



YouiDraw YouiDraw YouiDraw YouiDraw YouiE

BiteP

Sistema de Adoção de Pets



UNICENTRO

Campus CEDETEG

DECOMP -Departamento de Ciência da Computação

R. Simeão Varela de Sá, 03 - Vila Carli

Guarapuava - PR, 85040-080

Equipe:

João Gabriel de Deus

Julia Rodrigues Nascimento Mattos

Leonardo Elnisky

Thomas Nichay Ferreira

04.07.2018

Resumo

Atualmente há milhares de animais precisando de adoção por vários motivos, e esse número só tende a crescer. A ideia do BiteP surgiu visando resolver este problema. O software permite o cadastro de animais para adoção através do perfil de seu guardião atual e a adoção destes animais por um usuário comum. Esta documentação apresenta o processo de desenvolvimento, incluindo testes e implementação do sistema em um ambiente web. O sistema foi desenvolvido buscando atender qualquer pessoa que procura um companheiro de quatro patas.

1. INTRODUÇÃO

Este software tem como objetivo facilitar a conexão entre usuários para a adoção de animais de estimação como gatos e cachorros. Os clientes que utilizarão esta aplicação são qualquer um com acesso à internet, e terão um acesso simplificado as funcionalidades deste.

Os usuários terão a facilidade de doar ou buscar uma adoção de um pet sem sair de casa, usando um navegador. Será possível buscar em um catálogo de variedades de raças e outras características em uma cidade específica.

2. REQUISITOS

Esta seção tem como objetivo fazer a descrição dos requisitos e tem como meta descrevê-los de forma completa e consistente. Baseado nos ideais do usuário, cada requisito traz junto um conjunto de informações que podem ser alteradas durante o desenvolvimento do projeto.

2.1. Requisitos da Plataforma

Para o uso do sistema é necessário um dispositivo que possua um navegador com acesso à Internet já que o sistema é responsivo.

2.2. Requisitos do Software

2.2.1. Requisitos de Usuário (Visão de Usuário)

1. Cadastro de Usuário: É necessário que exista uma função para cadastrar as informações de um usuário para que este possa cadastrar seus pets;
2. Cadastro/Doação de Pets: É necessário que exista uma função para que o usuário cadastrado possa cadastrar seus pets para adoção, todo pet cadastrado é disponibilizado para outros usuários;
3. Pesquisar Pets: É necessário que exista uma função para buscar pets cadastrados no sistema;
4. Adotar Pets: É necessário que exista uma função que permita que usuários se conectem para negociar a doação/adoção;
5. Remover Pets: É necessário que exista uma função que permita a remoção de um pet cadastrado;
6. Remover Cadastro: É necessário que exista uma função que permita a remoção do cadastro completo do usuário.

Usuário	Cadastro Usuário	Remover Cadastro	Cadastro Pets	Remover Pets	Visualizar Pets	Adotar Pet
Visitante	x				x	
Cadastrado		x	x	x	x	x

2.2.2. Requisitos de Sistema (Visão de Desenvolvedor)

Os requisitos de sistema:

1. Cadastro de Usuário: Serão cadastrados, obrigatoriamente:
 - a. -nome String (40), não permitindo números;
 - b. -email String (255);
 - c. -senha String;
 - d. -telefone int (11), não permitindo letras;
 - e. -cidade String (40).
 - f. -estado String (40).
2. Cadastro/Doação de Pets: Serão cadastrados, obrigatoriamente:
 - a. -nome String (20), não permitindo números;
 - b. -genero char(1), sendo somente M ou F;
 - c. -espécie String (10), não permitindo números;

- d. -raça String (20), não permitindo números;
 - e. -idade int (8), não permitindo letras;
 - f. -observação String (200).
3. Pesquisar Pets: Haverá uma barra de busca para pesquisa com filtros como espécie, idade, raça e cidade.
 4. Adotar Pets: no perfil do pet desejado, há um botão onde o usuário pode solicitar acesso às informações pessoais do guardião para que possam efetuar a adoção.
 5. Remover Pets: Será exibida uma janela de confirmação e, caso confirmado, um retorno de que o pet foi removido do banco de dados.
 6. Remover Cadastro: Será exibida uma janela de confirmação e, caso confirmado, um retorno de que o usuário foi removido do banco de dados.

3. CRONOGRAMA

Segundo Sommerville [1] o cronograma de projeto mostra as dependências entre as atividades, a estimativa de tempo para cada atividade e a alocação das pessoas para as atividades.

Como parte do protocolo de formalização deste projeto, foi desenvolvido um cronograma que prevê tarefas de requisitos, projeto, codificação, implantação, ajustes de código, testes, validação e relatório final. A Figura 3.1 apresenta o Gráfico de Gantt do cronograma onde são mostradas as atividades, data inicial, duração e data final. São mostrados também a ocorrência de cada tarefa no espaço temporal do cronograma bem como as interdependências entre as atividades.

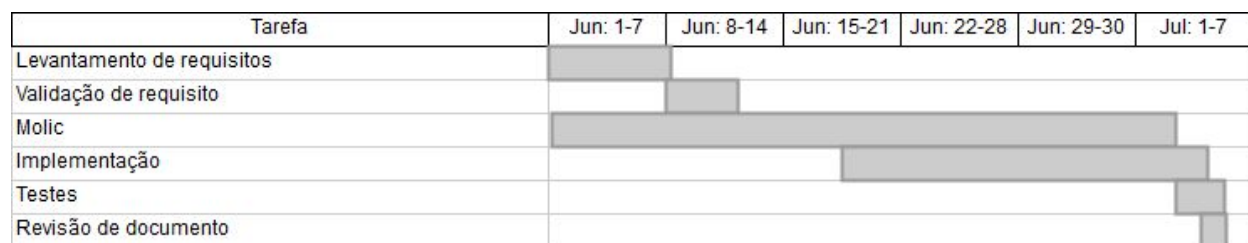


Figura 3.1 - Cronograma do projeto BiteP

4. PROJETO

Para Sommerville [1] O projeto descreve a estrutura do software a ser implementado, os modelos e estruturas de dados usados pelo sistema, as interfaces entre os componentes do sistema.

Através de uma abordagem iterativa, onde, a cada iteração, e por meio de revisões, os projetistas acrescentam formalidade e detalhes que refinam o projeto do software até chegar a um projeto final ideal.

A Figura 4.1 é um modelo abstrato de processo que contém as entradas para o processo de projeto, suas atividades e os documentos produzidos como saídas dele. Esta abordagem, sugerida por Sommerville [1], foi usada para o desenvolvimento deste projeto integrador.

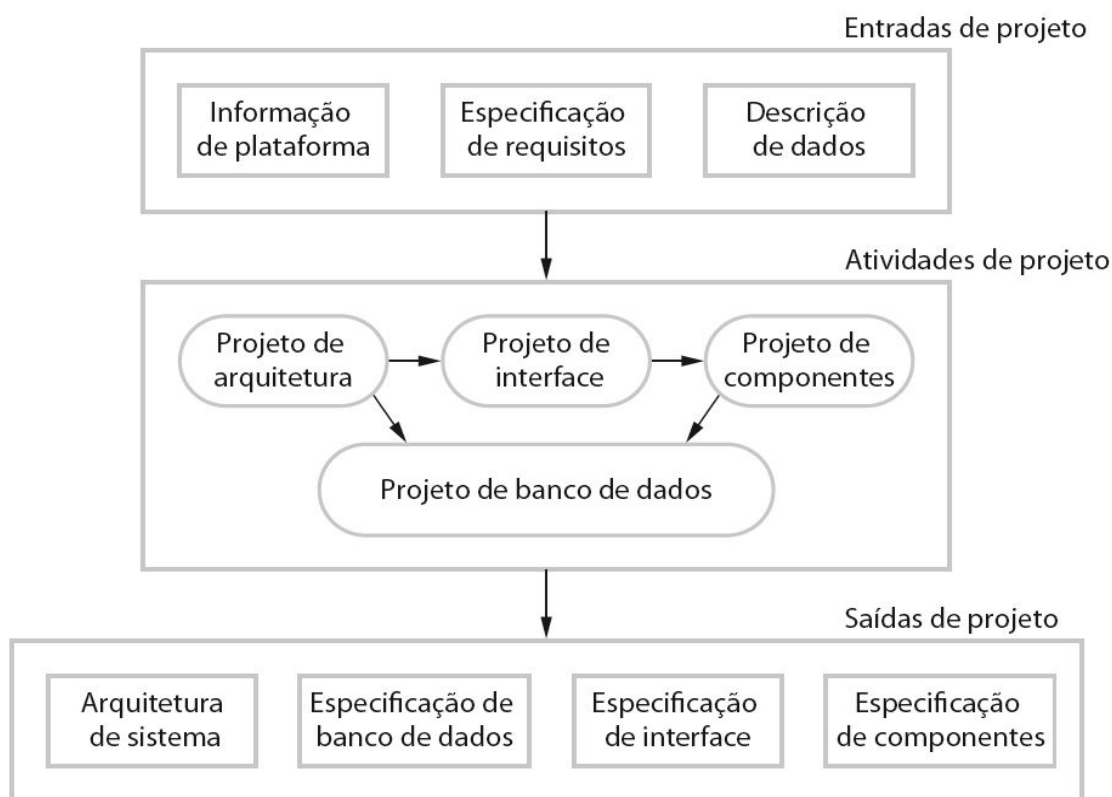


Figura 4.1 - Um modelo geral do processo de projeto [1]

4.1. Casos de Uso

O Diagrama de Casos de Uso, como visto na Figura 4.2, tem o objetivo de auxiliar a comunicação entre os analistas e o cliente. Um diagrama de Caso de Uso descreve um cenário que mostra as funcionalidades do sistema do ponto de vista do usuário. O cliente deve ver no diagrama de Casos de Uso as principais funcionalidades de seu sistema.[2]

4.1.1. Diagrama(s) de casos de uso

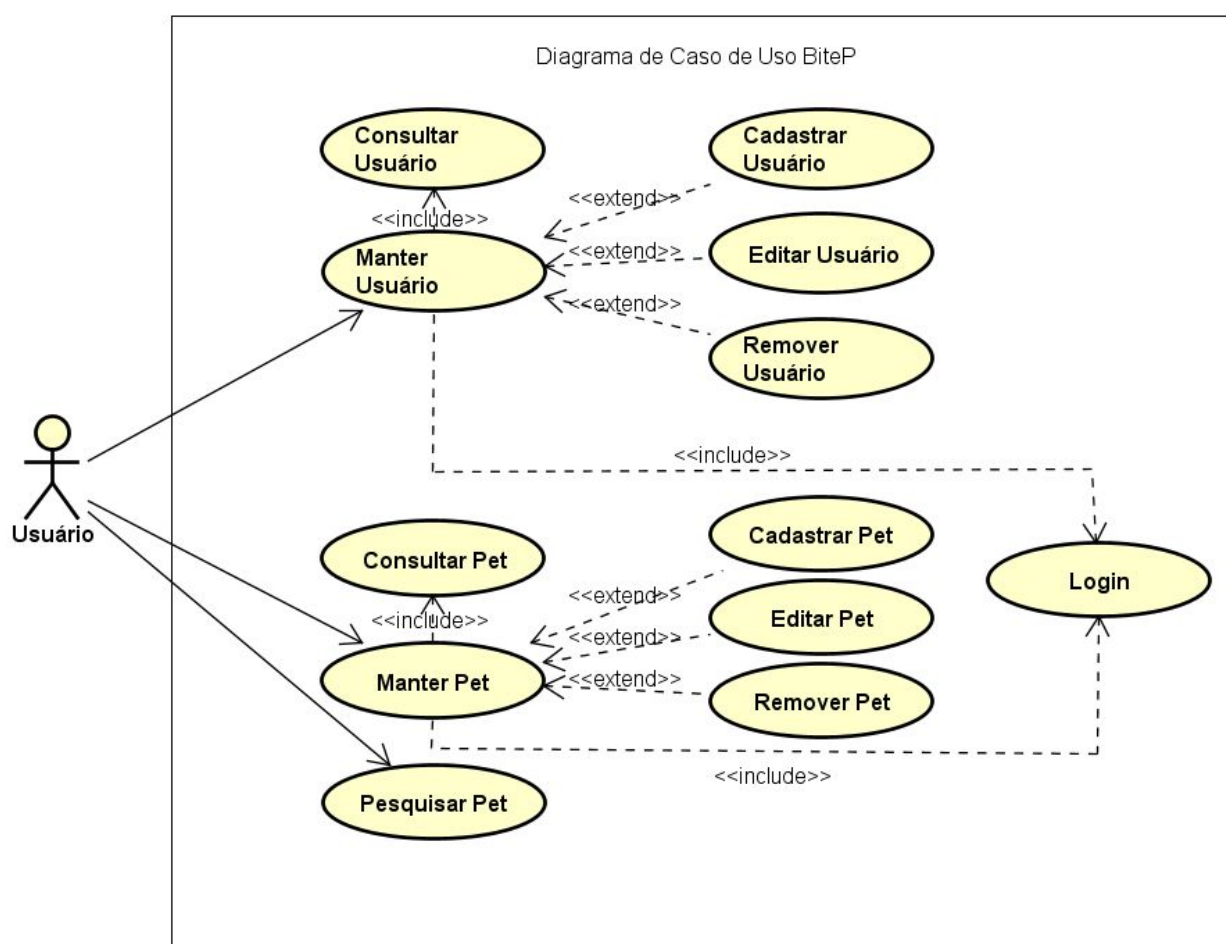


Figura 4.2 - Casos de uso do sistema BiteP

4.1.2. Descrição dos casos de uso

Os casos de uso tem como objetivo auxiliar na comunicação entre analistas e clientes, deixando claro através de um cenário as funcionalidades

do sistema a ser desenvolvido. Para que essa comunicação fique mais clara e a validação de requisitos se torne mais concreta, foram introduzidas as tabelas descritivas.

- Login: O usuário irá se conectar baseado em sua conta no sistema, para ter acesso às suas funções.
- Consultar Usuário: O sistema irá consultar o banco de dados em busca de um usuário.
- Cadastrar Usuário: O usuário poderá colocar suas informações para obter um perfil personalizado para si.
- Editar Usuário: O usuário poderá atualizar as informações pessoais.
- Remover Usuário: O usuário poderá excluir seu perfil da base de dados.
- Consultar Pet: O sistema irá consultar o banco de dados em busca de um pet.
- Cadastrar Pet: O usuário irá cadastrar seu Pet para ser adotado por outros usuários do sistema.
- Editar Pet: O usuário poderá alterar as informações de um Pet seu.
- Remover Pet: O usuário irá remover o Pet cadastrado da base de dados para adoção.
- Pesquisar Pets: O usuário irá fazer uma busca pelos Pets que estão na base de dados.

Tabela 4.1 - Caso de Uso Login

Nome do Caso de Uso	Login
Caso de Uso Geral	
Ator Principal	Usuário.
Atores Secundários	
Resumo	Este caso de uso descreve as etapas percorridas por um usuário para realizar o login.
Pré-condições	O usuário precisa possuir uma conta já cadastrada.
Pós-condições	
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Selecionar a opção de login.	
	2. Carregar/exibir interface de preenchimento dos dados .
3. Inserir dados do login.	
	4. Validar os dados preenchidos.
Fluxo Alternativo 1	
	5. Logar no site.
Fluxo Alternativo 2	
	5. Exibir mensagem de erro.

Tabela 4.2 - Caso de Uso Consultar Usuário

Nome do Caso de Uso	Consultar Usuário
Caso de Uso Geral	
Ator Principal	Sistema.
Atores Secundários	
Resumo	Este caso de uso descreve o trabalho do sistema em verificar a existência de um cadastro de um usuário.
Pré-condições	
Pós-condições	
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Confirmação do cadastro de um usuário.	
	2. Consultar o banco de dados e verificar se existe um usuário já cadastrado com os mesmos dados.
Fluxo Alternativo 1	
	3.Retornar mensagem de autorização de cadastro.
Fluxo Alternativo 2	
	3. Exibir mensagem de usuário já existente.

Tabela 4.3 - Caso de Uso Cadastrar Usuário.

Nome do Caso de Uso	Cadastrar Usuário		
Caso de Uso Geral			
Ator Principal	Usuário visitante.		
Atores Secundários			
Resumo	Este caso de uso descreve as etapas percorridas por um usuário para realizar o seu cadastro no site de adoção.		
Pré-condições			
Pós-condições			
Fluxo Principal			
Ações do Ator		Ações do Sistema	
1. Selecionar a opção de cadastro no menu.			
		2. Carregar/exibir interface de preenchimento dos dados .	
3. Inserir dados do usuário.			
		4. Validar os dados preenchidos .	
Fluxo Alternativo 1			
		5. Exibir confirmação de cadastro.	
Fluxo Alternativo 2			
		5. Exibir mensagem de erro.	
Restrições / Validações	Dados de preenchimento obrigatório.		

Tabela 4.4 - Caso de Uso Editar Usuário

Nome do Caso de Uso	Editar Usuário	
Caso de Uso Geral		
Ator Principal	Usuário cadastrado.	
Atores Secundários		
Resumo	Este caso de uso descreve as etapas percorridas por um usuário para editar as informações do seu perfil.	
Pré-condições	O usuário precisa estar logado no sistema.	
Pós-condições		
Fluxo Principal		
Ações do Ator	Ações do Sistema	
1. Visualizar perfil.		
	2. Carregar/exibir interface do perfil.	
3. Selecionar a opção de edição.		
	3. Carregar/exibir interface com dados para edição.	
4. Fazer as alterações desejadas e clicar no botão OK para confirmar.		
	5. Validar dados preenchidos.	
Fluxo Alternativo 1		
	6. Exibir confirmação de alteração.	
Fluxo Alternativo 2		
	6. Exibir mensagem de erro.	
Restrições / Validações	Dados de preenchimento obrigatório.	

Tabela 4.5 - Caso de Uso Remover Usuário

Nome do Caso de Uso	Remover Usuário		
Caso de Uso Geral			
Ator Principal	Usuário cadastrado.		
Atores Secundários			
Resumo	Este caso de uso descreve as etapas percorridas por um usuário para remover seu cadastro do site.		
Pré-condições	O usuário precisa estar logado no site.		
Pós-condições			
Fluxo Principal			
Ações do Ator		Ações do Sistema	
1. Visualizar perfil.			
		2. Carregar/exibir interface do perfil.	
3. Selecionar a opção de remover conta.			
		4. Informar o usuário da escolha e pedir confirmação.	
5. Confirmar.			
		6. Remover conta.	
Restrições / Validações	Verificar se a escolha foi intencional.		

Tabela 4.6 - Caso de Uso Consultar Pet

Nome do Caso de Uso	Consultar Pet		
Caso de Uso Geral			
Ator Principal	Sistema.		
Atores Secundários			
Resumo	Este caso de uso descreve o trabalho do sistema em verificar a existência de pets cadastrados no banco de dados.		
Pré-condições			
Pós-condições			
Fluxo Principal			
Ações do Ator		Ações do Sistema	
		1. Consultar o banco de dados e verificar pets já cadastrados para exibição.	
Fluxo Alternativo 1			
		2. Retornar pets cadastrados.	
Fluxo Alternativo 2			
		2. Exibir mensagem de erro.	

Tabela 4.7 - Caso de Uso Cadastrar Pet

Nome do Caso de Uso	Cadastrar Pet		
Caso de Uso Geral			
Ator Principal	Usuário cadastrado.		
Atores Secundários			
Resumo	Este caso de uso descreve as etapas percorridas por um usuário para realizar o cadastro do Pet no site de adoção.		
Pré-condições	O usuário precisa estar logado no sistema.		
Pós-condições			
Fluxo Principal			
Ações do Ator		Ações do Sistema	
1. Selecionar a opção de cadastro de pet no menu.			
		2. Carregar/exibir interface de preenchimento de dados.	
3. Preencher dados do pet a ser cadastrado.			
		4. Validar os dados preenchidos.	
Fluxo Alternativo 1			
		5. Exibir confirmação de cadastro.	
Fluxo Alternativo 2			
		5. Exibir mensagem de erro.	
Restrições / Validações	Dados de preenchimento obrigatório.		

Tabela 4.7 - Caso de Uso Editar Pet

Nome do Caso de Uso	Editar Pet
Caso de Uso Geral	
Ator Principal	Usuário cadastrado.
Atores Secundários	
Resumo	Este caso de uso descreve as etapas percorridas por um usuário para editar informações no perfil de um pet.
Pré-condições	O usuário precisa estar logado no sistema e também deve ter um pet já cadastrado em seu perfil.
Pós-condições	
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Visualizar a interface de meus pets.	
	2. Carregar pets cadastrados no perfil do usuário.
3. Selecionar o pet desejado para alteração.	
	4. Carregar/exibir interface com dados do pet selecionado para edição.
5. Fazer as alterações desejadas e confirmar.	
	6. Validar dados preenchidos.
Fluxo Alternativo 1	
	7. Exibir confirmação de alteração.
Fluxo Alternativo 2	
	7. Exibir mensagem de erro.
Restrições / Validações	Dados de preenchimento obrigatório.

Tabela 4.8 - Caso de Uso Remover Pet

Nome do Caso de Uso	Remover Pet	
Caso de Uso Geral		
Ator Principal	Usuário cadastrado.	
Atores Secundários		
Resumo	Este caso de uso descreve as etapas percorridas por um usuário para remoção do registro de pet do site.	
Pré-condições	O usuário precisa estar logado no sistema e também deve ter um pet já cadastrado em seu perfil.	
Pós-condições		
Fluxo Principal		
Ações do Ator		Ações do Sistema
1. Selecionar o pet desejado para remoção.		
		2. Carregar/exibir perfil do pet selecionado.
3. Selecionar a opção de remoção do pet.		
		4. Informar usuário da escolha e pedir confirmação da ação.
5. Confirmar.		
Fluxo Alternativo 1		
		6. Exibir confirmação de exclusão.
Fluxo Alternativo 2		
		6. Exibir mensagem de erro.
Restrições / Validações	Verificar se a escolha foi intencional.	

4.2. Interação Humano-Computador

A área de IHC estuda os fenômenos de comunicação entre pessoas e sistemas computacionais e “envolve todos os aspectos relacionados com a interação entre usuários e sistemas”[3].

4.2.1. Perfis de Usuário

- Usuário Cadastrado: pode visualizar animais para adoção, cadastrar animais para doação, editar o perfil de animais já cadastrados em seu perfil, deletar animais cadastrados em seu perfil, buscar por animais na ferramenta de pesquisa e editar seu próprio perfil.
- Usuário Visitante: pode visualizar a página inicial, fazer busca por Pets e cadastrar-se.

4.2.2. Personas

- Persona 1: Robertinho, 9 anos, estuda de manhã e joga Minecraft o dia inteiro, sua mãe gostaria que ele a ajudasse a cuidar melhor de seu cachorro, Bolota, pois o cachorro é muito indisciplinado e por não gastar sua energia acaba destruindo a casa. Robertinho não tem paciência com o animal pois ainda é muito novo e sua mãe que trabalha o dia inteiro gostaria de doar o cachorro pois não tem tempo de cuidar dele e limpar suas artimanhas sozinha.
- Persona 2: Maria, 25 anos, formada em Medicina Veterinária com especialização em animais de pequeno porte. É cirurgiã chefe em sua clínica e possui uma rotina apertada com compromissos ocupando seu tempo além das horas comerciais. Maria é familiarizada com tecnologia, pois possui um sistema de gerência em sua clínica. Gosta de passar sua folga em casa descansando assistindo Netflix e sente falta de uma companhia, há algum tempo procura um animalzinho para que a casa pareça menos vazia, mas como passa muito tempo trabalhando gostaria de um pet um pouco mais independente.
- Persona 3: Angelo, 72 anos, aposentado e viúvo, trabalhou como policial sua vida inteira e teve sua audição prejudicada durante uma ocorrência. Desde que sua mulher veio a falecer, ele tem contado com a ajuda de familiares em suas atividades diárias que são afetadas pela falta de audição. Teve pouco contato com tecnologia e possui dificuldade lidando com a área. Angelo soube da existência de cães-ouvintes e desde então vem buscando um animalzinho com este perfil para que o mesmo lhe ajude a ser mais independente.

4.2.3. Perguntas

1. Como pesquiso por um animal?
2. Como faço o cadastro no sistema?
3. Como faço login no sistema?
4. Como posso adotar um animal?
5. Como posso doar um animal?
6. Como posso cancelar uma doação?
7. Como posso remover o meu perfil do sistema?

4.2.4. Cenário (contendo personas, perguntas e signos)

- **Cenário 1:** Robertinho tinha um cão chamado Bolota, numa bela tarde ensolarada sua mãe o pressionou para encontrar alguém que adotasse o Bolota. Então ele saiu do Minecraft, foi para a Internet buscar por um site de adoção e encontrou o BiteP. A partir da tela inicial, há um botão para o cadastro e login de usuário, ao clicar neste botão, Robertinho foi redirecionado a uma tela de login onde poderia inserir seu email e senha[3], mas como ainda não possui conta, clicou no botão de cadastro oferecido na tela de login e criou sua conta preenchendo os campos com seus dados pessoais como: nome, email, telefone e estado, cidade, e senha[2]. Depois de criar sua conta e estar logado, Robertinho foi até Cadastrar Pets, preencheu os dados do Bolota, sendo nome, espécie, gênero, raça, data de nascimento, observações e imagens, confirmou no botão Confirmar e então Bolota está pronto para ser doado[5]. Após a doação do pet ser finalizada, ele gostaria de deletar seu perfil do sistema. Para isso, clicou em Meu Perfil, selecionou a opção Excluir[7] e confirmou a ação. Fechou o navegador e voltou para o Minecraft.
- **Cenário 2:** Maria já possui um cadastro no site, pois é uma veterinária informada. Para satisfazer seu objetivo de preencher o lar, resolve adotar um gatinho e, para tal, acessa o site BiteP. Ao chegar no site, ela faz o login clicando em Entrar/Cadastrar, selecionando a opção Login e inserindo seu e-mail e senha [3]. Voltando a página inicial, ela clica no símbolo da ampolheta escrito Buscar, que lhe permite pesquisar com filtros no site [1]. Escolhe suas preferências, como mostrar somente gatos. Confirmando a pesquisa, é retornado o resultado desta, contendo somente gatos. Olhando com carinho os animais, ela se apaixona por Banguela e clica em sua foto, que a

redireciona para seu perfil. Banguela está para adoção e tem algumas informações, como nome, idade, gênero, raça e observações, como ser bastante tranquilo. Percebendo que se encaixa em seu perfil, Maria coleta as informações de seu dono para entrar em contato [4]. Depois de tudo resolvido, Banguela é adotado por Maria.

- **Cenário 3:** Ângelo não tem muita intimidade com tecnologia, mas mesmo assim ele tenta buscar por um Pet para adoção online. Com algumas dificuldades, acaba por encontrar o BiteP. Após cadastrado no site, confunde-se na busca e acaba por clicar em "Doar Pet" e colocando as informações. Dando por si, realiza que pôs um pet fictício para doar e, percebendo tal erro, clica em seu Avatar, "Meus Pets" e remove o Pet criado sem intenção [6]. Procurando mais um pouco, encontra a ampulheta de busca e realiza uma busca. Angelo então pede ajuda para um filho seu, que o auxilia na busca de seu cão-ouvinte e acaba por encontrar e adotar seu novo animalzinho.

4.2.5. Tabela de Signos

Tabela 4.9 - Signos

Signos	Descrição	Tipo
Entrar	Login do Usuário	Aplicação
Senha	Senha do Usuário	Aplicação
Desconectar	Logout do Usuário	Aplicação
Signo Composto	Usuário	
Signos	Descrição	Tipo
ID	ID do Usuário	Aplicação
Nome	Nome do Usuário	Domínio
Email	Email do Usuário	Domínio
Telefone	Telefone do Usuário	Domínio
Estado	Estado do Usuário	Domínio
Cidade	Cidade do Usuário	Domínio

Signos	Desrição	Tipo
Signo Composto	Pet	
ID	ID do Pet	Aplicação
Imagem	Imagem do Pet	Domínio
Nome	Nome do Pet	Domínio
Espécie	Espécie do Pet	Domínio
Gênero	Gênero do Pet	Domínio
Raça	Raça do Pet	Domínio
Idade	Idade do Pet	Domínio
Observações	Informações extras do Pet	Domínio

4.2.6. Diagrama Hierárquico de Metas (DHM)

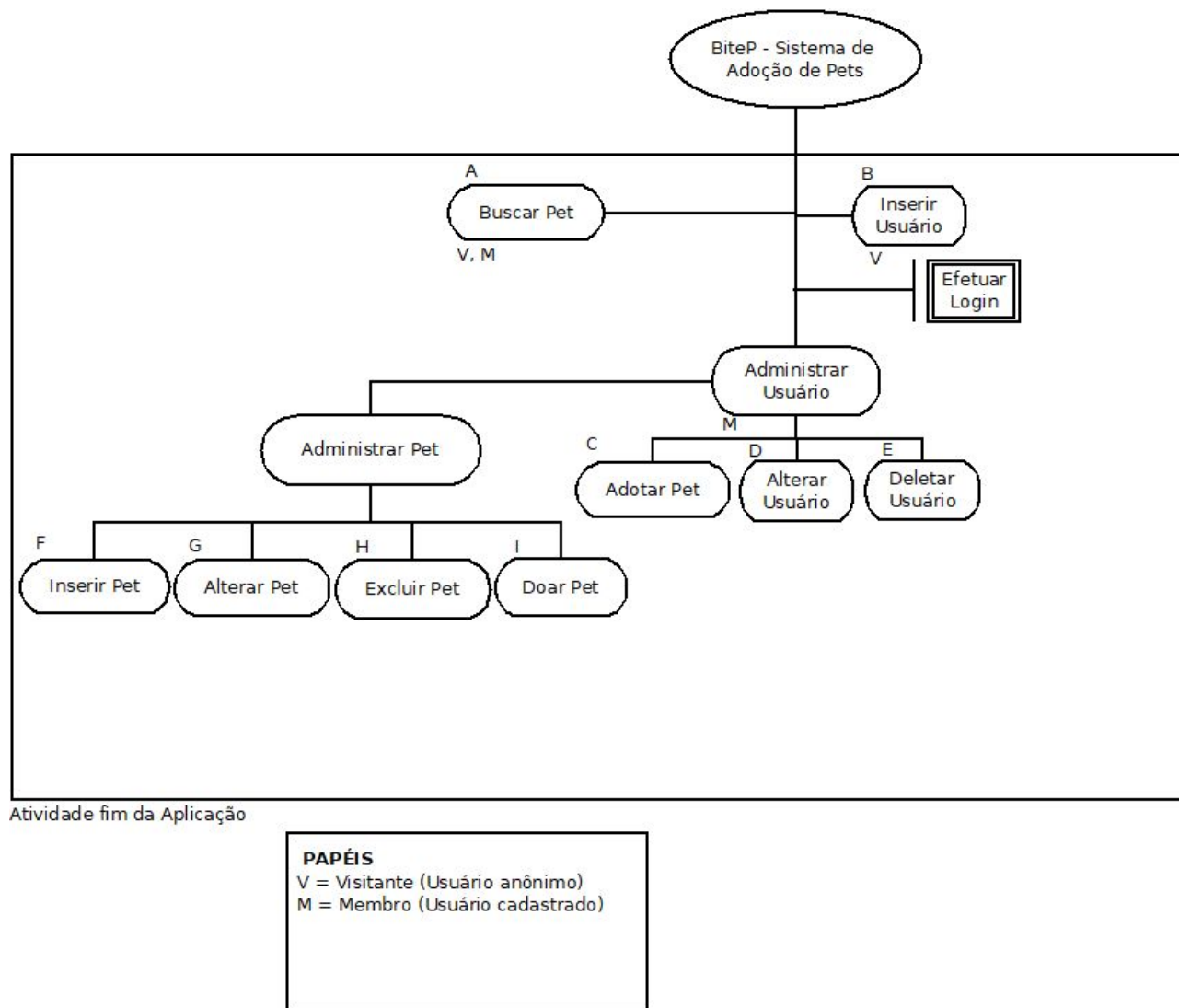


Figura 4.3 - Diagrama Hierárquico de Metas (DHM)

4.2.7. Diagrama Hierárquico de Tarefas (DHT) e especificação textual

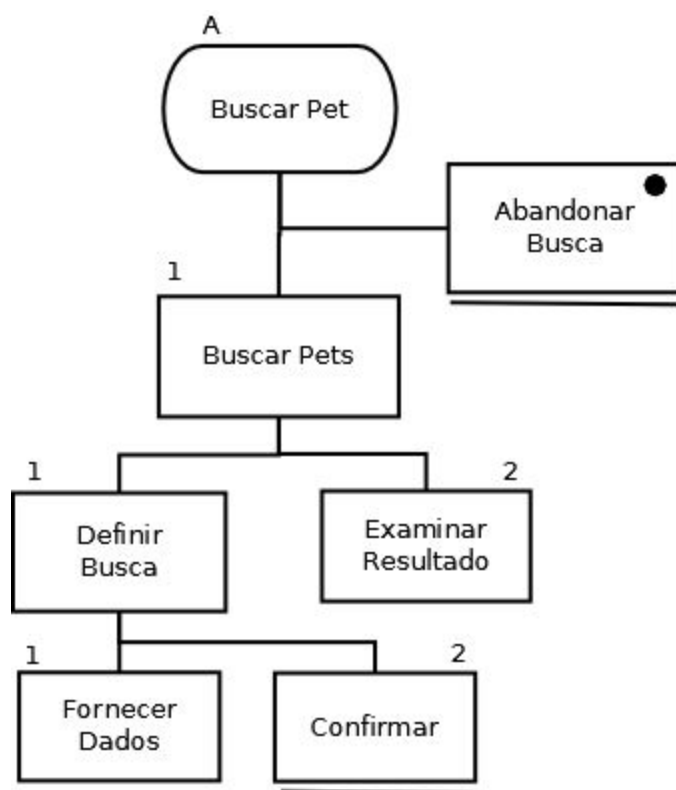


Figura 4.4 - Diagrama Hierárquico da Tarefa Buscar Pet

Buscar Pet

A.1.1.1 Fornecer Dados

SIGNOS: pet.espécie?, pet.raça?, pet.gênero?, pet.idade?

TRATAMENTO APOIADO: falta de informações obrigatórias, dados inconsistentes.

A.1.1.1.2 Confirmar

SIGNOS: pet.buscar().

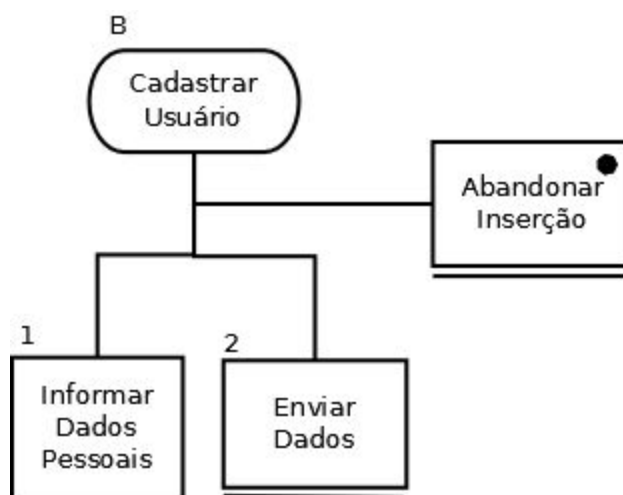


Figura 4.5 - Diagrama Hierárquico da Tarefa Cadastrar Usuário

Cadastrar Usuário

B.1 Informar dados pessoas

SIGNOS: usuário.nome?, usuário.email?, usuário.senha?, usuário.telefone?, usuário.cidade?, usuário.estado?

TRATAMENTO APOIADO: falta de informações obrigatórias, dados inconsistentes.

B.1.2 Enviar dados

SIGNOS: usuário.cadastrar().

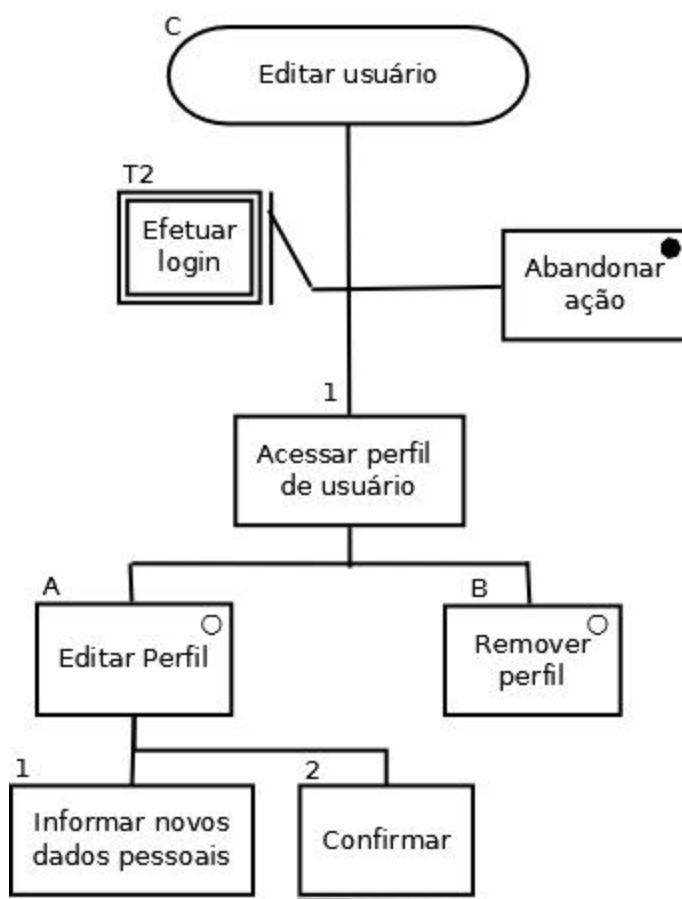


Figura 4.6 - Diagrama Hierárquico da Tarefa Editar Usuário

C.1.A.1 Informar novos dados pessoais

SIGNOS: usuário.nome?, usuário.email?, usuário.senha?, usuário.telefone?, usuário.cidade?, usuário.estado?

TRATAMENTO APOIADO: falta de informações obrigatórias, dados inconsistentes.

C.1.A.1.2 Confirmar

SIGNOS: usuario.editar().

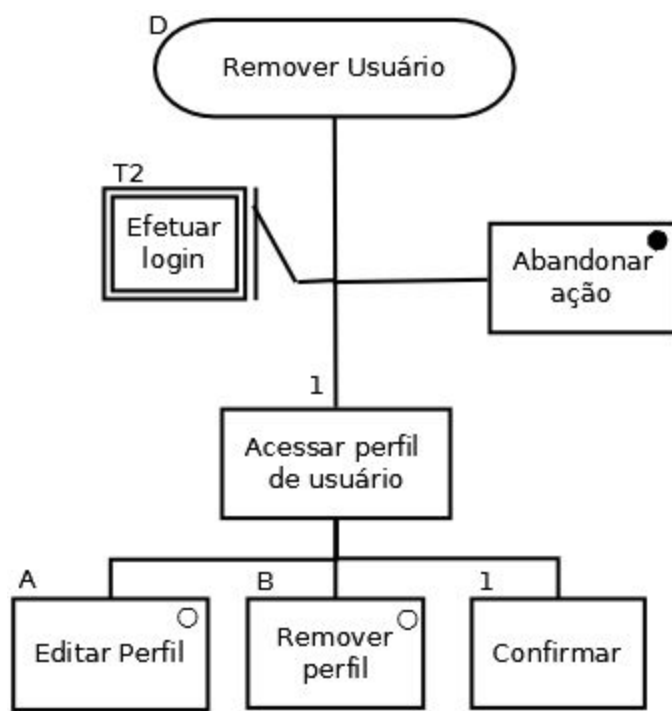


Figura 4.7 - Diagrama Hierárquico da Tarefa Remover Usuário

Remover Usuário

1.B.1 Confirmar

SIGNOS: usuário.remove().

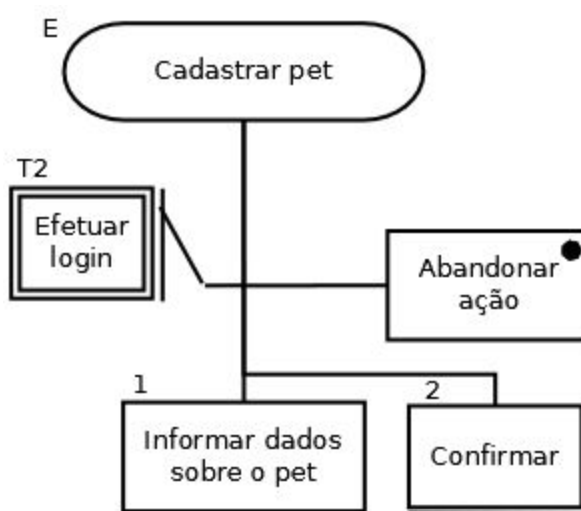


Figura 4.8 - Diagrama Hierárquico da Tarefa Cadastrar Pet

Cadastrar Pet

E.1 Informar dados sobre o pet

SIGNOS: pet.nome?, pet.espécie?, pet.raça?, pet.idade?, pet.observações?, pet.imagens?

TRATAMENTO APOIADO: falta de informações obrigatórias, dados inconsistentes.

E.1.2 Confirmar

SIGNOS: pet.cadastrar().

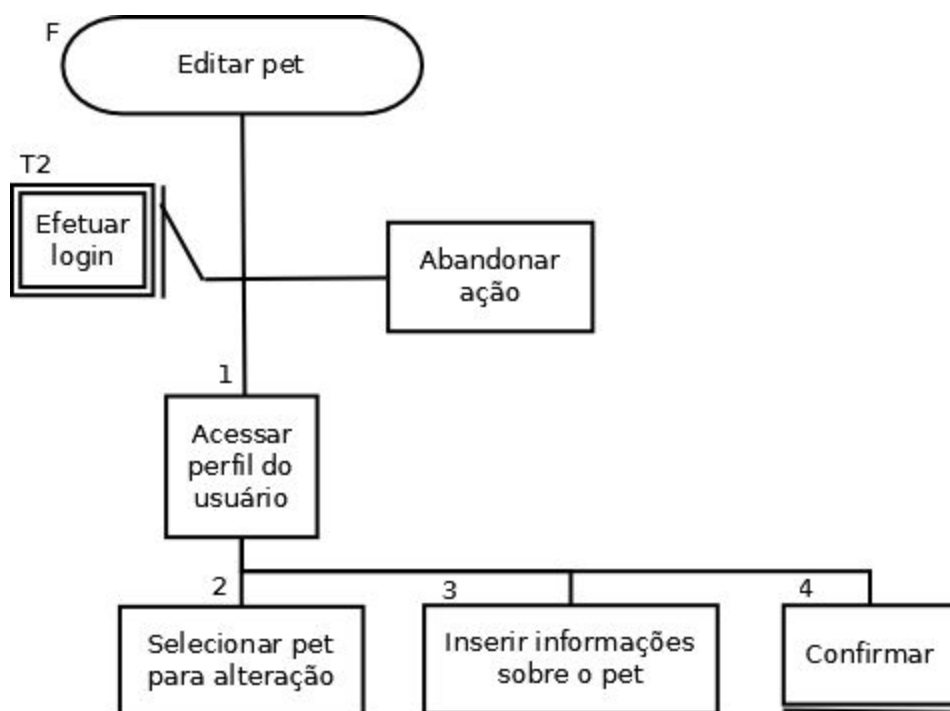


Figura 4.9 - Diagrama Hierárquico da Tarefa Editar Pet

Editar Pet

F.1.2 Selecionar pet para edição

SIGNOS: pet.id?

F.1.2.3 Inserir novas informações sobre o pet

SIGNOS: pet.nome?, pet.espécie?, pet.raça?, pet.idade?, pet.observações? pet.imagens?

TRATAMENTO APOIADO: falta de informações obrigatórias, dados inconsistentes.

F.1.2.3.4 Confirmar

SIGNOS: pet.editar().

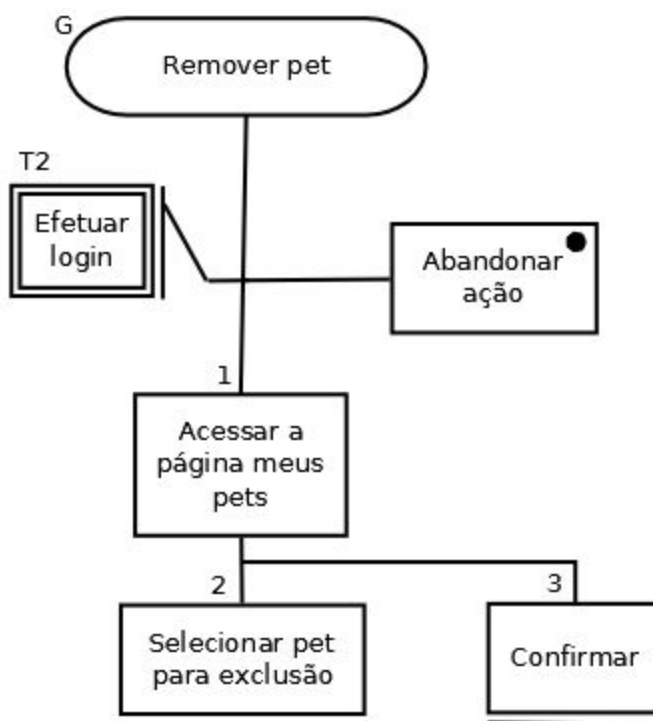


Figura 4.10 - Diagrama Hierárquico da Tarefa Remover Pet

Remover Pet

G.1.2 Selecionar pet para exclusão

SIGNOS: pet.id?

G.1.2.3 Confirmar

SIGNOS: pet.remover().

4.2.8. Diagrama de Interação (DI) e especificação textual

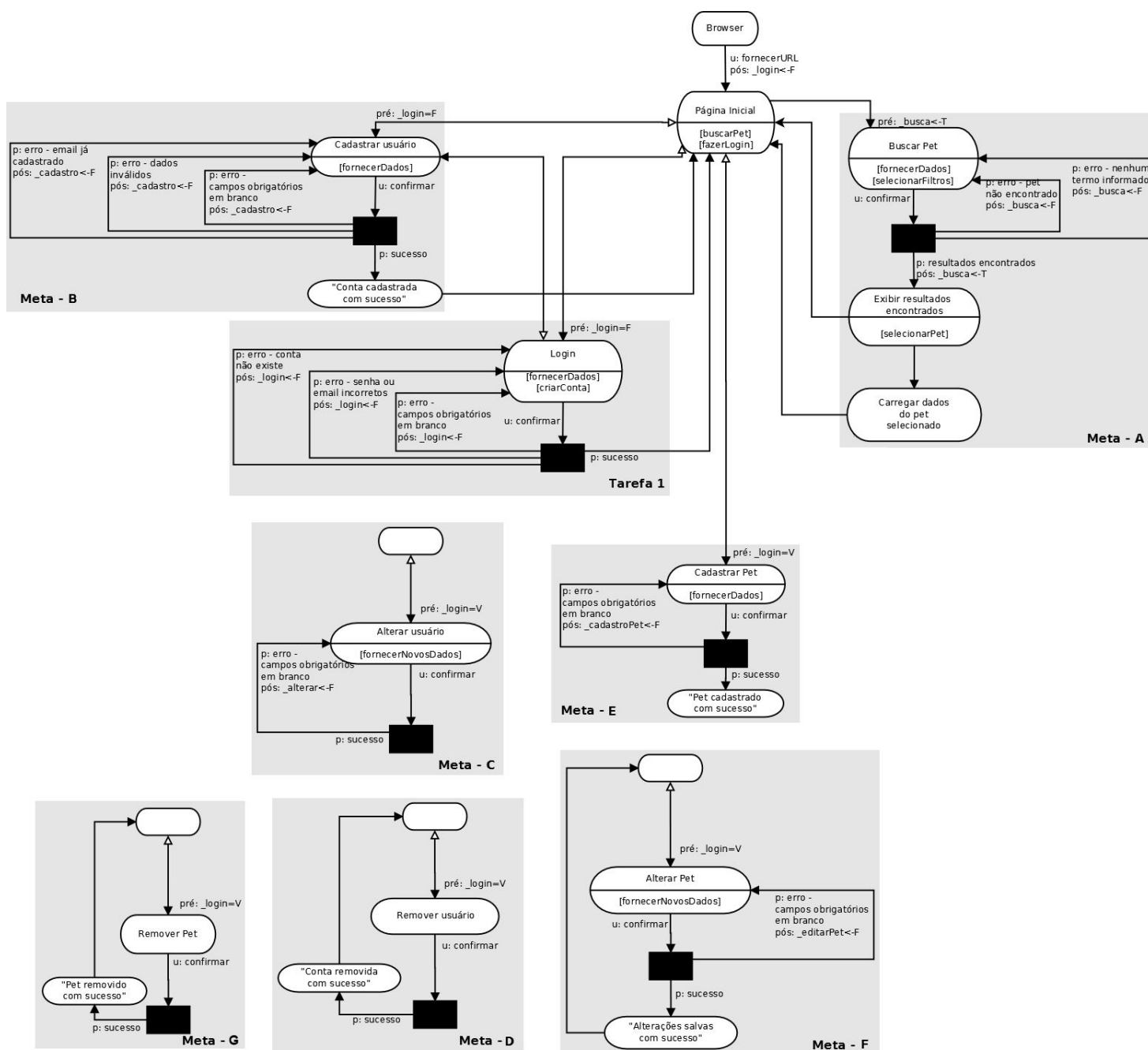



Figura 4.11 - Diagrama de Interação (DI) do Sistema BiteP

Cena: Cadastrar Usuário

A [fornecerDados]

usuário.nome? <texto restrito: obrigatório>

usuário.email? <texto livre: obrigatório>



usuário.senha? <texto livre: obrigatório>
usuário.telefone? <texto restrito: obrigatório>
usuário.cidade? <escolha simples: obrigatório>
usuário.estado? <escolha simples: obrigatório>

Cena: Remover Usuário

Cena: Cadastrar Pet

A [fornecerDados]

pet.nome? <texto livre: obrigatório>
pet.espécie? <texto restrito: obrigatório>
pet.raça? <texto restrito: obrigatório>
pet.idade? <texto restrito: obrigatório>
pet.observações? <texto livre: opcional>
pet.imagens?

Cena: Editar Pet

A [fornecerNovosDados]

pet.nome? <texto livre: obrigatório>
pet.espécie? <texto restrito: obrigatório>
pet.raça? <texto restrito: obrigatório>
pet.idade? <texto restrito: obrigatório>
pet.observações? <texto livre: opcional>
pet.imagens?

Cena: Remover Pet

Cena: Buscar Pet

A [fornecerDados]

pet.nome? <texto livre: opcional>

pet.espécie? <escolha simples: opcional>

pet.raça? <escolha simples: opcional>

pet.idade? <texto restrito: opcional>

B [selecionarFiltros]

Busca.filtros? <escolha múltipla>

4.2.9. Protótipos/Telas de Interface

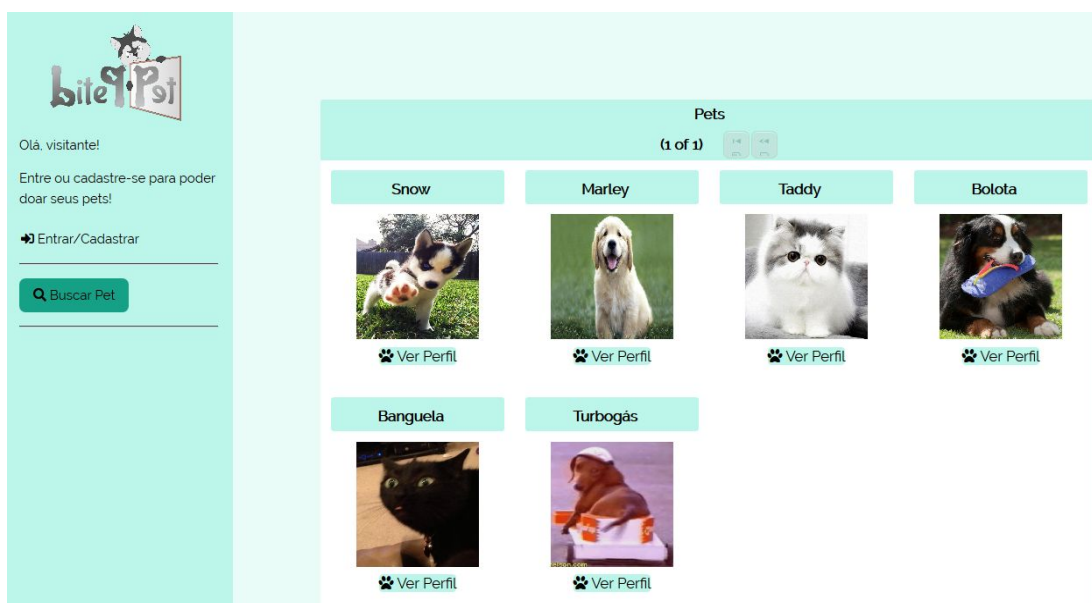


Figura 4.12: Tela inicial do sistema BiteP

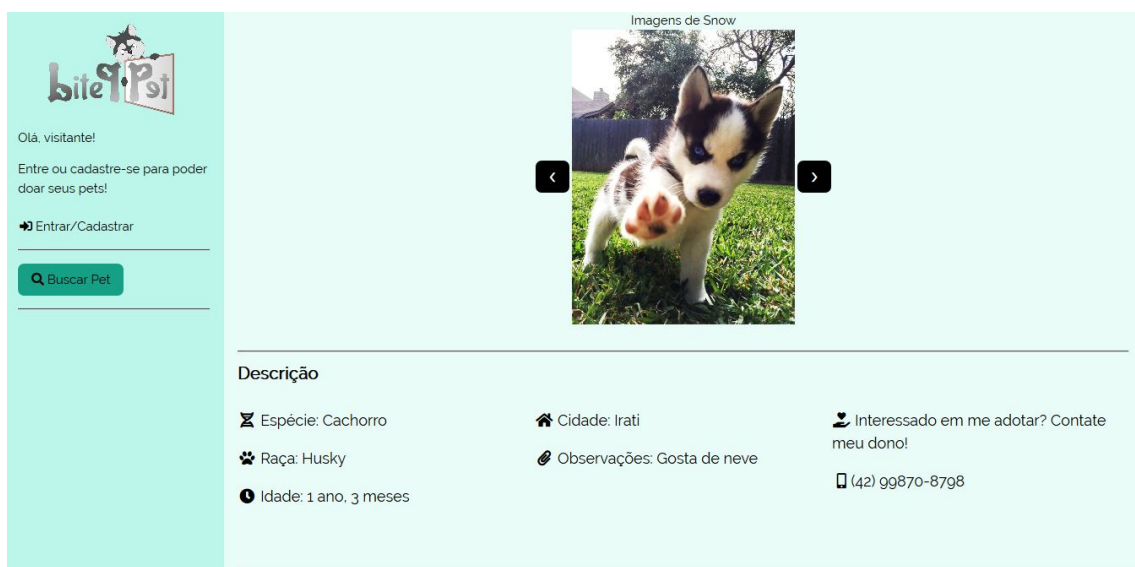


Figura 4.13: Tela Ver Perfil do Sistema BiteP

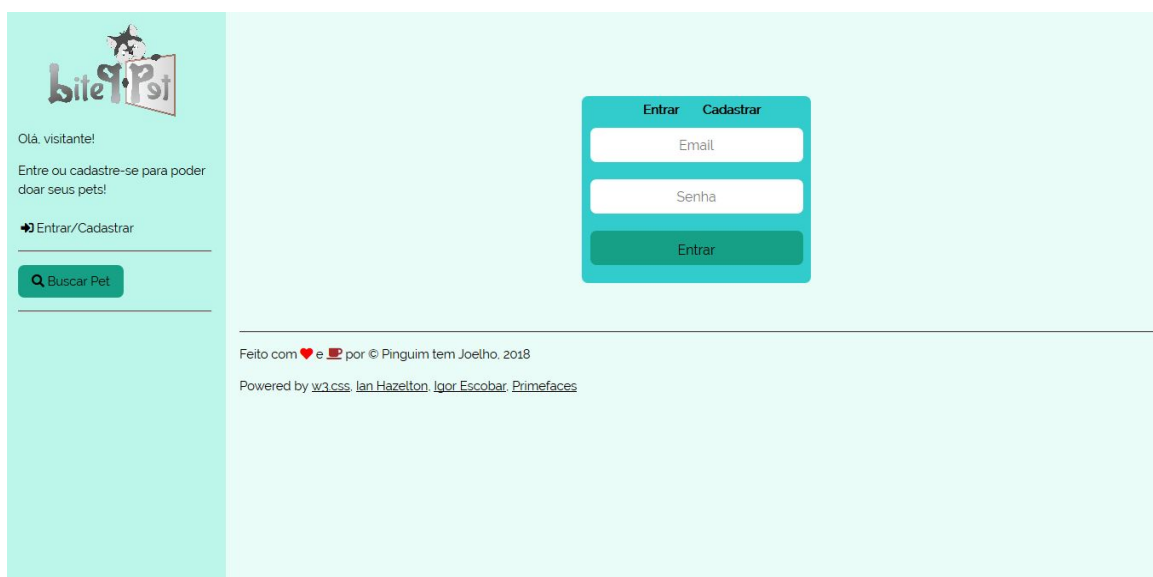


Figura 4.14: Tela Entrar do Sistema BiteP

The screenshot shows the 'Cadastrar' (Register) user screen in the BiteP system. On the left, there is a sidebar with the BiteP logo, a greeting 'Olá, visitante!', a prompt to log in or register, a link to 'Entrar/Cadastrar', a search bar labeled 'Buscar Pet', and a footer with the text 'Feito com ❤ e ☕ por © Pinguim tem Joelho, 2018'. The main content area features a registration form with the following fields: 'Nome' (Name), 'Email', 'Telefone' (Phone), 'Selecionar Estado' (Select State), 'Selecionar Cidade' (Select City), 'Senha' (Password), and 'Repetir Senha' (Repeat Password). A green 'Cadastrar' button is at the bottom of the form.

Figura 4.15: Tela Cadastrar Usuário do Sistema BiteP

The screenshot shows the 'Novo Pet' (New Pet) registration screen in the BiteP system. On the left, the sidebar is updated with 'Bem vindo, Anelle! :)', a link to '+ Cadastrar Pet', 'Meu perfil', 'Meus Pets', a search bar labeled 'Buscar Pet', and a 'Desconectar' link. The main content area features a 'Novo Pet' form with the following fields: 'Nome do Pet' (Pet Name), 'Espécie' (Species) with radio buttons for 'Cachorro' (Dog) and 'Gato' (Cat), 'Gênero' (Gender) with radio buttons for 'Fêmea' (Female) and 'Macho' (Male), 'Raça do Pet' (Pet Breed), 'Data Nascimento' (Date of Birth) with a date picker, and 'Observações' (Observations) with a text area. Below these fields are four buttons labeled 'Imagem 1', 'Imagem 2', 'Imagem 3', and 'Imagem 4' for uploading photos. At the bottom are green buttons for 'Cadastrar' (Register) and 'Cancelar' (Cancel).

Figura 4.16: Tela Cadastrar Pet do Sistema BiteP

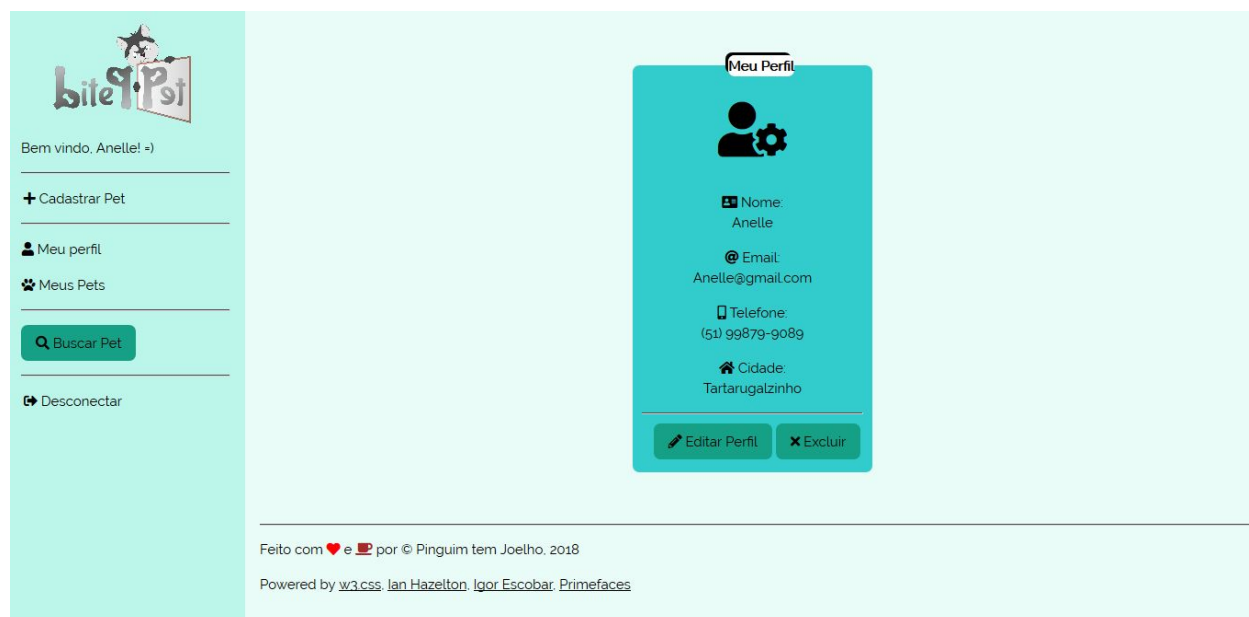


Figura 4.17: Tela Meu Perfil do Sistema BiteP

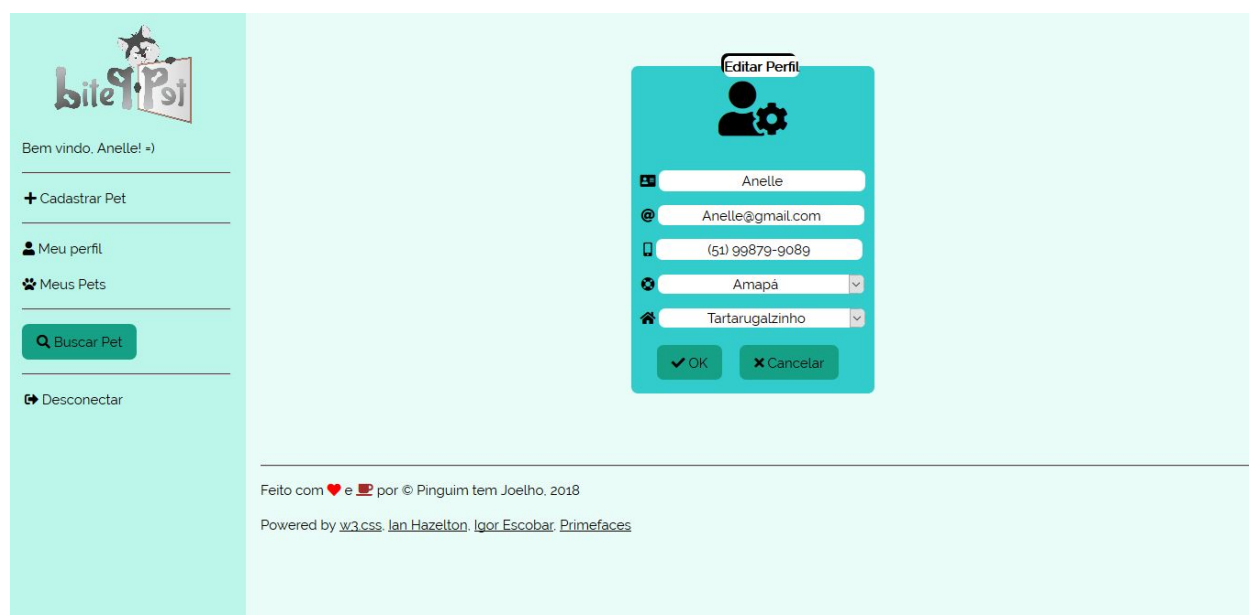


Figura 4.18: Tela Editar Perfil do Sistema BiteP

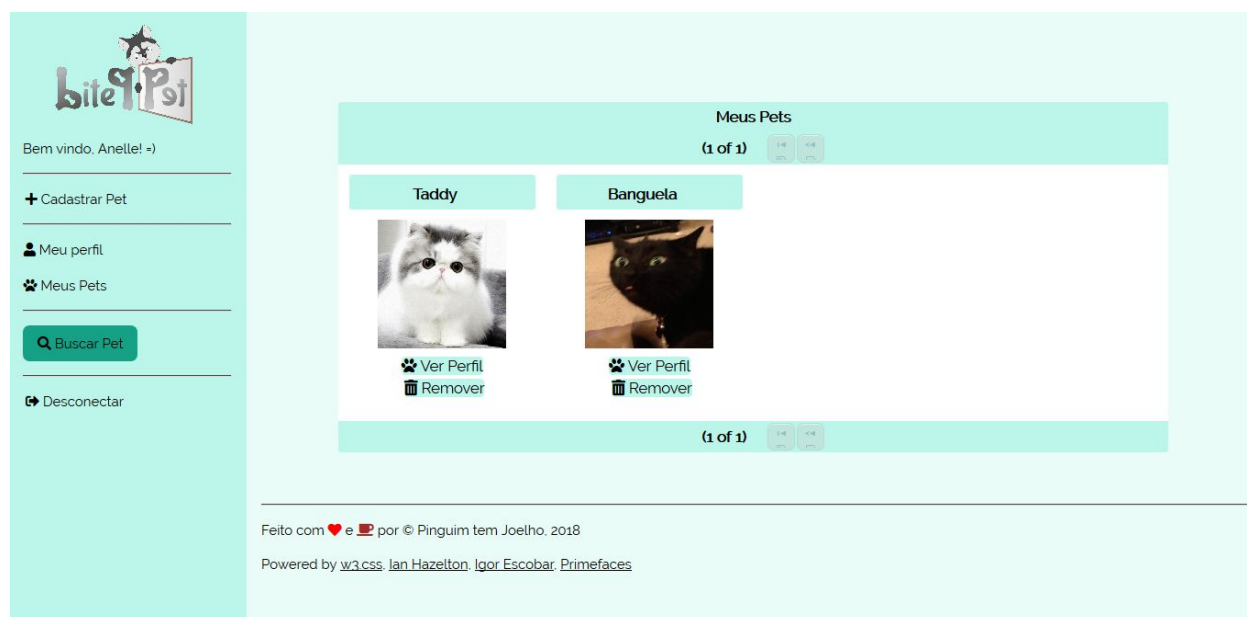


Figura 4.19: Tela Meus Pets do Sistema BiteP

4.2.10. Testes de Interface

4.3. Projeto de Classes e Componentes

Para Sommerville [1] os diagramas de classe são usados para desenvolver um modelo do sistema, na forma orientada a objetos, visando mostrar as classes de um sistema e as associações entre essas classes.

Neste projeto, a modelagem das classes foi feita de duas formas: através dos diagramas que mostram as classes e seus relacionamentos; e através da complementação que é feita pela descrição de cada classe, de seus atributos e de seus métodos.

4.3.1. Diagrama(s) de Classes

Com base nos requisitos levantados, nos casos de uso e nas interfaces, foram criados Diagramas de Classe do sistema BiteP usando a ferramenta de modelagem UML astah Professional.

A Figura 4.10 apresenta uma visão generalizada da arquitetura do Software, em pacotes. A figura 4.11 e 4.12 pode-se visualizar os Diagramas de Classes dos pacotes do sistema de forma expandida e detalhada.

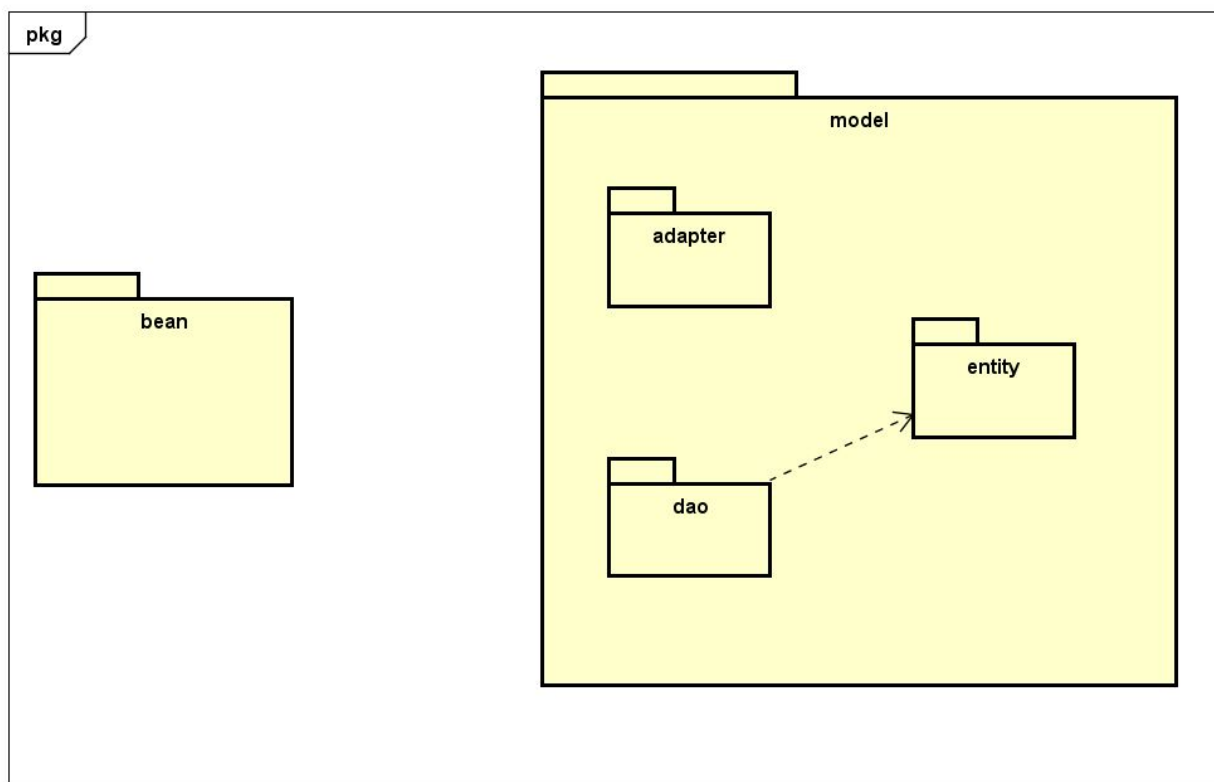


Figura 4.10 - Diagrama de Classes generalizado do sistema.

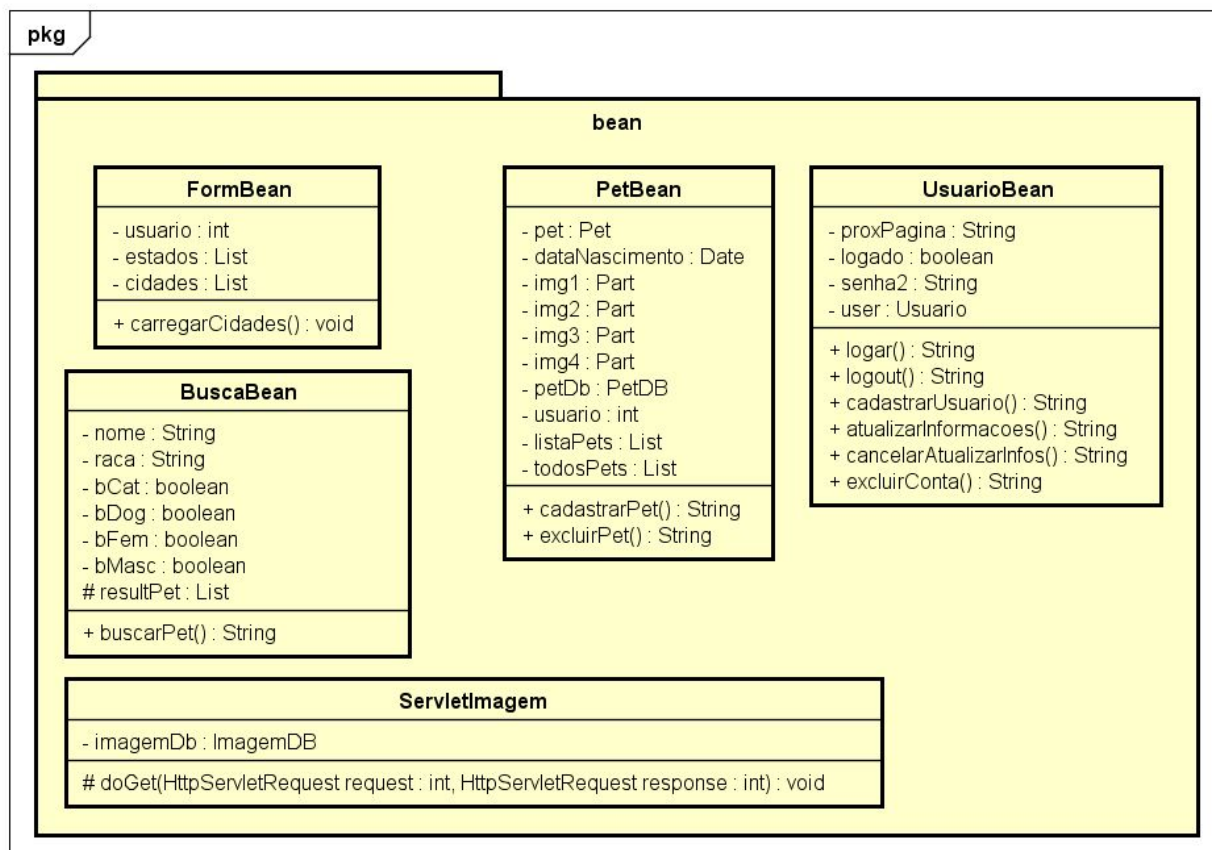


Figura 4.11 - Diagrama de classes do componente (pacote) Bean expandido

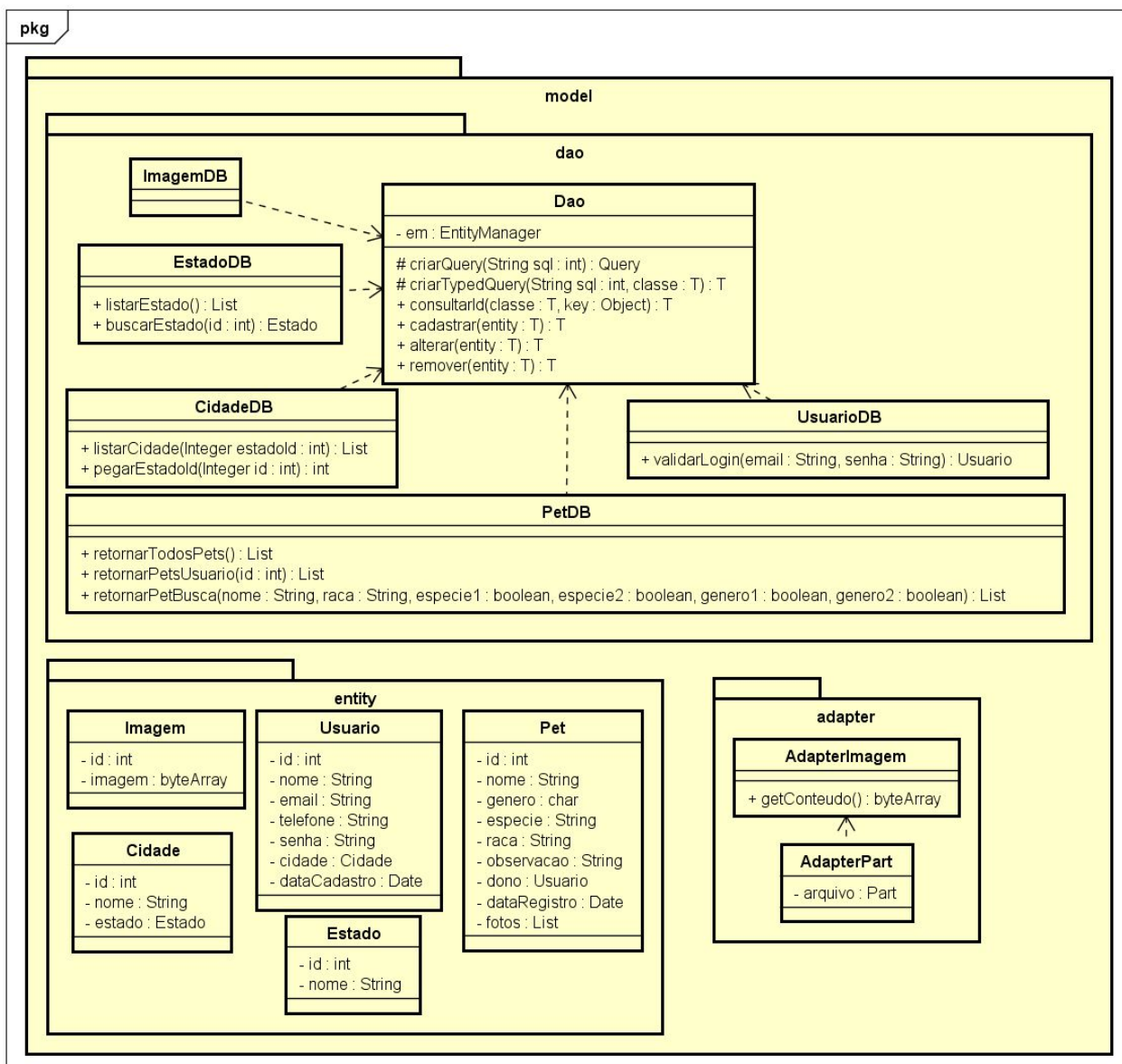


Figura 4.12 - Diagrama de classes do componente (pacote) Model expandido

4.3.2. Descrição das classes

Em projetos com várias pessoas trabalhando é comum se esquecer para que serve cada classe, ou atributo, ou método. Para resolver este problema, foi criada, para cada classe, uma tabela que a descreva. As tabelas 4.1 a 4.18 mostram, cada uma, as características de uma classe do sistema.

Tabela 4.1 - Tabela descritiva da classe FormBean

CLASSE		DESCRIÇÃO DA CLASSE	
FormBean		Esta classe foi criada para controlar os dados de usuário que a view precisa.	
CAMPO	TIPO	TAMANHO	DESCRIÇÃO
usuario	int		Recebe o usuário em sessão.
estados	List	Dinâmico.	Recebe todos os estados armazenados no banco.
cidades	List	Dinâmico.	Recebe todas as cidades armazenadas no banco de acordo com o estado.
MÉTODO		DESCRIÇÃO DO MÉTODO	
carregarCidades()		Busca todas as cidades.	

Tabela 4.2 - Tabela descritiva da classe BuscaBean

CLASSE		DESCRIÇÃO DA CLASSE	
BuscaBean		Esta classe foi criada para gerenciar a busca de pets no sistema.	
CAMPO	TIPO	TAMANHO	DESCRIÇÃO
nome	String	20	Recebe um nome de pet a ser buscado.
raca	String	20	Recebe uma raça de pet a ser buscada.
bCat	boolean		Define se o pet a ser buscado é um gato.
bDog	boolean		Define se o pet a ser buscado é um cão.
bFem	boolean		Define se o pet a ser buscado é fêmea.
bMasc	boolean		Define se o pet a ser buscado é macho.
resultPet	list	Dinâmico.	Lista com os pets que foram encontrados durante a busca.
MÉTODO		DESCRIÇÃO DO MÉTODO	
buscarPet()		Utilizado para buscar um pet cadastrado no banco de dados de acordo com os filtros de pesquisa.	

Tabela 4.3 - Tabela descritiva da classe ServletImagem

CLASSE		DESCRIÇÃO DA CLASSE	
ServletImagem		Esta classe foi criada para ficar responsável no retorno de imagens requisitadas pela view.	
CAMPO	TIPO	TAMANHO	DESCRIÇÃO
imagemDb	ImagemDB		Recebe a imagem requisitada pelo view.
MÉTODO		DESCRIÇÃO DO MÉTODO	
doGet()		Processa requisição de imagem e retorna a imagem diretamente a view sem necessidade de arquivo temporário.	

Tabela 4.4 - Tabela descritiva da classe PetBean

CLASSE		DESCRIÇÃO DA CLASSE	
PetBean		Esta classe foi criada para fazer o controle dos dados do pet.	
CAMPO	TIPO	TAMANHO	DESCRIÇÃO
pet	Pet		Armazena o número identificador do pet.
dataNascimento	Date		Armazena a data de nascimento do pet.
img1	Part		Armazena uma imagem do pet.
img2	Part		Armazena uma imagem do pet.
img3	Part		Armazena uma imagem do pet.
img4	Part		Armazena uma imagem do pet.
petDb	PetDB		Usado para instanciar classes Dao para ter acesso
usuario	int		Armazena o usuário dono do pet.
listaPets	List		Armazena uma lista de pets.
todosPets	List		Armazena uma lista com todos os pets cadastrados.
MÉTODO		DESCRIÇÃO DO MÉTODO	
cadastrarPet()		Utilizado para cadastrar um pet no sistema.	
excluirPet()		Utilizado para remover um pet do sistema.	

Tabela 4.5 - Tabela descritiva da classe UsuarioBean

CLASSE		DESCRIÇÃO DA CLASSE	
UsuarioBean		Esta classe foi criada para controlar os dados de acesso do usuário.	
CAMPO	TIPO	TAMANHO	DESCRIÇÃO
proxPagina	String		Controla a próxima página a ser carregada na view.
logado	boolean		Variável de flag para saber se o usuário iniciou uma sessão.
senha2	String		Variável que armazena a senha do usuário.
user	Usuario		Armazena qual o usuário que está ativo.
MÉTODO		DESCRIÇÃO DO MÉTODO	
logar()		Utilizado para que usuário inicie uma sessão no sistema.	
logout()		Utilizado para que o usuário termina uma sessão no sistema.	
cadastrarUsuario()		Utilizado para cadastrar um usuário.	
atualizarInformacoes()		Utilizado para atualizar as informações do usuário.	
cancelarAtualizarInfos()		Utilizado para interromper a edição de dados do usuário.	
excluirConta()		Utilizado para remover um usuário do sistema.	

Tabela 4.6 - Tabela descritiva da classe ImagemDB

CLASSE		DESCRIÇÃO DA CLASSE	
ImagemDB		Esta classe estende a classe abstrata Dao para utilização de seus métodos de armazenamento.	
CAMPO	TIPO	TAMANHO	DESCRIÇÃO
MÉTODO		DESCRIÇÃO DO MÉTODO	

Tabela 4.7 - Tabela descritiva da classe EstadoDB

CLASSE		DESCRIÇÃO DA CLASSE	
EstadoDB		Esta classe foi criada para armazenar todos os estados existentes no Brasil.	
CAMPO	TIPO	TAMANHO	DESCRIÇÃO
MÉTODO		DESCRIÇÃO DO MÉTODO	
listarEstado()		Utilizado para retornar todos os estados armazenados.	
buscarEstado()		Utilizado para buscar um estado armazenado no banco de dados através de seu número identificador.	

Tabela 4.8 - Tabela descritiva da classe CidadeDB

CLASSE		DESCRIÇÃO DA CLASSE	
CidadeDB		Esta classe foi criada para armazenar todas as cidades existentes no Brasil.	
CAMPO	TIPO	TAMANHO	DESCRIÇÃO
MÉTODO		DESCRIÇÃO DO MÉTODO	
listarCidade()		Utilizado para retornar todas as cidades de um determinado estado.	
pegarEstadoId()		Utilizado para buscar o estado a qual uma cidade pertence.	

Tabela 4.9 - Tabela descritiva da classe PetDB

CLASSE		DESCRIÇÃO DA CLASSE	
PetDB		Esta classe foi criada para o armazenamento de dados do pet no banco.	
CAMPO	TIPO	TAMANHO	DESCRIÇÃO
MÉTODO		DESCRIÇÃO DO MÉTODO	
retornarTodosPets()		Utilizado para retornar todos os pets cadastrados no sistema.	
retornarPetsUsuario()		Utilizado para retornar todos os pets cadastrados no sistema por um único usuário.	
retornarPetBusca()		Utilizado para retornar os pets cadastrados encontrados através da função de busca.	

Tabela 4.10 - Tabela descritiva da classe UsuarioDB

CLASSE		DESCRIÇÃO DA CLASSE	
UsuarioDB		Esta classe foi criada para o armazenamento e validação de logins de usuários cadastrados no sistema.	
CAMPO	TIPO	TAMANHO	DESCRIÇÃO
MÉTODO		DESCRIÇÃO DO MÉTODO	
validarLogin()		Utilizado para verificar se um determinado login é válido.	

Tabela 4.11 - Tabela descritiva da classe Dao

CLASSE		DESCRIÇÃO DA CLASSE	
Dao		Esta classe abstrata foi criada para implementar métodos básicos de CRUD	
CAMPO	TIPO	TAMANHO	DESCRIÇÃO
em	EntityManager		Lida com o banco de dados.
MÉTODO		DESCRIÇÃO DO MÉTODO	
criarQuery()		Cria uma query para interação com o banco de dados.	
criarTypedQuery()		Cria uma typedQuery para interação com o banco de dados.	
consultarId()		Utilizado para consultar a identificação de um determinado objeto no banco de dados.	
cadastrar()		Utilizado para cadastrar um determinado objeto no banco de dados.	
alterar()		Utilizado para editar os dados de um determinado objeto no banco de dados.	
remover()		Utilizado para remover um determinado objeto do banco de dados.	

Tabela 4.12 - Tabela descritiva da classe Imagem

CLASSE		DESCRIÇÃO DA CLASSE	
Imagem		Esta classe foi criada para armazenar imagens enviadas para o sistema.	
CAMPO	TIPO	TAMANHO	DESCRIÇÃO
id	int	PK	Armazena o número identificador de uma determinada imagem.
imagem	byteArray		Armazena uma imagem enviada ao sistema.
MÉTODO		DESCRIÇÃO DO MÉTODO	

Tabela 4.13 - Tabela descritiva da classe Cidade

CLASSE		DESCRIÇÃO DA CLASSE	
Cidade		Esta classe foi criada para gerenciar dados das cidades.	
CAMPO	TIPO	TAMANHO	DESCRIÇÃO
id	int	PK	Armazena o número identificador da cidade.
nome	String	20	Armazena o nome da cidade.
estado	Estado		Armazena o estado a qual a cidade pertence.
MÉTODO		DESCRIÇÃO DO MÉTODO	

Tabela 4.14 - Tabela descritiva da classe Estado

CLASSE		DESCRIÇÃO DA CLASSE	
Estado		Esta classe foi criada para gerenciar dados dos estados.	
CAMPO	TIPO	TAMANHO	DESCRIÇÃO
id	int	PK	Armazena o número identificador do estado.
nome	String	20	Armazena o nome do estado.
MÉTODO		DESCRIÇÃO DO MÉTODO	

Tabela 4.15 - Tabela descritiva da classe Pet

CLASSE		DESCRIÇÃO DA CLASSE	
Pet		Esta classe foi criada para gerenciar dados do pet a ser doado.	
CAMPO	TIPO	TAMANHO	DESCRIÇÃO
id	int	PK	Armazena o número identificador do pet.
nome	String	20	Armazena o nome do pet.
especie	String	10	Armazena a espécie do pet.
genero	char	1	Armazena o gênero do pet, F para fêmea e M para macho.
raca	String	20	Armazena a raça do pet.
observacao	String	200	Armazena dados extras do pet.
dono	usuario	FK	Armazena quem é o dono do pet.
dataRegistro	Date		Armazena a data de registro do pet no sistema.
fotos	List	4	Armazena um total de 4 fotos do pet.
MÉTODO		DESCRIÇÃO DO MÉTODO	

Tabela 4.16 - Tabela descritiva da classe Usuario

CLASSE		DESCRIÇÃO DA CLASSE	
Usuario		Esta classe foi criada para gerenciar dados do usuário.	
CAMPO	TIPO	TAMANHO	DESCRIÇÃO
id	int	PK	Armazena o número identificador do usuário.
nome	String	40	Armazena o nome do usuário.
email	String	255	Armazena o email/login do usuário.
senha	String		Armazena a senha de login do usuário.
telefone	int	11	Armazena o telefone de contato do usuário.
cidade	String	40	Armazena a cidade onde o usuário reside.
estado	String	40	Armazena o estado onde a cidade do usuário é localizada.
MÉTODO		DESCRIÇÃO DO MÉTODO	

Tabela 4.17 - Tabela descritiva da Interface AdapterImagem

CLASSE		DESCRIÇÃO DA CLASSE	
AdapterImagem		Esta classe foi criada para adaptar uma imagem para que ela possa ser armazenada no sistema.	
CAMPO	TIPO	TAMANHO	DESCRIÇÃO
MÉTODO		DESCRIÇÃO DO MÉTODO	
getConteudo()		Retorna um array de bites contendo a imagem desejada.	

Tabela 4.18 - Tabela descritiva da classe AdapterPart

CLASSE		DESCRIÇÃO DA CLASSE	
AdapterPart		Esta classe implementa a interface AdapterImagem, e foi criada para adaptar um arquivo do tipo Part.	
CAMPO	TIPO	TAMANHO	DESCRIÇÃO
arquivo	Part		Armazena o arquivo a passar pelo padrão adapter.
MÉTODO		DESCRIÇÃO DO MÉTODO	

4.4. Projeto de Banco de Dados

No projeto de banco de dados são criadas as estruturas de dados do sistema e como eles são representados no banco de dados [1]. neste trabalho, o projeto foi feito através do mapeamento do modelo conceitual (Diagrama de Classes), para o modelo de dados de implementação do sistema de banco de dados utilizado (Modelo Orientado à Objetos).

4.4.1. Projeto Lógico Relacional

O esquema gráfico do projeto lógico relacional do banco de dados foi criado com o uso de as regras de transformação entre modelos. A ferramenta MySQL Workbench exibe essas informações.

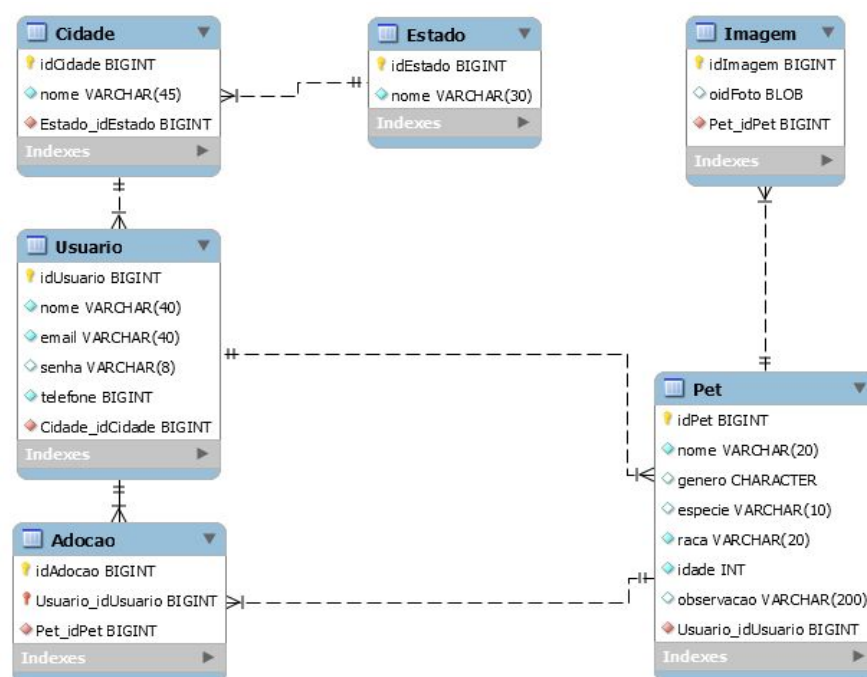


Figura 4.4.1.1 - Projeto Lógico Relacional

4.4.2. Dicionário de Dados

Foram criadas tabelas para descrever os metadados do banco de dados. As tabelas 4.1 a 4.2 descrevem os metadados da Figura 4.5.

Tabela 4.1 - Metadados da Tabela Usuário

TABELA		DESCRIÇÃO DA TABELA						
Usuário		Esta classe foi criada para guardar dados do usuário.						
CAMPO	DESCRIÇÃO	TIPO	TAM.	PK	FK	UK	RESTRIÇÕES	POSSÍVEIS VALORES
usuarioID	Código que identifica unicamente um usuário.	Inteiro		X				
nome	Nome do usuário.	String	40				Não permite números. Campo obrigatório.	
email	Email do usuário, utilizado como login.	String	255			X	Campo obrigatório.	
senha	Senha usada pelo usuário para efetuar login.	String					Campo obrigatório.	

telefone	Telefone do usuário disponibilizado para contato.	Inteiro	11				Não permite letras. Campo obrigatório.	
cidade	Cidade do usuário que pode ser usada para filtrar buscas.	String	40				Não permite números. Campo obrigatório.	

Tabela 4.2 - Metadados da Tabela Pet

TABELA		DESCRIÇÃO DA TABELA						
Pet		Esta classe foi criada para guardar dados do pet cadastrado para doação.						
CAMPO	DESCRIÇÃO	TIPO	TAM.	PK	FK	UK	RESTRIÇÕES	POSSÍVEIS VALORES
petID	Código que identifica unicamente um pet.	Inteiro		X				
nome	Nome do pet.	String	20				Não permite números. Campo obrigatório.	
gênero	Gênero do pet.	char	1				Aceita apenas os valores F e M referentes a Fêmea e Macho.	
espécie	Espécie do pet.	String					Não permite números. Campo obrigatório.	
raça	Raça do pet, podendo ser desconhecida.	String	20				Não permite números. Campo obrigatório.	
idade	Data de nascimento do pet.	int	8				Não permite letras. Campo obrigatório.	
observação	Informações adicionais sobre o pet, tal como personalidade etc.	String	200					

4.5. Projeto da Arquitetura

Para Bosch [2000] *apud* Sommerville [1] a definição da arquitetura do software é importante por afetar o desempenho e a robustez, bem como influenciar na capacidade de

distribuição e de manutenibilidade do sistema. Sabe-se também que é impossível que qualquer sistema seja otimizado em todos os atributos de sua arquitetura. Nas subseções a seguir são descritos alguns aspectos e características da arquitetura do software objeto de trabalho deste projeto.

4.5.1. Segurança

Para Sommerville [1], a propriedade da segurança reflete a capacidade do software operar sem falhas catastróficas, ou seja, não causar danos às pessoas ou ao seu ambiente.

O acesso ao sistema é assegurado por verificação de usuário, usuários visitantes não possuem acesso às funções mais específicas do site, sendo assim, necessário o cadastro e início de sessão para que o usuário tenha total acesso ao sistema e suas funcionalidades.

4.5.2. Disponibilidade

A disponibilidade diz respeito a probabilidade de um sistema estar ativo e funcionando para fornecer serviços aos usuários quando estes solicitarem em determinado momento [Sommerville, 1].

A disponibilidade do sistema é dependente da disponibilidade do servidor sendo utilizado.

4.5.3. Portabilidade

A portabilidade reflete a capacidade de um software poder ser portado de um ambiente para outro; capacidade de ser compilado e executado em diferentes arquiteturas.

O sistema é portátil apenas por navegador, seu design responsivo permite a adaptação a diferentes resoluções de telas.

4.5.4. Usabilidade

Essa propriedade reflete quão fácil é usar o sistema [Sommerville, 1]. Depende dos componentes técnicos de sistema, seus operadores e seu ambiente operacional.

A usabilidade é facilitada através do design implementado, funções importantes e muito utilizadas estão sempre visíveis na tela, independente de qual tela. Informações estão dispostas de forma clara e organizada para melhor entendimento do usuário.

5. VALIDAÇÃO DE REQUISITOS E PROJETO

Esta etapa é importante pois autoriza o prosseguimento do projeto, iniciando o desenvolvimento. Relatar atividades executadas para validar cada requisito.

É necessário que o usuário participe da validação dos requisitos, assinando um documento que aprova os requisitos, interfaces, e etc e concordando com o desenvolvimento do software.

6. PLANEJAMENTO/PROJETO DE CASOS DE TESTE

Desenvolva casos de teste que serão usados para testar o software. Exemplo:

Esta seção apresenta os casos de teste que foram desenvolvidos para o software BiteP. Foram desenvolvidos n casos de teste, cada um relativo a uma interface do software...

6.1 Tela home:

Exibindo todos os pets cadastrados por todos os usuarios.

6.2 Barra de Menus:

Botões de menu todos funcionando.

Caixas de pesquisa todos funcionando com checkbox marcados, fazendo as pesquisas necessárias.

6.3 Link de Teste da Pet sobrando:

Nome do usuário definido como "convidado" e exibido na primeira iteração do usuário com o sistema, o mesmo sendo substituído e exibindo o nome do usuário quando logado ao sistema.

6.4 Imagem:

Canto superior esquerdo que permite redirecionamento para tela home.

6.5 Menu entrar/cadastrar:

A interface é dividida em duas opções, login e cadastrar, no qual os campos de preenchimento de login são estendidos na opção de cadastrar aumentando a quantidade de dados.

Todos os campos de dados contém validações.

O campo de login contém a validação de verificar a propriedade escrita, ou seja, verificar os caracteres.

Não é possível fazer login se os dois campos não forem preenchidos.

No preenchimento incorreto de senha gera uma exceção.

No cadastro os campos estão sem as validações no qual aceitam o preenchimento de dados errôneos ou nulos, o qual somente acusa a necessidade do preenchimento das senhas.

A confirmação dos campos gera exceções(falhas).

O usuário quando logado é exibidos as opções de gerenciamento de perfil e de seus pets.

6.6 Usuário logado -> home:

Ao ser logado o sistema redireciona corretamente para tela home, no qual exibe todos os pets cadastrados.

É exibido no canto superior esquerdo o nome do usuário que está logado no sistema.

6.7 Usuário logado -> meu perfil:

Ao entrar na opção meu perfil são exibidos os dados do usuário cadastrado, também contém a opção de editar ou excluir.

Ao escolher editar é exibido uma interface com os dados cadastrados para serem modificados, o usuário poderá editar o que quiser, e então poderá salvar novamente.

Ao escolher excluir o sistema remove a conta do usuário, não há alerta antes de concluir.

6.8 Usuário logado -> meus pets:

São exibidos todos os pets cadastrados pelo usuário, assim como na tela home todos os registros estão organizados.

Cada pet exibido contém a opção de redirecionamento para suas informações como também a opção de exclusão.

Na opção de exclusão sistema não retorna um aviso ao usuário portanto excluindo imediatamente.

6.9 Usuário logado -> cadastrar pets:

Na opção de cadastrar pets o usuário realizará o cadastramento observando atentamente os campos no qual

Os campos precisam ser preenchidos de acordo com o item proposto e todas as imagens precisam ser carregadas, a quantidade definida para uploads são quatro.

A validação implicada no cadastro somente verifica os campos de senha preenchidos, portanto demais campos não

Foram avaliados, podendo assim não serem preenchidos.

As opções definidas podem ser tanto o cadastramento quanto o cancelamento de tal ação.

6.10 buscarPet:

Na Barra de Menus contém a opção de buscar pets, ele exibe os campos que podem ser preenchidos pelo usuário.

O método de busca baseia-se nos elementos escolhidos pelo usuário para a realização da tarefa.

Todos os campos podem ser ou não preenchidos, os ideais propostos podem ser aplicados na busca.

A tela de home retorna os dados dos pets de acordo com as informações definidas na busca.

6.11 perfilPet:

Nesse perfil são exibidos todas as informações do pet, no qual contém tanto as imagens cadastradas quanto

Os dados do pet e seu dono.

6.12 Logout:

Contém a opção no menu de desconectar no qual o sistema realiza o logout do usuário.

7. RELATÓRIO DA IMPLEMENTAÇÃO/DEPURAÇÃO E DA EXECUÇÃO DE CASOS DE TESTE PROJETADOS

O processo de implementação encontrou diversas barreiras, modificações precisaram ser feitas uma vez que componentes se mostraram difíceis de implementar.

Quanto aos views, JSF foi utilizado apenas no necessário. O framework se mostrou complicado de lidar, fazendo necessária sua substituição em certas funções por outros componentes como por exemplo o PrimeFaces. Incluímos solicitações Ajax para carregar uma view dentro de outra, para que a cada troca de página não seja necessário recarregar componentes em comum das telas causando perda de desempenho.

Existiram inúmeros conflitos entre dados de controladores e persistência no banco de dados, mostrou-se necessária a implementação do padrão adapter para lidar com o armazenamento de imagens. A estrutura do projeto foi baseada no padrão MVC, no qual definimos pacotes para model, view e controller. Dentro de cada um destes pacotes há subcomponentes que podem ser definidos como pacotes mais específicos como exemplo o entity, adapter e dao.

A reta-final da implementação foi focada na caça de bugs e implementação de validações.

8. RELATÓRIO DE IMPLANTAÇÃO

O processo de implementação sofreu diversas modificações ao passar do tempo, a sua implantação no ambiente de execução revelou diversos bugs, os quais foram tratados na reta-final da implementação do sistema.

9. CONCLUSÃO

Podemos concluir que o sistema desenvolvido cumpriu o seu objetivo de facilitar a doação e adoção de pets.

A estratégia utilizada para desenvolvimento tornou possível a aplicação de todos os requisitos fundamentais. Durante os 4 meses de implementação, o quesito de usabilidade sempre foi abordado para que o sistema não se mostrasse difícil para usuários inexperientes.

10. REFERÊNCIAS BIBLIOGRÁFICAS

[1] Sommerville, I. Engenharia de Software, 9ª Ed. Pearson, 2011.

[2] Diagrama de Casos de Uso. Disponível em: <http://www.dsc.ufcg.edu.br/~sampaio/cursos/2007.1/Graduacao/SI-II/Uml/diagramas/usecases/usecases.htm> - Acesso em 06/06/2018.

[3] SBC, Interação Humano-Computador. Disponível em: <http://www.sbc.org.br/14-comissoes/390-interacao-humano-computador> - Acesso em: 11/04/2018.

[x] Guedes, G. T. A. UML 2 - Uma Abordagem Prática, 2ª Ed. Novatec, 2011.

[3] McNulty, E. J. What Is This Thing Called Resilience?. 2014. Disponível em: <https://www.strategy-business.com/blog/What-is-This-Thing-Called-Resilience> - Acesso em: 01/11/2017.