****

**UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE**

**CAMPUS DE NATAL**

**DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA**

**DOCUMENTAÇÃO DO VOYAGE: BANCO DE DADOS**

**COMPONENTES: GEYSLAINE ANGELO DANTAS DE MORAIS, FELIPE FREITAS DE MEDEIROS**

**ORIENTADOR: BARTIRA PARAGUAÇU FALCÃO DANTAS ROCHA**

Natal – RN

2024

DOCUMENTAÇÃO DO VOYAGE

Trabalho apresentado ao Curso de Bacharelado

em Ciência da Computação como pré-requisito

à conclusão da disciplina de Banco de Dados.

Natal – RN

2024

DOCUMENTAÇÃO DO VOYAGE

**RESUMO**

Este trabalho tem como objetivo documentar sobre o banco de dados do software Voyage, que tem como propósito a criação de roteiros personalizados, juntando tudo que é necessário para ter uma experiência de uma viagem incrível em um só lugar, na plataforma é possível reservar e ver disponibilidade de passeios, hotéis, automóveis e guias turísticos. Entretanto, o foco principal deste documento é detalhar a estruturação do banco de dados, abrangendo o gerenciamento das informações relacionadas à criação de roteiros, interações com guias turísticos, e reservas de hotéis e passeios. Como resultado, a documentação apresenta uma descrição detalhada da estrutura do banco de dados, incluindo diagramas relacionais, definição de tabelas, restrições e relacionamentos, assegurando que as informações armazenadas atendam às necessidades do sistema e possam ser facilmente mantidas e expandidas no futuro.

**Palavras-chaves:** Roteiro de viagens; Roteiros personalizados; Banco de dados;

**LISTA DE IMAGENS**

Imagem 1 - Conectando banco de dados na API do sistema

Imagem 2 - Criação de rotas na API do sistema

Imagem 3 - Swagger, interface da api com toda a documentação do sistema

Imagem 4 - Interface da página HOME do site

Imagem 5 - Interface da página de login do site

Imagem 6 - Interface da página de registrar usuário do site

**LISTA DE TABELAS**

Tabela 1 - Dicionário de Dados de Pessoa

Tabela 2 - Dicionário de Dados de Usuário

Tabela 3 - Dicionário de Dados de Endereço

Tabela 4 - Dicionário de Dados de Guia

Tabela 5 - Dicionário de Dados de Destino

Tabela 6 - Dicionário de Dados de Passeio

Tabela 7 - Dicionário de Dados de Roteiro

Tabela 8 - Dicionário de Dados de Notificação

Tabela 9 - Dicionário de Dados de Área de Atuação

Tabela 10 - Dicionário de Dados de Guia Área de Atuação

Tabela 11 - Dicionário de Dados de Idioma

Tabela 12 - Dicionário de Dados de Idioma

Tabela 13 - Dicionário de Dados de Reserva

Tabela 14 - Dicionário de Dados de Avaliação

Tabela 15 - Tabela de Elaboração do Projeto

### 

# 

[**1. INTRODUÇÃO 6**](#_30j0zll)

[1.1 Justificativa 7](#_v0jnr7p96tit)

[**2. OBJETIVOS 7**](#_tyjcwt)

[2.1 Objetivos Gerais 7](#_dtjdub3g3xtc)

[2.2 Objetivos Específicos 8](#_ogg7i07ogtwb)

[**3. METODOLOGIA 8**](#_17dp8vu)

[**4. DISCUSSÃO E REQUISITOS DO SISTEMA 9**](#_1t3h5sf)

[4.1 Requisitos Funcionais 9](#_bc4xebevt85n)

[4.2 Requisitos Não Funcionais 10](#_86c19w5uj623)

[**5. MODELO ENTIDADE-RELACIONAMENTO 11**](#_j1uox7li6az0)

[5.1 Diagrama Entidade-Relacionamento 11](#_w6inex3ttmk4)

[**6. MODELO RELACIONAL NORMALIZADO E CONSULTAS 12**](#_fc9qy7181q1x)

[**7. DICIONÁRIO DE DADOS 16**](#_ibwk6e14o9cz)

[Tabela 1 - Dicionário de Dados de Pessoa 17](#_oftfsssf8b9f)

[Tabela 2 - Dicionário de Dados de Usuário 17](#_h13yl6a93j8j)

[Tabela 3 - Dicionário de Dados de Endereço 18](#_ed348ypaf2kd)

[Tabela 4 - Dicionário de Dados de Guia 19](#_6xp7fvci6r7a)

[Tabela 5 - Dicionário de Dados de Destino 21](#_l942ww4ml8fj)

[Tabela 6 - Dicionário de Dados de Passeio 21](#_ihn09s1w5npf)

[Tabela 7 - Dicionário de Dados de Roteiro 23](#_4yvf2t2n97gl)

[Tabela 8 - Dicionário de Dados de Notificação 24](#_yc4zxyxza0ok)

[**Tabela 9 - Dicionário de Dados de Área de Atuação 25**](#_4pddy3kzwac2)

[**Tabela 10 - Dicionário de Dados de Guia Área de Atuação 25**](#_on04x5z56t4n)

[**Tabela 11 - Dicionário de Dados de Idioma 26**](#_hk7vf6g1sxqv)

[**Tabela 12 - Dicionário de Dados de Idioma 26**](#_pf5pxmpt1ip)

[**Tabela 13 - Dicionário de Dados de Reserva 27**](#_41rblzoaf04k)

[**Tabela 14 - Dicionário de Dados de Avaliação 28**](#_t7pix2web88g)

[**8. SISTEMA E CONEXÃO COM O MYSQL 30**](#_f5jvgemh0blu)

[8.1 Estrutura do Sistema 30](#_qvhabav0l7nl)

[8.2 Integração com o MySQL 30](#_qohbfvjpyf1h)

[8.3 Segurança e Escalabilidade / Autenticação e Autorização 31](#_uihaa68af44b)

[8.4 Funcionamento Geral  
 Quando um usuário interage com a interface do Voyage, o front-end faz requisições para a API, que processa a solicitação, interage com o banco de dados MySQL e retorna a resposta ao front-end. Essa arquitetura desacoplada facilita a manutenção e evolução do sistema, permitindo que cada camada seja atualizada independentemente. 31](#_8esw4o7sm15s)

[8.5 Estrutura do Front-end 32](#_8rtqyp6e5vb8)

[**8.6 Capturas de tela do sistema 33**](#_vqfhhaxot2q8)

[**Imagem 6 - Interface da página de registrar usuário do site 36**](#_smajenoshqhe)

[**https://drive.google.com/file/d/1zr217\_HTi1gaokRc8-r1D9kPArr\_VLrv/view?usp=drive\_link 36**](#_uc9gtwnl4a9s)

[**9. RESULTADOS 37**](#_3rdcrjn)

[9.1 Resultados Iniciais 37](#_6006krdihxqe)

[9.2 Resultados Esperados 37](#_2rpby83rcsmf)

[**10. CRONOGRAMA 37**](#_289xlkrb54mz)

[Tabela 15 - Tabela de Elaboração do Projeto 37](#_37ghfyh6mgmc)

[**11.REPOSITÓRIOS 39**](#_gf4mvryirf1g)

# 1. INTRODUÇÃO

Um roteiro organizado não apenas otimiza tempo e recursos, mas também evita imprevistos, garantindo uma viagem tranquila e proveitosa. Durante a alta temporada, quando a procura por serviços aumenta e os preços sobem, o planejamento torna-se ainda mais importante para equilibrar diversão e custo-benefício.

Ao programar a viagem com antecedência, você tem a chance de aproveitar passagens mais baratas, pesquisar promoções e escolher entre diversas opções de hospedagem, garantindo acomodações que atendem ao seu orçamento e necessidades. Além disso, planejar permite criar um roteiro flexível, mas eficiente, organizando visitas aos principais pontos turísticos e otimizando o tempo para explorar o máximo que o destino oferece.

Outro benefício é a tranquilidade na organização dos documentos, principalmente em viagens internacionais, onde vistos e vacinas podem ser obrigatórios. Essa preparação reduz riscos de imprevistos burocráticos e assegura uma experiência mais tranquila. Pesquisar sobre o destino, sua cultura e atrações também enriquece a viagem, permitindo que você aproveite ao máximo cada momento.

O aplicativo Voyage, cujo nome é inspirado no termo francês *voyage* (viagem), reflete sua essência de promover experiências personalizadas e inovadoras no planejamento de roteiros de viagens. O principal objetivo do Voyage é ser uma plataforma intuitiva que conecta viajantes a guias locais, destinos e serviços, simplificando a organização de viagens de forma eficiente e interativa.

A plataforma busca atender tanto viajantes quanto guias turísticos. Os viajantes, público principal, podem planejar viagens personalizadas, criar roteiros e explorar passeios, enquanto os guias têm a oportunidade de divulgar seus serviços e atuar em áreas específicas. O público-alvo inclui turistas ocasionais, aventureiros frequentes e guias profissionais que desejam aproveitar uma ferramenta prática para organizar e promover viagens.

O Voyage aproveita associações bem definidas para proporcionar interações fluídas. A herança entre Usuário, Guia e Pessoa, por exemplo, permite o compartilhamento de atributos básicos, como nome e contatos. A associação entre Roteiro e Destino agrega múltiplos destinos a um único roteiro, enquanto Roteiro e Notificação garantem comunicação eficiente sobre eventos importantes.

Além disso, o aplicativo facilita a interação entre Usuários e Guias, permitindo que os viajantes encontrem profissionais especializados nas áreas de interesse. É uma ferramenta inovadora e eficiente, projetada para simplificar o planejamento de viagens enquanto conecta viajantes e guias turísticos. Baseado em uma modelagem de classes de domínio robusta, o site oferece uma experiência organizada e personalizada, atendendo às necessidades de um público diversificado e promovendo interações significativas entre seus usuários.

## 1.1 Justificativa

A documentação do banco de dados do Voyage é essencial para estabelecer uma estrutura sólida e organizada que suporte todas as funcionalidades necessárias para um sistema de planejamento de viagens eficiente. O setor de turismo demanda uma gestão precisa e integrada de diversos tipos de dados, desde informações de usuários até detalhes específicos de destinos e serviços.

A complexidade das interações entre viajantes, guias turísticos, roteiros e serviços exige um sistema de banco de dados bem estruturado e documentado. Essa documentação não apenas facilita o desenvolvimento inicial do sistema, mas também garante sua manutenção e evolução ao longo do tempo, permitindo que novos recursos sejam implementados de maneira consistente.

O mercado turístico está em constante evolução, com novas demandas surgindo frequentemente. Uma documentação detalhada do banco de dados permite que o sistema se adapte a essas mudanças de forma ágil e estruturada, mantendo a integridade e a eficiência das operações.

Além disso, a documentação serve como referência fundamental para desenvolvedores, administradores e outros profissionais envolvidos no projeto, assegurando que todos compreendam a estrutura dos dados e suas relações. Isso é particularmente importante em um sistema que lida com informações sensíveis como dados pessoais, reservas e transações financeiras.

A escolha de um modelo relacional para o banco de dados do Voyage permite estabelecer conexões lógicas entre diferentes tipos de informação, como usuários, destinos, roteiros e avaliações, garantindo a consistência e a confiabilidade dos dados armazenados. Esta documentação, portanto, visa detalhar essa estrutura e suas relações, fornecendo uma base sólida para o desenvolvimento e a manutenção contínua do sistema.

# 2. OBJETIVOS

## 2.1 Objetivos Gerais

O objetivo geral do Voyage é facilitar a criação de roteiros personalizados e acesso a variedades de passeios que é possível fazer em um determinados destino, além disso, conecta usuários a profissionais do setor turístico de forma simples e amigável, promovendo viagens organizadas, com ótimo custo-benefício e planejamento eficiente, garantindo uma experiência otimizada e sem preocupações.Logo, neste documento terá como objetivo informar detalhes sobre o desenvolvimento, implementação e estrutura do banco de dados desse sistema

## 2.2 Objetivos Específicos

* Armazenar e gerenciar dados pessoais dos usuários, incluindo preferências de viagem e histórico de interações.
* Registrar detalhes das atividades dentro dos roteiros, como horários, descrições e custos estimados.
* Desenvolver um sistema eficiente e intuitivo, garantindo usabilidade aprimorada e acesso ágil às funcionalidades, com foco na criação de roteiros personalizados e na conexão direta entre usuários e profissionais turísticos.

# 3. METODOLOGIA

Para o desenvolvimento do projeto, foi adotada uma abordagem estruturada, seguindo os passos necessários para garantir eficiência e funcionalidade durante o processo de desenvolvimento:

**Definição dos requisitos:**

Na definição dos requisitos, foi realizada uma análise detalhada das necessidades do sistema de gerenciamento de roteiros turísticos, identificando os principais atores (viajantes, guias e administradores), levantando as funcionalidades essenciais como autenticação, gestão de destinos, passeios e roteiros, além de definir as regras de negócio, incluindo permissões de acesso e fluxos de operação.

**Modelagem do Banco de Dados:**

A modelagem do banco de dados envolveu a criação do modelo entidade-relacionamento para representar a estrutura de dados, definindo as entidades principais (PESSOA, GUIA, DESTINO, PASSEIO, ROTEIRO) e estabelecendo seus relacionamentos. Foi realizada a normalização das tabelas para evitar redundância e garantir integridade dos dados, implementação de sistemas de autenticação e autorização (AUTH), além da definição de timestamps para rastreamento de criação e atualização de registros.

**Consultas Textuais:**

No desenvolvimento das consultas textuais, foram criadas queries para operações CRUD em todas as entidades, implementação de junções (JOINs) para relacionar informações entre tabelas, criação de consultas complexas para relatórios e listagens, além da otimização de queries para melhor performance.

**Implementação no SQL:**

A implementação no SQL envolveu a escolha do MySQL como sistema gerenciador de banco de dados, criação de scripts DDL para definição das estruturas das tabelas, implementação de chaves primárias e estrangeiras, definição de constraints e índices, criação de ENUMs para tipos específicos e implementação de triggers e procedimentos armazenados quando necessário.

**Desenvolvimento do Projeto:**O desenvolvimento da API foi realizado utilizando Node.js com Express para criar uma API RESTful, implementando autenticação JWT para segurança, criação de middlewares para controle de acesso, desenvolvimento de controllers para cada entidade principal, documentação da API usando Swagger e implementação de validações e tratamento de erros.

**Pesquisa Bibliográfica:**

A pesquisa bibliográfica incluiu o estudo de melhores práticas em desenvolvimento de APIs REST, pesquisa sobre padrões de projeto em Node.js, análise de documentações técnicas do MySQL e Express, investigação sobre sistemas similares de gerenciamento de roteiros turísticos e estudo sobre segurança em aplicações web e proteção de dados.

# 4. DISCUSSÃO E REQUISITOS DO SISTEMA

O sistema Voyage foi projetado para atender às necessidades específicas de usuários que buscam planejar viagens de forma eficiente e personalizada. A seguir, são apresentados os principais requisitos funcionais e não funcionais do sistema, organizados por categorias:

## 4.1 Requisitos Funcionais

**Gestão de Usuários:**

* Cadastro e autenticação de usuários
* Gerenciamento de perfis (viajantes e guias turísticos)
* Atualização de informações pessoais
* Sistema de recuperação de senha
* Histórico de viagens e roteiros

**Gestão de Roteiros:**

* Criação de roteiros personalizados
* Adição e remoção de destinos
* Planejamento de cronograma
* Cálculo de custos estimados
* Compartilhamento de roteiros
* Salvamento de rascunhos

**Gestão de Destinos e Passeios:**

* Cadastro de destinos turísticos
* Registro de passeios e atividades
* Informações sobre disponibilidade
* Sistema de reservas
* Registro de preços e horários
* Descrições detalhadas e categorização

**Interação com Guias:**

* Perfis profissionais de guias turísticos
* Sistema de avaliações e comentários
* Agendamento de serviços
* Comunicação direta entre usuários e guias
* Registro de especialidades e áreas de atuação

**Sistema de Notificações:**

* Alertas de reservas
* Confirmações de agendamentos
* Lembretes de viagens
* Atualizações de status
* Mensagens do sistema

## 4.2 Requisitos Não Funcionais

**Desempenho:**

* Tempo de resposta rápido para consultas
* Capacidade de processamento simultâneo
* Otimização de consultas ao banco de dados
* Escalabilidade do sistema

**Segurança:**

* Criptografia de dados sensíveis
* Controle de acesso por níveis de usuário
* Proteção contra injeção SQL
* Backup regular dos dados
* Conformidade com LGPD

**Usabilidade:**

* Interface intuitiva
* Responsividade
* Facilidade de navegação
* Acessibilidade
* Suporte multilíngue

**Confiabilidade:**

* Alta disponibilidade do sistema
* Tolerância a falhas
* Consistência dos dados
* Recuperação de erros

# 5. MODELO ENTIDADE-RELACIONAMENTO

Este tópico aborda a estruturação do banco de dados do Voyage, com a criação de um Diagrama Entidade-Relacionamento (DER). O objetivo deste modelo é representar de maneira clara e organizada as entidades centrais do sistema, como usuários, profissionais turísticos, roteiros, atividades, destinos e avaliações, bem como as relações entre essas entidades.

A finalidade é garantir que todas as interações entre as informações estejam corretamente estabelecidas, evitando redundâncias e inconsistências no banco de dados. O DER será a base para o desenvolvimento da estrutura de dados, proporcionando uma visão detalhada de como as informações serão armazenadas, acessadas e manipuladas dentro do sistema. Ademais, o diagrama assegura que os dados estejam interconectados de forma lógica e eficiente, atendendo tanto às necessidades dos usuários quanto dos profissionais turísticos, para garantir uma experiência fluida e funcional na plataforma.

## 5.1 Diagrama Entidade-Relacionamento

O Diagrama Entidade-Relacionamento (DER), ilustrado a seguir, foi desenvolvido para representar as interações e o fluxo de dados entre os usuários e os profissionais turísticos, como guias e prestadores de serviços. O diagrama exibe as principais entidades do sistema e suas relações, ao definir claramente essas relações, o sistema torna mais fácil a criação e gestão de roteiros personalizados, o acompanhamento das avaliações de serviços, o registro de atividades e a comunicação entre os usuários e os profissionais turísticos.

Essa modelagem tem como objetivo proporcionar uma visão clara e detalhada da estrutura do banco de dados, assegurando que todos os componentes do Voyage estejam interconectados de forma lógica e funcional, facilitando a gestão de dados e melhorando a experiência do usuário.

<https://drive.google.com/file/d/1bFgZ_25-CoJYY4BZWEiesWhYdFJ3PH1V/view?usp=sharing>

# 

# 6. MODELO RELACIONAL NORMALIZADO E CONSULTAS

6.1 Modelo Relacional Normalizado

É uma representação de dados em um banco de dados relacional que segue as regras de normalização, com o objetivo de reduzir a redundância de dados, evitar anomalias de atualização e garantir a integridade do banco.

Normalização:

**PESSOA**(id, nome, cpf, email, telefone, data\_nascimento, created\_at, updated\_at)

**AUTH**( id, pessoa\_id, password, role, is\_active, last\_login, created\_at, updated\_at)

pessoa\_id referencia PESSOA(id)

**ENDERECO**(id, cep, pais, estado, cidade, bairro, rua, numero, complemento, created\_at, updated\_at)

**GUIA**(id, pessoa\_id, endereco\_id, biografia, anos\_experiencia, documento\_verificacao, status\_verificacao, avaliacao\_media, numero\_avaliacoes, created\_at, updated\_at)

pessoa\_id referencia PESSOA(id)

endereco\_id referencia ENDERECO(id)

**AREA\_ATUACAO**(id, nome, descricao, created\_at)

**GUIA\_AREA\_ATUACAO**( guia\_id, area\_id)

guia\_id referencia GUIA(id)

area\_id referencia AREA\_ATUACAO(id)

**IDIOMA**( id, nome, codigo)

**GUIA\_IDIOMA**(guia\_id, idioma\_id, nivel)

guia\_id referencia GUIA(id)

idioma\_id referencia IDIOMA(id)

**DESTINO**(id, nome, estado, cidade, descricao, latitude, longitude, created\_at, updated\_at)

**PASSEIO**(id, nome, descricao, preco, duracao\_horas, nivel\_dificuldade, inclui\_refeicao, inclui\_transporte, destino\_id, guia\_id, capacidade\_maxima, created\_at, updated\_at)

destino\_id referencia DESTINO(id)

guia\_id referencia GUIA(id)

**ROTEIRO**(id, passeio\_id, data, hora\_inicio, hora\_fim, status, vagas\_disponiveis, created\_at, updated\_at)

passeio\_id referencia PASSEIO(id)

**RESERVA**(id, roteiro\_id, pessoa\_id, quantidade\_pessoas, valor\_total, status, data\_reserva, created\_at, updated\_at)

roteiro\_id referencia ROTEIRO(id)

pessoa\_id referencia PESSOA(id)

**AVALIACAO**(id, reserva\_id, nota, comentario, created\_at)

reserva\_id referencia RESERVA(id)

**NOTIFICACAO**(id, pessoa\_id, titulo, mensagem, tipo, status, data\_envio, created\_at)

pessoa\_id referencia PESSOA(id)

6.2 Consultas no Banco de Dados

As consultas abaixo foram realizadas para recuperar informações do banco de dados da aplicação, cada comando é específico para acessar determinada informação dentro do database. Além disso, cada consulta foi utilizada para extrair informações das tabelas de acordo com critérios determinados.

1. Listar todos os usuários com suas informações pessoais:

SELECT

p.id,

p.nome,

p.email,

p.telefone,

p.cpf,

p.data\_nascimento,

p.created\_at,

CASE

WHEN g.id IS NOT NULL THEN 'guia'

ELSE 'viajante'

END as tipo,

a.role,

a.is\_active,

a.last\_login

FROM PESSOA p

JOIN AUTH a ON p.id = a.pessoa\_id

LEFT JOIN GUIA g ON p.id = g.pessoa\_id

ORDER BY p.nome

1. Listar todos os passeios disponíveis:

SELECT

p.\*,

d.nome as destino\_nome,

d.cidade,

d.estado,

ps.nome as guia\_nome,

ps.email as guia\_email

FROM PASSEIO p

JOIN DESTINO d ON p.destino\_id = d.id

JOIN GUIA g ON p.guia\_id = g.id

JOIN PESSOA ps ON g.pessoa\_id = ps.id

WHERE 1=1

1. Consultar todos os roteiros agendados para um determinado usuário (por idusuario):

SELECT

r.\*,

p.nome as passeio\_nome,

p.descricao como passeio\_descricao,

p.preco,

p.duracao\_horas,

p.nivel\_dificuldade,

p.inclui\_refeicao,

p.inclui\_transporte,

d.nome as destino\_nome,

d.cidade,

d.estado,

g.id as guia\_id,

ps.nome as guia\_nome,

pc.nome as criador\_nome,

COALESCE(ar.media\_avaliacoes, 0) as avaliacao\_media,

COALESCE(ar.total\_avaliacoes, 0) as total\_avaliacoes

FROM ROTEIRO r

JOIN PASSEIO p ON r.passeio\_id = p.id

JOIN DESTINO d ON p.destino\_id = d.id

JOIN GUIA g ON p.guia\_id = g.id

JOIN PESSOA ps ON g.pessoa\_id = ps.id

JOIN PESSOA pc ON r.criador\_id = pc.id

LEFT JOIN (

SELECT

roteiro\_id,

AVG(nota) as media\_avaliacoes,

COUNT(\*) as total\_avaliacoes

FROM AVALIACAO\_ROTEIRO

GROUP BY roteiro\_id

) ar ON r.id = ar.roteiro\_id

WHERE r.criador\_id = ? AND r.status = 'agendado'

1. Listar todos os guias e suas áreas de atuação, incluindo endereço:

\*SELECT

g.id AS guia\_id,

p.nome AS nome\_guia,

ea.nome AS area\_atuacao,

e.rua, e.numero, e.bairro, e.cidade, e.estado, e.cep

FROM

GUIA g

JOIN

PESSOA p ON g.pessoa\_id = p.id

JOIN

ENDERECO e ON g.endereco\_id = e.id

JOIN

GUIA\_AREA\_ATUACAO ga ON g.id = ga.guia\_id

JOIN

AREA\_ATUACAO ea ON ga.area\_id = ea.id;

1. Contar o número total de passeios oferecidos por cidade:

SELECT

d.cidade,

COUNT(p.id) AS total\_passeios

FROM

DESTINO d

JOIN

PASSEIO p ON d.id = p.destino\_id

GROUP BY

d.cidade;

1. Listar todos os destinos com passeios disponíveis acima de um determinado preço:

SELECT

d.nome AS destino,

d.cidade,

d.estado,

p.nome AS passeio,

p.preco

FROM

DESTINO d

JOIN

PASSEIO p ON d.id = p.destino\_id

WHERE

p.preco > X;

1. Listar todos os roteiros e os respectivos guias por data de passeio:

SELECT

r.data,

p.nome AS nome\_guia,

pa.nome AS nome\_passeio,

r.hora\_inicio,

r.hora\_fim

FROM

ROTEIRO r

JOIN

PASSEIO pa ON r.passeio\_id = pa.id

JOIN

GUIA g ON pa.guia\_id = g.id

JOIN

PESSOA p ON g.pessoa\_id = p.id

ORDER BY

r.data, r.hora\_inicio;

1. Listar todos os usuários com pelo menos um roteiro agendado:

SELECT DISTINCT

p.id AS usuario\_id,

p.nome AS nome\_usuario,

p.email

FROM

PESSOA p

JOIN

RESERVA r ON p.id = r.pessoa\_id

JOIN

ROTEIRO ro ON r.roteiro\_id = ro.id

WHERE

ro.status IN ('agendado', 'confirmado');

1. Obter todas as notificações de um determinado roteiro:

SELECT NOTIFICACAO.id AS NotificacaoID, ROTEIRO.codigo\_roteiro, ROTEIRO.data, ROTEIRO.hora

FROM NOTIFICACAO

INNER JOIN ROTEIRO ON NOTIFICACAO.roteiroID = ROTEIRO.codigo\_roteiro

WHERE ROTEIRO.codigo\_roteiro = 1;

1. Obter números de notificações enviadas por roteiro:

SELECT ROTEIRO.codigo\_roteiro, COUNT(NOTIFICACAO.id) AS total\_notificacoes

FROM ROTEIRO

LEFT JOIN NOTIFICACAO ON ROTEIRO.codigo\_roteiro = NOTIFICACAO.roteiroID

GROUP BY ROTEIRO.codigo\_roteiro;

# 7. DICIONÁRIO DE DADOS

Esta seção apresenta a descrição detalhada de cada método presente nas classes do sistema, incluindo seus parâmetros, retornos e principais responsabilidades. Cada subseção corresponde a uma classe específica e seus respectivos métodos.

### 

### Tabela 1 - Dicionário de Dados de Pessoa

| Tabela | PESSOA | | | |
| --- | --- | --- | --- | --- |
| Descrição | Tabela que armazena os dados das pessoas para serem acessados nas tabelas de usuários ou guias. | | | |
|  | Campos | | | |
| Nome | Descrição | Tipo de dado | Tamanho | Restrições de integridade |
| id | identificador único para cada pessoa | Int | 10 | Não nulo, único, chave primária |
| nome | Nome completo da pessoa | Varchar | 100 | Não nulo |
| cpf | Número para identificação de cada pessoa | Varchar | 11 | Não nulo, único |
| email | Endereço de email | Varchar | 100 | Não nulo, único |
| telefone | Número de telefone | Varchar | 15 | Opcional |
| data\_nascimento | Data de nascimento | Date | ————— | Opcional |
| created\_at | Data de criação do registro | TIMESTAMP | ————— | Não nulo, valor padrão: CURRENT\_TIMESTAMP |
| updated\_at | Data de atualização do registro | TIMESTAMP | ————— | Atualizado automaticamente |

### Tabela 2 - Dicionário de Dados de Usuário

| Tabela | AUTH | | | |
| --- | --- | --- | --- | --- |
| Descrição | Tabela para autenticação e controle de acesso dos usuários do sistema | | | |
|  | Campos | | | |
| Nome | Descrição | Tipo de dado | Tamanho | Restrições de integridade |
| id | Identificador único da autenticação | INT | 10 | Não nulo, único, chave primária |
| pessoa\_id | Referência ao ID da pessoa | INT | 10 | Não nulo, chave estrangeira, referência PESSOA(id) |
| password | Senha criptografada | VARCHAR | 255 | Não nulo |
| role | Papel do usuário (ex: admin, user) | VARCHAR | 50 | Não nulo, valor padrão: 'user' |
| is\_active | Indica se a conta está ativa | BOOLEAN | ————— | Valor padrão: true |
| last\_login | Última data de login | TIMESTAMP | ————— | Opcional |
| created\_at | Data de criação do registro | TIMESTAMP | ————— | Valor padrão: CURRENT\_TIMESTAMP |
| updated\_at | Data de atualização do registro | TIMESTAMP | ————— | Atualizado automaticamente |

### Tabela 3 - Dicionário de Dados de Endereço

| Tabela | ENDERECO | | | |
| --- | --- | --- | --- | --- |
| Descrição | Tabela que armazena os dados de endereço | | | |
|  | Campos | | | |
| Nome | Descrição | Tipo de dado | Tamanho | Restrições de integridade |
| id | Identificador único de endereço | Int | 10 | Não nulo, único, chave primária |
| cep | Código de endereçamento postal | Varchar | 8 | Não nulo |
| pais | Nome do país | VARCHAR | 100 | Valor padrão: 'Brasil' |
| estado | Sigla do estado | Varchar | 2 | Não nulo |
| cidade | Nome da cidade | Varchar | 100 | Não nulo |
| bairro | Nome do bairro | VARCHAR | 100 | Opcional |
| rua | Nome da rua | Varchar | 100 | Não nulo |
| numero | Número da residência | Varchar | 10 | Não nulo |
| complemento | Complemento de endereço | Varchar | 100 | Opcional |
| created\_at | Data de criação do registro | TIMESTAMP | ————— | Valor padrão: CURRENT\_TIMESTAMP |
| updated\_at | Data de atualização do registro | TIMESTAMP | ————— | Atualizado automaticamente |

### Tabela 4 - Dicionário de Dados de Guia

| Tabela | GUIA | | | |
| --- | --- | --- | --- | --- |
| Descrição | Tabela que armazena os dados dos usuários cadastrados como guias no sistema | | | |
|  | Campos | | | |
| Nome | Descrição | Tipo de dado | Tamanho | Restrições de integridade |
| id | Referência ao ID da pessoa | INT | 10 | Não nulo, chave primaria |
| pessoa\_id | identificador único para cada pessoa | Int | 10 | Chave estrangeira, referência PESSOA(id) |
| endereco\_id | Referência ao ID do endereço | INT | 10 | Não nulo, chave estrangeira, referência ENDERECO(id) |
| biografia | Texto descritivo sobre o guia | TEXT | ————— | Opcional |
| anos\_experiencia | Quantidade de anos de experiência | INT | ————— | Opcional |
| avaliacao\_media | Nota média das avaliações recebidas | DECIMAL | 3,2 | Valor padrão: 0.00 |
| numero\_avaliacoes | Quantidade de avaliações recebidas | INT | ————— | Valor padrão: 0 |
| documento\_verificacao | Documento para verificação | VARCHAR | 255 | Opcional |
| status\_verificacao | Status da verificação (pendente/verificado/rejeitado) | ENUM | ————— | Valor padrão: 'pendente' |
| created\_at | Data de criação do registro | TIMESTAMP | ————— | Valor padrão: CURRENT\_TIMESTAMP |
| updated\_at | Data de atualização do registro | TIMESTAMP | ————— | Atualizado automaticamente |

### Tabela 5 - Dicionário de Dados de Destino

| Tabela | DESTINO | | | |
| --- | --- | --- | --- | --- |
| Descrição | Tabela que armazena os dados dos destinos cadastrados no sistema pelos usuários | | | |
|  | Campos | | | |
| Nome | Descrição | Tipo de dado | Tamanho | Restrições de integridade |
| id | Identificador único do destino | INT | 10 | Não nulo, único, chave primária |
| nome | Nome do destino | VARCHAR | 100 | Não nulo |
| estado | Sigla do estado | VARCHAR | 2 | Não nulo |
| cidade | Nome da cidade | VARCHAR | 100 | Não nulo |
| descricao | Descrição do destino | TEXT | ————— | Opcional |
| latitude | Coordenada de latitude | DECIMAL | 10, 8 | Opcional |
| longitude | Coordenada de longitude | DECIMAL | 11, 8 | Opcional |
| created\_at | Data de criação do registro | TIMESTAMP | ————— | Valor padrão: CURRENT\_TIMESTAMP |
| updated\_at | Data de atualização do registro | TIMESTAMP | ————— | Atualizado automaticamente |

### Tabela 6 - Dicionário de Dados de Passeio

| Tabela | PASSEIO | | | |
| --- | --- | --- | --- | --- |
| Descrição | Tabela que armazena os dados dos passeios disponíveis para fazer em um determinado destino. | | | |
|  | Campos | | | |
| Nome | Descrição | Tipo de dado | Tamanho | Restrições de integridade |
| id | Identificador único do passeio | INT | 10 | Não nulo, único, chave primária |
| nome | Nome do passeio | VARCHAR | 100 | Não nulo |
| descricao | Descrição do passeio | TEXT | - | Opcional |
| preco | Preço por pessoa | DECIMAL | 10, 2 | Não nulo |
| duracao\_horas | Duração em horas | INT | ————— | Opcional |
| nivel\_dificuldade | Nível de dificuldade | ENUM | ————— | Valor padrão: 'facil' |
| inclui\_refeicao | Indica se o passeio inclui refeição | BOOLEAN | ————— | Valor padrão: false |
| inclui\_transporte | Indica se o passeio inclui transporte | BOOLEAN | ————— | Valor padrão: false |
| destino\_id | Referência ao ID do destino | INT | 10 | Não nulo, chave estrangeira, referência DESTINO(id) |
| guia\_id | Referência ao ID do guia | INT | 10 | Não nulo, chave estrangeira, referência GUIA(id) |
| capacidade\_maxima | Capacidade máxima de participantes | INT | ————— | Opcional |
| created\_at | Data de criação do registro | TIMESTAMP | ————— | Valor padrão: CURRENT\_TIMESTAMP |
| updated\_at | Data de atualização do registro | TIMESTAMP | ————— | Atualizado automaticamente |

### Tabela 7 - Dicionário de Dados de Roteiro

| Tabela | ROTEIRO | | | |
| --- | --- | --- | --- | --- |
| Descrição | Tabela que armazena os dados dos roteiros planejados para determinado destino com passeios selecionados | | | |
|  | Campos | | | |
| Nome | Descrição | Tipo de dado | Tamanho | Restrições de integridade |
| id | Identificador único do roteiro | INT | 10 | Não nulo, único, chave primária |
| passeio\_id | Referência ao ID do passeio | INT | 10 | Não nulo, chave estrangeira, referência PASSEIO(id) |
| data | Data do roteiro | DATE | ————— | Não nulo |
| hora\_inicio | Hora de início do roteiro | TIME | ————— | Não nulo |
| hora\_fim | Hora de término do roteiro | TIME | ————— | Não nulo |
| status | Status do roteiro | ENUM | ————— | Valor padrão: 'agendado' |
| vagas\_disponiveis | Quantidade de vagas disponíveis | INT | ————— | Opcional |
| created\_at | Data de criação do registro | TIMESTAMP | ————— | Valor padrão: CURRENT\_TIMESTAMP |
| updated\_at | Data de atualização do registro | TIMESTAMP | ————— | Atualizado automaticamente |

### Tabela 8 - Dicionário de Dados de Notificação

| Tabela | NOTIFICACAO | | | |
| --- | --- | --- | --- | --- |
| Descrição | Tabela que armazena os dados notificações relacionadas ao roteiro, cada notificação está relacionado a um roteiro específico, informando detalhes dos passeios adicionados no roteiro | | | |
|  | Campos | | | |
| Nome | Descrição | Tipo de dado | Tamanho | Restrições de integridade |
| id | Identificador único da notificação | INT | 10 | Não nulo, único, chave primária |
| pessoa\_id | Referência ao ID da pessoa | INT | 10 | Não nulo, chave estrangeira, referência PESSOA(id) |
| titulo | Título da notificação | VARCHAR | 100 | Não nulo |
| mensagem | Conteúdo da notificação | TEXT | ————— | Não nulo |
| tipo | Tipo da notificação | ENUM | ————— | Valores: 'sistema', 'reserva', 'avaliacao', 'cancelamento'; não nulo |
| status | Status de leitura da notificação | ENUM | ————— | Valores: 'nao\_lida', 'lida'; padrão: 'nao\_lida' |
| data\_envio | Data e hora do envio da notificação | TIMESTAMP | ————— | Valor padrão: CURRENT\_TIMESTAMP |
| created\_at | Data de criação do registro | TIMESTAMP | ————— | Valor padrão: CURRENT\_TIMESTAMP |

# Tabela 9 - Dicionário de Dados de Área de Atuação

| Tabela | AREA\_ATUACAO | | | |
| --- | --- | --- | --- | --- |
| Descrição | Tabela que armazena as áreas de atuação disponíveis para guias turísticos. | | | |
|  | Campos | | | |
| Nome | Descrição | Tipo de dado | Tamanho | Restrições de integridade |
| id | Identificador único da área de atuação | INT | 10 | Não nulo, único, chave primária |
| nome | Nome da área de atuação | VARCHAR | 100 | Não nulo |
| descricao | Descrição da área | TEXT | ————— | Opcional |
| created\_at | Data de criação do registro | TIMESTAMP | ————— | Valor padrão: CURRENT\_TIMESTAMP |

# Tabela 10 - Dicionário de Dados de Guia Área de Atuação

| Tabela | GUIA\_AREA\_ATUACAO | | | |
| --- | --- | --- | --- | --- |
| Descrição | Tabela que relaciona guias turísticos com suas respectivas áreas de atuação. | | | |
|  | Campos | | | |
| Nome | Descrição | Tipo de dado | Tamanho | Restrições de integridade |
| guia\_id | Referência ao ID do guia | INT | 10 | Não nulo, chave estrangeira, referência GUIA(id) |
| area\_id | Referência ao ID da área de atuação | INT | 10 | Não nulo, chave estrangeira, referência AREA(id) |
| PRIMARY KEY | Combinação dos campos guia\_id e area\_id | ————— | ————— | Não nulo |

# Tabela 11 - Dicionário de Dados de Idioma

| Tabela | IDIOMA | | | |
| --- | --- | --- | --- | --- |
| Descrição | Tabela que armazena os idiomas disponíveis para guias. | | | |
|  | Campos | | | |
| Nome | Descrição | Tipo de dado | Tamanho | Restrições de integridade |
| id | Identificador único | INT | 10 | Não nulo, único, chave primária |
| nome | Nome do idioma | VARCHAR | 50 | Não nulo |
| codigo | Código abreviado do idioma | VARCHAR | 5 | Não nulo |

# Tabela 12 - Dicionário de Dados de Idioma

| Tabela | GUIA\_IDIOMA | | | |
| --- | --- | --- | --- | --- |
| Descrição | Tabela que relaciona guias turísticos com os idiomas que falam. | | | |
|  | Campos | | | |
| Nome | Descrição | Tipo de dado | Tamanho | Restrições de integridade |
| guia\_id | Referência ao ID do guia | INT | 10 | Não nulo, chave estrangeira, referência GUIA(id) |
| idioma\_id | Referência ao ID do idioma | INT | 10 | Não nulo, chave estrangeira, referência IDIOMA(id) |
| nivel | Nível de proficiência no idioma | ENUM | ————— | Não nulo (valores: 'basico', 'intermediario', 'avancado', 'fluente') |
| PRIMARY KEY | Combinação dos campos guia\_id e idioma\_id | ————— | ————— | Não nulo |

# Tabela 13 - Dicionário de Dados de Reserva

| Tabela | RESERVA | | | |
| --- | --- | --- | --- | --- |
| Descrição | Tabela que registra as reservas realizadas por pessoas para roteiros turísticos. | | | |
|  | Campos | | | |
| Nome | Descrição | Tipo de dado | Tamanho | Restrições de integridade |
| id | Identificador único da reserva | INT | 10 | Não nulo, único, chave primária |
| roteiro\_id | Referência ao ID do roteiro | INT | 10 | Não nulo, chave estrangeira, referência ROTEIRO(id) |
| pessoa\_id | Referência ao ID da pessoa | INT | 10 | Não nulo, chave estrangeira, referência PESSOA(id) |
| quantidade\_pessoas | Número de pessoas incluídas na reserva | INT | ————— | Valor padrão: 1, não nulo |
| valor\_total | Valor total da reserva | DECIMAL | 10, 2 | Não nulo |
| status | Status da reserva | ENUM | ————— | Valores: 'pendente', 'confirmada', 'cancelada'; padrão: 'pendente' |
| data\_reserva | Data e hora da reserva | TIMESTAMP | ————— | Valor padrão: CURRENT\_TIMESTAMP |
| created\_at | Data de criação do registro | TIMESTAMP | ————— | Valor padrão: CURRENT\_TIMESTAMP |
| updated\_at | Data de atualização do registro | TIMESTAMP | ————— | Atualizado automaticamente |

# Tabela 14 - Dicionário de Dados de Avaliação

| Tabela | GUIA\_IDIOMA | | | |
| --- | --- | --- | --- | --- |
| Descrição | Tabela que armazena as avaliações feitas por usuários sobre suas reservas. | | | |
|  | Campos | | | |
| Nome | Descrição | Tipo de dado | Tamanho | Restrições de integridade |
| id | Identificador único da avaliação | INT | 10 | Não nulo, único, chave primária |
| reserva\_id | Referência ao ID da reserva | INT | 10 | Não nulo, chave estrangeira, referência RESERVA(id) |
| nota | Nota da avaliação | DECIMAL | 3, 2 | Não nulo (valor entre 0 e 10) |
| comentario | Comentário opcional do avaliador | TEXT | ————— | Opcional |

# 

# 8. SISTEMA E CONEXÃO COM O MYSQL

O Voyage é um sistema dividido em duas principais camadas: Front-End e Back-End, integradas por uma API RESTful que gerencia a comunicação com o banco de dados MySQL. Este tópico descreve as tecnologias e abordagens utilizadas para desenvolver o sistema e estabelecer a conexão com o banco de dados.

## 8.1 Estrutura do Sistema

**Front-End**:  
 Desenvolvido em React com TypeScript, o front-end é responsável pela interface gráfica, garantindo que os usuários possam interagir de forma intuitiva e responsiva. Ele consome os serviços da API para exibir informações de roteiros, destinos, guias turísticos, e gerenciar reservas.

**Back-End**:  
 Implementado em Node.js utilizando o framework Express, o back-end serve como intermediário entre o front-end e o banco de dados. Sua arquitetura modular permite gerenciar autenticação, regras de negócio, e rotas para as diversas funcionalidades da aplicação.

**Banco de Dados (MySQL)**:  
 Utiliza-se um banco de dados relacional, estruturado com base em um Modelo Entidade-Relacionamento (MER) e normalizado para evitar redundâncias e garantir a integridade dos dados.

## 8.2 Integração com o MySQL

A conexão entre o back-end e o MySQL é estabelecida com o uso da biblioteca mysql2. A seguir, estão os principais passos para essa integração:

**Configuração:**

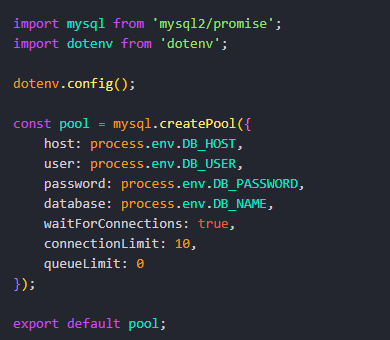
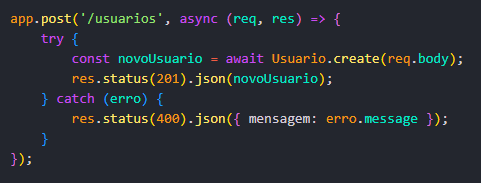
O arquivo de configuração inclui as credenciais de acesso ao banco de dados, como host, usuário, senha e nome do banco. Esses dados são armazenados em variáveis de ambiente para maior segurança.

Imagem 1 - Conectando banco de dados na API do sistema

**Definição e modelos:**

Definição de Modelos: No back-end, cada entidade do banco de dados (por exemplo, Pessoa, Roteiro, Guia) é representada por um modelo, utilizando frameworks ORM como o Sequelize. Isso facilita a manipulação de dados, permitindo que operações complexas sejam realizadas com comandos simples.

Métodos CRUD: As rotas da API incluem métodos de Create, Read, Update e Delete para cada entidade, como o exemplo abaixo para a criação de um novo usuário:

Imagem 2 - Criação de rotas na API do sistema

**Consultas e Relatórios:**

Consultas complexas, como junções e agregações, foram implementadas para atender aos requisitos do sistema, como relatórios de reservas e avaliações.

## 8.3 Segurança e Escalabilidade / Autenticação e Autorização

A API utiliza autenticação JWT (JSON Web Token) para gerenciar sessões de usuários, garantindo que apenas usuários autorizados acessem os recursos.

**Otimização:**

Consultas SQL foram otimizadas com o uso de índices e técnicas de cache para melhorar o desempenho em operações intensivas.

**Escalabilidade:**

O sistema foi projetado para ser escalável, permitindo a adição de novas funcionalidades e a integração com outros serviços no futuro.

## 8.4 Funcionamento Geral Quando um usuário interage com a interface do Voyage, o front-end faz requisições para a API, que processa a solicitação, interage com o banco de dados MySQL e retorna a resposta ao front-end. Essa arquitetura desacoplada facilita a manutenção e evolução do sistema, permitindo que cada camada seja atualizada independentemente.

## 8.5 Estrutura do Front-end

O front-end do Voyagee foi desenvolvido utilizando React com TypeScript, seguindo as melhores práticas de desenvolvimento e organização de código.

**Tecnologias utilizadas:**

* React: Biblioteca JavaScript para construção de interfaces de usuário.
* TypeScript: Superset de JavaScript que adiciona tipagem estática ao código.
* Tailwind CSS: Framework de CSS utilitário para estilização rápida e responsiva.
* Vite: Ferramenta de build rápida para desenvolvimento front-end.
* React Router: Biblioteca para gerenciamento de rotas no React.
* Axios: Biblioteca para fazer requisições HTTP.

# 

# 8.6 Capturas de tela do sistema

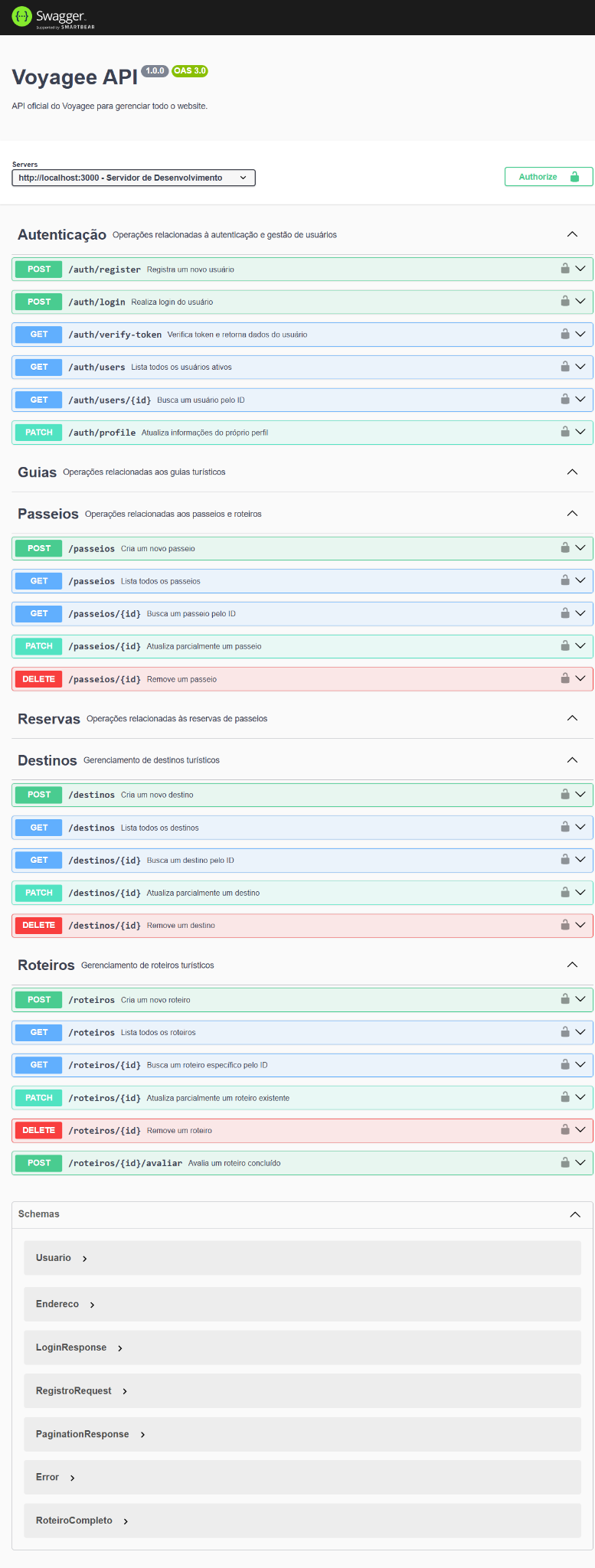
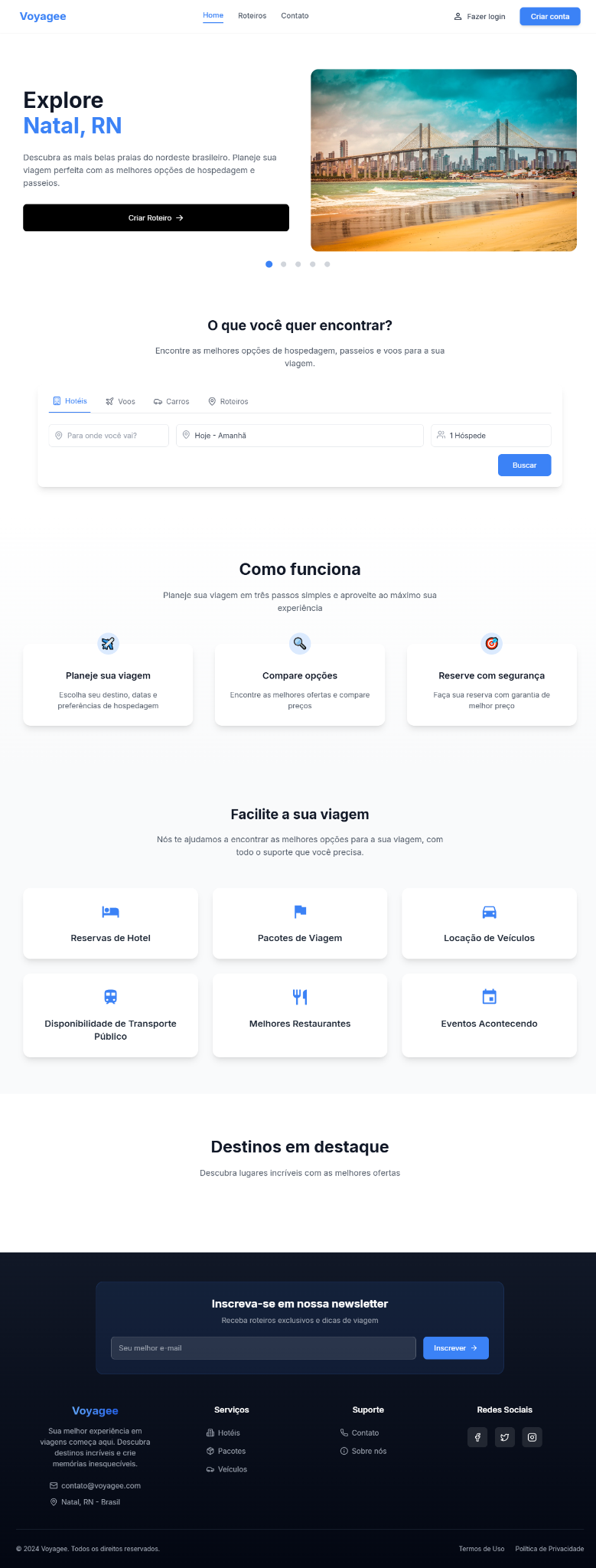
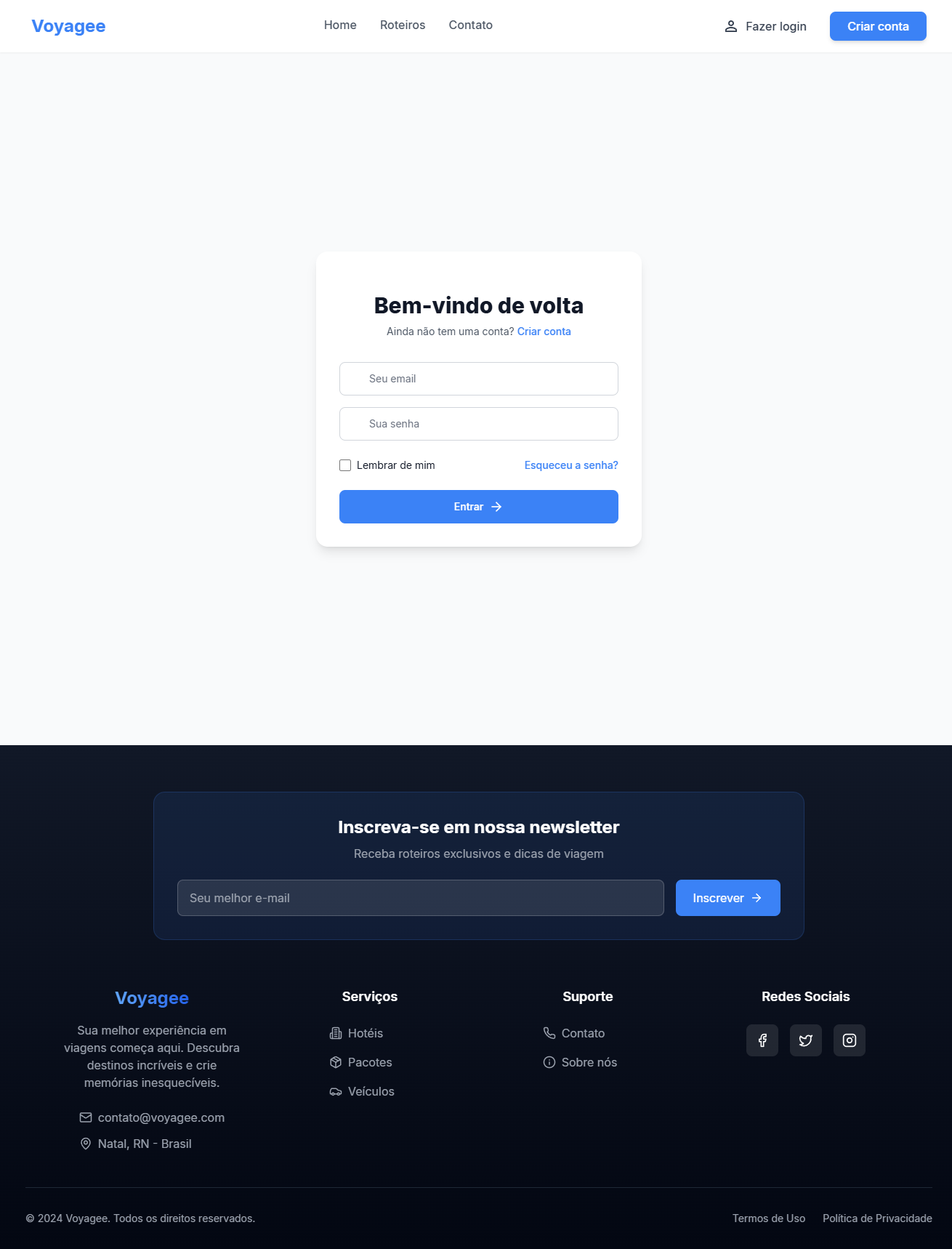


Imagem 3 - Swagger, interface da API com toda a documentação do sistema

Imagem 4 - Interface da página HOME do site



<https://drive.google.com/file/d/1GgWiAqwts-ttqwXmksFgVgpAS1XWQJ9S/view?usp=drive_link>

Imagem 5 - Interface da página de login do site

# Imagem 6 - Interface da página de registrar usuário do site

# <https://drive.google.com/file/d/1zr217_HTi1gaokRc8-r1D9kPArr_VLrv/view?usp=drive_link>

# 9. RESULTADOS

## 9.1 Resultados Iniciais

A implementação do banco de dados do Voyage foi realizada com êxito, permitindo uma integração harmoniosa entre as diversas funcionalidades do sistema. A estrutura deste banco foi concebida para assegurar a consistência das informações e minimizar redundâncias, garantindo assim a integridade dos dados. Através de uma modelagem meticulosa e da normalização, torna o acesso e a manipulação das informações dos utilizadores e profissionais turísticos mais simples e sem apresentar falhas.

## 9.2 Resultados Esperados

Com a conclusão do Voyage, espera-se que o planejamento de viagens se torne mais simples, personalizado e eficaz, permitindo que os usuários alcancem seus objetivos de otimização de tempo, praticidade e custo-benefício. Além disso, a plataforma visa facilitar a comunicação entre usuários e profissionais turísticos, agilizando o processo e promovendo passeios mais variados e viagens mais divertidas. A centralização de serviços como passeios, hotéis, guias e aluguel de automóveis em um único lugar, de forma remota, é um recurso que se espera seja altamente valorizado e amplamente utilizado pelos usuários.

# 10. CRONOGRAMA

O cronograma do projeto Voyage descreve as principais atividades para o desenvolvimento e entrega da plataforma. A seguir, estão os principais passos e suas respectivas datas:

### Tabela 15 - Tabela de Elaboração do Projeto

| **Atividades** | **Nov/24** | **Dez/24** |
| --- | --- | --- |
| 1. Definição de projetos | X |  |
| 1. Discussão e refinamento de requisitos | X |  |
| 1. Apresentação do MER | X |  |
| 1. Modelo Relacional Normalizado e definição de 10 a 20 consultas | X |  |
| 1. Apresentação das consultas SQL da etapa 4 | X |  |
| 1. Apresentação da interface do sistema | X |  |
| 1. Ajustes | X |  |
| 1. Ajustes | X |  |
| 1. Entrega da documentação |  | X |
| 1. Apresentação dos projetos |  | X |

# 11.REPOSITÓRIOS

[**https://github.com/FelipeFMedeiros/Voyagee**](https://github.com/FelipeFMedeiros/Voyagee)

[**https://github.com/FelipeFMedeiros/VoyageeAPI**](https://github.com/FelipeFMedeiros/VoyageeAPI)