
VVet Tech

PetPlace
Especificação de Requisitos de Software
Para Marketplace com Personalização Avançada

Versão 6.0

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

Histórico da Revisão

Data	Versão	Descrição	Autor
01/09/2024	1.0	Adição do conteúdo do documento.	Giovanna Santos Silva e Murilo Faveri.
02/09/2024	1.1	Alteração na Introdução, dos tópicos do 3.2 ao 4 e correção dos erros ortográficos. Adição do Diagrama de Caso de Uso e dos protótipos do logo e do site.	Giovanna Santos Silva e João Vítor Nogueira.
03/09/2024	1.2	Alteração do Índice.	Giovanna Santos Silva
23/09/2024	2.0	Adição do Diagrama de Casos de Uso.	Giovanna Santos Silva
23/09/2024	2.1	Adição da documentação dos casos de uso.	Enrico Genaro; Giovani Tirapani; João Nogueira.
24/09/2024	2.2	Correção da documentação dos casos de uso.	Murilo Faveri.
24/09/2024	2.3	Alteração no Histórico de Revisão.	Giovanna Santos Silva
25/09/2024	2.4	Alteração do Índice.	Giovanna Santos Silva
05/10/2024	3.0	Adição do Diagrama de Classe e da explicação do diagrama.	Giovanna Santos Silva; João Nogueira; Giovani Tirapani;
30/10/2024	4.0	Adição do Diagrama de Componentes e da explicação do diagrama.	Giovanna Santos Silva; Giovani Tirapani;
30/10/2024	5.0	Adição do Diagrama de Implementação e da explicação do diagrama.	João Nogueira; Giovanna Santos Silva;
30/10/2024	5.1	Correção e Revisão de toda a documentação do projeto.	Murilo Faveri; Enrico Genaro; Giovani Tirapani.
10/11/2024	6.0	Revisão final do projeto e últimos ajustes.	Giovanna Silva; Giovani Tirapani;

Equipe:

- Enrico Nalini Genaro;
- Giovani Maia Tirapani;
- Giovanna Santos Silva;
- João Vítor da Silva Nogueira;
- Murilo Vinicius de Faveri.

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

Índice

1.	Introdução	5
1.1	Objetivo	5
1.2	Escopo	5
1.3	Definições, Acrônimos e Abreviações	5
1.4	Referências	5
1.5	Visão Geral	6
2.	Descrição Geral	6
3.	Requisitos Específicos	7
3.1	Funcionalidade	7
3.1.1.	RF01 - Cadastro de Usuário	7
3.1.2.	RF02 - Login de Usuário	7
3.1.3.	RF03 - Cadastro de Pets	7
3.1.4.	RF04 - Catalogar Produtos e Serviços	7
3.1.5.	RF05 - Pesquisa e Navegação	7
3.1.6.	RF06 - Algoritmo e Recomendação (Anúncio)	7
3.1.7.	RF07 - Integração com Histórico de Saúde	7
3.1.8.	RF08 - Gestão de Carrinho de Compras	7
3.1.9.	RF09 - Gestão de Pedidos	7
3.1.10.	RF10 - Gestão de Entrega	7
3.1.11.	RF11 - Gestão de Endereços	8
3.1.12.	RF12 - Agendamento de Serviços	8
3.1.13.	RF13 - Pagamento Online	8
3.1.14.	RF14 - Chat	8
3.1.15.	RF15 - Resolução de Problemas	8
3.1.16.	RF16 - Avaliação e Feedback	8
3.1.17.	RF17 - Notificação Personalizadas	8
3.2	Utilidade	8
3.2.1	RNF01 - Usabilidade	8
3.2.2	RNF02 - Compreensibilidade	8
3.2.3	RNF03 - Aprendibilidade	9
3.2.4	RNF04 - Operabilidade	9
3.2.5	RNF05 - Atração	9
3.2.6	RNF06 - Proteção Contra Erros de Usuário	9
3.2.7	RNF07 - Eficiência de Uso	9
3.2.8	RNF08 - Satisfação do Usuário	10
3.3	Confiabilidade	10
3.3.1	RNF09 - Requisitos de Confiabilidade	10
3.3.2	RNF10 - Disponibilidade	10
3.3.3	RNF11 - Tolerância a Falhas	10
3.3.4	RNF12 - Tempo Médio Entre Falhas (MTBF)	11
3.3.5	RNF13 - Tempo Médio de Reparo (MTTR)	11
3.4	Desempenho	11
3.4.1	RNF14 - Requisitos de Desempenho	11
3.4.2	RNF15 - Tempo de Resposta	11
3.4.3	RNF16 - Taxa de Transferência	11

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

3.4.4 RNF17 - Utilização de Recursos	12
3.5 Suportabilidade	12
3.5.1 RNF18 - Requisitos de Suportabilidade	12
3.5.2 RNF19 - Manutenção e Atualizações	12
3.5.3 RNF20 - Modularidade	12
3.5.4 RNF21 - Monitoramento e Logging	12
3.6 Restrições de Design	13
3.6.1 RNF22 - Requisitos de Design	13
3.6.2 RNF23 - Arquitetura de Software	13
3.6.3 RNF24 - Escolha de Tecnologias	13
3.6.4 RNF25 - Escalabilidade	13
3.6.5 RNF26 - Segurança desde o Design	13
3.7 Documentação do Usuário On-line e Requisitos do Sistema de Ajuda	13
3.8 Componentes Comprados	14
3.9 Interfaces	14
3.9.1 Interfaces com o Usuário	14
3.9.2 Interfaces de Hardware	14
3.9.3 Interfaces de Software	14
3.9.4 Interfaces de Comunicações	15
3.10 Requisitos de Licença	15
3.11 Observações Legais, sobre Direitos Autorais e Outras Observações	15
3.12 Padrões Aplicáveis	16
4. Especificação dos Casos de Uso	16
4.1 Diagrama de Casos de Uso	16
4.2 Documentos de Casos de Uso	18
5. Especificação das Classes	35
5.1 Diagrama de Classe	36
5.2 Explicação das Classes	37
6. Especificação dos Componentes	40
6.1 Diagrama de Componentes	40
6.2 Explicação dos Componentes	40
7. Especificação das Implementações	42
7.1 Diagrama de Implementação	42
7.2 Explicação das Implementações	42

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

Especificação de Requisitos de Software

1.Introdução

Nos últimos anos, o mercado pet tem experimentado um crescimento exponencial, impulsionado pela crescente humanização dos animais de estimação e pela demanda por produtos e serviços de alta qualidade. Em paralelo, a digitalização e a personalização têm se tornado pilares essenciais para o desenvolvimento de plataformas que visam atender as necessidades específicas dos consumidores. Nesse contexto, surge a PetPlace, uma plataforma inovadora projetada para transformar a maneira como tutores de pets, veterinários e fornecedores de produtos interagem e fazem negócios.

Pet Place é um marketplace de serviços e produtos veterinários com personalização avançada, que utiliza tecnologias de ponta, como machine learning e inteligência artificial, para oferecer uma experiência única e personalizada para cada usuário. A plataforma conecta tutores de animais a uma vasta rede de profissionais e fornecedores, oferecendo recomendações personalizadas que vão desde alimentos específicos para as necessidades nutricionais de um pet até serviços de saúde e bem-estar, como consultas veterinárias e adestramento.

1.1.Objetivo

O objetivo deste documento de Especificação de Requisitos de Software (SRS) é descrever detalhadamente os requisitos funcionais e não funcionais para o desenvolvimento do Marketplace de Serviços e Produtos Veterinários com Personalização Avançada da VVet Tech. Este documento é destinado a fornecer uma base sólida para o design, desenvolvimento e teste do sistema.

1.2.Escopo

A SRS abrange o desenvolvimento de uma plataforma online que conectará tutores de pets a profissionais e fornecedores de produtos e serviços veterinários. A plataforma incluirá funcionalidades como cadastro de usuários, cadastro de pets, algoritmos de recomendação personalizados, gestão de compras, entre outras funcionalidades descritas neste documento. O sistema será desenvolvido para atender às necessidades específicas do mercado de saúde animal, conforme detalhado na seção de Requisitos Específicos.

1.3.Definições, Acrônimos e Abreviações

SRS: Especificação de Requisitos de Software.

PetPlace: Nome do projeto/plataforma desenvolvida.

ML: Machine Learning (Aprendizado de Máquina).

UX: User Experience (Experiência do Usuário).

UI: User Interface (Interface do Usuário).

API: Application Programming Interface (Interface de Programação de Aplicações).

JSON: JavaScript Object Notation, um formato de intercâmbio de dados.

JWT: JSON Web Token, um padrão para troca de informações de forma segura.

CI/CD: Continuous Integration/Continuous Deployment (Integração Contínua/Entrega Contínua).

LGPD: Lei Geral de Proteção de Dados, uma lei brasileira de proteção de dados pessoais.

GDPR: General Data Protection Regulation, regulamento de proteção de dados da União Europeia.

CDN: Content Delivery Network (Rede de Entrega de Conteúdo).

MQTT: Message Queuing Telemetry Transport, um protocolo leve de mensagens para a comunicação de dispositivos IoT.

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

REST: Representational State Transfer, um estilo arquitetural para web services.

AWS: Amazon Web Services, um provedor de serviços em nuvem.

1.4.Referências

- Fonte: "ISO/IEC/IEEE 29148:2018 - Systems and software engineering — Life cycle processes — Requirements engineering" fornece diretrizes para a especificação de requisitos de software e sistemas.
- Fonte: "ISO/IEC 25010:2011 - Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models".
- Fonte: "Market Trends: Global Pet Care Market Size, Growth, and Industry Analysis, 2023-2030" fornece insights sobre as tendências do mercado pet global.
- Fonte: "Pet Care and Veterinary Services Market Analysis: Digital Transformation and Emerging Technologies" aborda como a transformação digital está impactando o mercado de serviços veterinários.
- Fonte: "User Experience and Usability in Complex Systems: Approaches to Improving User Interfaces" explora as melhores práticas para garantir interfaces amigáveis e acessíveis.
- Fonte: "ISO/IEC/IEEE 42010:2011 - Systems and software engineering — Architecture description" discute a importância de definir a arquitetura de software de forma modular e escalável.
- Fonte: "Machine Learning for Recommender Systems: Foundations and Applications" de Charu Aggarwal fornece uma visão sobre como implementar sistemas de recomendação baseados em machine learning.
- Fonte: "Human-Computer Interaction: Theory, Design, Development and Practice" por Julie A. Jacko, que discute os princípios de design de interfaces intuitivas e a experiência do usuário.
- Fonte: "Software Reliability Engineering: More Reliable Software, Faster Development and Testing" por John D. Musa, que aborda a engenharia de confiabilidade para sistemas de software.
- Fonte: "Performance Engineering: State of the Art and Current Trends" oferece insights sobre as melhores práticas para garantir o desempenho de sistemas de software.
- Fonte: "Maintenance of Software" por Gerald J. Sussman e Jack Wisdom, que discutem as práticas para garantir a facilidade de manutenção de sistemas de software.
- Fonte: "Design Patterns: Elements of Reusable Object-Oriented Software" por Erich Gamma, que discute padrões de design que podem ser usados para criar uma arquitetura modular e escalável.
- Fonte: "The Handbook of Technical Writing" por Gerald J. Alred, que oferece diretrizes sobre como criar documentações técnicas de qualidade.
- Fonte: "Integrating Third-Party Components in Software Product Development" por Subashini Ganapathy, que discute os desafios e soluções na integração de componentes de terceiros em sistemas de software.
- Fonte: "Software Interfaces and APIs: The Art of Building Modular and Extensible Software" por Roman Zhukov, que aborda as melhores práticas para definir interfaces de software.
- Fonte: "Intellectual Property in Software Development: Understanding Legal Issues and Licensing" por Daniel C. Holbrook, que oferece orientações sobre como gerenciar licenças de software.
- Fonte: "Legal Aspects of Software Licensing and Compliance" fornece uma visão abrangente sobre questões legais relacionadas a software.
- Fonte: "ISO/IEC/IEEE 12207:2017 - Systems and software engineering — Software life cycle processes" que detalha os processos de ciclo de vida de software.

1.5.Visão Geral

Este documento está estruturado para fornecer uma visão abrangente dos requisitos do sistema. A seção 2 descreve a visão geral do sistema, enquanto a seção 3 especifica os requisitos detalhados, tanto funcionais quanto não funcionais.

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

2.Descrição Geral

2.1.Perspectiva do Produto

O marketplace proposto será uma solução inovadora voltada para o mercado veterinário, utilizando tecnologias como machine learning para oferecer recomendações personalizadas de produtos e serviços veterinários. O sistema será acessível através de dispositivos móveis e desktops, permitindo fácil navegação e compra de produtos e serviços.

2.2.Funções do Produto

As principais funcionalidades incluem:

- Cadastro de usuários e pets
- Algoritmos de recomendação personalizados
- Gestão de carrinho de compras, pedidos, entregas e endereços
- Agendamento de serviços veterinários
- Pagamento online
- Sistema de chat e suporte para resolução de problemas

2.3.Características do Usuário

O sistema será utilizado por tutores de pets, veterinários e fornecedores de produtos e serviços veterinários. Cada grupo de usuários terá funcionalidades específicas alinhadas às suas necessidades.

2.4.Restrições

O sistema deve ser compatível com as regulamentações de proteção de dados, como a LGPD, e funcionar em dispositivos móveis e desktops.

2.5.Premissas e Dependências

O desenvolvimento do sistema dependerá da disponibilidade de dados precisos para treinar os algoritmos de recomendação e da integração com sistemas de pagamento e entrega.

3.Requisitos Específicos

3.1.Funcionalidade

3.1.1.RF01 - Cadastro de Usuários

A plataforma deve permitir o cadastro de tutores, veterinários e fornecedores de produtos e serviços veterinários.

O cadastro deve incluir informações básicas como nome, e-mail, telefone, endereço e informações específicas relacionadas ao perfil (ex.: tipo de profissional, especialidades, produtos oferecidos).

3.1.2.RF02 - Login de Usuário

A plataforma deve permitir que os usuários façam login utilizando credenciais seguras (e-mail e senha), além de oferecer a opção de login via redes sociais ou autenticação de dois fatores para maior segurança.

3.1.3.RF03 - Cadastro de Pets

A plataforma deve permitir que os tutores cadastrem seus pets, incluindo informações como nome, espécie, raça, idade, peso, histórico de saúde e preferências alimentares.

3.1.4.RF04 - Catalogar Produtos e Serviços

O sistema deve permitir que fornecedores e veterinários cadastrem e cataloguem seus produtos e serviços na plataforma, incluindo detalhes como nome, descrição, preço, imagens, e especificações técnicas ou características relevantes.

3.1.5.RF05 - Pesquisa e Navegação

O sistema deve permitir que os usuários busquem por produtos e serviços veterinários utilizando

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

filtros como categoria, localização, preço, avaliação e necessidades específicas do pet.

3.1.6.RF06 - Algoritmo de Recomendação (Anúncio)

A plataforma deve utilizar algoritmos de machine learning para recomendar produtos e serviços personalizados para cada pet, com base em seu histórico de saúde, comportamento e preferências alimentares.

3.1.7.RF07 - Integração com Histórico de Saúde

O sistema deve integrar-se com prontuários eletrônicos para acessar o histórico de saúde dos pets, garantindo que as recomendações sejam precisas e alinhadas às necessidades dos animais.

3.1.8.RF08 - Gestão de Carrinho de Compras

A plataforma deve permitir que os usuários adicionem, removam e editem produtos e serviços em um carrinho de compras. Deve também calcular automaticamente o total da compra, incluindo impostos e taxas de entrega.

3.1.9.RF09 - Gestão de Pedidos

O sistema deve permitir que os usuários finalizem seus pedidos, incluindo a confirmação de pagamento, visualização de detalhes do pedido, status de processamento, e histórico de compras.

3.1.10.RF10 - Gestão de Entrega

A plataforma deve permitir que os usuários escolham métodos de entrega, insiram endereços de entrega, e rastreiem a entrega de seus produtos em tempo real. O sistema deve também gerenciar as opções de entrega disponíveis com base na localização do usuário.

3.1.11.RF11 - Gestão de Endereços

O sistema deve permitir que o usuário cadastre, edite e gerencie múltiplos endereços de entrega para facilitar o processo de compra e entrega de produtos.

3.1.12.RF12 - Agendamento de Serviços

A plataforma deve permitir que os usuários agendem serviços como consultas veterinárias, fisioterapia, adestramento, entre outros, diretamente através da interface do marketplace.

3.1.13.RF13 - Pagamento Online

O sistema deve permitir o pagamento online para produtos e serviços adquiridos, suportando múltiplas formas de pagamento, incluindo cartão de crédito, débito e PIX.

3.1.14.RF14 - Chat

A plataforma deve oferecer uma funcionalidade de chat que permita a comunicação em tempo real entre tutores, veterinários e fornecedores de serviços/produtos, para tirar dúvidas e realizar consultas rápidas.

3.1.15.RF15 - Resolução de Problemas

O sistema deve incluir um módulo de suporte para a resolução de problemas, onde os usuários podem reportar questões relacionadas a produtos, serviços ou entregas, e receber suporte até a resolução completa.

3.1.16.RF16 - Avaliação e Feedback

A plataforma deve permitir que os usuários avaliem e deixem feedback sobre os produtos e serviços adquiridos, ajudando outros usuários na tomada de decisão.

3.1.17.RF17 - Notificações Personalizadas

O sistema deve enviar notificações aos tutores sobre recomendações de produtos, lembretes de consultas e promoções, com base no perfil e comportamento do pet.

3.2. Utilidade

3.2.1. RNF01 - Usabilidade

- *A interface deve ser intuitiva e acessível, garantindo uma experiência de usuário positiva e eficiente.*
 - Design Responsivo: Utilizar frameworks de design responsivo como Bootstrap ou Tailwind CSS para garantir que a interface funcione bem em dispositivos móveis, tablets e desktops.

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

- Acessibilidade: Implementar práticas de acessibilidade de acordo com as diretrizes do WCAG (Web Content Accessibility Guidelines), incluindo suporte a leitores de tela, navegação por teclado, e contrastes de cores adequados.
- Testes de Usabilidade: Conduzir testes de usabilidade com usuários finais utilizando ferramentas como UserTesting ou Hotjar para identificar e corrigir problemas na interface.
- Interface de Usuário Consistente: Utilizar um sistema de design, como Material Design ou Design System da Atlassian, para manter a consistência visual e funcional da interface.

3.2.2. RNF02 - Compreensibilidade

- Os usuários devem ser capazes de entender facilmente o propósito do sistema e como utilizar suas funcionalidades, sem a necessidade de um extenso treinamento.
 - Design Intuitivo: Adotar um design de interface que siga padrões amplamente reconhecidos e intuitivos, como Material Design ou Human Interface Guidelines da Apple.
 - Uso de Ícones e Rótulos Claros: Implementar ícones e rótulos que sejam autoexplicativos, usando linguagem clara e direta para descrever funções e ações.
 - Feedback Visual e Auditivo: Fornecer feedback imediato e compreensível para as ações do usuário, como animações de carregamento, confirmações de ações, e alertas de erro, para ajudar o usuário a entender o estado atual do sistema.

3.2.3. RNF03 - Aprendibilidade

- O sistema deve ser fácil de aprender para novos usuários, permitindo que eles se tornem produtivos em pouco tempo.
 - Onboarding Interativo: Implementar um processo de onboarding que guie os novos usuários através das principais funcionalidades, utilizando tutoriais interativos ou walkthroughs.
 - Ajuda Contextual: Fornecer ajuda contextual diretamente na interface, como tooltips, dicas, ou um botão de "Ajuda" que explique as funcionalidades sem a necessidade de sair da tela atual.
 - Documentação e Recursos de Treinamento: Disponibilizar tutoriais, vídeos, e FAQs acessíveis diretamente a partir do sistema, garantindo que os usuários possam encontrar respostas rápidas para suas dúvidas.

3.2.4. RNF04 - Operabilidade

- O sistema deve ser fácil de operar, permitindo que os usuários executem suas tarefas de maneira eficaz e eficiente.
 - Navegação Simples: Estruturar a navegação de maneira lógica e intuitiva, utilizando menus claros e hierarquias de navegação que reflitam a forma como os usuários pensam em suas tarefas.
 - Fluxos de Trabalho Otimizados: Projetar fluxos de trabalho que minimizem o número de cliques e etapas necessários para completar tarefas comuns, garantindo que as ações mais frequentes estejam acessíveis.
 - Teclas de Atalho e Gestos: Implementar atalhos de teclado e gestos que permitam aos usuários avançados realizar operações comuns mais rapidamente, aumentando a eficiência.

3.2.5. RNF05 - Atração

- A interface do sistema deve ser atraente e agradável de usar, incentivando o uso contínuo e reduzindo a fadiga do usuário.
 - Design Visual Agradável: Utilizar princípios de design gráfico para criar uma interface visualmente atraente, com uma paleta de cores harmoniosa, tipografia legível, e layout equilibrado.
 - Consistência Visual: Garantir a consistência no design visual em toda a aplicação, incluindo cores, fontes, ícones e espaçamento, para criar uma experiência coesa e

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

profissional.

- Personalização da Interface: Permitir que os usuários personalizem aspectos da interface, como temas de cores ou layouts, para que possam adaptá-la ao seu gosto pessoal.

3.2.6. RNF06 - Proteção Contra Erros de Usuário

- O sistema deve prevenir erros de usuário sempre que possível e fornecer maneiras eficazes de corrigi-los caso ocorram.
 - Validação de Entrada de Dados: Implementar validação de dados em tempo real nos formulários para prevenir a entrada de dados incorretos, utilizando mensagens de erro claras que orientem o usuário sobre como corrigir o problema.
 - Confirmações para Ações Críticas: Solicitar confirmação do usuário antes de executar ações que possam ter consequências significativas, como deletar dados ou enviar informações importantes.
 - Desfazer e Refazer: Oferecer funcionalidades de desfazer e refazer para permitir que os usuários revertam facilmente ações acidentais.

3.2.7. RNF07 - Eficiência de Uso

- O sistema deve permitir que os usuários realizem tarefas com o mínimo de esforço e tempo, maximizando a produtividade.
 - Desempenho Rápido: Garantir que o sistema responda rapidamente às entradas do usuário, utilizando técnicas de otimização como cacheamento e pré-carregamento de dados.
 - Automatização de Tarefas Repetitivas: Implementar automações para tarefas comuns que podem ser repetitivas ou demoradas, como preenchimento automático de formulários ou sugestões de ações com base no histórico do usuário.
 - Navegação Rápida: Implementar navegação por abas, janelas modais, ou menus suspensos para permitir que os usuários acessem rapidamente diferentes partes do sistema sem interromper seu fluxo de trabalho.

3.2.8. RNF08 - Satisfação do Usuário

- O sistema deve proporcionar uma experiência de uso agradável e satisfatória, promovendo uma sensação positiva no usuário.
 - Testes de Satisfação: Conduzir pesquisas de satisfação do usuário regularmente para coletar feedback e identificar áreas de melhoria na interface e na experiência do usuário.
 - Animações e Efeitos de Transição: Usar animações suaves e transições naturais para tornar as interações mais agradáveis, desde que não comprometam o desempenho do sistema.
 - Gamificação: Considerar a inclusão de elementos de gamificação, como recompensas ou feedback positivo para a conclusão de tarefas, para aumentar o engajamento do usuário.

3.3. Confiabilidade

3.3.1. RNF09 - Requisitos de Confiabilidade

- A confiabilidade do sistema deve ser alta, garantindo que ele funcione corretamente e esteja disponível para os usuários em praticamente todos os momentos. Isso inclui resistência a falhas e capacidade de recuperação rápida.
 - Alta Disponibilidade: Implementar uma arquitetura de alta disponibilidade com servidores distribuídos em várias zonas de disponibilidade (AZs) usando provedores como AWS ou Google Cloud.
 - Monitoramento de Saúde: Utilizar ferramentas de monitoramento como Datadog ou Prometheus para monitorar a saúde do sistema em tempo real, detectando falhas antes que afetem os usuários.
 - MTBF e MTTR: Estabelecer métricas de tempo médio entre falhas (MTBF) e tempo médio de reparo (MTTR), com metas de pelo menos 99,9% de uptime e recuperação em menos de 2 horas.

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

- Redundância: Implementar redundância em componentes críticos do sistema, como servidores de banco de dados e serviços de backend, para garantir continuidade do serviço em caso de falhas.

3.3.2. RNF10 - Disponibilidade

- O sistema deve estar disponível para os usuários na maior parte do tempo, com o mínimo de interrupções.
 - Arquitetura de Alta Disponibilidade: Utilizar uma arquitetura de alta disponibilidade, distribuindo serviços em múltiplas zonas de disponibilidade (AZs) e utilizando balanceadores de carga.
 - Monitoramento Proativo: Implementar ferramentas de monitoramento, como Nagios ou Prometheus, para monitorar a saúde dos sistemas em tempo real e alertar sobre problemas antes que eles afetem os usuários.
 - Failover Automático: Configurar mecanismos de failover automático para serviços críticos, garantindo que o sistema continue operando em caso de falhas.

3.3.3. RNF11 - Tolerância a Falhas

- O sistema deve ser capaz de continuar funcionando corretamente, mesmo na presença de falhas em alguns de seus componentes.
 - Redundância de Componentes: Implementar redundância em componentes críticos, como servidores de banco de dados e servidores de aplicação.
 - Replicação de Dados: Utilizar replicação de dados síncrona ou assíncrona para garantir que dados críticos estejam disponíveis em múltiplos locais.
 - Testes de Resiliência: Realizar testes de resiliência, como injeção de falhas (usando ferramentas como Chaos Monkey) para simular falhas e verificar como o sistema responde.

3.3.4. RNF12 - Tempo Médio Entre Falhas (MTBF)

- O MTBF deve ser o mais longo possível, indicando que o sistema pode operar por um longo período antes de ocorrer uma falha.
 - Manutenção Preventiva: Implementar práticas de manutenção preventiva para identificar e corrigir potenciais problemas antes que causem falhas.
 - Qualidade de Código: Garantir que o código seja de alta qualidade, utilizando práticas de desenvolvimento como revisão de código e testes rigorosos.

3.3.5. RNF13 - Tempo Médio de Reparo (MTTR)

- O MTTR deve ser o menor possível, garantindo que falhas sejam corrigidas rapidamente.
 - Automação de Recuperação: Utilizar scripts e automações para recuperar sistemas falhos o mais rápido possível.
 - Equipe de Suporte 24/7: Garantir que haja uma equipe de suporte disponível 24/7 para lidar com problemas críticos.

3.4. Desempenho

3.4.1. RNF14 - Requisitos de Desempenho

- O tempo de resposta do sistema deve ser inferior a 2 segundos em 95% das interações, com capacidade para atender até 10.000 usuários simultâneos.
 - Escalabilidade Horizontal: Utilizar uma arquitetura que suporte escalabilidade horizontal, permitindo adicionar mais instâncias de servidor conforme a demanda aumenta. Ferramentas como Kubernetes podem ser usadas para orquestrar os contêineres.
 - Otimização de Consultas: Otimizar as consultas ao banco de dados, utilizando índices e consultas pré-compiladas (prepared statements) para reduzir a latência.
 - Cacheamento: Implementar cacheamento de dados frequentemente acessados usando Redis ou Memcached para diminuir a carga no banco de dados e acelerar o tempo de resposta.

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

- Testes de Carga: Realizar testes de carga e estresse utilizando ferramentas como Apache JMeter ou Gatling para simular diferentes cenários de uso e identificar possíveis gargalos.

3.4.2. RNF15 - Tempo de Resposta

- O sistema deve responder rapidamente às solicitações dos usuários, com tempos de resposta que atendam aos requisitos do projeto.
 - Otimização de Consultas: Otimizar as consultas ao banco de dados, utilizando índices adequados e procedimentos armazenados para melhorar o desempenho.
 - Cacheamento de Dados: Implementar cacheamento para dados frequentemente acessados usando Redis ou Memcached para reduzir a latência.
 - Minificação de Recursos: Minificar arquivos CSS, JavaScript e imagens para reduzir o tempo de carregamento das páginas.

3.4.3. RNF16 - Taxa de Transferência

- O sistema deve ser capaz de processar um grande número de transações por segundo (TPS) ou gigabytes por segundo (GB/s), dependendo do contexto.
 - Escalabilidade Horizontal: Adotar uma arquitetura que permita a adição de novos servidores conforme necessário para lidar com o aumento da carga.
 - Balanceamento de Carga: Utilizar balanceadores de carga como NGINX ou HAProxy para distribuir a carga de trabalho entre vários servidores.
 - Otimização de Rede: Implementar uma Content Delivery Network (CDN) para distribuir o tráfego e acelerar o acesso ao conteúdo.

3.4.4. RNF17 - Utilização de Recursos

- O sistema deve utilizar os recursos computacionais de forma eficiente, evitando sobrecargas e desperdícios.
 - Monitoramento de Recursos: Utilizar ferramentas como New Relic ou Datadog para monitorar o uso de CPU, memória e disco, e identificar gargalos.
 - Ajuste de Performance: Realizar tuning de desempenho em servidores de banco de dados e aplicação para otimizar o uso de recursos.
 - Profiling de Aplicação: Usar ferramentas de profiling como VisualVM ou Py-Spy para identificar e otimizar partes do código que consomem mais recursos.

3.5. Suportabilidade

3.5.1. RNF18 - Requisitos de Suportabilidade

O sistema deve ser facilmente mantido e atualizado, com suporte a monitoramento contínuo e integração de novos módulos sem interrupção.

3.5.2. RNF19 - Manutenção e Atualizações

- O sistema deve permitir atualizações e manutenção sem interrupções significativas no serviço.
 - CI/CD Pipeline: Implementar um pipeline de integração e entrega contínua (CI/CD) utilizando ferramentas como Jenkins, GitLab CI, ou CircleCI para automatizar o deploy de novas versões.
 - Deploys sem Downtime: Utilizar técnicas de deploy sem downtime, como Blue-Green Deployment ou Canary Releases, para introduzir atualizações sem afetar os usuários.
 - Documentação de Manutenção: Manter uma documentação completa de manutenção que cubra procedimentos para aplicar patches e realizar upgrades.

3.5.3. RNF20 - Modularidade

- O sistema deve ser modular, permitindo que componentes individuais sejam modificados ou substituídos sem impactar o sistema como um todo.
 - Arquitetura de Microserviços: Adotar uma arquitetura de microserviços onde cada serviço é autônomo e pode ser desenvolvido, implantado e escalado independentemente.
 - Interface de APIs Estável: Garantir que as interfaces de APIs sejam bem definidas e

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

estáveis, permitindo a modificação interna dos módulos sem impactar as integrações externas.

- Padrões de Design: Utilizar padrões de design como Factory ou Strategy para promover a modularidade e facilitar futuras modificações.

3.5.4. RNF21 - Monitoramento e Logging

- O sistema deve ter um monitoramento robusto e logging para facilitar a detecção e correção de problemas.
 - Logging Centralizado: Implementar um sistema de logging centralizado usando ELK Stack (Elasticsearch, Logstash, Kibana) ou Splunk para coletar, armazenar e analisar logs.
 - Alertas Proativos: Configurar alertas automáticos para eventos críticos ou comportamento anômalo utilizando Prometheus ou Grafana.
 - Painéis de Monitoramento: Criar painéis de monitoramento com Grafana para visualizar métricas em tempo real e facilitar o diagnóstico de problemas.

3.6. Restrições de Design

3.6.1. RNF22 - Requisitos de Design

O sistema deve ser desenvolvido utilizando uma arquitetura modular, suportando escalabilidade e integração com APIs externas.

3.6.2. RNF23 - Arquitetura de Software

- O sistema deve ter uma arquitetura que suporte escalabilidade, flexibilidade e manutenção.
 - Arquitetura de Microserviços: Utilizar uma arquitetura de microserviços para permitir o desenvolvimento e a implantação independentes de diferentes partes do sistema.
 - Desacoplamento e Interfaces Bem Definidas: Assegurar que os componentes do sistema sejam desacoplados e se comuniquem por meio de interfaces bem definidas (por exemplo, APIs RESTful).
 - Camadas de Abstração: Implementar camadas de abstração para isolar dependências e facilitar a substituição de componentes ou tecnologias sem impactar o restante do sistema.

3.6.3. RNF24 - Escolha de Tecnologias

- As tecnologias utilizadas no sistema devem ser escolhidas com base em sua adequação ao projeto e suporte a longo prazo.
 - Linguagens e Frameworks Consolidados: Utilizar linguagens de programação e frameworks amplamente adotados e com forte comunidade de suporte, como JavaScript/TypeScript para frontend (React ou Angular) e Python ou Node.js para backend.
 - Padrões de Mercado: Seguir padrões de mercado, como OAuth 2.0 para autenticação e JWT para tokens de sessão, para garantir interoperabilidade e segurança.
 - Ferramentas de Desenvolvimento: Definir um conjunto de ferramentas de desenvolvimento padrão, como Visual Studio Code para IDE, Git para controle de versão, e Docker para containerização.

3.6.4. RNF25 - Escalabilidade

- O design do sistema deve suportar escalabilidade, tanto horizontal quanto vertical.
 - Design para Escalabilidade Horizontal: Estruturar o sistema de forma que ele possa escalar horizontalmente, adicionando mais servidores ou instâncias conforme necessário.
 - Balanceamento de Carga: Implementar balanceamento de carga para distribuir o tráfego entre múltiplos servidores, garantindo que nenhum servidor individual se torne um gargalo.
 - Distribuição de Dados: Utilizar técnicas de sharding ou particionamento de dados para garantir que o banco de dados possa crescer conforme necessário sem comprometer o desempenho.

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

3.6.5. RNF26 - Segurança desde o Design

- A segurança deve ser integrada desde o início do design do sistema.
 - Princípios de Segurança por Design: Aplicar princípios de segurança por design, como o princípio do menor privilégio, proteção por padrão e defesa em profundidade.
 - Criptografia de Dados: Incorporar criptografia de dados em repouso e em trânsito desde o início do projeto, utilizando padrões como AES-256 para dados em repouso e TLS para dados em trânsito.
 - Testes de Segurança: Realizar testes de segurança como análise de vulnerabilidades e testes de penetração regularmente para identificar e mitigar riscos de segurança.

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

A documentação do usuário e sistemas de ajuda devem estar disponíveis online, acessíveis diretamente pela interface da plataforma.

3.8.Componentes Comprados

A plataforma pode integrar componentes comprados, como gateways de pagamento e APIs de rastreamento de entregas, desde que atendam aos requisitos de segurança e desempenho.

3.9.Interfaces

3.9.1. Interfaces com o Usuário

- A interface deve ser responsiva, garantindo uma experiência consistente em dispositivos móveis e desktops.
 - Tipo de Interface: Interface gráfica do usuário (GUI).
 - Tecnologias Utilizadas: A interface será desenvolvida utilizando HTML5, CSS3, JavaScript/TypeScript com frameworks como React ou Angular.
 - Design Responsivo: A interface será responsiva, garantindo compatibilidade com desktops, tablets, e smartphones. Isso será alcançado utilizando frameworks de design responsivo como Bootstrap ou Tailwind CSS.
 - Acessibilidade: A interface deverá seguir as diretrizes WCAG 2.1 (Web Content Accessibility Guidelines) para garantir a acessibilidade para usuários com deficiências, incluindo suporte a leitores de tela e navegação por teclado.
 - Navegação: O sistema terá uma navegação intuitiva com menus claros, breadcrumbs e uma barra de pesquisa acessível em todas as páginas.
 - Elementos de Interface: Todos os elementos da interface, como botões, formulários, e tabelas, serão estilizados de forma consistente, seguindo o sistema de design Material Design.
 - Suporte a Múltiplos Idiomas: A interface deve suportar múltiplos idiomas, com tradução dinâmica das strings de interface utilizando i18next ou React Intl.

3.9.2. Interfaces de Hardware

- O sistema deve suportar integração com dispositivos IoT relacionados à saúde dos pets.
 - Dispositivos Suportados: O sistema suportará dispositivos IoT para monitoramento de saúde animal, como coleiras inteligentes e sensores de ambiente.
 - Protocolo de Comunicação: Será utilizado o protocolo MQTT para comunicação entre os dispositivos IoT e o backend do sistema, devido à sua eficiência em cenários de baixa largura de banda.
 - Endereçamento: Os dispositivos serão identificados e autenticados por meio de endereços MAC e certificados digitais, garantindo uma comunicação segura.
 - Drivers de Hardware: O sistema deverá incluir drivers específicos para integração com sensores de temperatura, frequência cardíaca e GPS. Esses drivers serão desenvolvidos ou integrados utilizando SDKs fornecidos pelos fabricantes dos

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

dispositivos.

- Comportamento Esperado: Os dispositivos de hardware devem enviar dados em intervalos regulares ou em resposta a eventos específicos (por exemplo, detecção de anomalias). O sistema processará esses dados em tempo real, armazenando-os no banco de dados e notificando os usuários conforme necessário.

3.9.3. Interfaces de Software

- Deve haver suporte para integração com sistemas de pagamento, histórico de saúde e logística.
 - APIs RESTful: O sistema exporá e consumirá APIs RESTful para comunicação entre os diferentes microserviços. As APIs serão documentadas utilizando Swagger/OpenAPI.
 - Autenticação e Autorização: A comunicação com as APIs será segura, utilizando OAuth 2.0 para autenticação e tokens JWT (JSON Web Tokens) para autorização.
 - Integração com Sistemas de Pagamento: O sistema integrará com provedores de pagamento como Stripe ou PayPal. A interface será projetada para suportar chamadas assíncronas e manuseio de callbacks de pagamento.
 - Componentes de Terceiros: O sistema utilizará componentes de terceiros para funcionalidades específicas, como Twilio para envio de mensagens SMS e AWS S3 para armazenamento de arquivos. As integrações serão feitas via SDKs fornecidos pelos provedores.
 - Logs e Monitoramento: Os logs de API serão centralizados usando ELK Stack (Elasticsearch, Logstash, Kibana), permitindo análise em tempo real e identificação de problemas de integração.

3.9.4. Interfaces de Comunicações

- O sistema deve suportar protocolos seguros de comunicação entre clientes e servidores.
 - Protocolos de Comunicação: O sistema suportará comunicação por HTTPS para todas as interações via internet, garantindo segurança com criptografia TLS 1.3.
 - Portas Utilizadas: As comunicações padrão utilizarão a porta 443 para HTTPS. Dispositivos IoT poderão se comunicar através da porta 8883, que é utilizada para MQTT sobre TLS.
 - Rede Local: O sistema permitirá a comunicação com dispositivos em uma rede local via TCP/IP, utilizando endereços IP estáticos ou dinâmicos (via DHCP) conforme necessário.
 - Comunicação Assíncrona: Para garantir alta disponibilidade e resposta rápida, a comunicação entre serviços poderá utilizar filas de mensagens como RabbitMQ ou Kafka, especialmente para processamento de tarefas em segundo plano ou integração de sistemas.
 - Segurança de Rede: Implementação de firewalls e políticas de segurança de rede para proteger a comunicação entre servidores e dispositivos.

3.10.Requisitos de Licença

- Licenciamento de Componentes de Terceiros: Todos os componentes de terceiros utilizados no sistema, como bibliotecas, frameworks, e SDKs, deverão ser licenciados conforme as exigências das suas respectivas licenças (por exemplo, MIT, Apache 2.0, GPL).
- Controle de Licenças: O sistema deverá incluir uma lista completa de todas as licenças de software utilizadas, monitorando a conformidade com as mesmas utilizando ferramentas como FOSSA ou WhiteSource.
- Restrições de Uso: O software final deverá ter uma licença de uso que permita sua utilização em conformidade com os termos acordados (por exemplo, licença comercial, licença para uso interno, etc.). Qualquer uso além do escopo autorizado pela licença será considerado uma violação.
- Licença de Usuário Final (EULA): Um contrato de licença de usuário final deve ser apresentado e aceito pelos usuários antes da instalação ou uso do software. Esse contrato deve detalhar os direitos e responsabilidades do usuário em relação ao software.

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

3.11. Observações Legais, sobre Direitos Autorais e Outras Observações

- **Direitos Autorais:** Todo o código desenvolvido internamente será protegido por direitos autorais, com um aviso de copyright incluído no início de cada arquivo de código-fonte, conforme as práticas recomendadas.
- **Marcas Registradas:** Quaisquer logotipos, nomes comerciais ou outros identificadores associados ao software que estejam protegidos por marcas registradas devem ser identificados e utilizados de acordo com as diretrizes de marca.
- **Isonções e Garantias:** O software será fornecido "como está", sem garantias expressas ou implícitas, exceto aquelas estipuladas por lei. A isenção de responsabilidade será incluída nos contratos de licença e na documentação.
- **Conformidade com Leis Locais:** O software deverá ser compatível com todas as leis de propriedade intelectual e proteção de dados aplicáveis nas jurisdições onde for comercializado, incluindo conformidade com a LGPD (Lei Geral de Proteção de Dados) no Brasil e GDPR (General Data Protection Regulation) na União Europeia.
- **Avisos Legais:** Todos os avisos legais relevantes, incluindo informações sobre licenciamento, copyright, e direitos de patente, devem ser exibidos ao usuário final, geralmente durante a instalação do software ou no início de sua utilização.

3.12. Padrões Aplicáveis

- **ISO/IEC 25000 (SQuaRE):** O software será desenvolvido em conformidade com a norma ISO/IEC 25000, garantindo que os requisitos de qualidade, usabilidade, confiabilidade, e desempenho sejam atendidos.
- **ISO/IEC 27001:** Para garantir a segurança das informações, o sistema deve seguir as diretrizes da norma ISO/IEC 27001 para gerenciamento de segurança da informação.
- **OWASP Top 10:** O desenvolvimento do software seguirá as práticas de segurança recomendadas pela OWASP Top 10, para mitigar os riscos mais comuns de segurança.

4. Especificação dos Casos de Uso

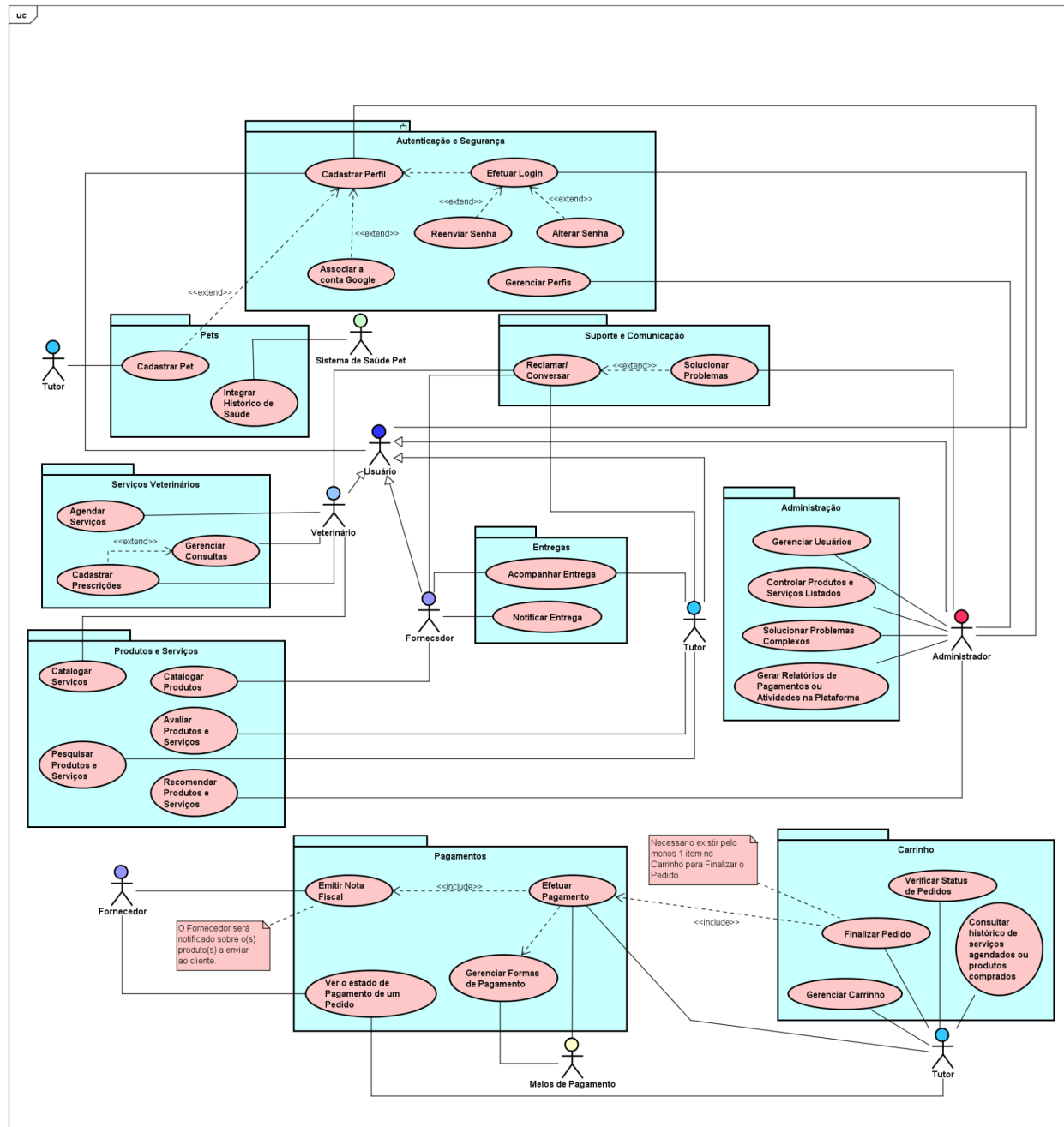
Este Capítulo do Documento de Especificação de Requisitos irá detalhar por intermédio da UML 2.5 as funcionalidades descritas sob o Capítulo 3, na Seção de Requisitos Funcionais. Entende-se que o conjunto dos Requisitos Funcionais do Software estão contemplados no Diagrama de Casos de Uso, identificado de forma completa todos os Atores que interagem com o Software a ser desenvolvido no projeto.

4.1. Diagrama de Casos de Uso

Incluir nesta seção o desenho do Diagrama de Casos de Uso, elaborado com o apoio de uma ferramenta de Modelagem como o Astah ou o Draw.io, de acordo com a sintaxe e a semântica da UML 2.5.

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

Figura 1 - Diagrama de Casos de Uso do MarketPlace da PetPlace



Fonte: autoria própria (2024).

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

4.2. Documentos de Casos de Uso

Quadro 1 - UC01 - Cadastrar Perfil

Caso de Uso: UC01 – Cadastrar Perfil	
Resumo: Este caso de uso permite que o Administrador do sistema registre novos perfis de usuários, incluindo dados cadastrais básicos (nome, e-mail, telefone) e configurações de permissões de acesso. O objetivo é garantir que cada usuário tenha uma identificação única no sistema e acesso às funcionalidades permitidas para seu perfil.	
Atores Participantes: Usuário (Tutor, Veterinário, Fornecedor, Administrador)	
Casos de Uso Relacionados: UC02 - Efetuar Login (para que o usuário possa acessar o sistema após o cadastro). UC06 - Gerenciar Perfis (para que o administrador possa posteriormente alterar ou remover perfis cadastrados).	
Pré-condição: O Administrador deve estar autenticado no sistema e ter privilégios de gerenciamento de perfis de usuário. É necessário que o sistema possua validação de dados para evitar duplicidade de informações e garantir a segurança.	
Fluxo Principal:	
Ações do Ator:	Ações do Sistema:
1. O Administrador acessa a interface de cadastro de usuários.	3. O sistema verifica se o e-mail já está cadastrado. Caso o e-mail seja único, prossegue com a verificação do telefone.
2. Informa os dados cadastrais do usuário: nome, endereço de e-mail, telefone, e tipo de perfil de acesso (por exemplo, Tutor, Veterinário, Fornecedor, Administrador).	4. O sistema atribui uma identificação única ao novo usuário e gera uma senha temporária para o primeiro acesso.
	5. O sistema confirma o cadastro e envia uma notificação para o e-mail registrado com as instruções de acesso.
Fluxo Alternativo:	
A1 - Usuário já cadastrado (Alteração de dados):	
1. Se o usuário já está cadastrado, o sistema permite ao administrador consultar o perfil existente e exibir os dados atuais.	
2. O administrador pode optar por alterar dados como nome, e-mail ou tipo de perfil de acesso, conforme necessário.	
3. O sistema salva as alterações e notifica o administrador do sucesso na operação.	
Fluxo de Exceção:	
E1 - E-mail Inválido:	
1. Caso o formato do e-mail informado não esteja de acordo com o padrão esperado (por exemplo, não possuir um "@" ou um domínio válido), o sistema exibe uma mensagem de erro: "Endereço de e-mail inválido".	
2. Caso o e-mail informado já esteja em uso, o sistema exibe uma mensagem de erro: "Endereço de e-mail já em uso".	
3. Requisito de Negócio (RN01): Não poderão existir usuários distintos com o mesmo endereço de e-mail cadastrado no sistema.	
E2 - Telefone Inválido:	
1. Se o formato do telefone estiver incorreto ou se o número já estiver em uso, o sistema exibe uma mensagem de erro: "Número de telefone inválido".	

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

2. RN02: Não podem existir dois usuários com o mesmo número de telefone cadastrado.
Pós-condição: O usuário é cadastrado no sistema com uma identificação única e uma senha inicial gerada. Ele recebe um e-mail com as instruções de acesso, incluindo a necessidade de trocar a senha no primeiro login.

Quadro 2 - UC02 - Efetuar Login

Caso de Uso: UC02 - Efetuar Login			
Resumo: Este caso de uso permite que qualquer usuário cadastrado efetue login no sistema utilizando suas credenciais, garantindo acesso controlado com base no tipo de perfil. O sistema autentica o usuário e fornece acesso às funcionalidades conforme suas permissões.			
Atores Participantes: Usuário (Tutor, Veterinário, Fornecedor, Administrador)			
Casos de Uso Relacionados: UC01 - Cadastrar Perfis de Usuário (para que o usuário já esteja registrado antes de fazer login). UC03 - Reenviar Senha (caso o usuário esqueça a senha e precise redefini-la).			
Pré-condição: O usuário deve ter uma conta válida e já cadastrada no sistema. Ele também deve possuir um meio de autenticação, como senha ou associação com uma conta Google.			
Fluxo Principal: <table border="0" style="width: 100%;"> <tr> <td style="vertical-align: top; width: 50%;"> Ações do Ator: <ol style="list-style-type: none"> O usuário acessa a interface de login e insere suas credenciais (endereço de e-mail e senha). </td><td style="vertical-align: top; width: 50%;"> Ações do Sistema: <ol style="list-style-type: none"> O sistema verifica se o e-mail e a senha fornecidos correspondem a uma conta ativa. Se as credenciais são válidas, o sistema autentica o usuário e redireciona para a página inicial, conforme o perfil de acesso (ex.: Tutor é redirecionado para a área de Pets, enquanto o Administrador para a área de Administração). </td></tr> </table>		Ações do Ator: <ol style="list-style-type: none"> O usuário acessa a interface de login e insere suas credenciais (endereço de e-mail e senha). 	Ações do Sistema: <ol style="list-style-type: none"> O sistema verifica se o e-mail e a senha fornecidos correspondem a uma conta ativa. Se as credenciais são válidas, o sistema autentica o usuário e redireciona para a página inicial, conforme o perfil de acesso (ex.: Tutor é redirecionado para a área de Pets, enquanto o Administrador para a área de Administração).
Ações do Ator: <ol style="list-style-type: none"> O usuário acessa a interface de login e insere suas credenciais (endereço de e-mail e senha). 	Ações do Sistema: <ol style="list-style-type: none"> O sistema verifica se o e-mail e a senha fornecidos correspondem a uma conta ativa. Se as credenciais são válidas, o sistema autentica o usuário e redireciona para a página inicial, conforme o perfil de acesso (ex.: Tutor é redirecionado para a área de Pets, enquanto o Administrador para a área de Administração). 		
Fluxo Alternativo: A1 - Recuperação de senha: <ol style="list-style-type: none"> Se o usuário esqueceu a senha, ele pode clicar na opção "Esqueci minha senha". O sistema redireciona para o caso de uso UC03 - Reenviar Senha, onde ele poderá iniciar o processo de redefinição. 			
Fluxo de Exceção: E1 - Credenciais Inválidas: <ol style="list-style-type: none"> Se o e-mail ou a senha fornecidos não correspondem a uma conta no sistema, o sistema exibe uma mensagem de erro: "E-mail ou senha incorretos". Após três tentativas falhas de login consecutivas, o sistema bloqueia temporariamente o acesso e exibe uma mensagem de segurança: "Muitas tentativas falhas. Tente novamente em 15 minutos." 			
Pós-condição: O usuário é autenticado e redirecionado à interface principal conforme seu tipo de perfil, com as permissões apropriadas de acesso.			

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

Quadro 3 - UC03 - Reenviar Senha

Caso de Uso: UC03 – Reenviar Senha			
Resumo: Permite que o usuário solicite o envio de um link para redefinição de senha caso tenha esquecido a atual. O link será enviado para o e-mail cadastrado e terá um tempo de expiração para segurança.			
Atores Participantes: Usuário (Tutor, Veterinário, Fornecedor, Administrador).			
Casos de Uso Relacionados: UC02 - Efetuar Login (pois o usuário pode esquecer a senha ao tentar login). UC04 - Alterar Senha (para que o usuário defina uma nova senha após o reenvio).			
Pré-condição: O e-mail do usuário deve estar registrado no sistema e válido.			
Fluxo Principal: <table border="0" style="width: 100%;"> <tr> <td style="vertical-align: top; width: 50%;"> Ações do Ator: <ol style="list-style-type: none"> O usuário seleciona a opção "Esqueci minha senha" na interface de login. </td><td style="vertical-align: top; width: 50%;"> Ações do Sistema: <ol style="list-style-type: none"> O sistema solicita ao usuário que insira seu e-mail cadastrado. Verifica se o e-mail está registrado. Se estiver, o sistema gera um link único de recuperação com um token de segurança. Envia um e-mail ao usuário com o link e instruções para redefinição de senha. </td></tr> </table>		Ações do Ator: <ol style="list-style-type: none"> O usuário seleciona a opção "Esqueci minha senha" na interface de login. 	Ações do Sistema: <ol style="list-style-type: none"> O sistema solicita ao usuário que insira seu e-mail cadastrado. Verifica se o e-mail está registrado. Se estiver, o sistema gera um link único de recuperação com um token de segurança. Envia um e-mail ao usuário com o link e instruções para redefinição de senha.
Ações do Ator: <ol style="list-style-type: none"> O usuário seleciona a opção "Esqueci minha senha" na interface de login. 	Ações do Sistema: <ol style="list-style-type: none"> O sistema solicita ao usuário que insira seu e-mail cadastrado. Verifica se o e-mail está registrado. Se estiver, o sistema gera um link único de recuperação com um token de segurança. Envia um e-mail ao usuário com o link e instruções para redefinição de senha. 		
Fluxo Alternativo: <p>A1 - Cancelamento da solicitação:</p> <ol style="list-style-type: none"> O usuário pode optar por cancelar a solicitação de redefinição de senha, retornando à página de login. 			
Fluxo de Exceção: <p>E1 - E-mail não cadastrado:</p> <ol style="list-style-type: none"> Se o e-mail fornecido não estiver cadastrado, o sistema exibe a mensagem: "E-mail não encontrado. Verifique e tente novamente." Caso de segurança: Após várias tentativas com e-mails inexistentes, o sistema pode bloquear temporariamente a função de recuperação. 			
Pós-condição: O usuário recebe o e-mail com o link de recuperação e tem um período limitado para redefinir sua senha. Após esse prazo, o link expira, exigindo que uma nova solicitação seja feita, se necessário.			

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

Quadro 4 - UC04 - Alterar Senha

Caso de Uso: UC04 – Alterar Senha	
Resumo: Permite ao usuário alterar sua senha por uma nova para manter a segurança de sua conta.	
Atores Participantes: Usuário (Tutor, Veterinário, Fornecedor, Administrador).	
Casos de Uso Relacionados: UC02 - Efetuar Login UC03 - Reenviar Senha	
Pré-condição: O usuário deve estar autenticado no sistema.	
Fluxo Principal:	
Ações do Ator: <ol style="list-style-type: none"> 1. O usuário acessa a seção de configurações e escolhe a opção de alteração de senha. 2. Insere a senha atual e a nova senha desejada. 	Ações do Sistema: <ol style="list-style-type: none"> 3. O sistema verifica a senha atual para garantir que está correta. 4. Atualiza a senha no sistema e confirma a alteração.
Fluxo Alternativo: Esquecer a senha atual: <ul style="list-style-type: none"> Se o usuário não se lembra da senha atual, ele pode utilizar o caso de uso UC03 - Reenviar Senha. 	
Fluxo de Exceção: E1 - Nova senha inválida: <ul style="list-style-type: none"> Se a nova senha não atender aos requisitos de segurança (por exemplo, comprimento mínimo), o sistema exibe uma mensagem de erro: "Senha inválida. Insira uma senha que atenda aos requisitos de segurança." 	
Pós-condição: A senha é alterada, e o usuário pode utilizá-la no próximo login.	

Caso de Uso: UC05 - Associar a Conta Google

Caso de Uso: UC05 – Associar a Conta Google	
Resumo: Permite ao usuário vincular sua conta do sistema à conta Google, facilitando futuros logins.	
Atores Participantes: Usuário (Tutor, Veterinário, Fornecedor, Administrador).	
Casos de Uso Relacionados: UC02 - Efetuar Login.	
Pré-condição: O usuário deve estar autenticado no sistema e ter uma conta Google ativa.	
Fluxo Principal:	
Ações do Ator: <ol style="list-style-type: none"> 1. O usuário seleciona a opção para associar a conta Google. 2. Autoriza o sistema a acessar os dados da conta Google. 	Ações do Sistema: <ol style="list-style-type: none"> 3. Verifica se a conta Google já está vinculada a outra conta do sistema. 4. Conclui a associação e exibe uma mensagem de confirmação.
Fluxo Alternativo: Conta já vinculada a outro usuário: <ul style="list-style-type: none"> Se a conta Google já está associada a outro perfil, o sistema exibe uma mensagem de erro. 	
Fluxo de Exceção: Falha na autenticação do Google:	

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

<ul style="list-style-type: none"> Se a autenticação com o Google falhar, exibe uma mensagem de erro: "Falha ao vincular a conta Google. Tente novamente."
Pós-condição: A conta Google é vinculada ao perfil do usuário no sistema.

Caso de Uso: UC06 - Gerenciar Perfis

Caso de Uso: Caso de Uso: UC06 - Gerenciar Perfis.
Resumo: Permite que o Administrador visualize, edite ou remova perfis de usuários conforme necessário.
Atores Participantes: Administrador.
Casos de Uso Relacionados: UC01 - Cadastrar Perfis de Usuários.
Pré-condição: O Administrador deve estar autenticado e possuir privilégios de gerenciamento.
Fluxo Principal: <div> <div> Ações do Ator: <ol style="list-style-type: none"> O administrador acessa a lista de perfis e escolhe um perfil para editar ou remover. </div> <div> Ações do Sistema: <ol style="list-style-type: none"> Exibe os detalhes do perfil selecionado. Permite a edição dos dados ou a exclusão do perfil. </div> </div>
Fluxo Alternativo: Confirmação de exclusão: <ul style="list-style-type: none"> O sistema solicita uma confirmação antes de excluir o perfil.
Fluxo de Exceção: Erro de permissão: <ul style="list-style-type: none"> Se o administrador tenta alterar um perfil com permissões maiores, o sistema bloqueia a ação e exibe uma mensagem de erro.
Pós-condição: O perfil é atualizado ou removido conforme solicitado.

Reclamar Caso de Uso: UC07 - Cadastrar Pet

Caso de Uso: Caso de Uso: UC07 - Cadastrar Pet
Resumo: Permite que o Tutor cadastre seu pet no sistema com informações relevantes como nome, espécie, idade e histórico de saúde.
Atores Participantes: Tutor.
Casos de Uso Relacionados: UC08 - Integrar Histórico de Saúde.
Pré-condição: O Tutor deve estar autenticado.
Fluxo Principal: <div> <div> Ações do Ator: <ol style="list-style-type: none"> O Tutor acessa a opção de cadastro de pet e preenche os dados solicitados. </div> <div> Ações do Sistema: <ol style="list-style-type: none"> O sistema valida os dados e salva o pet no perfil do Tutor. </div> </div>

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

Fluxo Alternativo: Alteração de dados do pet: <ul style="list-style-type: none"> Caso o pet já esteja cadastrado, o Tutor pode optar por atualizar as informações existentes.
Fluxo de Exceção: Dados insuficientes: <ul style="list-style-type: none"> Se algum dado obrigatório não for preenchido, o sistema exibe uma mensagem de erro.
Pós-condição: O pet é cadastrado e vinculado ao perfil do Tutor.

Caso de Uso: UC08 - Integrar Histórico de Saúde

Caso de Uso: UC08 – Integrar Histórico de Saúde.
Resumo: Permite ao Tutor adicionar e manter o histórico de saúde do pet, incluindo consultas, vacinas e exames.
Atores Participantes: Tutor, Veterinário.
Casos de Uso Relacionados: UC07 - Cadastrar Pet.
Pré-condição: O pet já deve estar cadastrado no sistema.
Fluxo Principal: <div style="display: flex; justify-content: space-between;"> <div> Ações do Ator: <ol style="list-style-type: none"> O Tutor ou Veterinário acessa o histórico de saúde do pet e adiciona uma nova entrada. </div> <div> Ações do Sistema: <ol style="list-style-type: none"> O sistema salva as informações no histórico do pet. </div> </div>
Fluxo Alternativo: Edição de uma entrada: <ul style="list-style-type: none"> Permite editar ou remover entradas no histórico de saúde.
Fluxo de Exceção: Dados incompletos: <ul style="list-style-type: none"> Caso os dados inseridos sejam insuficientes, o sistema exibe uma mensagem de erro.
Pós-condição: O histórico de saúde é atualizado com sucesso.

Caso de Uso: UC09 - Reclamar/Conversar

Caso de Uso: UC09 – Reclamar/Conversar
Resumo: Permite ao Tutor registrar reclamações ou iniciar conversas para suporte.
Atores Participantes: Tutor.
Casos de Uso Relacionados: UC10 - Solucionar Problemas.
Pré-condição: O Tutor deve estar autenticado.
Fluxo Principal: <div style="display: flex; justify-content: space-between;"> <div> Ações do Ator: <ol style="list-style-type: none"> O Tutor acessa a interface de suporte e inicia uma nova conversa ou reclamação. </div> <div> Ações do Sistema: <ol style="list-style-type: none"> O sistema registra a conversa e exibe o status. </div> </div>
Fluxo Alternativo: Reabertura de conversa: <ul style="list-style-type: none"> Caso a conversa tenha sido encerrada, permite reabrir se necessário.

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

Fluxo de Exceção:
Conexão perdida:

- Se houver problema na conexão, o sistema exibe uma notificação e permite retomar mais tarde.

Pós-condição:

A conversa ou reclamação é registrada e exibida no histórico do usuário.

Caso de Uso: UC10 - Solucionar Problemas
Caso de Uso:

UC10 – Solucionar Problemas

Resumo:

Permite que o Administrador revise e resolva problemas reportados pelos usuários, incluindo reclamações e pedidos de suporte.

Atores: Participantes:

Administrador.

Casos de Uso Relacionados:

UC09 - Reclamar/Conversar.

Pré-condição:

O Administrador deve estar autenticado e possuir permissões para resolver problemas de suporte.

Fluxo Principal:
Ações do Ator:

- O Administrador acessa a lista de reclamações e problemas reportados pelos usuários.
- Seleciona um problema específico para revisão e resolução.

Ações do Sistema:

- O sistema exibe as informações detalhadas do problema.
- O Administrador fornece uma resposta ou toma uma ação para resolver o problema.

Fluxo Alternativo:
Delegar problema a outro administrador:

- Caso o problema seja complexo, o Administrador pode delegá-lo para outro especialista ou administrador.

Fluxo de Exceção:
Falha na resolução:

- Se a solução não for efetiva, o sistema permite reabrir o problema para investigação adicional.

Pós-condição:

O problema é resolvido ou marcado como resolvido, e o Tutor é notificado.

Caso de Uso: UC11 - Agendar Serviços
Caso de Uso:

Caso de Uso: UC11 - Agendar Serviços

Resumo:

Permite que o Tutor ou Veterinário agende serviços veterinários, como consultas, exames e vacinas.

Atores Participantes:

Tutor, Veterinário.

Casos de Uso Relacionados:

UC12 - Gerenciar Consultas

Pré-condição:

O Tutor ou Veterinário deve estar autenticado.

Fluxo Principal:
Ações do Ator:

- O Tutor ou Veterinário acessa a opção de agendamento de serviços.
- Seleciona o tipo de serviço, data e horário desejados.

Ações do Sistema:

- O sistema verifica a disponibilidade do serviço e do profissional.
- Confirma o agendamento e envia uma notificação ao Tutor ou Veterinário.

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

Fluxo Alternativo: Escolha de horário alternativo: <ul style="list-style-type: none"> Caso o horário desejado não esteja disponível, o sistema sugere horários alternativos.
Fluxo de Exceção: Serviço indisponível: <ul style="list-style-type: none"> Se o serviço estiver indisponível temporariamente, o sistema exibe uma mensagem de erro e sugere tentar novamente mais tarde.
Pós-condição: O serviço é agendado e registrado no sistema, com notificações enviadas aos envolvidos.

Caso de Uso: UC12 - Gerenciar Consultas

Caso de Uso: Caso de Uso: UC12 - Gerenciar Consultas	
Resumo: Permite que o Veterinário acompanhe e gerencie as consultas agendadas com os pets dos tutores.	
Atores Participantes: Veterinário.	
Casos de Uso Relacionados: UC11 - Agendar Serviços.	
Pré-condição: O Veterinário deve estar autenticado.	
Fluxo Principal:	
Ações do Ator: <ol style="list-style-type: none">O Veterinário acessa a lista de consultas agendadas.Visualiza os detalhes de cada consulta e atualiza seu status (ex.: realizada, cancelada).	Ações do Sistema: <ol style="list-style-type: none">O sistema exibe a lista de consultas e permite atualizações de status.Notifica o Tutor sobre alterações, como cancelamentos ou confirmações.
Fluxo Alternativo: Remarcar consulta: <ul style="list-style-type: none">Se necessário, o Veterinário pode remarcar a consulta para uma data futura.	
Fluxo de Exceção: Erro ao atualizar status: <ul style="list-style-type: none">Caso ocorra um erro na atualização do status da consulta, o sistema exibe uma mensagem e sugere tentar novamente.	
Pós-condição: A consulta é atualizada no sistema, e o Tutor é notificado de quaisquer mudanças.	

Caso de Uso: UC13 - Cadastrar Prescrições

Caso de Uso: Caso de Uso: UC13 - Cadastrar Prescrições.	
Resumo: Permite que o Veterinário registre prescrições médicas para os pets no sistema, que estarão acessíveis ao Tutor.	
Atores Participantes: Veterinário	
Casos de Uso Relacionados: UC12 - Gerenciar Consultas.	
Pré-condição: O Veterinário deve estar autenticado e ter uma consulta registrada com o pet.	
Fluxo Principal:	
Ações do Ator:	Ações do Sistema:

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

1. O Veterinário deve estar autenticado e ter uma consulta registrada com o pet.	2. O sistema armazena a prescrição e a vincula ao perfil do pet.
	3. Notifica o Tutor sobre a nova prescrição.
Fluxo Alternativo: Editar prescrição: <ul style="list-style-type: none"> O Veterinário pode acessar prescrições anteriores e editá-las se necessário. 	
Fluxo de Exceção: Falha na criação da prescrição: <ul style="list-style-type: none"> Caso ocorra um erro, o sistema exibe uma mensagem e solicita que o Veterinário tente novamente. 	
Pós-condição: A prescrição é cadastrada e fica disponível para o Tutor acessar.	

Caso de Uso: UC14 - Catalogar Produtos

Caso de Uso: Caso de Uso: UC14 - Catalogar Produtos	
Resumo: Permite que o Fornecedor cadastre e organize produtos disponíveis no sistema para os usuários.	
Atores Participantes: Fornecedor.	
Casos de Uso Relacionados: UC15 - Catalogar Serviços	
Pré-condição: O Fornecedor deve estar autenticado e ter permissões para gerenciar produtos.	
Fluxo Principal: <div> <div> Ações do Ator: <ol style="list-style-type: none"> O Fornecedor acessa a seção de catálogo de produtos e adiciona novos produtos. </div> <div> Ações do Sistema: <ol style="list-style-type: none"> O sistema valida os dados do produto e o adiciona ao catálogo disponível para os usuários. </div> </div>	
Fluxo Alternativo: Atualizar informações do produto: <ul style="list-style-type: none"> O Fornecedor pode editar informações do produto, como preço e descrição. 	
Fluxo de Exceção: Produto duplicado: <ul style="list-style-type: none"> Se o produto já existe no catálogo, o sistema notifica o fornecedor. 	
Pós-condição: O produto é cadastrado e exibido no catálogo.	

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

Caso de Uso: UC15 - Catalogar Serviços

Caso de Uso: Caso de Uso: UC15 - Catalogar Serviços	
Resumo: Permite que o Fornecedor adicione e organize serviços disponíveis na plataforma.	
Atores Participantes: Fornecedor.	
Casos de Uso Relacionados: UC14 - Catalogar Produtos.	
Pré-condição: O Fornecedor deve estar autenticado e possuir permissões para adicionar serviços.	
Fluxo Principal:	
Ações do Ator: 1. O Fornecedor acessa a seção de catálogo de serviços e adiciona os detalhes do serviço.	Ações do Sistema: 2. O sistema valida e armazena o serviço no catálogo.
Fluxo Alternativo:	
Editar informações do serviço: <ul style="list-style-type: none"> Permite que o Fornecedor atualize dados como preço ou disponibilidade. 	
Fluxo de Exceção:	
Serviço duplicado: <ul style="list-style-type: none"> Se o serviço já está listado, o sistema exibe uma mensagem de erro. 	
Pós-condição: O serviço é adicionado ao catálogo e disponível para os usuários.	

Quadro 16 - UC16 - Avaliar Produtos e Serviços

Caso de Uso: UC016 – Avaliar Produtos e Serviços	
Resumo: Permite que o Tutor avalie produtos e serviços com notas e comentários, contribuindo para a experiência de outros usuários na plataforma.	
Atores Participantes: Tutor.	
Casos de Uso Relacionados: UC14 - Catalogar Produtos UC15 - Catalogar Serviços	
Pré-condição: O Tutor deve estar autenticado e ter adquirido ou utilizado o produto ou serviço.	
Fluxo Principal:	
Ações do Ator: 1. O Tutor acessa a seção de avaliações para um produto ou serviço. 2. Insere uma nota e um comentário.	Ações do Sistema: 3. O sistema armazena a avaliação e a associa ao produto ou serviço.
Fluxo Alternativo:	
Editar avaliação: <ul style="list-style-type: none"> O Tutor pode atualizar sua avaliação posteriormente. 	
Fluxo de Exceção:	
Erro na submissão da avaliação: <ul style="list-style-type: none"> Se ocorrer um erro, o sistema exibe uma mensagem de falha. 	

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

Pós-condição:

A avaliação é registrada e exibida para outros usuários.

Quadro 17 - UC17 - Recomendar Produtos e Serviços

Caso de Uso: UC017 – Recomendar Produtos e Serviços.	
Resumo: Permite que o Tutor recomende produtos e serviços a outros usuários, com base em suas próprias experiências.	
Atores Participantes: Tutor	
Casos de Uso Relacionados: UC14 - Catalogar Produtos UC15 - Catalogar Serviços	
Pré-condição: O Tutor deve estar autenticado.	
Fluxo Principal:	
Ações do Ator: 1. O Tutor acessa um produto ou serviço e opta por recomendá-lo.	Ações do Sistema: 2. O sistema registra a recomendação e a exibe no perfil do Tutor.
Fluxo Alternativo: Remover recomendação: <ul style="list-style-type: none"> O Tutor pode retirar a recomendação a qualquer momento. 	
Fluxo de Exceção: E1: Se não houver dados suficientes para recomendação, o sistema informa que não há recomendações disponíveis no momento.	
Pós-condição: O produto ou serviço é recomendado pelo Tutor, sendo exibido aos outros usuários.	

Quadro 18 - UC18 - Consultar Histórico de Compras, agendamentos e serviços.

Caso de Uso: UC018 – Consultar Histórico de Compras, agendamentos e serviços.	
Resumo: Permite que o Tutor visualize o histórico de compras e serviços adquiridos anteriormente.	
Atores Participantes: Tutor.	
Casos de Uso Relacionados: UC19 - Finalizar Pedido	
Pré-condição: O Tutor deve estar autenticado.	
Fluxo Principal:	
Ações do Ator: 1. O Tutor acessa a seção de histórico de compras.	Ações do Sistema: 2. O sistema exibe a lista de compras e serviços passados com detalhes de cada transação.
Fluxo Alternativo: A1: Se o histórico estiver vazio, o sistema informa que não há registros para exibir.	
Fluxo de Exceção:	

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

E1: Se ocorrer um erro de sistema ao buscar o histórico, o sistema exibe uma mensagem de erro.

Pós-condição:

O Tutor pode revisar seu histórico de compras e serviços.

Quadro 19 - UC19 - Finalizar Pedido

Caso de Uso: UC019 – Finalizar Pedido	
Resumo: Permite que o Tutor finalize uma compra ou pedido de serviço, registrando as informações para pagamento e entrega.	
Atores Participantes: Tutor.	
Casos de Uso Relacionados: UC20 - Consultar Status de Pedido.	
Pré-condição: O Tutor deve ter itens no carrinho de compras.	
Fluxo Principal:	
Ações do Ator: 1. O Tutor acessa o carrinho e confirma os itens selecionados.	Ações do Sistema: 2. O sistema calcula o valor total e solicita a forma de pagamento e endereço de entrega. 3. Finaliza o pedido e gera uma confirmação.
Fluxo Alternativo: A1: Se o carrinho estiver vazio, o sistema informa que é necessário adicionar itens para finalizar o pedido.	
Fluxo de Exceção: E1: Se ocorrer um erro de sistema ao processar o pedido, o sistema exibe uma mensagem de erro.	
Pós-condição: Pós-condição: O pedido é registrado e aguardando o processamento de entrega.	

Quadro 20 - UC20 - Consultar Status de Pedido

Caso de Uso: UC020 – Consultar Status de Pedido	
Resumo: Permite que o Tutor verifique o status atual de um pedido em andamento, incluindo informações sobre a etapa de entrega.	
Atores Participantes: Tutor.	
Casos de Uso Relacionados: UC19 - Finalizar Pedido.	
Pré-condição: O Tutor deve ter um pedido em andamento.	
Fluxo Principal:	
Ações do Ator: 1. O Tutor acessa o status do pedido.	Ações do Sistema: 2. O sistema exibe o status atual do pedido (ex.: em processamento, enviado).

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

Fluxo Alternativo:

A1: Se o pedido não for encontrado, o sistema exibe uma mensagem de erro.

Fluxo de Exceção:

E1: Se ocorrer um erro ao consultar o status, o sistema exibe uma mensagem de erro.

Pós-condição:

O Tutor obtém informações atualizadas sobre o andamento do pedido.

Quadro 21 - UC21 - Notificar Entrega

Caso de Uso:

UC021 – Notificar Entrega

Resumo:

Permite que o sistema notifique o Tutor sobre o status da entrega de um pedido.

Atores Participantes:

Sistema, Tutor.

Casos de Uso Relacionados:

UC20 – Consultar Status de Pedido.

Pré-condição:

O pedido deve estar em processo de entrega.

Fluxo Principal:

Ações do Ator:

1. O Fornecedor notifica o sistema sobre a entrega realizada.

Ações do Sistema:

2. O sistema envia notificações ao Tutor conforme o status de entrega do pedido muda.

Fluxo Alternativo:

A1: Se o pedido não estiver na lista de entregas, o sistema informa que o pedido não está qualificado para notificação de entrega.

Fluxo de Exceção:

E1: Se houver falha ao registrar a notificação, o sistema exibe uma mensagem de erro.

Pós-condição:

O Tutor é notificado das atualizações de entrega do pedido.

Quadro 22 - UC22 - Emitir Nota Fiscal

Caso de Uso:

UC22 – Emitir Nota Fiscal

Resumo:

Permite que o sistema emita uma nota fiscal para compras realizadas.

Atores Participantes:

Sistema, Tutor, Fornecedor.

Casos de Uso Relacionados:

UC19 - Finalizar Pedido.

Pré-condição:

O pedido deve estar finalizado e pronto para ser faturado.

Fluxo Principal:

Ações do Ator:

Ações do Sistema:

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

1. O Fornecedor solicita a emissão da nota fiscal para um pedido.	2. O sistema emite a nota fiscal e a associa ao pedido do Tutor.
Fluxo Alternativo: A1: Se o pedido já tiver uma nota fiscal emitida, o sistema informa que a nota já foi gerada.	
Fluxo de Exceção: E1: Se ocorrer um erro ao emitir a nota, o sistema exibe uma mensagem de erro.	
Pós-condição: A nota fiscal é gerada e disponível para o Tutor e Fornecedor.	

Quadro 23 - UC23 - Gerenciar Formas de Pagamento

Caso de Uso: UC23 – Gerenciar Formas de Pagamento
Resumo: Permite que o Tutor gerencie as formas de pagamento salvas em seu perfil.
Atores Participantes: Tutor.
Casos de Uso Relacionados: UC19 - Finalizar Pedido.
Pré-condição: O Tutor deve estar autenticado.
Fluxo Principal: <div style="display: flex; justify-content: space-between;"> <div> Ações do Ator: 1. O Tutor acessa as configurações de pagamento e adiciona ou remove formas de pagamento. </div> <div> Ações do Sistema: 2. O sistema atualiza as informações de pagamento no perfil do Tutor. </div> </div>
Fluxo Alternativo: A1: Se o Tutor tentar remover todas as formas de pagamento, o sistema informa que é necessário manter pelo menos uma forma de pagamento ativa.
Fluxo de Exceção: E1: Se houver erro ao atualizar as formas de pagamento, o sistema exibe uma mensagem de erro.
Pós-condição: As formas de pagamento do Tutor são atualizadas.

Quadro 24 - UC24 - Acompanhar Entrega

Caso de Uso: UC24 – Acompanhar Entrega.
Resumo: Permite que o Tutor acompanhe a localização e andamento da entrega de um pedido em tempo real.
Atores Participantes: Tutor, Sistema de Entregas.
Casos de Uso Relacionados: UC21 - Notificar Entrega.
Pré-condição: O Tutor deve ter um pedido em entrega.

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

Fluxo Principal:	
Ações do Ator:	Ações do Sistema:
1. O Tutor acessa a seção de acompanhamento de entrega.	2. O sistema exibe informações de rastreamento.
Fluxo Alternativo:	
A1: Se o Tutor tenta acompanhar um pedido que ainda não foi enviado, o sistema exibe uma mensagem informando que o pedido ainda está em preparação e que o acompanhamento estará disponível após o envio.	
Fluxo de Exceção:	
E1: Se ocorrer um erro ao buscar o status da entrega, o sistema exibe uma mensagem de erro e solicita que o Tutor tente novamente mais tarde.	
E2: Se o pedido não for encontrado, o sistema exibe uma mensagem de erro informando que o pedido especificado não existe ou não foi localizado.	
Pós-condição:	
O Tutor visualiza o status atualizado da entrega.	

Quadro 25 - UC25 - Gerar Relatórios de Pagamentos ou Atividades na Plataforma

Caso de Uso:
UC25 –Gerar Relatórios de Pagamentos ou Atividades na Plataforma
Resumo:
Permite que o Administrador gere relatórios financeiros e de atividade de usuários na plataforma para fins de análise.
Atores Participantes:
Administrador.
Casos de Uso Relacionados:
UC06 - Gerenciar Perfis.
Pré-condição:
O Administrador deve estar autenticado e ter permissões para acesso a dados de relatórios.
Fluxo Principal:
Ações do Ator:
1. O Administrador acessa a seção de relatórios e escolhe o tipo de relatório a ser gerado.
Ações do Sistema:
2. O sistema gera o relatório com os dados solicitados e os exibe ao Administrador.
Fluxo Alternativo:
A1: Se não houver dados para o período selecionado, o sistema informa que não há registros para gerar o relatório.
Fluxo de Exceção:
E1: Se ocorrer um erro ao gerar o relatório, o sistema exibe uma mensagem de erro.
Pós-condição:
O Administrador recebe o relatório gerado.

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

Quadro 26 - UC26 - Controlar Produtos e Serviços Listados

Caso de Uso: UC26 – Finalizar Pedido.	
Resumo: Permite que o Administrador controle os produtos e serviços listados na plataforma, podendo ocultar ou excluir itens se necessário.	
Atores Participantes: Administrador.	
Casos de Uso Relacionados: UC14 - Catalogar Produtos UC15 - Catalogar Serviços	
Pré-condição: O Administrador deve estar autenticado.	
Fluxo Principal:	
Ações do Ator: 1. O Administrador acessa a lista de produtos e serviços e seleciona aqueles que precisam ser controlados.	Ações do Sistema: 2. O sistema permite que o Administrador oculte ou remova itens conforme necessário.
Fluxo Alternativo: A1: Se o Administrador tentar editar um produto inexistente, o sistema informa que o produto não está disponível.	
Fluxo de Exceção: E1: Se houver erro ao atualizar a lista de produtos, o sistema exibe uma mensagem de erro.	
Pós-condição: Os produtos e serviços são atualizados conforme o controle do Administrador.	

Quadro 27 - UC27 - Gerenciar Usuários

Caso de Uso: UC27 – Gerenciar Usuários	
Resumo: Este caso de uso permite ao Administrador gerenciar as informações dos usuários cadastrados na plataforma, incluindo adicionar, modificar e remover perfis de usuários.	
Atores Participantes: Administrador.	
Casos de Uso Relacionados: UC01 - Cadastrar Perfis de Usuários	
Pré-condição: O Administrador deve estar autenticado no sistema.	
Fluxo Principal:	
Ações do Ator: 1. O Administrador solicita a lista de usuários cadastrados. 2. O Administrador seleciona um usuário para visualizar, alterar ou excluir 3. O Administrador confirma as alterações realizadas ou a exclusão do usuário.	Ações do Sistema: 4. O sistema exibe a lista completa de usuários registrados. 5. O sistema permite a edição ou remoção do usuário selecionado. 6. O sistema atualiza as informações do usuário ou realiza a exclusão.
Fluxo Alternativo: A1: Se o Administrador tenta visualizar um usuário inexistente, o sistema exibe uma mensagem informando que o usuário não foi encontrado.	
Fluxo de Exceção:	

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

E1: Se ocorrer um erro de sistema ao buscar ou atualizar os dados do usuário, o sistema exibe uma mensagem de erro e solicita ao Administrador que tente novamente.

Pós-condição:

Os dados do usuário são atualizados ou removidos conforme solicitado pelo Administrador.

Quadro 28 - UC28 - Gerenciar Carrinho

Caso de Uso:

UC28 – Gerenciar Carrinho

Resumo:

Este caso de uso permite ao Tutor adicionar, remover e visualizar itens no carrinho antes de concluir a compra.

Atores Participantes:

Tutor.

Casos de Uso Relacionados:

UC19 - Finalizar Pedido.

Pré-condição:

O Tutor deve estar autenticado no sistema.

Fluxo Principal:

Ações do Ator:

1. O Tutor adiciona um item ao carrinho.
2. O Tutor remove um item do carrinho.
3. O Tutor visualiza o conteúdo do carrinho para confirmar os itens.

Ações do Sistema:

4. O sistema adiciona o item ao carrinho e exibe a lista atualizada.
5. O sistema remove o item e exibe a lista atualizada.
6. O sistema exibe o resumo dos itens e o valor total.

Fluxo Alternativo:

A1: Se o Tutor tenta remover um item inexistente, o sistema exibe uma mensagem de erro informando que o item não está no carrinho.

Fluxo de Exceção:

E1: Se houver uma falha ao adicionar ou remover itens, o sistema exibe uma mensagem de erro.

Pós-condição:

O carrinho é atualizado conforme as ações realizadas pelo Tutor.

Quadro 29 - UC29 - Pesquisar Produtos e Serviços

Caso de Uso:

UC29 – Pesquisar Produtos e Serviços

Resumo:

Este caso de uso permite ao Tutor buscar por produtos e serviços disponíveis na plataforma.

Atores Participantes:

Tutor.

Casos de Uso Relacionados:

UC17 - Recomendar Produtos e Serviços.

Pré-condição:

O Tutor deve estar autenticado no sistema.

Fluxo Principal:

Ações do Ator:

Ações do Sistema:

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

<ol style="list-style-type: none"> 1. O Tutor digita uma palavra-chave na barra de pesquisa. 2. O Tutor seleciona um item para visualizar detalhes. 3. O sistema processa a pesquisa e exibe a lista de produtos e serviços correspondentes. 4. O sistema exibe as informações detalhadas do item selecionado.
Fluxo Alternativo: A1: Se o Tutor não digita nada e tenta pesquisar, o sistema exibe uma mensagem pedindo para inserir uma palavra-chave.
Fluxo de Exceção: E1: Se ocorrer um erro ao realizar a pesquisa, o sistema exibe uma mensagem de erro e solicita que o Tutor tente novamente.
Pós-condição: O Tutor visualiza os resultados da pesquisa e, se desejar, pode selecionar um item para mais informações.

Quadro 30 - UC30 - Ver o Estado de Pagamento de um Pedido

Caso de Uso: UC30 – Ver o Estado de Pagamento de um Pedido			
Resumo: Este caso de uso permite ao Tutor verificar o status de pagamento de um pedido específico.			
Atores Participantes: Tutor.			
Casos de Uso Relacionados: UC19 - Finalizar Pedido UC24 - Acompanhar Entrega			
Pré-condição: O Tutor deve ter realizado um pedido e estar autenticado no sistema.			
Fluxo Principal: <table> <tr> <td> Ações do Ator: <ol style="list-style-type: none"> 1. O Tutor seleciona um pedido para verificar o estado de pagamento. </td><td> Ações do Sistema: <ol style="list-style-type: none"> 2. O sistema exibe o status atual do pagamento do pedido, indicando se foi pago, pendente ou em processamento. </td></tr> </table>		Ações do Ator: <ol style="list-style-type: none"> 1. O Tutor seleciona um pedido para verificar o estado de pagamento. 	Ações do Sistema: <ol style="list-style-type: none"> 2. O sistema exibe o status atual do pagamento do pedido, indicando se foi pago, pendente ou em processamento.
Ações do Ator: <ol style="list-style-type: none"> 1. O Tutor seleciona um pedido para verificar o estado de pagamento. 	Ações do Sistema: <ol style="list-style-type: none"> 2. O sistema exibe o status atual do pagamento do pedido, indicando se foi pago, pendente ou em processamento. 		
Fluxo Alternativo: A1: Se o Tutor seleciona um pedido inexistente, o sistema exibe uma mensagem informando que o pedido não foi encontrado.			
Fluxo de Exceção: E1: Se ocorrer um erro ao buscar o estado de pagamento, o sistema exibe uma mensagem de erro e solicita ao Tutor que tente novamente.			
Pós-condição: O Tutor visualiza o estado atual do pagamento do pedido selecionado.			

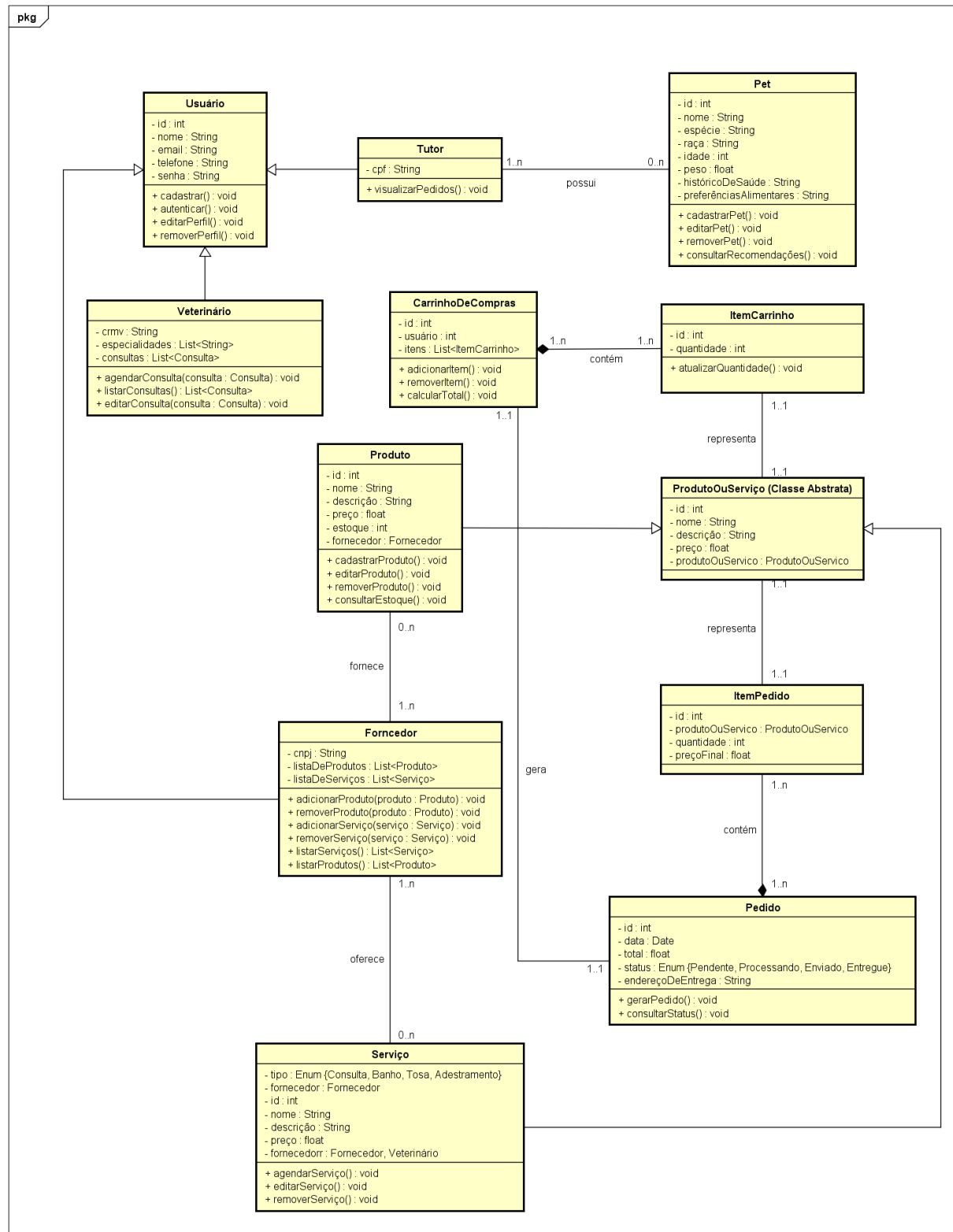
5. Especificação das Classes

Este Capítulo do Documento de Especificação de Requisitos irá detalhar por intermédio da UML 2.5 as funcionalidades descritas sob o Capítulo 3, na Seção de Requisitos Funcionais. Entende-se que o conjunto dos Requisitos Funcionais do Software estão contemplados no Diagrama de Classe, identificando de forma completa todas as entidades e suas associações que interagem com o Software a ser desenvolvido no projeto.

5.1. Diagrama de Classe

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

Figura 2 - Diagrama de Classe do MarketPlace da PetPlace



PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

5.2 Explicação das Classes

1. Usuário

- **Descrição:** Classe genérica que representa qualquer tipo de usuário do sistema (Tutor, Fornecedor e Veterinário).
- **Atributos:**
 - 'id: int': Identificador único do usuário.
 - 'nome: String': Nome completo do usuário.
 - 'email: String': Endereço de e-mail do usuário.
 - 'telefone: String': Número de telefone do usuário.
 - 'senha: String': Senha usada para autenticação no sistema.
- **Métodos:**
 - cadastrar(): Método para registrar um novo usuário.
 - autenticar(): Método para verificar se o usuário pode acessar o sistema.
 - editarPerfil(): Método para modificar os dados do usuário.
 - removerPerfil(): Método para remover o perfil do usuário do sistema.

2. Tutor

- **Descrição:** Subclasse de Usuário, representa o responsável pelo pet no sistema.
- **Atributos:**
 - 'cpf: String': Número do CPF do Tutor.
- **Métodos:**
 - visualizarPedido(): Exibe os pedidos feitos pelo Tutor no sistema.

3. Veterinário

- **Descrição:** Subclasse de Usuário, representa o veterinário que oferece serviços no sistema.
- **Atributos:**
 - 'crm: String': Número de registro profissional do veterinário.
 - 'especialidades: List<String>': Lista de especialidades do veterinário.
 - 'consultas: List<Consulta>': Lista de consultas realizadas pelo veterinário.
- **Métodos:**
 - 'agendarConsulta(consulta: Consulta)': Agendar uma nova consulta para um pet.
 - 'listarConsultas()': Listar as consultas realizadas.
 - 'editarConsulta(consulta: Consulta)': Editar os detalhes de uma consulta.

4. Fornecedor

- **Descrição:** Representa o fornecedor de produtos e/ou serviços no marketplace.
- **Atributos:**
 - 'cnpj: String': CNPJ do fornecedor.
 - 'listaDeProdutos: List<Produto>': Lista de produtos fornecidos.
 - 'listaDeServicos: List<Servico>': Lista de serviços oferecidos.
- **Métodos:**
 - 'adicionarProduto(produto: Produto)': Adicionar um novo produto ao catálogo.
 - 'removerProduto(produto: Produto)': Remover um produto do catálogo.
 - 'adicionarServico(servico: Servico)': Adicionar um novo serviço ao catálogo.
 - 'removerServico(servico: Servico)': Remover um serviço do catálogo.
 - 'listarServicos()': Listar os serviços disponíveis.
 - 'listarProdutos()': Listar os produtos disponíveis.

5. Pet

- **Descrição:** Representa o animal de estimação do Tutor no sistema.
- **Atributos:**
 - 'id: int': Identificador único do pet.

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

- `nome: String`: Nome do pet.
 - `especie: String`: Espécie do pet (cão, gato, etc.).
 - `raca: String`: Raça do pet.
 - `idade: int`: Idade do pet.
 - `peso: float`: Peso do pet.
 - `historicoSaude: String`: Histórico médico do pet.
 - `preferenciasAlimentares: String`: Preferências alimentares do pet.
 - **Métodos:**
 - `cadastrarPet()`: Método para cadastrar um novo pet no sistema.
 - `editarPet()`: Método para editar as informações de um pet.
 - `removerPet()`: Método para remover um pet do sistema.
 - `consultarRecomendacoes()`: Consultar recomendações de cuidados e produtos para o pet.
- 6. CarrinhoDeCompras**
- **Descrição:** Representa o carrinho de compras de um usuário (Tutor).
 - **Atributos:**
 - `id: int`: Identificador único do carrinho.
 - `usuarioId: int`: Identificador do usuário proprietário do carrinho.
 - `items: List<ItemCarrinho>`: Lista de itens (produtos e/ou serviços) no carrinho.
 - **Métodos:**
 - `adicionarItem()`: Método para adicionar um item ao carrinho.
 - `removerItem()`: Método para remover um item do carrinho.
 - `calcularTotal()`: Calcula o valor total dos itens no carrinho.
- 7. ItemCarrinho**
- **Descrição:** Representa um item (produto ou serviço) no carrinho de compras.
 - **Atributos:**
 - `id: int`: Identificador único do item no carrinho.
 - `quantidade: int`: Quantidade do item.
 - **Métodos:**
 - `atualizarQuantidade()`: Atualiza a quantidade de um item no carrinho.
- 8. Produto**
- **Descrição:** Representa um produto disponível para compra no marketplace.
 - **Atributos:**
 - `id: int`: Identificador único do produto.
 - `nome: String`: Nome do produto.
 - `descricao: String`: Descrição do produto.
 - `preco: float`: Preço do produto.
 - `estoque: int`: Quantidade disponível em estoque.
 - **Métodos:**
 - `cadastrarProduto()`: Cadastrar um novo produto no sistema.
 - `editarProduto()`: Editar as informações de um produto.
 - `removerProduto()`: Remover um produto do sistema.
 - `consultarEstoque()`: Consultar a quantidade disponível em estoque.
- 9. ProdutoOuServico (Classe Abstrata)**
- **Descrição:** Classe abstrata que serve como base para produtos e serviços.
 - **Atributos:**
 - `id: int`: Identificador único.
 - `nome: String`: Nome do produto ou serviço.
 - `descricao: String`: Descrição do produto ou serviço.

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

- `preco: float`: Preço do produto ou serviço.

10. Serviço

- **Descrição:** Representa um serviço disponível no marketplace (como consulta, banho, tosa, etc.).
- **Atributos:**
 - `tipo: Enum`: Tipo de serviço (Consulta, Banho, Tosa, Adestramento).
 - `fornecedor: Fornecedor`: Fornecedor do serviço.
 - `id: int`: Identificador único do serviço.
 - `nome: String`: Nome do serviço.
 - `descricao: String`: Descrição do serviço.
 - `preco: float`: Preço do serviço.
- **Métodos:**
 - `agendarServico()`: Agendar um serviço.
 - `editarServico()`: Editar os detalhes de um serviço.
 - `removerServico()`: Remover um serviço do sistema.

11. ItemPedido

- **Descrição:** Representa um item dentro de um pedido, podendo ser um produto ou serviço.
- **Atributos:**
 - `id: int`: Identificador único do item.
 - `produtoOuServico: ProdutoOuServico`: Produto ou serviço associado ao item.
 - `quantidade: int`: Quantidade do item.
 - `precoFinal: float`: Preço final do item no pedido.

12. Pedido

- **Descrição:** Representa um pedido feito pelo tutor no sistema.
- **Atributos:**
 - `id: int`: Identificador único do pedido.
 - `data: Date`: Data do pedido.
 - `total: float`: Valor total do pedido.
 - `status: Enum`: Status do pedido (Pendente, Processando, Enviado, Entregue).
 - `enderecoDeEntrega: String`: Endereço para entrega do pedido.
- **Métodos:**
 - `gerarPedido()`: Gerar um novo pedido.
 - `consultarStatus()`: Consultar o status do pedido.

Relações Entre as Classes:

- Usuário - Tutor: Herança simples, onde Tutor é uma especialização de Usuário.
- Usuário - Veterinário: Herança simples, onde Veterinário é uma especialização de Usuário.
- Tutor - Pet: Um tutor pode possuir vários pets, indicando uma relação de um-para-muitos.
- CarrinhoDeCompras - ItemCarrinho: Um carrinho de compras pode conter vários itens, indicando uma relação de um-para-muitos.
- ItemCarrinho - ProdutoOuServico: Um item no carrinho representa um produto ou um serviço.
- Fornecedor - Produto/Servico: Um fornecedor pode oferecer vários produtos e serviços, indicando uma relação de um-para-muitos.
- Pedido - ItemPedido: Um pedido pode conter vários itens, indicando uma relação de um-para-muitos.
- ItemPedido - ProdutoOuServico: Um item no pedido representa um produto ou um serviço.

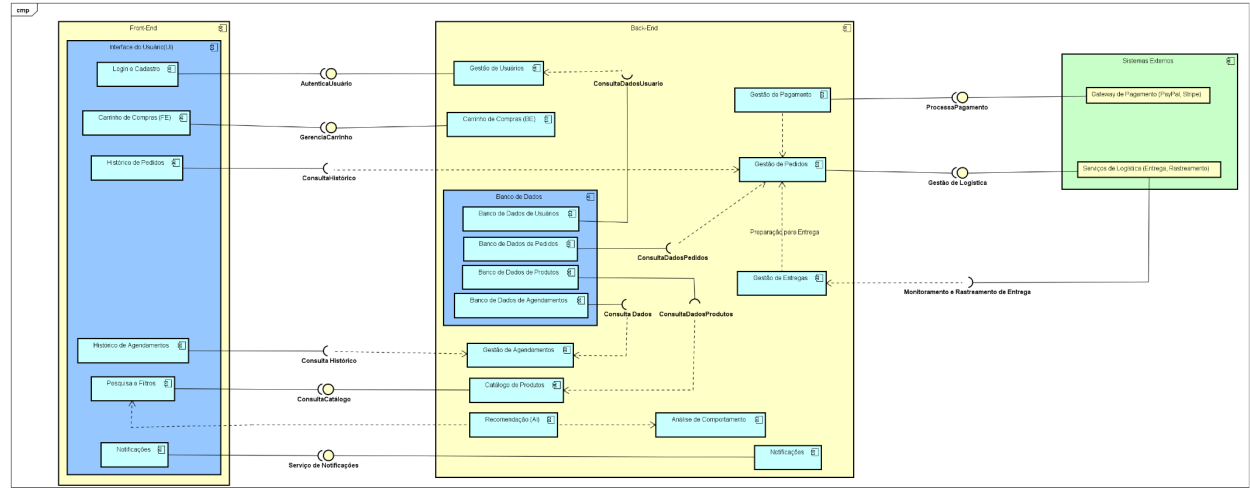
PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

6. Especificação dos Componentes

Este Capítulo do Documento de Especificação de Requisitos irá detalhar, por intermédio da UML 2.5, as funcionalidades descritas no Capítulo 3, na Seção de Requisitos Funcionais. Entende-se que o conjunto dos Requisitos Funcionais do Software está contemplado no Diagrama de Componentes, identificando de forma completa todos os componentes do sistema e suas interações que compõem o Software a ser desenvolvido no projeto.

6.1. Diagrama de Componentes

Figura 3 - Diagrama de Componentes do MarketPlace da PetPlace



6.2. Explicação dos componentes

O diagrama representa a arquitetura de componentes de um sistema dividido entre **Front-End**, **Back-End**, e **Sistemas Externos**, abordando os componentes principais e as interações entre eles.

1. Front-End

O Front-End é a parte visível ao usuário, composta por vários componentes que tratam da interação direta com os usuários e da interface gráfica.

- **Interface de Usuário (UI):** Este componente principal representa a camada com a qual o usuário final interage diretamente. Ele engloba a apresentação e a interação visual usando tecnologias como HTML, CSS, e JavaScript.
 - **Login e Cadastro:** Está ligado à **Gestão de Usuários** no Back-End através da interface **AutenticaUsuário**, onde os dados de login são validados e os usuários são autenticados.
 - **Carrinho de Compras (FE):** Conectado ao componente **Carrinho de Compras (BE)** no Back-End pela interface **GerenciaCarrinho**, é responsável por adicionar, remover e manipular produtos no carrinho.
 - **Histórico de Pedidos:** Este componente permite que o usuário visualize seus pedidos passados. Ele consulta dados do **Banco de Dados de Pedidos** através da interface **ConsultaHistórico**.
 - **Histórico de Agendamentos:** Similar ao histórico de pedidos, consulta os dados de agendamentos anteriores e interage com o Back-End via a interface **ConsultaHistórico**.
 - **Pesquisa e Filtros:** Este componente lida com as consultas de produtos, realizando pesquisas no **Catálogo de Produtos** através da interface **ConsultaCatálogo**.

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

- **Notificações:** O Front-End recebe notificações enviadas do Back-End via o **Serviço de Notificações**.

2. Back-End

O Back-End é onde as principais operações do sistema são processadas, desde o gerenciamento de usuários até a logística de entrega.

- **Gestão de Usuários:** Responsável por toda a administração dos usuários, como autenticação e gerenciamento de perfis. Está ligado ao banco de dados através da interface **ConsultaDadosUsuario** e ao **Login e Cadastro** no Front-End.
- **Carrinho de Compras (BE):** Interage com o Front-End para manter o estado do carrinho e realizar as operações necessárias. Conecta-se com o **Banco de Dados de Produtos** para garantir a disponibilidade de produtos.
- **Gestão de Agendamentos:** Gerencia os agendamentos de entregas ou outros serviços relacionados e armazena essas informações no **Banco de Dados de Agendamentos**.
- **Gestão de Pagamento:** Interage com **Gateways de Pagamento** como PayPal ou Stripe para processar pagamentos através da interface **ProcessaPagamento**.
- **Gestão de Pedidos:** Coordena o processamento dos pedidos, lidando com a etapa de preparação para entrega. Este componente prepara os dados para serem enviados ao sistema de logística.
- **Gestão de Entregas:** Conecta-se aos **Serviços de Logística Externos** (como sistemas de rastreamento) para coordenar as entregas. A interface **Monitoramento e Rastreamento de Entrega** é usada para monitorar o status das entregas.
- **Catálogo de Produtos:** Fornece informações sobre os produtos disponíveis para compra, sendo consultado pelo Front-End através da interface **ConsultaCatalogo**.
- **Recomendação (AI):** Utiliza algoritmos de inteligência artificial para gerar recomendações personalizadas para o usuário. Este componente faz análises comportamentais baseadas nos dados dos usuários.
- **Notificações:** Lida com o envio de notificações para os usuários, seja para informar sobre status de pedidos ou promoções. O **Serviço de Notificações** envia essas mensagens ao Front-End.
- **Banco de Dados:** Está subdividido em quatro bancos de dados principais:
 - **Banco de Dados de Usuários:** Armazena dados relacionados a perfis de usuários.
 - **Banco de Dados de Pedidos:** Guarda informações sobre os pedidos realizados.
 - **Banco de Dados de Produtos:** Mantém dados dos produtos disponíveis no catálogo.
 - **Banco de Dados de Agendamentos:** Armazena as informações de agendamentos realizados pelos usuários.

3. Sistemas Externos

Os **Sistemas Externos** são serviços terceirizados com os quais o sistema interage para realizar algumas operações, como pagamentos e logística.

- **Gateway de Pagamento (PayPal, Stripe):** Conecta-se à **Gestão de Pagamento** no Back-End através da interface **ProcessaPagamento**, possibilitando a validação e execução de transações financeiras.
- **Serviços de Logística (Entrega, Rastreamento):** Interagem com o componente **Gestão de Entregas** para coordenar o envio e rastreamento de pedidos, usando a interface **Monitoramento e Rastreamento de Entrega**.

4. Conexões Importantes

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

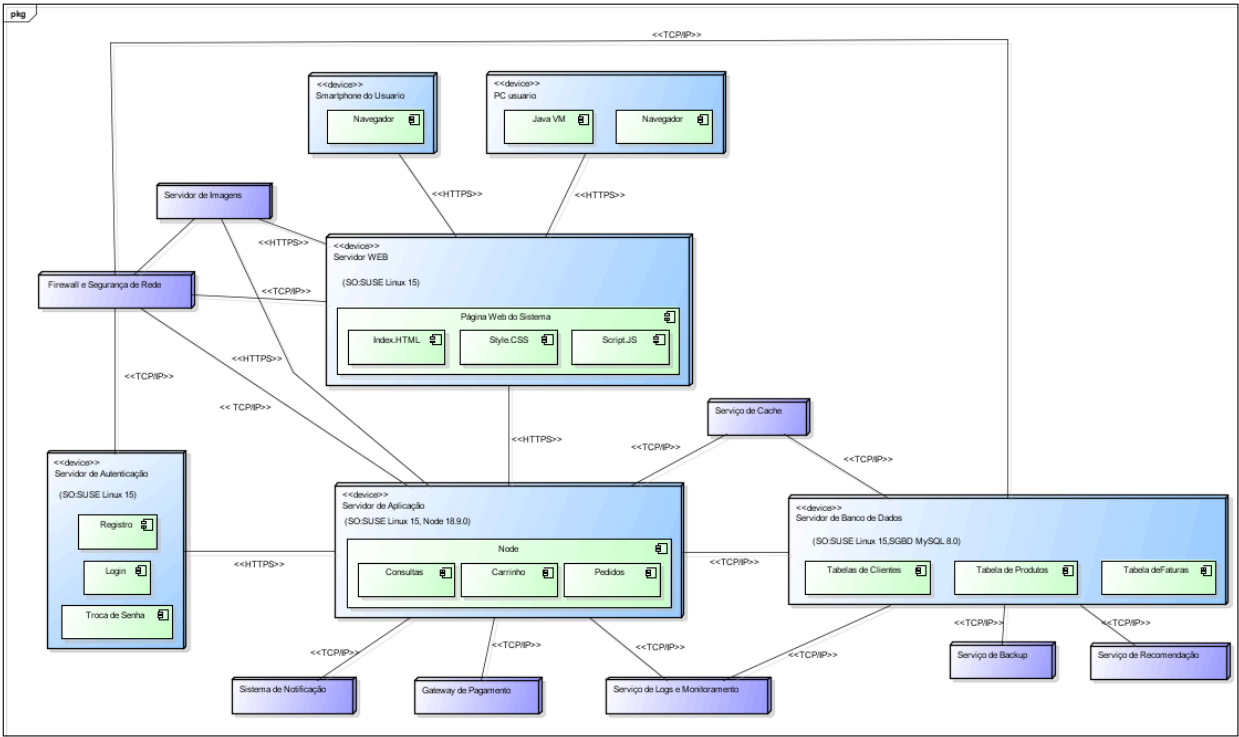
- **AutenticaUsuário:** Conecta a **Gestão de Usuários** com o componente de **Login e Cadastro**, fornecendo autenticação de usuários.
- **GerenciaCarrinho:** Faz a ponte entre o **Carrinho de Compras (FE)** no Front-End e o **Carrinho de Compras (BE)** no Back-End.
- **ConsultaHistórico:** Ligação entre o Front-End e o banco de dados de pedidos, permitindo que o histórico seja visualizado pelos usuários.
- **ConsultaCatalogo:** Utilizada para realizar buscas no catálogo de produtos a partir do Front-End.
- **Serviço de Notificações:** Envia notificações geradas no Back-End para o Front-End.

7. Especificação das Implementações

Este Capítulo do Documento de Especificação de Requisitos irá detalhar, por intermédio da UML 2.5, as funcionalidades descritas no Capítulo 3, na Seção de Requisitos Funcionais. Entende-se que o conjunto dos Requisitos Funcionais do Software está contemplado no Diagrama de Implementação, identificando de forma completa todos os artefatos e a infraestrutura necessários para a implantação do Software a ser desenvolvido no projeto, bem como suas interações e dependências.

7.1. Diagrama de Implementação

Figura 4 - Diagrama de Implementação do MarketPlace da PetPlace



7.2. Explicação das Implementações

1. Firewall e Segurança de Rede

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

- **Descrição:** Atua como a primeira linha de defesa do sistema, protegendo contra acessos não autorizados e ataques de rede.
- **Função:** Filtra o tráfego de entrada e saída, assegurando que apenas requisições válidas atinjam o sistema. Serve como uma camada de segurança essencial para evitar invasões e acesso não autorizado aos servidores internos.
- **Conexões:**
 - Estabelece conexões com o Servidor Web e o Servidor de Autenticação via **TCP/IP**.
 - Controla as solicitações feitas por dispositivos dos usuários, oferecendo um ponto inicial de segurança para proteger a rede interna.

2. Servidor de Autenticação

- **Configuração:** Executa o sistema operacional SUSE Linux 15.
- **Componentes:**
 - **Registro:** Módulo que permite a criação de contas para novos usuários.
 - **Login:** Responsável pela verificação de credenciais para conceder acesso ao sistema.
 - **Troca de Senha:** Oferece a funcionalidade de redefinição de senha para aumentar a segurança.
- **Função:** Valida as credenciais dos usuários antes de permitir o acesso ao sistema, garantindo que apenas usuários autenticados possam interagir com outros serviços.
- **Conexões:**
 - Comunica-se com o Servidor de Aplicação e o Servidor Web via **HTTPS** para proteger a transmissão de dados de autenticação.
 - Conecta-se ao Firewall e Segurança de Rede para proteger as comunicações.

3. Servidor Web

- **Configuração:** Executa SUSE Linux 15 e hospeda a interface web da aplicação.
- **Componentes:**
 - **Index.html:** Arquivo HTML que representa a página principal da interface do sistema.
 - **Style.css:** Arquivo CSS que controla o estilo e a aparência da interface.
 - **Script.js:** Arquivo JavaScript responsável pela interatividade e funcionalidades dinâmicas do front-end.
- **Função:** Exibe a interface para os usuários e trata as solicitações HTTP recebidas dos dispositivos do usuário (smartphone e PC).
- **Conexões:**
 - Estabelece conexões **HTTPS** com dispositivos dos usuários para garantir a segurança da comunicação.
 - Comunica-se com o Servidor de Aplicação e o Servidor de Autenticação para processar e responder às solicitações dos usuários.
 - Conecta-se ao Serviço de Cache via **TCP/IP** para armazenamento temporário de dados.
 - Estabelece conexão **TCP/IP** com o Servidor de Imagens para fornecer imagens na interface.

4. Dispositivos do Usuário (Smartphone e PC)

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

- **Componentes:**
 - **Navegador:** Interface usada pelos usuários para interagir com o sistema.
 - **Java VM (no PC):** Permite a execução de aplicações Java, adicionando funcionalidades adicionais.
- **Função:** Permite que o usuário final acesse o sistema por meio de uma interface amigável, utilizando o navegador como ponto de interação.
- **Conexões:** Realizam conexões **HTTPS** com o Servidor Web para garantir que a comunicação seja segura e os dados do usuário estejam protegidos.

5. Servidor de Aplicação

- **Configuração:** Executa SUSE Linux 15 com Node.js (versão 18.9.0) para processar a lógica de negócios.
- **Componentes:**
 - **Consultas:** Módulo responsável por buscar informações no sistema, como produtos e dados de pedidos.
 - **Carrinho:** Gerencia as operações do carrinho de compras, incluindo adição e remoção de itens.
 - **Pedidos:** Processa os pedidos de compra, incluindo atualizações e controle de status.
- **Função:** Processa as solicitações dos usuários, executa a lógica de negócios e se comunica com o banco de dados para acessar ou atualizar informações.
- **Conexões:**
 - Comunica-se com o Servidor Web e o Servidor de Autenticação via **HTTPS** para segurança da troca de informações.
 - Conecta-se ao Servidor de Banco de Dados e a serviços de apoio como Sistema de Notificação, Gateway de Pagamento e Serviço de Logs e Monitoramento via **TCP/IP**.

6. Servidor de Banco de Dados

- **Configuração:** Executa SUSE Linux 15 com o SGBD MySQL 8.0 para gerenciar e armazenar dados.
- **Componentes:**
 - **Tabela de Clientes:** Armazena informações sobre os clientes do sistema.
 - **Tabela de Produtos:** Contém detalhes sobre os produtos disponíveis.
 - **Tabela de Faturas:** Registra as transações e faturamentos.
- **Função:** Centraliza o armazenamento de dados do sistema, permitindo que o Servidor de Aplicação acesse, insira e atualize informações conforme necessário.
- **Conexões:** Estabelece conexões via **TCP/IP** com o Servidor de Aplicação e outros serviços, como Serviço de Backup e Serviço de Recomendação, para facilitar o fluxo de dados.

7. Serviços de Apoio

PetPlace	Versão: 6.0
Especificação de Requisitos de Software	Data: 31/10/2024
ER-MP01	

- **Serviço de Cache:**
 - **Função:** Armazena temporariamente dados frequentemente acessados para reduzir o tempo de resposta e a carga no banco de dados.
 - **Conexões:** Comunica-se com o Servidor de Aplicação via **TCP/IP** para facilitar o acesso rápido a dados armazenados.
- **Sistema de Notificação:**
 - **Função:** Envia notificações aos usuários sobre eventos importantes, como confirmações de pedidos.
 - **Conexões:** Conecta-se ao Servidor de Aplicação para enviar notificações em tempo real.
- **Gateway de Pagamento:**
 - **Função:** Gerencia transações financeiras e integrações com serviços de pagamento externos para realizar transações de forma segura.
 - **Conexões:** Comunica-se com o Servidor de Aplicação para processar e confirmar transações de pagamento.
- **Serviço de Logs e Monitoramento:**
 - **Função:** Armazena registros de atividades e monitora o desempenho do sistema, permitindo diagnósticos e auditorias.
 - **Conexões:** Conecta-se ao Servidor de Aplicação e ao Servidor de Banco de Dados via **TCP/IP** para registrar e monitorar atividades.
- **Serviço de Backup:**
 - **Função:** Realiza backups periódicos dos dados armazenados, garantindo a segurança e recuperação em caso de falhas.
 - **Conexões:** Conecta-se ao Servidor de Banco de Dados via **TCP/IP** para efetuar cópias de segurança dos dados críticos.
- **Serviço de Recomendação:**
 - **Função:** Fornece recomendações personalizadas aos usuários, com base em suas interações e histórico de compras.
 - **Conexões:** Conecta-se ao Servidor de Aplicação e ao Servidor de Banco de Dados para acessar dados necessários e gerar recomendações personalizadas.