Projeto Integrador 3º Semestre - DSM

**Disciplinas:**

Gestão Ágil de Projeto de Software

Banco de Dados Não Relacional

Desenvolvimento Web III

**Professores:**

Renato Cividini Matthiesen

Thiago Mendes

Orlando Saraiva do Nascimento Junior

**Grupo(n) / Nome da Empresa: Inova6**

**Sistema:** **Gestão e Vendas – Farofa Trip 019**

|  |  |
| --- | --- |
| **Integrantes** | **Papel Principal** |
| Willian Paulino | Design UI/UX  Documentação |
| Bruno Rodrigues | Developer Database |
| Eduardo Martins | Desenvolvedor Front-end/ Back-end |
| Marcos Zambom | Analista de Dados |

**Fatec Araras**

**2024**

FICHA DE CONTROLE - PROJETO INTERDISCIPLINAR

DISCIPLINA CHAVE: Gestão Ágil de Projetos - PI III

PROFESSOR: Renato Cividini matthiesen

GRUPO: Nome do grupo SEMESTRE: Escolher um item.

TÍTULO DO PROJETO: SSU – xxx

DATA DA APRESENTAÇÃO: xxx

NOTA:

INTEGRANTES DO GRUPO: Nome grupo

|  |  |
| --- | --- |
| Nome | Nota Individual |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Araras, 03 de dezembro de 2024

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Professor Renato Cividini Matthiesen

**Sumário**

[1. Gestão Ágil de Projeto de Software 3](#_Toc2096596499)

[1.1 Apresentação do projeto e considerações sobre a empresa (fictícia) que está desenvolvendo o sistema 4](#_Toc1692377678)

[1.2 Visão do produto 5](#_Toc1123193362)

[1.3 Persona + Mapa de Empatia 5](#_Toc1002673643)

[1.4 Canvas do projeto/produto 6](#_Toc818226710)

[1.5 Mapa mental do sistema 7](#_Toc238234536)

[1.6 Backlog do Produto 8](#_Toc349035082)

[Épico 001 – Gestão de Clientes 8](#_Toc1967533144)

[Épico 002 – Gestão de Excursões e Ingressos 9](#_Toc75761381)

[Épico 003 – Sistema de Pagamento 9](#_Toc1955354068)

[Épico 004 – Autenticação e Acesso 9](#_Toc1597055283)

[Épico 005 – Painel Administrativo 9](#_Toc1154845235)

[Épico 006 – Interface e Experiência do Usuário 9](#_Toc1137096720)

[Épico 007 – Documentação e Infraestrutura 9](#_Toc1166330980)

[1.7 Backlog dos Sprints 10](#_Toc834313082)

[1.8 Entregáveis – Minimum Viable Product (MVP) 10](#_Toc438080664)

[1.9 Retrospectiva dos Sprints 20](#_Toc728113466)

[1.10 Retrospectiva do Projeto. 22](#_Toc81590724)

[2. Desenvolvimento Web III 23](#_Toc885143893)

[2.1 Visão Geral 23](#_Toc838281241)

[2.2 Tecnologias Utilizadas 23](#_Toc2090356321)

[2.3 Justificativa Técnica 24](#_Toc1250150686)

[2.4 Estrutura do Projeto 24](#_Toc1258027698)

[2.5 Entregáveis – Minimum Viable Product (MVP) 27](#_Toc450436978)

[3. Banco de dados Não Relacional 35](#_Toc1383873262)

[3.1 Visão Geral. 36](#_Toc1730427757)

[3.2 Tecnologias Utilizadas. 36](#_Toc1723341104)

[3.3 Justificativa Técnica. 36](#_Toc2067324133)

[3.4 Estrutura do projeto. 37](#_Toc1851095033)

[3.5 Modelo de Dados. 38](#_Toc93744931)

[3.6 Entregáveis – Minimum Viable Product (MVP) 40](#_Toc1077457471)

[4. Considerações Finais 43](#_Toc308688951)

[Referências Bibliográficas 44](#_Toc111421389)

[Anexo I - Diário de bordo 45](#_Toc37042916)

[Anexo II – Cronograma efetivo 45](#_Toc551471440)

[Anexo III – Evidências 46](#_Toc1284227131)

# Gestão Ágil de Projeto de Software

A gestão ágil de projetos de software surgiu como alternativa aos modelos tradicionais, que muitas vezes são lentos e pouco flexíveis. O **Manifesto Ágil (2001)** propôs uma nova forma de trabalho baseada em **colaboração, adaptação e entregas incrementais**, sempre com foco em gerar valor ao cliente.

Entre os métodos ágeis, o **Scrum** é um dos mais utilizados. Ele organiza o trabalho em ciclos curtos chamados **Sprints**, nos quais a equipe entrega partes do sistema de forma contínua e revisa o que pode ser melhorado. Autores como **Sutherland (2019)** e **Cruz (2018)** destacam que essa abordagem traz maior transparência, rapidez e capacidade de adaptação a mudanças.

No caso do projeto da **Farofa Trip**, a gestão ágil permitirá priorizar as funcionalidades mais importantes (como reservas e pagamentos), testar soluções com os clientes e evoluir o sistema de forma iterativa, garantindo maior alinhamento com as necessidades do negócio.

## 1.1 Apresentação do projeto e considerações sobre a empresa (fictícia) que está desenvolvendo o sistema

A **Farofa Trip** atua no setor de turismo, com foco em excursões para festivais de música eletrônica e passeios em geral. Atualmente, o processo de atendimento é feito por **WhatsApp e Instagram**, dificultando a organização e rastreabilidade das informações.

**Principais desafios identificados:**

* Inexistência de **base de dados centralizada** para clientes.
* Ausência de **sistema de gestão de reservas e pagamentos**.
* Falta de **plataforma digital unificada** para divulgação e vendas.

**Objetivo geral do projeto:** Criar uma solução tecnológica que otimize a **gestão de excursões e ingressos**, melhore o **atendimento ao cliente** e apoie o **crescimento da empresa**.

O projeto tem como objetivo a criação de uma solução tecnológica que otimize a gestão da empresa, facilite o atendimento e melhore a experiência do cliente.

## 1.2 Visão do produto

Aplicação para Farofa Trip.

A empresa Farofa Trip atua no setor de turismo, com foco em venda de ingressos, excursões para festivais de música eletrônica e passeios em geral.

O projeto tem como objetivo a criação de uma solução tecnológica que otimize a gestão da empresa, facilite o atendimento e melhore a experiência do cliente.

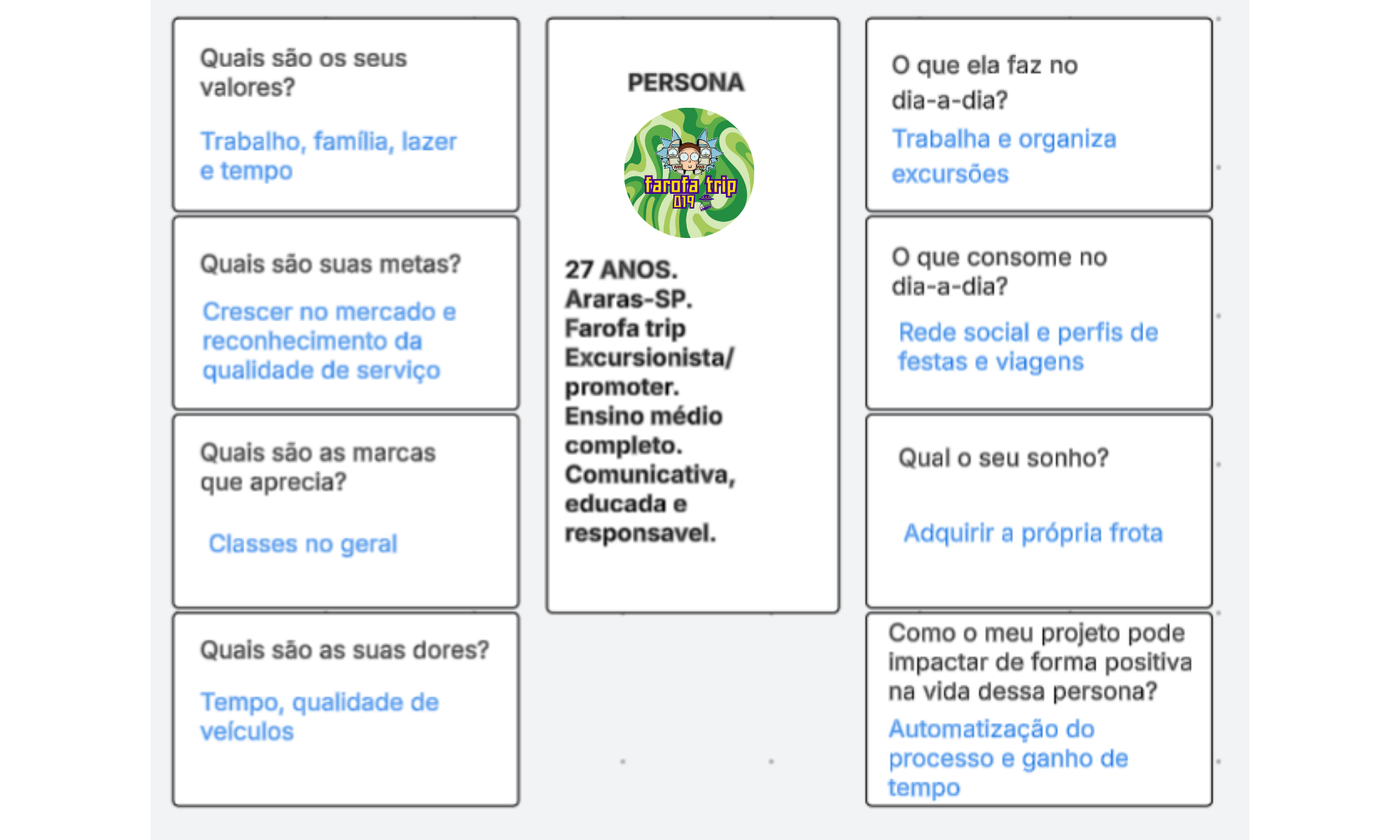
⦁ Proporcionar agilidade no atendimento.

⦁ Oferecer um cadastro estruturado de clientes.

⦁ Disponibilizar uma plataforma digital para consulta, reserva e compra de ingressos/excursões.

## 1.3 Persona + Mapa de Empatia

Persona Farofa Trip

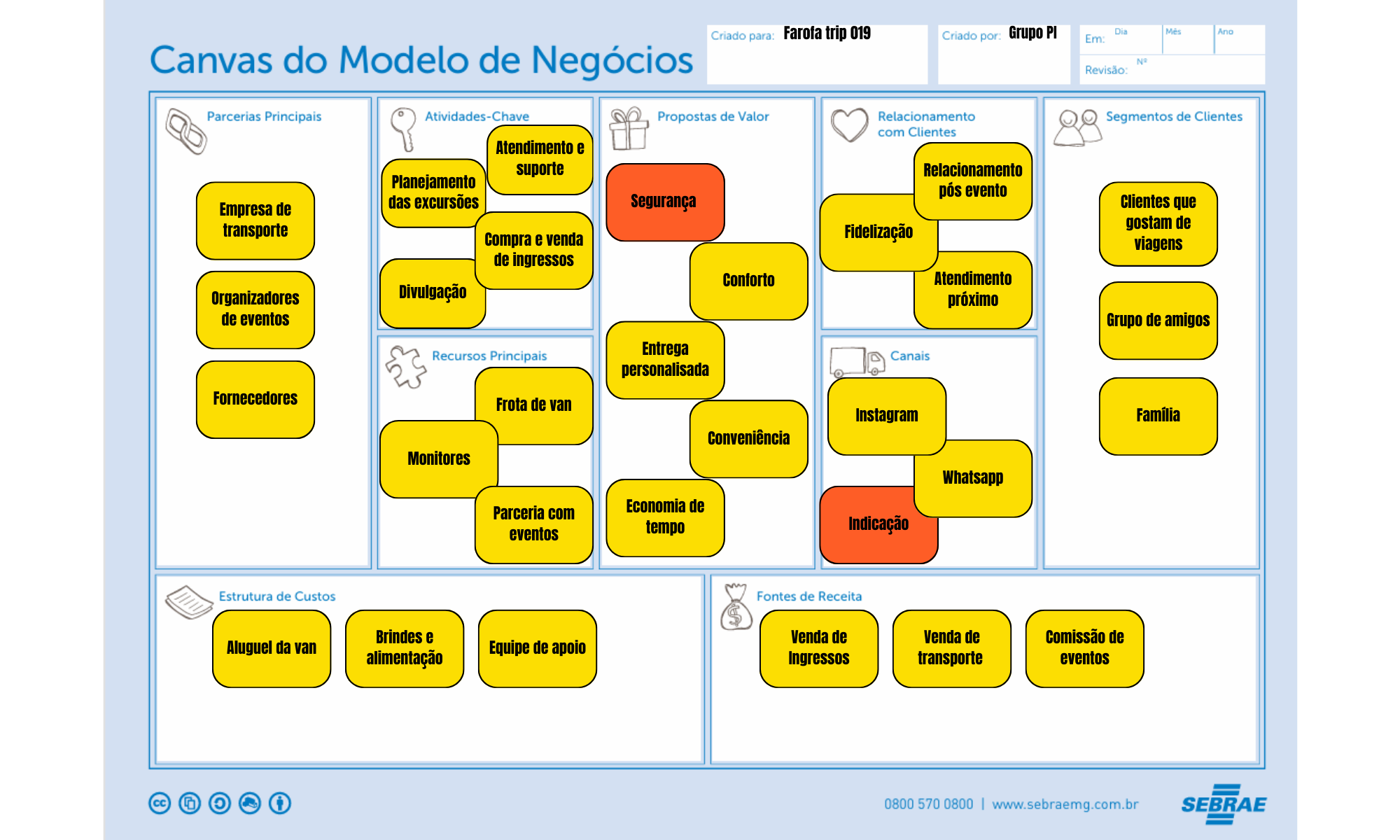


Mapa de Empatia Farofa Trip



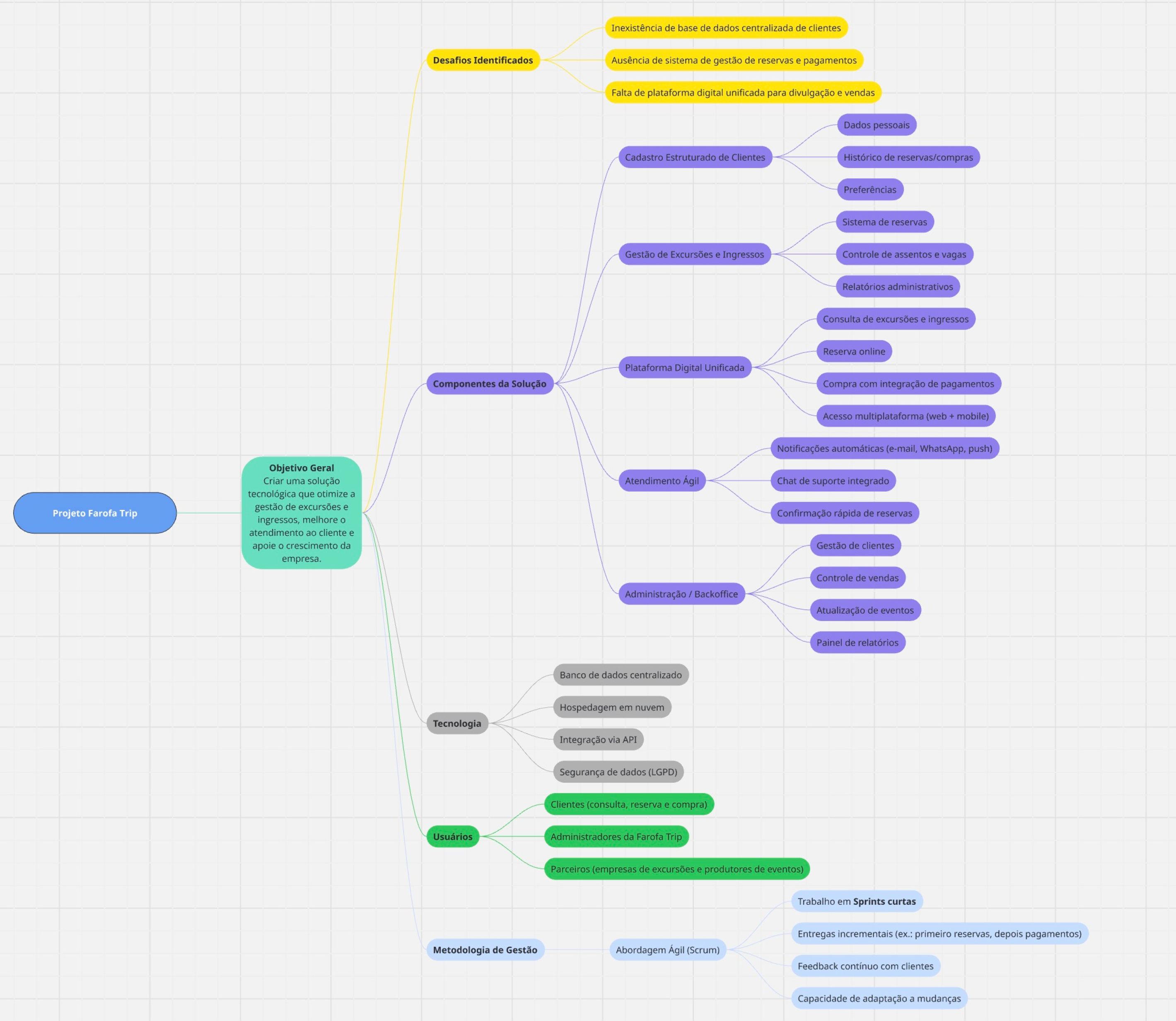
## 1.4 Canvas do projeto/produto

Canvas Model Farofa Trip



## 1.5 Mapa mental do sistema

Mapa Mental Farofa Trip



## 1.6 *Backlog* do Produto

O **Backlog do Produto** apresenta a lista de funcionalidades e requisitos priorizados que serão desenvolvidos ao longo do projeto:

### **Épico 001 – Gestão de Clientes**

* Como administrador, quero cadastrar clientes para manter um controle organizado das informações.
* Como cliente, quero me cadastrar e atualizar meus dados pessoais.
* Como administrador, quero visualizar e editar informações dos clientes para manter o banco de dados atualizado.

### **Épico 002 – Gestão de Excursões e Ingressos**

* Como administrador, quero cadastrar novas excursões e eventos (data, local, preço, vagas disponíveis).
* Como cliente, quero visualizar as excursões e eventos disponíveis.
* Como cliente, quero reservar vagas na excursão e comprar ingressos de forma online.
* Como administrador, quero gerenciar as reservas e controlar o número de vagas.

### **Épico 003 – Sistema de Pagamento**

* Como cliente, quero escolher a forma de pagamento (PIX ou cartão).
* Como administrador, quero acompanhar o status dos pagamentos realizados.

### **Épico 004 – Autenticação e Acesso**

* Como usuário, quero acessar o sistema com login e senha para segurança dos meus dados.

### **Épico 005 – Painel Administrativo**

* Como administrador, quero visualizar um painel com métricas (número de excursões, vendas, clientes).
* Como administrador, quero gerar relatórios de vendas e reservas.

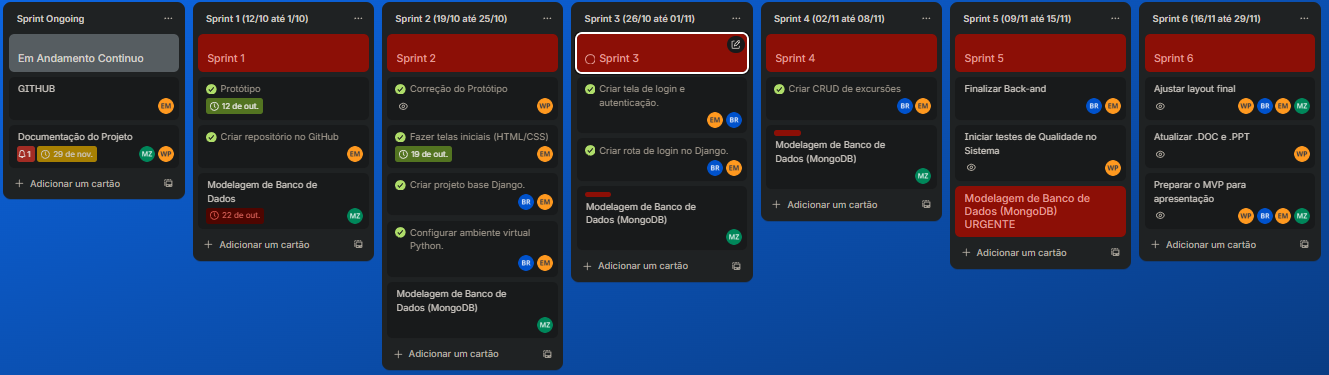
### **Épico 006 – Interface e Experiência do Usuário**

* Como usuário, quero acessar uma interface simples e intuitiva, com design responsivo.
* Como cliente, quero visualizar informações detalhadas dos eventos com fotos e descrições.

### **Épico 007 – Documentação e Infraestrutura**

* Como equipe, queremos manter o código e a documentação organizados e atualizados via versionamento de código com Git.
* Como equipe, queremos manter o banco de dados estruturado no MongoDB para fácil acesso e manutenção.

## 1.7 *Backlog* dos *Sprints*



## 1.8 Entregáveis – *Minimum Viable Product* (MVP)

**Sprint 1**



**Objetivo:** Estruturação inicial do projeto e definição dos requisitos.

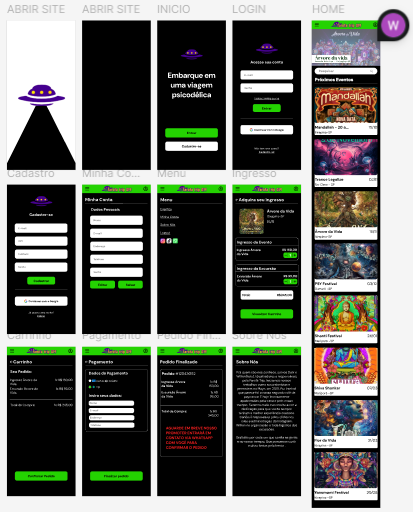
⦁Foi feito o levantamento de requisitos com o cliente.

⦁Definição da arquitetura (Python + Django + MongoDB + HTML/CSS).

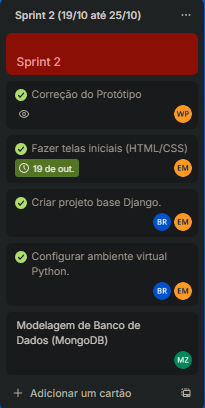
⦁Alinhamento sobre funções de cada membro.

⦁Criação do primeiro protótipo básico no Figma.

Protótipo do sistema



**Sprint 2**



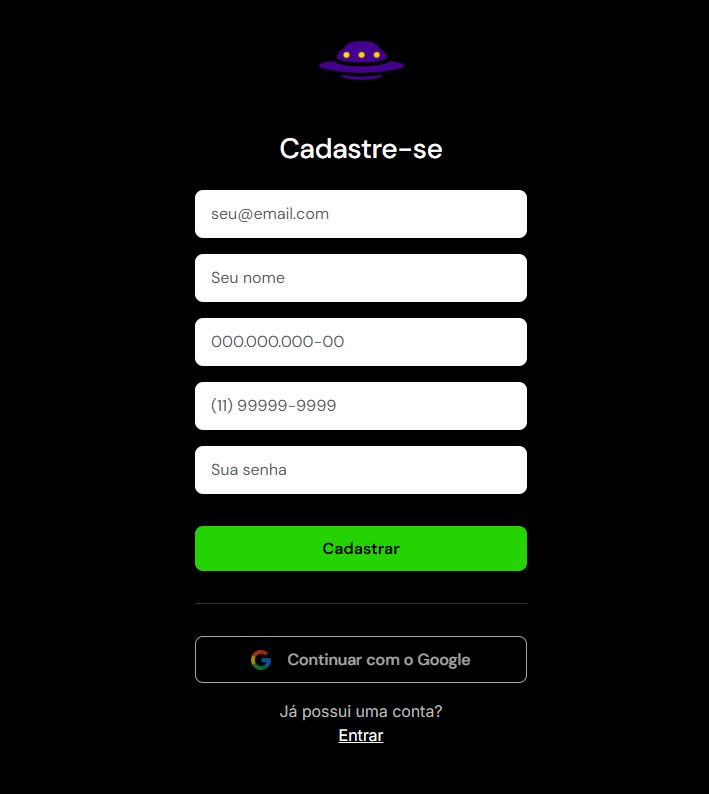
**Objetivo**: Criar as coleções básicas do banco de dados e configurar o backend.

⦁Foi criado a tela de cadastro do sistema (HTML/CSS).

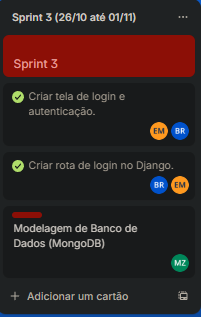
⦁Foi realizada algumas correções necessarias no protótipo.

⦁Foi configurado o ambiente virtual.

Tela de Cadastro:



**Sprint 3**



Objetivo: Criar as APIs principais e telas de front-end iniciais.

⦁Foi feito a API de cadastro de clientes.

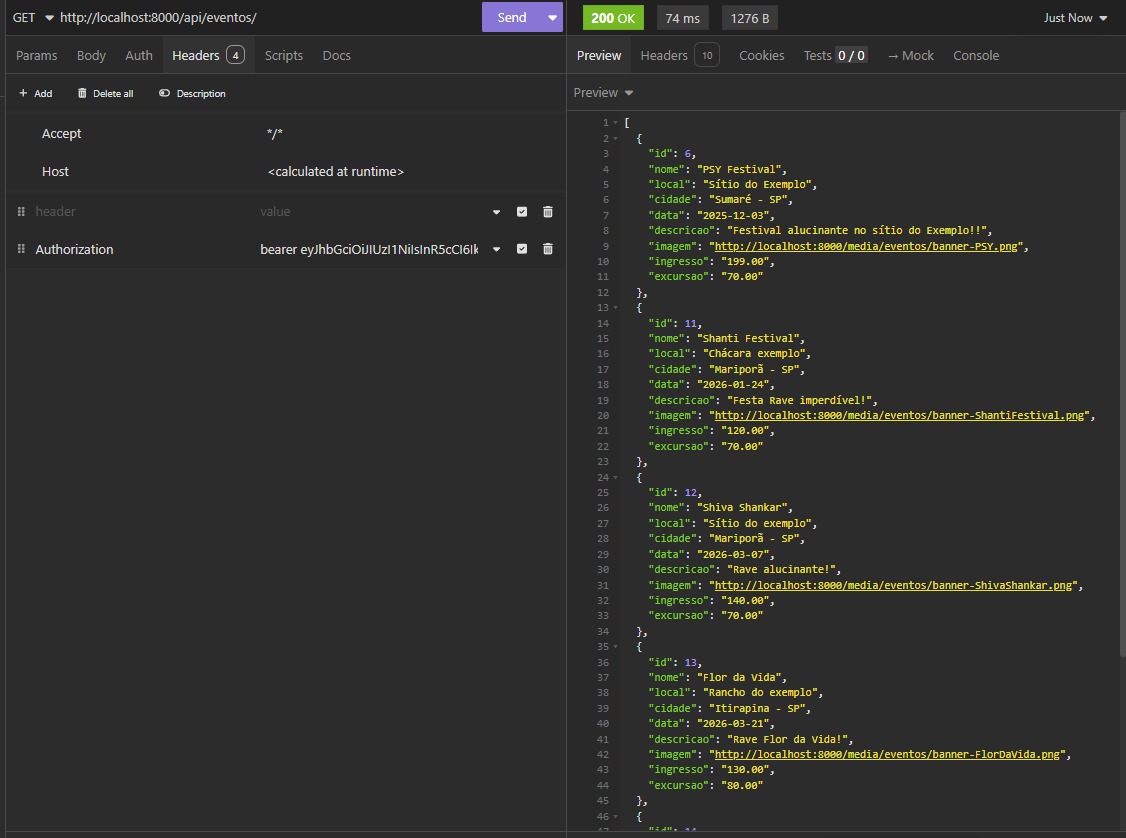
⦁Foi feito a API de listagem de eventos e excursões.

⦁Foi feito a API criação de pedidos.

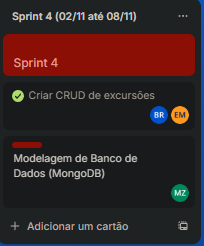
(OBS: APIS retornando status 200)

⦁Desenvolvimento da tela de login.

⦁Desenvolvimento da Home do sistema.

Teste de Listagem de Eventos: 

**Sprint 4**

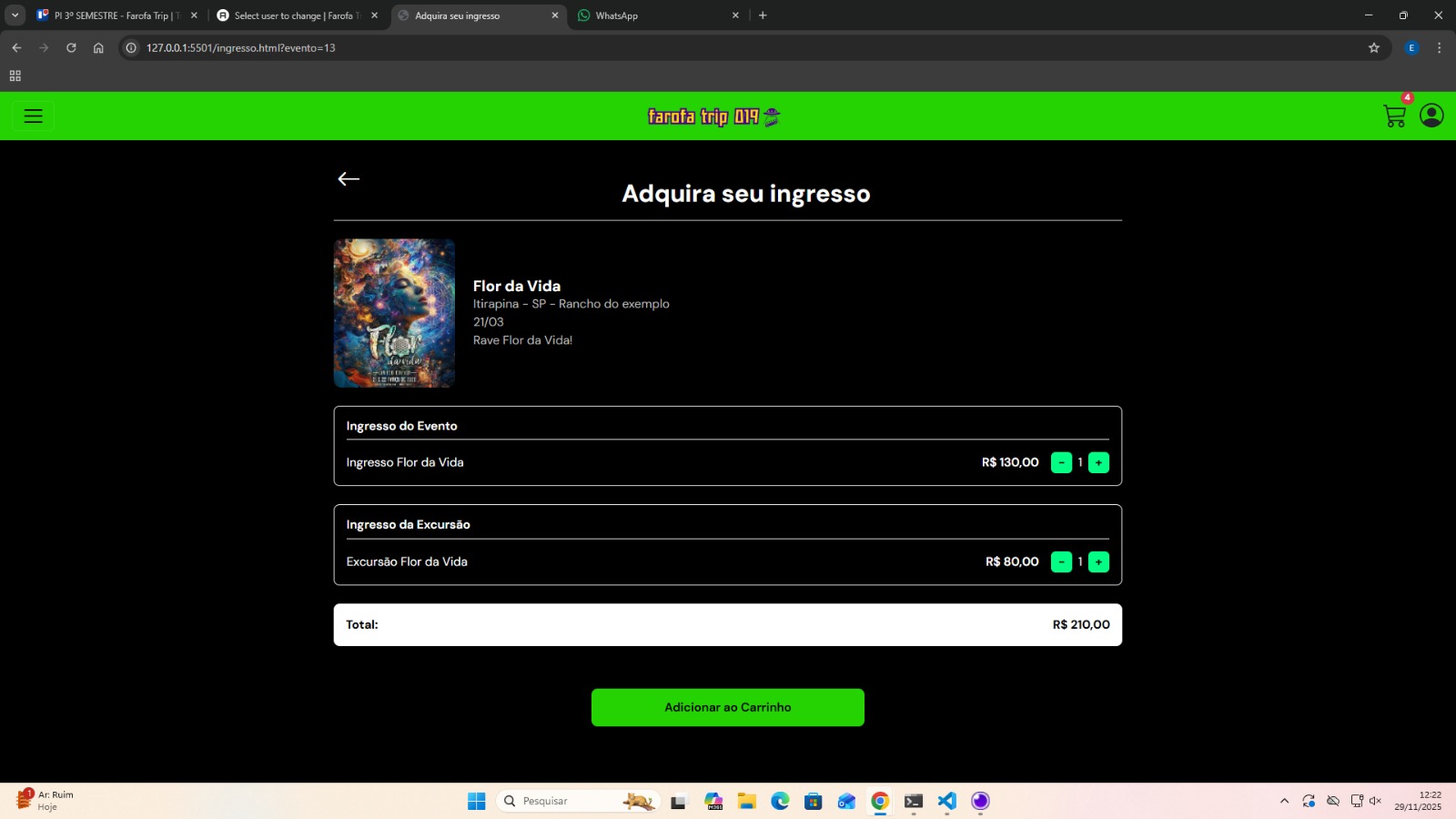


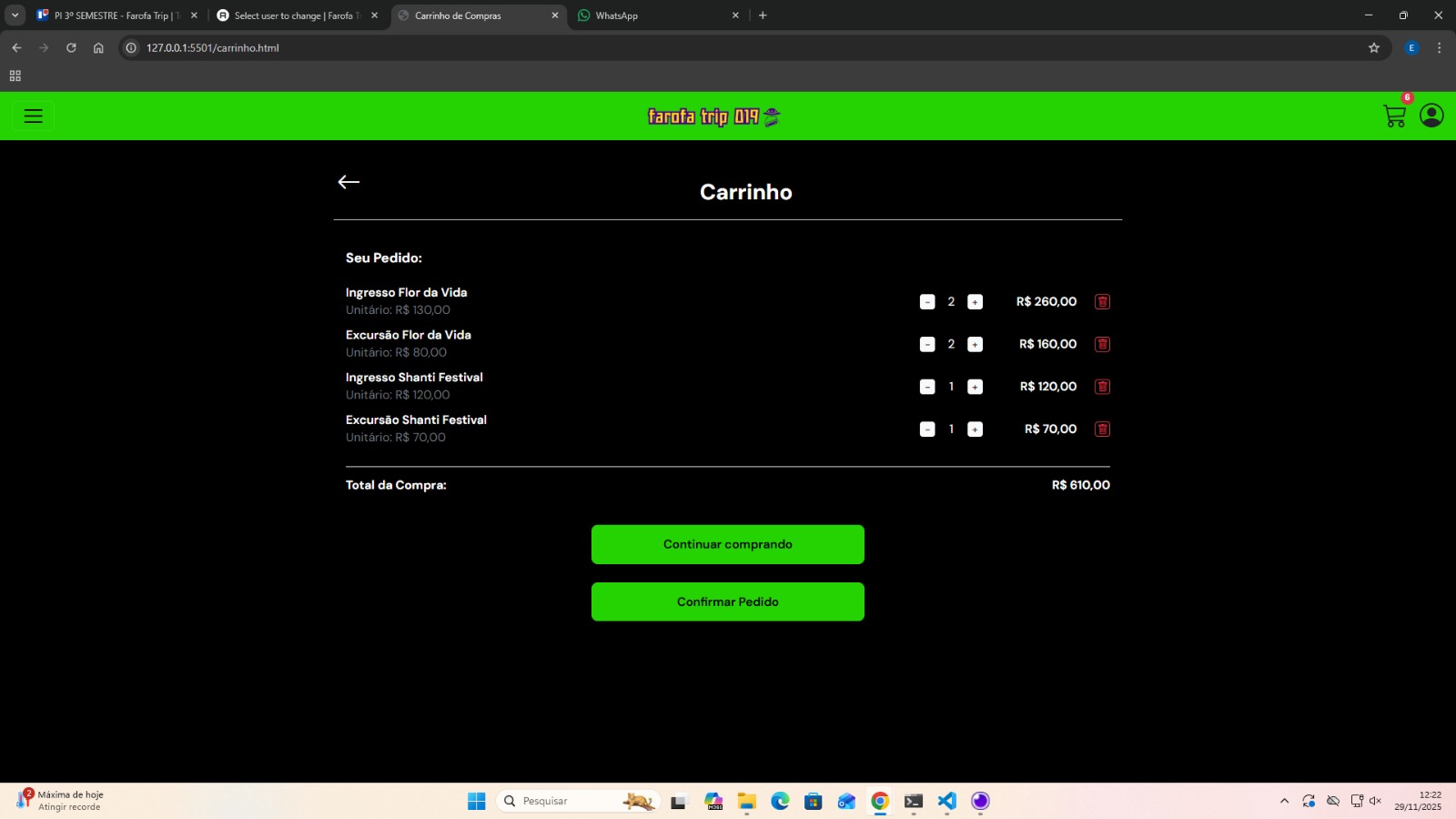
Objetivo: Criar fluxo completo de venda (ingresso/excursão).

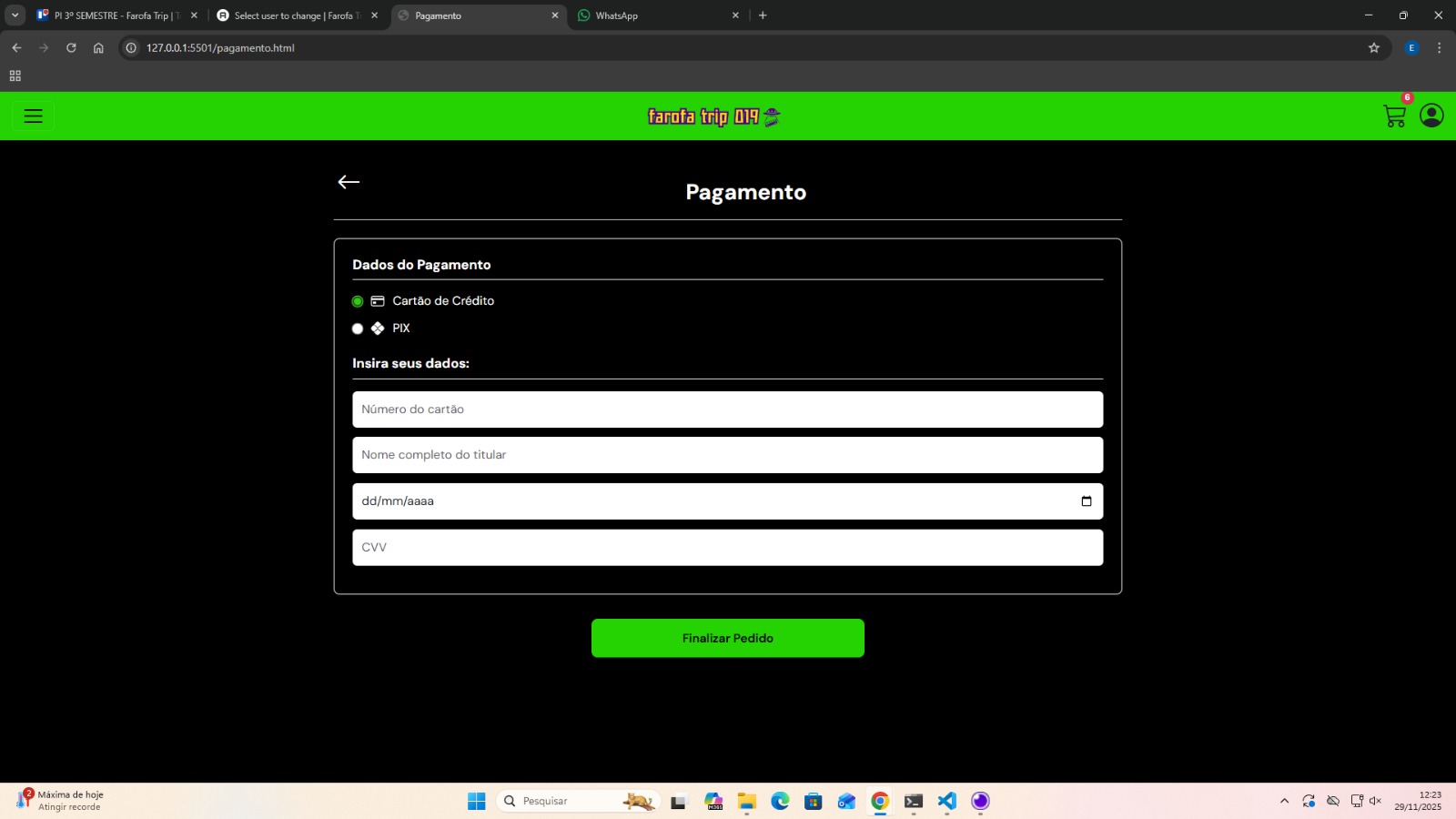
⦁Criação do carrinho de compras.

⦁Criação do processo de compra.

⦁Tela de detalhes do evento/excursão.

Fluxo de compras 1: 

Fluxo de compras 2: 

Fluxo de compras 3: 

Fluxo de compras 4: 

**Sprint 5**



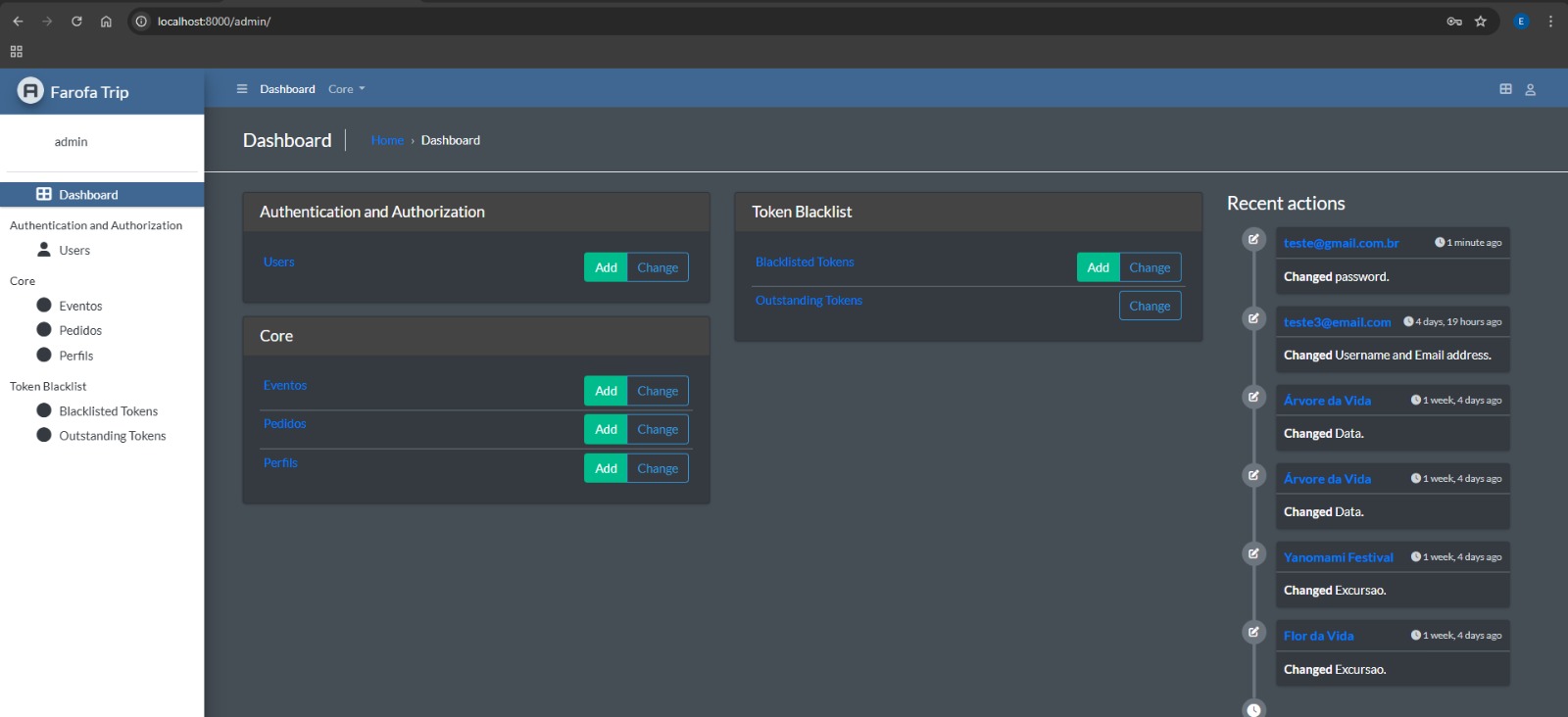
Objetivo: Correção de bugs e testes no sistema.

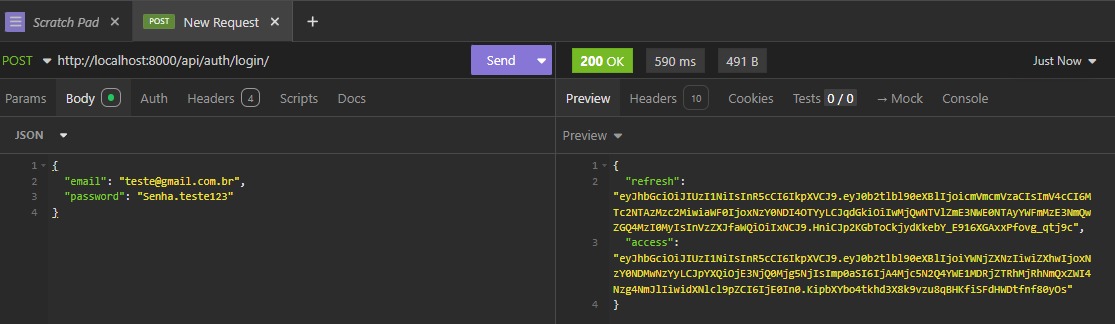
⦁Criação da tela de dashboard.

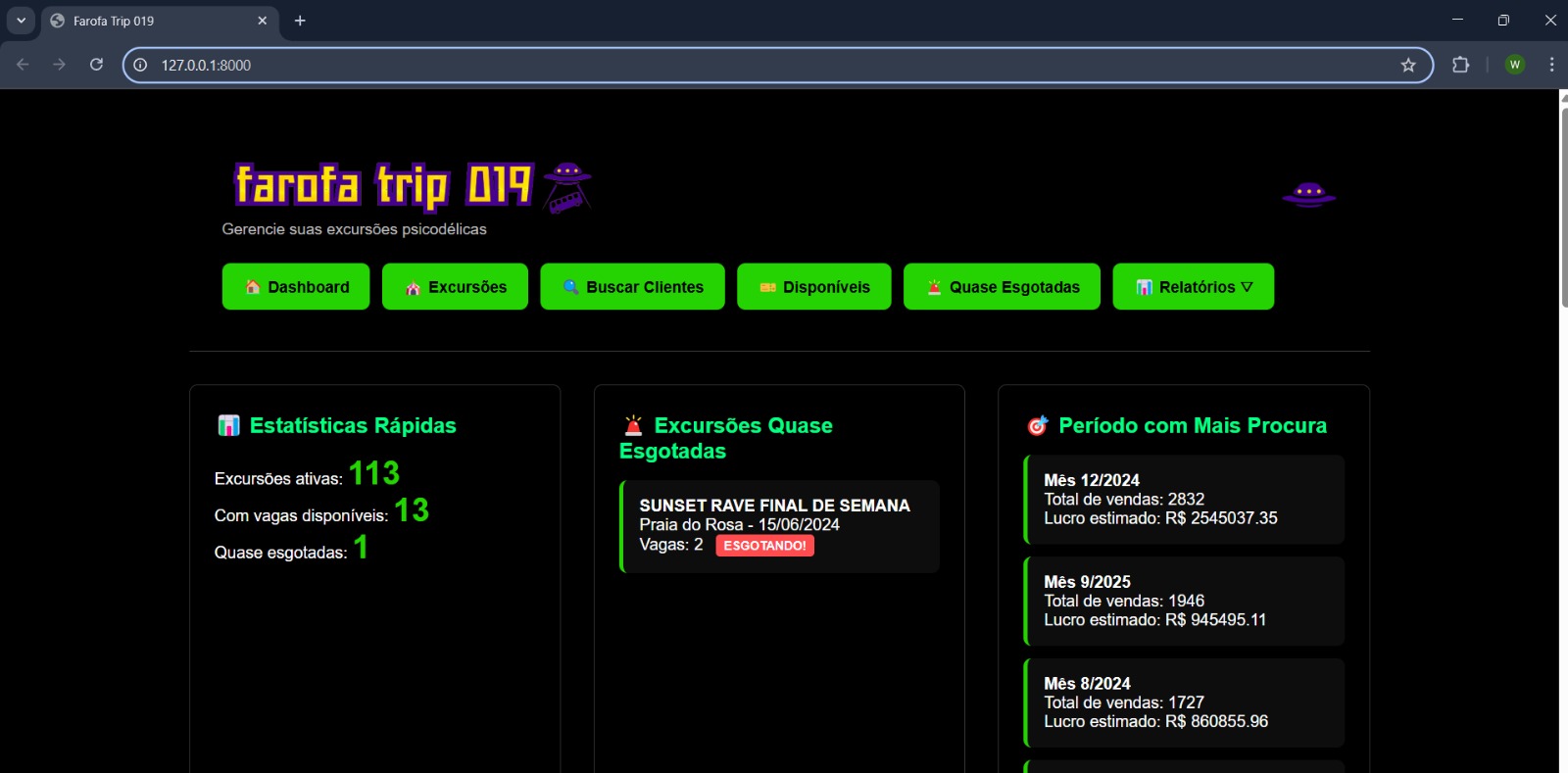
⦁Realização de testes.

⦁Implementação no Django Admin.

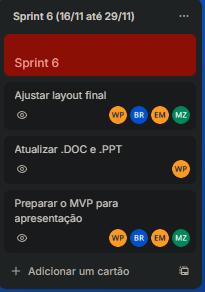
⦁Modelagem de Banco de dados (NoSQL).

Dashboard Admin: 

Teste de Login via insomnia: 

Dashboard Banco de Dados NoSQL: 

**Sprint 6**



Objetivo: Finalização, testes e documentação.

⦁Testes de integração backend.

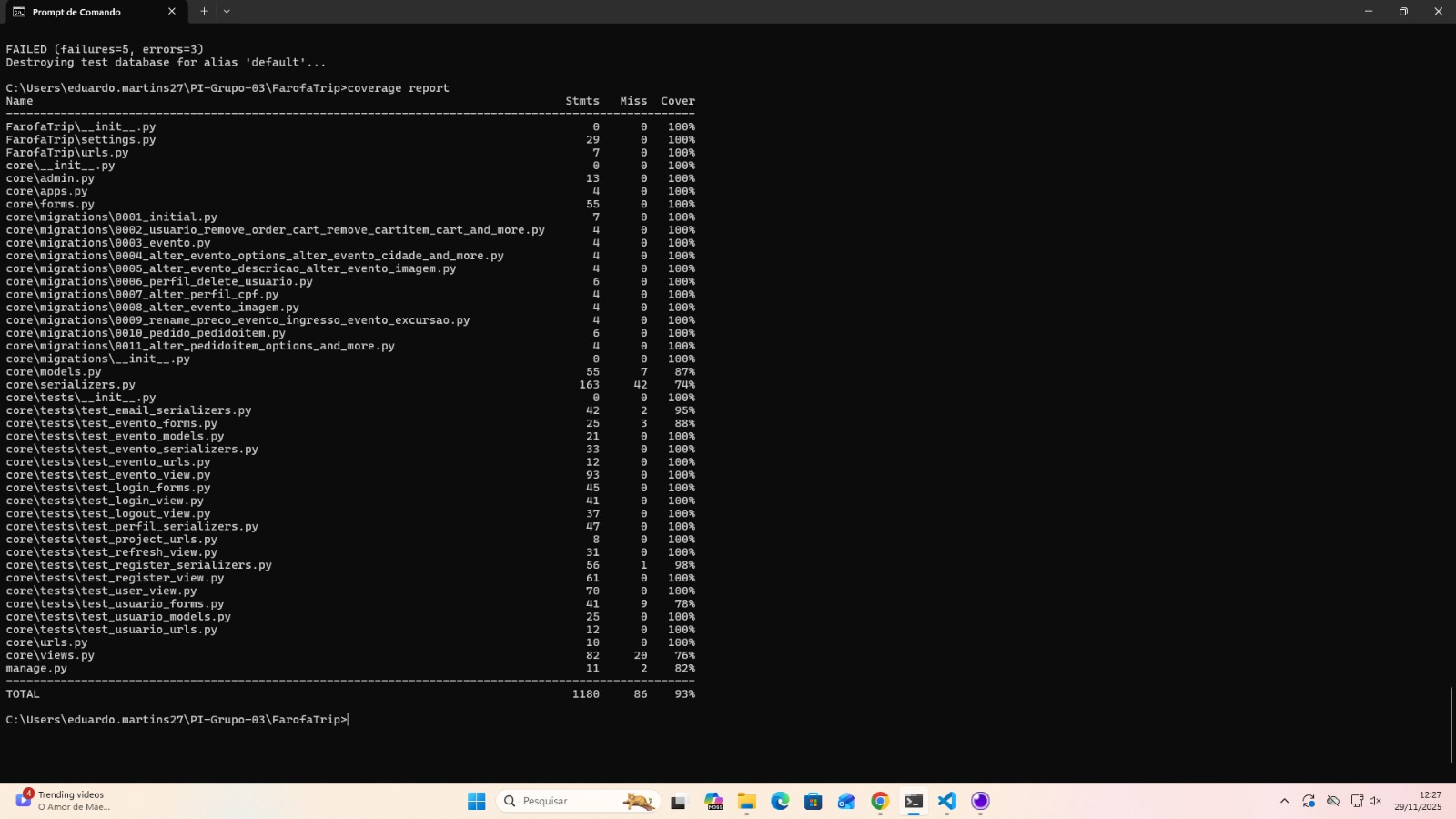
⦁Testes das telas principais.

⦁Ajustes finais do protótipo.

⦁Finalização da documentação.

⦁Revisão geral do código.

Resultado dos testes do Sistema:



## 1.9 Retrospectiva dos *Sprints*

1. **Aspectos que funcionaram bem:**

⦁ O desenvolvimento do protótipo inicial no Figma dentro do prazo estabelecido, facilitando a visualização das primeiras funcionalidades.

⦁ A criação do banco de dados no MongoDB ocorreu de forma satisfatória, com boa modelagem das coleções necessárias ao sistema.

⦁ As APIs iniciais foram implementadas de maneira eficaz e apresentaram bom desempenho nos testes realizados, contribuindo para a integração dos módulos.

1. **Aspectos que requer melhoria:**

Embora os resultados gerais tenham sido positivos, alguns desafios foram identificados ao longo dos sprints.

⦁ Determinadas tarefas demandaram mais tempo do que o previsto, especialmente no que diz respeito à implementação de funcionalidades mais complexas no backend.

⦁ Ajustes no protótipo impactaram o cronograma inicial, ocasionando pequenos atrasos.

⦁ A integração entre o front-end e back-end apresentou dificuldades iniciais, exigindo retrabalho e ajustes na comunicação das APIs.

⦁ A realização de testes dentro do prazo estabelecido mostrou-se desafiadora em alguns momentos, devido à carga de tarefas acumulada.

1. **Aprendizados obtidos ao longo do projeto:**

Ao longo dos sprints, a equipe adquiriu conhecimentos relevantes tanto na parte técnica quanto na parte organizacional.

⦁ Compreedemos a importância de revisões frequentes de requisitos para evitar inconsistências durante a implementação.

⦁ A equipe amadureceu na utilização da prática do Scrum, principalmente no planejamento e priorização de tarefas.

⦁ Houve aprimoramento no domínio de tecnologias como MongoDB, Python, frameworks (Django).

⦁ Observamos a importância de realizar testes ao longo do desenvolvimento para reduzir retrabalhos.

1. **Ações de melhoria para os próximos ciclos:**

⦁ Aperfeiçoar a estimativa de esforço das tarefas para tornar o planejamento mais preciso.

⦁ Melhorar a comunicação entre a equipe.

⦁ Fortalecer o processo de documentação contínua, especialmente de APIs e fluxos do sistema.

⦁ Melhorar o fluxo de commits e gerenciamento de branches no GitHub, visando maior organização e rastreabilidade.

⦁ Intensificar a realização de testes automatizados e manuais durante a execução de cada Sprint, reduzindo riscos de falhas tardias.

## 1.10 Retrospectiva do Projeto.

A realização do projeto Farofa Trip 019 permitiu a integração prática dos conhecimentos adquiridos ao longo do curso, possibilitando o desenvolvimento de uma solução tecnológica voltada ao gerenciamento de excursões e vendas de ingressos. Durante o processo, foi possível observar avanços significativos na capacidade da equipe em planejar, estruturar e implementar um sistema funcional baseado em tecnologias modernas como Python, MongoDB, HTML/CSS e princípios da metodologia Scrum.

Ao final do projeto, verificou-se que os principais objetivos iniciais foram alcançados. O sistema desenvolvido apresentou funcionalidades essenciais, como cadastro de clientes, eventos e excursões, criação de pedidos, controle de pagamentos, geração de relatórios e um dashboard administrativo capaz de auxiliar na tomada de decisões da empresa. O fluxo de compra foi implementado com sucesso, demonstrando a aplicabilidade real da solução criada.

Os aprendizados obtidos foram expressivos. A equipe aprimorou suas habilidades em programação, modelagem de dados, criação de APIs e testes, além de vivenciar de forma prática a dinâmica de trabalho em equipe orientado por Scrum. Ficou evidente a importância da comunicação contínua, da documentação adequada e dos testes frequentes para garantir entregas mais consistentes e com menor índice de retrabalhos.

Como reflexão final, identificou-se que uma gestão de tempo mais rigorosa e uma definição inicial mais precisa dos requisitos poderiam ter evitado alguns retrabalhos. A organização dos commits, o controle de versões e a integração contínua são pontos que, se melhor estruturados desde o início, poderiam ter otimizado ainda mais o fluxo de desenvolvimento.

# Desenvolvimento Web III

Durante o semestre aprendemos a trabalhar a linguagem Python e o framework Django, essa tecnologia está sendo aplicada no projeto Farofa Trip. O framework Django, desenvolvido em Python, é uma ferramenta poderosa para construção de aplicações web modernas, seguras e escaláveis. Ele segue o padrão de arquitetura MVT (Model–View–Template), permitindo organizar o código de forma clara e eficiente.

## 2.1 Visão Geral

A plataforma oferece funcionalidades voltadas tanto para os administradores quanto para os clientes. Para o público interno, o sistema disponibiliza um painel administrativo que permite gerenciar excursões, lotes de ingressos, clientes, pagamentos e histórico de viagens. Já para o cliente final, a aplicação apresenta uma interface intuitiva onde é possível visualizar eventos disponíveis, consultar detalhes das excursões e realizar reservas.

A arquitetura do sistema foi planejada para garantir flexibilidade e performance. O backend em Django segue o padrão MVT (Model–View–Template), garantindo organização no fluxo de dados e facilitando futuras expansões do projeto.

## 2.2 Tecnologias Utilizadas

**Backend**

Python 3.12+ - Linguagem principal

Django 5.2 - Framework web

**Frontend**

HTML5 - Estrutura das páginas

CSS3 - Estilização customizada

Bootstrap 5 - Layouts responsivos

## 2.3 Justificativa Técnica

Porque Python + Django?

1. Desenvolvimento ágil e organizado.

* O Django oferece uma estrutura pronta, com boas práticas, reduzindo tempo de desenvolvimento.

1. Segurança nativa.

* Criptografia automática de senhas.

1. Integração com API REST.

* Com Django REST Framework, fica fácil criar endpoints.

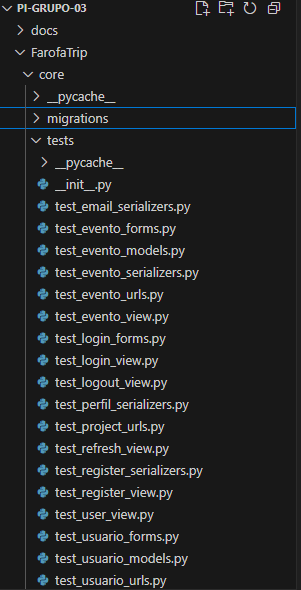
1. Admin Panel pronto.

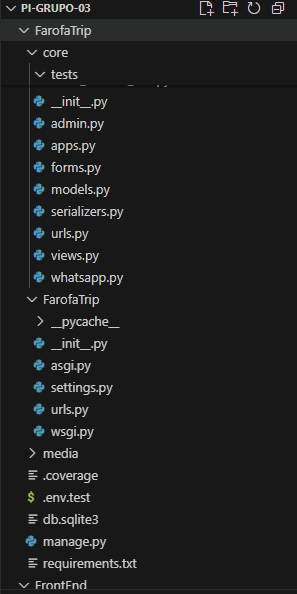
* O Django Admin permite:  
   cadastrar excursões e eventos;  
   gerenciar clientes;  
   visualizar pagamentos.

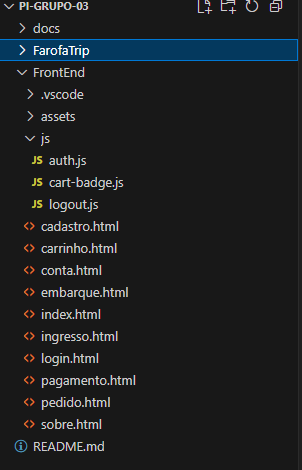
1. Escalabilidade e boa performance.

* Ideal para sistemas que vão crescer, com muitos usuários realizando reservas e compras.

## 2.4 Estrutura do Projeto







## 2.5 Entregáveis – Minimum Viable Product (MVP)

Diagrama de Casos de uso (Administrador):

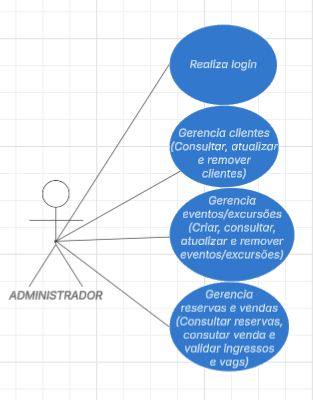


Diagrama de Casos de Uso (Cliente):

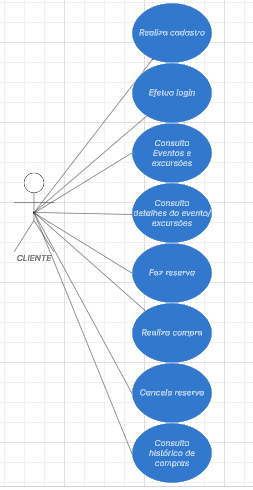
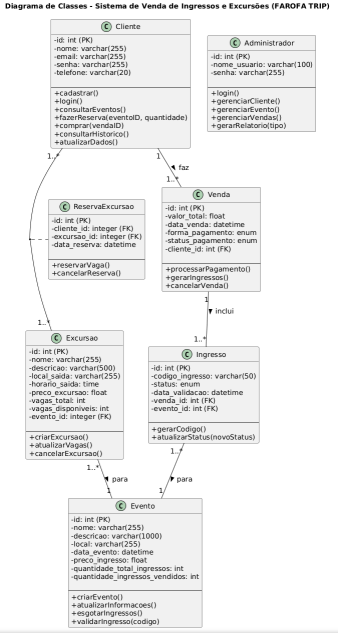
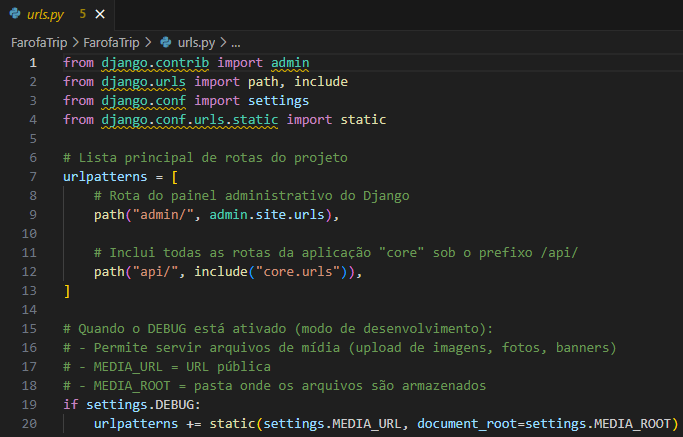


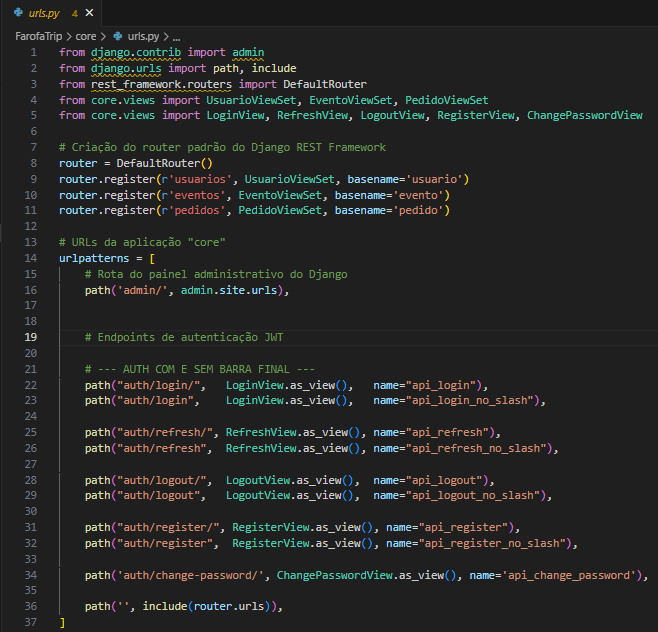
Diagrama de Classe:



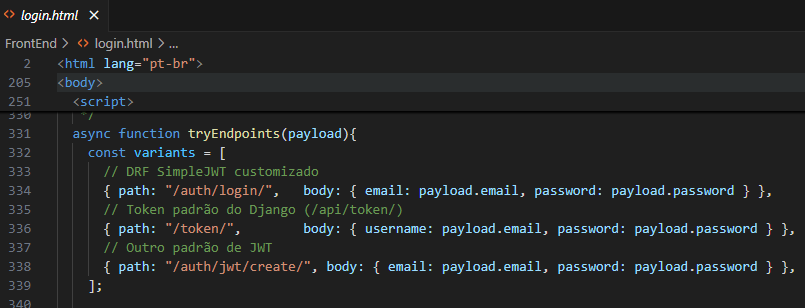
Urls do projeto:



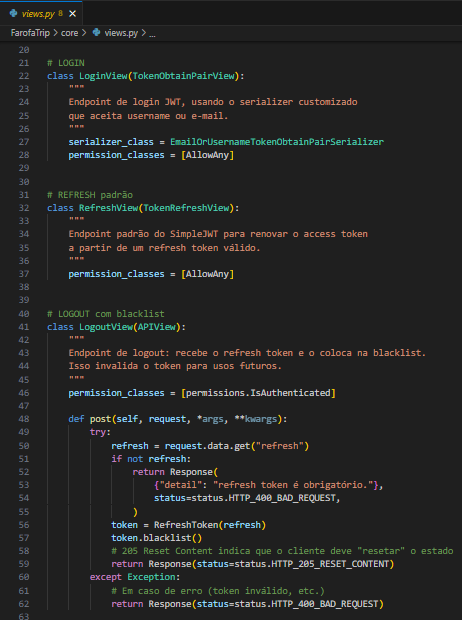
Urls do api (core)



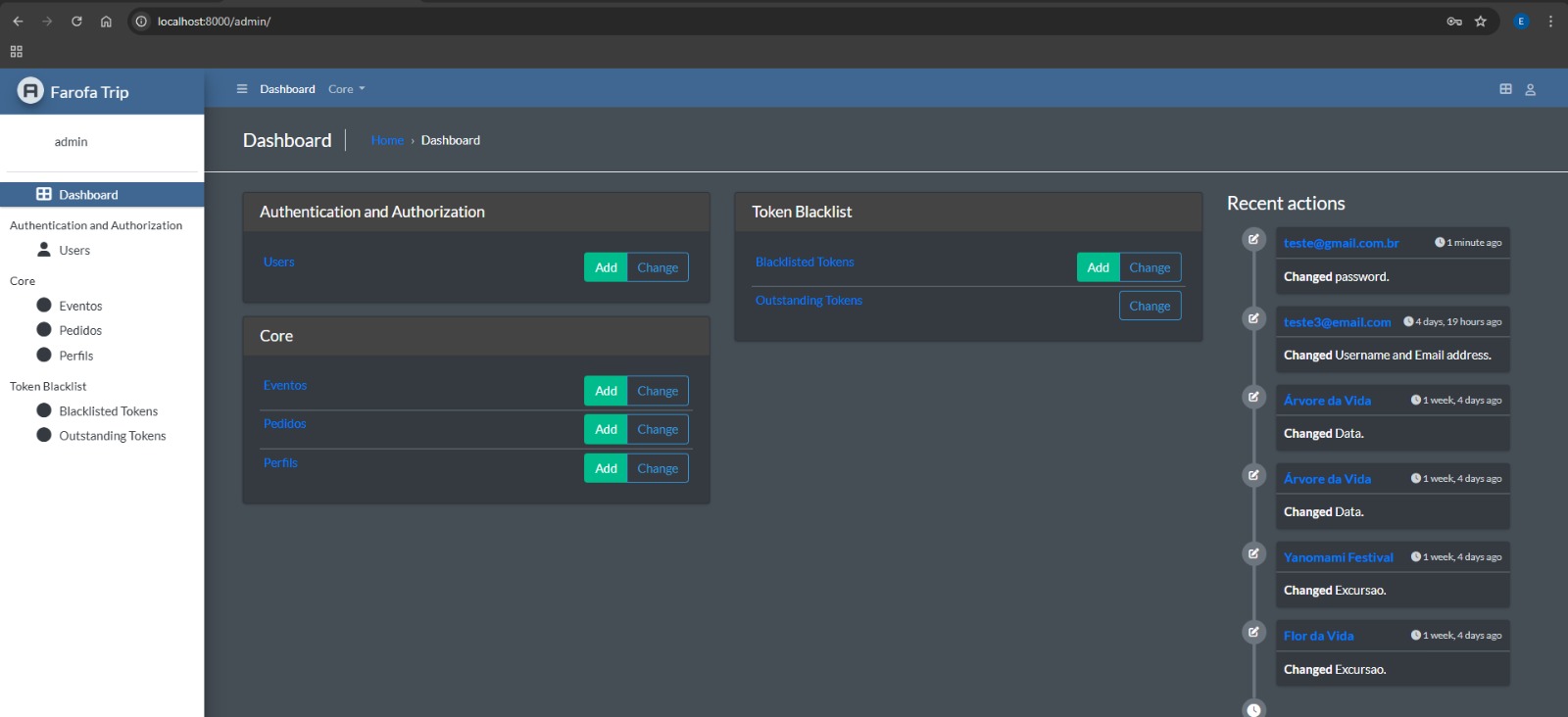
Integração do Frontend com o Backend



Autenticação com o recurso nativo do Django (Login/Logout)



Dashboard Admin:



# Banco de dados Não Relacional

Os bancos de dados não relacionais, também conhecidos como NoSQL, são sistemas de armazenamento de dados que não utilizam o modelo tradicional de tabelas, linhas e colunas encontrado nos bancos relacionais.

Em vez disso, esses bancos adotam modelos mais flexíveis, como:

⦁ Documentos (ex.: MongoDB)

⦁ Chave–valor

⦁ Grafos

⦁ Família de colunas

Nesse tipo de banco, os dados não precisam seguir um esquema rígido, podendo ser armazenados com estruturas variadas, o que facilita a adaptação às necessidades do sistema.

No contexto do projeto Farofa Trip, o uso de um banco de dados não relacional traz benefícios como:

⦁ Estrutura flexível para armazenar clientes, eventos e excursões.

⦁ Facilidade para adicionar novos campos conforme o sistema evolui.

⦁ Melhor performace para buscas rápidas e dashboards.

⦁ Integração direta com o backend em Python e APIs REST.

## 3.1 Visão Geral.

Sistema desenvolvido para gerenciamento e análise de dados de excursões destinadas a eventos de musicas eletrônicas (raves). Foi desenvolvido um dashboard que fornece insights em tempo real sobre vendas, clientes, disponibilidade e períodos de alta demanda, auxiliando na tomada de decisões estratégicas para o negócio.

## 3.2 Tecnologias Utilizadas.

**Backend**

Python 3.12+ - Linguagem principal

Django 5.2 - Framework web

PyMongo 4.3 - Driver oficial MongoDB

MongoDB Atlas - Banco de dados cloud

Django Decouple - Gerenciamento de variáveis de ambiente

**Frontend**

HTML5 - Estrutura das páginas

CSS3 - Estilização customizada

Django Template Language - Renderização server-side

## 3.3 Justificativa Técnica.

Por que MongoDB (NoSQL)?

1. Estrutura Flexível de Dados

⦁ Os dados das excursões possuem atributos variáveis (tags, descrições, informações extras).

⦁ O MongoDB permite esquemas dinâmicos sem necessidade de migrações complexas.

1. Desempenho em Consultas Complexas

⦁ Agregações multidimensionais (períodos com mais procura, lucro estimado).

⦁ Índices eficientes para buscas em grandes volumes de dados.

1. Escalabilidade Horizontal

⦁ Capacidade de escalar para milhões de registros com sharding.

⦁ Replicação automática para alta disponibilidade.

1. Adequação ao Domínio

⦁ Relacionamentos simples entre excursões, clientes e vendas.

⦁ Consultas geográficas futuras (localização das raves).

1. JSON Nativo

⦁ Facilidade de integração com APIs RESTful.

⦁ Compatibilidade com frameworks modernos de frontend.

## 3.4 Estrutura do projeto.

rave-excursions-dashboard/

├── rave\_agenda/ # Projeto Django principal

│ ├── **init**.py

│ ├── settings.py # Configurações do projeto

│ ├── urls.py # URLs principais

│ └── wsgi.py

├── excursions/ # Aplicação principal

│ ├── **init**.py

│ ├── models.py # Modelos Python (não-ORM)

│ ├── views.py # Views do Django

│ ├── urls.py # URLs da aplicação

│ ├── repository.py # Lógica de acesso ao MongoDB

│ └── mongo\_connection.py # Conexão com MongoDB

├── templates/ # Templates HTML

│ ├── base.html # Template base

│ ├── dashboard.html # Dashboard principal

│ ├── analytics.html # Página de análises

│ ├── buscar\_evento.html # Busca de clientes

│ ├── clientes\_evento.html # Lista de clientes

│ ├── disponiveis.html # Excursões disponíveis

│ └── quase\_esgotadas.html # Excursões quase esgotadas

├── scripts/ # Scripts auxiliares

│ └── populate\_complete\_test\_data.py # População de dados

├── static/ # Arquivos estáticos

├── .env # Variáveis de ambiente (não versionado)

├── .gitignore # Arquivos ignorados pelo Git

├── requirements.txt # Dependências Python

├── manage.py # Script de gerenciamento Django

└── README.md # Este arquivo

## 3.5 Modelo de Dados.

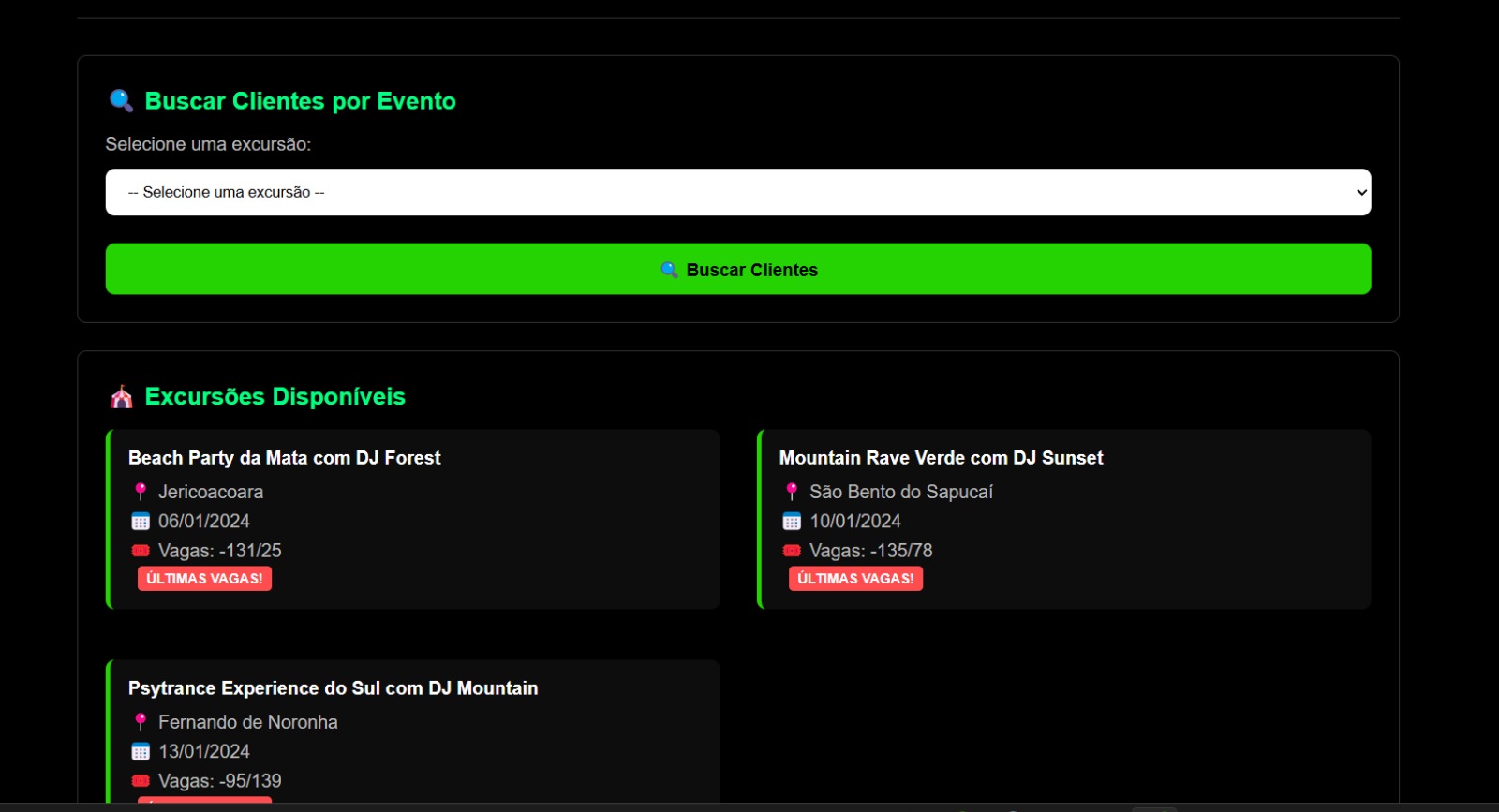




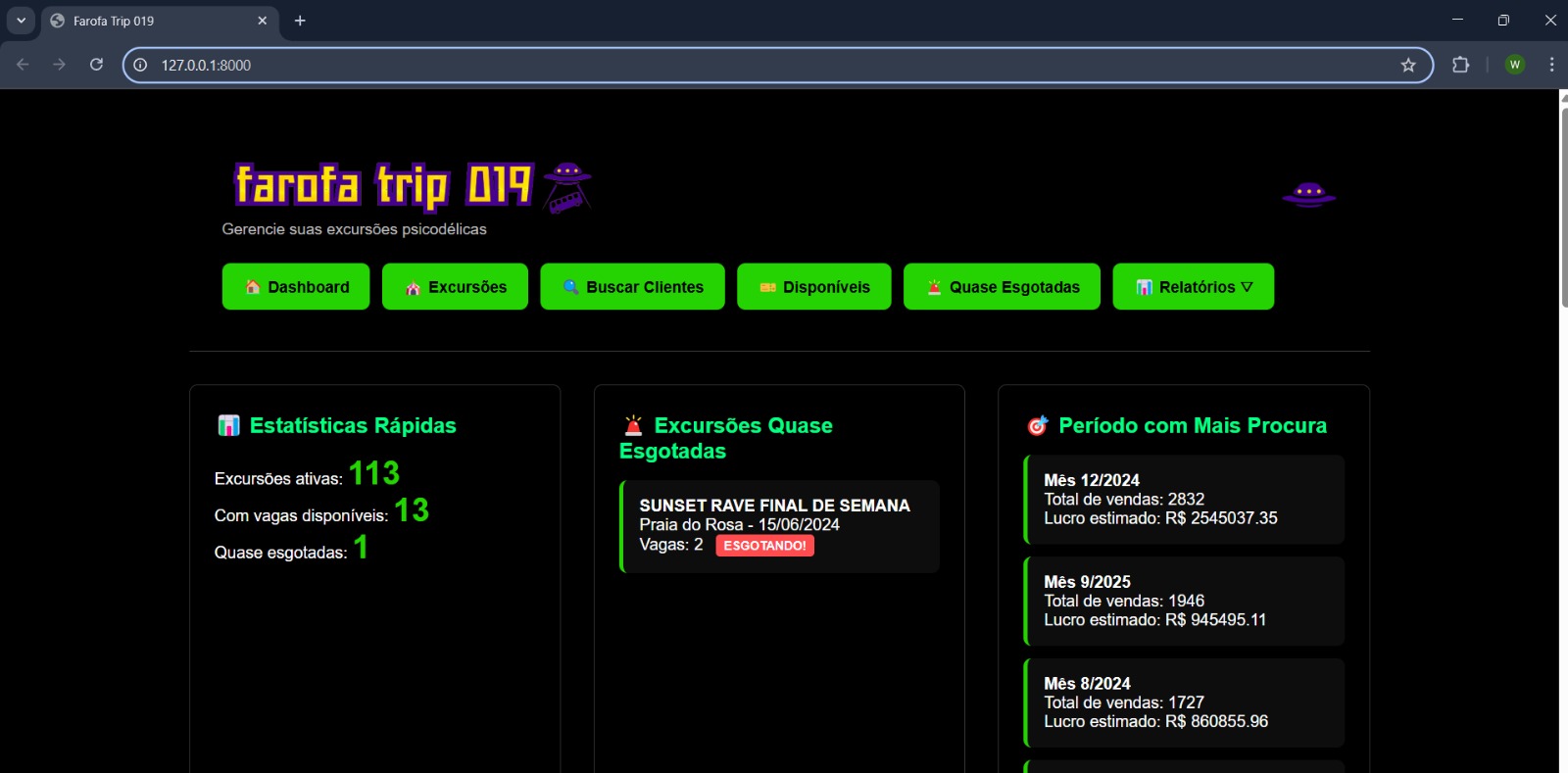


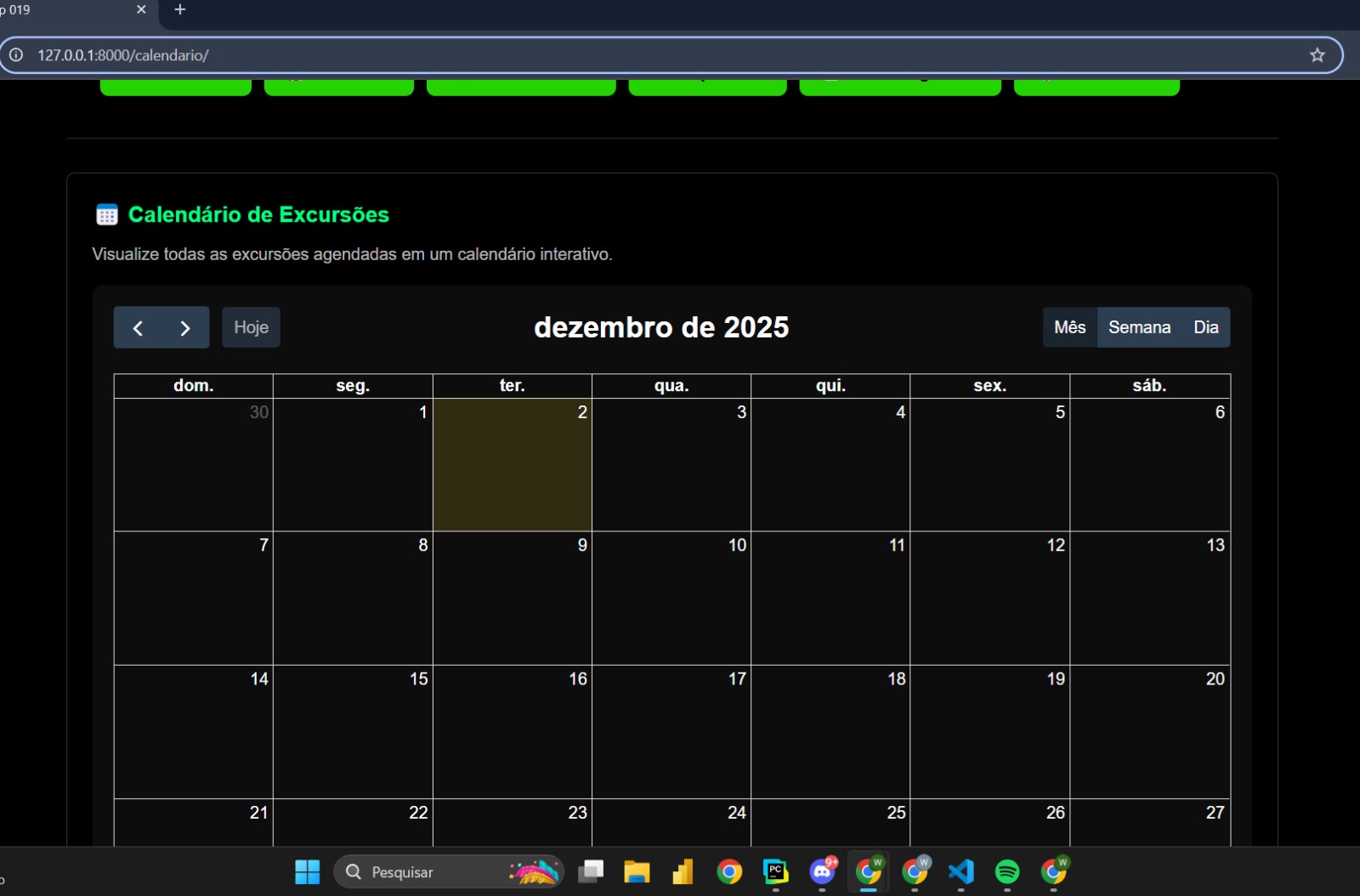
## 3.6 Entregáveis – Minimum Viable Product (MVP)

Busca de clientes

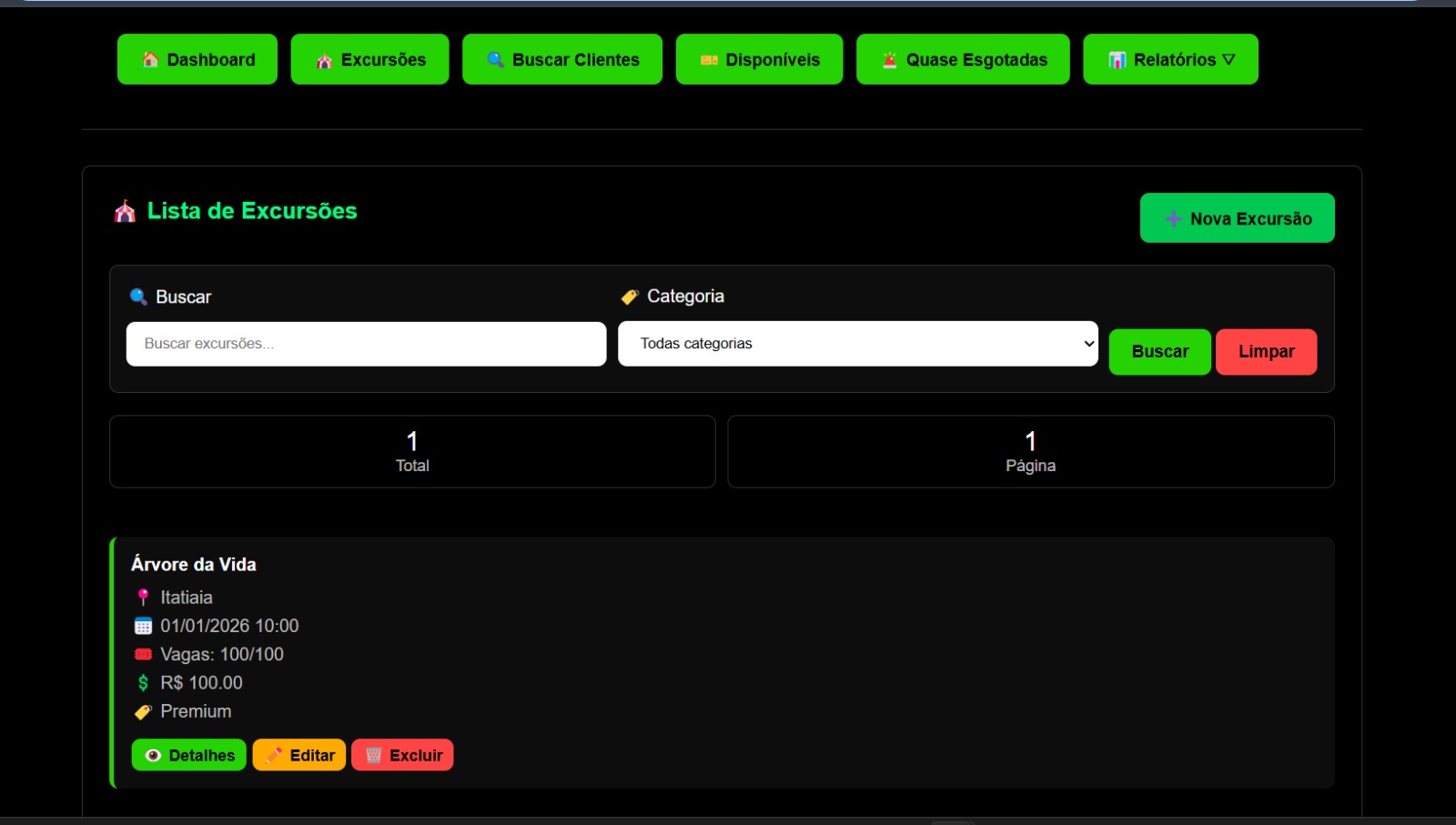


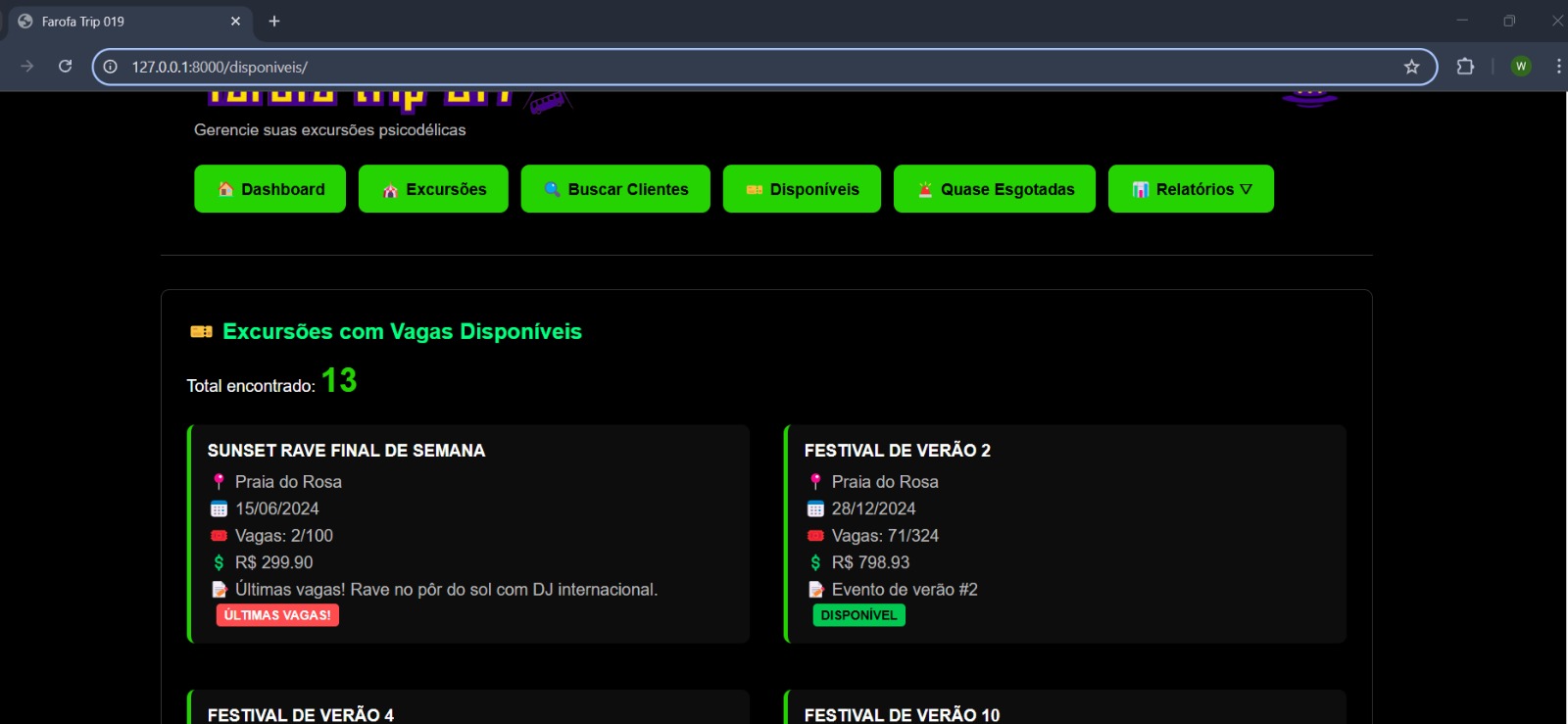
Dashboard



Calendario de excursões

Lista de excursões



Excursões com vagas disponiveis

# Considerações Finais

O desenvolvimento do sistema Farofa Trip permitiu aplicar, de forma prática, os conceitos estudados ao longo do semestre, principalmente no que diz respeito à modelagem de sistemas, arquitetura de software, banco de dados não relacional, metodologias ágeis e desenvolvimento web. Ao longo do projeto, a equipe enfrentou desafios reais, como dificuldades de comunicação, atrasos nas entregas e necessidade de reorganização do planejamento, o que proporcionou um aprendizado significativo sobre trabalho em equipe e gestão de projetos.

A utilização do Scrum como metodologia principal contribuiu para estruturar as etapas do trabalho, permitindo acompanhar a evolução do projeto através de sprints, backlog de tarefas e reuniões de planejamento. A cada Sprint, foi possível revisar processos, identificar pontos de melhoria e alinhar as prioridades, fortalecendo não apenas o produto final, mas também a maturidade da equipe.

Apesar dos obstáculos encontrados, o grupo conseguiu consolidar um sistema funcional, com base de dados organizada, interface estruturada e funcionalidades essenciais já implementadas e prototipadas. O projeto demonstrou a importância de uma comunicação clara, divisão equilibrada de responsabilidades e registro contínuo do progresso.

**Contribuições Individuais**

Willian - Design UI/UX.

⦁ Atualização periódica da documentação do projeto.

⦁ Organizou o fluxo do projeto e conduziu rituais ágeis.

⦁ Desenvolveu protótipos das interfaces e estruturou grande parte da documentação (requisitos, sprints, backlog, evidências, relatórios).

⦁ Apoio no alinhamento do time e acompanhamento das entregas.

Eduardo - Desenvolvedor Frontend/ Backend e controle de versionamento.

⦁ Implementação das telas em HTML/CSS e integração com o back-end.

⦁ Implementação das regras de negócio e estrutura das API's.

⦁ Desenvolvimento de endpoints.

⦁ Suporte em testes e ajustes das funcionalidades.

⦁ Responsável pela organização do repositório e versionamento no GitHub.

Bruno - Developer Database

⦁ Desenvolvimento da modelagem do banco de dados não relacional.

⦁ Criação das coleções, relacionamentos e otimização das estruturas.

⦁ Definição de padrões de consulta, armazenamento e consistência dos dados.

Marcos - Analista de Dados

⦁ Suporte na modelagem de Banco de Dados NoSQL.

⦁ Suporte na documentação do projeto.

# Referências Bibliográficas

AMARAL, D. C. et al. **Gerenciamento Ágil de Projetos**: aplicações em produtos inovadores. São Paulo: Saraiva, 2011.

BROWN, T. ***Desing thinking*:** uma metodologia poderosa para decretar o fim das velhas ideias**.** Rio de Janeiro: Editora Alta Books, 2020.

BUZAN, T. **Mapas Mentais Para os Negócios**: Revolucione Sua Atividade Empresarial e a Maneira Como Você Trabalha. São Paulo: Cultrix, 2017.

CAROLI, P. **Direto ao ponto: criando produtos de forma enxuta**. São Paulo: Editora Casa do Código, 2015.

CAMARGO, R. **PM Visual**: Project Model Visual – Gestão de projetos simples e eficaz. 2. ed. São Paulo: Saraiva Educação, 2019.

COUTINHO, Heitor. **Da estratégia ágil aos resultados**. São Paulo: Saraiva, 2019.

CRUZ, F. PMO Ágil: **Escritório Ágil de Gerenciamento de Projetos**. São Paulo: BRASPORT, 2016.

CRUZ, F. **SCRUM e Agile em Projetos Guia Completo**: conquiste sua certificação e aprenda a usar métodos ágeis no seu dia a dia. 2 ed. São Paulo: Brasport, 2018.

CRUZ, F. **Scrum e PMBOK unidos no Gerenciamento de Projetos**. São Paulo: Brasport, 2013.

MATSUMOTO, Mayra, **MVP**: a forma eficiente de criar, validar e comercializar um novo produto ou serviço, Kindle: Amazon.com, 2020.

OSTERWALDER, A. **Business model generation: inovação em modelos de negócios**. Rio de Janeiro: Editora Alta Books, 2011.

PMI. **Guia PMBOK®**: U**m Guia para o Conjunto de Conhecimentos em Gerenciamento de Projetos**. 7ª. ed. Pennsylvania: PMI, 2021.

PMI. **Guia de Prática Ágil**. EUA: Project Management Institute, 2018.

RIES, E. **A Startup Enxuta.** São Paulo: Editora Leya, 2019.

SBROCCO, Jose. H. T. C.; MACEDO, Paulo C. de. **Metodologias ágeis: Engenharia de Software sob medida**. São Paulo: Érica, 2012.

SEBRAE, CANVAS: Como estruturar seu modelo de negócio. **SEBRAEPR,** 2024. Disponível em: <https://www.sebraepr.com.br/canvas-como-estruturar-seu-modelo-de-negocios/>. Acesso em: 25 maio 2024.

SCRUMSTUDY, **Guia SBOK: Guia para o Conhecimento em Scrum**, Arizona, EUA: SCRUMstudy, 2016. <https://www.Scrumstudy.com/SBOK/SCRUMstudy-SBOK-Guide-2016-Portuguese.pdf>. Acesso em: 08 mar. 2024.

SUTHERLAND, J. **Scrum**: a arte de fazer o dobro do trabalho na metade do tempo. Rio de Janeiro: Sextante, 2019.

SUTHERLAND, J. J. **Scrum**: guia prático, Sextante, Rio de Janeiro, 2020.

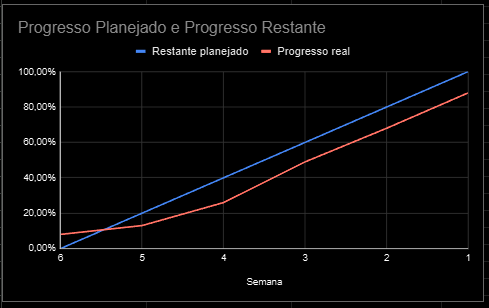
# Anexo I - Diário de bordo

([PI 3º SEMESTRE - Farofa Trip | Trello)](https://trello.com/b/LwBZYqVF/pi-3-semestre)

([Eduardo1Martins/PI-Grupo-03: Projeto Integrador do Curso de Graduação em Desenvolvimento de Software Multiplataforma do 2º semestre de 2025](https://github.com/Eduardo1Martins/PI-Grupo-03))

([brunorod07/Farofa-Trip-Dashboard: Respositório destinado a armazenar o dashboard do Pprojeto Farofa Trip para o P.I do 3º SEM DSM - FATEC Araras](https://github.com/brunorod07/Farofa-Trip-Dashboard))

# Anexo II – Cronograma efetivo



# Anexo III – Evidências

<https://fatecspgov.sharepoint.com/:w:/r/sites/Section_AGO021.A888.N.139.290.20252/_layouts/15/Doc2.aspx?action=edit&sourcedoc=%7Bced94108-cf4f-42b1-97a3-c299b89f9ac7%7D&wdOrigin=TEAMS-MAGLEV.teamsSdk_ns.rwc&wdExp=TEAMS-TREATMENT&wdhostclicktime=1764793147995&web=1>