DESENVOLVIMENTO WEB DE UM CONVERSOR DE TEXTO EM ÁUDIO: AUDIFLY

Pedro Paulo Macedo de Oliveira[1]; Arthur Pereira Morais[2];
Raul Sérgio Reis Rezende[3]

[1] [2] [3] *Universidade de Uberaba – UNIUBE*

E-mails: pedropaulomacedo.oliveira@gmail.com; raul.rezende@uniube.br

Resumo
O presente trabalho descreve o desenvolvimento do Audifly, um sistema web voltado à conversão de texto em áudio destinado a ampliar a acessibilidade digital. O objetivo principal foi projetar e implementar uma aplicação que possibilite a transformação de conteúdos textuais em fala natural diretamente no navegador, sem necessidade de softwares adicionais. A solução adotou uma arquitetura cliente-servidor baseada em PHP para o *back-end*, com banco de dados MySQL, e uma interface responsiva construída em HTML5, CSS3 e JavaScript. A síntese de voz foi realizada por meio da API Web Speech Synthesis, permite seleção de idioma, ajuste de velocidade e controle de volume. Foram realizados testes de compatibilidade e desempenho em Google Chrome, Mozilla Firefox e Microsoft Edge, avaliando tempo de conversão, estabilidade e qualidade perceptiva da fala. Os resultados indicam que o Audifly converte textos curtos em menos de dois segundos e mantém estabilidade em diferentes plataformas, embora haja variações na naturalidade da voz devido à dependência das vozes nativas dos navegadores. Conclui-se que o Audifly constitui uma solução prática e escalável para acessibilidade web, com potencial de integração a serviços de síntese avançada para melhorias futuras.

Palavras-chave: acessibilidade; conversão de texto em áudio; web speech api.

Introdução

A rápida evolução das tecnologias digitais têm transformado profundamente a forma como as pessoas interagem com informações e sistemas computacionais. Com o advento da Web 3.0, da inteligência artificial (IA) e da computação em nuvem, cresce a necessidade de soluções que aliem inovação, acessibilidade e praticidade. Nesse cenário, a comunicação homem-máquina vem sendo aprimorada por meio de interfaces naturais, incluindo reconhecimento de fala, visão computacional e síntese de voz (Text-to-Speech – TTS). Essas tecnologias são fundamentais para garantir o acesso igualitário à informação, especialmente para pessoas com deficiência visual ou limitações cognitivas, cumprindo o princípio do Desenho Universal (Silva *et al*., 2021).

A conversão de texto em áudio é uma das aplicações mais relevantes dessa tendência. Ela permite que conteúdos textuais sejam transformados em fala natural, possibilitando uma experiência auditiva e inclusiva. A crescente demanda por acessibilidade digital impulsionou o desenvolvimento de soluções como o Audifly, um sistema *web* concebido para realizar conversão de texto em áudio de forma simples e direta. O grande desafio técnico atual em TTS é equilibrar a alta qualidade de voz dos modelos neurais (processados no lado do servidor) com a baixa latência exigida por aplicações *web* interativas. O Audifly propõe enfrentar este desafio utilizando uma arquitetura *client-side*.

O objetivo principal deste artigo é apresentar o desenvolvimento e a avaliação de desempenho do sistema *web* “Audifly”, um conversor de texto em áudio de baixa latência. O trabalho foca em projetar e validar uma solução acessível que utilize a API Web Speech

Synthesis nativa, integrada a um *back-end* estruturado em PHP no padrão MVC, para promover acessibilidade e inclusão digital com prioridade na velocidade de resposta.

As principais contribuições deste trabalho para a área de desenvolvimento de *software* e acessibilidade são:

- A validação da arquitetura *client-side* com a Web Speech API como uma alternativa eficaz e de custo zero para o desenvolvimento de soluções TTS que exigem baixa latência, em oposição às soluções complexas baseadas em *cloud computing*.
- O desenvolvimento de um sistema funcional e escalável baseado no padrão MVC (PHP), que facilita a manutenção e permite futuras integrações com módulos de alta fidelidade de voz.
- A demonstração da compatibilidade e estabilidade da Web Speech API em navegadores *mainstream* (Chrome, Firefox e Edge), fornecendo dados quantitativos sobre o tempo de conversão (latência) para embasar futuras escolhas arquitetônicas.

O artigo está estruturado da seguinte forma: a Seção 2 apresenta o referencial teórico, discutindo conceitos e tecnologias relacionadas à síntese de voz e acessibilidade digital, incluindo a legislação pertinente; a Seção 3 detalha a metodologia adotada, descrevendo a arquitetura *cliente-servidor* e o padrão MVC; a Seção 4 apresenta os resultados obtidos nos testes de desempenho; a Seção 5 aprofunda a discussão dos achados, confrontando latência e qualidade; e, por fim, a Seção 6 apresenta as conclusões e as perspectivas para trabalhos futuros.

Acessibilidade Digital e Normativas

A acessibilidade digital refere-se à capacidade de todos os usuários, independentemente de suas habilidades ou deficiências, perceberem, entenderem, navegarem e interagirem com a web. No Brasil, normativas como a Lei nº 13.146/2015 (Estatuto da Pessoa com Deficiência) promovem a inclusão. O desenvolvimento de soluções como o Audifly, ao fornecer uma alternativa auditiva ao texto, atua como uma tecnologia assistiva essencial (Saouza *et al*., 2020).

O Text-to-Speech (TTS), nesse contexto, permite que o conteúdo textual seja assimilado por usuários com deficiência visual ou dificuldades de leitura, garantindo a equidade no acesso à informação e aos serviços digitais.

A implementação da interface do Audifly buscou aderência às diretrizes de interface, como as definidas pelas Web Content Accessibility Guidelines (WCAG) 2.1 (W3C, 2018).

O Text-to-Speech (TTS), nesse contexto, permite que o conteúdo textual seja assimilado por usuários com deficiência visual ou dificuldades de leitura, garantindo a equidade no acesso à informação e aos serviços digitais.

Fundamentos da Síntese de Voz (Text-to-Speech)

A tecnologia TTS é o processo de transformar texto escrito em fala. Nos últimos anos, houve um avanço significativo com o uso de modelos neurais, permitindo vozes cada vez mais naturais.

A síntese neural (Neural TTS) utiliza redes neurais profundas para gerar áudio de alta fidelidade, produzindo vozes que são quase indistinguíveis da fala humana natural. A complexidade computacional desses modelos (ex: WaveNet - Van Den Oord *et al*., 2016) geralmente exige poder de processamento significativo, o que é tipicamente realizado no

lado do servidor (Server-Side Rendering – SSR), introduzindo latência devido à rede. O aprimoramento do estilo e expressividade da voz em idiomas específicos, como o Português Brasileiro, também é um desafio ativo na pesquisa de TTS (Tunnermann, 2021).

Web Speech API e a Arquitetura Client-Side

No contexto *web*, a API Web Speech Synthesis do W3C possibilita que navegadores modernos executem a conversão de texto em áudio diretamente no cliente. Essa abordagem (*Client-Side Rendering* – CSR) torna a tecnologia mais acessível e leve.

A escolha da Web Speech API é um posicionamento que prioriza a baixa latência e a acessibilidade imediata, pois utiliza os recursos de TTS nativos do sistema operacional, minimizando a latência e a carga do servidor. O *trade-off* é que a qualidade da voz fica atrelada às implementações do navegador, o que pode causar variação na naturalidade da fala.

MATERIAIS E MÉTODOS

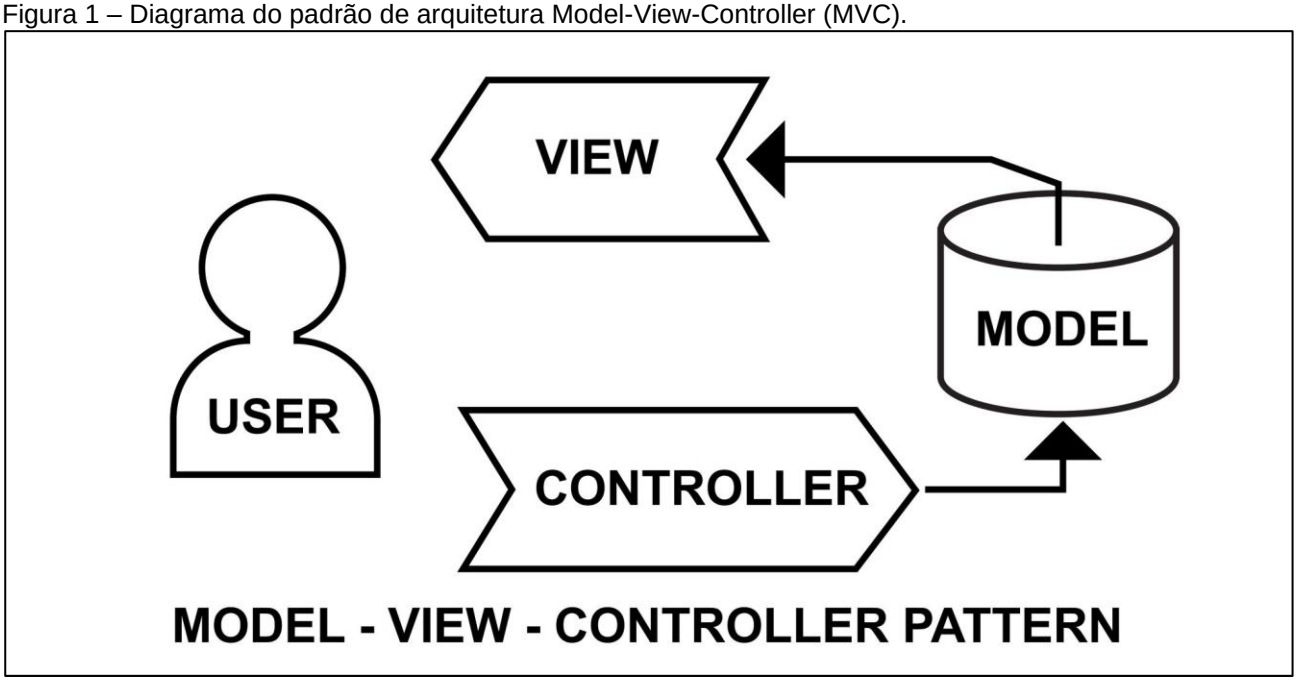
A metodologia adotada para o desenvolvimento do sistema Audifly foi de natureza pragmática e experimental, focada na implementação de um protótipo funcional e na sua validação através de métricas de desempenho em tempo real. O processo foi guiado por princípios da Engenharia de Software (ES), utilizando o Modelo Iterativo e Incremental.

Fundamentação Teórica das Escolhas: Esta subseção estabelece as bases conceituais que justificaram as decisões tecnológicas e arquiteturais do Audifly.

Arquitetura e Padrão de Projeto: A natureza da aplicação, que prioriza a baixa latência e a acessibilidade imediata, guiou a escolha por uma Arquitetura Cliente-Servidor Híbrida.

- Decisão Client-Side (CSR): A adoção da Web Speech Synthesis API para a síntese de voz é um posicionamento que prioriza o desempenho percebido. Esta abordagem de processamento no cliente (CSR) elimina a latência de comunicação com serviços de nuvem de alta fidelidade, conforme é comparado em estudos sobre APIs de voz.
- Padrão MVC: O *back-end* foi rigidamente estruturado no padrão Model-View-Controller (MVC)

Figura 1 – Diagrama do padrão de arquitetura Model-View-Controller (MVC).



Fonte: Elaborada pelos autores (2025).

Reconhecido por separar a lógica de negócio (Model), a interface do usuário (View) e o gerenciamento de requisições (Controller). A escolha do MVC, justifica-se pela necessidade de garantir a modularização, manutenção e escalabilidade do sistema.

Métricas de Desempenho: As métricas utilizadas (tempo de conversão, estabilidade e qualidade perceptiva) são cruciais para a validação de sistemas interativos. A medição da latência (*Time-to-First-Token*) é um fator determinante na usabilidade de tecnologias assistivas de fala.

Processo de Desenvolvimento e Métodos de ES: O desenvolvimento do Audifly seguiu o Modelo Iterativo e Incremental, adaptado para o desenvolvimento de protótipos funcionais.

Análise e Definição de Requisitos: Esta fase inicial aplicou métodos de Engenharia de Requisitos, estabelecendo:

1. Requisitos Funcionais: Conversão de texto em áudio no navegador, controle de parâmetros (velocidade, volume, idioma), persistência de logs (MySQL).
2. Requisitos Não-Funcionais: Baixa latência (tempo < 2.5s), compatibilidade *cross-browser*, interface responsiva (Acessibilidade - WCAG 2.1).

Projeto e *Design*: Com base nos requisitos, foi realizado o projeto detalhado da arquitetura, implementando a separação do padrão MVC. Foram desenhados o Esquema do Banco de Dados (Model) e o Layout da Interface (View), garantindo que o design estivesse centrado no usuário, conforme princípios da usabilidade.

Implementação e Codificação: Esta fase correspondeu à construção do sistema, onde o código foi dividido em camadas de forma modular.

- Ambiente de Desenvolvimento: Servidor Apache (XAMPP), Back-end em PHP 8 e persistência de dados em MySQL.

- Módulo *View* (HTML5/CSS3/JS): Desenvolvimento da interface e a integração-chave: o script JavaScript que utiliza o objeto `SpeechSynthesisUtterance` e a `speechSynthesis` (Web Speech API) para realizar a conversão diretamente no cliente.

Testes e Validação: A validação utilizou métodos de Testes de Caixa-Preta, focados na funcionalidade e desempenho do sistema do ponto de vista do usuário.

- Testes de Compatibilidade: Verificação da estabilidade em Google Chrome, Mozilla Firefox e Microsoft Edge.
- Testes de Desempenho: Medição da latência de conversão em ambiente controlado, utilizando textos de diferentes comprimentos.
- Avaliação Qualitativa: Inclusão da avaliação subjetiva da clareza da voz (Seção 4), um critério essencial para validação de tecnologias de voz.

Ambiente de Desenvolvimento e Arquitetura

O desenvolvimento do Audifly foi realizado em um ambiente local de desenvolvimento. A *stack* tecnológica empregada compreendeu:

- Servidor: Apache, operando dentro do pacote XAMPP.
- Back-end: Implementado em PHP 8. A aplicação é rigidamente estruturada seguindo o padrão arquitetural MVC (Model-View-Controller).
- Persistência de Dados: Utilizou-se o MySQL para o gerenciamento de dados. O banco de dados foi projetado para persistir dados essenciais como preferências de usuário e *logs* de conversão.
- *Front-end*: A interface cliente foi construída com HTML5, CSS3 e JavaScript (ES6), garantindo responsividade e ampla compatibilidade com diferentes dispositivos.

A arquitetura MVC foi adotada para separar a lógica de negócio *Model* e o gerenciamento de requisições *Controller* da interface do usuário *View* facilitando a manutenção e futuras integrações.

Implementação do Módulo de Síntese de Voz

O módulo de conversão de texto em áudio foi implementado integralmente no lado do cliente *Client-Side* utilizando a tecnologia nativa do navegador.

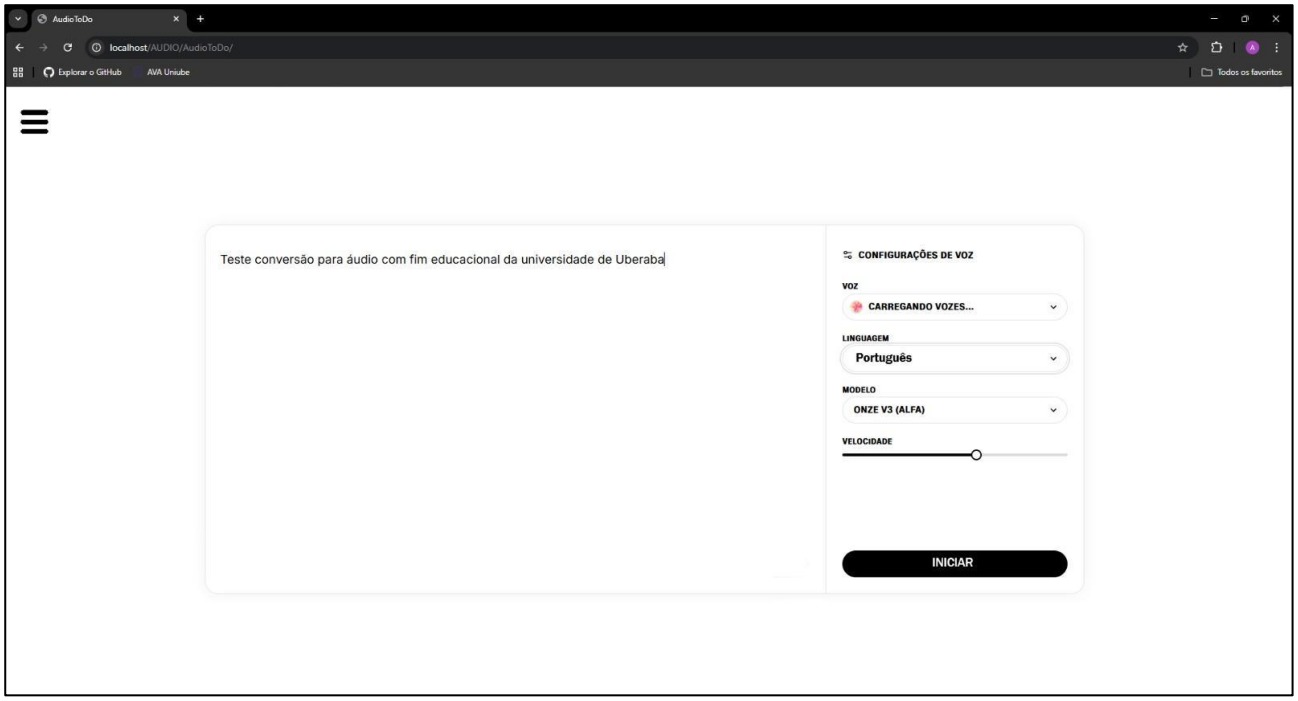
- **API Principal:** Emprega-se a **Web Speech Synthesis API** do W3C, que fornece os métodos nativos para a conversão de texto em áudio em navegadores compatíveis.
- **Configuração da Fala:** O código JavaScript é responsável por instanciar o objeto de fala e configurar parâmetros como seleção de idioma, ajuste de velocidade e controle de volume.
- **Procedimento de Conversão:** O módulo principal captura o texto inserido pelo usuário na *View*, envia o conteúdo para o mecanismo TTS do navegador e reproduz o áudio sintetizado.

Critérios e Procedimentos de Teste

A validação do sistema focou em métricas de desempenho e qualidade perceptiva. O ambiente de teste foi isolado para garantir a precisão da medição de desempenho:

- *Hardware* de Teste: As medições foram realizadas em uma máquina com processador Intel i5, 8 GB de memória RAM e sistema operacional Windows 10. A escolha do ambiente local e a conexão direta sem latência de rede garantiram que as métricas de tempo refletissem puramente o processamento *client-side*.
- Métricas de Desempenho:
  - ✓ (i) Medição da Latência: O principal foco foi o tempo de conversão, medido em segundos (s), desde o acionamento do comando até o início da reprodução do áudio (Time-to-First-Token). Foram utilizados diferentes tamanhos de texto (curto e longo) para avaliar a escalabilidade da latência.
  - ✓ (ii) Avaliação Qualitativa: Incluiu a avaliação da clareza da voz.
  - ✓ (iii) Compatibilidade: Foi verificada a estabilidade e funcionalidade do sistema em navegadores *mainstream*: Google Chrome, Mozilla Firefox e Microsoft Edge.
- Limitações Observadas: As restrições do modelo incluíram a variação na qualidade de voz observada entre os sistemas operacionais e navegadores, devido à dependência dos motores TTS nativos.

Figura 2 – Aplicação: desenvolvimento web de conversor de texto em áudio



Fonte: Elaborada pelos autores (2025).

Resultados (ou resultados esperados)

Os testes mostraram desempenho consistente do Audifly. Textos de aproximadamente 300 a 500 palavras apresentaram tempo médio de conversão entre 1,6 e 2,1 segundos no

Chrome e Edge, enquanto no Firefox observou-se variação próxima a 2,3 segundos. O Quadro 1 resume os resultados de média obtidos.

Quadro 1 – Resultados de desempenho por navegador.

Navegador	Tempo Médio (s)	Classificação da Qualidade
Google Chrome	1.8	Alta
Mozilla Firefox	2.3	Média
Microsoft Edge	1.9	Alta

Fonte: Elaborado pelos autores (2025).

Discussão

Observa-se que o Chrome apresentou o menor tempo de conversão e melhor naturalidade de voz, enquanto o Firefox apresentou leve atraso e timbre mais artificial. Essas diferenças se devem às variações nas implementações da API Web Speech Synthesis entre os navegadores. No geral, os resultados indicam que a solução é viável para uso educacional, corporativo e pessoal.

Durante a fase de testes, o sistema mostrou-se responsivo, funcionando em diferentes tamanhos de tela e dispositivos móveis. Além disso, a interface simples contribuiu para uma experiência de uso intuitiva, reforçando a importância do *design* centrado no usuário no desenvolvimento de aplicações acessíveis.

.

Conclusão

O presente artigo apresentou o desenvolvimento e a avaliação do Audifly, um sistema web projetado para realizar a conversão de texto em áudio utilizando tecnologias open source e recursos nativos dos navegadores modernos. Os resultados obtidos demonstraram que o sistema é funcional, acessível e de fácil utilização, atendendo aos objetivos propostos de promover acessibilidade digital e inclusão social.

As principais contribuições deste trabalho incluem a integração de tecnologias web com recursos de síntese de voz, a validação da API Web Speech Synthesis em diferentes navegadores e a criação de uma ferramenta prática para uso cotidiano. Como limitações, identificou-se a dependência das vozes nativas do sistema operacional e a falta de padronização entre navegadores. Como trabalhos futuros, propõe-se a integração com APIs baseadas em inteligência artificial, como Google Cloud TTS e Amazon Polly, para aprimorar a naturalidade e a expressividade das vozes geradas.

.

Referências

BRASIL. Lei n° 13.146, de 6 de julho de 2015. Institui a Lei Brasileira de Inclusão da Pessoa com Deficiência (Estatuto da Pessoa com Deficiência). Brasília: Presidência da República, 2015. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13146.htm.

CENTRO DE ESTUDOS SOBRE AS TECNOLOGIAS DA INFORMAÇÃO E DA COMUNICAÇÃO (CETIC.br). Cartilha de Acessibilidade na Web: Fascículo IV. [S.l.]:

Comitê Gestor da Internet no Brasil (CGI.br), 2022. Disponível em: https://nic.br/media/docs/publicacoes/13/20220221104917/cartilha-acessibilidade-web-fasciculo-IV.pdf.

GAMMA, E. *et al.* Design Patterns: Elements of Reusable Object-Oriented Software. Reading: Addison-Wesley, 1994.

SHEN, J. et al. Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. arXiv preprint, arXiv:1802.06006, 2018.

SILVA, J. A.; VIEIRA, M. F.; SANTOS, P. M. Desenvolvimento de um Sistema de Apoio a Leitura Baseado em Síntese de Fala para Usuários com Deficiência Visual. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE), 32., 2021, Natal. Anais... Natal: Sociedade Brasileira de Computação (SBC), 2021. p. 1-10.

SOUZA, A. C. B.; MARTINS, G. R. S.; ALVES, P. R. de S. Acessibilidade Web e a Tecnologia Assistiva para Leitura de Conteúdo Textual em um Projeto de Extensão. Interfaces Tecnológicas, Quatro Barras, v. 1, n. 2, p. 111-120, 2020. Disponível em: https://revista.fatectq.edu.br/interfacetecnologica/pt_BR/article/view/1641.

VAN DEN OORD, A. et al. WaveNet: A Generative Model for Raw Audio. arXiv preprint, arXiv:1609.03499, 2016.

Web Content Accessibility Guidelines (WCAG) 2.1. W3C Recommendation, 2018. Disponível em: https://www.w3.org/TR/WCAG21/.

TUNNERMANN, Daniel. Controle de Estilo na Síntese de Voz em Português Brasileiro usando Redes Neurais Profundas. 2021. Dissertação (Mestrado em Ciência da Computação) - Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Goiás (UFG), Goiânia, 2021. Disponível em: https://repositorio.bc.ufg.br/tedeserver/api/core/bitstreams/a54f3de1-c41b-4527-b4f7-1e1b2690c8fe/content.