

DOCUMENTO DE PROJETO DE SOFTWARE: SISTEMA DE GERENCIAMENTO DE CUIDADOS PARA PETS

SISTEMA DE GERENCIAMENTO DE CUIDADOS PARA PETS



Curso: Ciências da Computação

UC: Modelos, métodos e técnicas de engenharia de software

Alunos: Gustavo Abreu Pereira, Lucas Dal Jovem.
Professor: Henrique Neves.
Campus: Dib Mussi.

Alunos: Lucas Dal Jovem Neyssinger - 1072317036, Gustavo Abreu Pereira - 1072316195

SISTEMA DE GERENCIAMENTO DE CUIDADOS PARA PETS

Documento de Projeto de Software apresentado como requisito para a disciplina de Engenharia de Software.

Resumo

Este documento detalha a proposta de desenvolvimento de um sistema de gerenciamento de cuidados para pets, acessível via aplicativo móvel, destinado a clientes de um pet shop. A problemática central abordada é a dificuldade dos tutores em manter um controle organizado sobre a saúde e os cuidados de seus animais, resultando em esquecimentos de vacinas, consultas e medicação, o que pode levar a problemas de saúde e gastos desnecessários além de registros desorganizados e risco à saúde dos animais. O objetivo do software é oferecer uma ferramenta ao cliente que facilite a sua rotina e, consequentemente, do próprio pet shop, promovendo a fidelização e aprimorando a gestão do relacionamento. A solução proposta envolve o desenvolvimento de um sistema robusto de cadastro e monitoramento, com funcionalidades de alerta e lembrete acessível por computador e dispositivos móveis. O sistema permitirá o cadastro de usuários e pets, o registro de vacinas, consultas e medicações, além de gerar alertas e lembretes automáticos com antecedência mínima de 24 horas.

A metodologia Scrum será empregada para garantir flexibilidade e entrega contínua de valor, com uma equipe multifuncional.

Sumário

DOCUMENTO DE PROJETO DE SOFTWARE: SISTEMA DE GERENCIAMENTO DE CUIDADOS PARA PETS	1
1. INTRODUÇÃO	4
1.1. Contexto da Problemática	4
1.2. Objetivo do Cliente	4
1.3. Justificativa do Projeto	5
2. PROPOSTA DE SOLUÇÃO	6
2.1. Descrição Detalhada da Proposta	6
2.2. Estrutura e Arquitetura do Software	7
2.3. Metodologia de Desenvolvimento	7
2.4. Equipe do Projeto e Atribuições	8
3. REQUISITOS DO SISTEMA	12
3.1. Requisitos Funcionais (RF)	12
3.2. Requisitos Não Funcionais (RNF)	14
4. MODELAGEM DE SOFTWARE	16
4.1. Caso de Uso (UML)	16
4.2. Diagrama de Classes (UML)	17
5. OBJETIVO DO SOFTWARE DESENVOLVIDO	19
5.1. O que ele atende	19
5.2. Melhorias e facilidades implementadas	19
5.3. Implementações futuras	20
6. CUSTOS E PRAZOS	21
6.1. Custos Estimados	21
6.2. Prazos Estimados	25
7. CANAIS DE COMUNICAÇÃO	26
7.1. Comunicação Interna da Equipe	27
7.2. Comunicação com o Cliente (Product Owner)	28
8. CONCLUSÃO	29

1. INTRODUÇÃO

1.1. Contexto da Problemática

A crescente integração dos animais de estimação na vida familiar moderna tem intensificado a preocupação com seu bem-estar e saúde. Contudo, muitos donos de pets enfrentam desafios significativos na gestão eficiente dos cuidados de seus animais. É comum que informações cruciais, como datas de vacinação, agendamentos de consultas veterinárias, dosagens e periodicidade de medicações, e outros cuidados essenciais, sejam esquecidas ou registradas de forma desorganizada. Essa falta de controle pode acarretar sérias consequências, desde o comprometimento da saúde do animal devido à interrupção de tratamentos ou à perda de prazos de vacinação, até gastos financeiros desnecessários com tratamentos emergenciais que poderiam ter sido prevenidos.

A ausência de um sistema centralizado e proativo para o acompanhamento desses cuidados não só gera estresse para os donos, mas também limita a capacidade dos pet shops de oferecer um serviço mais integrado e personalizado aos seus clientes.

A análise do cenário revela uma importante oportunidade de mercado e de engajamento do cliente para o pet shop. Embora a problemática inicial seja a dificuldade dos tutores em gerenciar os cuidados de seus pets, a oferta de um aplicativo que solucione essa questão para os clientes do pet shop transforma a ferramenta em um ativo estratégico. Ao fornecer uma solução valiosa que facilita a rotina dos donos, o aplicativo se torna um ponto de contato constante e uma fonte de valor percebido. Este valor percebido não apenas resolve um problema imediato para o usuário final, mas também fomenta um aumento no engajamento e na lealdade dos clientes em relação ao pet shop. Além disso, a plataforma pode coletar dados vitais, como o histórico de vacinas, a frequência de consultas e as necessidades de medicação, que podem ser utilizados pelo pet shop para aprimorar suas operações, oferecer serviços personalizados e desenvolver campanhas de marketing direcionadas. Desse modo, o aplicativo transcende a função de uma simples utilidade, evoluindo para uma ferramenta de gestão de relacionamento com o cliente (CRM) é uma fonte de inteligência de negócios, proporcionando uma vantagem competitiva e fortalecendo o vínculo com a clientela a longo prazo.

1.2. Objetivo do Cliente

O cliente, um pet shop, busca o desenvolvimento de um aplicativo para

dispositivos móveis que será disponibilizado aos seus próprios clientes. Seu propósito fundamental é permitir que os usuários registrem seus dados pessoais e, principalmente, as informações de seus pets, facilitando a gestão dos cuidados e a rotina do pet shop. Este objetivo visa não apenas otimizar a experiência do cliente final, proporcionando-lhes uma ferramenta eficaz para o controle da saúde de seus animais, mas também aprimorar a eficiência operacional do pet shop. Ao centralizar as informações dos pets e de seus cuidados, o aplicativo contribuirá para um atendimento mais ágil e personalizado, reduzindo a burocracia e fortalecendo o relacionamento entre o pet shop e seus clientes.

1.3. Justificativa do Projeto

A implementação deste sistema de gerenciamento de cuidados para pets é justificada por seus múltiplos benefícios, tanto para os usuários finais quanto para o pet shop, e pelo impacto positivo na saúde e bem-estar animal.

Para o usuário final, o software proporcionará um controle centralizado e altamente organizado das informações de saúde e cuidados de seus pets. A capacidade de registrar vacinas aplicadas, datas de consultas, medicações em uso e outras necessidades do animal em um único local reduzirá significativamente o estresse e a probabilidade de esquecimentos. A inclusão de alertas e lembretes proativos, com antecedência mínima de 24 horas para vacinas, consultas e medicações, assegurará que os cuidados essenciais sejam realizados dentro do prazo adequado, prevenindo problemas de saúde e evitando gastos desnecessários com tratamentos emergenciais.

Para o pet shop, o aplicativo representa uma ferramenta estratégica para a fidelização de clientes. Ao oferecer um serviço de valor agregado. O sistema otimiza o atendimento, ao permitir que a equipe do pet shop acesse informações relevantes sobre os pets dos clientes de forma rápida e eficiente. Isso pode reduzir a carga administrativa associada ao histórico dos pets e fornecer dados valiosos para a tomada de decisões estratégicas, como a identificação de padrões de saúde, a demanda por determinados produtos ou serviços, e a personalização de ofertas.

Em um nível mais amplo, o impacto social e na saúde animal é notável. A disponibilização de uma ferramenta que facilita o acompanhamento rigoroso dos cuidados contribui diretamente para a melhoria da saúde e bem-estar dos animais de estimação. Isso promove a guarda responsável, incentivando os tutores a manterem seus pets com a vacinação em dia, a realizarem consultas preventivas e a administrar medicações conforme a prescrição, auxiliando na prevenção de doenças e na

promoção de uma vida mais saudável para os animais.

2. PROPOSTA DE SOLUÇÃO

2.1. Descrição Detalhada da Proposta

A proposta consiste no desenvolvimento de um sistema web responsivo, seguro e intuitivo, acessível por computador e dispositivos móveis. O sistema permitirá o cadastro de usuários e pets, o registro de vacinas, consultas e medicações, além de gerar alertas e lembretes automáticos com antecedência mínima de 24 horas.

abrangente para cadastro e monitoramento de pets, Este sistema permite aos donos registrar e consultar de forma organizada todas as informações críticas relacionadas à saúde e ao bem-estar de seus companheiros. As funcionalidades centrais incluem o registro de vacinas aplicadas, com detalhes como nome da vacina, data de aplicação e periodicidade; o acompanhamento de consultas veterinárias, com campos para data, motivo e observações relevantes; e o controle de medicações em uso, especificando nome do medicamento, dosagem, frequência e período de uso.

Um dos pilares do sistema é a funcionalidade de alertas e lembretes personalizáveis. O sistema será capaz de emitir notificações proativas, seja por meio de alertas no próprio aplicativo ou via e-mail, com uma antecedência mínima de 24 horas, garantindo que os tutores sejam lembrados de vacinas, consultas agendadas e a administração de medicações. Esse recurso é fundamental para evitar esquecimentos e assegurar a continuidade dos cuidados essenciais.

O fluxo básico de funcionamento do aplicativo inicia-se com o cadastro do usuário, onde serão solicitados dados pessoais como nome completo, e-mail, número de telefone e CPF, com validação de campos obrigatórios e verificação de e-mail duplicado para garantir a unicidade do registro. Após o cadastro e a criação de uma senha de acesso, o usuário poderá fazer login no sistema. Uma vez autenticado, o usuário terá a capacidade de criar múltiplos perfis para seus pets, informando detalhes como nome, espécie, raça, sexo, idade e data de nascimento, com validação para impedir datas futuras. A partir daí, o registro do histórico de vacinas, consultas e medicação, será facilitado por interfaces dedicadas. O sistema processará essas informações para gerar os alertas automáticos mencionados. Adicionalmente, o aplicativo permitirá a edição e exclusão de informações previamente cadastradas, tanto de pets quanto de seus registros de cuidados, conforme a necessidade do

usuário.

2.2. Estrutura e Arquitetura do Software

A arquitetura do software será cuidadosamente projetada para garantir robustez, escalabilidade, segurança e fácil manutenção a longo prazo. Será adotada uma abordagem de arquitetura em camadas, com uma clara separação de responsabilidades entre os componentes.

- **Aplicativo Móvel (Frontend):** Para o aplicativo móvel, utilizaremos **React Native**. Isso nos permitirá desenvolver uma interface amigável que funcione bem tanto em celulares Android quanto iOS, otimizando o tempo de desenvolvimento. A interface será responsável pela apresentação visual e interação com o usuário.
- **Servidor (Backend):** A lógica do aplicativo, o processamento das informações e a conexão com o banco de dados ficarão no backend. Ele será desenvolvido com **Node.js** e o **framework Express.js**. Essa escolha nos oferece uma base sólida, segura e que pode crescer conforme a necessidade.
- **Banco de Dados:** As informações dos usuários, pets, vacinas, consultas e medicamentos serão armazenadas no banco de dados **PostgreSQL**. Ele foi escolhido por garantir a integridade e consistência desses dados, que são muito importantes para o funcionamento do sistema.
- **Hospedagem em Nuvem:** O sistema completo (backend e banco de dados) será hospedado em serviços de nuvem (por exemplo, AWS, Azure ou GCP). Isso garante que o aplicativo esteja disponível para os usuários a maior parte do tempo (alta disponibilidade, ex: 99,5%), que os dados estejam seguros através de backups automáticos, e que possamos aumentar a capacidade do sistema se o número de usuários crescer.

A escolha dessas tecnologias considera também a facilidade de manutenção e futuras atualizações do aplicativo, aproveitando comunidades de desenvolvedores ativas e documentação acessível.

2.3. Metodologia de Desenvolvimento

Para guiar o processo de desenvolvimento do software, será adotada uma metodologia ágil, especificamente o framework **Scrum**. Esta escolha é fundamentada na capacidade do Scrum de entregar valor de forma incremental e adaptativa, características essenciais para projetos de software em ambientes dinâmicos.

O Scrum é um framework iterativo e incremental que organiza o trabalho em

ciclos curtos e de tempo fixo, conhecidos como Sprints, geralmente com duração de duas semanas. Esta abordagem permite um feedback contínuo e uma flexibilidade inerente para adaptar-se a requisitos que podem evoluir ao longo do tempo.¹² É uma metodologia particularmente adequada para o desenvolvimento de aplicativos móveis, dada a natureza volátil dos requisitos em projetos de software e a necessidade de prototipagem e validação rápidas com os usuários. Diferentemente de abordagens sequenciais tradicionais, o Scrum promove a auto-organização da equipe e a comunicação frequente entre todos os membros.¹²

Os princípios do Scrum serão aplicados da seguinte forma:

- **Sprints:** O desenvolvimento será dividido em iterações de duas semanas, ao final das quais um incremento do produto, potencialmente utilizável e com valor agregado, será entregue.
- **Daily Scrums:** Reuniões diárias curtas, com duração máxima de 15 minutos, serão realizadas para que o Time de Desenvolvimento sincronize suas atividades, discuta o progresso em relação ao objetivo da sprint e identifique quaisquer impedimentos que possam surgir.
- **Feedback Contínuo:** A cada final de sprint, serão realizadas as Revisões de Sprint, onde o incremento do produto será demonstrado aos stakeholders (incluindo o cliente) para coleta de feedback e validação. As retrospectivas de Sprint, por sua vez, permitirão que a equipe reflita sobre o processo de desenvolvimento e identifique oportunidades de melhoria contínua.

A aplicação do Scrum em projetos menores, como o aplicativo para pet shop, demandará uma adaptação pragmática, focando nos princípios essenciais de transparência, inspeção e adaptação para evitar sobrecarga. A otimização pode envolver a simplificação de cerimônias, mantendo valor e colaboração, e equilibrando a flexibilidade ágil com documentação ABNT e expectativas claras do cliente. O engajamento do cliente como Product Owner é crucial para o alinhamento do produto às suas necessidades.

2.4. Equipe do Projeto e Atribuições

A equipe de desenvolvimento será composta por profissionais multifuncionais, organizados de acordo com os papéis definidos pelo framework Scrum. Esta estrutura visa promover a colaboração intensa, a auto-organização e a responsabilidade coletiva pela entrega de um produto de alta qualidade.

Papéis e Responsabilidades:

- **Product Owner (PO):**

- **Responsabilidades:** O Product Owner é o principal responsável por maximizar o valor do produto resultante do trabalho do Time de Desenvolvimento. Suas atribuições incluem a gestão do Product Backlog, garantindo que os itens sejam claramente definidos, priorizados e visíveis para toda a equipe. Atua como o elo central de comunicação entre o Time Scrum e os stakeholders, representando a voz do cliente e suas necessidades.
- **Custo/Hora Estimado:** Com base no salário médio anual de R\$ 84.299 para Product Owners no Brasil, e considerando uma jornada de 22 dias úteis por mês e 8 horas por dia (totalizando 176 horas mensais), o custo horário estimado para este papel é de aproximadamente **R\$ 47,89/hora**.

- **Scrum Master (SM):**

- **Responsabilidades:** O Scrum Master é o facilitador do processo Scrum, responsável por educar e treinar a equipe e os stakeholders sobre a teoria e as práticas do Scrum. Sua função primordial é remover impedimentos que possam atrapalhar o progresso do Time de Desenvolvimento e assegurar que os eventos Scrum ocorram de forma produtiva e dentro das diretrizes. O Scrum Master não possui responsabilidades de gerenciamento de pessoas tradicionais, focando na otimização do processo.
- **Custo/Hora Estimado:** Considerando o salário médio anual de R\$ 120.000 para Scrum Masters no Brasil, o custo horário estimado é de aproximadamente **R\$ 68,18/hora**.

- **Time de Desenvolvimento (Developers):**

- **Responsabilidades:** O Time de Desenvolvimento é composto pelos profissionais que realizam o trabalho de criar um incremento "Done" (pronto para uso) a cada Sprint. São auto-organizados e multifuncionais, possuindo todas as habilidades necessárias para desenvolver o produto de ponta a ponta. As especialidades que compõem este time são:
 - **Desenvolvedor Frontend (Mobile):** Responsável pela implementação da interface do usuário do aplicativo, garantindo a responsividade e a melhor experiência visual e interativa para o usuário em diferentes dispositivos.
 - **Custo/Hora Estimado:** O salário médio para um programador frontend no Brasil é de R\$ 32,32 por hora.
 - **Desenvolvedor Backend:** Encarregado da construção da lógica de negócio, das APIs (Application Programming Interfaces) e da integração com o banco de dados, assegurando a segurança, performance e escalabilidade do sistema.
 - **Custo/Hora Estimado:** O salário médio para um desenvolvedor

backend no Brasil é de R\$ 32,32 por hora.

- **Designer UI/UX:** Responsável pela pesquisa de usuário, criação de wireframes, protótipos e o design visual da interface, com foco primordial na usabilidade e na experiência do usuário.
 - **Custo/Hora Estimado:** Para um profissional de nível médio no Brasil, o custo horário varia entre **R\$ 29 e R\$48.30** Será utilizada uma média de **R\$ 38,50/hora** para as estimativas de custo.
- **QA Tester:** Responsável por desenvolver e executar planos de teste, casos de teste e scripts para avaliar a qualidade e funcionalidade do software, identificando e documentando defeitos para garantir que o produto final atenda aos mais altos padrões de qualidade.
 - **Custo/Hora Estimado:** O salário médio para QA Testers no Brasil varia entre **R\$ 14,60 e R\$ 19,75** por hora.
- **Analista de Requisitos:** Colabora ativamente na elicitacão, análise, especificacão e validacão dos requisitos do sistema, atuando como uma ponte essencial entre o cliente e a equipe técnica, garantindo que as necessidades de negócio sejam traduzidas em requisitos claros e compreensíveis.
 - **Custo/Hora Estimado:** Não há um valor explícito nos dados fornecidos para o custo de um Analista de Requisitos. No entanto, dada a natureza especializada e a importância da função para a clareza do projeto, estima-se um custo horário de **R\$40,00/hora**, alinhado com a média de profissionais técnicos experientes.

A otimização da equipe para o orçamento e o escopo do projeto é uma consideracão crucial. Embora os dados salariais permitam uma estimativa detalhada dos custos por função, a alocação de uma pessoa dedicada em tempo integral para cada especialidade pode resultar em custos excessivos ou subutilizacão de recursos, especialmente em um projeto de pequena escala a média. Em equipes ágeis, é comum que as funções sejam combinadas ou compartilhadas; por exemplo, o Product Owner pode assumir algumas responsabilidades de análise de requisitos detalhada, ou um desenvolvedor pode atuar tanto no frontend quanto no backend para funcionalidades mais simples. As taxas horárias fornecem a flexibilidade necessária para alocar esforços parciais, como um Designer UI/UX que pode ser mais intensamente envolvido durante os sprints de design e menos durante os sprints de desenvolvimento puro. Portanto, a estrutura da equipe e a alocação de horas devem ser cuidadosamente otimizadas para se alinharem ao orçamento do projeto e ao

princípio de operações enxutas da metodologia ágil.

Tabela 1 – Estimativa de Custos da Equipe por Função (Valores Horários)

Função	Custo/Hora (BRL)
Product Owner	R\$ 47,89
Scrum Master	R\$ 68,18
Desenvolvedor Frontend	R\$ 32,32
Desenvolvedor Backend	R\$ 32,32
Designer UI/UX	R\$ 38,50
QA Tester	R\$ 85,88
Analista de Requisitos	R\$ 40,00

Nota: Os custos horários são baseados em salários médios anuais no Brasil, convertidos para base horária considerando 22 dias úteis e 8 horas de trabalho por dia. A estimativa de horas por fase será detalhada na seção de Custos e Prazos.

3. REQUISITOS DO SISTEMA

A definição clara dos requisitos é uma etapa fundamental no desenvolvimento de software, pois estabelece o escopo e as funcionalidades esperadas do sistema. Os requisitos são categorizados em funcionais e não funcionais.

3.1. Requisitos Funcionais (RF)

Requisitos funcionais (RF) descrevem as funções e serviços específicos que o sistema deve oferecer aos usuários, detalhando o "o quê" o sistema fará para atender às necessidades do negócio e do usuário. Eles são as ações que o sistema deve ser capaz de executar.

A lista detalhada de requisitos funcionais para o sistema de gerenciamento de cuidados para pets é apresentada a seguir:

- **RF<001.01> Informar dados pessoais:**

Objetivo: Permitir que o usuário forneça suas informações de identificação.

Funcionalidade: O sistema deve permitir que o usuário informe nome completo, e-mail, número de telefone e CPF.

- **RF<001.02> Criar senha de acesso:**

Objetivo: Assegurar a autenticação segura do usuário.

Funcionalidade: O sistema deve permitir que o usuário crie uma senha para autenticação no sistema.

- **RF<001.03> Validar dados obrigatórios (usuário):**

Objetivo: Garantir a completude das informações de cadastro do usuário.

Funcionalidade: O sistema deve validar se todos os campos obrigatórios estão preenchidos antes de permitir o cadastro.

- **RF<001.04> Validar e-mail duplicado:**

Objetivo: Prevenir cadastros de usuários com e-mails já em uso.

Funcionalidade: O sistema deve verificar se o e-mail informado está duplicado/em uso, não permitindo o cadastro duplicado.

- **RF Login e Autenticação:**

Objetivo: Permitir o acesso seguro dos usuários ao sistema.

Funcionalidade: O sistema deve permitir que usuários façam login utilizando

e-mail e senha previamente cadastrados.

- **RF<003.01> Informar dados do pet:**

Objetivo: Permitir o registro das informações básicas de um animal de estimação.

Funcionalidade: O sistema deve permitir informar nome, espécie, raça, sexo, idade e data de nascimento do pet.

- **RF<003.02> Validar data de nascimento (pet):**

Objetivo: Garantir a consistência da data de nascimento do pet.

Funcionalidade: O sistema deve verificar se a data de nascimento não é posterior à data atual.

- **RF<003.03> Cadastro múltiplo de pets:**

Objetivo: Permitir que um único usuário gerencie múltiplos animais.

Funcionalidade: O sistema deve permitir que um usuário cadastre mais de um pet.

- **RF<003.04> Validar campos obrigatórios (pet):**

Objetivo: Assegurar a completude das informações de cadastro do pet.

Funcionalidade: O sistema deve impedir o cadastro se campos obrigatórios estiverem vazios.

- **RF Edição de dados:**

Objetivo: Permitir a atualização de informações registradas.

Funcionalidade: O sistema deve permitir que o usuário edite informações previamente cadastradas dos pets e seus registros.

- **RF Exclusão de Registros:**

Objetivo: Permitir a remoção de informações desnecessárias ou incorretas.

Funcionalidade: O sistema deve permitir que o usuário exclua pets ou registros de cuidados que não são mais necessários.

- **RF Registro de Vacinas:**

Objetivo: Manter um histórico detalhado das vacinações dos pets.

Funcionalidade: O sistema deve permitir o cadastro de vacinas aplicadas aos pets, com nome, data de aplicação e periodicidade.

- **RF Registro de Consultas:**

Objetivo: Registrar e consultar o histórico de atendimentos veterinários.

Funcionalidade: O sistema deve permitir que o usuário registre consultas veterinárias, informando data, motivo e observações.

- **RF Registro de Medicações:**

Objetivo: Manter um controle sobre os medicamentos em uso pelos pets.

Funcionalidade: O sistema deve permitir o registro de medicamentos, com nome, dosagem, frequência e período de uso.

- **RF Alertas e Lembretes:**

Objetivo: Notificar o usuário sobre eventos importantes relacionados aos cuidados dos pets.

Funcionalidade: O sistema deve emitir alertas e lembretes para vacinas, consultas e medicações com antecedência mínima de 24 horas.

3.2. Requisitos Não Funcionais (RNF)

Requisitos não funcionais (RNF) especificam os critérios de qualidade do sistema, ou seja, definem "como" o sistema deve se comportar, em vez de "o quê" ele deve fazer. Eles são cruciais para a experiência do usuário, a segurança e a sustentabilidade do software.

A lista detalhada de requisitos não funcionais para o sistema de gerenciamento de cuidados para pets é a seguinte:

- **RNF Interface Responsiva:**

Objetivo: Garantir a adaptabilidade da interface do usuário a diferentes dispositivos.

Funcionalidade: A interface do sistema deve ser responsiva aos diversos tamanhos de telas de computadores, tablets e celulares.

- **RNF Segurança de Dados:**

Objetivo: Proteger a confidencialidade e integridade das informações dos usuários e pets.

Funcionalidade: O sistema deve armazenar os dados dos usuários e pets de

forma segura, utilizando criptografia.

- **RNF Backup de Dados:**

Objetivo: Assegurar a recuperação de dados em caso de falhas ou perdas.

Funcionalidade: O sistema deve realizar backup automático das informações periodicamente.

- **RNF Desempenho:**

Objetivo: Garantir uma experiência de usuário fluida e eficiente.

Funcionalidade: O sistema deve responder às requisições dos usuários em no máximo 2 segundos.

- **RNF Usabilidade:**

Objetivo: Facilitar a interação do usuário com o sistema.

Funcionalidade: O sistema deve apresentar interface amigável e intuitiva, acessível a usuários sem conhecimentos técnicos.

- **RNF Escalabilidade:**

Objetivo: Permitir que o sistema suporte um aumento no volume de usuários e dados.

Funcionalidade: O sistema deve ser capaz de escalar horizontalmente para suportar aumento de usuários.

- **RNF Compatibilidade:**

Objetivo: Assegurar o funcionamento correto em diferentes ambientes de execução.

Funcionalidade: O sistema deve funcionar corretamente nos principais navegadores modernos (Chrome, Firefox, Edge, Safari).

- **RNF Disponibilidade:**

Objetivo: Garantir que o sistema esteja acessível para os usuários a maior parte do tempo.

Funcionalidade: O sistema deve estar disponível 99,5% do tempo, exceto durante janelas de manutenção planejadas.

4. MODELAGEM DE SOFTWARE

A modelagem de software é uma etapa crucial para visualizar e entender o sistema antes de sua implementação. Utiliza-se a Linguagem de Modelagem Unificada (UML) para representar diferentes perspectivas do sistema.

4.1. Caso de Uso (UML)

Os diagramas de caso de uso são ferramentas da UML que descrevem a funcionalidade do sistema do ponto de vista do usuário (ator). Eles ilustram como os usuários interagem com o sistema para realizar tarefas específicas. Para o sistema de gerenciamento de cuidados para pets, os principais casos de uso e seus atores são:

- **Ator Principal:** Tutor (Dono do Pet)
- **Outros atores:** Sistema

Objetivo: Permitir que o tutor registre, consulte e organize todas as informações cruciais sobre a saúde e o bem-estar de seus pets, garantindo que nenhum cuidado essencial seja esquecido.

Pré-condições:

- O Tutor já está cadastrado no sistema.
- O Tutor fez login no sistema.
- O pet para o qual os cuidados serão gerenciados já está cadastrado no sistema.

Pós-condições:

- As informações de vacinas, consultas ou medicações são registradas, atualizadas ou excluídas no sistema.
- Lembretes automáticos são configurados/atualizados conforme os registros.

Fluxo Principal:

1. O Tutor acessa o painel principal do aplicativo após o login.
2. O Tutor seleciona o pet desejado na lista de pets cadastrados.

3. O Sistema exibe as opções de gerenciamento de cuidados para o pet selecionado (Registrar Vacina, Registrar Consulta, Registrar Medicação, Visualizar Histórico, Editar Informações, Excluir Registro).
4. O Tutor pode escolher uma das seguintes ações:
 - Ação A: Registrar Vacina
 1. O Tutor seleciona a opção "Registrar Vacina".
 2. O Sistema apresenta um formulário para o registro da vacina.
 3. O Tutor informa o nome da vacina, a data de aplicação e a periodicidade.
 4. O Sistema valida os dados informados.
 5. O Sistema armazena o registro da vacina.
 6. O Sistema configura um lembrete automático para a próxima dose (com antecedência mínima de 24 horas).
 7. O Sistema exibe uma mensagem de sucesso.
 - Ação B: Registrar Consulta Veterinária
 1. O Tutor seleciona a opção "Registrar Consulta Veterinária".
 2. O Sistema apresenta um formulário para o registro da consulta.
 3. O Tutor informa a data da consulta, o motivo e observações adicionais.
 4. O Sistema valida os dados informados.
 5. O Sistema armazena o registro da consulta.
 6. O Sistema configura um lembrete automático para a consulta (com antecedência mínima de 24 horas).
 7. O Sistema exibe uma mensagem de sucesso.
 - Ação C: Registrar Medicação
 1. O Tutor seleciona a opção "Registrar Medicação".
 2. O Sistema apresenta um formulário para o registro da medicação.
 3. O Tutor informa o nome do medicamento, dosagem, frequência e período de uso.
 4. O Sistema valida os dados informados.
 5. O Sistema armazena o registro da medicação.

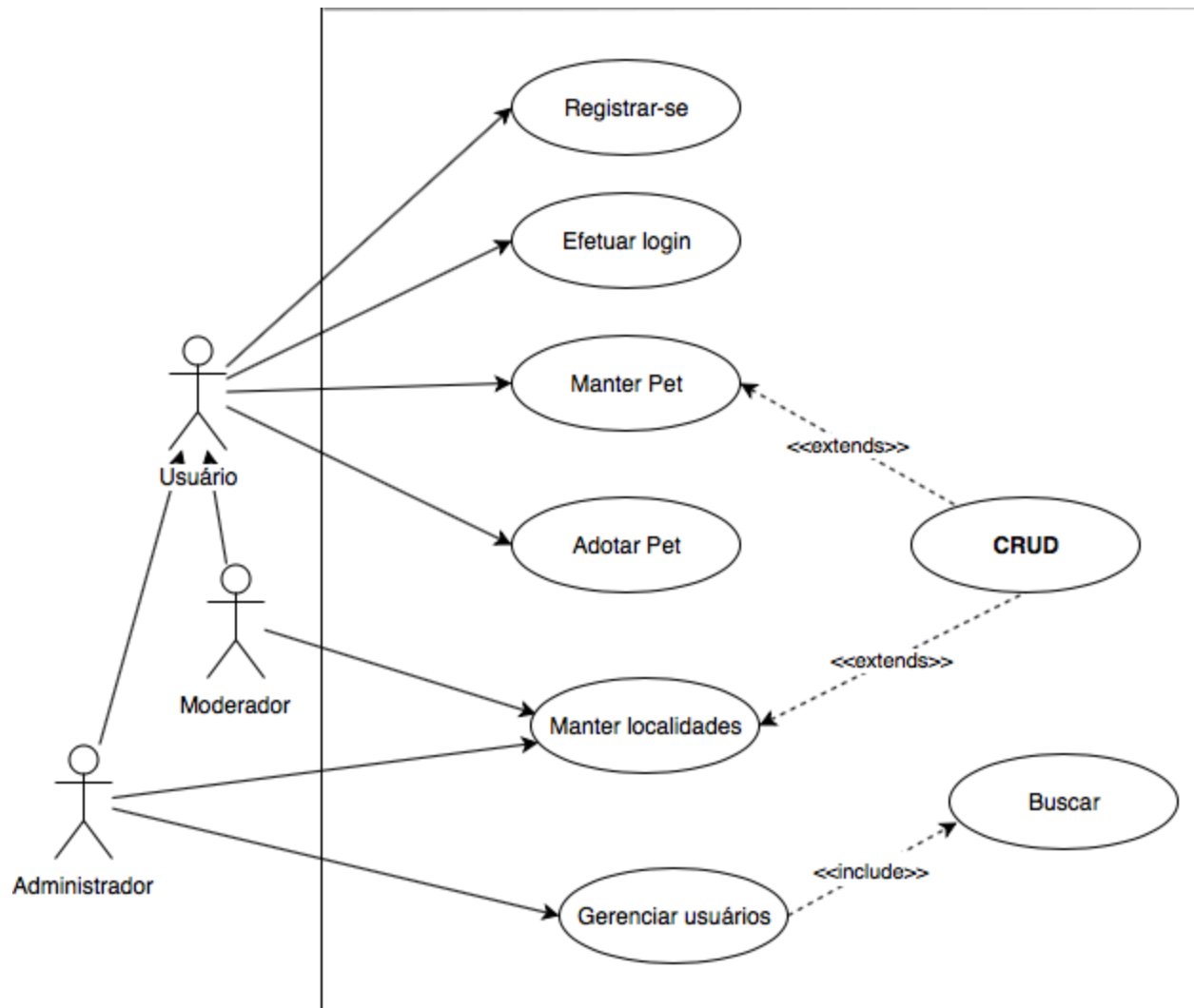
6. O Sistema configura lembretes automáticos para a administração da medicação (com antecedência mínima de 24 horas).
7. O Sistema exibe uma mensagem de sucesso.
- Ação D: Visualizar Histórico
 1. O Tutor seleciona a opção "Visualizar Histórico".
 2. O Sistema exibe o histórico completo de vacinas, consultas e medicações do pet selecionado em ordem cronológica.
- Ação E: Editar Informações de Cuidado
 1. O Tutor seleciona a opção "Editar Informações".
 2. O Sistema apresenta uma lista dos registros de cuidados (vacinas, consultas, medicações).
 3. O Tutor seleciona o registro que deseja editar.
 4. O Sistema carrega o formulário com as informações existentes do registro.
 5. O Tutor modifica as informações necessárias e as submete.
 6. O Sistema valida as novas informações e as atualiza no banco de dados.
 7. O Sistema atualiza os lembretes automáticos, se aplicável.
 8. O Sistema exibe uma mensagem de sucesso.
- Ação F: Excluir Registro de Cuidado
 1. O Tutor seleciona a opção "Excluir Registro".
 2. O Sistema apresenta uma lista dos registros de cuidados (vacinas, consultas, medicações).
 3. O Tutor seleciona o registro que deseja excluir.
 4. O Sistema solicita uma confirmação de exclusão.
 5. O Tutor confirma a exclusão.
 6. O Sistema remove o registro do banco de dados.
 7. O Sistema remove quaisquer lembretes associados a esse registro.
 8. O Sistema exibe uma mensagem de sucesso.

Fluxos Alternativos:

- **Validação de Dados:**

- Se, em qualquer etapa de registro/edição, os dados informados pelo Tutor forem inválidos (e.g., campos obrigatórios vazios, data futura), o Sistema exibirá uma mensagem de erro indicando o problema e o Tutor deverá corrigi-lo.
- **Cancelamento de Operação:**
 - Em qualquer formulário de registro ou edição, o Tutor pode optar por cancelar a operação, retornando à tela anterior sem salvar as alterações.

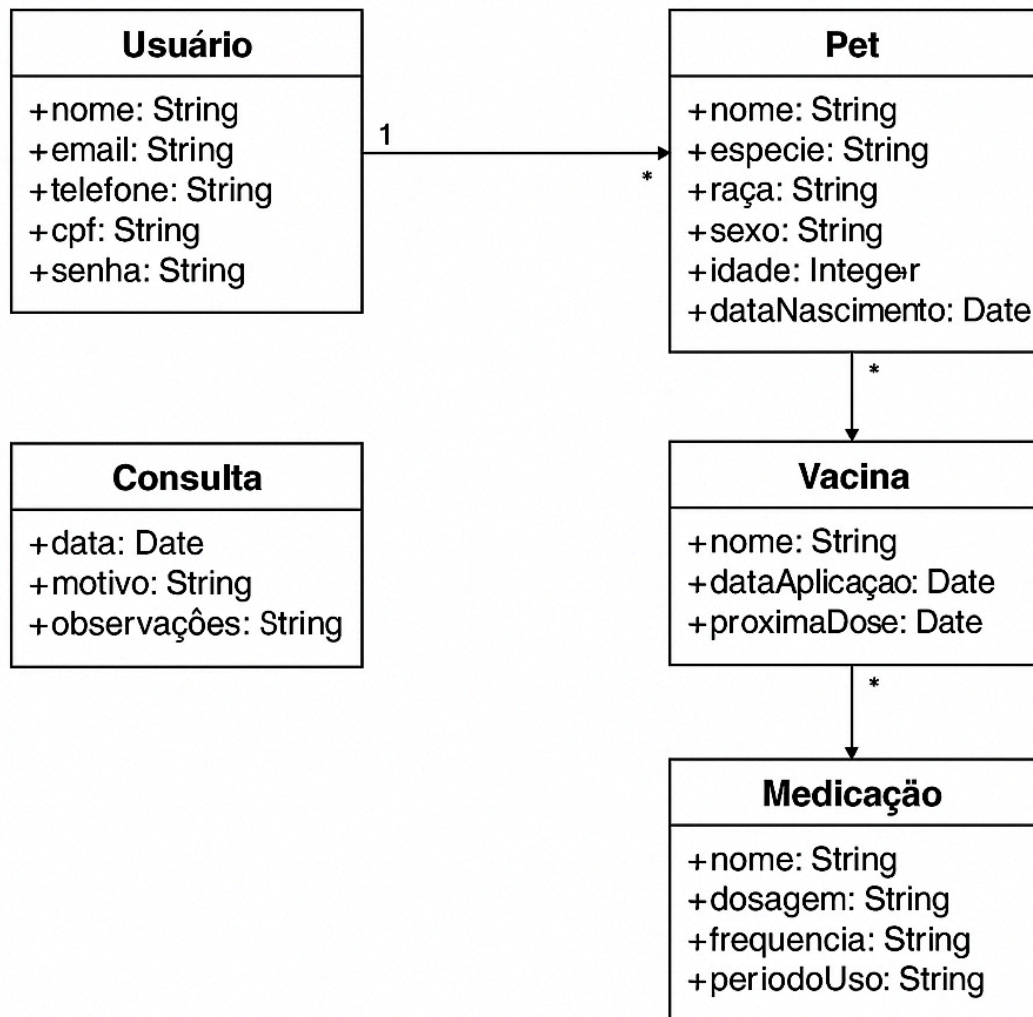
Representação do Caso de Uso utilizando [Draw.io](https://draw.io):



4.2. Diagrama de Classes (UML)

Os diagramas de classes são fundamentais para a modelagem orientada a objetos, representando a estrutura estática do sistema em termos de classes, seus atributos, operações e os relacionamentos entre elas. Para o sistema de gerenciamento de cuidados para pets, as principais classes e seus relacionamentos

seriam:



Essas classes representam as entidades centrais do sistema e suas interconexões, fornecendo uma base para o design do banco de dados e a implementação da lógica de negócio.

5. OBJETIVO DO SOFTWARE DESENVOLVIDO

O software de gerenciamento de cuidados para pets é desenvolvido com o objetivo primordial de simplificar e otimizar a rotina de cuidados com animais de estimação, tanto para os tutores quanto para o pet shop.

5.1. O que ele atende

O sistema centraliza o histórico de cuidados dos pets (vacinas, consultas, medicações), reduzindo o estresse dos donos e assegurando cuidados preventivos no tempo certo. Para o pet shop, melhora a experiência do cliente, otimiza a rotina operacional e fortalece o relacionamento com a clientela, servindo como um diferencial competitivo.

5.2. Melhorias e facilidades implementadas

As melhorias e facilidades implementadas pelo software são diversas e impactam positivamente a vida dos tutores e a gestão do pet shop:

- **Controle Centralizado e Organizado:** Todas as informações de saúde e cuidados do pet (vacinas, consultas, medicações) são armazenadas em um único local de fácil acesso, eliminando a necessidade de registros físicos desorganizados.
- **Alertas e Lembretes Proativos:** O sistema emite notificações automáticas para vacinas, consultas e medicações com antecedência mínima de 24 horas, assegurando que nenhum cuidado importante seja esquecido.
- **Suporte a Múltiplos Pets:** Um único usuário pode cadastrar e gerenciar informações para vários pets, tornando o aplicativo versátil para famílias com mais de um animal.
- **Edição e Exclusão Flexíveis:** A capacidade de editar e excluir registros permite que os usuários mantenham as informações sempre atualizadas e precisas.
- **Interface Amigável e Intuitiva:** O design do aplicativo prioriza a usabilidade, tornando-o acessível mesmo para usuários sem conhecimentos técnicos.
- **Fidelização de Clientes para o Pet Shop:** Ao oferecer um serviço de valor agregado, o pet shop fortalece o relacionamento com seus clientes, incentivando a recorrência e a lealdade.
- **Otimização da Rotina do Pet Shop:** Com o acesso facilitado às informações dos pets dos clientes, o pet shop pode oferecer um atendimento mais personalizado e eficiente, reduzindo o tempo gasto na busca por históricos e otimizando o

agendamento de serviços.

5.3. Implementações futuras

O sistema foi concebido com uma arquitetura escalável, permitindo futuras implementações que expandirão sua funcionalidade e valor. Algumas das melhorias e funcionalidades futuras que podem ser consideradas incluem:

- **Integração com Serviços do Pet Shop:**
 - **Agendamento Online:** Permitir que os usuários agendem banho e tosa, consultas veterinárias e outros serviços diretamente pelo aplicativo.
 - **Histórico de Serviços do Pet Shop:** Incluir um histórico de serviços realizados no pet shop, como banhos, tosas, hospedagens, etc.
 - **Loja Virtual Integrada:** Possibilitar a compra de produtos (rações, brinquedos, acessórios) diretamente pelo aplicativo, com opções de entrega ou retirada no pet shop.
- **Funcionalidades de Telemedicina Veterinária:**
 - **Consultas Online:** Implementar a possibilidade de consultas veterinárias por vídeo, para casos de menor complexidade ou acompanhamento.
 - **Chat com Veterinário:** Um canal de comunicação direto com veterinários para dúvidas rápidas.
- **Recursos Sociais e Comunitários:**
 - **Perfis Compartilháveis:** Opção de compartilhar o perfil do pet com outros membros da família ou cuidadores.
 - **Comunidade de Tutores:** Fóruns ou grupos para troca de experiências e dicas entre donos de pets.
- **Monitoramento Avançado e Inteligência Artificial:**
 - **Monitoramento de Peso e Dieta:** Ferramentas para acompanhar o peso do pet e gerenciar sua dieta.
 - **Análise de Tendências de Saúde:** Utilização de IA para identificar padrões de saúde e alertar sobre possíveis problemas com base nos dados registrados.
 - **Recomendações Personalizadas:** Sugestões de produtos, serviços ou cuidados com base nas características e histórico do pet.

Essas implementações futuras visam manter o aplicativo relevante e valioso para os usuários, expandindo as oportunidades de negócio para o pet shop e consolidando sua posição no mercado de cuidados animais.

6. CUSTOS E PRAZOS

A fase de planejamento de custos e prazos é crucial para a viabilidade e gestão do projeto, especialmente considerando o cenário econômico atual no Brasil. As estimativas a seguir são baseadas nos salários médios por hora já definidos na seção 2.4 e na experiência comum em projetos de desenvolvimento ágil.

6.1. Custos Estimados

A estimativa de custos foi segmentada por fase do projeto e por papel na equipe, considerando a intensidade de envolvimento de cada profissional em diferentes etapas. É importante notar que, em uma equipe Scrum otimizada para orçamentos, um mesmo profissional pode ter responsabilidades em diferentes funções, especialmente em projetos de pequena a média escala.

Fase do Projeto	Duração Estimada (Sprints)	Duração (Semanas)
1. Planejamento	1	2
2. Design e UI/UX	2	4
3. Desenvolvimento	6	12
4. Teste de controle de qualidade	2	4
5.Implantação e Go-Live	0,5	1
Total:	11,5	23

Custo Total Estimado por Fase e Papel:

Considerando 40 horas de trabalho por semana, os custos por fase são:

Função	Custo/Hora (BRL)	Salário Mensal Estimado (BRL)	Horas/Sprint	Custo/Sprint (BRL)
Proprietário do produto	R\$ 47,89	R\$ 8.429,90	20	R\$ 957,80
Mestre Scrum	R\$ 68,18	R\$ 12.000,00	15	R\$ 1.022,70
Desenvolvedor Frontend	R\$ 32,32	R\$ 5.700,00	40	R\$ 1.292,80
Desenvolvedor Backend	R\$ 32,32	R\$ 5.700,00	40	R\$ 1.292,80
Designer de UI/UX	R\$ 38,50	R\$ 6.776,00	25	R\$ 962,50
Testador de QA	R\$ 19,75	R\$ 3.476,00	30	R\$ 592,50
Analista de Requisitos	R\$ 40,00	R\$ 7.040,00	10	R\$ 400,00

Nota: Foi utilizado o valor máximo da faixa de R\$ 14,60 e R\$ 19,75, conforme a documentação, para uma estimativa mais conservadora, arredondado para R\$ 19,75. Os salários mensais estimados são baseados nos custos por hora e uma jornada de 176 horas mensais (22 dias úteis x 8 horas/dia). As horas por sprint são uma média

que pode variar dependendo da intensidade da fase.

Estrutura de Custos por Fase e Função:

1. Planejamento (1 Sprint)

- Product Owner: R\$ 957,80
- Scrum Master: R\$ 1.022,70
- Desenvolvedor Frontend: R\$ 646,40
- Desenvolvedor Backend: R\$ 646,40
- Designer UI/UX: R\$ 481,25
- QA Tester: R\$ 296,25
- Analista de Requisitos: R\$ 400,00
- **Total da Fase:** R\$ 4.450,80

2. Design e UI/UX (2 Sprints)

- Product Owner: R\$ 1.915,60
- Scrum Master: R\$ 2.045,40
- Desenvolvedor Frontend: R\$ 1.292,80
- Desenvolvedor Backend: R\$ 1.292,80
- Designer UI/UX: R\$ 1.925,00
- QA Tester: R\$ 1.185,00
- Analista de Requisitos: R\$ 800,00
- **Total da Fase:** R\$ 10.456,60

3. Desenvolvimento (6 Sprints)

- Product Owner: R\$ 5.746,80
- Scrum Master: R\$ 6.136,20
- Desenvolvedor Frontend: R\$ 7.756,80
- Desenvolvedor Backend: R\$ 7.756,80
- Designer UI/UX: R\$ 2.887,50
- QA Tester: R\$ 3.555,00
- Analista de Requisitos: R\$ 2.400,00
- **Total da Fase:** R\$ 36.239,10

4. Testes e QA (2 Sprints)

- Product Owner: R\$ 1.915,60
- Scrum Master: R\$ 2.045,40

- Desenvolvedor Frontend: R\$ 1.292,80
- Desenvolvedor Backend: R\$ 1.292,80
- Designer UI/UX: R\$ 481,25
- QA Tester: R\$ 2.370,00
- Analista de Requisitos: R\$ 800,00
- **Total da Fase:** R\$ 10.197,85

5. Implantação e Go-Live (0.5 Sprint)

- Product Owner: R\$ 478,90
- Scrum Master: R\$ 511,35
- Desenvolvedor Frontend: R\$ 646,40
- Desenvolvedor Backend: R\$ 646,40
- Designer UI/UX: R\$ 240,63
- QA Tester: R\$ 296,25
- Analista de Requisitos: R\$ 200,00
- **Total da Fase:** R\$ 3.019,13

Custo Total do Projeto: R\$ 64.363,48

Além dos custos com a equipe, é fundamental considerar outros custos indiretos:

- **Hospedagem em Nuvem:** Os custos de hospedagem (AWS, Azure ou GCP) podem variar, mas para um MVP, estima-se um custo inicial de **R\$ 300,00 a R\$ 800,00 por mês**, aumentando conforme a escalabilidade e o volume de dados. Para o período do projeto, considerando 6 meses de desenvolvimento e implantação, seria em torno de **R\$ 1.800,00 a R\$ 4.800,00**.
- **Licenças de Software/Ferramentas (se aplicável):** Embora muitas ferramentas ágeis e de desenvolvimento tenham versões gratuitas, licenças premium para funcionalidades avançadas (Jira, ferramentas de design, etc.) podem adicionar um custo. Estimativa de **R\$ 100,00 a R\$ 300,00 por mês**, totalizando **R\$ 600,00 a R\$ 1.800,00** para o período do projeto.
- **Marketing e Lançamento (pós-desenvolvimento):** Embora não faça parte do custo direto de desenvolvimento, é um custo essencial para o sucesso do aplicativo após o lançamento. Não incluído na estimativa acima.

Custo Total Estimado (incluindo indiretos):

R\$ 64.363,48 (equipe) + R\$ 4.800,00 (hospedagem, máximo) + R\$ 1.800,00 (licenças, máximo) = R\$ 70.963,48

6.2. Prazos Estimados

A metodologia Scrum, com seus ciclos de Sprints, proporciona um cronograma flexível e adaptável. O prazo total estimado para o desenvolvimento do MVP (Produto Mínimo Viável) é de **aproximadamente 23 semanas (cerca de 5 a 6 meses)**, divididas nas seguintes fases:

- **1. Planejamento (1 Sprint / 2 semanas):** Definição inicial do Product Backlog, entendimento dos requisitos de alto nível e estruturação do projeto.
- **2. Design e UI/UX (2 Sprints / 4 semanas):** Criação de wireframes, protótipos, fluxos de usuário e o design visual da interface.
- **3. Desenvolvimento (6 Sprints / 12 semanas):** Implementação das funcionalidades core do sistema (cadastro de usuários e pets, registro de vacinas, consultas, medicações, alertas e lembretes).
- **4. Testes e QA (2 Sprints / 4 semanas):** Realização de testes de funcionalidade, usabilidade, segurança e desempenho para garantir a qualidade do software.
- **5. Implantação e Go-Live (0.5 Sprint / 1 semana):** Preparação e lançamento do aplicativo nos ambientes de produção (lojas de aplicativos e servidores).

Este cronograma permite entregas incrementais ao longo do tempo, possibilitando que o cliente (Product Owner) visualize o progresso e forneça feedback contínuo. É importante ressaltar que os prazos podem ser otimizados ou estendidos dependendo da complexidade de funcionalidades adicionais que possam surgir ou de ajustes prioritários durante as Sprints.

7. CANAIS DE COMUNICAÇÃO

A comunicação eficaz é um pilar fundamental para o sucesso de qualquer projeto de software, especialmente em abordagens ágeis como o Scrum. Ela garante que todos os envolvidos estejam alinhados, que as informações fluam livremente e que os problemas sejam resolvidos de forma proativa.

7.1. Comunicação Interna da Equipe

Para a equipe de desenvolvimento, serão estabelecidos diversos canais e práticas de comunicação para garantir a colaboração e o alinhamento contínuos:

- **Daily Scrums:** Reuniões diárias curtas (até 15 minutos) onde cada membro da equipe compartilha o que fez no dia anterior, o que fará no dia atual e quaisquer impedimentos. Isso promove a transparência e a rápida identificação de problemas.
- **Ferramentas de Colaboração e Gerenciamento de Projetos:**
 - **Jira/Trello:** Utilizadas para gerenciar o Product Backlog, Sprint Backlog e o fluxo de trabalho das tarefas, proporcionando visibilidade sobre o progresso e as responsabilidades.
 - **Slack/Microsoft Teams:** Canais de comunicação instantânea para discussões rápidas, compartilhamento de arquivos e resolução de dúvidas em tempo real.
 - **Confluence/Wiki:** Plataforma para documentação colaborativa, onde especificações técnicas, decisões de design, guias de uso e outros documentos importantes serão armazenados e mantidos atualizados.
- **Revisões de Código (Code Reviews):** Prática regular onde os desenvolvedores revisam o código uns dos outros para garantir a qualidade, identificar bugs e compartilhar conhecimento.
- **Retrospectivas de Sprint:** Reuniões ao final de cada sprint para que a equipe reflita sobre o processo de trabalho, identifique o que funcionou bem e o que pode ser melhorado, promovendo a melhoria contínua.

7.2. Comunicação com o Cliente (Product Owner)

A comunicação com o cliente, representado pelo Product Owner, é de suma importância para garantir que o produto desenvolvido atenda às suas expectativas e necessidades de negócio.

- **Revisões de Sprint:** Ao final de cada sprint, o incremento do produto será demonstrado ao Product Owner e outros stakeholders relevantes. Esta é uma oportunidade formal para o cliente inspecionar o trabalho concluído, fornecer feedback e solicitar ajustes no Product Backlog.
- **Reuniões de Refinamento do Backlog:** Encontros regulares entre o Product Owner e o Time de Desenvolvimento para discutir e detalhar os itens do Product Backlog, garantindo que estejam claros e prontos para serem desenvolvidos em futuras sprints.
- **Canais de Comunicação Dedicados:**
 - **E-mail:** Para comunicações formais, envio de relatórios de progresso e documentação.
 - **Ferramentas de Chat (Slack/Teams):** Para comunicação rápida e informal, resolução de dúvidas urgentes e alinhamentos diários.
 - **Reuniões Regulares (Semanais/Quinzenais):** Além das Revisões de Sprint, reuniões periódicas para discutir o status geral do projeto, alinhar expectativas e tomar decisões estratégicas.
- **Transparência do Backlog:** O Product Backlog será mantido em uma ferramenta acessível ao Product Owner, permitindo que ele acompanhe a priorização e o status dos itens a qualquer momento.

A manutenção de canais de comunicação abertos e transparentes, tanto dentro da equipe quanto com o cliente, é crucial para a adaptabilidade do projeto, a resolução proativa de desafios e a entrega de um produto que realmente gere valor.

8. CONCLUSÃO

O sistema de gerenciamento de cuidados para pets visa solucionar a desorganização e esquecimento de cuidados essenciais, oferecendo registros organizados e alertas. A metodologia Scrum será utilizada para garantir entregas contínuas. A arquitetura de software, focada em tecnologias robustas e escaláveis na nuvem, assegura performance e segurança. Requisitos funcionais e não funcionais foram detalhados, guiando o desenvolvimento. A modelagem de software oferece uma visão clara da interação do usuário e da organização interna dos dados. A estimativa de custos e prazos inicial fornece uma base para o planejamento, priorizando a entrega de um MVP. O projeto alinha a resolução de uma necessidade do usuário final com uma estratégia de negócio para o pet shop, fortalecendo a marca e otimizando operações.