

UNIBRASIL

SISTEMA DE VIAGEM CORPORATIVA

VITOR HUGO VIANA MONDEQUI

EDUARDO HENRIQUE ELER

PROFESSOR: RODRIGO NASCIMENTO

DISCIPLINA: PRÁTICA PROFISSIONAL EM PROGRAMAÇÃO WEB

CURITIBA- PR

**ABRIL DE 2025**

## Sumário

1. Introdução .....	2
2. Wireframes .....	4
3. Arquitetura de Software .....	4
4. Entidades, Atributos e Relacionamentos .....	5
5. Repositório GitHub .....	8
6. Delineamento de Desenvolvimento .....	8
7. Divisão da Equipe .....	8

## 1. Introdução

O presente projeto tem como objetivo desenvolver um sistema de gestão de viagens corporativas, com o intuito de otimizar e digitalizar os processos relacionados a deslocamentos, reservas e reembolsos de colaboradores em viagens a trabalho. O sistema proporcionará mais controle, segurança e agilidade aos processos internos das empresas.

Funcionalidades:

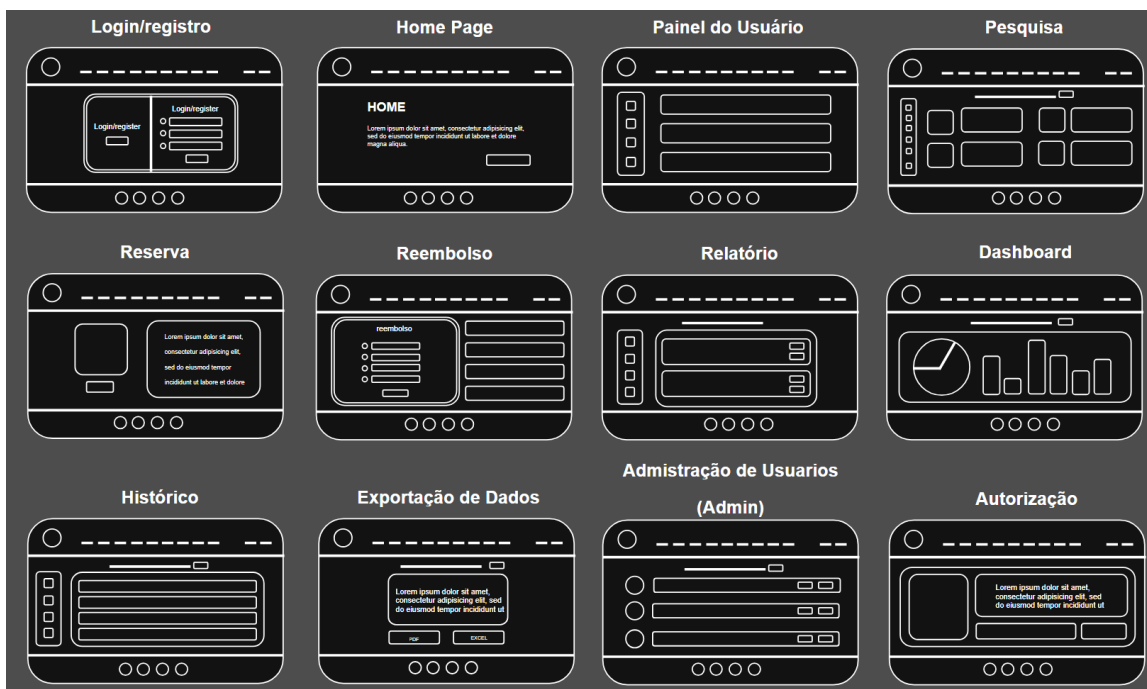
- Cadastro de usuários;
- Autenticação de autorização;
- Gerenciamento de viagem;
- Aprovação de viagem;
- Pesquisa e reserva de transporte e hospedagem;
- Reembolso;
- Relatório e Dashboard;
- Histórico de viagem;
- Exportação de dados.

Paginas:

- Home page;
- Login/Registro;
- Painel do usuário;
- Pesquisa de transporte e hospedagem;
- Reserva de transporte e hospedagem;
- Reembolso;
- Relatório;
- Dashboard;
- Histórico de viagem;
- Exportação de dados;
- Administração de usuário;
- Autorização de viagem;

## 2. Wireframes

Nesta seção, apresentamos os wireframes das principais telas do sistema, como: Tela de Login, Cadastro de Viagens, Reserva, Reembolso, Dashboard, entre outras.



## 3. Arquitetura de Software

A arquitetura utilizada é baseada no padrão MVC (Model-View-Controller), com o frontend construído em HTML, CSS e JavaScript, e o backend utilizando PHP e MySQL.

A arquitetura MVC (Model-View-Controller) foi escolhida para o desenvolvimento do **Sistema de Viagem Corporativa** devido à sua clara separação de responsabilidades, que facilita a manutenção, escalabilidade e flexibilidade do sistema. O padrão MVC organiza o código em três componentes principais:

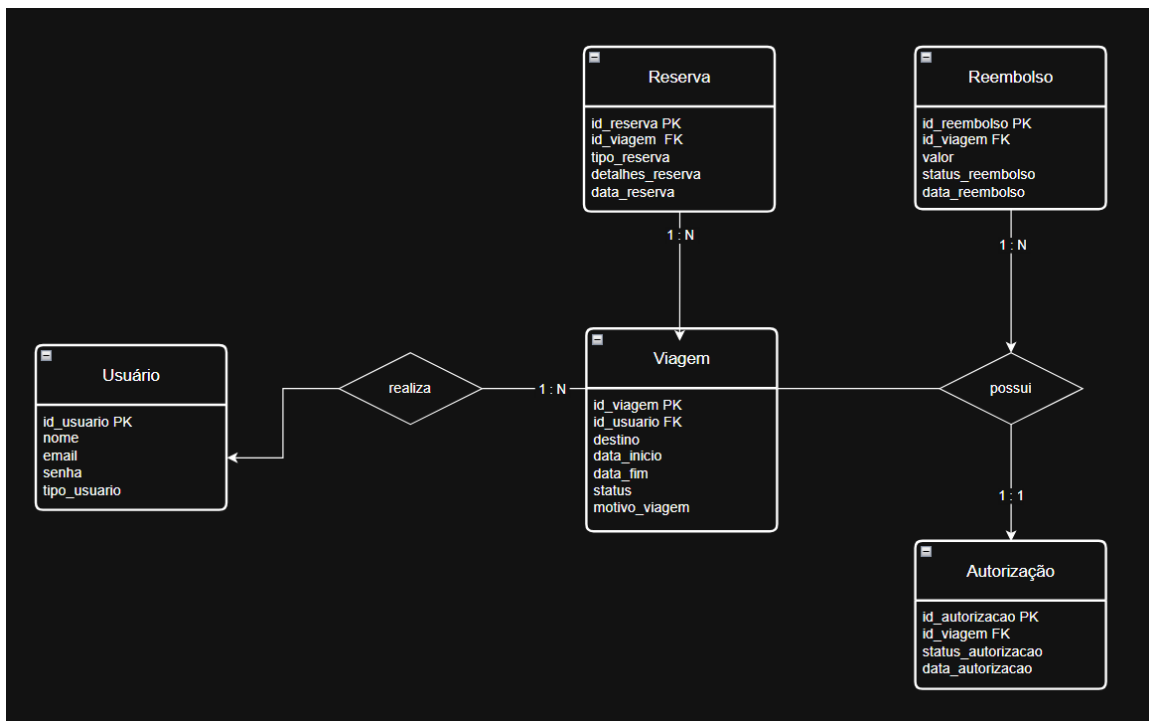
1. **Model:** Representa a lógica de dados e as regras de negócios do sistema. O Model é responsável por interagir com o banco de dados, recuperar e manipular dados, e garantir a integridade dessas informações.
2. **View:** Refere-se à interface do usuário, ou seja, os elementos visuais com os quais o usuário interage. A View é responsável por exibir os dados recebidos do Model de maneira intuitiva e agradável.

3. **Controller:** Atua como intermediário entre o Model e a View. O Controller processa as entradas do usuário, atualiza o Model e, em seguida, escolhe a View apropriada para exibir os resultados.

A principal vantagem do uso do MVC é que ele permite uma organização modular do código, onde a lógica de negócios, a interface do usuário e o controle das interações ficam separados. Isso facilita a manutenção do sistema, já que qualquer alteração na interface (View) ou na lógica de dados (Model) pode ser feita de maneira independente, sem afetar as outras partes. Além disso, a reutilização de código se torna mais simples, pois cada componente tem sua própria responsabilidade bem definida.

#### 4. Entidades, Atributos e Relacionamentos

A modelagem de dados envolve entidades como Usuário, Viagem, Reserva, Reembolso, entre outras, com seus respectivos atributos e relacionamentos. O DER será apresentado com as ligações entre essas tabelas.



## Entidades e Atributos:

### 1. Usuário

#### a. Atributos:

- i. id\_usuario (PK)
- ii. nome
- iii. email
- iv. senha
- v. tipo\_usuario (Ex: Administrador, Funcionário)

### 2. Viagem

#### a. Atributos:

- i. id\_viagem (PK)
- ii. id\_usuario (FK - referente ao Usuário)
- iii. destino
- iv. data\_inicio
- v. data\_fim
- vi. status (Ex: Aprovada, Pendente, Finalizada)
- vii. motivo\_viagem

### 3. Reserva

#### a. Atributos:

- i. id\_reserva (PK)
- ii. id\_viagem (FK - referente à Viagem)
- iii. tipo\_reserva (Ex: Hospedagem, Transporte)
- iv. detalhes\_reserva
- v. data\_reserva

### 4. Reembolso

#### a. Atributos:

- i. id\_reembolso (PK)
- ii. id\_viagem (FK - referente à Viagem)
- iii. valor
- iv. status\_reembolso (Ex: Pendente, Pago)
- v. data\_reembolso

### 5. Autorização

a. Atributos:

- i. id\_autorizacao (PK)
- ii. id\_viagem (FK - referente à Viagem)
- iii. status\_autorizacao (Ex: Aprovada, Rejeitada)
- iv. data\_autorizacao

### Relacionamentos:

- Um **Usuário** pode ter várias **Viagens** (Relacionamento: 1:N)
- Uma **Viagem** pode ter várias **Reservas** (Relacionamento: 1:N)
- Uma **Viagem** pode ter um ou mais **Reembolsos** (Relacionamento: 1:N)
- Uma **Viagem** tem uma **Autorização** (Relacionamento: 1:1)

### Modelo de Relacionamento:

- Usuário (1) — (N) Viagem
- Viagem (1) — (N) Reserva
- Viagem (1) — (N) Reembolso
- Viagem (1) — (1) Autorização

## 5. Repositório GitHub

O repositório do projeto está disponível em: <https://github.com/vitogo22/sistema-corporativo-viagem>

## 6. Delineamento de Desenvolvimento

O desenvolvimento será dividido nas seguintes etapas:

1. levantamento de requisitos;
2. criação dos wireframes;
3. modelagem do banco de dados;

4. desenvolvimento frontend e backend;

5. testes e implantação;

## 7. Divisão da Equipe

Cada integrante da equipe ficará responsável por uma parte do sistema. A divisão detalhada será apresentada na tabela a seguir.

Vitor Hugo	Frontend
Eduardo	Backend
Vitor Hugo	Modelagem de dados e wireframes
Eduardo	Documentação e apresentação