Inclusão Através da Voz: Acessibilidade em Jogos Educativos Estilo Quiz com Reconhecimento de Voz

João Victor Alves Cerqueira[1], **Victor Travassos Sarinho**[1]

[1]Laboratório de Entretenimento Digital Aplicado (LEnDA)
Universidade Estadual de Feira de Santana (UEFS)
Feira de Santana, BA - Brasil

jvacerqueira@ecomp.uefs.br, vsarinho@uefs.br

Abstract. *Accessibility in digital games is still limited, especially for people with visual or motor impairments, restricting their participation in gamified interactive experiences. This work proposes the integration of speech recognition and screen reading into the AsKME platform, aiming to promote inclusive interaction in quiz-style games developed on the platform. A hybrid architecture was developed combining Web Speech API, SpeechSynthesis, and a complementary module using FastAPI and Google Speech-to-Text, enabling voice input and audio feedback for games configured within the AsKME platform. The solution allows users to interact with games autonomously through voice commands, providing accessibility across different environments and contributing to digital inclusion in quiz-style games that can be produced at scale.*

Resumo. *A acessibilidade em jogos digitais ainda é limitada, especialmente para pessoas com deficiência visual ou motora, restringindo sua participação em experiências interativas gamificadas. Este trabalho propõe a integração de reconhecimento de voz e leitura de tela na plataforma AsKME, visando promover interação inclusiva em jogos estilo quiz produzidos. Para tal, foi desenvolvida uma arquitetura híbrida combinando Web Speech API, SpeechSynthesis e um módulo complementar com FastAPI e Google Speech-to-Text, permitindo entrada por voz e feedback em áudio para jogos configurados na plataforma AsKME. A solução permite que usuários interajam com jogos de forma autônoma por meio de comandos de voz, fornecendo acessibilidade em diferentes ambientes e contribuindo para a inclusão digital em jogos estilo quiz capazes de serem produzidos em larga escala.*

1. Introdução

Jogos digitais têm sido amplamente utilizados em contextos educacionais devido ao seu potencial de engajamento e promoção da aprendizagem ativa [Prensky 2003], [Gee 2003]. Entretanto, a acessibilidade ainda representa um desafio significativo, especialmente para pessoas com deficiência visual ou motora, que podem encontrar dificuldades em interfaces que dependem predominantemente de teclado, mouse, toque ou informações apresentadas visualmente [Yuan et al. 2011]. Nesse contexto, a acessibilidade em jogos envolve a redução de barreiras relacionadas à percepção, compreensão e controle das experiências interativas, demandando alternativas às modalidades convencionais de entrada e saída.

Entre os diferentes formatos de jogos digitais utilizados na educação, os jogos estilo *quiz* destacam-se por sua versatilidade e aplicabilidade à revisão e fixação de conteúdos. Caracterizados pela apresentação de perguntas e pela seleção de alternativas ou fornecimento de respostas, esses jogos favorecem a avaliação imediata do conhecimento e o reforço contínuo da aprendizagem [Deterding et al. 2011], [Hamari et al. 2014]. Além disso, sua estrutura relativamente simples torna possível mapear as principais ações de interação para diferentes modalidades de entrada, incluindo comandos de voz.

Nesse contexto, a plataforma AsKME (*Assessment of Knowledge Multiplatform Environment*) foi proposta como uma abordagem baseada em features para o desenvolvimento de jogos estilo quiz multiplataforma [Sarinho et al. 2018]. A plataforma organiza características comuns e variáveis desse domínio, permitindo a construção de diferentes jogos a partir de uma infraestrutura compartilhada. Essa característica favorece a reutilização de funcionalidades entre diferentes jogos e constitui uma estratégia alinhada aos princípios de Linhas de Produto de Software (LPS), nos quais produtos relacionados são desenvolvidos a partir de ativos reutilizáveis e mecanismos de variabilidade [Clements and Northrop 2002], [Pohl et al. 2005].

Embora a abordagem da AsKME favoreça a produção e reutilização de jogos estilo quiz, sua interação originalmente baseada em mecanismos convencionais de entrada não contempla, de forma específica, alternativas de interação voltadas a usuários que apresentam limitações para utilizar teclado, mouse ou interfaces predominantemente visuais. Essa lacuna é particularmente relevante em uma plataforma orientada à produção de múltiplos jogos, pois a implementação individual de recursos de acessibilidade em cada jogo poderia aumentar o esforço de desenvolvimento e dificultar sua reutilização. Assim, a incorporação desses recursos diretamente à infraestrutura da plataforma apresenta-se como uma alternativa para ampliar a acessibilidade de diferentes jogos sem modificar sua lógica de interação individualmente.

Diante desse problema, este trabalho propõe a integração de mecanismos de reconhecimento de voz e feedback auditivo à plataforma AsKME, permitindo que ações originalmente realizadas por meio de entradas convencionais também possam ser executadas por comandos de voz. A proposta utiliza uma arquitetura híbrida, combinando recursos de reconhecimento e síntese de fala disponíveis no ambiente web com um mecanismo complementar baseado em servidor. Além disso, é introduzido um mecanismo de escuta contínua capaz de identificar palavras-chave e interpretar comandos de acordo com o estado atual do jogo. Dessa forma, busca-se ampliar as possibilidades de interação em jogos estilo quiz produzidos na plataforma, preservando sua lógica original e favorecendo a reutilização dos recursos de acessibilidade.

2. Fundamentação Teórica

2.1. Acessibilidade em Jogos Digitais

A acessibilidade em jogos digitais envolve a redução de barreiras que podem dificultar ou impedir a participação de pessoas com diferentes tipos de deficiência. Essas barreiras podem estar relacionadas tanto à percepção das informações apresentadas pelo jogo quanto aos mecanismos utilizados para interagir com seus elementos. [Yuan et al. 2011] discutem diferentes desafios de acessibilidade em jogos digitais, destacando limitações

relacionadas às modalidades de entrada e saída e à dependência de dispositivos e interfaces convencionais. Nesse contexto, a adoção de modalidades alternativas de interação pode contribuir para ampliar as possibilidades de acesso aos jogos.

[Cairns and Cox 2008] também destacam a importância de considerar diferentes necessidades dos jogadores durante o projeto de jogos acessíveis. A disponibilização de recursos de áudio, comandos de voz e diferentes formas de *feedback* pode reduzir a dependência de uma única modalidade de interação e favorecer experiências mais inclusivas. Essa perspectiva está alinhada aos princípios gerais de acessibilidade para conteúdos digitais, que recomendam que informações e funcionalidades não dependam exclusivamente de uma única capacidade sensorial ou motora [Schwartz et al. 2025]. Além disso, estudos sobre acessibilidade em jogos digitais indicam que a integração de múltiplas modalidades de interação, tais como áudio, voz e *feedback* visual, é fundamental para ampliar o acesso e melhorar a experiência de usuários com deficiência [Yuan et al. 2011], [Cairns and Cox 2008].

No contexto de jogos educacionais, essas questões assumem importância adicional, uma vez que as barreiras de interação podem comprometer não apenas a experiência de jogo, mas também o acesso ao conteúdo pedagógico. Entre os diferentes gêneros utilizados para fins educacionais, jogos estilo *quiz* apresentam características particularmente favoráveis à adaptação acessível. Sua interação é geralmente estruturada em torno da apresentação de perguntas, seleção de alternativas e fornecimento de *feedback*, permitindo que ações originalmente associadas a teclado, mouse ou toque sejam mapeadas para modalidades alternativas, como comandos de voz.

2.2. Interfaces por Voz e Tecnologias Assistivas

O reconhecimento automático de fala tem sido explorado como uma alternativa aos mecanismos convencionais de interação humano-computador, especialmente em situações nas quais teclado, mouse ou outras formas de entrada podem representar barreiras. Sistemas de reconhecimento convertem a fala do usuário em texto ou comandos que podem ser processados por uma aplicação, permitindo a construção de interfaces baseadas em linguagem natural ou em conjuntos restritos de comandos.

Entre as tecnologias utilizadas para esse propósito estão motores de reconhecimento executados localmente, como o CMU Sphinx[1], e serviços de reconhecimento disponibilizados em nuvem, como o *Google Speech-to-Text*[2]. Essas tecnologias têm sido empregadas em diferentes contextos, incluindo assistentes virtuais, aplicações educacionais e tecnologias assistivas. Sua utilização permite substituir ou complementar modalidades convencionais de entrada, oferecendo alternativas de interação que podem ser particularmente relevantes para usuários com limitações motoras ou visuais.

Interfaces de voz, entretanto, apresentam características e limitações próprias. O desempenho do reconhecimento pode ser influenciado por fatores como ruído ambiental, qualidade do dispositivo de captura, pronúncia, velocidade da fala e características individuais dos usuários. Além disso, abordagens baseadas em serviços remotos introduzem dependências relacionadas à conectividade, latência e ao processamento de dados de voz

[1] `https://cmusphinx.github.io/`
[2] `https://cloud.google.com/speech-to-text/docs`

por terceiros. Dessa forma, embora o reconhecimento de fala possa ampliar as possibilidades de interação, sua utilização como recurso de acessibilidade requer a consideração das condições nas quais o sistema será utilizado.

No contexto deste trabalho, a interação por voz é utilizada como modalidade complementar, e não como substituição obrigatória das formas convencionais de interação. Essa característica permite preservar o funcionamento original dos jogos e, simultaneamente, disponibilizar uma alternativa para usuários que apresentem dificuldades na utilização de teclado, mouse ou outras modalidades tradicionais.

2.3. Web Speech API

No contexto de aplicações web, a *Web Speech API*[3] disponibiliza mecanismos para reconhecimento de voz e síntese de fala diretamente no navegador. A API é composta principalmente pelos recursos de *Speech Recognition*, responsável pela conversão da fala em texto, e *Speech Synthesis*, responsável pela conversão de texto em fala. Essa integração permite incorporar interação por voz e *feedback* auditivo a aplicações web sem que esses mecanismos precisem ser implementados integralmente pela própria aplicação.

O uso de interfaces de voz em aplicações interativas também tem sido investigado em contextos de uso cotidiano e educacional. [Porcheron et al. 2018], por exemplo, analisam a interação de usuários com sistemas baseados em voz, evidenciando características específicas dessa modalidade de interação em comparação com interfaces gráficas convencionais. Esses estudos reforçam o potencial das interfaces de voz como uma modalidade adicional de interação, mas também evidenciam a necessidade de considerar aspectos relacionados ao contexto de uso.

Apesar das vantagens proporcionadas pela Web Speech API, seu emprego em aplicações que demandam ampla compatibilidade apresenta limitações. O suporte ao reconhecimento de voz não é uniforme entre navegadores, e determinadas funcionalidades podem depender de implementações específicas. Além disso, o comportamento do reconhecimento pode estar associado a serviços externos e, consequentemente, à disponibilidade de conectividade. Essas características dificultam a obtenção de um comportamento completamente uniforme entre diferentes dispositivos e ambientes de execução.

Nesse sentido, a utilização da Web Speech API é adequada para explorar a interação por voz diretamente no ambiente web, mas aplicações que necessitam de maior controle sobre o processo de reconhecimento podem recorrer a mecanismos complementares. Essa possibilidade fundamenta a adoção, neste trabalho, de uma arquitetura híbrida que combina o reconhecimento disponibilizado pelo navegador com um mecanismo alternativo baseado em servidor.

2.4. Linhas de Produto de Software e a Plataforma AsKME

A reutilização de mecanismos de acessibilidade é particularmente relevante quando diferentes jogos compartilham uma mesma infraestrutura. Nesse contexto, Linhas de Produto de Software (LPS) constituem uma abordagem para o desenvolvimento sistemático de famílias de produtos relacionados, explorando características comuns e variáveis e promovendo a reutilização de ativos de software [Clements and Northrop 2002],

[3] `https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API`

[Pohl et al. 2005]. Em vez de desenvolver cada produto de forma independente, uma LPS permite organizar funcionalidades reutilizáveis que podem ser combinadas para produzir diferentes produtos de uma mesma família.

A plataforma AsKME foi proposta nesse contexto como uma abordagem baseada em *features* para o desenvolvimento de jogos estilo *quiz* multiplataforma [Sarinho et al. 2018]. Sua arquitetura permite organizar características comuns e variáveis dos jogos estilo quiz, favorecendo a reutilização de componentes e a geração de diferentes produtos a partir de uma infraestrutura compartilhada.

Essa característica torna a AsKME um contexto adequado para a incorporação de recursos de acessibilidade. Em uma abordagem convencional, cada jogo poderia implementar individualmente mecanismos de reconhecimento de voz e *feedback* auditivo. Em uma arquitetura baseada em reutilização, entretanto, esses mecanismos podem ser incorporados à infraestrutura comum da plataforma e posteriormente disponibilizados aos diferentes jogos que fazem parte da família de produtos.

A proposta apresentada neste trabalho explora justamente essa possibilidade. Os mecanismos de reconhecimento de voz e síntese de fala são integrados à infraestrutura da AsKME, enquanto os comandos reconhecidos são convertidos para as mesmas entradas utilizadas pela lógica original dos jogos. Dessa forma, procura-se separar a modalidade de interação da lógica do jogo, permitindo que a interação por voz seja incorporada sem a necessidade de redefinir individualmente as regras e funcionalidades de cada *quiz*.

Essa abordagem também estabelece uma relação direta entre os conceitos discutidos nesta seção: a acessibilidade em jogos fornece a motivação para disponibilizar modalidades alternativas de interação; as interfaces de voz oferecem uma possibilidade de entrada e saída complementar; a Web Speech API fornece mecanismos para sua implementação em aplicações web; e a abordagem de Linha de Produto de Software oferece o suporte arquitetural para reutilizar esses mecanismos em diferentes jogos da plataforma AsKME.

3. Metodologia

O desenvolvimento do sistema de acessibilidade foi conduzido por meio de uma abordagem incremental, organizada em ciclos iterativos de implementação, teste e refinamento contínuo. A metodologia combinou análise estrutural de código, integração de tecnologias e experimentação prática, com o objetivo de garantir compatibilidade, desempenho e usabilidade na plataforma AsKME, voltada ao desenvolvimento de jogos educativos interativos.

Inicialmente, foi realizada uma análise da arquitetura interna da plataforma AsKME e de seu cliente web, desenvolvido com base no motor de jogo *Phaser*[4] . Essa etapa permitiu mapear as variáveis e funções responsáveis pelas interações de entrada e saída de dados, viabilizando a adaptação do sistema originalmente baseado em teclado para suportar comandos de voz. A variável de entrada foi estendida para armazenar também comandos processados por fala, de modo que expressões como “opção dois” fossem convertidas para “2” e tratadas como entradas equivalentes às digitadas, mantendo a compatibilidade com a lógica original dos jogos. Adicionalmente, o estado atual do jogo

[4]`https://github.com/phaserjs/phaser`

passou a condicionar os comandos válidos em cada momento, contribuindo para maior precisão no reconhecimento.

Paralelamente, foi implementado o recurso de leitura de tela utilizando a API *SpeechSynthesis*, responsável por converter automaticamente os textos exibidos na interface em áudio narrado. Esse mecanismo foi integrado ao fluxo da aplicação, permitindo que instruções, perguntas, alternativas e *feedbacks* fossem vocalizados em tempo real. A variável de mensagens da interface passou a alimentar diretamente esse recurso, garantindo acesso contínuo às informações por usuários com deficiência visual.



Figura 1. Fluxos de entrada e saída da arquitetura híbrida produzida.

O reconhecimento de voz foi incorporado por meio da *Web Speech API*, utilizando a interface *webkitSpeechRecognition*, configurada para o idioma português brasileiro, com suporte a resultados parciais e operação em modo contínuo. Esse modo permite a escuta permanente do usuário, capturando comandos em tempo real sem necessidade de reativação manual, o que torna a interação mais fluida e natural. Entretanto, devido às limitações de suporte da *Web Speech API* em diferentes navegadores, foi adotada uma arquitetura híbrida, através de um módulo *backend* em *Python* utilizando o *framework FastAPI*, integrado à biblioteca *SpeechRecognition*[5] e ao serviço *Google Cloud Speech-to-Text*. Nesse mecanismo de *fallback*, o *frontend* envia trechos de áudio em blocos para o servidor, que realiza a transcrição e retorna os resultados ao cliente, garantindo o funcionamento do sistema em ambientes sem suporte nativo e com maior latência. A Figura 1 ilustra os fluxos de entrada e saída trabalhados pelas tecnologias da arquitetura híbrida produzida.

Um elemento central da solução é o mecanismo de escuta contínua, estruturado em ciclos de vigília, análise e comando. No estado de vigília, o sistema captura trechos curtos de áudio e verifica a presença de palavras-chave. Ao detectar uma palavra-chave, entra no

[5] `https://pypi.org/project/SpeechRecognition/`

modo de comando, suspendendo a detecção contínua para focar na captura da instrução principal. Após a execução ou após um tempo limite, o sistema retorna automaticamente ao estado de vigília (Figura 2). Em navegadores compatíveis, esse processo ocorre localmente e em tempo real, com reinicialização automática em caso de interrupções. No modo *fallback*, uma *thread* dedicada processa blocos de áudio de curta duração (entre 1 e 4 segundos), reproduzindo o mesmo comportamento de forma compatível.



Figura 2. Estados de transição do mecanismo de escuta contínua.

Adicionalmente, foram incorporados mecanismos de *feedback* multimodal para informar o estado do sistema. Para usuários com visão funcional, um ícone de microfone indica visualmente quando o reconhecimento de voz está ativo. Para usuários com deficiência visual, foram implementados *feedbacks* auditivos que sinalizam a ativação, desativação e reconhecimento de comandos por meio de mensagens e sons curtos, aumentando a clareza e a segurança durante a interação. Por fim, testes iterativos de usabilidade e acessibilidade foram realizados ao longo do desenvolvimento, orientando ajustes na precisão da transcrição de voz, na naturalidade da síntese de fala, na confiabilidade do mecanismo de escuta contínua e na clareza dos *feedbacks*. Cada componente foi validado individualmente antes de sua integração ao sistema completo, resultando em uma solução alinhada aos princípios de inclusão digital.

4. Resultados Obtidos

A versão final produzida[6] foi integrada com sucesso à plataforma AsKME, permitindo interação por voz e *feedback* auditivo sem comprometer o funcionamento original do sistema. Os usuários podem realizar ações como selecionar opções e navegar nos jogos utilizando apenas comandos de voz, enquanto a funcionalidade de leitura de tela garante que todas as informações relevantes sejam acessíveis por meio de áudio. A arquitetura híbrida mostrou-se eficaz ao garantir funcionamento em diferentes ambientes. Enquanto

[6]Disponível em: `https://github.com/[OMITIDO]`.

a *Web Speech API* oferece baixa latência em navegadores compatíveis, o módulo em nuvem com *Python/FastAPI* integrado ao *Google Cloud Speech-to-Text* assegura suporte em contextos sem suporte nativo, embora com latência ligeiramente maior. Essa abordagem tornou o sistema amplamente acessível, validando a proposta de inclusão digital.

A integração com as variáveis internas do AsKME funcionou conforme esperado, permitindo que os comandos de voz substituíssem plenamente as entradas de teclado e que a leitura de tela via *SpeechSynthesis* mantivesse os usuários informados em tempo real sobre cada atualização da interface. O mecanismo de escuta contínua, aliado às palavras-chave adotadas (e.g. "ok google", "alexa", "escutar" e "ativar") e aos *feedbacks* auditivos, trouxe confiabilidade ao sistema permitindo que usuários com deficiência visual consigam navegar pelos menus e interagir com o jogo de forma autônoma, eliminando a tentativa de ativações manuais frequentes.

Como exemplo de funcionamento, a tela inicial do jogo "*GuessMyNumberBot*" (Figura 3) mostra a operação do sistema, incluindo no rodapé os botões visuais para ativação e desativação da leitura de tela e da transcrição de voz que indicam o estado ativo ou inativo dessas funcionalidades. Quando o sistema reconhece uma palavra-chave, um ícone de microfone verde sinaliza que o microfone está em modo de captação de comandos. Esse recurso visual, em conjunto com o feedback sonoro, reforça a percepção do usuário sobre o funcionamento do sistema e evita inseguranças quanto à captura de comandos. Nessa mesma tela, o usuário pode iniciar a partida falando "Opção S", visualizar pontuações com "Opção H", acessar informações com "Opção A" ou sair com "Opção X", sendo que as instruções de voz são reconhecidas, interpretadas e traduzidas para a variável de entrada, simulando exatamente o mesmo fluxo da interação por teclado. Quando o jogador seleciona uma opção de jogo, a interação por voz continua ativa, permitindo que as escolhas sejam realizadas sem o uso do teclado.

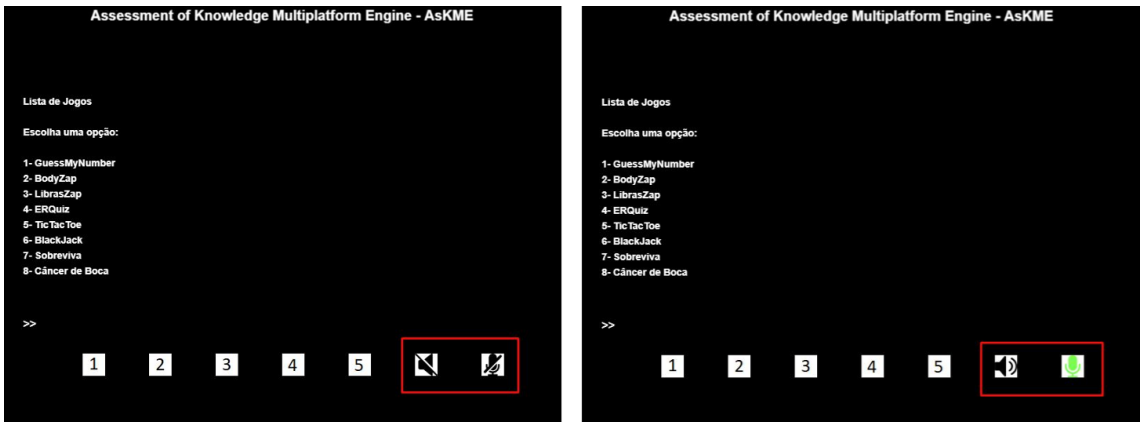


Figura 3. Tela inicial do jogo "GuessMyNumberBot" no AsKME inclusivo ilustrando uma transição de modo inativo para o modo de captação de comandos após o reconhecimento de uma palavra-chave.

5. Conclusões e Trabalhos Futuros

Este trabalho apresentou uma solução de acessibilidade baseada em reconhecimento de voz e leitura de tela integrada à plataforma AsKME, evidenciando a viabilidade de incorporar tecnologias assistivas em jogos estilo quiz. A adoção de uma arquitetura

híbrida, combinando a *Web Speech API* em navegadores compatíveis e uma API em *Python/FastAPI* integrada ao *Google Cloud Speech-to-Text* como *fallback*, mostrou-se eficaz para garantir ampla compatibilidade e desempenho consistente no sentido de promover maior autonomia para usuários com deficiência visual ou motora.

Além de contribuir diretamente para a acessibilidade, a solução desenvolvida fortalece o papel das tecnologias assistivas no contexto de jogos estilo quiz e na inclusão digital. Para tal, a solução proposta introduziu um mecanismo de escuta contínua baseado em palavras-chave e estados de interação, permitindo que comandos sejam processados sem a necessidade de acionamentos manuais frequentes. Assim, os comandos reconhecidos são convertidos para a mesma representação lógica utilizada pelas entradas convencionais, permitindo preservar a lógica original dos jogos estilo quiz desenvolvidos.

A integração com a AsKME também constitui uma característica importante da proposta, pois permite incorporar os mecanismos de acessibilidade à infraestrutura compartilhada de uma plataforma baseada em *features*, favorecendo sua reutilização em diferentes jogos estilo *quiz*. Trata-se de uma abordagem que reduz a necessidade de implementar individualmente recursos semelhantes em cada jogo e possibilita a produção em maior escala de experiências acessíveis. Dessa forma, a AsKME apresenta-se como uma alternativa alinhada aos princípios de inclusão e acessibilidade, ao possibilitar a interação com os jogos por meio de comandos de voz e *feedback* auditivo.

Os testes funcionais realizados demonstraram a viabilidade técnica da abordagem e sua integração com o jogo "GuessMyNumberBot". Entretanto, a avaliação atual ainda não fornece evidências quantitativas suficientes para afirmar a efetividade da solução em termos de taxa de reconhecimento, latência, compatibilidade entre navegadores ou melhoria da experiência de usuários com deficiência. Assim, a principal conclusão deste trabalho é que a arquitetura proposta constitui uma alternativa tecnicamente viável para incorporar interação por voz e feedback auditivo a jogos estilo quiz produzidos na AsKME. A validação de sua efetividade como recurso de acessibilidade, entretanto, requer avaliações técnicas quantitativas mais abrangentes e, principalmente, estudos com usuários reais com deficiência visual ou motora.

Por fim, apesar dos avanços, ainda persistem desafios relacionados à latência em soluções baseadas em nuvem e à heterogeneidade de suporte entre navegadores. Como trabalhos futuros, destacam-se a realização de avaliações empíricas com usuários, a medição sistemática da taxa de reconhecimento e da latência, a avaliação em diferentes navegadores e condições ambientais, a comparação com a interação por teclado e mouse, a análise do esforço de integração em diferentes jogos, a otimização do desempenho do sistema, a investigação de mecanismos de reconhecimento local e de novas formas de interação acessível, bem como a avaliação da compatibilidade com leitores de tela e demais tecnologias assistivas. Esses desdobramentos podem contribuir para aprimorar a experiência dos usuários e ampliar o alcance da plataforma, fortalecendo sua contribuição para a inclusão digital.

Referências

Cairns, P. and Cox, A. L. (2008). *Research methods for human-computer interaction*, volume 10. Cambridge University Press Cambridge.

Clements, P. and Northrop, L. (2002). *Software product lines*, volume 1. Addison-wesley Boston.

Deterding, S., Dixon, D., Khaled, R., and Nacke, L. (2011). From game design elements to gamefulness: defining"gamification". In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, pages 9–15.

Gee, J. P. (2003). What video games have to teach us about learning and literacy. *Computers in entertainment (CIE)*, 1(1):20–20.

Hamari, J., Koivisto, J., and Sarsa, H. (2014). Does gamification work?–a literature review of empirical studies on gamification. In *2014 47th Hawaii international conference on system sciences*, pages 3025–3034. Ieee.

Pohl, K., Böckle, G., and Van Der Linden, F. (2005). *Software product line engineering: foundations, principles, and techniques*, volume 1. Springer.

Porcheron, M., Fischer, J. E., Reeves, S., and Sharples, S. (2018). Voice interfaces in everyday life. In *proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–12.

Prensky, M. (2003). Digital game-based learning. *Computers in entertainment (CIE)*, 1(1):21–21.

Sarinho, V. T., de Azevedo, G. S., and Boaventura, F. M. (2018). Askme: A feature-based approach to develop multiplatform quiz games. In *2018 17th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, pages 38–3809. IEEE.

Schwartz, R., Monge Roffarello, A., and De Russis, L. (2025). Bridging digital wellbeing and accessibility: An analysis of the w3c web content accessibility guidelines. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–7.

Yuan, B., Folmer, E., and Harris Jr, F. C. (2011). Game accessibility: a survey. *Universal Access in the information Society*, 10(1):81–100.