visualizem blocos de código em um ambiente tridimensional, promovendo uma compreensão mais profunda dos conceitos de lógica e estruturação. Além disso, essa abordagem imersiva pode aumentar o engajamento dos estudantes, tornando o aprendizado mais dinâmico e atrativo.

Dado o cenário atual, o uso da RA para ensinar lógica de programação oferece uma oportunidade valiosa para superar as limitações do ensino tradicional e preparar melhor os estudantes para as demandas tecnológicas do futuro. Este trabalho propõe o desenvolvimento de um aplicativo para smartphones com sistema operacional Android, que utiliza realidade aumentada para interagir com *targets*. Os *targets*, nesse contexto, são elementos físicos ou digitais que servem como marcadores ou pontos de referência para o aplicativo reconhecer e sobre os quais projetar elementos virtuais. Eles podem ser imagens, objetos ou padrões específicos que, quando detectados pela câmera do dispositivo, acionam a interação ou a visualização dos conteúdos, em RA, previamente criados por desenvolvedores ou artistas gráficos. O objetivo é ensinar e reforçar conceitos de lógica de programação de forma inovadora, por meio de elementos visuais e práticos que tornam o aprendizado mais intuitivo, dinâmico e acessível.

## **2. Objetivos**

### **2.1. Objetivo geral**

O principal objetivo desta proposta é desenvolver um aplicativo para *Android* que utilize realidade aumentada como ferramenta interativa no ensino e reforço da lógica de programação. A proposta busca proporcionar uma experiência de aprendizado mais dinâmica e envolvente, explorando os elementos de RA para facilitar a compreensão dos conceitos de programação, tornando o processo mais eficiente, acessível e atrativo para os usuários.

### **2.2. Objetivos específicos**

- Criar e implementar cenas interativas que utilizem realidade aumentada para facilitar o ensino de conceitos fundamentais de lógica de programação.
- Desenvolver quizzes interativos para reforçar o aprendizado e avaliar o conhecimento dos usuários sobre lógica de programação e conceitos relacionados.

## **3. Trabalhos relacionados**

Em uma análise realizada por [Aureliano and Tedesco 2012] sobre o ensino de programação para iniciantes, foi observado que 65% dos trabalhos revisados propõem o uso de ferramentas de software como parte do processo de ensino-aprendizagem. A revisão sistemática considerou publicações brasileiras nos Anais do Simpósio Brasileiro de Informática na Educação (SBIE) e na Revista Brasileira de Informática na Educação (RBIE), abrangendo o período de 2005 a 2010. Essa preferência por soluções tecnológicas reflete a importância dessas ferramentas no apoio ao ensino de programação. No entanto, os autores enfatizam que tais ferramentas não têm como objetivo substituir os métodos tradicionais, mas sim atuar como suporte adicional no processo de ensino-aprendizagem.

Dentre as tecnologias emergentes com grande potencial de apoio educacional, destaca-se a realidade aumentada (RA). Segundo [Billinghamurst and Duenser 2012], a RA facilita a compreensão de fenômenos complexos ao fornecer experiências visuais e interativas únicas que combinam o real e o virtual, além de auxiliar na comunicação de conceitos abstratos aos aprendizes. A RA tem sido aplicada em diversas áreas de ensino, como no estudo de [Silva et al. 2014], que utilizou essa tecnologia no ensino de geografia, associando conceitos de quebra-cabeça físicos ao uso da RA para fornecer feedbacks visuais, táteis e sonoros. Essa abordagem despertou maior interesse e curiosidade nos alunos sobre os temas tratados.

Além do ensino básico, a RA também tem mostrado utilidade em áreas mais complexas. Um exemplo disso é o estudo de [Alves et al. 2020], no qual a RA foi proposta para facilitar a compreensão de conceitos de bioquímica em cursos de saúde e ciências naturais. Conceitos abstratos, como estruturas químicas e fórmulas matemáticas, são frequentemente considerados desafiadores por estudantes iniciantes. No estudo, o autor desenvolveu um jogo em realidade aumentada que apresenta perguntas sobre estruturas químicas de aminoácidos. O aluno recebe 20 cartas com modelos bidimensionais das estruturas e, ao escolher a resposta correta apontando a câmera do dispositivo para a carta, um modelo tridimensional da estrutura é exibido, oferecendo maior clareza sobre sua forma física.

Esses estudos demonstram que a realidade aumentada pode ser uma ferramenta poderosa em diversos contextos educacionais, desde o ensino básico até disciplinas mais especializadas. O uso de RA, ao proporcionar visualizações tridimensionais, interatividade e feedbacks em tempo real, pode auxiliar na superação de barreiras de aprendizado comuns, especialmente em áreas onde conceitos abstratos e visuais são predominantes. Segundo uma revisão de 25 anos de uso de RA na educação, a tecnologia tem mostrado ser eficaz no engajamento dos alunos e na melhoria da compreensão de conceitos complexos, criando novas oportunidades para o aprendizado interativo [Garzón 2021].

Embora os trabalhos citados tenham explorado diferentes áreas de aplicação, todos compartilham um ponto em comum: a RA oferece uma nova dimensão de suporte ao aprendizado, que vai além dos métodos tradicionais de ensino. Assim, a integração da RA no ensino de programação apresenta uma oportunidade valiosa para enriquecer a experiência de aprendizagem. A RA pode ser utilizada para visualizar algoritmos, facilitar a compreensão de conceitos lógicos e proporcionar feedback instantâneo aos alunos, tornando o aprendizado mais dinâmico e envolvente.

Neste sentido, o presente trabalho propõe o uso da RA como uma ferramenta inovadora no ensino de lógica de programação, com o objetivo de facilitar a aprendizagem de conceitos fundamentais para estudantes iniciantes. A proposta visa integrar a tecnologia ao currículo de ensino de computação de forma complementar, oferecendo atividades interativas que tornem o processo de aprendizado mais dinâmico e acessível.

## 4. Metodologia

### 4.1. Escolha dos desafios

Lahtinen, Ala-Mutka e Järvinen (2005) analisaram as correlações entre diferentes conceitos de programação com base nas respostas de mais de 500 alunos e professores universitários. Eles constataram que questões relacionadas a estruturas básicas, como *loops*, condicionais e operadores lógicos, estão fortemente interligadas de maneira positiva. Em outras palavras, um aluno que aprende facilmente uma dessas estruturas tende a aprender as demais com a mesma facilidade, enquanto aquele que enfrenta dificuldades em uma dessas áreas provavelmente terá problemas semelhantes com as outras.

Para abordar essas dificuldades, foram desenvolvidas três cenas focadas em temas relacionados às estruturas básicas. Nessas cenas, os alunos interagem com alvos (*targets*) de realidade aumentada para completar desafios específicos. Essa abordagem permite que os alunos reforcem seu conhecimento, além de possibilitar a geração de dúvidas e curiosidades sobre os temas abordados.

- **Cena 1:** Expressões lógicas

Nesta cena, o primeiro desafio consiste em organizar uma expressão lógica utilizando os *targets* fornecidos. Para esta demonstração, a expressão escolhida é "if (n % 2 == 0)", que determina se um número é par ou ímpar. O símbolo "%" é conhecido como operador de módulo e calcula o resto da divisão de um número por outro. Se o resto da divisão for 0, isso indica que o número é par. A correção da organização é verificada por meio de um questionário de múltipla escolha, no qual o aluno deve identificar a função da expressão. Se a resposta estiver correta, uma mensagem de sucesso é exibida e o aluno recebe autorização para avançar para a próxima cena.

- **Cena 2:** Quiz de múltipla escolha  
Aqui, um quiz de múltipla escolha controla dois "pokémons" em realidade aumentada, criaturas fictícias do universo da franquia *Pokémon*, criada pela Nintendo, Game Freak e Creatures [Nintendo, Game Freak, Creatures 1996]. O jogo *Pokémon GO* é amplamente reconhecido como um marco no gênero de jogos de realidade aumentada (RA) baseados em localização. De acordo com [Alha et al. 2019], ele foi o primeiro jogo desse tipo a alcançar popularidade global, tornando-se um dos títulos mais bem-sucedidos para dispositivos móveis. Antes do lançamento do *Pokémon GO*, os jogos de RA baseados em localização eram, em sua maioria, protótipos experimentais ou não conseguiam atingir um impacto comercial significativo [Alha et al. 2023]. Inspirado nesse sucesso, optou-se por utilizar o tema de *Pokémon* em uma das cenas do jogo desenvolvido neste trabalho. A escolha justifica-se pela familiaridade do público com o conceito de *Pokémon*, o que pode facilitar a imersão e o engajamento dos alunos no aprendizado de programação básica. Além disso, a integração de elementos de realidade aumentada, como no *Pokémon GO*, pode tornar a experiência de aprendizado mais interativa e envolvente. São apresentadas 4 opções de resposta, das quais apenas uma está correta. A resposta correta reduz um ponto de vida do oponente, enquanto a incorreta reduz um ponto de vida do aluno. O aluno deve acertar 3 das 5 perguntas para avançar para a próxima cena. Se o aluno perder 3 pontos de vida, uma mensagem de falha é exibida e ele poderá reiniciar a cena. Caso contrário, receberá a permissão para avançar para a cena seguinte.
- **Cena 3:** Portas lógicas  
Neste último desafio, o aluno deve posicionar 5 *targets* representando portas lógicas em um circuito lógico para gerar a saída esperada. Não há *feedback* imediato sobre a correção do posicionamento; a confirmação só ocorre quando todos os *targets* estão corretamente posicionados. Uma animação de vitória é exibida após o sucesso e assim encerra-se o ciclo do aplicativo.

## 4.2. Configuração do ambiente de desenvolvimento

O desenvolvimento do software envolveu a preparação do ambiente para realidade aumentada, utilizando tecnologias cuidadosamente selecionadas. Essa etapa é crucial para o sucesso da aplicação, pois impacta diretamente aspectos como escalabilidade, desempenho e manutenção. Para este projeto, foram escolhidos a Unity como plataforma principal para desenvolvimento 3D, o Vuforia pela sua integração eficiente com a Unity para realidade aumentada, além do Git para controle de versão e o Visual Studio para codificação.

### 4.2.1. Tecnologias utilizadas

- **Unity:**  
Fundada como Over the Edge Entertainment (OTEE) em Copenhague em 2004, por David Helgason, Nicholas Francis e Joachim Ante, a Unity Technologies tem como um de seus produtos, a plataforma Unity, um software de desenvolvimento de jogos 2D e 3D, que permite criar jogos para diversas plataformas, como computadores, celulares, óculos de realidade virtual, e aplicações de realidade aumentada. A escolha da Unity, em sua versão 2021.3.7f1, como plataforma de desenvolvimento do projeto foi baseada em sua robustez em aplicações 3D e suporte

completo à realidade aumentada. Utilizou-se a versão recomendada pela própria marca, na data do presente artigo, por sua estabilidade e confiabilidade.



**Figura 1. Unity.**

**Fonte:** [https://upload.wikimedia.org/wikipedia/commons/thumb/1/19/Unity\\_Technologies\\_Logo.svg/1280px-Unity\\_Technologies\\_Logo.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/1/19/Unity_Technologies_Logo.svg/1280px-Unity_Technologies_Logo.svg.png)

- **Visual Studio 2022:**

O Visual Studio é uma IDE (Ambiente de Desenvolvimento Integrado, do inglês *Integrated Development Environment*), que oferece integração com a Unity, independentemente da versão do seu lançamento, desde que seja a versão 2022. A Unity realiza a compilação de código em tempo de execução utilizando a técnica de compilação Just-in-Time (JIT). Essa técnica permite que o código seja compilado durante a execução do programa, o que possibilita a otimização dinâmica com base no contexto específico de execução, resultando em um potencial aumento de desempenho do software [Oliveira and da Silva 2013]. A plataforma de software utilizada para essa compilação é o Mono, um projeto de código aberto voltado para o desenvolvimento multiplataforma, baseado no .NET Framework [Mono Project 2024]. Criado pela Microsoft, o .NET Framework é uma plataforma de desenvolvimento que fornece um ambiente para a construção, implementação e execução de aplicativos, incluindo serviços web, aplicativos desktop e móveis. Ele oferece uma ampla biblioteca de classes reutilizáveis e componentes que facilitam o desenvolvimento [Microsoft 2024]. Com recursos de depuração, refatoração de código e ferramentas de gerenciamento de projetos, o Visual Studio 2022 se destaca como um ambiente de desenvolvimento completo e integrado à Unity, sendo a escolha ideal para o projeto em questão.



**Figura 2. Visual Studio 2022.**

**Fonte:** Fonte: <https://www.aomeitech.com/screenshot/windows-tips/others/visual-studio/visual-studio-logo.png>

- **Vuforia:**

O Vuforia Engine é uma plataforma amplamente usada para desenvolvimento de

realidade aumentada, com suporte para a maioria dos telefones, tablets e óculos. Os desenvolvedores podem facilmente adicionar funcionalidade avançada de visão computacional a aplicativos Android, iOS e UWP (*Universal Windows Platform*), uma plataforma da Microsoft que permite criar aplicativos para diversos dispositivos Windows, como PCs, *smartphones* e Xbox, utilizando uma base de código única. O Vuforia permite criar experiências de realidade aumentada que interagem de forma realística com objetos e o ambiente [Vuforia 2024]. Neste trabalho, foi utilizada a versão 11 do SDK, à qual oferece todo o suporte necessário para a aplicação.



**Figura 3. Vuforia Engine**

**Fonte:** <https://integralplm.com/wp-content/uploads/2021/12/vuforia-engine-logo.png>

- **Git:**

O Git é um sistema de controle de versão distribuído, gratuito e de código aberto, projetado para gerenciar uma ampla gama de projetos, desde os mais simples até os mais complexos, com alta velocidade e eficiência [Git 2024]. Neste trabalho, utilizou-se também o GitHub, uma plataforma de hospedagem de código-fonte que serve como repositório para armazenar e gerenciar o código-fonte do aplicativo, aumentando a produtividade no desenvolvimento.



**Figura 4. Git 2024.**

**Fonte:** <https://git-scm.com/images/logo@2x.png>

#### **4.2.2. Desenvolvimento para dispositivos Android**

Para o desenvolvimento de aplicativos Android na Unity, é necessário instalar o Android Build Support. Este módulo inclui componentes cruciais como o Android SDK e o Android NDK. Além disso, ele inclui bibliotecas de suporte que garantem a compatibilidade com diferentes versões do Android, facilitando a depuração e testes em dispositivos reais e emuladores.

- **OpenJDK:** O OpenJDK (Open Java Development Kit) é uma implementação de código aberto da linguagem Java, necessária para compilar e executar código Java. Ele fornece as bibliotecas e ferramentas necessárias para a criação e execução de código Java, que é um requisito para o desenvolvimento de aplicativos Android. O OpenJDK é essencial para a configuração e execução das builds Android na Unity, pois muitas funcionalidades do Android SDK e do próprio Unity dependem do Java.
- **Android SDK:** O Android SDK (Software Development Kit) é um conjunto abrangente de ferramentas e bibliotecas para o desenvolvimento de aplicativos Android. Inclui ferramentas de compilação, emulação e depuração, além de APIs e pacotes de suporte para diversas versões do Android. O SDK permite que o Unity interaja e teste o aplicativo em dispositivos Android e emuladores. Ferramentas importantes incluem o Android Debug Bridge (ADB) para comunicação com dispositivos e o Android Emulator para simulação do ambiente Android.
- **Android NDK:** O Android NDK (Native Development Kit) permite o desenvolvimento de código nativo em C e C++. Ele é útil para integrar partes do código que exigem alto desempenho ou funcionalidades específicas do sistema. No Unity, o NDK é utilizado para criar e gerenciar bibliotecas nativas que podem ser chamadas a partir do código C# do Unity, proporcionando melhor desempenho para tarefas críticas e acesso a APIs nativas do Android que não estão disponíveis através do SDK.

## 5. Resultado

Os resultados alcançados com o desenvolvimento do protótipo foram positivos, com a criação de uma versão inicial dos três problemas abordados na metodologia. O jogo foi projetado para utilizar a realidade aumentada no ensino de conceitos introdutórios de lógica de programação, permitindo aos usuários interagir com *targets* e resolver desafios visuais. As três cenas desenvolvidas abordam aspectos fundamentais do ensino de lógica, como expressões condicionais, declarações de variáveis e o uso de portas lógicas. Embora a avaliação com usuários não tenha sido realizada nesta etapa, futuros trabalhos poderão explorar essa análise para verificar o impacto da aplicação no aprendizado dos usuários.

### 5.1. Seleção dos *Targets*

Neste trabalho, optou-se pelo uso de *targets* de Realidade Aumentada (RA) no formato de QR Codes devido à sua simplicidade e praticidade. Esses padrões são amplamente reconhecíveis, fáceis de escanear e compatíveis com diversas plataformas. Contudo, é importante destacar que outras imagens também podem ser utilizadas como *targets* RA, desde que sejam devidamente avaliadas quanto à sua qualidade. A plataforma Vuforia, por exemplo, avalia a eficácia dos *targets* com uma escala de 0 a 5 estrelas, indicando sua adequação para rastreamento, como pode ser observado na Figura 5. O conceito de eficácia e os critérios utilizados para essa avaliação serão discutidos com mais detalhes na Seção 5.4.

### 5.2. Uso da Plataforma Vuforia

A plataforma Vuforia é uma ferramenta poderosa para o desenvolvimento de aplicações de Realidade Aumentada. Para utilizar o Vuforia, os *targets* (QR Codes ou

outras imagens) devem ser carregados na plataforma, onde serão processados e avaliados quanto à sua qualidade de rastreamento. O Vuforia fornece um SDK (Software Development Kit), que é um conjunto de ferramentas de desenvolvimento que permite a criação de aplicações de software. Esse SDK pode ser integrado ao *Unity Engine*, permitindo o desenvolvimento de aplicações interativas de RA.

### 5.3. Eficácia dos *Targets*

A eficácia dos *targets* refere-se à sua capacidade de ser detectado e rastreado pelo sistema de RA. A Vuforia utiliza uma escala de 0 a 5 estrelas para avaliar essa eficácia, onde *targets* com uma pontuação maior são considerados mais adequados para rastreamento. Essa avaliação, que é realizada automaticamente pelo algoritmo da plataforma ao realizar o *upload* da imagem que será utilizada como *target*, o qual não é disponibilizado ao público, leva em conta fatores como contraste, complexidade e repetitividade do padrão visual do *target*, que afetam diretamente a precisão e estabilidade do

<input type="checkbox"/>	Image	Target Name	Type	Rating	Status	Date Modified
<input type="checkbox"/>		qr6	Image	★★★★☆	Active	Jun 11, 2024
<input type="checkbox"/>		qr5	Image	★★★★☆	Active	Jun 11, 2024
<input type="checkbox"/>		qr4	Image	★★★★☆	Active	Jun 11, 2024
<input type="checkbox"/>		qr3	Image	★★★★☆	Active	Jun 11, 2024
<input type="checkbox"/>		qr2	Image	★★★★☆	Active	Jun 11, 2024
<input type="checkbox"/>		qr1	Image	★★★★☆	Active	Jun 11, 2024

**Figura 5. Ranking de *targets* disponíveis ao realizar upload de imagens na plataforma Vuforia.**

**Fonte:** Autor, gerada na conta pessoal da plataforma Vuforia.

*Targets* com maior pontuação apresentam características como alto contraste, texturas complexas e detalhes não repetitivos, fatores que facilitam o rastreamento pelo algoritmo de RA fornecido junto ao SDK da plataforma Vuforia, o qual dispensa implementações complexas. É necessário apenas referenciar os objetos e configurá-los conforme as especificações da plataforma, como o tamanho real dos *targets*, a disposição vertical ou horizontal dos objetos virtuais, entre outros parâmetros. Por outro lado, imagens com grandes áreas monocromáticas ou padrões repetitivos tendem a prejudicar a experiência, dificultando o rastreamento e reduzindo a precisão. Assim, apesar da conveniência dos *QR Codes*, a flexibilidade da plataforma permite explorar outros tipos de imagens, desde que sejam otimizadas para o desempenho no contexto de RA. A plataforma Vuforia disponibiliza a análise e a classificação dos *targets* logo após o upload, como mostrado na figura 6.

qr6

Edit Name Remove



**Figura 6. Target com as marcações realizadas pelo algoritmo da plataforma Vuforia.**

**Fonte:** Autor, gerada na conta pessoal da plataforma Vuforia.

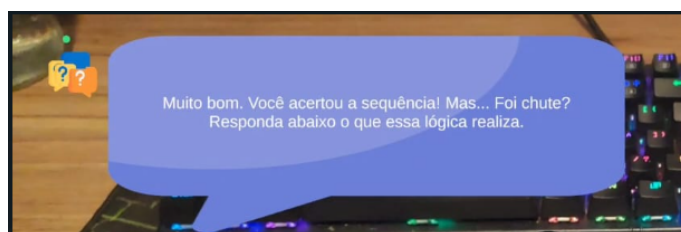
#### **5.4. Desenvolvimento dos desafios propostos**

Ao iniciar a aplicação, o usuário é apresentado à tela inicial, onde a câmera do telefone exibe sua imagem. Essa tela foi propositalmente mantida limpa de elementos visuais para maximizar o espaço interativo para o uso da RA. Na figura 7, é possível visualizar a tela inicial, que exibe um botão de informações no canto superior esquerdo. Ao clicar nesse botão, é mostrado um texto explicando o desafio atual. Na parte superior direita, há botões para sair da aplicação ou avançar para o próximo desafio. É importante ressaltar que o teclado, os *targets*, a mesa e demais componentes presentes nas imagens, com exceção dos botões virtuais, são todos itens físicos capturados em tempo real pela câmera durante a execução da aplicação por meio de captura de tela.



**Figura 7. Tela inicial da aplicação.**

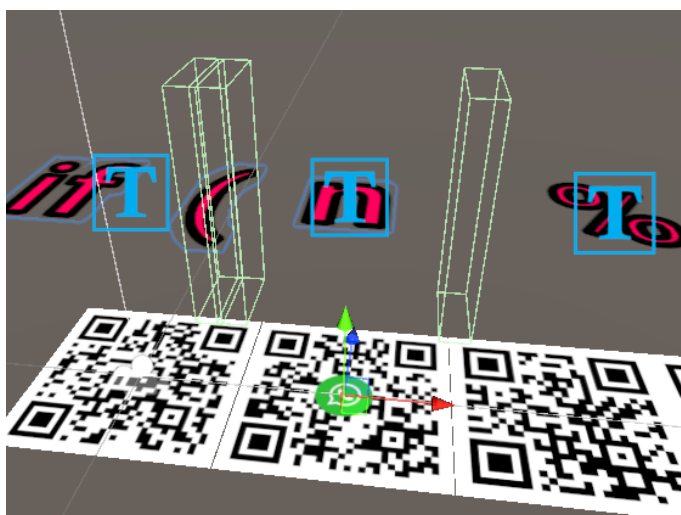
**Fonte:** Autor



**Figura 8. Botão informativo aberto na tela inicial.**

**Fonte:** Autor

O primeiro desafio proposto consiste em posicionar os *targets* para formar uma expressão condicional válida. Cada *target* possui, virtualmente, um colisor em suas extremidades. Os colisores, elementos do sistema de física da Unity, detectam interações entre objetos. Quando os colisores entram em contato, um evento é disparado, permitindo a execução da lógica programada.



**Figura 9. Colisores (retângulos) vistos dentro da Unity.**

**Fonte:** Autor

Ao posicionar corretamente os *targets*, os colisores ativam a validação lógica do jogo. Apenas uma configuração correta é reconhecida como válida. Para garantir o aprendizado e evitar tentativas aleatórias, o usuário precisa responder um quiz de múltipla escolha antes de avançar para o próximo desafio. Após responder corretamente, uma mensagem aparece no balão informativo, indicando que o próximo desafio está liberado. Caso o quiz, contendo uma sequência de 4 possíveis respostas para o significado da expressão do desafio proposto, onde apenas uma das respostas é correta, não seja concluído com sucesso, o avanço não será permitido.

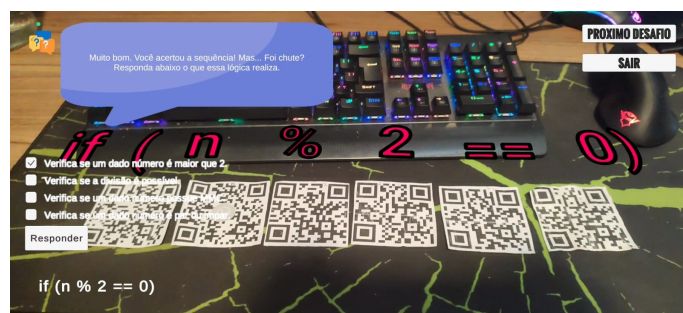


Figura 10. Mensagem de sucesso após responder corretamente o quiz.

Fonte: Autor

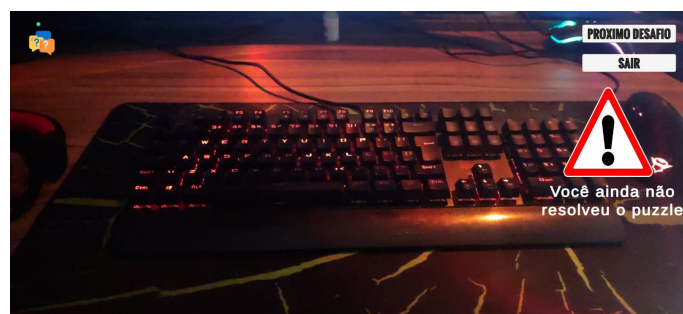


Figura 11. Mensagem de erro ao tentar avançar para o desafio seguinte sem concluir o atual.

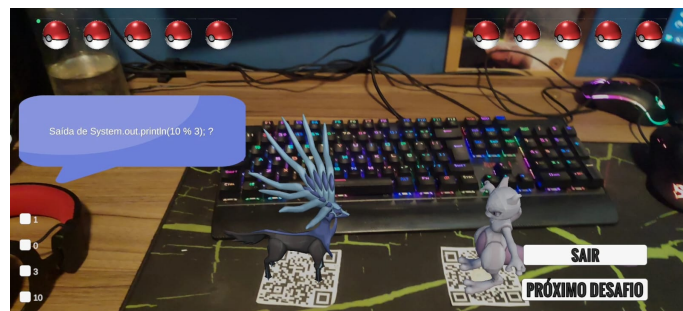
Fonte: Autor

Após concluir o primeiro desafio, o usuário pode clicar no botão para entrar na segunda cena. Nela, o desafio consiste em responder corretamente a um quiz de múltipla escolha. Cada resposta correta reduz um ponto do "adversário" (um componente passivo), enquanto respostas erradas reduzem pontos do próprio usuário. Para avançar, é necessário acertar 3 de 5 perguntas. Caso não consiga, o quiz pode ser reiniciado. As *pokebolas*, conhecidas na franquia *Pokémon* [Nintendo, Game Freak, Creatures 1996], são dispositivos usados para capturar e armazenar *Pokémon*. No contexto deste jogo, as *pokebolas* representam os pontos de vida dos jogadores, sendo reduzidas de acordo com as respostas corretas ou erradas. Cada jogador começa com um número determinado de *pokebolas*, e perde uma *pokebola* a cada erro, enquanto acertos diminuem as *pokebolas* do adversário.



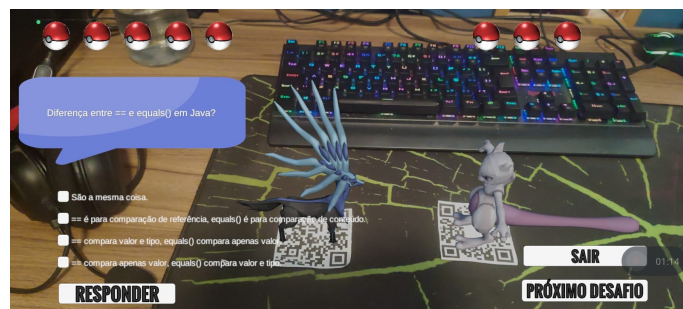
**Figura 12. Pokebola.**

**Fonte:** <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRGGj4XKRiKLnTWr5MFOu6Ud1KWGNM3hs9Lpg&s>



**Figura 13. Cena 2 após a realização do escaneamento dos *targets*.**

**Fonte:** Autor



**Figura 14. Cena 2 após acertar duas respostas. As *pokebolas* que representam os pontos são removidas.**

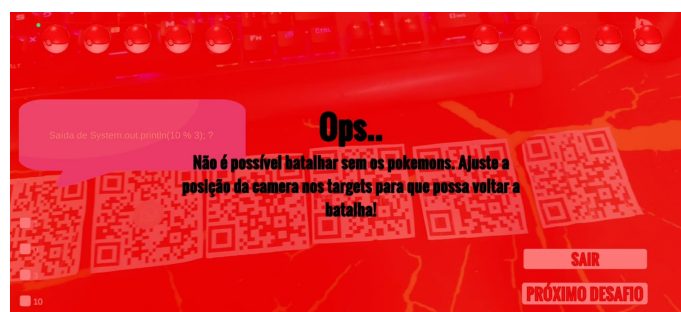
**Fonte:** Autor



**Figura 15.** Tela escurificada (sem imagem da câmera) após errar mais de 3 respostas do quiz. Apresenta dois botões de ação para o usuário.

**Fonte:** Autor

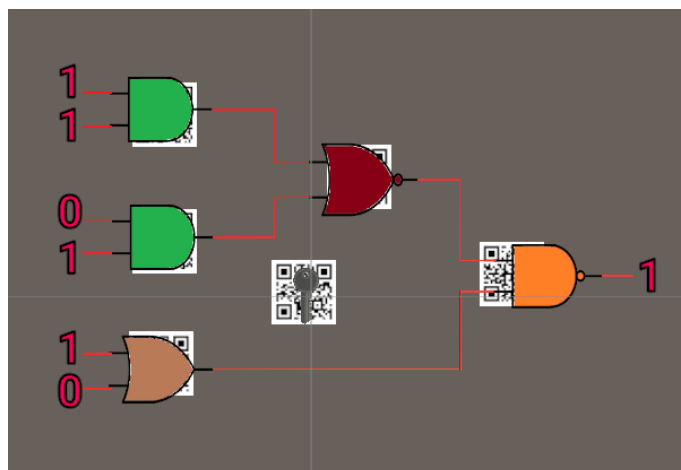
É imprescindível que o *targets* de referência (*QR Codes*) estejam visíveis e devidamente escaneados na tela. Caso contrário, o avanço será bloqueado, e uma mensagem de erro orientará o usuário a realizar o posicionamento da câmera sobre os *targets* corretamente, permitindo que os modelos virtuais associados a eles sejam exibidos. Isso é utilizado para que em outras versões possam ser implementados efeitos visuais como animações, sons e outros, além de manter a interatividade da realidade aumentada.



**Figura 16.** Mensagem de erro bloqueando o usuário até a correta realização do escaneamento.

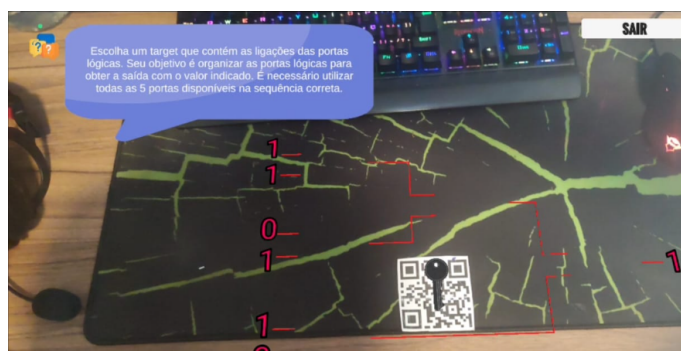
**Fonte:** Autor

Ao completar o segundo desafio, o usuário prossegue para a terceira cena, que envolve o posicionamento correto de *targets* para formar um circuito lógico. Nessa etapa, um *target* (*QR code*) principal é utilizado para representar o circuito vazio, funcionando como uma base central sobre a qual os outros *targets*, que representam portas lógicas, devem ser posicionados. Esses *targets* adicionais precisam ser dispostos adequadamente para produzir a saída desejada. Atualmente, a validação considera apenas uma sequência previamente definida, o que simplifica o desenvolvimento. No entanto, há potencial para incluir a validação de outras combinações no futuro, tornando o desafio mais dinâmico e flexível.



**Figura 17. Sequência da cena 3 montada corretamente dentro da Unity.**

**Fonte:** Autor



**Figura 18. Tela da cena 3 com *target* especial utilizado como referência.**

**Fonte:** Autor

Por fim, ao completar a terceira cena, uma animação é exibida, e a tela final da aplicação aparece. Nessa tela, o usuário pode optar por sair ou reiniciar a aplicação. Vale ressaltar que a reinicialização resulta na perda de todo o progresso, pois este protótipo não implementa persistência de dados.



**Figura 19. Tela final da aplicação.**

**Fonte:** Autor

## 6. Conclusão

O protótipo desenvolvido neste trabalho demonstrou o potencial da realidade aumentada como uma ferramenta eficaz para o ensino de programação, oferecendo uma experiência interativa e envolvente para os usuários. Ao utilizar o Vuforia e a Unity, foi possível criar um ambiente no qual os alunos podem explorar conceitos de lógica de programação de forma prática e visual. A escolha de *targets* visuais em formato de *QR Code*, além de simples e acessível, mostrou-se eficiente para a interação com o conteúdo virtual, facilitando a aprendizagem e mantendo o foco do usuário.

Os desafios implementados, como construção de expressões condicionais, quizzes e circuitos lógicos, foram essenciais para a progressão no aprendizado. Além disso, a interatividade com objetos virtuais proporcionou uma aprendizagem mais dinâmica e menos tradicional.

Contudo, o protótipo ainda possui amplo espaço para aprimoramentos. Possibilidades futuras incluem melhorias específicas para cada cena. Na Cena 1, a implementação de uma opção para os usuários escolherem as expressões ou optarem por expressões aleatórias, além do ajuste na quantidade de *targets*, pode aumentar a flexibilidade. Na Cena 2, a ampliação da variedade de perguntas com seleção aleatória pode tornar as atividades mais desafiadoras. Na Cena 3, a introdução de uma lógica que simule o funcionamento de portas lógicas possibilitaria a criação de circuitos dinâmicos, com funcionalidades como alertas sobre erros, identificação de falhas e dicas. Para todas as cenas, a inclusão de efeitos visuais e sonoros mais elaborados, como animações, sons dinâmicos e elementos tridimensionais, pode aumentar a imersão e o engajamento dos usuários.

O suporte multiplataforma pode ser expandido para incluir iOS, PCs e óculos de realidade virtual. Embora cada plataforma exija adaptações específicas, essas modificações são viáveis e podem aumentar significativamente o alcance e impacto do protótipo.

Para o iOS, é necessário ajustar o protótipo para garantir sua compatibilidade com dispositivos como iPhones e iPads. Embora o desenvolvimento para essa plataforma demande um esforço considerável, o grande público de usuários da Apple oferece uma oportunidade de expandir o acesso à tecnologia, permitindo que mais pessoas, além dos usuários Android, se beneficiem da abordagem educacional inovadora do protótipo.

Nos computadores, a diversidade de sistemas operacionais e configurações de hardware exige uma abordagem robusta para garantir a compatibilidade com Windows, macOS e Linux. O uso da Unity, que já oferece suporte a essas plataformas, facilita essa adaptação. Além disso, a versão para desktop proporciona uma nova dinâmica de interação, com o uso de mouse e teclado, o que permite a realização de exercícios mais complexos e interações detalhadas no processo de aprendizagem.

A expansão para óculos de realidade virtual, por sua vez, oferece uma experiência imersiva e interativa. A utilização de realidade virtual no ensino de lógica de programação pode transformar a visualização dos conceitos, criando uma conexão mais tangível com o conteúdo. A interação com ambientes virtuais e a simulação de circuitos lógicos em 3D são exemplos de como essa plataforma pode enriquecer o aprendizado, tornando-o mais dinâmico e envolvente.

Embora cada plataforma demande ajustes específicos, as tecnologias atuais tornam essas modificações viáveis, ampliando o alcance do protótipo. Com suporte multiplataforma, o protótipo pode atingir diferentes públicos, desde usuários móveis até aqueles que tem acesso a dispositivos que permitam uma experiência imersiva com realidade virtual, aumentando seu impacto e abrindo novas oportunidades no ensino de lógica de programação.

Este trabalho explorou o potencial da tecnologia de realidade aumentada como uma aliada no ensino de lógica de programação, destacando sua capacidade de estimular a aprendizagem ativa e proporcionar um ambiente mais interativo e motivador. Com os aprimoramentos propostos, o protótipo tem potencial para oferecer uma abordagem ainda mais robusta e transformadora no ensino de programação.

## Referências

- Alha, K., Koskinen, E., Paavilainen, J., and Hamari, J. (2019). Why do people play location-based augmented reality games: A study on pokémon go. *Computers in Human Behavior*, 93:114–122.
- Alha, K., Leorke, D., Koskinen, E., and Paavilainen, J. (2023). Augmented play: An analysis of augmented reality features in location-based games. *Convergence: The International Journal of Research into New Media Technologies*, 29:342–361.
- Alves, A., Machado, L., and Felipe, C. (2020). Arminogame: Um jogo colaborativo com realidade aumentada para ensino de bioquímica estrutural. In *Anais do V Congresso sobre Tecnologias na Educação*, pages 482–491, Porto Alegre, RS, Brasil. SBC.
- Aureliano, V. and Tedesco, P. (2012). Ensino-aprendizagem de programação para iniciantes: uma revisão sistemática da literatura focada no sbie e wie. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, 23(1).
- Billinghamurst, M. and Duenser, A. (2012). Augmented reality in the classroom. *Computer*, 45(7):56–63.
- Garzón, J. (2021). An overview of twenty-five years of augmented reality in education. *Multimodal Technologies and Interaction*, 5(7).
- Git (2024). Git - the simple guide. <https://git-scm.com/>. Acesso em: 23 ago. 2024.
- Martins, V. F. and Guimarães, M. P. (2012). Desafios para o uso de realidade virtual e aumentada de maneira efetiva no ensino. In *Anais do Workshop de Desafios da Computação Aplicada à Educação (desafie)*, pages 100–109, São Paulo. Sociedade Brasileira de Computação.
- Microsoft (2024). .net framework — download, instalação e suporte. <https://dotnet.microsoft.com/pt-br/download/dotnet-framework>. Acesso em: 23 ago. 2024.
- Mono Project (2024). Mono: A cross platform .net framework. <https://www.mono-project.com/>. Acesso em: 23 ago. 2024.
- Nintendo, Game Freak, Creatures (1996). Pokémon. <https://www.pokemon.com>. Acesso em: 23 dez. 2024.

- Oliveira, G. S. and da Silva, A. F. (2013). Compilação just-in-time: Histórico, arquitetura, princípios e sistemas. *Revista de Informática Teórica e Aplicada*, 20(2):174–213.
- Scaico, P. D., Henrique, M. S., Miranda Cunha, F. O., and de Alencar, Y. M. (2012). Um relato de experiências de estagiários da licenciatura em computação com o ensino de computação para crianças. *Revista Novas Tecnologias na Educação*, 10(3).
- Silva, M., Vilar Neto, E., Reis, G., Lima, J., and Teichrieb, V. (2014). Ar jigsaw puzzle: Potencialidades de uso da realidade aumentada no ensino de geografia. page 194.
- Vuforia (2024). Vuforia developer library. <https://developer.vuforia.com/library/>. Acesso em: 23 ago. 2024.