



**UNIVERSIDADE NOVE DE JULHO**  
**PROJETO EM SISTEMAS INTELIGENTES**

**DEISE SANTOS DA SILVA - RA 922114940**

**IZAEL ALVES SILVA - RA 922114939**

**PAULO ROBERTO FILHO - RA 921116535**

**GABRIEL SANTOS DE CAMPOS - RA 922113898**

**EVERSSON SANTANA SERRA BELA - RA 922107547**

**TRAVEL ASSISTANT**

**SÃO PAULO**

**2025**

**DEISE SANTOS DA SILVA - RA 922114940**  
**IZAEL ALVES SILVA - RA 922114939**  
**PAULO ROBERTO FILHO - RA 921116535**  
**GABRIEL SANTOS DE CAMPOS - RA 922113898**  
**EVERSSON SANTANA SERRA BELA - RA 922107547**

**TRAVEL ASSISTANT**

Trabalho apresentado à Universidade Nove de Julho, UNINOVE, em cumprimento parcial às exigências da disciplina de Projeto em sistemas inteligentes, sob orientação do Prof. Dr. Edson Melo de Souza.

**SÃO PAULO**

**2025**

## SUMÁRIO

1. RESUMO	4
2. OBJETIVOS	6
3. DESCRIÇÃO DA EMPRESA	7
4. MISSÃO, VISÃO E VALORES DA EMPRESA	9
4.1. MISSÃO	9
4.2. VISÃO	9
4.3. VALORES	9
5. DEFINIÇÃO DA EQUIPE, DIVISÃO DE PAPÉIS E TAREFAS, CRONOGRAMA DE DESENVOLVIMENTO	10
6. DESENVOLVIMENTO (METODOLOGIA)	12
<b>7. DESENVOLVIMENTO DO ASSISTENTE DE VIAGENS (FRONTEND)</b>	<b>15</b>
7.1. Introdução	15
7.2. Tecnologias Utilizadas	15
7.3. Arquitetura do Projeto	15
7.3.1. Estrutura de Diretórios	15
7.3.2. Configuração do Ambiente	16
7.4. Implementação	16
7.4.1. Interface do Usuário	16
7.4.2. Funcionalidades Principais	16
7.5. Considerações Técnicas	16
7.5.1. Performance	16
7.5.2. Acessibilidade	17
7.6. Conclusão	17
8. DESENVOLVIMENTO DO ASSISTENTE DE VIAGENS (BACKEND)	18
8.1. Visão Geral do Sistema	18
8.2. Arquitetura do Sistema	18
8.2.1. Estrutura do Projeto	18
8.2.2. Tecnologias Utilizadas	18

8.3. Funcionalidades Principais	18
8.3.1. Geração de Roteiros	18
8.3.2. Sistema de Busca Inteligente	19
8.4. Implementação	19
8.4.1. Configuração do Ambiente	19
8.4.2. Fluxo de Processamento	19
8.5. Considerações de Segurança	19
8.6. Conclusão	20
<b>9. ANÁLISE DE RESULTADOS</b>	<b>21</b>
7.1 AVALIAÇÃO	21
7.2 COMPARAÇÃO	21
7.3 SITE DA EMPRESA	21
7.4 GITHUB DO PROJETO	21
<b>10. CONCLUSÃO</b>	<b>22</b>
8.1 CONTRIBUIÇÕES	22
8.2 TRABALHOS FUTUROS	22
<b>11. REFERÊNCIAS</b>	<b>23</b>

## **1. RESUMO**

Neste projeto, desenvolvemos uma assistente de viagem composta por um frontend em JavaScript, e um backend em python, integrando com ferramentas como LLMs da OpenAI, LangChain, pesquisando na web eventos e informações atuais com o DuckDuckGo criando assim um chatbot web. O objetivo é criar um assistente de viagens para ajudar os usuários a gerenciar suas viagens de forma prática, permitindo que os usuários planejem viagens de forma rápida e eficiente, sem a necessidade de pesquisas manuais em vários sites, integração de informações como voos, hospedagens, roteiros e preços em um único lugar. Com uma interface intuitiva e recursos avançados, o assistente de viagem oferece uma solução completa para a gestão de viagens, ajudando os usuários a encontrar seus destinos de forma rápida e com um baixo custo. A comparação automática de preços entre diferentes provedores, como hotéis, companhias aéreas, sites e eventos atuais garante as melhores ofertas. A conexão entre o usuário e o agente de viagens foi essencial para criar roteiros personalizados e garantir uma experiência de viagem satisfatória. O desenvolvimento do Travel Assistant utiliza JavaScript, Python, e ferramentas como LLMs da OpenAI, LangChain, utilizando o navegador DuckDuckGo para pesquisa na web eventos e informações atuais garantindo uma experiência de usuário fluida e interativa. A escolha dessas tecnologias possibilita a criação de um chatbot rápido e eficiente, podendo ser acessado de qualquer lugar pela web.

Palavras-chave: JavaScript, Python, Web, chatbot, OpenIA, DuckDuckGo, LangChain.

## **2. OBJETIVOS**

O principal objetivo do Travel Assistant é facilitar e otimizar a experiência de viagem dos clientes, transformando seus desejos em roteiros práticos, personalizados e memoráveis. Ele atua como um consultor especializado, garantindo que cada aspecto da viagem seja planejado com eficiência e segurança, criando roteiros alinhados com o perfil e interesses dos usuários, indicando destinos, atrações e experiências exclusivas com base em conhecimento de mercado e tendências atuais. Com foco em garantir economia e valor, negociando preços, pacotes promocionais, garantindo viagens mais tranquilas, bem-estruturadas e adaptadas às suas expectativas.

### **3. DESCRIÇÃO DA EMPRESA**

A StarTec é uma empresa de tecnologia pioneira em soluções digitais inteligentes, fundada em 2022 com a missão de transformar desafios complexos, problemas do dia a dia, em oportunidades inovadoras. Combinando IA, Internet das Coisas (IoT) e Big Data, desenvolvemos produtos e serviços que revolucionam indústrias, comércios e varejos, desde saúde, educação, e-commerces e negócios digitais.

Nossos diferenciais são construídos sobre pilares que nos tornam únicos no mercado de tecnologia. Unimos inovação técnica com designs intuitivos através da nossa Tecnologia Humanizada, colocando a experiência do usuário no centro de tudo o que desenvolvemos. Nossas Soluções Customizáveis são adaptadas às necessidades específicas de cada cliente, garantindo que cada projeto atenda perfeitamente aos objetivos do negócio. Investimos continuamente no futuro, destinando 20% do nosso faturamento anual para Pesquisa & Desenvolvimento, o que nos permite antecipar tendências globais e oferecer soluções de ponta. Além disso, comprometemo-nos com a Sustentabilidade Digital, implementando práticas que promovem eficiência energética e redução de carbono em todas as nossas plataformas, porque acreditamos que a tecnologia deve evoluir em harmonia com o meio ambiente. Esses princípios nos permitem entregar não apenas produtos, mas experiências transformadoras que fazem a diferença no dia a dia das pessoas e das empresas.

#### **Principais Produtos**

- Travel Assistant - Plataforma de inteligência artificial para criação de roteiros de viagens.
- Speak4Text - Plataforma que utiliza inteligência artificial para converter áudio em texto.
- Hémedios - Aplicativo Móvel multiplataforma, que cria de cronograma para medicações e remédios de uso contínuo.
- Ecoleta - Aplicativo para cadastrar e localizar pontos de coleta próximo ao usuário.

Na StarTec, cultivamos uma cultura organizacional fundamentada em valores sólidos: Inovação, Transparência e Impacto Social. Usamos esses princípios para cada decisão e projeto que realizamos, criando um ambiente onde ideias revolucionárias nascem com integridade e propósito. Nossa equipe global conta com mais de 500 talentosos colaboradores espalhados pelo mundo, incluindo engenheiros especializados, desenvolvedores, cientistas de

dados e designers, todos unidos pelo mesmo objetivo de transformar o futuro através da tecnologia. Com sede principal em São Paulo, Brasil, expandimos nossa presença com filiais estratégicas em Berlim, Tóquio e Austin (EUA), conectando diferentes culturas e perspectivas em um ecossistema tecnológico verdadeiramente global.

Acreditamos profundamente que a tecnologia deve ser uma força inclusiva e transformadora. Por isso, criamos o programa "Tech for All", nossa iniciativa de responsabilidade social que oferece capacitação digital gratuita para comunidades em desenvolvimento, abrindo portas para oportunidades antes inacessíveis. Além disso, comprometemo-nos a destinar 5% de nossos lucros anuais para apoiar iniciativas de educação tecnológica, reforçando nosso compromisso com um futuro mais justo e equitativo. Na StarTec, não apenas desenvolvemos tecnologia, usamos nosso conhecimento e recursos para construir um impacto social positivo que ultrapassa fronteiras e transforma vidas.



## **4. MISSÃO, VISÃO E VALORES DA EMPRESA**

### **4.1. MISSÃO**

Desenvolver soluções tecnológicas inovadoras que transformem desafios complexos em oportunidades acessíveis, promovendo a inclusão digital e impulsionando o progresso sustentável em escala global.

### **4.2. VISÃO**

Ser reconhecida como líder mundial em tecnologia, criando um ecossistema onde inovação e impacto social andam lado a lado, inspirando mudanças positivas na sociedade e no mercado.

### **4.3. VALORES**

Acreditamos na Inovação com Propósito, buscando constantemente avanços tecnológicos que vão além da simples criação, desenvolvemos soluções que resolvem problemas concretos e trazem melhorias significativas para a vida das pessoas. Praticamos a Transparência como um pilar essencial, mantendo relações baseadas em ética, clareza absoluta e responsabilidade em todos os níveis da organização.

5. CRONOGRAMA DE DESENVOLVIMENTO

A equipe será composta por profissionais multidisciplinares, cada um será responsável por uma área do projeto:

Nome	Cargo	Responsabilidades
Deise Santos da Silva	CEO / Líder do Projeto	responsável pelas entregas, com visão estratégica, tomada de decisões
Izael Alves Silva	CTO (Tech Lead)	Arquitetura técnica, escolha de tecnologias, coordenação do time de desenvolvimento
Paulo Roberto Filho	Desenvolvedor	implementação do backend
Gabriel S. De Campos	Desenvolvedor	implementação do frontend
Eversson S. Serra Bela	Suporte & Operações	implementação e monitoramento

Cronograma de Desenvolvimento

O projeto será dividido em 5 fases principais, com entregas e marcos definidos:

Fase	Duração	Principais Tarefas
Concepção e pesquisa	2 semanas	Validação da ideia, análise e fase de requisitos
Prototipagem	2 semanas	desenvolvimento, testes de usabilidade e ajustes de design
Desenvolvimento	2 meses	criação da versão beta, integração com APIs e banco de dados
Implementação	1 mês	lançamento e monitoramentos inicial
Operação	Contínuo	melhorias iterativas, correção de bugs e otimização de custos

## **DESENVOLVIMENTO (METODOLOGIA)**

Neste capítulo, descrevemos a abordagem metodológica adotada para o desenvolvimento do projeto, incluindo as tecnologias utilizadas, a estrutura do projeto e as ferramentas de desenvolvimento empregadas. A metodologia seguida neste projeto inclui a análise de requisitos, escolha das tecnologias, desenvolvimento incremental dos componentes, e testes contínuos para garantir a funcionalidade e performance da aplicação.

### **1. DESENVOLVIMENTO**

A aplicação foi desenvolvida utilizando uma abordagem modular, onde cada componente foi implementado e testado de forma independente antes de ser integrado ao projeto principal.

### **2. VISÃO GERAL**

Fornecer uma estrutura sólida para o desenvolvimento do projeto, delineando os princípios, ferramentas e processos-chave que serão seguidos para alcançar os objetivos estabelecidos.

### **3. TESTES**

Os testes foram conduzidos para garantir que todas as funcionalidades da aplicação funcionassem conforme o esperado. Foram realizados testes unitários nos componentes individuais e testes de integração na aplicação como um todo.

### **4. FRONTEND**

O frontend é a parte de um sistema de software que os usuários interagem diretamente. Também é conhecido como a interface do usuário (UI). Ele consiste em todos os elementos visíveis e interativos de um aplicativo ou site, incluindo botões, menus, formulários, gráficos, entre outros.

As principais responsabilidades do frontend incluem:

1. **Apresentação de Conteúdo:** O frontend é responsável por exibir informações e conteúdo de maneira clara e organizada para os usuários.

2. **Interatividade:** Ele fornece meios para os usuários interagirem com o sistema, como clicar em botões, preencher formulários e navegar por diferentes páginas.
3. **Usabilidade:** O frontend é projetado para ser intuitivo e fácil de usar, proporcionando uma experiência agradável ao usuário.
4. **Estilo e Design:** Ele trata do layout, estilo e design visual do aplicativo ou site, garantindo uma aparência atraente e coesa.
5. **Performance:** O frontend também é responsável por garantir uma resposta rápida e eficiente às interações do usuário, minimizando os tempos de carregamento e proporcionando uma experiência fluida.

## 5. BACKEND

O backend, também conhecido como servidor ou camada de servidor, é a parte de um sistema de software que lida com o processamento dos dados, a lógica de negócios e a interação com o banco de dados. Ele é responsável por receber requisições do frontend (interface do usuário) e retornar as respostas apropriadas.

Em uma arquitetura de software típica, o frontend é a parte da aplicação com a qual o usuário interage diretamente, enquanto o backend é responsável por fornecer os recursos e funcionalidades necessários para que o frontend funcione corretamente.

- **Python**

Python é uma linguagem de programação de uso geral, é uma linguagem que funciona para criar uma grande variedade de aplicações diferentes. Essa versatilidade e a facilidade de uso para pessoas iniciantes, fez com que se tornasse uma das linguagens mais comuns atualmente. Por ser uma linguagem interpretada ela não precisa passar pelo processo de compilação, sua sintaxe simples é fácil de aprender, muito próxima à linguagem humana

- **OpenAI API**

Integrada para fornecer recursos avançados de processamento de linguagem natural. Segundo a documentação oficial, OpenAI API é uma interface de programação de aplicativos que permite que os desenvolvedores usem os modelos de linguagem de

ponta da OpenAI em seus próprios aplicativos. Os modelos de linguagem são treinados em uma ampla gama de textos para gerar respostas coerentes e úteis a uma variedade de solicitações de texto.

- **LangChain**

LangChain é um framework open-source que simplifica a criação de aplicações avançadas com modelos de linguagem (LLMs), ele permite conectar LLMs a bancos de dados, APIs e outras ferramentas, criando sistemas inteligentes com memória contextual, agentes autônomos e respostas baseadas em dados externos (RAG), automação de tarefas e soluções empresariais escaláveis, o LangChain aumenta a eficiência e a personalização no uso de IA generativa.

- **Retrivial De Documentos**

Retrieval de documentos é a espinha dorsal de sistemas que precisam encontrar informações em meio a grandes volumes de dados não estruturados, combinando técnicas de NLP e aprendizado de máquina para entregar resultados precisos e contextualizados.

- **DuckDuckGo**

DuckDuckGo é um mecanismo de busca alternativo que prioriza a privacidade do usuário, diferenciando-se dos buscadores tradicionais por não rastrear, armazenar dados pessoais ou criar perfis de usuários, desenvolvido como alternativa ao Google, possui uma busca privada não coleta histórico, IP ou informações pessoais

## 6. DESENVOLVIMENTO DO ASSISTENTE DE VIAGENS (FRONTEND)

### 6.1. Introdução

O Travel Assistant Web é uma aplicação web moderna desenvolvida com o objetivo de auxiliar usuários no planejamento e organização de suas viagens. Este capítulo descreve detalhadamente a implementação técnica do projeto, suas tecnologias e arquitetura.

### 6.2. Tecnologias Utilizadas

O projeto foi desenvolvido utilizando um conjunto moderno de tecnologias e frameworks, incluindo:

- **React 18:** Framework JavaScript para construção de interfaces de usuário
- **TypeScript:** Superset tipado do JavaScript
- **Vite:** Ferramenta de build moderna para desenvolvimento front-end
- **Tailwind CSS:** Framework CSS utilitário para estilização
- **Radix UI:** Biblioteca de componentes acessíveis e customizáveis
- **Axios:** Cliente HTTP para requisições à API

### 6.3. Arquitetura do Projeto

A arquitetura do projeto segue os princípios modernos de desenvolvimento web, organizando-se da seguinte forma:

#### 6.3.1. Estrutura de Diretórios

O projeto é organizado em uma estrutura modular, com separação clara de responsabilidades:

travel-assistant-front/

```
├── src/           # Código fonte principal
├── public/        # Arquivos estáticos
├── node_modules/  # Dependências
└── configurações # Arquivos de configuração
```

### 6.3.2. Configuração do Ambiente

O ambiente de desenvolvimento é configurado através de diversos arquivos de configuração:

- **vite.config.ts**: Configuração do bundler Vite
- **tsconfig.json**: Configuração do TypeScript
- **tailwind.config.js**: Configuração do Tailwind CSS
- **postcss.config.js**: Configuração do PostCSS

## 6.4. Implementação

### 6.4.1. Interface do Usuário

A interface do usuário é construída utilizando componentes React modernos e estilizados com Tailwind CSS. A aplicação implementa:

- Design responsivo
- Componentes acessíveis (via Radix UI)
- Animações suaves
- Feedback visual através de toasts

### 6.4.2. Funcionalidades Principais

O sistema implementa diversas funcionalidades para auxiliar no planejamento de viagens, incluindo:

- Integração com APIs externas
- Interface intuitiva e moderna
- Feedback em tempo real

## 6.5. Considerações Técnicas

### 6.5.1. Performance

O projeto foi otimizado para performance através de:

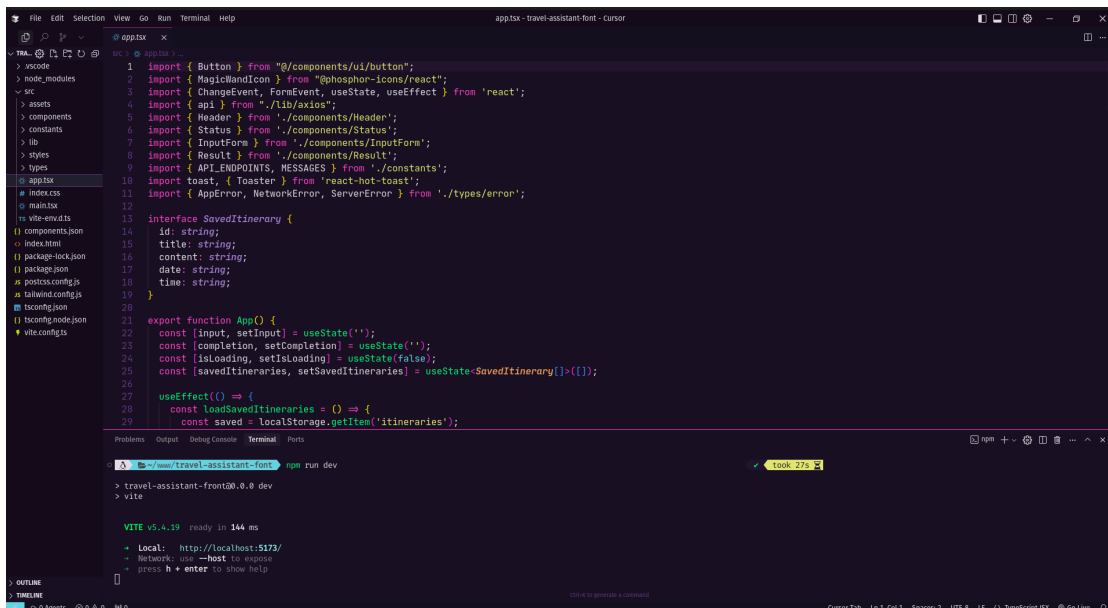
- Bundling eficiente com Vite
- Lazy loading de componentes
- Otimização de assets
- Minificação de código

## 6.5.2. Acessibilidade

A acessibilidade foi priorizada através de:

- Componentes Radix UI
- Semântica HTML adequada
- Suporte a leitores de tela
- Contraste adequado

## 6.6. Imagens do backend



```
1 import { Button } from "@components/ui/button";
2 import { Magic WandIcon } from "@phosphor-icons/react";
3 import { ChangeEvent, FormEvent, useState, useEffect } from "react";
4 import { api } from "../lib/axios";
5 import { Header } from "../components/Header";
6 import { Status } from "../components/Status";
7 import { InputForm } from "../components/InputForm";
8 import { Result } from "../components/Result";
9 import { API_ENDPOINTS, MESSAGES } from "../constants";
10 import toast, { Toaster } from "react-hot-toast";
11 import { AppError, NetworkError, ServerError } from "../types/error";
12
13 interface SavedItinerary {
14   id: string;
15   title: string;
16   content: string;
17   date: string;
18   time: string;
19 }
20
21 export function App() {
22   const [input, setInput] = useState("");
23   const [completion, setCompletion] = useState("");
24   const [isLoading, setIsLoading] = useState(false);
25   const [savedItineraries, setSavedItineraries] = useState<SavedItinerary[]>([]);
26
27   useEffect(() => {
28     const loadSavedItineraries = () => {
29       const saved = localStorage.getItem("itineraries");
```

Problems Output Debug Console Terminal Ports

npm run dev

travel-assistant-front@0.0.0 dev

vite

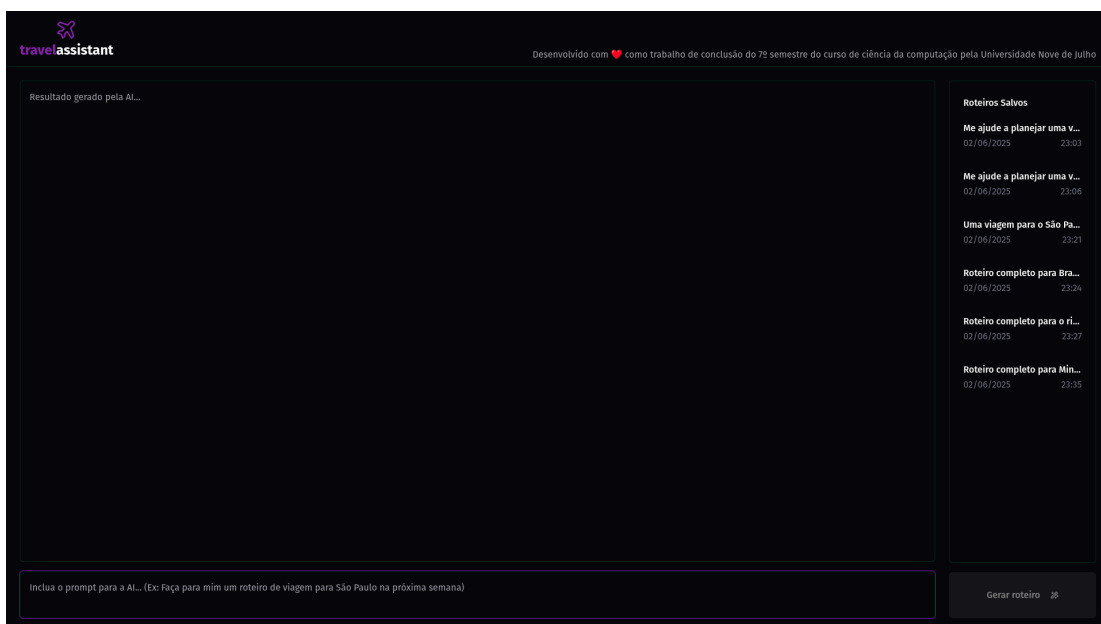
VITE v5.4.19 ready in 144 ms

Local: http://localhost:5173/

Networks: use --host to expose

press h + enter to show help

fonte: imagem/printscreens



fonte: imagem/printscreens



## 7. DESENVOLVIMENTO DO ASSISTENTE DE VIAGENS (BACKEND)

### 7.1. Visão Geral do Sistema

O Travel Assistant é uma aplicação desenvolvida em Python que utiliza inteligência artificial para criar roteiros de viagem personalizados. O sistema foi projetado para combinar informações da Wikipedia e buscas na web em tempo real, gerando itinerários detalhados e relevantes para os usuários.

### 7.2. Arquitetura do Sistema

#### 7.2.1. Estrutura do Projeto

O projeto foi organizado seguindo uma arquitetura modular, dividida nos seguintes componentes principais:

- **API:** Responsável pelos endpoints da aplicação
- **Core:** Contém a lógica principal do assistente
- **Config:** Gerencia as configurações do sistema
- **Services:** Implementa serviços auxiliares
- **Utils:** Fornece utilitários para o sistema
- **Chroma DB:** Banco de dados vetorial para armazenamento de informações

#### 7.2.2. Tecnologias Utilizadas

O sistema foi desenvolvido utilizando as seguintes tecnologias:

- Python 3.10+
- FastAPI para desenvolvimento da API REST
- Google Gemini para processamento de linguagem natural
- Chroma DB para armazenamento vetorial
- Wikipedia API para busca de informações turísticas

### 7.3. Funcionalidades Principais

#### 7.3.1. Geração de Roteiros

O sistema é capaz de gerar roteiros personalizados considerando:

- Preferências do usuário
- Restrições de tempo
- Orçamento disponível

- Pontos turísticos relevantes
- Recomendações de restaurantes e hospedagens
- Informações sobre transportes

#### 7.3.2. Sistema de Busca Inteligente

O assistente implementa um sistema de busca que:

- Realiza pesquisas na Wikipedia
- Executa buscas na web em tempo real
- Armazena informações relevantes em banco de dados vetorial
- Utiliza processamento de linguagem natural para compreensão das consultas

### 7.4. Implementação

#### 7.4.1. Configuração do Ambiente

O sistema requer:

- Ambiente virtual Python
- Chave de API do Google para o modelo Gemini
- Dependências listadas no arquivo requirements.txt
- Configuração de variáveis de ambiente

#### 7.4.2. Fluxo de Processamento

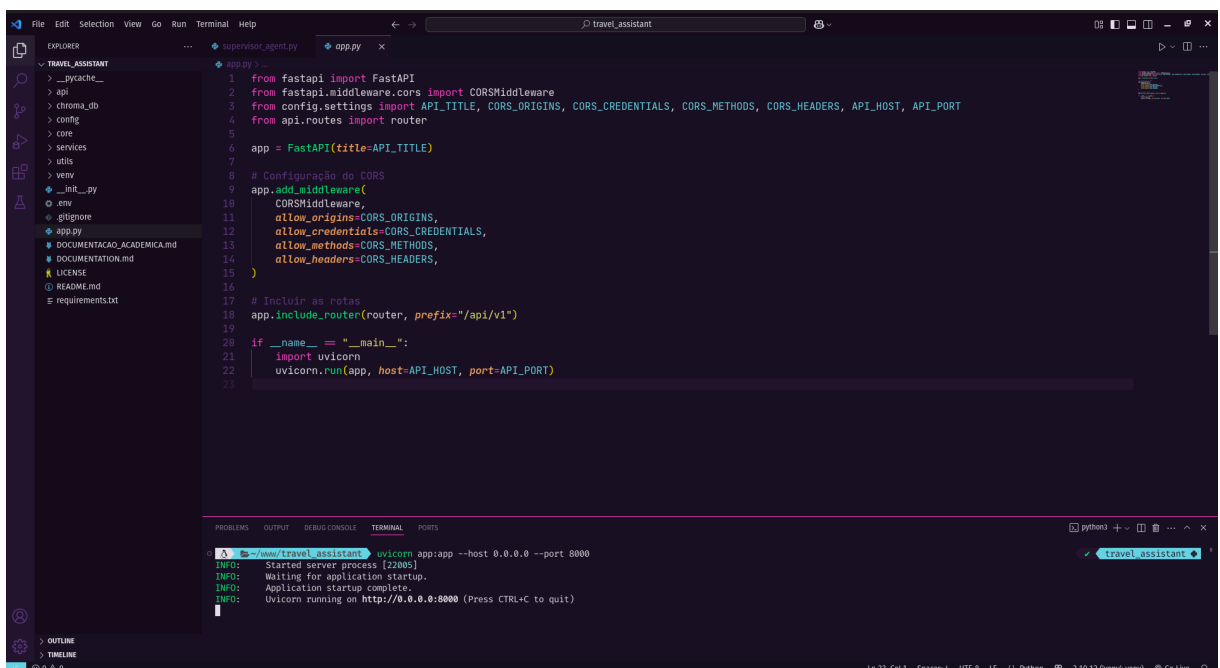
- Recebimento da consulta do usuário
- Processamento da solicitação via API
- Busca de informações relevantes
- Geração do roteiro personalizado
- Retorno da resposta formatada

## 7.5. Considerações de Segurança

O sistema implementa medidas de segurança como:

- Proteção de chaves de API
- Validação de entradas
- Tratamento de erros
- Logs de sistema

## 7.6. Imagens do backend



```
1 from fastapi import FastAPI
2 from fastapi.middleware.cors import CORSMiddleware
3 from config.settings import API_TITLE, CORS_ORIGINS, CORS_CREDENTIALS, CORS_METHODS, CORS_HEADERS, API_HOST, API_PORT
4 from api.routes import router
5
6 app = FastAPI(title=API_TITLE)
7
8 # Configuração do CORS
9 app.add_middleware(
10     CORSMiddleware,
11     allow_origins=CORS_ORIGINS,
12     allow_credentials=CORS_CREDENTIALS,
13     allow_methods=CORS_METHODS,
14     allow_headers=CORS_HEADERS,
15 )
16
17 # Incluir as rotas
18 app.include_router(router, prefix="/api/v1")
19
20 if __name__ == "__main__":
21     import uvicorn
22     uvicorn.run(app, host=API_HOST, port=API_PORT)
23
```

Terminal output:

```
INFO: Started server process [22005]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
```

fonte: imagem/printscreens

## 7.7. Conclusão

O Travel Assistant representa uma solução inovadora na área de assistentes de viagem, combinando tecnologias modernas de inteligência artificial com uma arquitetura robusta e escalável. O sistema demonstra a aplicação prática de conceitos de processamento de linguagem natural e busca de informações em um contexto real de uso.

## **8. ANÁLISE DE RESULTADOS**

Os resultados do desenvolvimento indicam que a aplicação é capaz de buscar ofertas e descontos de forma rápida e eficiente. O tempo de busca e a qualidade da pesquisa foram satisfatórios, atendendo aos requisitos definidos. Nossos próximos passos é elevar este produto a outros patamares, até porque no momento o software só faz buscas usando o DuckDuckGo, e documentos na web, queremos que ele use várias bases de dados.

### **7.1 AVALIAÇÃO**

A aplicação mostrou-se eficiente, porém foram identificados alguns desafios, como a limitação de performance ao buscar destinos com nome de cidades parecidas no navegador

### **7.2 COMPARAÇÃO**

Os resultados obtidos estão alinhados com os objetivos propostos. A aplicação oferece uma interface intuitiva e funcional, permitindo a busca por um roteiro simples e custo abaixo do mercado de forma eficiente.

### **7.3 SITE DA EMPRESA**

- <https://dezyzy.github.io/StarTec/>

### **7.4 GITHUB DO PROJETO**

- Código do projeto no github  
[https://github.com/ias4g/projeto\\_em\\_sistemas\\_inteligentes\\_uni9\\_7s](https://github.com/ias4g/projeto_em_sistemas_inteligentes_uni9_7s)

## **9. CONCLUSÃO**

O projeto atingiu seus objetivos principais, demonstrando que é possível criar um chatbot eficiente para criação de cronograma e roteiros de viagem. A integração do LangChain com Python e OpenAI API mostrou-se uma solução viável e eficaz.

### **8.1 CONTRIBUIÇÕES**

Este trabalho contribui para a área de automação e desenvolvimento, mostrando como tecnologias modernas podem ser utilizadas para resolver problemas complexos de forma eficiente.

### **8.2 TRABALHOS FUTUROS**

Sugestões para trabalhos futuros incluem a otimização de performance para processamento de perguntas e informações e a adição de mais funcionalidades, como mostrar fotos do local, hotéis e resorts.

10. REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023: informação e documentação: referências: elaboração.** Rio de Janeiro, 2002.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6028: informação e documentação: resumo: apresentação.** Rio de Janeiro, 2003.

**DuckDuckGo:** o mecanismo de busca que respeita sua privacidade. [S. l.], 2024. Disponível em: <<https://duckduckgo.com/>>. Acesso em: 3 abr. 2025.

**LangChain Documentation:** building applications with LLMs through composability. [S. l.], 2024. Disponível em: <[https://python.langchain.com/docs/introduction/?\\_gl=1\\*ixlplq\\*\\_ga\\*MTA3OTc3MDI0Ni4xNzQ2NjIzMDI4\\*\\_ga\\_47WX3HKKY2\\*czE3NDY2MjMwMjgkbzEkZzAkdDE3NDY2MjMwMjgkajAkbDAkaDA](https://python.langchain.com/docs/introduction/?_gl=1*ixlplq*_ga*MTA3OTc3MDI0Ni4xNzQ2NjIzMDI4*_ga_47WX3HKKY2*czE3NDY2MjMwMjgkbzEkZzAkdDE3NDY2MjMwMjgkajAkbDAkaDA)>. Acesso em: 7 Mar. 2025.

**OpenAI Platform Documentation.** 2024. Disponível em: <<https://platform.openai.com/docs/overview>>. Acesso em: 3 Mar. 2025.

**Artigos sobre Python: Alura,** 2024. Disponível em: <[https://www.alura.com.br/artigos/python?srsId=AfmBOoqoaAP4SGzFXAvNiCDDFeOOQnupo\\_mhODmtC\\_9i8jNBBTTqJn5f](https://www.alura.com.br/artigos/python?srsId=AfmBOoqoaAP4SGzFXAvNiCDDFeOOQnupo_mhODmtC_9i8jNBBTTqJn5f)>. Acesso em: 3 abr. 2025.

**Blog da Asimov Academy: aprendizado em IA,** 2024. Disponível em: <<https://hub.asimov.academy/blog/>>. Acesso em: 3 Mar. 2025.

SEVERINO, Antônio Joaquim. **Metodologia do trabalho científico.** 22. ed. rev. ampl. São Paulo: Cortez, 2002.