

Trabalho 1

Camila Costa Durante

Felipe Natan Zanqueta Macaubas

5 de maio de 2025

1 Introdução e Objetivos da Aplicação

Visando o desenvolvimento de uma aplicação web interativa voltada para o domínio da Educação, utilizando HTML, CSS, JavaScript e um modelo de Inteligência Artificial pré-treinado, foi criada a aplicação "Performance de Centavos": uma página web gamificada que utiliza o reconhecimento de voz por meio do modelo *speech-commands* do TensorFlow.js, permitindo que crianças pratiquem operações matemáticas básicas de forma lúdica e dinâmica, além de treinarem um vocabulário básico porém essencial do idioma inglês.

Os objetivos do trabalho consistem em:

- Desenvolver uma aplicação web educativa que auxilie crianças na prática de operações matemáticas simples (adição, subtração, multiplicação e divisão);
- Integrar o modelo de reconhecimento de voz, por meio do TensorFlow.js, para permitir respostas faladas utilizando comandos em inglês ("up", "down", "left", "right");
- Estimular o aprendizado simultâneo de matemática e vocabulário básico em inglês, por meio de uma abordagem lúdica e interativa;
- Oferecer *feedbacks* imediatos (visuais e sonoros) com o objetivo de sinalizar acertos e erros de forma clara;
- Criar um ambiente digital amigável, funcional e atrativo para o público infantil, respeitando princípios de usabilidade e acessibilidade.

2 Descrição Técnica da Implementação

Foram criados quatros arquivos:

- **home.html**: responsável pelo conteúdo disponível na página inicial do site;
- **jogo.html**: arquivo que define a interface da página do jogo;
- **script.js**: contém as funções responsáveis pela lógica do jogo;
- **styles.css**: define o estilo visual das páginas em HTML, com foco em legibilidade e atratividade infantil.

No arquivo **home.html** foram inseridos os principais elementos da página inicial da aplicação: a imagem do logotipo do site, um botão que redireciona o usuário para a próxima etapa (o jogo), uma `<div>` contendo uma lista com instruções de uso, explicando de forma simples e acessível como interagir com a aplicação, além dos áudios utilizados na interface. Também

foi implementado um pequeno *script* responsável por tocar o áudio de fundo da página inicial. Quando o usuário clica no botão "Jogar", esse áudio de fundo é interrompido, um segundo áudio de saudação é reproduzido e, após seu término, o usuário é automaticamente redirecionado para a página do jogo.

Já no arquivo `jogo.html`, foram criadas diversas `<div>`s com finalidades específicas: uma para exibir a equação matemática a ser resolvida, quatro para apresentar as possíveis respostas — posicionadas em forma de cruz (acima, abaixo, à esquerda e à direita da tela) — e outra para fornecer um *feedback* visual ao usuário, indicando se a resposta foi correta ou incorreta. Também foram incluídos elementos de áudio responsáveis por fornecer *feedbacks* auditivos, bem como a imagem do logotipo para manter a identidade visual do site.

No arquivo `script.js`, foram implementadas as principais funções responsáveis pela lógica do jogo. A função `gerarPergunta()` é responsável por gerar dois números aleatórios entre 1 e 10 e selecionar aleatoriamente um operador matemático (adição, subtração, multiplicação ou divisão). Com base no operador sorteado, a operação é realizada e o resultado é armazenado. Cuidados foram tomados para evitar resultados negativos em subtrações e divisões não exatas.

Após o cálculo, a equação é exibida na tela e três respostas incorretas são geradas. Para isso, é sorteado um valor aleatório entre 1 e 5, que é somado ou subtraído do resultado correto de forma aleatória. Se o valor gerado for diferente da resposta correta e maior ou igual a zero, ele é adicionado à lista de alternativas incorretas.

Em seguida, a posição da resposta correta é aleatoriamente definida entre as quatro disponíveis (*up*, *down*, *left* e *right*), e as respostas incorretas são alocadas nas demais posições.

Já a função `"iniciarReconhecimento()"` carrega o modelo pré-treinado `"speech-commands"` do TensorFlow.js, e ativa para escutar o usuário pelo microfone. O resultado do reconhecimento de voz é primeiramente extraído os escores de probabilidade de cada comando de voz reconhecido, depois é salvo a posição do comando com maior probabilidade reconhecido pelo modelo, por fim, o comando é salvo baseado na posição da maior probabilidade no vetor de comandos existentes. Com o comando provável reconhecido pelo modelo, é verificado se o comando é um entre as opções (*up*, *down*, *left* e *right*), exibido para o usuário qual o comando o modelo entendeu, e então ocorre a verificação se a posição entendida pelo modelo, e falada pelo usuário corresponde a posição da alternativa correta. Se a posição que o usuário disse corresponde a posição da alternativa correta, é exibido e tocado o *feedback* de que o usuário acertou, e então a próxima pergunta é feita. Se o usuário falou uma posição diferente da posição da alternativa correta, os *feedbacks* visuais e auditivos são dados mostrando que o usuário errou a resposta e o jogo é encerrado, voltando para a página inicial.

Por fim, a função `iniciarReconhecimento()` é responsável por carregar o modelo pré-treinado `speech-commands` do TensorFlow.js e ativar o microfone para captar a fala do usuário. Ao receber um comando de voz, o modelo retorna os escores de probabilidade para cada palavra reconhecida. Em seguida, é identificada a posição (índice) da palavra com maior probabilidade, e o comando correspondente é extraído com base nesse índice.

Com o comando mais provável identificado, o programa verifica se ele corresponde a uma das opções válidas (*up*, *down*, *left* ou *right*). Caso corresponda, o comando é exibido para o usuário e comparado com a posição da resposta correta. Se o comando falado for a resposta certa, o sistema fornece *feedback* visual e sonoro de acerto e gera uma nova pergunta. Caso contrário, os *feedbacks* de erro são apresentados, e o jogo é encerrado, redirecionando o usuário para a página inicial.

No arquivo `styles.css`, foram definidos estilos visuais aplicados tanto à página inicial quanto à página do jogo. Para garantir uma padronização e gerar atratividade, as colorações utilizadas foram extraídas da paleta do logotipo da aplicação, reforçando a identidade visual

do projeto.

Os tamanhos das fontes foram ajustados para proporcionar clareza na leitura, com destaque na coloração do texto em contraste com o fundo, garantindo acessibilidade e legibilidade. Na página do jogo, as áreas de resposta foram dispostas em formato de cruz, com o logotipo centralizado, reforçando a consistência visual e a orientação espacial da atividade.

O *feedback* visual, responsável por indicar acertos ou erros, foi posicionado logo abaixo do logotipo, facilitando sua visualização imediata após cada interação. A equação matemática foi colocada na parte superior da tela, utilizando fonte em tamanho ampliado e em negrito, assegurando total clareza durante o exercício.

3 Tecnologias utilizadas

A aplicação foi desenvolvida utilizando tecnologias web, com ênfase na acessibilidade e na interatividade por meio de comandos de voz. As principais tecnologias utilizadas incluem:

- **HTML:** Estruturação da interface e aplicação.
- **CSS:** Estilização e responsividade de elementos gráficos.
- **JavaScript:** Lógica do jogo, manipulação de eventos e integração com bibliotecas externas.
- **TensorFlow.js (TENSORFLOW TEAM, 2023):** Biblioteca de aprendizado de máquina executada no navegador, utilizada para o reconhecimento de voz em tempo real.

Dentre os modelos oferecidos pelo TensorFlow.js, foi utilizado o pacote *speech-commands*, que fornece um modelo pré-treinado capaz de identificar comandos simples de voz diretamente no navegador. Este modelo opera com base em Transformadas Rápidas de Fourier via *BROWSER_FFT*, permitindo que a análise do espectrograma do áudio seja realizado localmente, sem a necessidade de servidores externos. A escolha por tecnologias que rodam diretamente no navegador proporciona maior acessibilidade, privacidade do usuário e facilidade de uso, eliminando a necessidade de instalações adicionais.

4 Instruções de Execução

Execução Local

Para a execução da aplicação localmente é necessário fazer o download de todos os arquivos do projeto, isso inclui todos os arquivos *html*, *css*, *js* assim como os arquivos de imagens e áudios que são utilizados na aplicação. Também é necessário utilizar algum tipo de navegador com acesso a internet, uma vez que estamos utilizando bibliotecas hospedadas em redes de distribuição de conteúdos, como o *Tensorflow.js*. Para a inicialização da aplicação, basta abrir o arquivo *home.html* no navegador escolhido.

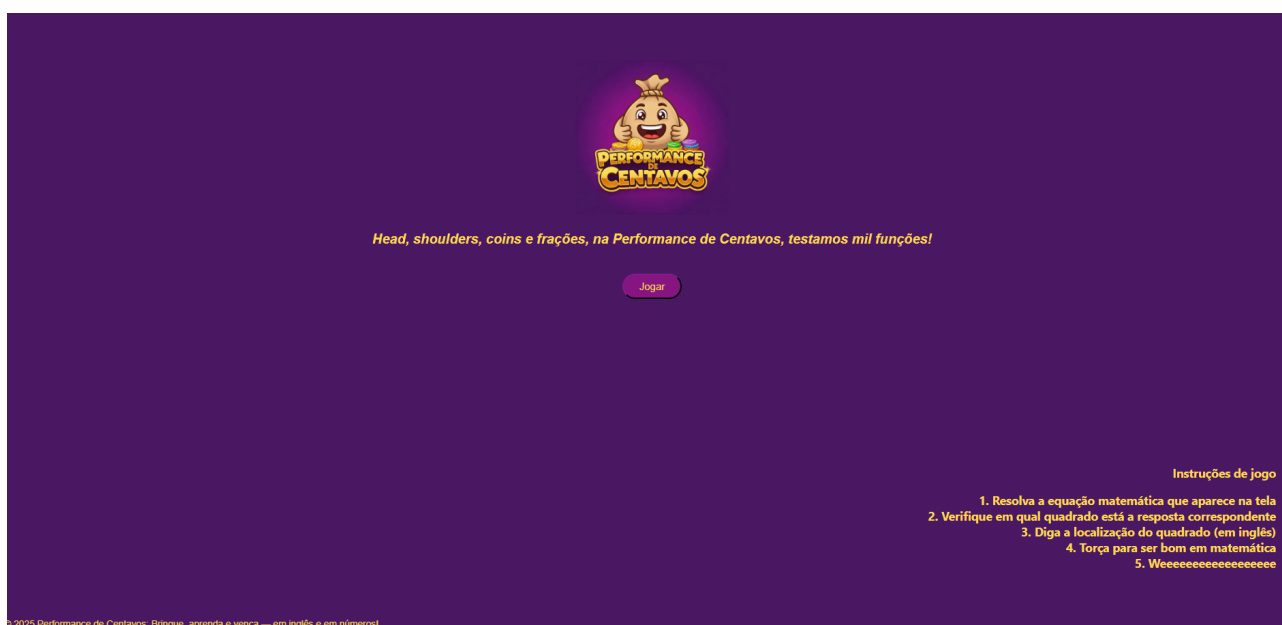
Modo de Utilização

Ao carregar a página inicial, o usuário deve clicar no botão "Jogar" para acessar o ambiente interativo, a aplicação irá apresentar uma equação matemática e quatro alternativas de respostas distribuídas na tela. Após realizar a equação matemática o usuário deve identificar

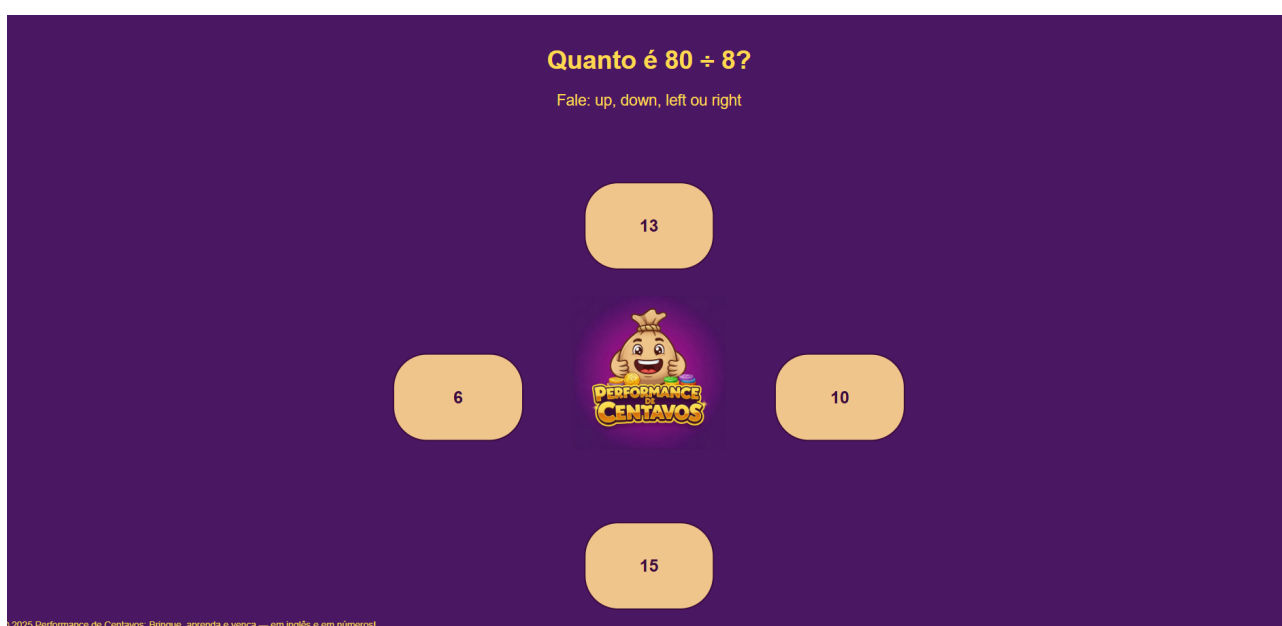
em qual posição está a resposta correta e verbalizar a direção em inglês. Em caso de acerto, uma nova pergunta será gerada automaticamente, caso a resposta seja incorreta, o sistema irá direcionar o usuário para a página inicial após alguns segundos, em ambos os casos há uma confirmação visual e auditiva sobre o acerto ou erro da resposta.

5 Adicionais

Imagens da Aplicação



Home



Jogo

Referências

TENSORFLOW TEAM. @tensorflow-models/speech-commands. Internet: GitHub, 2023. Disponível em:
<<https://github.com/tensorflow/tfjs-models/tree/master/speech-commands>>.