
<Pet>

<My little pet>
Especificação de Requisitos de Software
Para <Subsistema ou Recurso>

Versão <1.0>

My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	

Histórico da Revisão

Data	Versão	Descrição	Autor
01/07/2024	0.0	Criação do documento, iniciando a introdução e objetivo.	Beatriz Omori
08/07/2024	0.1	Iniciando descrição dos requisitos específicos.	Beatriz Omori e Eduardo Nakai
17/07/2024	0.2	Finalizando requisitos.	Beatriz Omori e Eduardo Nakai

My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	

Índice

1.	Introdução	4	
1.1	Objetivo	4	
1.2	Escopo	4	
1.3	Definições, Acrônimos e Abreviações	4	
1.4	Referências	4	
1.5	Visão Geral	5	
2.	Descrição Geral	5	
3.	Requisitos Específicos	5	
3.1	Funcionalidade	5	
3.1.1	<Requisito Funcional Um>		6
3.2	Utilidade	6	
3.2.1	<Requisito de Utilidade Um>		7
3.3	Confiabilidade	7	
3.3.1	<Requisito de Confiabilidade Um>		8
3.4	Desempenho	8	
3.4.1	<Requisito de Desempenho Um>		8
3.5	Suportabilidade	9	
3.5.1	<Requisito de Suportabilidade Um>		10
3.6	Restrições de Design	10	
3.6.1	<Restrição de Design Um>		10
3.7	Documentação do Usuário On-line e Requisitos do Sistema de Ajuda	10	
3.8	Componentes Comprados	10	
3.9	Interfaces	10	
3.9.1	Interfaces com o Usuário		11
3.9.2	Interfaces de Hardware		12
3.9.3	Interfaces de Software		12
3.9.4	Interfaces de Comunicações		12
3.10	Padrões Aplicáveis	12	

My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	

Especificação de Requisitos de Software

1. Introdução

É um aplicativo gamificado destinado a incentivar os proprietários a cuidar de seus animais de estimação, gerenciar consultas veterinárias, despesas, cuidados (alimentação, hidratação), interação lúdica com o animal e tempo dedicado a ele. Após a finalização das tarefas, o aplicativo disponibiliza um cupom para comprar produtos com desconto. Atualmente, muitas pessoas enfrentam dificuldades em dedicar tempo para a gestão dos cuidados com seus pets, e essa plataforma serviria como um estímulo para que reservassem momentos do seu dia para se dedicarem a eles.

1.1 Objetivo

O objetivo do aplicativo é proporcionar um incentivo divertido e sistematizado para que os proprietários de animais de estimação cuidem de seus pets de maneira mais eficiente e regular. Através de um sistema de gamificação, o aplicativo visa facilitar a alimentação, hidratação, interação e tempo de cuidado, recompensando os usuários com cupons de compra. Assim, buscando promover o bem-estar dos animais e estimular um compromisso contínuo dos proprietários com a saúde e felicidade de seus companheiros.

1.2 Escopo

Objetivo do projeto é finalizar o sistema com todas as funcionalidades descritas no documento funcionando.

Recursos

- Equipe 2 pessoas, 20 horas de trabalho semanal durante 16 semanas.

Roteiro do projeto e cronograma:

- **01 de julho:** começar a definir o escopo.
- **02 de julho a 18 de julho:** fazer o documento de requisitos
- **19 de julho:** revisar o documento de requisitos
- **20 de julho a 03 de agosto:** fazer o protótipo.
- **05 de agosto a 17 de agosto:** começar back-end do site
- **19 de agosto a 31 de agosto:** começar front-end do site
- **02 de setembro a 11 de setembro:** começar back-end do app
- **12 de setembro a 21 de setembro:** começar front-end do app
- **31 de novembro:** finalizar o site
- **14 de novembro:** finalizar app

My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	

Modelo de Caso de Uso



1.3 Definições, Acrônimos e Abreviações

- **SRS:** Software Requirements Specification - Documento que descreve de forma detalhada os requisitos de software.
- **RNF:** Requisito Não Funcional - Requisitos que não estão relacionados diretamente às funcionalidades do sistema, mas a aspectos como desempenho, segurança, usabilidade, etc.
- **API:** Application Programming Interface - Conjunto de funções e procedimentos que permitem a criação de aplicativos que acessam dados ou recursos de um serviço operacional.
- **PDF:** Portable Document Format - Formato de arquivo utilizado para representar documentos de maneira independente de software, hardware e sistema operacional.

My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	

- **CDN:** Content Delivery Network - Rede de distribuição de conteúdo usada para entregar páginas e outros conteúdos da Web aos usuários com maior velocidade e eficiência.

1.4 Referências

- PINHO, Diego. Engenharia de Software: 3 Requisitos Não Funcionais. DevMedia. Disponível em: <https://www.devmedia.com.br/artigo-engenharia-de-software-3-requisitos-nao-funcionais/9525#modulo-mvp>. Acesso em: 21 jul. 2024
- UNIVERSIDADE FEDERAL DE PERNAMBUCO. Requisitos. Centro de Informática – UFPE, s.d. Disponível em: https://www.cin.ufpe.br/~gta/rup-vc/core.base_rup/guidances/concepts/requirements_62E28784.html#Functionality. Acesso em: 21 jul. 2024.

1.5 Visão Geral

No documento será apresentado os tópicos listados a seguir:

Descrição Geral: Introduce o sistema e seus principais objetivos, sem focar em requisitos específicos. Fornece uma visão geral das funcionalidades e características desejadas, facilitando a compreensão dos requisitos detalhados que serão descritos nas próximas seções.

Requisitos Não Funcionais: Definem características que não estão diretamente ligadas às funcionalidades do sistema, mas que são essenciais para garantir a qualidade e usabilidade do produto. Os requisitos incluem:

- **Funcionalidade:** Refere-se às características que o sistema deve oferecer, como segurança no acesso, registro de ações e geração de relatórios.
- **Utilidade:** Enfatiza a facilidade de uso do sistema, garantindo que os usuários possam aprender e operar o sistema de forma eficiente, como uma interface simples e um sistema de suporte para dúvidas e problemas.
- **Confiabilidade:** Relaciona-se com a disponibilidade do sistema, tempo de funcionamento e taxas de erro, assegurando que o sistema funcione de forma estável e confiável na maior parte do tempo.
- **Desempenho:** Define o tempo de resposta, a capacidade de processamento e a escalabilidade do sistema, como suportar muitos usuários simultâneos sem perder desempenho.
- **Suportabilidade:** Foca na manutenção e atualização do sistema sem comprometer a integridade dos dados, assegurando a compatibilidade com versões anteriores e a adaptação a futuras atualizações.
- **Restrições de Design:** Especifica as tecnologias e frameworks que serão usados no desenvolvimento do sistema, como Java, Golang, React Native, PostgreSQL e Docker.
- **Documentação do Usuário On-line e Requisitos do Sistema de Ajuda:** Inclui a criação de guias, FAQs e notificações para auxiliar os usuários a utilizarem o sistema de forma eficaz.
- **Componentes Comprados:** Define se o sistema usará ou não componentes prontos adquiridos de terceiros.
- **Interfaces:** Descreve a interação do sistema com outros sistemas, APIs e componentes externos.
- **Padrões Aplicáveis:** Normas ou guidelines que o sistema deve seguir, seja em termos de desenvolvimento ou de segurança.

My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	

2. Descrição Geral

Perspectiva do Produto

O produto é um sistema gamificado voltado para donos de animais de estimação, com o objetivo de ajudá-los a organizar e gerenciar os cuidados com seus pets. O sistema será uma aplicação multiplataforma (web e móvel) que permitirá o agendamento de tarefas relacionadas ao bem-estar dos animais, como alimentação, hidratação e atividades recreativas.

Funções do Produto

As principais funções do produto incluem:

- Gerenciamento de perfis de pets, com informações detalhadas sobre cada animal.
- Registro e acompanhamento de atividades diárias, como alimentação e hidratação.
- Sistema de missões, onde o usuário precisa finalizar todas as atividades do dia e no final ganha um cupom para usar na compra de produtos.
- Geração de relatórios detalhados sobre vendas, anúncios e pedidos.

Características do Usuário

Os usuários do sistema são, principalmente, donos de animais de estimação que buscam melhorar a organização e gestão do cuidado com seus pets. Esses usuários podem ter diferentes níveis de familiaridade com tecnologia, variando de pessoas com experiência limitada em aplicativos móveis até usuários avançados. Portanto, o sistema precisa ser intuitivo e fácil de usar, garantindo uma curva de aprendizado rápida.

Restrições

- O sistema deve ser compatível com diferentes plataformas (web, iOS, Android) para alcançar uma ampla gama de usuários.
- Precisa operar dentro de restrições de tempo de resposta e desempenho, especialmente durante períodos de alta demanda.
- A segurança dos dados dos usuários é uma prioridade, especialmente em relação a informações sensíveis, como dados pessoais e detalhes financeiros.
- O design do sistema deve considerar a escalabilidade, permitindo a adição de novas funcionalidades e a integração com sistemas de terceiros, como clínicas veterinárias e plataformas de pagamento.

Premissas e Dependências

- A disponibilidade de infraestrutura adequada (servidores, banco de dados) para garantir o desempenho e escalabilidade necessários.
- A integração com APIs externas para notificação por push, envio de e-mails, e possivelmente integração com sistemas veterinários.
- O produto deve contar com uma equipe de suporte ativa e uma infraestrutura de manutenção para atualizações contínuas e correções de bugs.
- O sucesso do sistema depende da adesão dos usuários à plataforma, o que exige uma interface atraente e funcional.

My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	

3. Requisitos Específicos

3.1 Funcionalidade

Esta seção descreve os requisitos funcionais do sistema gamificado para cuidados com pets. A organização segue a estrutura baseada em recursos, detalhando as funcionalidades principais do sistema, incluindo segurança, relatórios, agendamento de missões e outros aspectos críticos.

3.1.1 <Requisito Funcional Um>

Autenticação de Usuário

O sistema deve exigir autenticação para todos os usuários, incluindo administradores, para acessar suas funcionalidades. O login será feito utilizando e-mail e senha.

- **Regras de Negócio:**
 - Caso o usuário insira a senha incorretamente mais de 3 vezes, será enviada uma autenticação via código para o e-mail cadastrado.
 - Todos os acessos e alterações no sistema devem gerar um log contendo o nome do usuário, a data e a hora.
- **Requisitos Não Funcionais:**
 - A autenticação deve ser processada em menos de 2 segundos.

Trilhas de Auditoria

O sistema deve registrar logs de auditoria para todas as ações importantes realizadas pelos usuários. Isso inclui login, edição e exclusão de dados.

- **Regras de Negócio:**
 - Cada log de auditoria deve conter o nome do usuário, a data, a hora e a descrição da ação realizada.
- **Requisitos Não Funcionais:**
 - Os logs devem ser armazenados por um período mínimo de 2 anos.

Relatórios de Faturamento e Vendas

O sistema deve oferecer a opção de gerar relatórios de faturamento, vendas, e dados dos anúncios. Esses relatórios devem ser exportáveis e impressos no formato PDF.

- **Regras de Negócio:**
 - O sistema deve permitir a exportação dos relatórios de faturamento, vendas e anúncios em PDF.
 - Relatórios de vendas e faturamento devem incluir informações detalhadas sobre os produtos vendidos, valores, e datas.

My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	

- **Requisitos Não Funcionais:**
 - A geração de relatórios deve ser concluída em até 7 segundos.

Segurança dos Dados

Os dados do sistema, especialmente os dados sensíveis, como informações de login e transações, devem ser protegidos.

- **Regras de Negócio:**
 - Todo dado sensível deve ser criptografado tanto em repouso quanto em trânsito.
 - O sistema deve seguir as regulamentações de segurança de dados (GDPR).
- **Requisitos Não Funcionais:**
 - A criptografia deve ser feita com padrões AES-256 para dados em repouso e TLS 1.2+ para dados em trânsito.

Sistema de Missões Gamificadas

O sistema deve permitir que os usuários participem de missões relacionadas ao cuidado de seus pets.

- **Regras de Negócio:**
 - Missões devem incluir atividades como alimentação, hidratação, brincadeiras e caminhadas.
- **Requisitos Não Funcionais:**
 - O processamento das missões deve ser feito em menos de 1 segundo após a conclusão.

3.2 Utilidade

Esta seção descreve os requisitos de utilidade do sistema, que visam garantir que o aplicativo seja fácil de usar, eficiente e agradável para os usuários. A interface deve ser simples e intuitiva, e os usuários devem poder aprender a utilizá-la em pouco tempo, além de contar com um suporte acessível.

3.2.1 <Requisito de Utilidade Um>

Facilidade de Aprendizagem

A interface do aplicativo será desenhada para ser fácil de aprender, permitindo que novos usuários se familiarizem com suas funcionalidades em até 1 hora. Isso garantirá que os usuários consigam utilizar o sistema sem a necessidade de um treinamento extenso, o que facilita a adoção do aplicativo.

Suporte ao Usuário

O sistema fornecerá uma área de suporte acessível dentro da aplicação, onde os usuários podem tirar dúvidas, relatar erros ou resolver problemas que possam surgir durante o uso do aplicativo. Isso é essencial

My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	

para manter a satisfação do usuário e assegurar que eventuais problemas sejam rapidamente solucionados.

Simplicidade e Intuição na Interface

A interface do sistema será simples e intuitiva, garantindo um entendimento rápido e fácil por parte dos usuários. Uma interface bem projetada e de fácil uso aumenta a eficiência do usuário nas tarefas e reduz a necessidade de treinamento adicional, proporcionando uma experiência agradável e motivando o uso contínuo do aplicativo.

Eficiência na Tarefa

Os usuários devem ser capazes de realizar tarefas comuns (como registrar o cuidado com seus pets ou agendar missões) com o mínimo de cliques e em um tempo otimizado. A meta é que os usuários completem tarefas principais em até 3 cliques e dentro de um prazo de 5 segundos, desde que os dados já estejam disponíveis.

Facilidade de Recordação

Após um período de não utilização do sistema, os usuários devem ser capazes de recordar rapidamente como realizar tarefas básicas, como agendar missões ou visualizar o histórico de atividades de seus pets. O sistema será projetado para minimizar a necessidade de re-aprendizado, o que aumenta a eficiência e reduz a frustração dos usuários.

Entendimento das Mensagens e Alertas

Mensagens e alertas dentro do sistema devem ser claros e fáceis de entender, comunicando exatamente o que aconteceu e o que o usuário deve fazer a seguir. As mensagens de erro seguirão um padrão consistente: "Erro: (tipo do erro)", garantindo que os usuários compreendam rapidamente a situação e tomem as ações corretas.

3.3 Confiabilidade

Esta seção descreve os requisitos de confiabilidade do sistema, garantindo que o aplicativo seja altamente disponível, com uma baixa taxa de falhas e erros, assegurando uma experiência contínua e confiável para os usuários.

3.3.1 <Requisito de Confiabilidade Um>

Disponibilidade

O sistema deve estar disponível 99% do tempo, o que garante alta disponibilidade para os usuários realizarem suas atividades de gerenciamento dos cuidados com seus pets. A manutenção programada será realizada durante períodos de menor uso para minimizar impactos.

Tempo Médio Entre Falhas (MTBF)

My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	

O sistema deve ter um tempo médio entre falhas (MTBF) de 1488 horas (aproximadamente 2 meses). Isso assegura que o aplicativo terá uma alta confiabilidade, oferecendo uma experiência estável e contínua para os usuários.

Tempo Médio para Reparos (MTTR)

Em caso de falha, o sistema tem um tempo máximo de 2 horas para ser restaurado e voltar a funcionar. Isso garante que o impacto sobre os usuários seja minimizado e que a disponibilidade do sistema seja rapidamente restabelecida.

Taxa Máxima de Erros ou Defeitos

A taxa máxima de erros permitida no código é de 0,5 erros por mil linhas de código (KLOC). Isso assegura que o código do sistema será de alta qualidade, minimizando a ocorrência de problemas que possam comprometer a experiência do usuário.

Taxa de Erros ou Defeitos Críticos

Um erro crítico é definido como qualquer falha que cause perda completa de dados ou inabilidade total de uso de funcionalidades principais do sistema. A ocorrência de erros críticos deve ser extremamente rara, assegurando a confiança e o engajamento dos usuários.

3.4 Desempenho

Esta seção detalha os requisitos de desempenho do sistema, assegurando que o aplicativo funcione com rapidez e eficiência, mesmo sob alta demanda.

3.4.1 <Requisito de Desempenho Um>

Tempo de Resposta

Todas as interações entre o usuário e o sistema devem ter um tempo de resposta máximo de 2 segundos, garantindo uma experiência de uso ágil e sem atrasos.

Taxa de Transferência

O sistema deve ser capaz de processar 100 transações por segundo, assegurando uma alta capacidade de processamento para múltiplos usuários simultâneos.

Partida

O tempo necessário para o sistema entrar em funcionamento, após ser iniciado, não deve ultrapassar 2 segundos, proporcionando uma inicialização rápida.

Parada

My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	

O sistema deve ser capaz de encerrar suas atividades em até 2 segundos, garantindo que o fechamento do aplicativo ocorra rapidamente.

Capacidade

O sistema deve suportar até 20.000 usuários simultâneos no período de pico entre 7:30h e 11:00h, e até 15.000 usuários simultâneos nos demais períodos, assegurando que o desempenho não será comprometido mesmo com alta demanda.

3.5 Suportabilidade

A seção de suportabilidade estabelece os requisitos para garantir a fácil manutenção, atualização, e escalabilidade do sistema, permitindo que ele seja adaptado e compatível com futuros desenvolvimentos e ambientes de execução.

3.5.1 <Requisito de Suportabilidade Um>

Adaptabilidade

- **Atualizações:** O sistema deve permitir a aplicação de atualizações sem perda de dados ou interrupção prolongada dos serviços.
- **Dependências e Frameworks:** O sistema deve ser preparado para futuras atualizações de dependências e frameworks nativos nas plataformas suportadas.
- **Migração de Dados:** Deve haver suporte para migração de dados entre versões anteriores e posteriores do aplicativo, assegurando que dados de perfil, progresso de pets permaneçam intactos.

Compatibilidade

- **Integração com Sistemas Legados:** O aplicativo deve ser compatível com sistemas legados em caso de integração com ferramentas de terceiros, como plataformas de pagamentos para serviços relacionados à pets.

Configurabilidade

- **Personalização:** O produto deve ser configurável após a instalação, permitindo que os usuários personalizem certos aspectos da interface e as notificações de lembretes.

Instalação

- **Verificação de Dependências:** O instalador deve verificar se todas as dependências necessárias (como bibliotecas de sistema e permissões) estão presentes antes de completar a instalação.
- **Automação:** O processo de instalação deve ser automatizado e incluir verificação de pré-requisitos para todas as plataformas suportadas (Android, iOS, Windows).

Nível de Suporte

- **Help-Desk:** O sistema deve fornecer suporte via chat ou e-mail com tempo de resposta de até 24

My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	

horas. Além disso, deve haver uma base de conhecimento online com artigos de solução de problemas e FAQs.

Manutenção

- **Liberações:** As liberações de manutenção devem ser oferecidas uma vez por trimestre, com correções de bugs e melhorias incrementais de desempenho e segurança.
- **Automatização:** A manutenção do sistema deve ser automatizada na medida do possível, com atualizações automáticas para dispositivos móveis, sempre que aplicável.

Testabilidade

- **Testes Automatizados:** Deve existir um conjunto de testes automatizados para validar a integridade dos dados, desempenho e segurança do sistema após cada atualização.
- **Testes Unitários e de Integração:** O sistema deve permitir testes unitários e de integração, simulando o comportamento do usuário para garantir que todas as funcionalidades principais (registro de pets, pontuação, notificações) estejam funcionando corretamente.
- **Testes de Carga e Estresse:** O aplicativo deve suportar testes de carga e estresse para garantir que a infraestrutura backend possa suportar a escalabilidade planejada.

3.6 Restrições de Design

Esta seção detalha as restrições de design obrigatórias para o sistema em desenvolvimento, incluindo decisões sobre linguagens de programação, ferramentas de desenvolvimento, e componentes utilizados.

3.6.1 <Restrição de Design Um>

Linguagens de Programação:

- **Back-End:** Java - Utilizado no desenvolvimento de back-end devido à sua robustez, portabilidade e ampla comunidade de suporte. E Golang - Ideal para partes do back-end que requerem processamento rápido e simultâneo.
- **Front-End:** JavaScript - Amplamente suportado pelos navegadores, permitindo criar experiências dinâmicas e responsivas no lado do cliente.

Documentação de API:

- **Swagger:** Utilizado para descrever e documentar a API. Permite criar documentação interativa e legível para APIs, facilitando a compreensão e o uso das APIs por desenvolvedores e outras partes interessadas.

Testes Automatizados:

My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	

- **Vitest:** Utilizado para criar testes automatizados em aplicações JavaScript. Permite escrever e executar testes automatizados para garantir que o código esteja funcionando corretamente e para detectar regressões.

Containerização:

- **Docker:** Utilizado para o desenvolvimento do aplicativo. Facilita a configuração de ambientes de desenvolvimento e produção consistentes e portáteis, isolando dependências e garantindo que o aplicativo funcione da mesma maneira em diferentes ambientes.

Execução de JavaScript no Servidor:

- **Node.js:** Utilizado para executar JavaScript no servidor. Permite que o JavaScript seja utilizado para construir aplicações de back-end, oferecendo uma plataforma eficiente e escalável para execução de código JavaScript fora do navegador.

Desenvolvimento de Aplicativos Móveis:

- **React Native:** Utilizado para criar aplicativos móveis. Possibilita criar aplicativos para iOS e Android com uma base de código única, aproveitando a experiência de desenvolvimento web com React.

Validação, transformação, Envio de dados HTTP

- **Zod:** biblioteca feita para gerenciar dados e validações;
- **React-hook-form:** Biblioteca de gerenciamento de formulários;
- **React-query:** Biblioteca para gerenciamento de requisições HTTP;
- **Axios:** Client HTTP para fazer requisições e consumo de APIs;

Desenvolvimento de Aplicações React:

- **Next.js:** Utilizado para desenvolver aplicações React. Framework para React que oferece renderização do lado do servidor e geração de sites estáticos.

Storage de arquivos:

- **Supabase:** Utilizando para Storage de gerenciamento de Storage de arquivos.

Gerenciamento de Banco de Dados:

- **PostgreSQL:** Sistema de gerenciamento de banco de dados utilizado. Escolha do PostgreSQL para gerenciamento de banco de dados devido à sua robustez, escalabilidade e suporte para operações complexas.

My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	

3.7 Documentação do Usuário On-line e Requisitos do Sistema de Ajuda

Para garantir que os usuários tenham acesso fácil e eficiente às informações necessárias para utilizar o sistema, serão implementados os seguintes requisitos de documentação e suporte:

- **Guia de Início Rápido:** Para ajudar novos usuários a configurar perfis de pets e entender o sistema de pontos e níveis.
- **Notificações In-App e Push:** Lembrar os usuários de tarefas pendentes, como alimentar o pet ou oferecer água.

3.8 Componentes Comprados

Nenhum componente comprado será utilizado na aplicação. Todos os componentes de software integrados serão de código aberto ou desenvolvidos internamente pela equipe do projeto, garantindo flexibilidade e controle sobre as funcionalidades do sistema.

3.9 Interfaces

Esta seção especifica as interfaces que o aplicativo deve suportar, incluindo protocolos, portas, endereços lógicos e outras informações relevantes para garantir a integração e a comunicação adequada entre o sistema e outros componentes ou serviços.

1. Interface de API

- **Descrição:** Interface para comunicação entre o cliente e o servidor, permitindo operações como criação de perfis de pets, pontuação, lembretes, e visualização de progresso.
- **Protocolos:** HTTP/HTTPS
- **Porta:** 80 (HTTP), 443 (HTTPS)
- **Endereço Lógico:** <https://mylittlepet.com/>
- **Endpoints Principais:**
 - /pets
 - /usuario
 - /missao
 - /produtos
 - /pedidos
 - /banner

2. Interface de Banco de Dados

- **Descrição:** Interface para comunicação com o sistema de gerenciamento de banco de dados PostgreSQL.
- **Protocolos:** TCP/IP
- **Porta:** 5432
- **Endereço Lógico:** <jdbc:postgresql://localhost:5432/>

My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	

- **Schemas:**
 - usuarios
 - banners
 - pets
 - missão
 - produtos
 - categories
 - stores
 - orders
 - productinorders

3.9.1 Interfaces com o Usuário

O aplicativo será utilizado por proprietários de pets para gerenciar atividades diárias de cuidado, saúde e bem-estar dos seus animais. As interfaces aqui descritas permitirão ao usuário registrar, consultar, modificar e gerenciar tarefas e informações relacionadas ao seu pet. O foco será na gamificação, oferecendo uma experiência interativa para aumentar o engajamento do usuário nos cuidados dos seus pets, como alimentação, exercícios, entre outros.

Identificador	Nome	Descrição
UI.0001	Tela de cadastro de usuário	Interface para cadastrar o usuário com as informações de nome, email, data de nascimento, CPF e idade.
UI.0002	Tela de missões	Tela exibe as missões diárias, junto com missões extras para o usuário completar.
UI.0004	Tela de produtos	Tela onde o usuário poderá escolher produtos para colocar dentro do carrinho de compras.
UI.0005	Tela de cadastro de pet	Interface para registrar e consultar informações sobre o pet, como nome, data de nascimento, raça, idade e porte.
UI.0006	Tela de cadastro de produto	Interface para cadastrar informações do produto como a descrição, código, quantidade, tipo e valor.
UI.0007	Tela de cadastro de anúncio	Interface para cadastrar os anúncios exibidos na tela inicial do app.

My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	

Aspectos Visuais

- Cores principais: O aplicativo utilizará como cores principais um verde-azulado (#008080) e laranja (#F7AB2A), criando uma aparência autêntica e agradável.
- Logo: O sistema terá um logo de um cachorro da raça Shiba Inu, que visa chamar a atenção de usuários que possuem pets.
- Menu: No app mobile, o menu será posicionado na parte inferior da tela, enquanto na interface web, o menu estará à esquerda.
- Cabeçalho: O cabeçalho terá 72 pixels de altura com a cor laranja (#F7AB2A), garantindo destaque visual.
- Menu no app: Terá 120 pixels de altura com fundo branco (#FFFFFF) para facilitar a navegação e manter a interface limpa.
- Responsividade: Os elementos da interface serão adaptáveis para diferentes tamanhos de dispositivos, proporcionando uma experiência fluida tanto em smartphones quanto na web.
- Destaque de dados importantes: Dados relevantes serão exibidos em negrito e com cores chamativas para facilitar a identificação pelo usuário.

Requisitos de Layout e Navegação

- O layout será responsivo e manterá consistência em diferentes plataformas. O menu ficará na parte inferior no app mobile e à esquerda na versão web.
- Informações importantes, como alertas ou dados cruciais sobre o pet, serão destacados para melhorar a experiência do usuário.
- Mensagens de erro seguirão o formato padrão: "Erro: (tipo do erro)", garantindo clareza e uniformidade.
- Mensagens de confirmação devem ser claras para evitar mal-entendidos e garantir que o usuário esteja ciente das ações tomadas.

Consistência

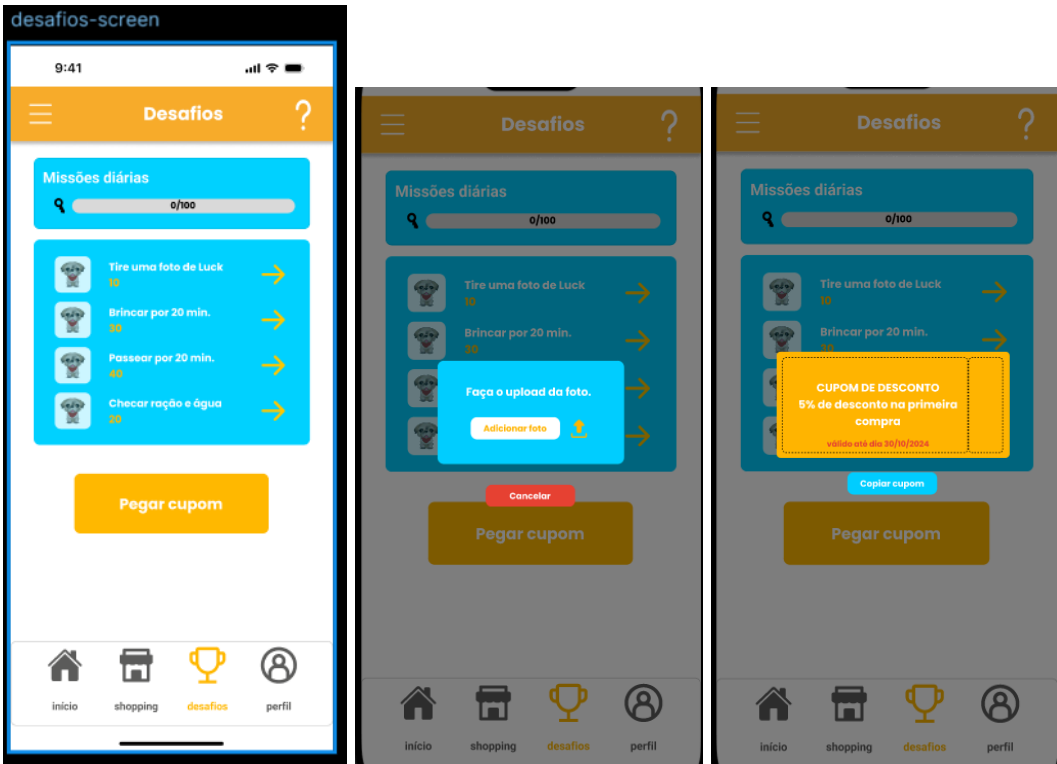
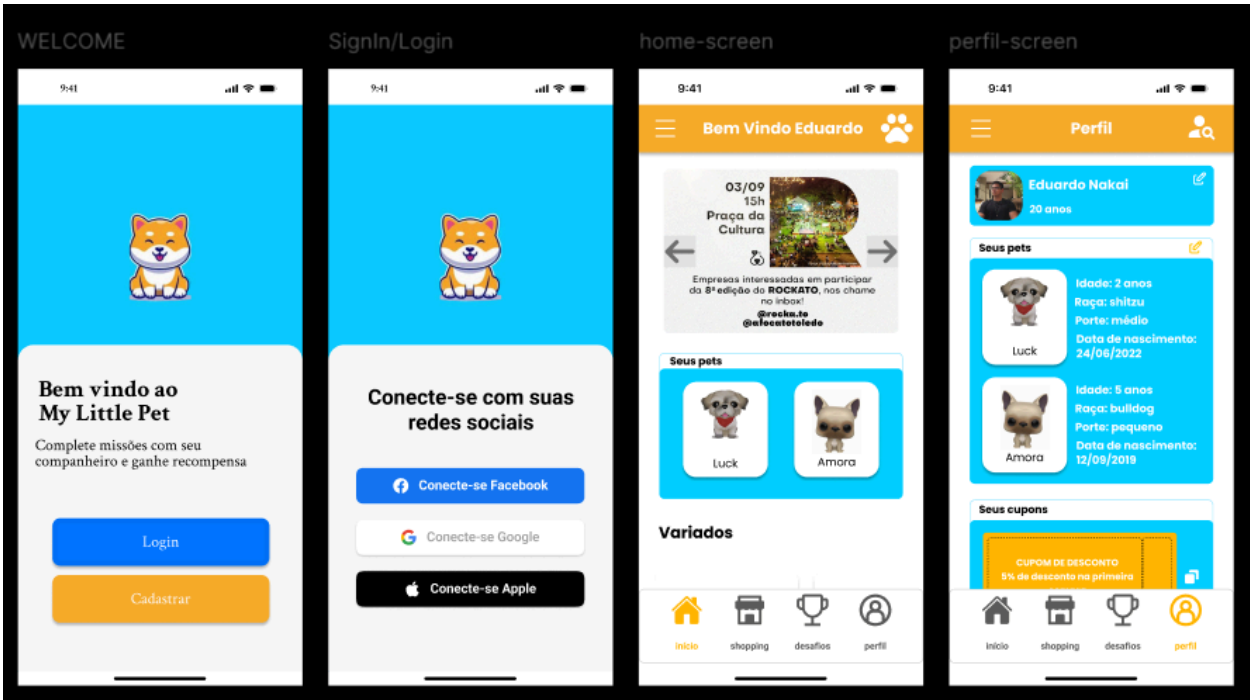
- A consistência será mantida em todos os elementos visuais e funcionais, garantindo que o usuário saiba o que esperar ao navegar pelo sistema. A navegação, botões de ação e áreas de destaque seguirão um padrão claro e uniforme.

Requisitos de Personalização

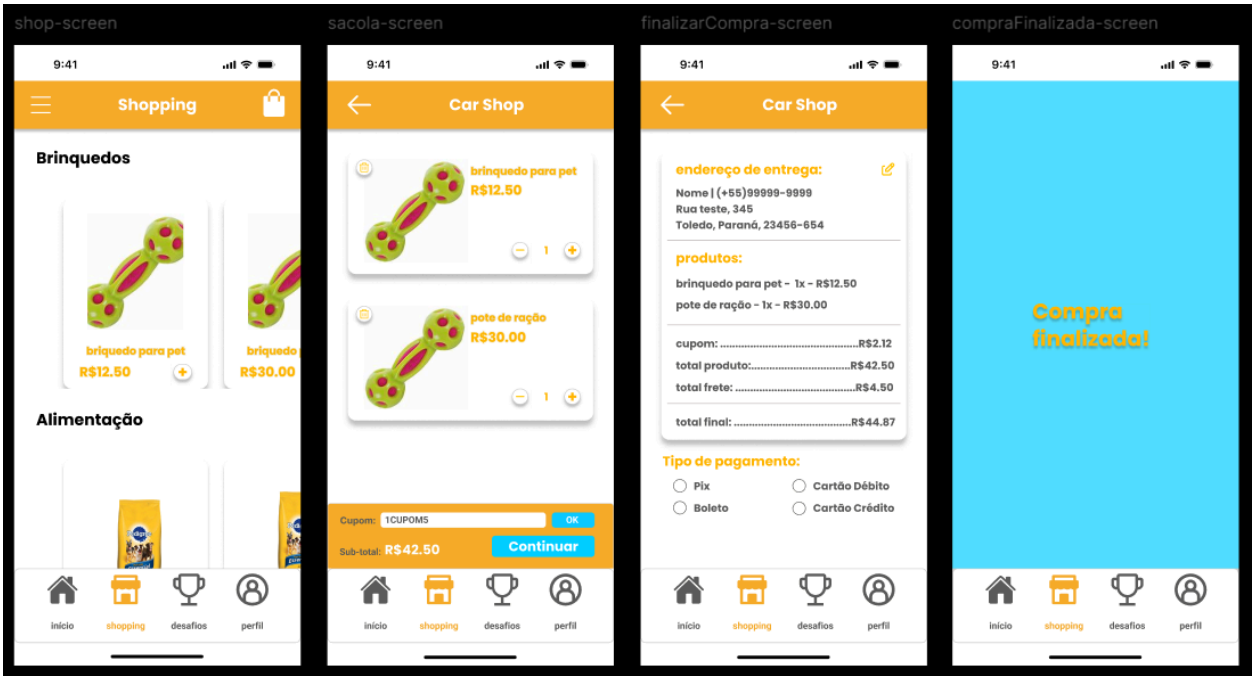
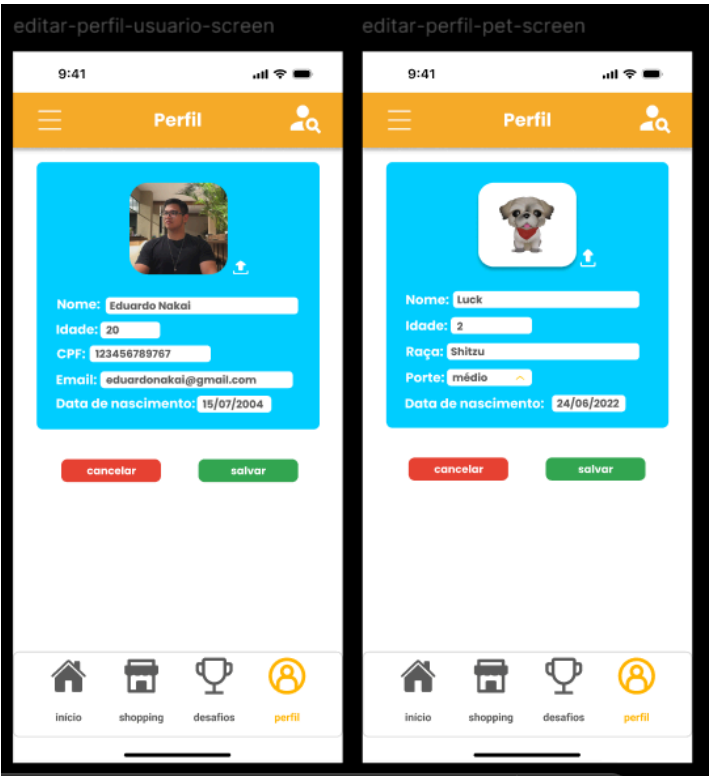
- Missões e notificações: O sistema exibirá missões inacabadas na página inicial e permitirá que os usuários configurem notificações, personalizando o conteúdo com base em preferências de cuidado ou eventos importantes relacionados ao pet.

Protótipo Aplicativo

My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	



My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	




My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	

pag-pix

←

Valor R\$44.87



copiar código

pag-cartão

←

Valor R\$44.87

Número

Nome

CVV

Vencimento:

Mês


Ano

Pagar

pag-boleto

←

Valor R\$44.87

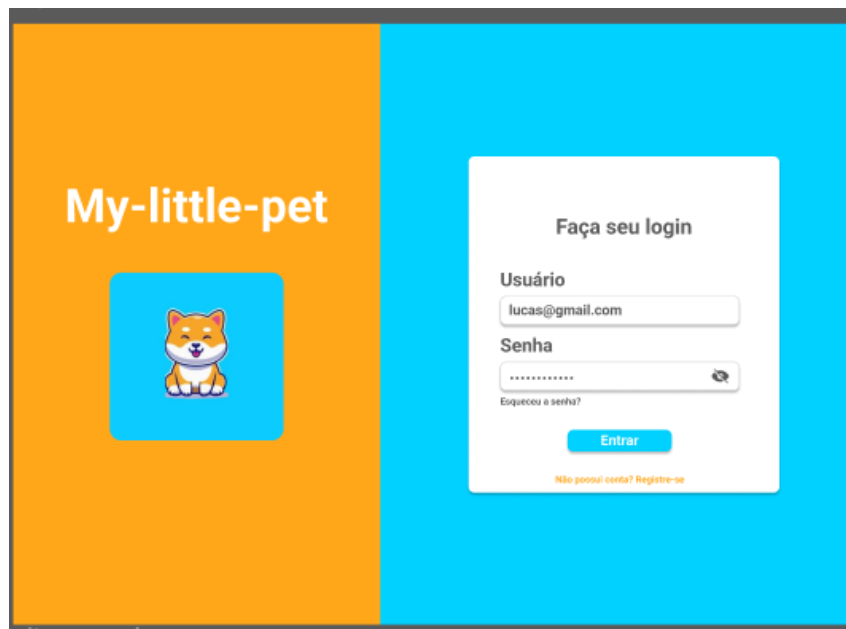


baixar boleto

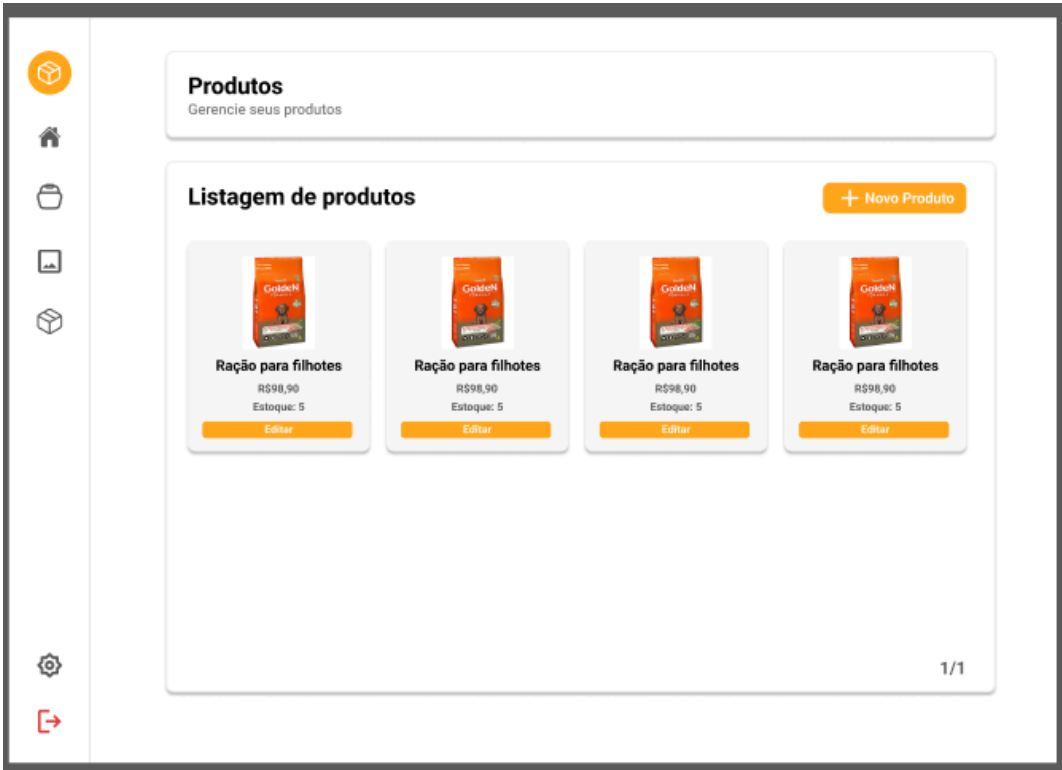
copiar código

Protótipo Site

My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	



My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	



My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	

Novo Produto

Título

Descrição

Preço R\$

Estoque

Categoria

Selecionar categoria

Arquivo

Cancelar

Salvar

Editar Produto

Título

Ração para filhotes

Descrição

Saco de ração de 10Kg

Preço R\$

R\$98,90

Estoque

5

Categoria

Alimentação

Arquivo

racao.png

Cancelar

Salvar

Anúncios

Gerencie seus anúncios

Listagem de anúncios

+ Novo Anúncio

id	título	status	crédito	prazo final	saiba mais
8dec09c-8bf6-4246-9feb-d6533263d93b	Adote um amigo	Ativo	R\$100,00	25/09/2024	Saiba mais

My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	

Cadastrar um anúncio

Título

Descrição

Link externo

Data limite

Crédito

Imagem

Cancelar

Salvar

Anúncios

Gerencie seus anúncios

Informação do anúncio

Título

Adote um amigo

Descrição

Adote um amigo

Link externo

adoteumavida.com.br

Data limite

25/09/2024

Crédito

R\$100,00

Status pagamento: Pago

Status: Ativo

Cliques: 5

ADOTE UM AMIGO

Alterar imagem

Salvar

Desativar

Confidencial

©<Pet>, 2022

Página 24

My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	



3.9.2 Interfaces de Hardware

Esta seção define as interfaces de hardware que o software gamificado para cuidados com pets deve suportar, garantindo a interação adequada com os dispositivos físicos utilizados pelos usuários.

Estrutura Lógica

O aplicativo será executado em dispositivos móveis (smartphones e tablets) e desktops, sendo compatível com sistemas operacionais Android, iOS e Windows. Ele deve se adaptar a diferentes tamanhos de tela e resoluções, mantendo a responsividade para garantir uma boa experiência do usuário.

Endereços Físicos

Dispositivos móveis: O aplicativo acessa o armazenamento interno do dispositivo para salvar dados do usuário, como imagens dos pets e histórico de atividades, usando permissões de armazenamento fornecidas pelo sistema operacional.

3.9.3 Interfaces de Software

Esta seção descreve as interfaces de software que o aplicativo de cuidados com pets terá com outros componentes, incluindo sistemas externos, APIs e serviços necessários para a funcionalidade completa.

Integrações com Sistemas Externos

- **Serviços de Pagamento:** O sistema utilizará APIs de pagamento (Stripe) para permitir compras de produtos relacionados ao cuidado dos pets diretamente no aplicativo.

My little pet	Versão: <1.0>
Especificação de Requisitos de Software	Data: 01/07/2024
<identificador do documento>	

Componentes Reutilizados

- **Integração com Sistema de Login:** O sistema utilizará serviços de autenticação de terceiros (Google, Facebook, Apple ID) para facilitar o login do usuário, utilizando protocolos OAuth2.0.

Protocolos e Formatos de Dados

- **Formato de Dados:** O aplicativo fará uso de JSON como formato de dados para troca de informações entre o backend e as APIs externas.
- **Protocolos de Comunicação:** Todas as comunicações com APIs e sistemas externos ocorrerão via HTTPS para garantir segurança e criptografia dos dados do usuário e do pet.

3.9.4 Interfaces de Comunicações

Esta seção descreve as interfaces de comunicação que o software utilizará para interagir com outros sistemas e dispositivos em redes locais ou remotas.

Redes de Área Local (LANs)

O aplicativo deve ser capaz de sincronizar dados em tempo real com servidores remotos via redes locais ou conexões Wi-Fi, garantindo que as informações sejam atualizadas instantaneamente nos perfis dos pets.

Dispositivos Remotos

- **Cloud Services:** O sistema deve se comunicar com servidores em nuvem (AWS ou Google Cloud) para armazenar dados de histórico de cuidados, imagens e progresso de missões, utilizando APIs RESTful.

3.10 Padrões Aplicáveis

Esta seção descreve os padrões que o software gamificado para cuidados com pets deve seguir, garantindo conformidade com normativas técnicas, jurídicas e de mercado.

Padrões de Interoperabilidade

- **OAuth 2.0:** O sistema de autenticação deve estar em conformidade com o protocolo OAuth 2.0 para garantir a segurança nas integrações com serviços de login de terceiros (Google, Facebook, etc.).
- **OpenAPI (Swagger):** A documentação das APIs do sistema seguirá o padrão OpenAPI (Swagger) para garantir clareza e interoperabilidade nas integrações com desenvolvedores externos.