ENGENHARIA DE SOFTWARE: PROJETO PETCARE

1. Resumo do MVP

O sistema conecta tutores de animais a cuidadores.

- **Cadastro:** Usuários se cadastram como Tutor ou Cuidador.
- **Gestão de Pets:** Tutores cadastram seus animais.
- **Busca:** Tutores localizam cuidadores próximos.
- **Agendamento:** Tutores solicitam horários; Cuidadores aceitam ou recusam.
- **Avaliação:** Tutores avaliam o serviço após a conclusão.

---

2. User Stories (Histórias de Usuário)

Autenticação

- Como **usuário**, quero me cadastrar e fazer login para acessar o sistema.
- Como **usuário**, quero escolher meu perfil (Tutor ou Cuidador) durante o cadastro.
- Como **usuário**, quero gerenciar um perfil simples com minhas informações básicas.

Gestão de Pets

- Como **Tutor**, quero cadastrar meus pets para vinculá-los aos agendamentos.
- Como **Tutor**, quero editar ou remover as informações dos meus pets.

Busca de Cuidadores

- Como **Tutor**, quero buscar cuidadores baseados em minha localização geográfica.
- Como **Tutor**, quero visualizar informações essenciais do cuidador (nome, localização, nota média).

Agendamento

- Como **Tutor**, quero solicitar um agendamento com um cuidador específico.
- Como **Tutor**, quero acompanhar o status das minhas solicitações.
- Como **Cuidador**, quero ter a opção de aceitar ou recusar solicitações de agendamento.

Avaliações (Reviews)

- Como **Tutor**, quero avaliar o cuidador após o término do serviço.
- Como **Cuidador**, quero que as avaliações recebidas sejam exibidas no meu perfil público.

---

3. Diagrama de Entidades e Relacionamentos

Entidades

USUARIO

- id
- nome
- email
- senha
- tipo (Enum: TUTOR | CUIDADOR)
- localizacao

PET

- id
- nome
- tipo
- observacoes
- tutor_id (FK -> USUARIO)

AGENDAMENTO

- id
- data_inicio
- data_fim
- status (Enum: PENDENTE, ACEITO, RECUSADO, CONCLUIDO)
- tutor_id (FK -> USUARIO)
- cuidador_id (FK -> USUARIO)
- pet_id (FK -> PET)

REVIEW

- id
- nota
- comentario
- agendamento_id (FK -> AGENDAMENTO)
- cuidador_id (FK -> USUARIO)

Relacionamentos

- **Usuário (Tutor) 1:N Pets** (Um tutor pode ter vários pets)
- **Usuário (Tutor) 1:N Agendamentos** (Um tutor pode fazer vários agendamentos)
- **Usuário (Cuidador) 1:N Agendamentos** (Um cuidador pode receber vários agendamentos)
- **Agendamento 1:1 Review** (Cada agendamento pode ter apenas uma avaliação)

---

4. Definição da API (Endpoints)

Autenticação

Método	Rota	Objetivo
POST	/api/auth/register	Cadastrar novo usuário (Tutor ou Cuidador)
POST	/api/auth/login	Autenticar usuário e gerar token
GET	/api/auth/me	Retornar dados do usuário logado

Usuários

Método	Rota	Objetivo
GET	/api/users/{id}	Visualizar perfil público de um usuário
PUT	/api/users/{id}	Atualizar dados do próprio perfil

Pets (Módulo do Tutor)

Método	Rota	Objetivo
POST	/api/pets	Cadastrar um novo pet
GET	/api/pets	Listar todos os pets do tutor logado

PUT	/api/pets/{id}	Atualizar dados de um pet
DELETE	/api/pets/{id}	Remover um pet do sistema

Busca de Cuidadores

Método	**Rota**	**Objetivo**
GET	/api/cuidadores	Buscar cuidadores (Filtros: localização)
GET	/api/cuidadores/{id}	Ver detalhes do perfil de um cuidador

Exemplos de Query Params:

- /cuidadores?cidade=SãoPaulo
- /cuidadores?lat=-23.5&lng=-46.6&raio=5

Agendamentos (Booking)

Método	**Rota**	**Objetivo**
POST	/api/agendamentos	Solicitar um novo agendamento
GET	/api/agendamentos	Listar agendamentos (Histórico do usuário)
GET	/api/agendamentos/{id}	Ver detalhes de um agendamento específico
PATCH	/api/agendamentos/{id}/status	Alterar status (Aceitar/Recusar/Concluir)

Reviews

Método	Rota	Objetivo
POST	/api/reviews	Criar avaliação para um serviço concluído
GET	/api/reviews/cuidador/{id}	Listar todas as avaliações de um cuidador

---

5. Fluxo Principal do MVP

Sequência sugerida de implementação e teste das rotas:

1. POST /auth/register (Criar contas)
2. POST /auth/login (Obter acesso)
3. POST /pets (Tutor cadastra animal)
4. GET /cuidadores (Tutor encontra profissional)
5. POST /agendamentos (Solicitação de serviço)
6. PATCH /agendamentos/{id}/status (Confirmação do serviço)
7. POST /reviews (Fechamento do ciclo com avaliação)

---

6. Estrutura de Diretórios (Backend Java)

src/main/java/com/meuprojeto

```
├── config/          # Configurações gerais (Beans, Swagger, etc.)
├── controller/      # Camada de API (Endpoints REST)
├── dto/             # Objetos de Transferência de Dados (Requests/Responses)
├── entity/          # Classes de Domínio (JPA/Hibernate)
├── repository/      # Acesso a Dados (Interfaces Repository)
├── service/         # Regras de Negócio
├── security/        # Configuração de Autenticação e Tokens (JWT)
└── exception/       # Tratamento de Erros Personalizados
```