

**UNIVERSIDADE DE SÃO PAULO**

ARTHUR YANG TUNG - 14559819

FILIPPE VALERIANO DE OLIVEIRA – 14570701

RENAN MOURA NASCIMENTO - 14748921

THE DJR DJRJAN BRITO – 13688367

KEVIN TAMAYOSE - 14669711

**Relatório da entrega da apresentação de andamento do EP da disciplina  
Banco de Dados 2**

SÃO PAULO

2025

ARTHUR YANG TUNG - 14559819

FILIPPE VALERIANO DE OLIVEIRA – 14570701

RENAN MOURA NASCIMENTO - 14748921

THE DJR DJRJAN BRITO – 13688367

KEVIN TAMAYOSE - 14669711

**Relatório da entrega da apresentação de andamento do EP da disciplina  
Banco de Dados 2**

Relatório entrega parcial do exercício  
programa da disciplina ACH2025 do  
curso de Sistemas de Informação da  
Escola de Artes Ciências e Humanidades  
da Universidade de São Paulo.

Professor: Luciano Vieira Araújo

SÃO PAULO

2025

## Sumário

1. Introdução .....	4
2. Arquitetura do Sistema.....	4
3. Modelagem de Dados .....	4
3.1 Modelo Entidade-Relacionamento (DER) .....	4
3.2 Modelo Lógico.....	5
3.3 Principais Tabelas .....	6
4. Funcionalidades Implementadas.....	6
4.1 Adoção de Animais.....	6
4.2 Loja Virtual.....	6
4.3 Serviços .....	6
5. Uso de Inteligência Artificial .....	6
6. Backup e Restauração .....	7
7. Otimizações Futuras .....	7
8. Próximos Passos.....	8
9. Considerações Finais.....	8
Divisão de Tarefas entre os Membros.....	8

# 1. INTRODUÇÃO

O projeto **PetMatch** foi idealizado com o objetivo de desenvolver uma plataforma web para intermediar a adoção de animais e oferecer serviços e produtos voltados ao público pet. A solução abrange funcionalidades completas de cadastro, adoção, venda de produtos e agendamento de serviços com prestadores verificados.

A motivação principal foi desenvolver um sistema que ajudasse a resolver um problema real — o abandono de animais e a dificuldade em encontrar serviços confiáveis — combinando tecnologia com responsabilidade social.

## 2. ARQUITETURA DO SISTEMA

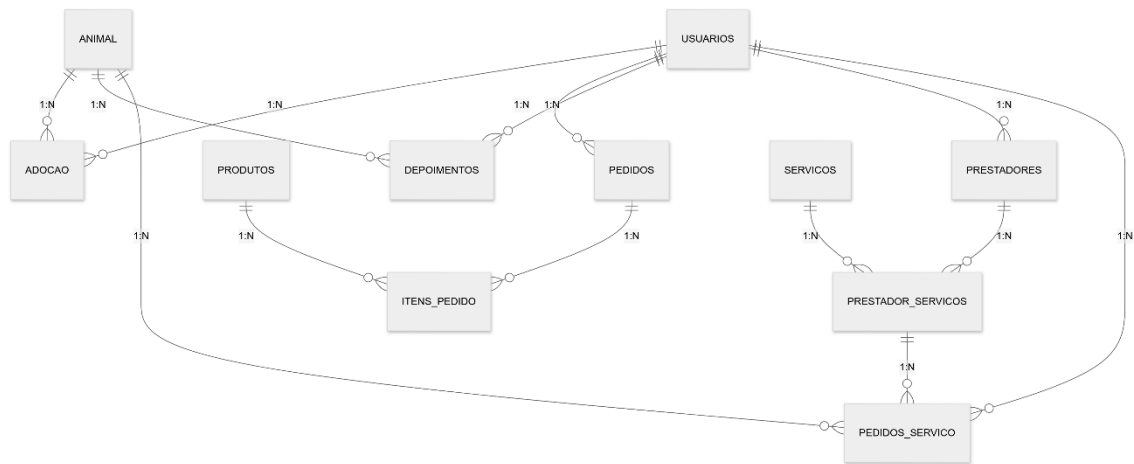
O sistema foi estruturado em camadas, com a seguinte arquitetura:

- **Frontend:** HTML, CSS, com templates Jinja2 integrados ao Flask.
- **Backend:** Python com Flask, responsável pelo roteamento, renderização de páginas, integração com banco de dados e API.
- **Banco de Dados:** PostgreSQL, modelado com foco em integridade referencial, normalização e escalabilidade.
- **Ferramentas de apoio:** WSL para ambiente Linux, pgAdmin para gerenciamento de dados, e pg\_restore para restauração de backups.

## 3. MODELAGEM DE DADOS

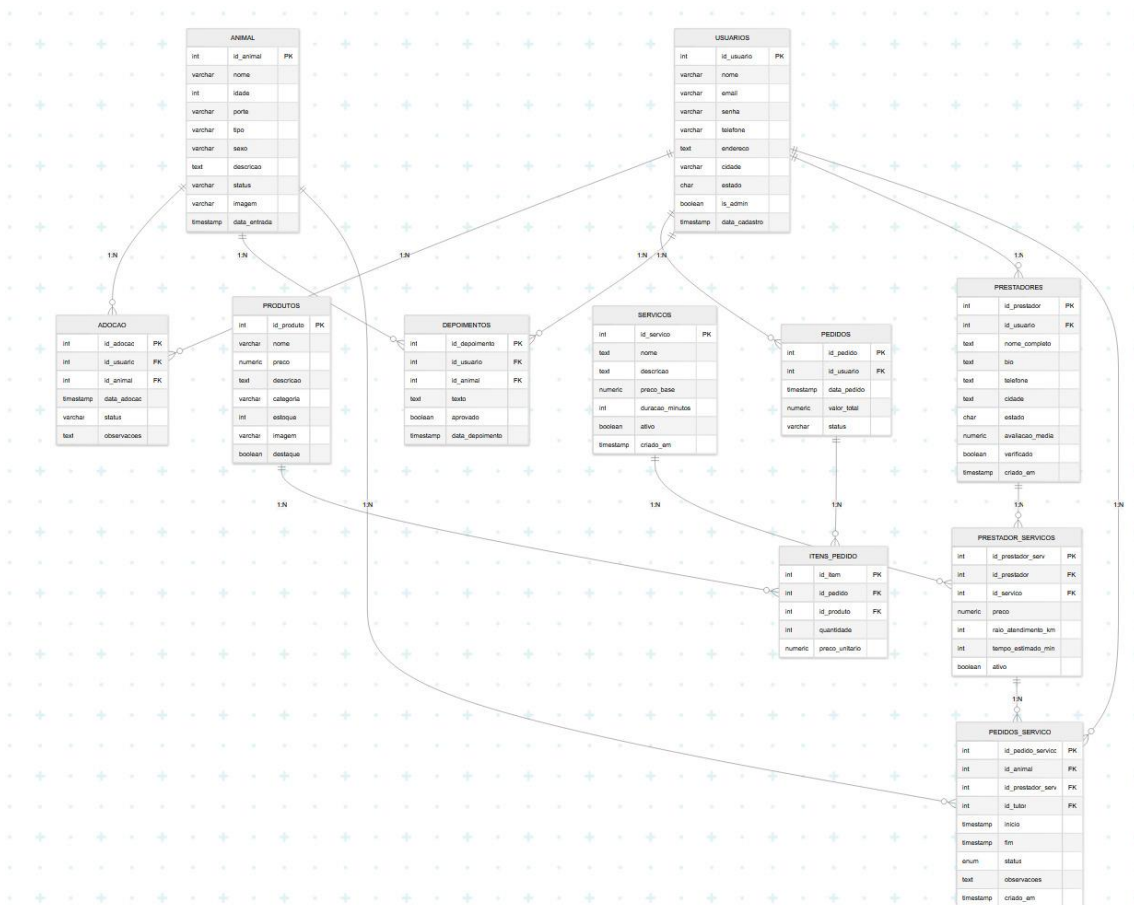
### 3.1 MODELO ENTIDADE-RELACIONAMENTO (DER)

O DER representa as entidades principais do sistema, como usuários, animais, produtos e serviços, e suas respectivas relações. A modelagem seguiu as boas práticas de normalização e uso de chaves estrangeiras.



### 3.2 MODELO LÓGICO

O modelo lógico foi descrito em Mermaid (formato ERD) com todos os atributos das tabelas e relacionamentos. A construção do modelo permitiu visualizar as dependências e estruturas antes da implementação física no banco de dados.



### **3.3 PRINCIPAIS TABELAS**

- usuarios: armazena informações de tutores, administradores e prestadores.
- animal: animais disponíveis para adoção.
- adocao: histórico e status de adoções.
- produtos: catálogo de itens disponíveis para venda.
- pedidos e itens\_pedido: pedidos feitos pelos usuários.
- servicos, prestadores, prestador\_servicos e pedidos\_servico: estrutura completa de agendamento e oferta de serviços.

## **4. FUNCIONALIDADES IMPLEMENTADAS**

### **4.1 ADOÇÃO DE ANIMAIS**

- Cadastro de animais com status, tipo, porte e imagem.
- Listagem dos animais em destaque na página inicial.
- Página individual para cada pet com botão "Quero Adotar".

### **4.2 LOJA VIRTUAL**

- Exibição de produtos com imagem, descrição e preço.
- Carrinho funcional implementado com JavaScript.
- Estrutura para pedidos e itens de pedido já presente no banco.

### **4.3 SERVIÇOS**

- Cadastro de serviços com preço base e duração.
- Relacionamento com prestadores.
- Tabela de pedidos de serviços com status de confirmação.

## **5. USO DE INTELIGÊNCIA ARTIFICIAL**

Diversas IAs foram utilizadas para acelerar e otimizar o desenvolvimento:

- **Claude:** auxílio na escrita de código Python, rotas Flask e integração com o banco.
- **Copilot:** sugestões de código em tempo real dentro do VSCode.
- **ChatGPT:** modelagem de banco, explicações técnicas, correções de erro, otimização de queries e geração de dados de teste.
- **Mermaid :** geração automática do modelo lógico visual com base nas queries SQL.
- **Claude:** Foi especialmente útil para gerar e revisar trechos de código mais complexos, como funções de backend em Flask e estruturas de banco de dados. Ele foi consultado principalmente para verificar a lógica dos relacionamentos e propor soluções para problemas no código.

## 6. BACKUP E RESTAURAÇÃO

Para garantir a persistência e recuperação de dados, foram utilizados comandos PostgreSQL como:

- `pg_dump` para gerar o backup completo do banco.
- `pg_restore` para restaurar o banco com estrutura e dados a partir do `.dump`.

## 7. OTIMIZAÇÕES FUTURAS

Três estratégias principais foram definidas:

1. **Índices:** criar índices para colunas frequentemente utilizadas em joins e filtros.
2. **Cache:** evitar consultas repetidas utilizando cache local ou via Redis.
3. **Views materializadas:** gerar visões prontas para consultas agregadas ou complexas.

## 8. PRÓXIMOS PASSOS

- Finalizar CRUD completo de usuários e pets.
  - Concluir o frontend das páginas de produtos, serviços e pedidos.
  - Implementar autenticação segura com hashing de senha e login de administrador.
  - Criar painel administrativo para controle das adoções e produtos.
- 

## 9. CONSIDERAÇÕES FINAIS

O PetMatch é um projeto ambicioso que une responsabilidade social com desenvolvimento tecnológico. Durante o processo, foram empregadas boas práticas de modelagem, programação, organização do código e uso de IA para apoiar decisões técnicas.

O sistema já está estruturado com forte base de dados, arquitetura bem definida e frontend funcional. As próximas etapas serão focadas na usabilidade, segurança e experiência do usuário.

## DIVISÃO DE TAREFAS ENTRE OS MEMBROS

### **Renan Moura Nascimento - 14748921**

Responsável pela estrutura geral do projeto, organização do banco de dados, criação dos modelos lógico e DER, além da implementação do backend em Flask. Renan também coordenou o uso de IA para apoiar o desenvolvimento, além de montar e ajustar os modelos no Mermaid.

### **The Djrdjrjan Brito - 13688367**

Contribuiu ativamente na construção do frontend, utilizando HTML e CSS. Participou da estruturação das páginas de adoção, loja de produtos e exibição dos serviços. Também colaborou na adaptação visual para responsividade inicial do site.

### **Kevin Tamayose - 14669711**



Focado na parte de modelagem do banco e integração com o PostgreSQL. Ajudou na definição das entidades e na criação dos scripts SQL. Também participou dos testes iniciais de inserção de dados e restauração do banco com pg\_restore.

**Arthur Yang Tung - 14559819**

Trabalhou na implementação da lógica de agendamento de serviços e no relacionamento entre prestadores, serviços e usuários. Também participou do desenvolvimento das rotas e da lógica de pedidos no backend.

**Filipe Valeriano Batista de Oliveira - 14570701**

Responsável pelos testes, verificação da consistência dos dados no banco e parte da documentação. Ajudou na escrita de trechos descritivos do projeto e na montagem da justificativa e motivação da ideia, além de contribuir com o roteiro para a apresentação.