#### UNIVERSIDADE FEDERAL DE JUIZ DE FORA CAMPUS JUIZ DE FORA BACHARELADO EM SISTEMAS DE INFORMAÇÃO

# BERNARDO ORTIZ CAIO VIEIRA LUCAS DUARTE RHUAN NASCIMENTO FERREIRA

**RELATÓRIO DO TRABALHO 1: PET SHOP** 

JUIZ DE FORA 2022 SUMÁRIO

SUMÁRIO	1
INTRODUÇÃO	2
DOMÍNIO DA APLICAÇÃO	3
INTERFACE	4
DashBoard	4
Agenda	4
- Agenda(Editar)	5
- Agenda(Cadastrar)	5
Cliente	6
- Cliente(Cadastrar)	6
- Cliente(Editar)	7
Animais	7
- Animais(Cadastrar)	8
- Animais(Editar)	8
<ul> <li>Pendências</li> </ul>	9
<ul><li>Produtos</li></ul>	9
- Produtos(Cadastrar)	10
- Produtos(Editar)	10
<ul> <li>Serviços</li> </ul>	11
- Serviços(Cadastrar)	11
- Serviços(Editar)	12
DESCRIÇÃO DAS CLASSES E PACOTES UTILIZADOS	13
INSTRUÇÃO PARA COMPILAR E EXECUTAR O CÓDIGO	13

# INTRODUÇÃO

Este documento trata de um relatório contendo informações sobre um software para gerenciar um comércio "Pet Shop". Tal sistema foi desenvolvido a pedido de Gleiph Ghiotto, professor de Orientação a Objetos na Universidade Federal de Juiz de Fora. O documento em questão deve conter: definição do domínio da aplicação, o projeto da interface de interação com usuário, descrição das classes e pacotes utilizados e instrução para compilar e executar o código.

#### DOMÍNIO DA APLICAÇÃO

O projeto tem como objetivo o gerenciamento interno de um "Pet Shop", com funções de agendamento, gerenciamento de cliente, animais relacionados com seus respectivos donos (clientes), controle de estoque e serviços prestados.

Dentre as funções do software, estão:

**DashBoard:** Trata-se da tela inicial, contém um painel com um resumo de quais clientes estão agendados para o dia, uma lista com pendências a realizar e botões de acesso às telas (Agenda, Clientes, Animais, Pendências, Produtos e Serviços);

**Agenda:** Permite visualizar, criar, editar e remover agendamentos. Um agendamento possui cliente, animal, serviço, horário e data. Na tela de criação e edição de um agendamento temos a opção de redirecionar para as telas de cadastro de animal, cliente e serviço - caso algum desses ainda não esteja registrado no sistema;

**Serviços:** Permite visualizar, criar, editar e remover serviços. Um serviço possui tipo, frequência e preço. A frequência diz respeito ao tempo (em dias) que o serviço deve ser realizado novamente. Caso o animal esteja passando por algum tratamento, o cliente deve retornar ao Pet Shop depois de "X" dias, por exemplo.

Cliente: Permite visualizar, cadastrar, editar e remover clientes. Um cliente possui nome, CPF, email, telefone, animal e endereço. Para visualizar todas estas informações temos que entrar na tela de edição; na tela de exibição temos apenas um resumo contendo nome, telefone e animais. Ao cadastrar ou editar dados do cliente não é possível adicionar ou remover animais. Esta opção somente pode ser feita na tela de cadastro/edição de animais;

**Animais:** Permite visualizar, cadastrar, editar e remover animais. Um animal possui nome, espécie, raça e dono;

**Produtos:** Permite visualizar, criar, editar e remover produtos. Um produto possui nome, quantidade e preço;

**Pendências:** Possibilita visualizar, adicionar e excluir pendências. Uma pendência possui nome e estado - pendente ou realizada - mediado através de um *checkbox*. Não é possível editar uma pendência, apenas excluir e adicionar novas.

#### **INTERFACE**

#### • Início



Fonte: Elaborado pelos autores, 2022.

# - Agenda(Editar)

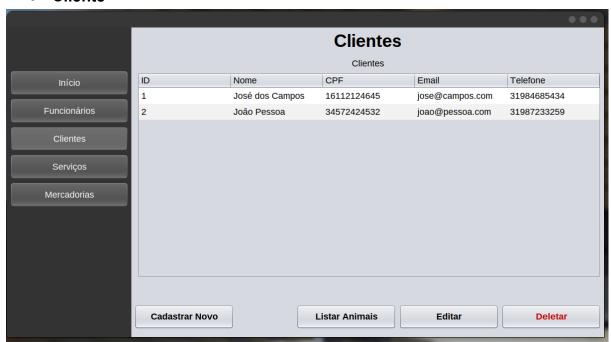


#### - Agenda(Cadastrar)

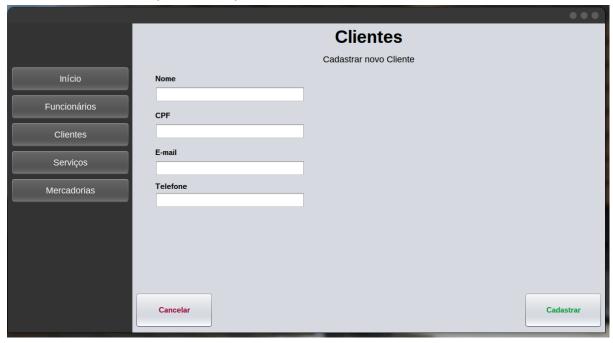


Fonte: Elaborado pelos autores, 2022.

#### Cliente

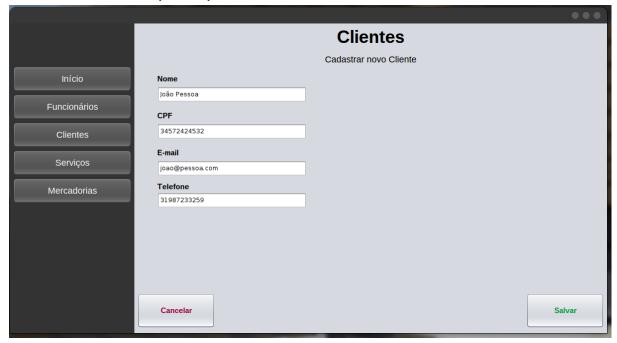


# - Cliente(Cadastrar)

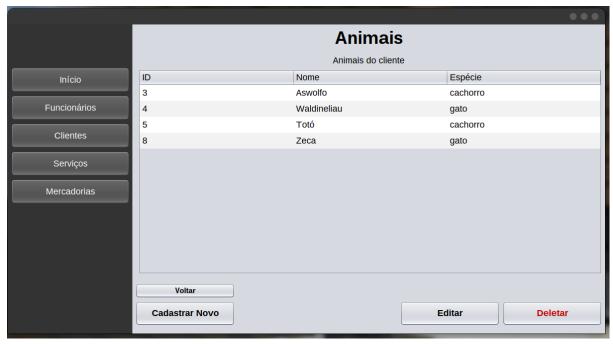


Fonte: Elaborado pelos autores, 2022.

# - Cliente(Editar)

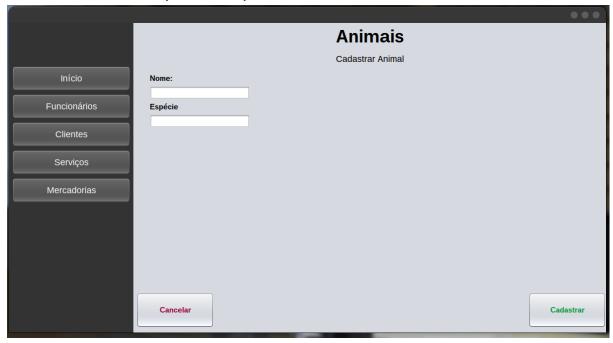


#### Animais

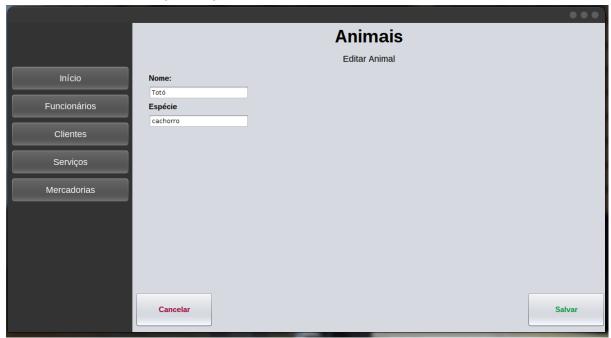


Fonte: Elaborado pelos autores, 2022.

# - Animais(Cadastrar)

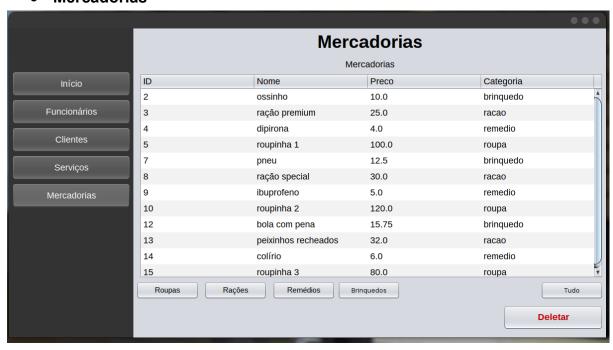


#### - Animais(Editar)



Fonte: Elaborado pelos autores, 2022.

#### Mercadorias

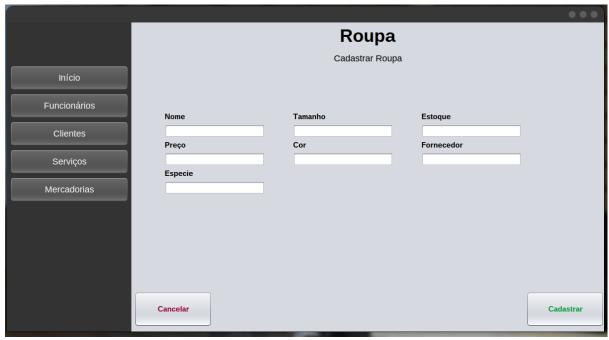


- Roupas



Fonte: Elaborado pelos autores, 2022.

- Cadastrar Roupa



#### - Editar Roupa



Fonte: Elaborado pelos autores, 2022.

- Rações



# - Editar Rações



Fonte: Elaborado pelos autores, 2022.

- Cadastrar Ração



#### - Remédios



Fonte: Elaborado pelos autores, 2022.

- Editar Remédio



#### - Cadastrar remédio



Fonte: Elaborado pelos autores, 2022.

- Brinquedos



# - Editar Brinquedo



Fonte: Elaborado pelos autores, 2022.

- Cadastrar Brinquedo

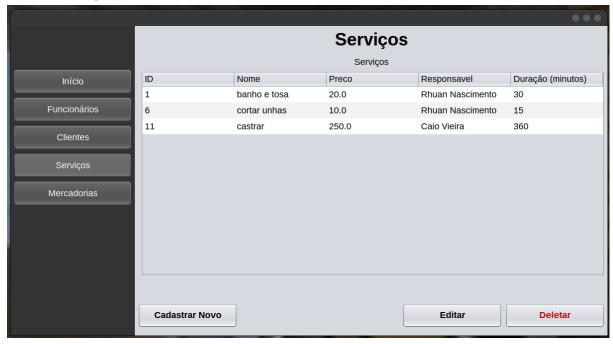


# - Produtos(Vender, todos)

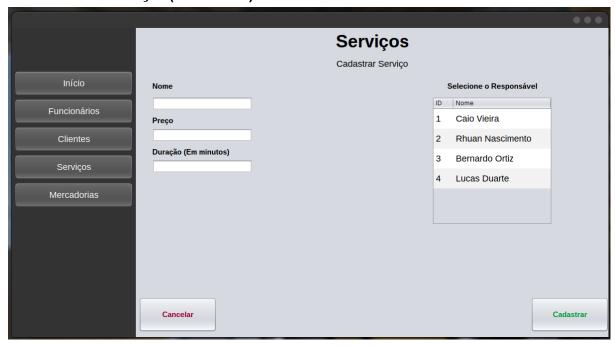


Fonte: Elaborado pelos autores, 2022.

#### Serviços

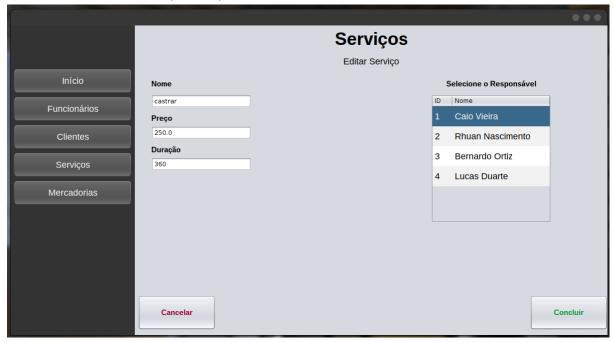


# - Serviços(Cadastrar)



Fonte: Elaborado pelos autores, 2022.

# - Serviços(Editar)



#### DESCRIÇÃO DAS CLASSES E PACOTES UTILIZADOS

O projeto será organizado no padrão MVC (*Model-View-Controller*), logo serão utilizados 3 pacotes principais: pacote model, responsável pela manipulação da informação do sistema; pacote view, responsável pela interação do usuário; e o pacote controller, para a administração do projeto em geral.

No pacote model, temos as classes Agenda, Agendamento, Animal, Cliente, Produto, Serviço e Pendência. Cada uma com seus respectivos métodos para adição, edição e remoção, como descrito no domínio da aplicação (página 3), e detalhados no código fonte disponibilizado.

#### INSTRUÇÃO PARA COMPILAR E EXECUTAR O CÓDIGO

O projeto foi construído com Maven, utilizando o comando **mvn install** através da configuração do arquivo **pom.xml** com o plugin de construção com dependências disponível no link (https://stackoverflow.com/questions/574594/how-can-i-create-an-executable-runnab le-jar-with-dependencies-using-maven) para a versão do java até 17. Para a execução do código, o usuário deve digitar no terminal o seguinte comando:

#### mvn exec:java -Dexec.mainClass=com.mycompany.petshop.PETSHOP

Como o projeto também acompanha o build, o usuário também poderá rodá-lo utilizando o seguinte comando:

#### java -jar target/PETSHOP-1.0-SNAPSHOT-jar-with-dependencies.jar

É possível passar argumentos para executar ações pré programadas de banco de dados. São elas:

- migrate: cria as tabelas do banco de dados, caso elas não existam
- **seed**: popula as tabelas do banco de dados com informações suficientes para executar qualquer operação da aplicação
- test efetua um teste de cada operação dos repositórios, de forma a validar toda a aplicação. Esse argumento faz o banco sofrer diversas alterações, pensadas nos dados criados pelo argumento seed. Por alterar bastante o banco, ao final da operação, os bancos são limpos.

Pelo fato do arg **test** limpar os bancos, é possível populá-los novamente, usando novamente o seed.

Um exemplo de comando para rodar a aplicação, migrar os bancos, popular os bancos, testar a aplicação e popular os bancos novamente é:

java -jar target/PETSHOP-1.0-SNAPSHOT-jar-with-dependencies.jar migrate seed test seed

# Logins para a aplicação:

Ao rodar o argumento "seed", são cadastrados os seguintes logins:

- caio.costa senhacaio -> gerente
- rhuan.nascimento senharhuan -> funcionario
- bernardo.ortiz senhabernardo -> funcionario
- lucas.duarte senhalucas -> funcionario

Link para repositório no Github: (https://github.com/rhuannascimento/Pet\_Shop\_Java)