

# Desenvolvimento de uma Inteligência Artificial Tutora de Inglês

EP - Programação para Web

Arthur Rosa dos Santos

Bruno Queiroz Castro

2025.1

## Resumo

Este trabalho apresenta o desenvolvimento de uma aplicação web que integra uma inteligência artificial com o objetivo de atuar como tutora de inglês. A proposta envolve a criação de uma interface interativa e acessível, onde o usuário pode praticar leitura, escrita e compreensão do idioma por meio de interações com a IA. A aplicação será desenvolvida utilizando tecnologias web, o projeto visa explorar o potencial de sistemas inteligentes no apoio educacional, promovendo o aprendizado de forma autônoma, eficaz e motivadora.

## Introdução

### Motivação

O domínio da língua inglesa é uma habilidade fundamental em diversas áreas do conhecimento e do mercado de trabalho. No entanto, muitos estudantes enfrentam dificuldades de acesso a métodos eficazes de aprendizagem, especialmente em ambientes autodidatas. Nesse contexto, a utilização de uma inteligência artificial como tutora surge como uma solução inovadora, proporcionando um acompanhamento personalizado, constante e adaptativo. A proposta deste trabalho é desenvolver uma ferramenta interativa, acessível via web, que atue como um assistente virtual voltado ao ensino do inglês, promovendo autonomia e engajamento no processo de aprendizado.

### Problema

O aprendizado de inglês como segunda língua ainda apresenta desafios significativos, especialmente para alunos autodidatas ou com recursos educacionais limitados. Muitos métodos tradicionais exigem acompanhamento constante de um professor ou uso de plataformas pagas, o que pode não ser acessível para todos. Além disso, as ferramentas disponíveis geralmente não oferecem respostas personalizadas em tempo real nem interatividade suficiente para manter o engajamento do usuário.

## Trabalhos Relacionados

Diversas plataformas e aplicativos vêm explorando o uso da inteligência artificial no ensino de línguas, como o Duolingo, que utiliza gamificação e algoritmos adaptativos para guiar o aluno. Ferramentas como o ChatGPT também são utilizadas de forma informal por estudantes para praticar conversação. No entanto, há uma lacuna na criação de soluções focadas exclusivamente em tutoria de inglês com orientação didática específica e integração direta com interfaces web personalizadas. Nosso projeto se diferencia ao aplicar uma IA treinada com foco educacional, acessível gratuitamente por navegador.

## Contribuição do Trabalho

A principal contribuição deste trabalho está no desenvolvimento de uma aplicação web funcional que integra uma inteligência artificial como tutora virtual de inglês, com interface intuitiva, backend seguro e comunicação fluida entre usuário e IA. A plataforma permite a prática do idioma de maneira natural, personalizada e acessível, utilizando tecnologias modernas e seguindo boas práticas de segurança. Além disso, o projeto serve como base didática para futuros aprimoramentos, como integração com reconhecimento de voz ou expansão para outros idiomas.

## Organização do Restante do Trabalho

Este artigo está organizado da seguinte maneira: na seção de **Fundamentação Teórica**, discutimos os conceitos essenciais sobre aprendizado de máquina, segurança web e tecnologias utilizadas. A seção **Materiais e Métodos** detalha os frameworks, bibliotecas, APIs e o fluxo de implementação. Em **Resultados**, são apresentadas a estrutura do projeto, capturas de tela, problemas encontrados e medidas de segurança adotadas. Por fim, em **Conclusão**, refletimos sobre o impacto do trabalho, suas limitações e possibilidades de evolução futura.

## Fundamentação Teórica

O projeto desenvolvido baseia-se na interseção entre inteligência artificial (IA), aplicações web modernas e ensino de línguas. A IA aplicada à educação utiliza algoritmos de aprendizado de máquina e processamento de linguagem natural (PLN) para interpretar, responder e adaptar-se ao usuário. Neste trabalho, a IA é baseada em modelos de linguagem como o GPT (Generative Pre-trained Transformer), que são capazes de compreender e gerar textos com coerência contextual, permitindo conversas mais naturais com o aluno.

Do ponto de vista do desenvolvimento web, utilizamos o padrão cliente-servidor. O cliente (frontend) é responsável pela interface gráfica e comunicação com o usuário, enquanto o servidor (backend) gerencia a lógica de negócio, autenticação e persistência dos dados.

Além disso, princípios de segurança web foram aplicados com o uso de validação de entradas, criptografia de senhas com `bcrypt`, proteção contra injeções com `express-validator`, e autenticação via JSON Web Tokens (JWT). Tais práticas são fundamentais para proteger os dados e garantir uma experiência confiável ao usuário.

# Materiais e Métodos

## Frameworks e Tecnologias Utilizadas

- **Frontend:**
  - **React.js:** biblioteca JavaScript para construção da interface de usuário de forma reativa e modular.
  - **Axios:** cliente HTTP para comunicação com o backend.
  - **React Router:** gerenciamento de rotas da aplicação.
- **Backend:**
  - **Node.js:** ambiente de execução JavaScript do lado do servidor.
  - **Express.js:** framework para gerenciamento de rotas e middlewares.
  - **MongoDB com Mongoose:** banco de dados NoSQL e ODM para persistência dos dados dos usuários.
  - **jsonwebtoken:** geração e verificação de tokens de autenticação.
  - **bcryptjs:** hashing seguro de senhas.
  - **express-validator:** sanitização e validação de entradas no backend.
  - **dotenv:** carregamento de variáveis de ambiente a partir do arquivo `.env`.
  - **cors:** liberação de requisições cross-origin entre frontend e backend.

## Metodologia

A metodologia de desenvolvimento adotada foi baseada na prototipagem incremental. Inicialmente foi criado o layout básico da interface. Em seguida, foram implementadas rotas de autenticação e comunicação com a IA. Cada funcionalidade foi testada isoladamente e depois integrada ao fluxo geral da aplicação.

O modelo de IA utilizado (como base teórica) assume um comportamento tutor, com foco em respostas didáticas, correção de erros gramaticais e prática conversacional contextualizada.

## Roteiro de Instalação e Configuração

### 1. Clone o repositório:

```
git clone https://github.com/Arthurdarosa/projetoweb
```

### 2. Backend:

```
cd backend
npm install
cp .env.example .env
# configure a variável JWT_SECRET e a URL do MongoDB no .env
```

```
npm start
```

### 3. Frontend:

```
cd frontend  
npm install  
npm run dev
```

### 4. Acesse no navegador:

```
http://localhost:5173
```

## Link do Projeto Online

O projeto está hospedado em ambiente de testes no seguinte endereço:

**<https://chat-tutora.vercel.app>**

O código-fonte completo está disponível no GitHub: **<https://github.com/ArthurDarosa/projetowe>**

## Resultados

### Estrutura do Projeto

A seguir, apresenta-se a árvore de diretórios da aplicação, contendo tanto o frontend quanto o backend:

```
1 chat-tutora-ingles/  
2   backend/  
3     controllers/  
4     models/  
5     routes/  
6     middleware/  
7     .env.example  
8     server.js  
9     package.json  
10  frontend/  
11    public/  
12    src/  
13      assets/  
14      components/  
15      pages/  
16      services/  
17      App.jsx  
18      .env  
19      vite.config.js  
20      package.json
```

## Discussão dos Resultados

A aplicação foi testada em diferentes navegadores e dispositivos, demonstrando funcionalidade estável tanto no desktop quanto em navegadores móveis. O chat interativo permitiu conversas em inglês com a IA de forma coerente, com respostas instantâneas e sugestões didáticas em tempo real.

Durante os testes, observou-se que a personalização da IA com um foco educacional (tutora) aumentou a retenção do usuário e gerou maior engajamento nas sessões de prática do idioma.

## Capturas de Tela

### Problemas Encontrados

Durante o desenvolvimento, os seguintes desafios foram enfrentados:

- Integração entre backend e frontend com variáveis de ambiente exigiu ajustes no ‘CORS’.
- O uso incorreto de ‘localStorage’ causava persistência de tokens inválidos.
- A IA apresentava respostas desconexas ao sair do contexto educacional — solucionado por engenharia de prompt.

## Segurança e Vulnerabilidades

A aplicação foi protegida contra ameaças básicas com as seguintes práticas:

- Uso de ‘express-validator’ para sanitização contra NoSQL Injection.
- Criptografia de senhas com ‘bcrypt’.
- Tokens JWT com expiração e middleware de verificação.
- CORS configurado para restringir acessos fora do domínio autorizado.

## Conclusão

O projeto demonstrou como ferramentas modernas de desenvolvimento web e IA podem ser integradas para criar experiências educacionais significativas. A tutora virtual de inglês cumpre o objetivo de auxiliar usuários no aprendizado do idioma de forma prática e acessível. O sistema é escalável e pode ser adaptado para outros idiomas e áreas de ensino.

## Referências

- OpenAI. ChatGPT API. Disponível em: <https://platform.openai.com>
- Node.js. Documentação oficial. <https://nodejs.org>
- React. Framework JavaScript. <https://react.dev>

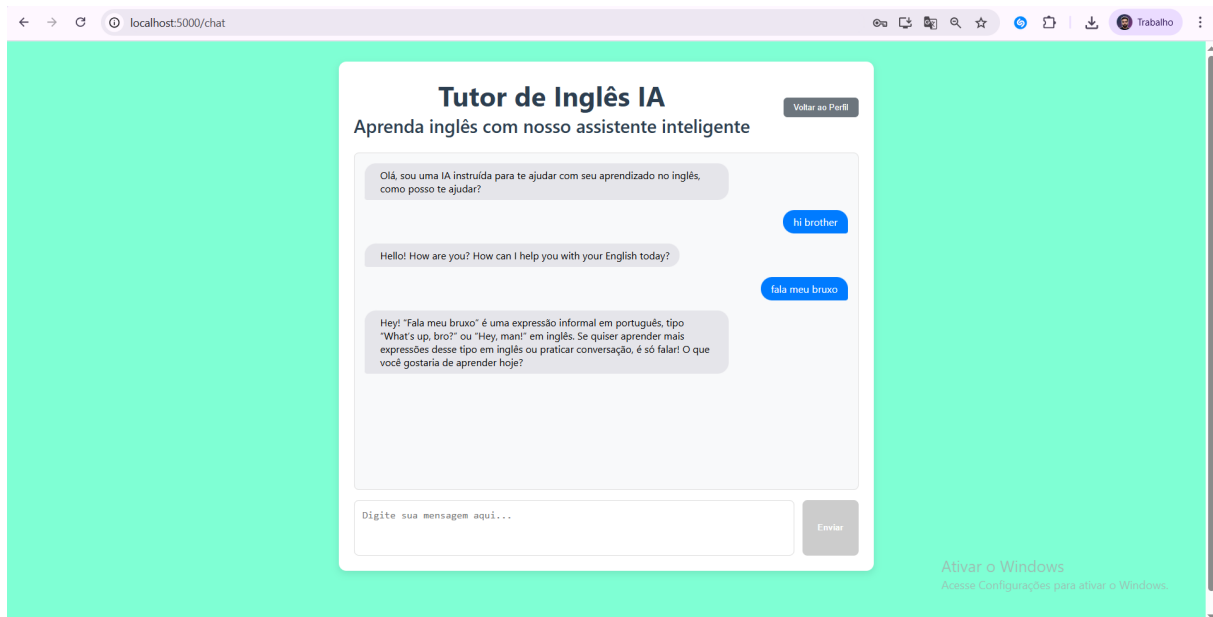


Figura 1: Tela do chat com IA tutora de inglês

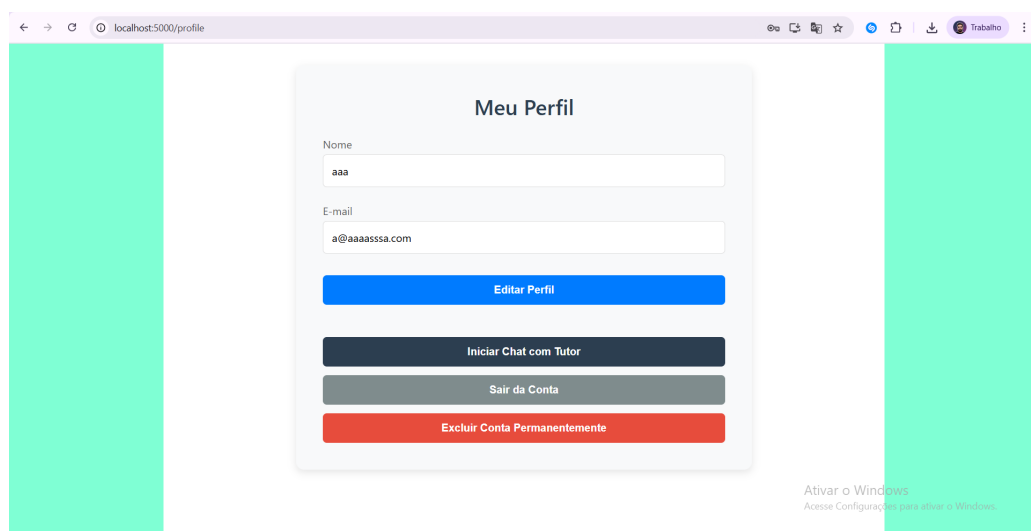


Figura 2: Página de perfil com edição de dados protegida por token JWT

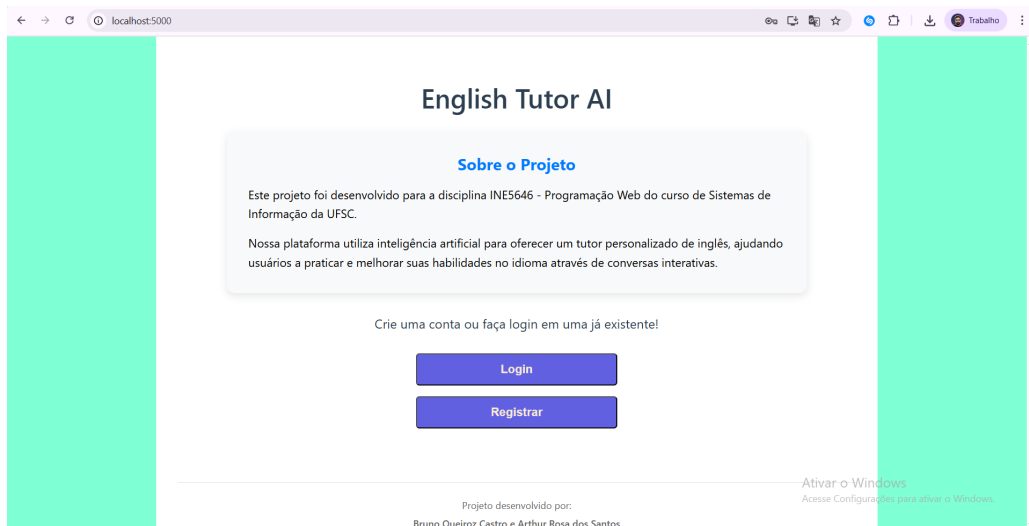


Figura 3: Página inicial

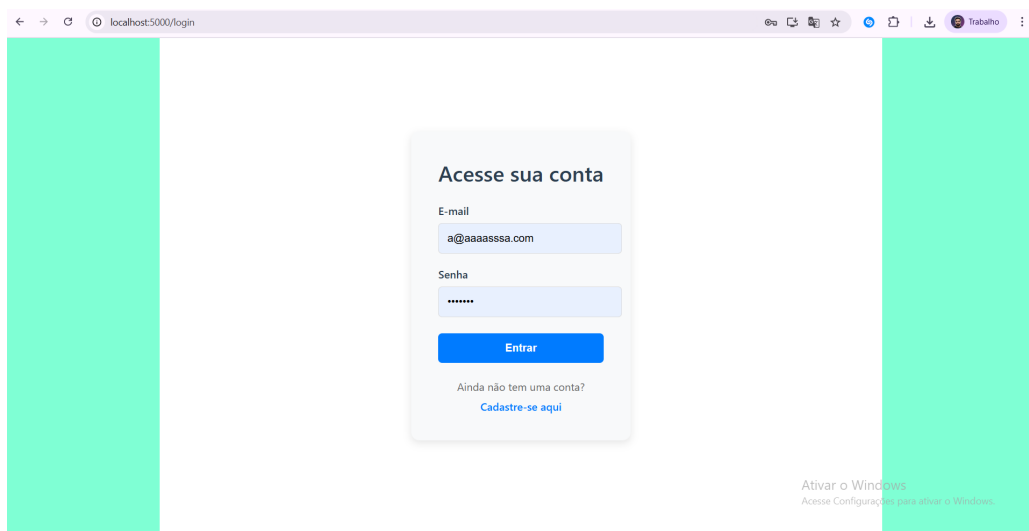


Figura 4: Página de login

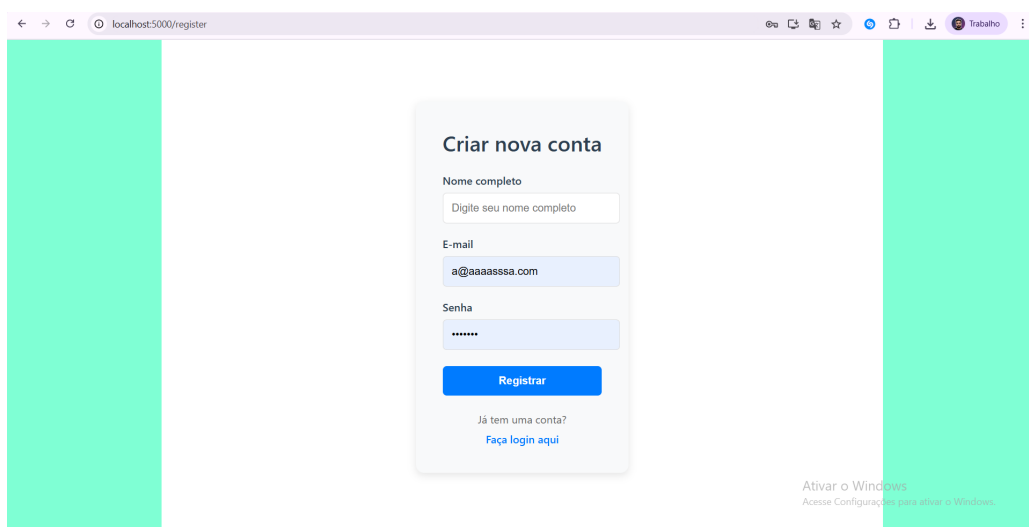


Figura 5: Página de registro

- Express.js. Documentação. <https://expressjs.com>
- bcrypt.js. Hashing de senhas. <https://github.com/kelektiv/node.bcrypt.js>

## Apêndice

### Exemplo de Código - Middleware de Autenticação

Listing 1: Middleware de autenticação JWT

```
1 const jwt = require('jsonwebtoken');
2
3 function autenticarToken(req, res, next) {
4   const authHeader = req.headers['authorization'];
5   const token = authHeader && authHeader.split(' ')[1];
6
7   if (!token) return res.status(401).json({ error: 'Token n o
   fornecido' });
8
9   jwt.verify(token, process.env.JWT_SECRET, (err, payload) => {
10     if (err) return res.status(403).json({ error: 'Token
       inv lido' });
11
12     req.userId = payload.userId;
13     next();
14   });
15 }
```

### Link do repositório

repositório: <https://github.com/Arthurdarosa/projetoweb>

## Anexos

### Código de Terceiros Utilizados

- react-router-dom – Navegação SPA.
- axios – Requisições HTTP.
- jsonwebtoken – Geração e verificação de tokens.
- express-validator – Validação e sanitização de inputs.
- dotenv – Configuração de variáveis de ambiente.

Listing 2: Exemplo de uso de express-validator

```
1 const { body } = require('express-validator');
2
3 router.post('/register', [
```



```
4     body('email').isEmail().withMessage('Email inv lido '),
5     body('senha').isLength({ min: 6 }).withMessage('Senha fraca')
6 ], registrarUsuario);
```