

Detalhamento Técnico: VetCare+

1. Função: Veterinário

****Responsabilidades:****

- Gerenciar agendamentos.
- Manter histórico médico dos pets.
- Implementar teleconsultas e pagamentos.

****Classes Necessárias:****

1. `Pet`

- ****Atributos:**** `id`, `nome`, `especie`, `raca`, `idade`, `historicoMedico`.
- ****Métodos:**** `adicionarHistorico(medico)`, `listarHistorico()`.

2. `Veterinario`

- ****Atributos:**** `id`, `nome`, `especialidade`, `agenda`.
- ****Métodos:**** `agendarConsulta(pet, horario)`, `listarConsultas()`.

3. `Consulta`

- ****Atributos:**** `id`, `pet`, `veterinario`, `dataHora`, `status`, `detalhes`.
- ****Métodos:**** `marcarConsulta()`, `cancelarConsulta()`, `alterarHorario()`.

4. `Pagamento`

- ****Atributos:**** `id`, `valor`, `metodo`, `status`, `data`.
- ****Métodos:**** `processarPagamento()`, `gerarRecibo()`.

5. `Teleconsulta`

- ****Atributos:**** `id`, `linkVideo`, `pet`, `veterinario`, `horario`.
- ****Métodos:**** `iniciarChamada()`, `finalizarChamada()`.

****Estrutura de Banco de Dados:****

- Tabelas: Pets, Veterinários, Consultas, Histórico Médico, Pagamentos.
- Relacionamentos:
- `Pet` (1:N) `Consulta`
- `Veterinario` (1:N) `Consulta`
- `Consulta` (1:1) `Pagamento`

2. Função: Adoção de Pets

****Responsabilidades:****

- Exibir lista de pets disponíveis.
- Facilitar comunicação com ONGs.
- Agendar visitas.

****Classes Necessárias:****

1. `PetAdocao` (herda de `Pet`)

- ****Atributos:**** `id`, `status`, `descricao`, `fotos`, `ong`.
- ****Métodos:**** `atualizarStatus(status)`, `adicionarFoto(foto)`.

2. `ONG`

- ****Atributos:**** `id`, `nome`, `endereco`, `contato`, `petsDisponiveis`.
- ****Métodos:**** `listarPets()`, `adicionarPet()`, `removerPet()`.

3. `Adocao`

- ****Atributos:**** `id`, `pet`, `adotante`, `status`, `dataInicio`.
- ****Métodos:**** `iniciarProcesso()`, `atualizarStatus()`.

4. `AgendamentoVisita`

- ****Atributos:**** `id`, `pet`, `adotante`, `dataHora`, `status`.

- ****Métodos:**** `marcarVisita()`, `cancelarVisita()`.

5. `Adotante`

- ****Atributos:**** `id`, `nome`, `contato`, `endereco`.

- ****Métodos:**** `enviarFormularioAdocao()`,
`consultarStatusAdocao()`.

****Estrutura de Banco de Dados:****

- Tabelas: Pets para adoção, ONGs, Adotantes, Agendamentos.

- Relacionamentos:

- `PetAdocao` (N:1) `ONG`

- `PetAdocao` (1:1) `Adocao`

- `Adocao` (1:1) `AgendamentoVisita`

3. Função: Venda de Produtos

****Responsabilidades:****

- Gerenciar catálogo de produtos.
- Facilitar compras e entrega.
- Implementar sistema de fidelidade.

****Classes Necessárias:****

1. `Produto`

- ****Atributos:**** `id`, `nome`, `descricao`, `preco`, `estoque`,
`categoria`.
- ****Métodos:**** `atualizarEstoque()`, `calcularDesconto()`.

2. `Carrinho`

- ****Atributos:**** `id`, `itens`, `total`.
- ****Métodos:**** `adicionarProduto(produto)`,

``removeProduto(produto)`, `calcularTotal()`.`

3. ``Pedido``

- ****Atributos:**** ``id`, `cliente`, `produtos`, `status`, `data`, `metodoPagamento`.`

- ****Métodos:**** ``finalizarPedido()`, `cancelarPedido()`.`

4. ``Entrega``

- ****Atributos:**** ``id`, `pedido`, `endereço`, `status`, `rastreamento`.`

- ****Métodos:**** ``atualizarStatus()`, `gerarLinkRastreamento()`.`

5. ``ProgramaFidelidade``

- ****Atributos:**** ``id`, `cliente`, `pontos`.`

- ****Métodos:**** ``acumularPontos(valorCompra)`, `resgatarPontos()`.`

****Estrutura de Banco de Dados:****

- Tabelas: Produtos, Pedidos, Clientes, Fidelidade.

- Relacionamentos:

- ``Pedido` (N:M) `Produto``

- ``Pedido` (1:1) `Entrega``

- ``Cliente` (1:N) `Pedido``

4. Funcionalidades Gerais

****Classes Necessárias:****

1. ``Usuario``

- ****Atributos:**** ``id`, `nome`, `email`, `senha`, `tipo` (veterinário, ONG, cliente).`

- ****Métodos:**** ``autenticar()`, `alterarSenha()`.`

2. ``Notificacao``

- ****Atributos:**** ``id`, `usuario`, `mensagem`, `dataHora`, `tipo`.`

- ****Métodos:**** `enviarNotificacao()`.

3. `Relatorio`

- ****Atributos:**** `id`, `tipo`, `dados`, `data`.

- ****Métodos:**** `gerarRelatorio()`.

4. `RedeSocial`

- ****Atributos:**** `id`, `postagem`, `linkCompartilhamento`.

- ****Métodos:**** `compartilhar()`.

Fluxo de Desenvolvimento

1. ****Backend:****

- Framework: Spring Boot.

- APIs REST: Criar endpoints para operações (ex.: criar consulta, listar produtos, processar adoções).

2. ****Frontend:****

- Framework: React Native.

- Componentes: Interfaces baseadas nas funcionalidades descritas.

3. ****Banco de Dados:****

- Modelo Relacional: PostgreSQL.

- ORM: Hibernate.

4. ****Integrações:****

- Gateway de pagamento, geolocalização, notificação.