UNIBRASIL

Eros Netto, Gabriel, Luigi Karwel, Ronald Lucas, Victor Hugo

**Cantinho Verde**

Curitiba, 2025

UNIBRASIL

Eros Netto, Gabriel, Luigi Karwel, Ronald Lucas, Victor Hugo

**Cantinho Verde**

Relatório técnico apresentado como requisito parcial para obtenção de aprovação na disciplina Prática Profissional em Programação WEB, no Curso de Engenharia de Software, na Faculdade Unibrasil.

Prof. Rodrigo Da Silva Do Nascimento

Curitiba, 2025

**RESUMO**



Este relatório apresenta o desenvolvimento completo de um workshop voltado à temática de plantas, abrangendo espécies variadas e suas aplicações. O projeto está sendo construído do zero, englobando desde a concepção dos wireframes e definição da arquitetura de software até a organização da equipe e definição de papeis

Palavras-chave: Workshop. Plantas. Desenvolvimento.

**SUMÁRIO**

1 INTRODUÇÃO ...................................................................................................5  
2 DESENVOLVIMENTO DE TODOS OS WIREFRAMES DO PROJETO ............6  
  2.1 Página Inicial ............................................................................................6  
  2.2 Produtos…................................................................................................6  
  2.3 Categorias……………...............................................................................7  
  2.4 Sobre.........................................................................................................8  
  2.5 Entrar em contato......................................................................................8

  2.6 Conta/Usuário ...........................................................................................9  
  2.7 Carrinho de Compras.................................................................................9  
  2.8 Entrega……..............................................................................................10  
  2.9 Checkout...................................................................................................10  
  2.0.1 Confirmação...........................................................................................11

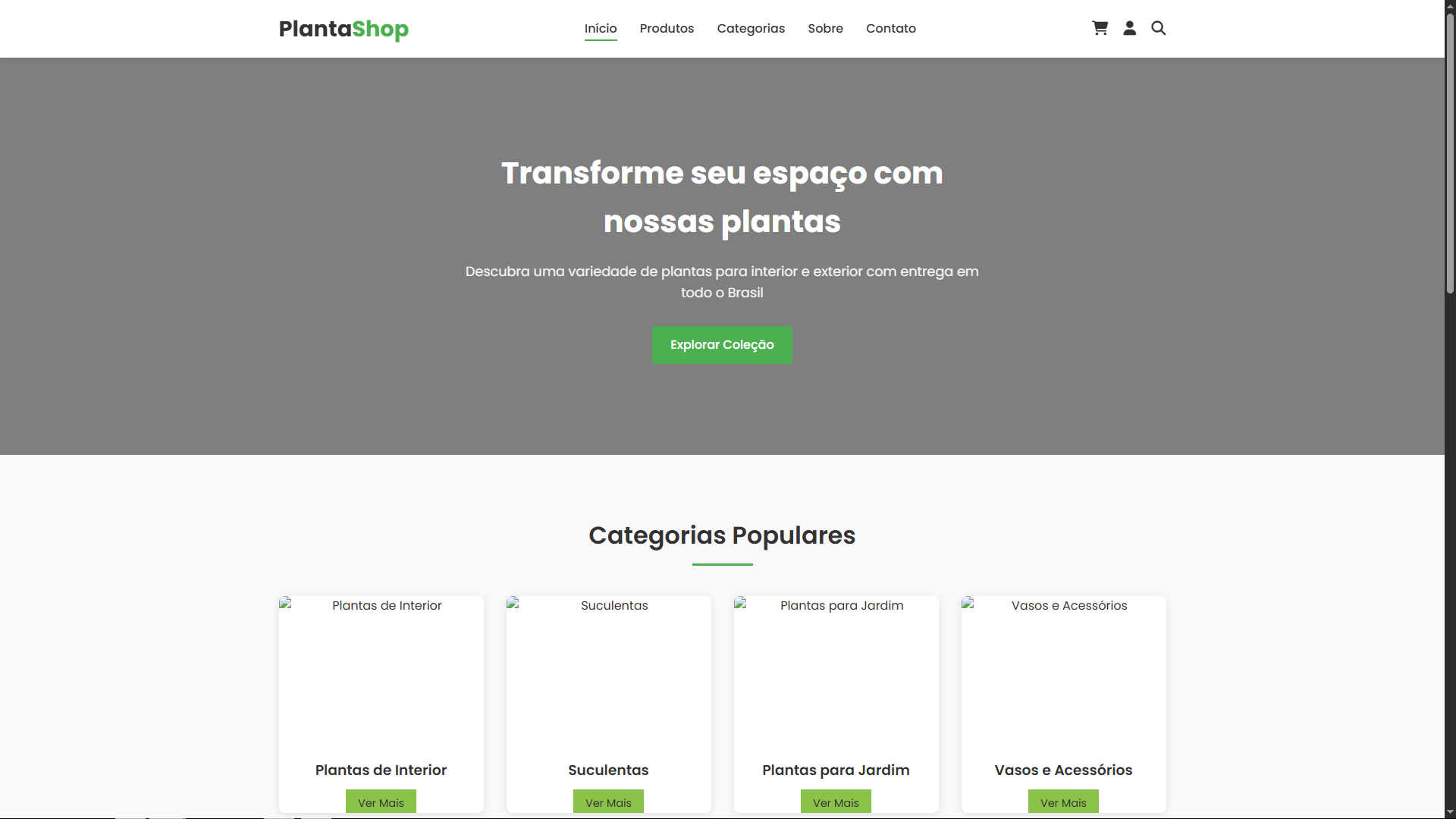
  2.0.2 Login………............................................................................................11  
  2.0.3 Admin…..................................................................................................12  
3 ARQUITETURA ..................................................................................................13  
  3.1 Componentes do Sistema ........................................................................14  
  3.2 Banco de Dados .......................................................................................15  
4 DELINEAMENTO DO PROJETO .......................................................................19  
  4.1 Divisão das Tarefas ..................................................................................19  
  4.2 Tecnologias Utilizadas ..............................................................................21  
5 Git e GitHub ........................................................................................................21

6 REFERÊNCIAS ..................................................................................................22

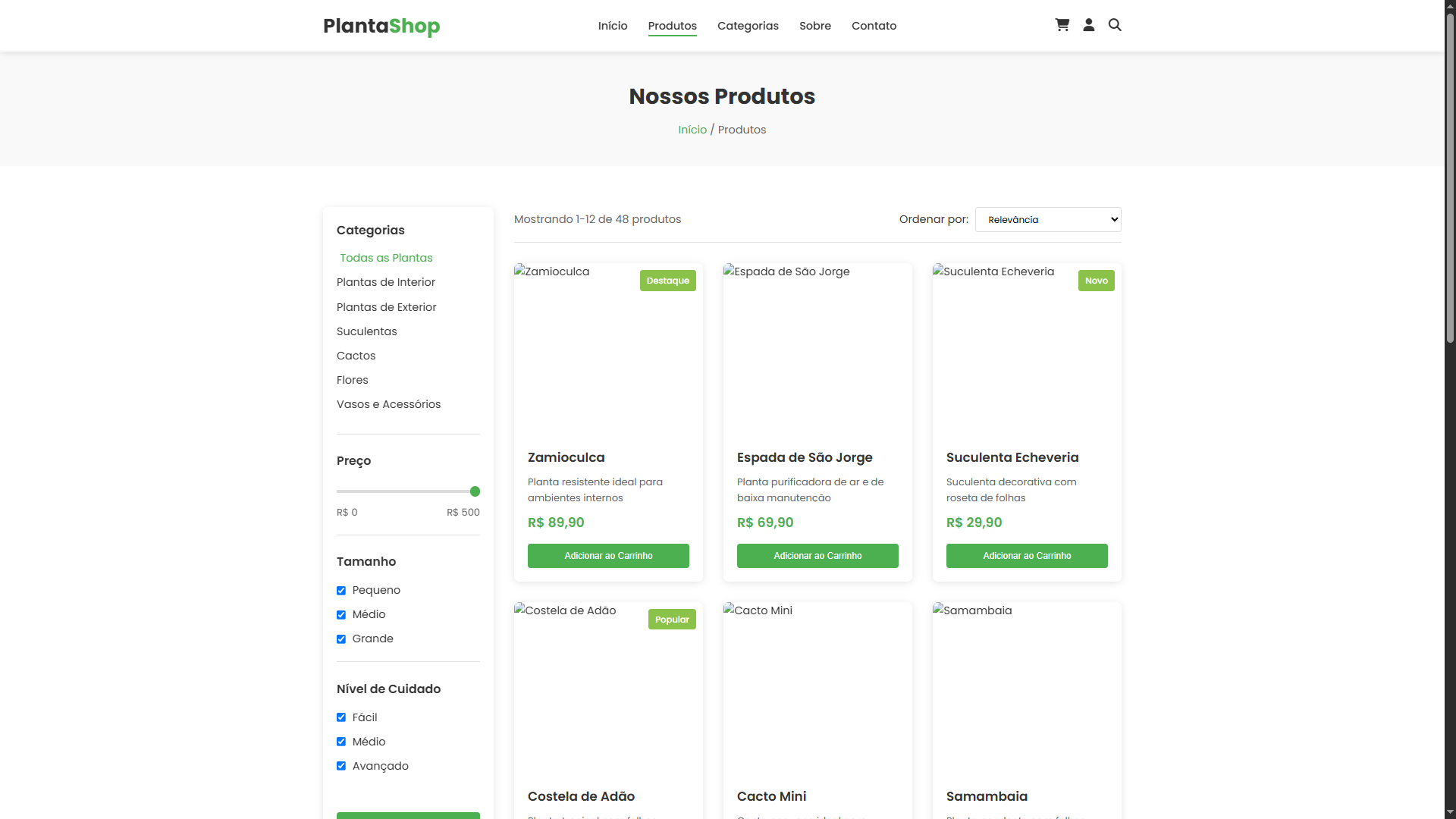
1. **INTRODUÇÃO**

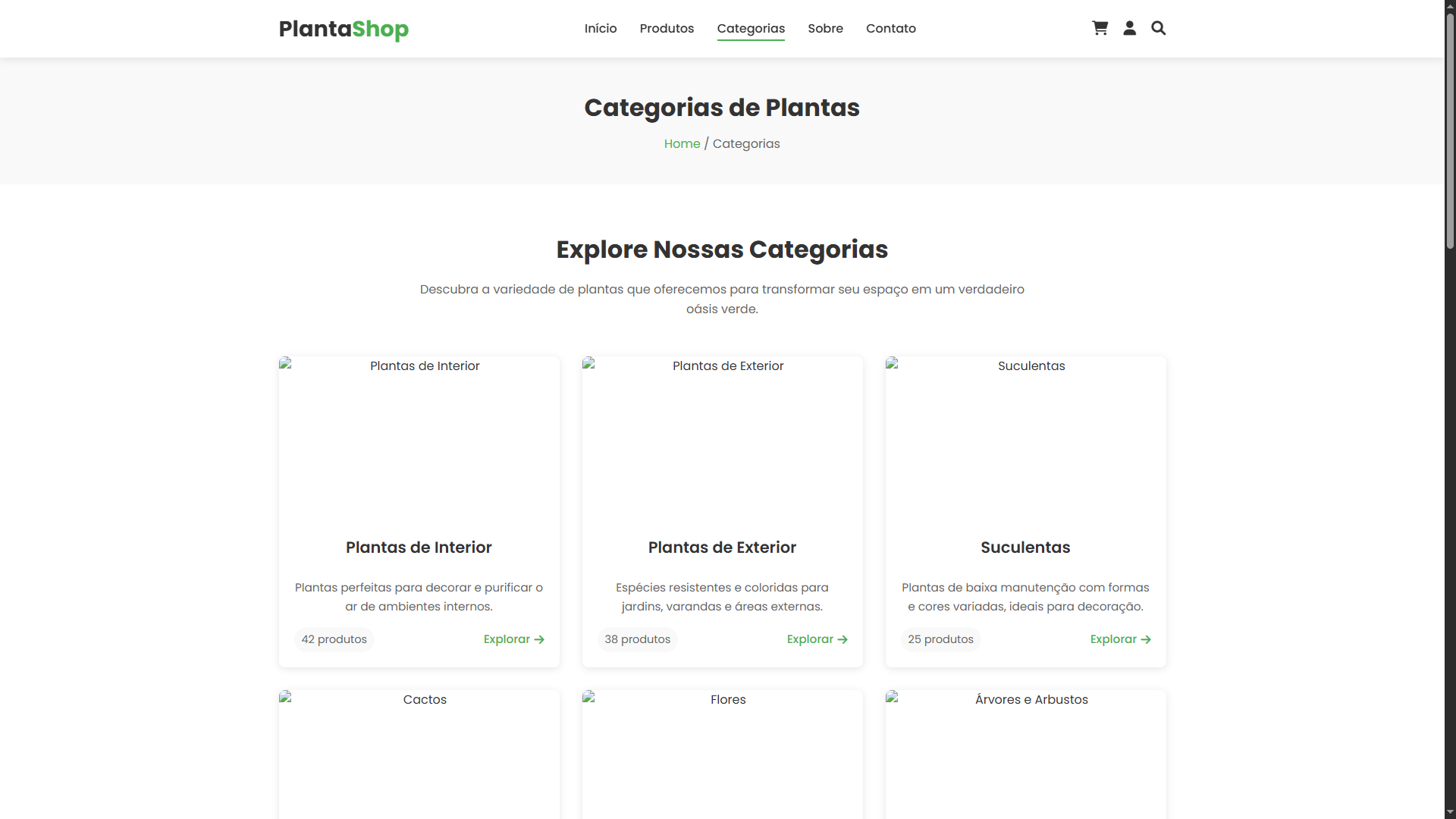
A nossa proposta é desenvolver um sistema que envolve um workshop intuitivo e funcional, que oferece aos usuários a possibilidade de navegar por categorias de plantas, visualizar detalhes dos produtos, adicionar itens ao carrinho, realizar login ou cadastro, além de simular um processo de compra. Em paralelo, contempla um painel administrativo para gestão de produtos e usuários.

**2 Desenvolvimento de todos os wireframes do projeto**

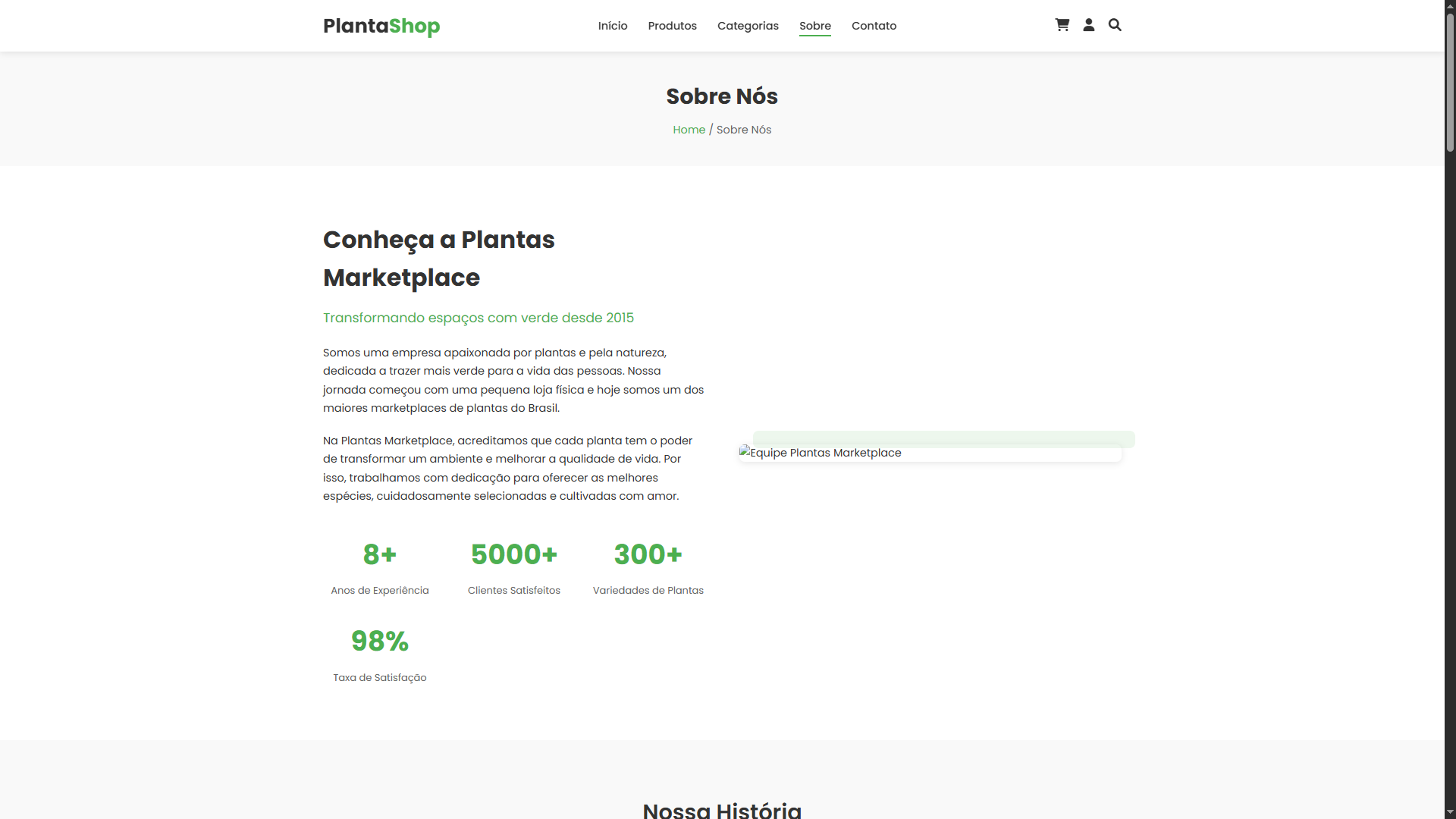
2.1 Página Inicial ****

2.2 Produtos

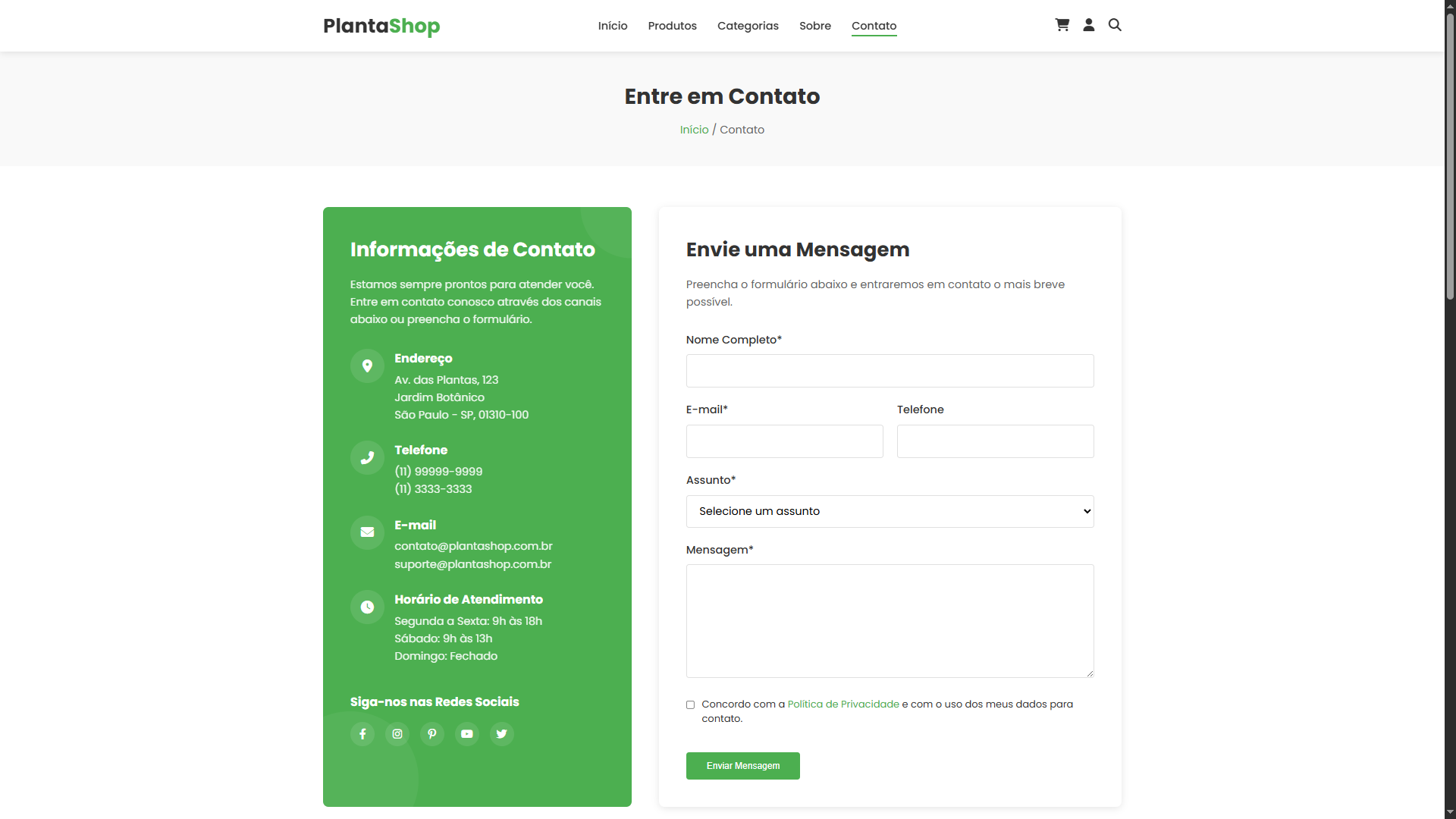
****

2.3 Categorias

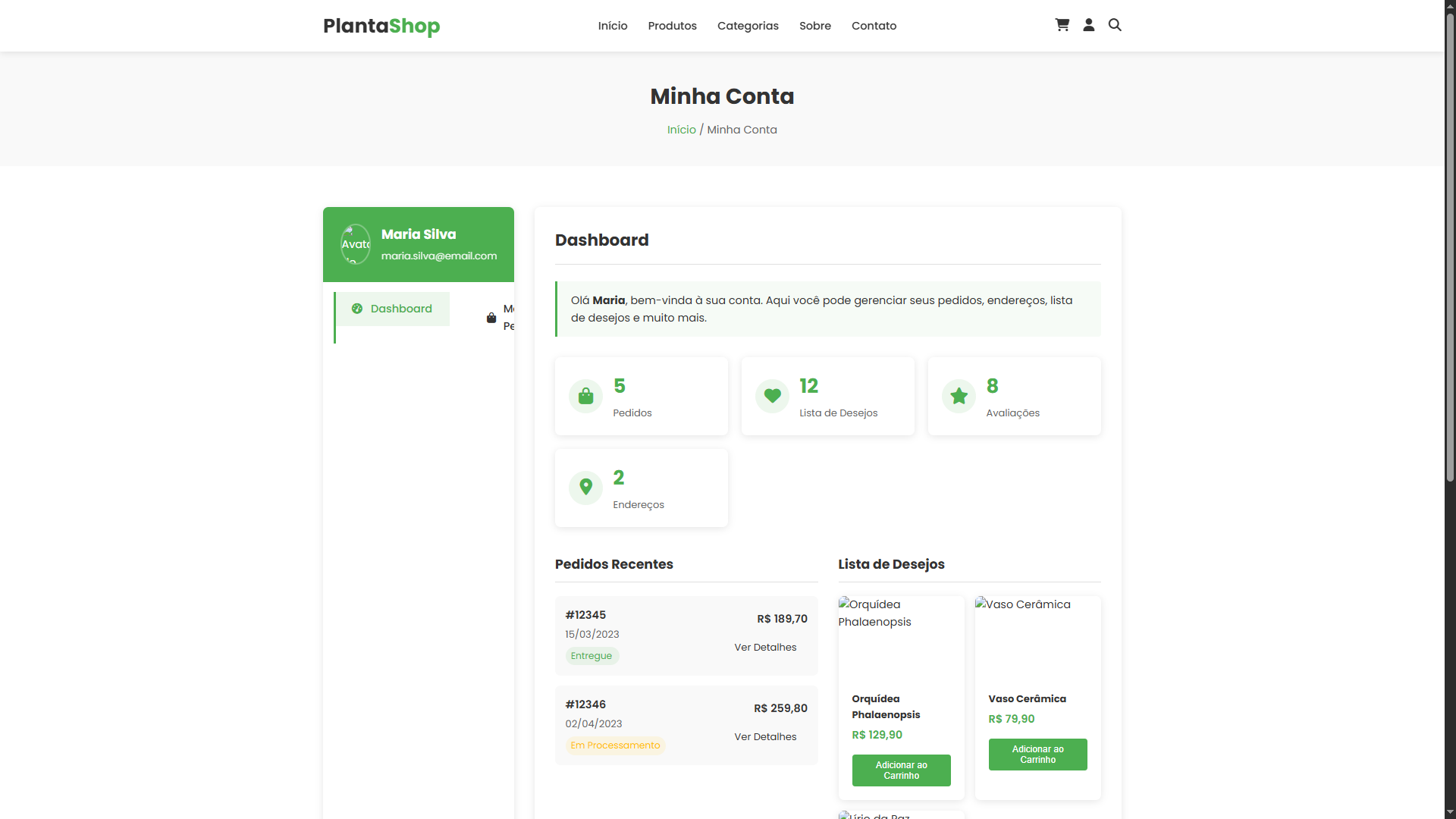
2.4 Sobre



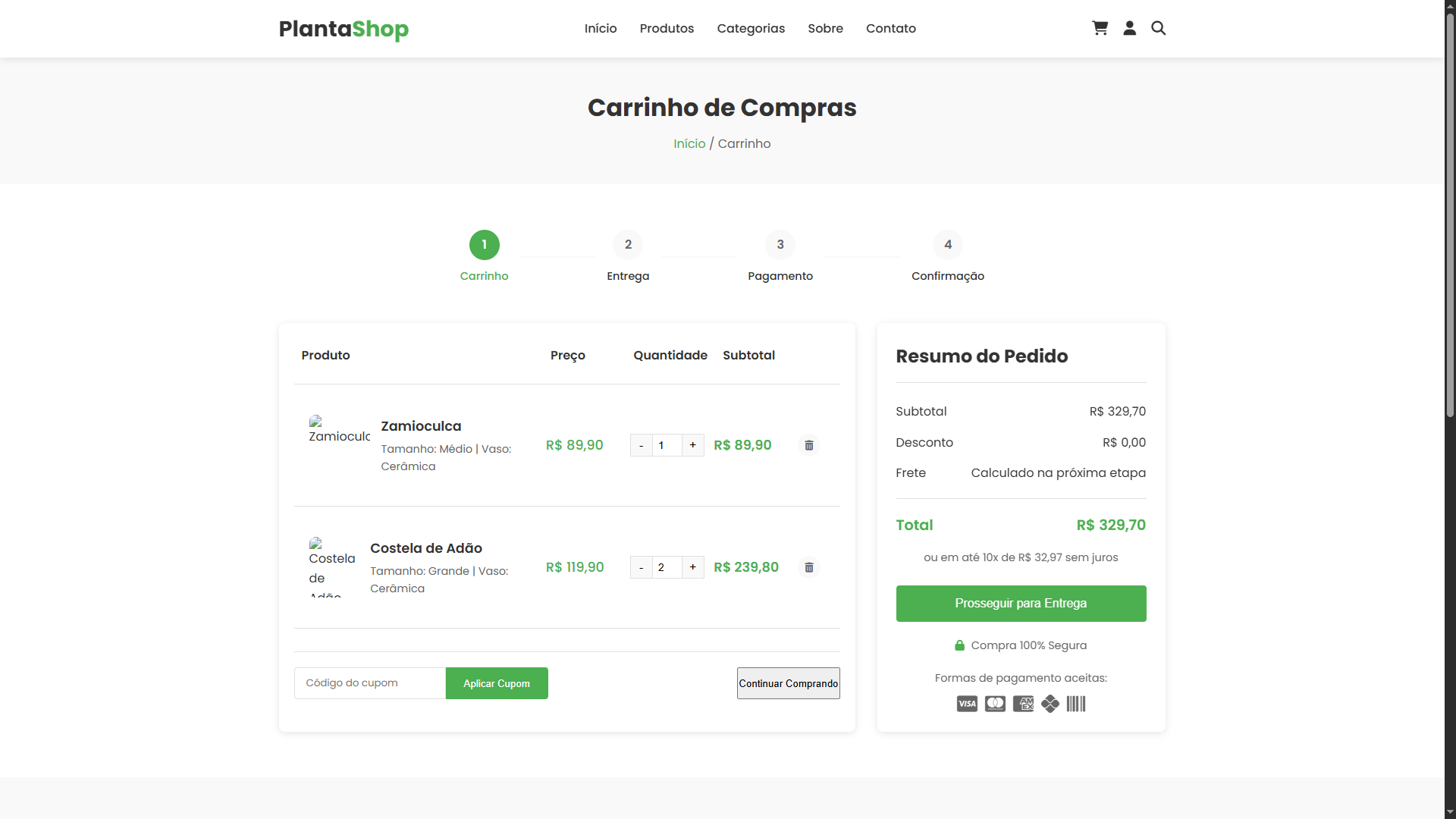
2.5 Entrar em contato



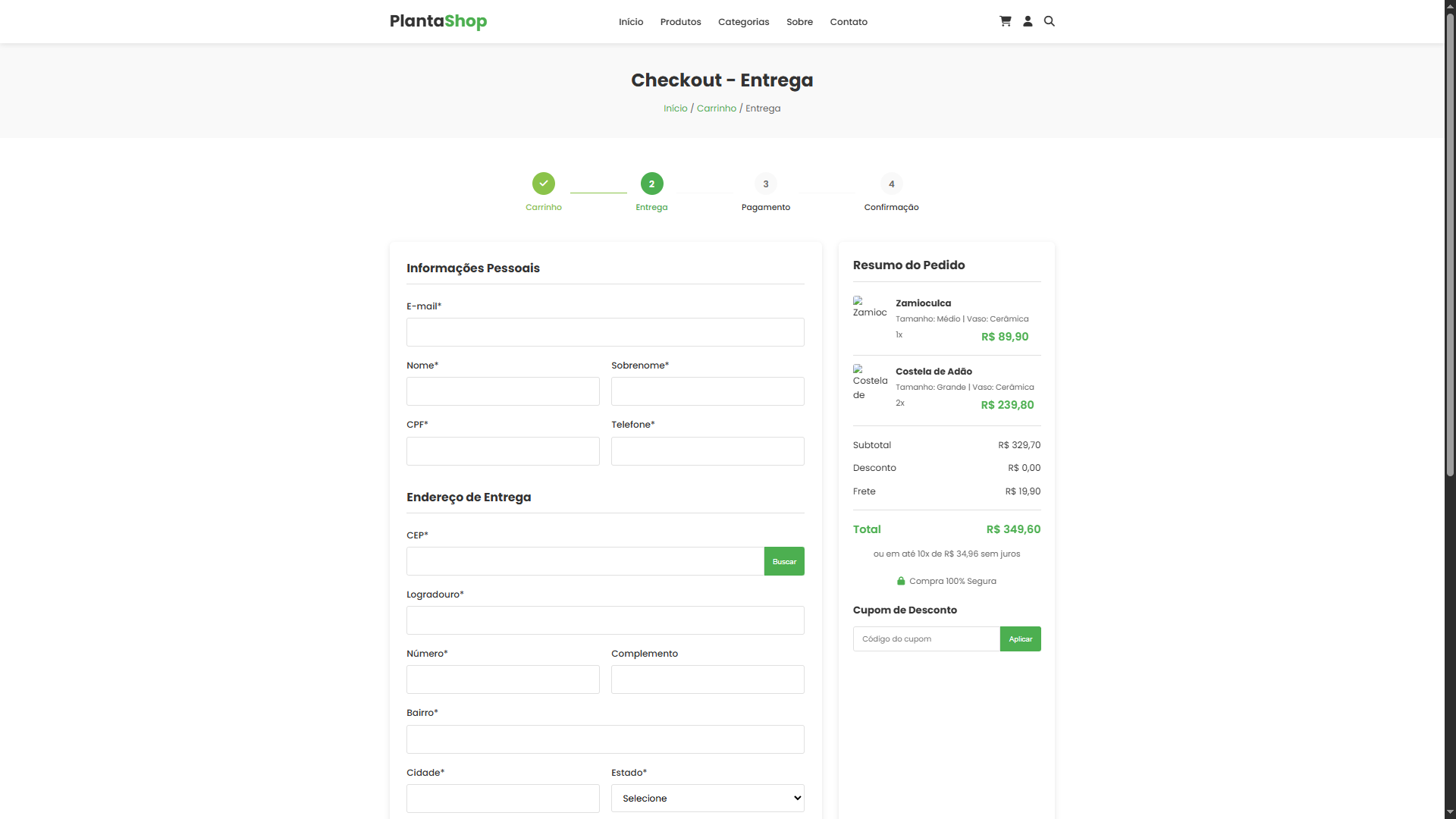
2.6 Conta / Usuário

****

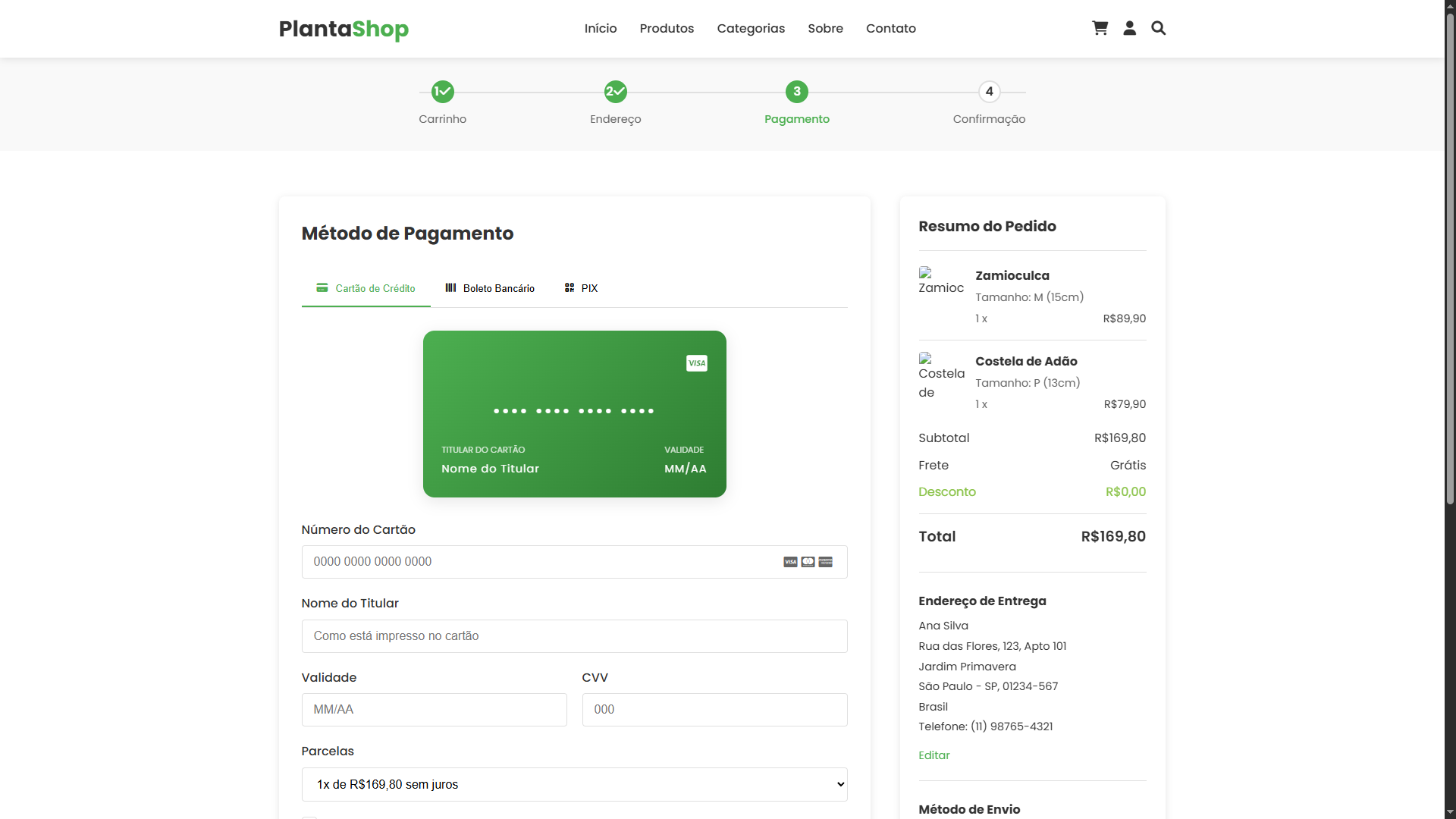
2.7 Carrinho de Compras

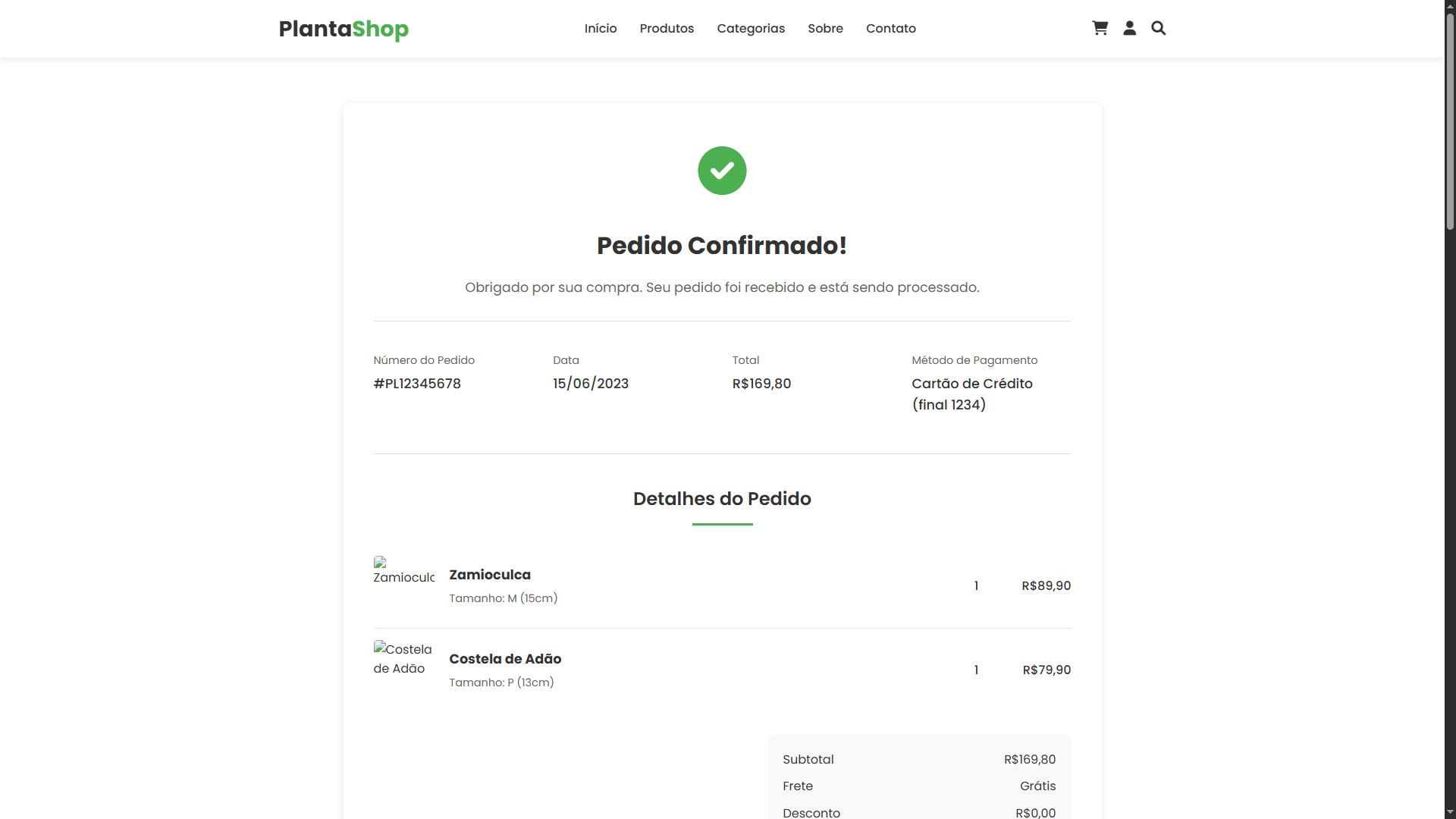
****

2.8 Entrega

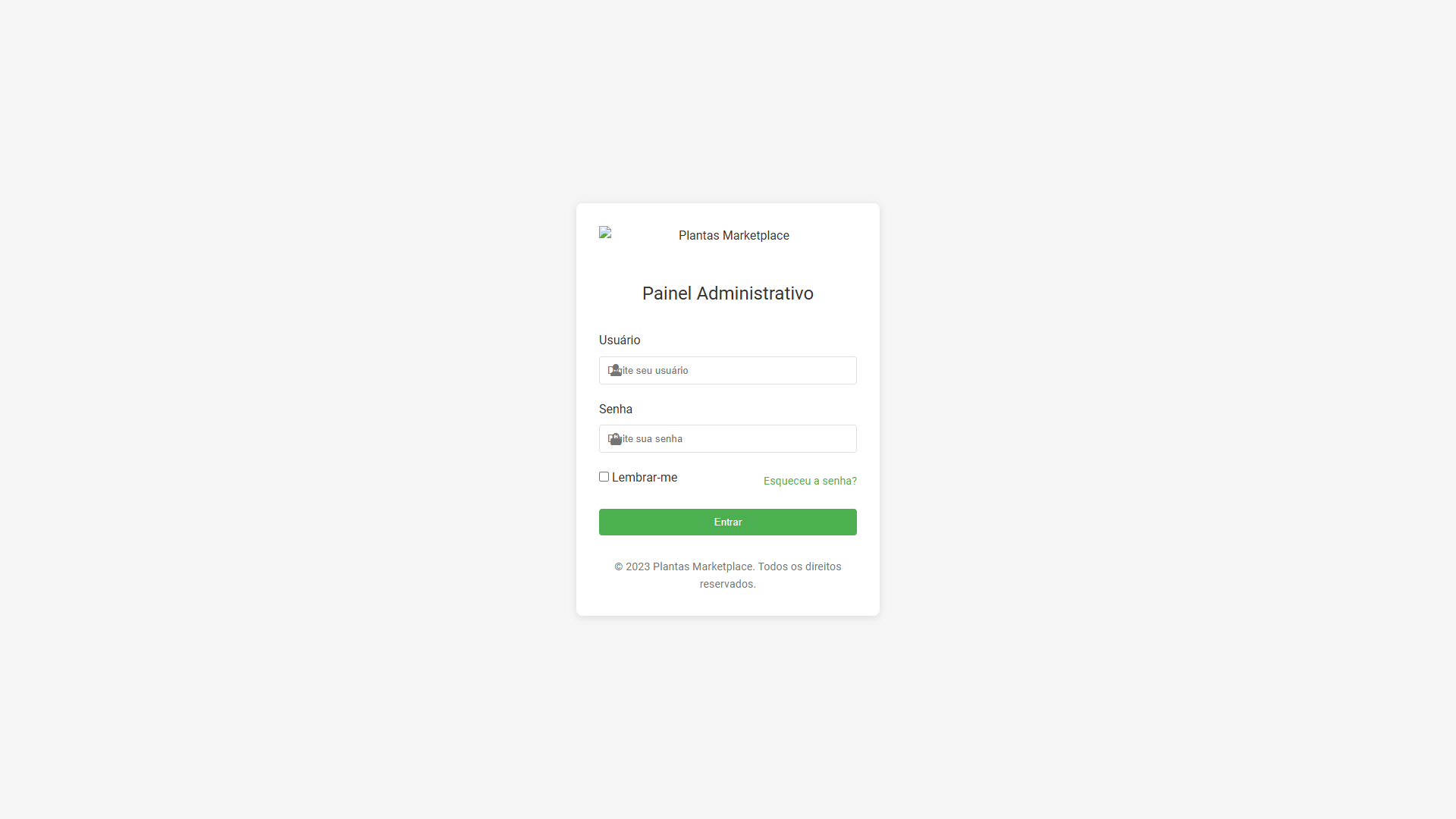
****

2.9 Checkout

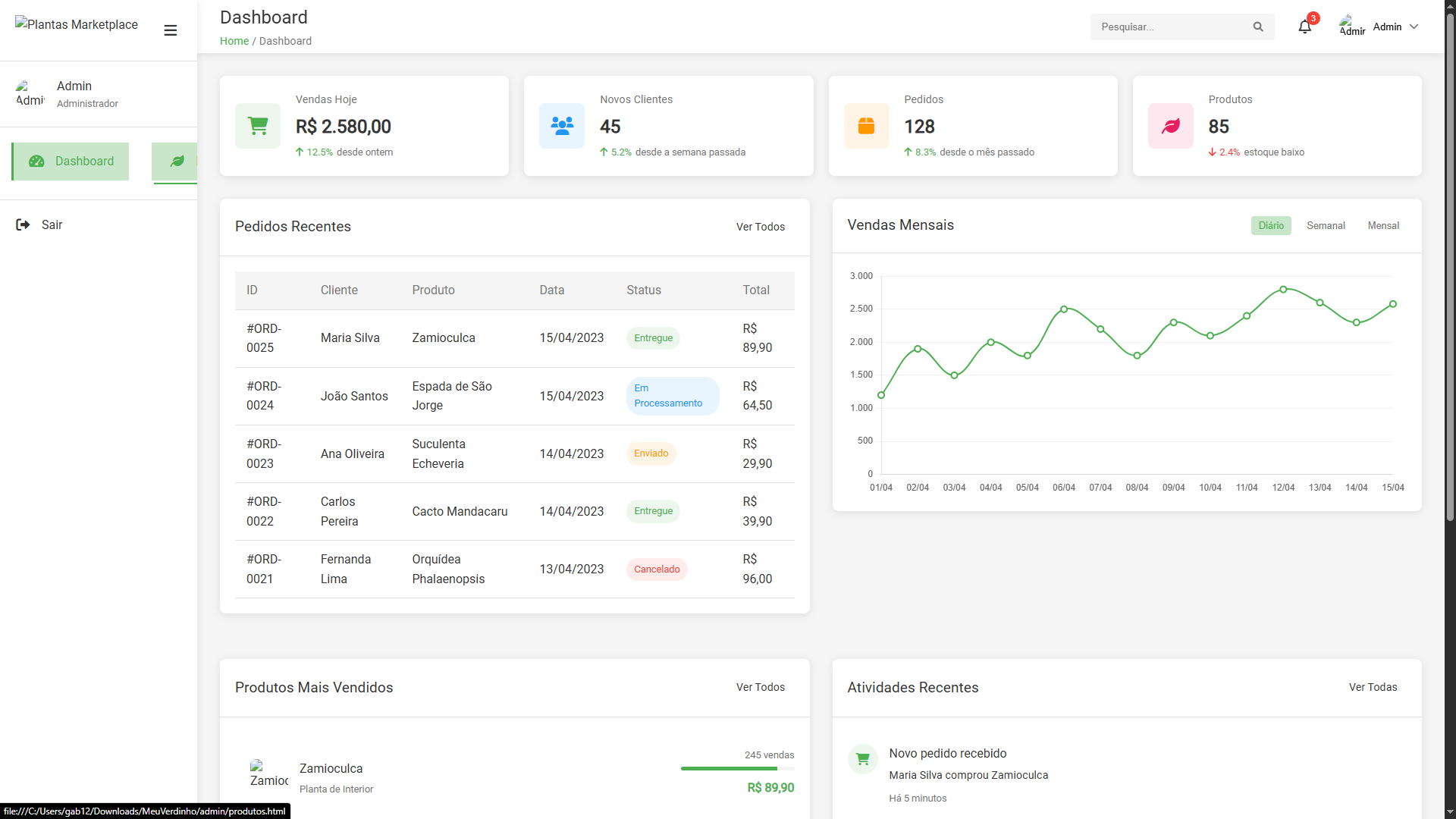


2.0.1 Confirmação

2.0.2 Login

****

2.0.3 Admin



**3. ARQUITETURA**

A arquitetura do sistema é dividida em três camadas principais, cada uma com sua função e tecnologias específicas. A seguir, descrevem-se as camadas que compõem o projeto:

1. Modelo MVC
   * Estrutura MVC para organizar código e separar responsabilidades na aplicação.
2. Camada de Apresentação (Frontend)
   * Tecnologias: HTML5, CSS3, JavaScript, Font Awesome.
   * Função: Esta camada é responsável pela interface gráfica e interação com o usuário. Ela exibe os dados fornecidos pela camada de lógica de negócio e possibilita a navegação entre as páginas do sistema.
3. Camada de Lógica de Negócio (Backend)
   * Tecnologia: PHP.
   * Função: A camada de lógica de negócio é encarregada do processamento das regras de negócio. Ela também gerencia as requisições e respostas do sistema, interligando a camada de apresentação com a camada de dados.
4. Camada de Dados (Persistência)
   * Tecnologia: MySQL e XAMPP.
   * Função: A camada de dados tem como função o armazenamento e a recuperação dos dados estruturados, como produtos, categorias, usuários e pedidos, garantindo a persistência e a integridade dos dados dentro do sistema.

3.1 COMPONENTES DO SISTEMA

O sistema é composto pelos seguintes arquivos e diretórios, cada um com sua função específica:

1. **index.php** – Página inicial com destaques.  
   Função: Exibe a página inicial do sistema, com informações e destaques sobre os produtos.
2. **produtos.php** – Lista de produtos.  
   Função: Apresenta uma lista de todos os produtos disponíveis no sistema.
3. **categorias.php** – Lista de produtos filtrados por categoria.  
   Função: Exibe produtos de uma categoria específica, permitindo o filtro dos mesmos.
4. **conta.php** – Página de login/cadastro.  
   Função: Interface para login de usuários existentes ou para cadastro de novos usuários.
5. **carrinho.php** – Exibição de produtos no carrinho.  
   Função: Exibe os produtos adicionados ao carrinho de compras do usuário.
6. **admin/** – Painel de administração.  
   Função: Área destinada aos administradores para gerenciar os produtos, categorias e usuários.
7. **db.php** – Conexão com banco de dados.  
   Função: Arquivo responsável pela conexão entre o sistema e o banco de dados MySQL.
8. **style.css** – Estilização da interface.  
   Função: Arquivo CSS responsável pela aparência visual do sistema, incluindo layout e design.

3.2 BANCO DE DADOS

O banco de dados é estruturado com tabelas específicas para o armazenamento de informações do sistema. A seguir, estão descritas as tabelas principais:

**Modelagem de Banco de Dados - Sistema de E-commerce**

O modelo de dados apresentado representa a estrutura de um sistema de e-commerce, com foco em gerenciamento de usuários, produtos, pedidos, pagamentos e demais recursos essenciais para uma loja virtual. Abaixo, descrevemos as entidades, seus principais atributos e os relacionamentos entre elas.

**Entidade**: usuarios

Armazena os dados dos usuários da plataforma.

**Atributos:** id, nome, sobrenome, email, senha, telefone, data\_nascimento, tipo

**Observações:**

O campo tipo diferença entre clientes e administradores. O email é único.

**Entidade:** enderecos

Contém os endereços vinculados a um usuário.

**Atributos:** id, usuario\_id, tipo, cep, logradouro, numero, complemento, bairro, cidade, estado

**Relacionamento:** Muitos endereços para um usuário (usuario\_id → usuarios.id)

**Regra:** Um endereço pode ser de entrega, cobrança ou ambos.

**Entidade:** categorias

Representa as categorias dos produtos.

**Atributos:** id, nome, descricao, imagem, status

**Entidade:** produtos

Armazena os produtos disponíveis na loja.

**Atributos:** id, nome, descricao, preco, quantidade\_estoque, peso, dimensoes, nivel\_cuidado, tamanho, ambiente, luz, agua, ativo, data\_cadastro, data\_atualizacao

**Observações:** O produto pode ter diversos atributos específicos relacionados a cuidados, ambiente, luz e água, reforçando a ideia de que o sistema é voltado a plantas ou itens de jardinagem.

**Entidade:** produto\_imagens

Armazena as imagens associadas a produtos.

**Atributos:** id, produto\_id, url

**Relacionamento:** Muitos para um produto (produto\_id → produtos.id)

**Entidade:** produto\_categoria

Relacionamento N:N entre produtos e categorias.

**Atributos:** produto\_id, categoria\_id

**Relacionamento:** Produto pode pertencer a várias categorias e vice-versa.

**Entidade:** carrinhos

Representa o carrinho de compras de um usuário.

**Atributos:** id, usuario\_id, data\_criacao, data\_atualizacao

**Relacionamento:** Um carrinho por usuário (pode ser nulo se anônimo).

**Entidade:** carrinho\_itens

Itens adicionados ao carrinho.

**Atributos:** id, carrinho\_id, produto\_id, quantidade, preco\_unitario

**Relacionamentos:** Muitos itens para um carrinho. Cada item refere-se a um produto.

**Entidade:** lista\_desejos

Armazena os produtos que o usuário deseja futuramente.

**Atributos:** id, usuario\_id, produto\_id, data\_adicao

**Relacionamentos:** Um usuário pode ter vários produtos na lista de desejos. Cada produto pode estar na lista de vários usuários.

**Regra:** A combinação usuario\_id + produto\_id é única.

**Entidade:** pedidos

Representa os pedidos feitos por usuários.

**Atributos:** id, usuario\_id, endereco\_entrega\_id, codigo, subtotal, total, status, metodo\_pagamento, codigo\_rastreio, data\_pedido, data\_atualizacao

**Relacionamentos:** Um pedido está ligado a um usuário e a um endereço de entrega.

**Entidade:** pedido\_itens

Itens contidos em cada pedido.

**Atributos:** id, pedido\_id, produto\_id, quantidade, preco\_unitario, subtotal

**Relacionamentos:** Muitos itens para um pedido, cada item refere-se a um produto.

**Entidade:** pedido\_status\_historico

Mantém o histórico de status dos pedidos.

**Atributos:** id, pedido\_id, status, comentario, data\_alteracao

**Relacionamento:** Um histórico por status de um pedido.

**Entidade:** pagamentos

Armazena os dados dos pagamentos.

**Atributos:** id, pedido\_id, metodo, status, valor, codigo\_transacao, data\_pagamento, data\_atualizacao

**Relacionamento:** Cada pagamento está vinculado a um pedido.

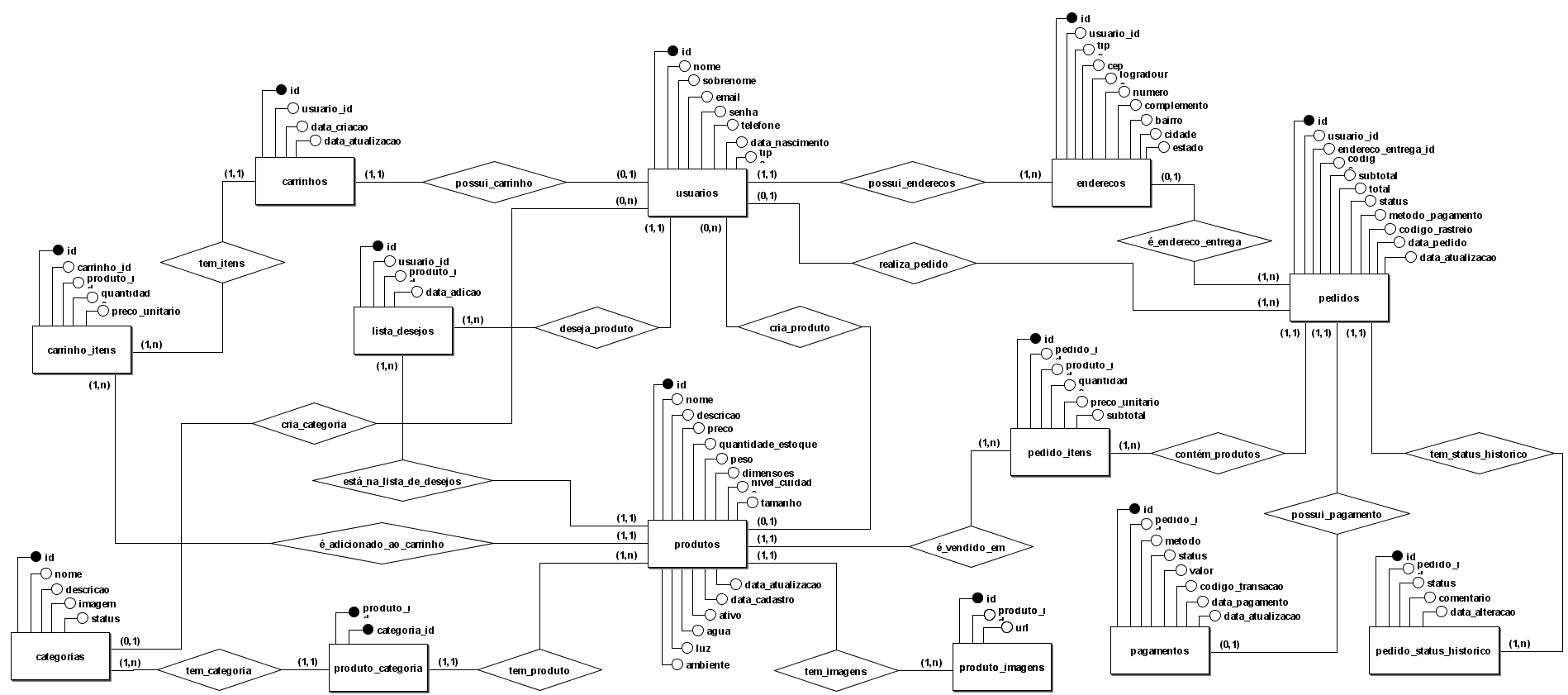
**Resumo dos Relacionamentos**

Usuário pode ter: Vários endereços, um carrinho, vários pedidos, uma lista de desejos com vários produtos.

Produto pode: Ter várias imagens, estar em várias categorias, estar em carrinhos, pedidos e listas de desejos.

Pedido: Tem um ou mais itens, tem um endereço de entrega, pode ter vários status históricos, está ligado a um pagamento

**Diagrama de Relacionamento**



**4. Delineamento do Projeto**

A equipe foi estruturada com responsabilidades bem definidas para garantir o bom andamento do desenvolvimento do projeto.

4.1 Divisão das tarefas

**Desenvolvimento dos Wireframes do Projeto – Responsável: [Gabriel]**

O membro ficou encarregado de:

* Criar os wireframes das principais telas do projeto: página inicial, catálogo de plantas, página de produto, carrinho de compras e páginas de login/cadastro.
* Utilizar ferramentas como Excalidraw ou draw.io para elaboração dos wireframes.
* Validar os fluxos de navegação e usabilidade com o grupo, garantindo que as interações fossem intuitivas e eficientes.

**Levantamento da Arquitetura de Software – Responsável: [Ronald]**

Este membro realizou:

* Descrição da arquitetura de software adotada para o desenvolvimento do projeto.
* Definição da arquitetura em camadas (apresentação, lógica de negócio e persistência).
* Estudo do fluxo de requisições dentro do sistema e definição de como os dados fluem entre as camadas.
* Identificação dos componentes do sistema e sua interação.
* Modelagem do banco de dados e definição das tecnologias a serem utilizadas (Frontend, Backend, Banco de Dados, Hospedagem).

**Levantamento das Entidades, Atributos e Relacionamentos – Responsável: [Eros]**

Atividades incluíram:

* Modelagem do banco de dados, incluindo a definição das entidades principais (como Usuário, Produto, Pedido, etc.).
* Definição dos atributos de cada entidade e os relacionamentos entre elas.
* Garantir que os dados necessários para o sistema fossem adequadamente representados e estruturados no banco de dados.

**Criação do Repositório Git e Contato com o GitHub – Responsável: [Eros]**

Responsabilidades:

* Criação do repositório principal no GitHub para versionamento do código.
* Definição de convenções de commits e organização das branches de desenvolvimento.
* Gerenciamento dos pull requests, garantindo que o código fosse revisado e aprovado antes de ser integrado ao repositório principal.

**Delineamento do Desenvolvimento pela Equipe – Responsável: [Vitor Hugo]**

Este membro coordenou:

* Descrição da função de cada membro dentro do projeto, garantindo que todas as responsabilidades estivessem bem definidas.
* Definição dos objetivos do projeto e organização do fluxo de trabalho.
* Acompanhamento do andamento das atividades, garantindo que os prazos fossem cumpridos e que o progresso fosse monitorado.
* Criação dos slides para a apresentação final do projeto, reunindo os principais pontos para a defesa.

**Divisão dos Papéis na Equipe – Responsável: [Luigi]**

Responsável por:

* Definir as funções de cada integrante da equipe (por exemplo, desenvolvedores Front-end, Back-end, e Design).
* Garantir a comunicação eficiente entre todos os membros da equipe, promovendo reuniões de alinhamento e resolução de dúvidas.
* Formatar o relatório e fazer os ajustes necessários.

4.2 TECNOLOGIAS UTILIZADAS

O desenvolvimento do sistema utilizou as seguintes tecnologias:

* **Frontend**: HTML, CSS, JavaScript, Font Awesome.
* **Backend**: PHP.
* **Banco de dados**: MySQL.
* **Hospedagem**: Apache / XAMPP / LAMP.

5 GIT E GITHUB

Para organização e versionamento do projeto, foi criado um repositório no GitHub, uma plataforma amplamente utilizada para hospedagem de código e colaboração em projetos de software. O repositório permite o controle de versões, facilitando o desenvolvimento em equipe, o registro de alterações, e o acompanhamento da evolução do projeto.

Durante a criação, foi definido um nome representativo do projeto e adicionado um arquivo README.md com a descrição do sistema, além de uma. O repositório também foi configurado como público, permitindo o acesso ao código por membros da equipe ou professores para fins de avaliação.

Link para o repositório no GitHub:

🔗 https://github.com/ErosNetto/e-commerce-prog-web

**6. REFERÊNCIAS**

ALMEIDA, Maria Elizabeth Bianconcini de. *Engenharia de software: conceitos e práticas*. 2. ed. São Paulo: Saraiva, 2018.

SOMMERVILLE, Ian. *Engenharia de software*. 10. ed. São Paulo: Pearson, 2019.

WORLD WIDE WEB CONSORTIUM (W3C). *HTML5 – A vocabulary and associated APIs for HTML and XHTML*. 2014. Disponível em: https://www.w3.org/TR/html5/. Acesso em: 15 abr. 2025.

MOZILLA. *CSS: cascading style sheets*. [S. l.]: MDN Web Docs, [2024?]. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/CSS. Acesso em: 15 abr. 2025.

ORACLE. *MySQL 8.0 Reference Manual*. [S. l.], 2024. Disponível em: https://dev.mysql.com/doc/. Acesso em: 15 abr. 2025.

PHP DOCUMENTATION GROUP. *PHP Manual*. [S. l.], 2024. Disponível em: https://www.php.net/manual/pt\_BR/. Acesso em: 15 abr. 2025.

GITHUB. *Getting started with GitHub*. [S. l.]: GitHub Docs, 2024. Disponível em: https://docs.github.com/pt. Acesso em: 15 abr. 2025.

DRAW.IO. *Documentação oficial*. [S. l.]: diagrams.net, [2024?]. Disponível em: https://www.diagrams.net/doc/. Acesso em: 15 abr. 2025.

FONT AWESOME. *Font Awesome icons documentation*. [S. l.], 2024. Disponível em: https://fontawesome.com/. Acesso em: 15 abr. 2025.