CENTRO UNIVERSITÁRIO DE BRASÍLIA - UniCEUB
CURSO DE ENGENHARIA DE COMPUTAÇÃO

JOÃO VICTOR MARQUES DOS SANTOS

CONTROLE DE ATIVIDADES PARA PETSHOP UTILIZANDO RFID

Orientadora: MsC Prof. Maria Marony Sousa Farias

Brasília

Dezembro, 2013

JOÃO VICTOR MARQUES DOS SANTOS

CONTROLE DE ATIVIDADES PARA PETSHOP UTILIZANDO RFID

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB) como pré-requisito para a obtenção de Certificado de Conclusão de Curso de Engenharia de Computação.

Orientadora: MsC Prof. Maria Marony Sousa Farias

Brasília

Dezembro, 2013

JOÃO VICTOR MARQUES DOS SANTOS

CONTROLE DE ATIVIDADES PARA PETSHOP UTILIZANDO RFID

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB) como pré-requisito para a obtenção de Certificado de Conclusão de Curso de Engenharia de Computação.

Orientadora: Profª. MsC Maria Marony Sousa Farias

Este Trabalho foi julgado adequado para a obtenção do Título de Engenheiro de Computação, e aprovado em sua forma final pela Faculdade de Tecnologia e Ciências Sociais Aplicadas - FATECS.

_____

Prof. Abiezer Amarilia Fernandez
Coordenador do Curso

Banca Examinadora:

_____

MsC. Profª. Maria Marony Sousa Farias, mestre em Engenharia Elétrica – UFPB – PB.
Orientadora - UniCEUB

_____

Prof. Luis Cláudio Lopes de Araújo, mestre em Matemática.
UniCEUB

_____

Prof. Luciano Duque, mestre em Engenharia Elétrica.
UniCEUB

_____

Profª. Vera Farini, mestre em Matemática.
UniCEUB

DEDICATÓRIA

Dedico este trabalho a minha família que tanto me apoiou, principalmente, às minhas mães(Antônia e Nilde), meu Pai Geraldino e àqueles que acreditaram em mim. Muito obrigado!

AGRADECIMENTOS

Agradeço a ajuda de Deus e a todos os professores envolvidos neste trabalho.

À paciência e apoio da minha Orientadora Profª Marony e do Prof. Javier

A todo o corpo docente da instituição UniCEUB.

Agradeço ao meu Tio Luiz Henrique de Sá que deu início a este projeto de formar um menino em um Engenheiro.

SUMÁRIO

LISTA DE FIGURAS 8
LISTA DE TABELAS 10
LISTA DE ABREVIAÇÕES E SIGLAS 11
RESUMO 12
ABSTRACT 13
CAPÍTULO 1 – INTRODUÇÃO AO PROJETO 14
1.1. Apresentação do Problema 15
1.2. Objetivos 15
1.3. Justificativa e Relevância do Trabalho 16
1.4. Escopo 16
1.5. Resultados Esperados 16
1.6. Estrutura do Trabalho 17
CAPÍTULO 2 – APRESENTAÇÃO DO PROBLEMA 18
2.1. A Correta Identificação dos animais de estimação 18
CAPÍTULO 3 – BASES TEÓRICAS E METODOLÓGICAS PARA RESOLUÇÃO DO PROBLEMA 20
3.1. Identificação por rádio frequência 20
3.2. TAG's e Leitoras RFID 20
3.3. Microcontrolador Arduíno 21
3.4. Banco de Dados – Oracle 22
3.5. A Linguagem PL/SQL 22
3.6. A Linguagem Java 23
CAPÍTULO 4 – PROTÓTIPO DO SISTEMA DE CONTROLE DE ATIVIDADES – PETSHOP 24
4.1. Apresentação Geral do Projeto Proposto 24
4.2. Descrição dos Hardwares envolvidos 25
4.2.1. Arduíno Uno R3 25
4.2.2. Leitora RFID – MF522-AN 26
4.3. O Software de controle leitura – Arduíno 27
4.4. O Software de Interface com o Usuário 28
4.5. Criação do Banco de Dados 31
4.6. Conexão com o banco de dados 34
CAPÍTULO 5 – A IMPLEMENTAÇÃO 36
5.1. Modelagem da interface com o usuário 36

5.2. AUTENTICAÇÃO 37
5.2.1. CADASTRAR CLIENTE 37
5.2.2. CONSULTAR CLIENTE 38
5.2.3. ADICIONAR PET 41
5.2.4. CRIAR ORDEM DE SERVIÇO 41
5.2.5. CONSULTAR ORDEM DE SERVIÇO 43
5.2.6. TELA DIÁRIA 45
5.2.7. CADASTRAR TAG 45
5.2.8. CADASTRAR USUÁRIO 46
5.3. M ER – BANCO DE DADOS ORACLE 47
5.4. TESTES DO SISTEMA DE CONTROLE DE ATIVIDADE DE UM PET SHOP 48
5.5. DIFICULDADES ENCONTRADAS 56
5.6. RESULTADOS OBTIDOS 57
5.7. PRODUTO GERADO 58
CAPÍTULO 6 – CONSIDERAÇÕES FINAIS 59
6.1. CONCLUSÃO 59
6.2. PROPOSTAS PARA FUTUROS PROJETOS 59
REFERÊNCIAS BIBLIOGRÁFICAS 60
APÊNDICE A 62
APÊNDICE B 64

LISTA DE FIGURAS

Figura 3.1 – Código PL/SQL alternando entre o motor PL/SQL e SQL 23
Figura 4.1 – Diagrama geral do Sistema de controle de Atividades de um Pet Shop 24
Figura 4.2 – Arduíno Uno R3 25
Figura 4.3 – ATmega328 – Microship 25
Figura 4.4 – Placa MF522-AN – MIFARE 26
Figura 4.5 – Código-fonte utilizado para fazer a comunicação entre o leitor e o computador utilizando o arduíno 27
Figura 4.6 – IDE Eclipse Kepler – Projeto: Controle de Atividades de um Pet Shop 29
Figura 4.7 – Ações do administrador 30
Figura 4.8 – Ações do Atendente 30
Figura 4.9 – Ações do Tratador 31
Figura 4.10 – Início da criação do banco de dados 31
Figura 4.11 – Fim da criação do banco de dados 32
Figura 4.12 – Serviço TNSListener iniciado 33
Figura 4.13 – IDE SQL Navigator – Procedure intp_gera_osrfid sendo desenvolvida 34
Figura 4.14 – Cada DriverManager possui seu driver JDBC específico 35
Figura 5.1 – Fluxograma geral do Sistema de Controle de Atividades 36
Figura 5.2 – Autenticação 37
Figura 5.3 – Cadastro de um novo cliente 38
Figura 5.4 – Consultar Cliente – Procurar Cliente 39
Figura 5.5 – Consultar Cliente – Procurar Pets 39
Figura 5.6 – Consultar Cliente – Procurar O.S 40
Figura 5.7 – Consultar Cliente – Detalhes da O.S 40
Figura 5.8 – Fluxo adicionar animais de estimação associado a um cliente 41
Figura 5.9 – Criar Ordem de Serviço – Procurar 42
Figura 5.10 – Criar Ordem de Serviço – Procurar Pets 42
Figura 5.11 – Criar Ordem de Serviço – Selecionar Serviços 43
Figura 5.12 – Criar Ordem de Serviço – Gerar Ordem de Serviço 43
Figura 5.13 – Consultar Ordem de Serviço – Procurar por tag 44
Figura 5.14 – Consultar Ordem de Serviço – Procurar por id O.S 44
Figura 5.15 – Tela Diária 45
Figura 5.16 – Cadastrar tag - Leitura 46

Figura 5.17 – Cadastrar tag – Inserção 46
Figura 5.18 – Cadastrar Usuário 47
Figura 5.19 – MER – CRMRFID 47
Figura 5.20 – Tela Inicial 48
Figura 5.21 – Autenticação 49
Figura 5.22 – Autenticação – aviso 49
Figura 5.23 – Incluir Cliente 50
Figura 5.24 – Gerar Ordem de Serviço – Aviso 51
Figura 5.25 – Geração de ordem de serviço com sucesso 52
Figura 5.26 – Consultar O.S por tag 53
Figura 5.27 – Consultar Cliente 54
Figura 5.28 – Consultar Cliente, tramitação de O.S 55
Figura 5.29 – Tela Diária 55
Figura 5.30 – Montagem do Arduíno, Placa leitora RFID e o computador 58

LISTA DE TABELAS

Tabela 4.1 – Configuração dos Pinos MF522-AN – Arduíno Uno R3 28
Tabela 4.2 – Objetos do banco de dados – CRMRFID 32

LISTA DE ABREVIAÇÕES E SIGLAS

ISO	–	Internacional Standards Organization
ANSI	–	American National Standards Institute
SGBD	–	Sistema Gerenciador de Banco de Dados
PL/SQL	–	Procedural Language / Structured Query Language
GUI	–	Graphical User Interfaces
SPI	–	Serial Peripherical Interface
RFID	–	Radio Frequency Identification
MER	–	Modelo de Entidade de Relacionamento

RESUMO

Este trabalho apresenta um sistema de controle para a identificação e controle de atividades realizadas com animais de estimação em um estabelecimento Pet Shop, por rádio frequência – RFID, integrado a um microcontrolador arduíno R3 Uno, realizando uma comunicação serial com um computador que contém um sistema de controle de atividades. O animal de estimação é identificado através de uma tag – etiqueta eletrônica – enquanto o animal estiver no ambiente do Pet Shop. Os dados que contêm informações do dono, animal de estimação e as atividades, ficam armazenados em um banco de dados ORACLE 10G, e o sistema é construído pela linguagem de computação JAVA, versão 1.7, utilizando a API Swing.

Palavras Chave: RFID, Pet Shop, Arduíno, Oracle 10g, Java Swing.

ABSTRACT

This work aims to identify and control activities with pets Pet Shop in an establishment, using radio frequency identification - RFID, integrated with a microcontroller Arduino Uno R3, performing serial communication with a computer that contain a control system activity. The pet is identified by a tag - electronic tag, while the animal is in the environment of the Pet Shop. The data that contain information of the owner, pet and activities, are stored in a database ORACLE 10G, the system is built by computer language JAVA, version 1.7 using the Swing API.

Keywords: RFID, Arduino, ORACLE , JAVA , Pet Shop.

CAPÍTULO 1 – INTRODUÇÃO AO PROJETO

Os dados da indústria Pet, desde que começaram a ser medidos em 1998, apresenta um vertiginoso crescimento. Um mercado promissor como este exige soluções e inovações que venham consolidar ainda mais este ramo. (BRANCO, 2013)

O empreendedor no ramo de Pet Shop deve ficar atento às novidades e tendências do mercado em si quanto ao atendimento e à qualidade dos serviços oferecidos, pois neste mercado em franco crescimento, a atração dos clientes é definida nestes termos. O público alvo desse mercado encontra-se nas classes A e B, que estão dispostos a pagar por novidades, qualidade e comodidade. (SEBRAE, 2011)

Segundo Taufenbach (2004), a tecnologia de identificação através de rádio frequência - RFID vem sendo usada em identificação de objetos desde 1969 e patenteadas em 1973, mas, somente a partir de 2004, estão se tornando comercialmente e tecnologicamente viáveis.

De acordo com Pressman (2011), os RFID's trazem a computação para uma base industrial e para o ramo de produtos de consumo. Sendo assim, com este trabalho, será trazido para o ramo de serviços de Pet Shop. A tecnologia da Informação vem auxiliando as indústrias, os negócios de uma empresa no sentido de aumentar a produtividade e os lucros.

No ano de 2003, o WalMart anunciou, o que ficou conhecido como "o mandato", onde o objetivo era que todos os seus fornecedores deveriam etiquetar seus suprimentos com a etiqueta eletrônica. Esse mandato incentivou muita inovação por parte da indústria de RFID, em que foram realizados investimentos e pesquisa que, comprovadamente, aceleraram o avanço da tecnologia. Foram superadas três importantes barreiras: a percepção, a funcionalidade e o preço. Antes, havia a ideia errônea de substituir os códigos de barras em toda a cadeia de suprimentos. Havia, também, um alto índice de erro nas leituras e o preço em relação ao código de barra era cerca de 400% mais caro. Aconteceu que os leitores foram melhor projetados, o silício utilizado nas etiquetas é de melhor qualidade, reduzindo as leituras espúrias. O RFID, após essas melhorias, vem auxiliando o ramo de logística e varejo, aumentando o volume de escoamento dos produtos. (MOURA, 2013)

Quanto à percepção de onde e como deve ser utilizar o RFID, chegou-se a conclusão que a convergência era a melhor saída juntamente com aplicação desta tecnologia em circuitos fechados controlados por uma empresa, especialmente o fabricante. Atualmente, a etiqueta eletrônica pode ser associada a gerenciamento de ativos, associações a dados coletados por sensores ou pode ser associada a um banco de dados. (MOURA, 2013)

1.1. APRESENTAÇÃO DO PROBLEMA

Com o crescimento do mercado de Pet Shop, os serviços prestados aos animais de estimação precisam de um melhor controle de atividades e identificação. Tratando-se de um pequeno Pet Shop, é possível controlar a identificação do animal de estimação com uma etiqueta escrita manualmente. Porém, quando falamos de agilidade crescimento e expansão do negócio, surge a necessidade de um melhor controle nas atividades oferecidas pelo estabelecimento.

A identificação dos animais de estimação e controle de atividades, quando estão no playground, hotel, tornam-se precárias ante a atual forma de apresentação.

1.2. OBJETIVOS

O objetivo deste trabalho é controlar as atividades realizadas com um animal de estimação, extraindo a função principal da etiqueta eletrônica que é identificar e informar, utilizando um software dentro de um estabelecimento de Pet Shop, identificando os animais, seus donos, bem como as atividades realizadas com os dados armazenados em um banco de dados. Os objetivos específicos são:

1. Utilizar um hardware de leitor RFID e uma tag;
2. Desenvolver uma comunicação entre o leitor de RFID e o computador, utilizando o microcontrolador arduíno uno R3;
3. Desenvolver um software para o cadastramento e controle de atividades utilizando a linguagem Java 1.7, PL/SQL e banco de dados Oracle 10G;
4. Analisar um cadastro, atendimento e conclusão de um serviço realizado com um animal de estimação.

1.3. JUSTIFICATIVA E RELEVÂNCIA DO TRABALHO

O desenvolvimento deste sistema surgiu da necessidade de agregar valor ao negócio de Pet Shop, auxiliando esse ramo de negócios com as ferramentas que a tecnologia da informação pode oferecer. O mercado de pet nacional vem se profissionalizando a cada ano e possui o segundo maior mercado mundial no setor. O mercado de pet possui ramificações, entre elas estão o *pet food* (alimentação), *pet care* (acessórios, produtos para higiene e beleza e equipamentos), *pet vet* (produtos veterinários) e *pet serv* (serviços). À soma do volume negociado em todas essas ramificações, verifica-se um crescimento de 17,3% em relação ao ano de 2011 para 2012, com um volume de negócios no ano de 2012 de aproximadamente de R$ 14 bilhões. (ABINPET, 2013).

Tendo a noção do tamanho do mercado de pet brasileiro e seu potencial, este projeto visa agregar valor ao ramo de serviço do mercado de pet, aumentando a confiabilidade e qualidade dos estabelecimentos de Pet Shop, pois quando há um grupo de pets reunidos, a correta identificação do pet, do dono do pet e das atividades realizadas com cada um, se faz necessária através de um sistema administrativo à altura.

1.4. ESCOPO

Neste trabalho é desenvolvido um sistema para identificar um animal de estimação que está sendo tratado em um estabelecimento de Pet Shop através de uma tag, utilizando a identificação por rádio frequência.

O software projetado aqui neste trabalho não tem pretensão de controlar o fluxo de caixa, contas a pagar e controle de fornecedores.

1.5. RESULTADOS ESPERADOS

Espera-se, com este trabalho, construir um software capaz de controlar as atividades realizadas em animal de estimação, utilizando a identificação por RFID. O animal será identificado, enquanto ele estiver no estabelecimento de Pet Shop.

1.6. ESTRUTURA DO TRABALHO

O trabalho aqui apresentado está estruturado em seis capítulos:

Capítulo 1 - introdutório, onde é apresentada a tecnologia RFID, o mercado de pet shop, objetivo deste trabalho, o problema a ser revolvido e a motivação deste trabalho.

Capítulo 2 - trata do problema identificado, onde os aspectos e situações são descritos, visando levar à compreensão dos fatores que envolvem este projeto.

Capítulo 3 - refere-se a métodos tecnológicos que são utilizados no RFID e linguagens de programação

Capítulo 4 - trata da proposta do projeto, o escopo, implementação e como foi desenvolvido.

Capítulo 5 - apresenta um caso de aplicação do trabalho detalhado e evidenciados os resultados obtidos.

Capítulo 6 - apresenta a conclusão e sugestões de trabalhos futuros.

CAPÍTULO 2 – APRESENTAÇÃO DO PROBLEMA

Este capítulo tem como finalidade mostrar consequências de uma identificação errônea por parte dos funcionários de um estabelecimento de pet shop, e apresentar os possíveis problemas gerados por esta ação.

Com uma declaração de um consumidor podemos ter em mente o tipo de problema a ser resolvido. (CONSUMIDOR, 2010)

> "Comprei um Lhasa Apso, em um Pet Shop próximo a minha residência, como ele é filhote, tem que tomar as 3 doses de vacinas, para que não contraia doenças, infecções nem nada do tipo, pois bem, ele tomou a 1ª vacina ainda no Pet Shop em que o comprei, foi marcado o retorno e já na 2ª dose o levei para tomar no Pet Center Marginal, ele tomou corretamente a vacina e assim marcamos a 3ª dose.!. Quando voltamos para tomar a 3ª dose, por erro do Pet Shop, não aplicaram a 3ª dose e sim a vacina antirrábica, deixando então o meu bichinho com ciclo de vacinação incompleto.!"
> "Meu cachorro vem sofrendo de vômitos e diarreias, por conta de infecção, causado pela baixa imunidade do cachorro, atribuído é claro, pelo erro na vacinação".

2.1. A CORRETA IDENTIFICAÇÃO DOS ANIMAIS DE ESTIMAÇÃO

O relato de um dono de animal de estimação descreve um erro de procedimento a um cachorro, realizado em um Pet Shop. Os erros de procedimento têm acontecido nos estabelecimentos de Pet Shop devido à falha na identificação do animal. Conforme pergunta feita a clientes de estabelecimentos Pet Shop – por mais que haja esforços por parte da concorrência para que me torne cliente de outros Pet Shop e/ou Agroveterinária, haveria troca de Pet shop X e/ou Agroveterinária X por outra? (CRESTANI, 2012, p. 56). Como resposta, obteve-se 82.5% dos entrevistados afirmando que não trocariam. Porém, ao realizar a entrevista com o grupo de consumidores assíduos, ficou evidenciado que, em se tratando de serviços prestados em relação à saúde e o bem estar de um animal de estimação, não podem ocorrer erros como maus tratos, cortes, medicamentos indicados de forma errada, e que isso sim é um fator que todos avaliam como grave, o que os faz trocar de estabelecimento. (CRESTANI, 2012, p. 58)

Uma reportagem conta um fato que aconteceu a um cachorro. (TRIBUNA ANIMAL, 2012)

> “Andrade dono de Tony diz que o cachorro foi levado para banho e morreu trancado no veículo da empresa, onde ficou horas esquecido. Já a proprietária do pet shop admite ter esquecido o animal, mas diz que ele ficou dentro da loja, não do veículo. Tony foi levado para o banho às 9h segundo Andrade, horas se passaram, e sua irmã resolveu ligar para o pet shop, por volta das 16h, e que chegou a ser informada que o animal já havia sido entregue.”

Nesta reportagem é relatado o descuido por parte do funcionário que ao trancar o carro, não observou que havia um cachorro lá dentro, e mostra o descontrole administrativo ao dar ao cliente com uma informação não verdadeira.

Em seu trabalho, Crestani (2012) entrevistou um gestor de um estabelecimento de Pet Shop. Ele diz se preocupar com os fatores de qualidade e cuidados, e constantemente investe na diversidade de produtos e no treinamento de seus colaboradores para que se atualizem em relação ao atendimento e ao conhecimento sobre os produtos.

É dever do gestor facilitar o trabalho e oferecer aos funcionários, ferramentas que evitem tais descuidos.

Mas, ao utilizar o sistema de controle de atividades em um Pet Shop, tais descuidos podem ser minimizados e até extintos quando associados ao treinamento e à atenção dos funcionários.

CAPÍTULO 3 – BASES TEÓRICAS E METODOLÓGICAS PARA RESOLUÇÃO DO PROBLEMA

3.1. IDENTIFICAÇÃO POR RÁDIO FREQUÊNCIA

Neste contexto, vemos dois importantes substantivos, Identificação e Controle. Conforme (Ferreira, 1988. p. 349), identificação é “S.F 1. Ato ou efeito de identificar(se). 2. Reconhecimento duma coisa ou dum indivíduo como os próprios” e Controle é “S.M Fiscalização ou monitoramento exercidos sobre certas atividades, ou o poder de exercê-los”. (AULETE, 2013)

Desde tempos remotos, o ser humano sente a necessidade de identificar-se e identificar seus bens. A busca por novas tecnologias, e novas formas de resolver os problemas atuais, reflete em nosso cotidiano mudando os costumes.

De acordo com Matsubayashi (2004).:apud Taufenbach, (2004), “RFID é a tecnologia que permite identificação por rádio frequência, ou seja, permite a leitura sem contato visual direto. Afirma ainda o autor que essa nova tecnologia será rotina e terá impacto direto no cotidiano das pessoas e nos processos logísticos de toda cadeia de abastecimento”.

3.2. TAG’S E LEITORAS RFID

O leitor de radiofrequência emite ondas magnéticas que acionam a tag. As tags RFID são *chips* com capacidade de armazenarem informações que podem ser lidas dentro do campo magnético provido pelos leitores RFID - *transceivers* - leitores de etiquetas eletrônicas. Quando a tag passa dentro da zona eletromagnética gerada pelo leitor RFID, são dectados, lidos e decodificados os dados que estão armazenados em sua memória, passando-o para um computador realizar o processamento. A fonte de energia provém, do próprio sinal de rádio que as consultam e têm, consequentemente, sua resposta. Tais tags são denominadas passivas. Quando as tags possuem bateria interna, são denominadas tags ativas. Esses transmissores podem ser lidos a pequenas ou longas distâncias dependendo do tamanho da antena ou da potência. As tags existem em vários tamanhos e formas, em vários níveis de frequências de rádio: (TAUFENBACH, 2004)

- Temos as tags que operam na faixa de 125 a 134 KHz – baixa frequência.
- Temos as tags que operam na faixa de 13,56 MHz – Alta frequência.
- Temos as tags que operam na faixa de 868 a 956 MHz – Ultra-elevada frequência.
- Temos as tags que operam na faixa de 2,45 GHz – Microondas.

O leitor RFID não precisa de campo visual para realizar a leitura da tag. Essa leitura pode ser feita através de diversos materiais como plásticos, madeira, vidro, papel, cimento. Esse campo magnético precisa ser forte o bastante para englobar a tag. (PINHEIRO, 2004).

3.3. MICROCONTROLADOR ARDUÍNO

O microcontrolador é um dispositivo semicondutor em forma de C.I - Circuito Integrado - que integra as partes básicas de um microcomputador para aplicações específicas. Entre estas partes básicas estão: - o processador, memórias e portas de entrada e saída. Suas vantagens giram em torno de baixo custo e o baixo consumo de energia. (GIMENEZ, 2002).

O microcontrolador escolhido para este trabalho foi o arduíno uno. É um placa baseada na arquitetura do *microship* ATmega328 produzido pela *Atmel Corporation*. Este *microship* possui uma interface serial de periféricos, conhecida como SPI, que permite uma transferência de dados síncrona de alta velocidade entre dispositivos externos. Essa interface foi utilizada neste projeto para fornecer a comunicação entre o arduíno uno e o computador. (ARDUINO1, 2013).

3.4. BANCO DE DADOS – ORACLE

O foco nesta parte foi pensar em como as coisas se relacionam, segundo Oliveira (2005, p.31):

> Modelar dados significa atravessar o árido território que se estende dos requisitos dos usuários à implementação física dos dados. O SGBD – Sistema Gerenciador de Banco de Dados - é um software responsável pela definição, manipulação, controle e armazenamento de informações relacionadas, controlando a segurança, a consistência e a integridade de dados.

Desde que bem projetados, estas características podem ser atingidas. Para isso é necessário a definição documentada sobre as características do produto – os requisitos.

A partir dos requisitos, nasce o modelo conceitual e posteriormente o modelo de dados que enfatiza a representação e organização dos dados armazenados, enquanto o modelo conceitual visa representar a compreensão da informação. (OLIVEIRA, 2005)

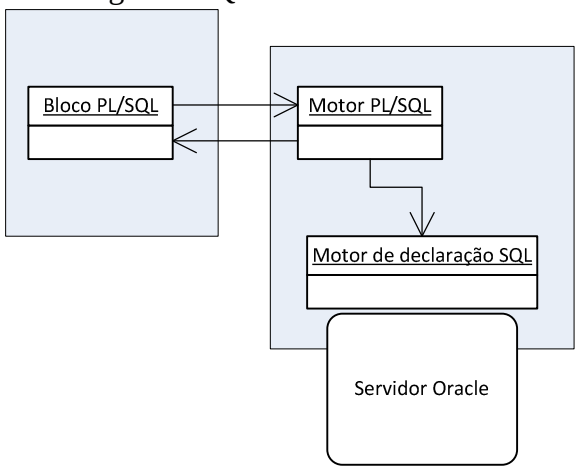
O projeto físico finaliza o processo de desenvolvimento de um banco de dados. Isso significa definir como os dados serão armazenados nos dispositivos disponíveis, quais os tipos de dados e como as regras estabelecidas pelo negócio serão tratadas pelo SGBD.

3.5. A LINGUAGEM PL/SQL

PL/SQL é uma linguagem de programação utilizada pelo SGBD da Oracle. É uma extensão da linguagem de procedimentos desenvolvida pela Oracle para o padrão SQL, para fornecer um modo de executar a lógica de procedimentos no banco de dados. A SQL se tornou a linguagem franca das linguagens de acesso a banco de dados. Ela foi adotada pela ISO e também foi adotada pela ANSI e fornece maior segurança ao banco de dados. (GENNICK, 2000, p.5).

"*Independentemente da ferramenta front-end que está usando, você pode usar a PL/SQL para executar o processamento no servidor em vez de executá-lo no cliente e encapsular regras de negócios e outras lógicas que o negócio exige*". (GENNICK, 2000, p.4).

Figura 3.1 – Código PL/SQL alternando entre o motor PL/SQL e SQL.



Fonte: (Autor)

3.6. A LINGUAGEM JAVA

A linguagem de programação Java, versão 1.7, foi utilizada para a construção da interface com o usuário. A interface gráfica com o usuário – GUI – apresenta um mecanismo amigável ao usuário para interagir com um aplicativo. Uma GUI dá ao aplicativo uma aparência de um comportamento distinto. Fornece aos diferentes aplicativos componentes de interface com o usuário, consistentes e intuitivos. (DEITEL, 2005)

Neste projeto foram utilizados componentes como *JFrame*, *JPanel*, *JCombo*, *JLabel*, *JList*, *JTextFields* entre outros. Para complementar o software, algumas API's foram utilizadas como a:

- AWT – para a criação de eventos dos componentes;
- Swing – para a criação dos componentes;
- Ojdbc14 – para a comunicação com o banco de dados;
- Log4J – para a geração de *logs* customizados;
- RxTxcomm – para a leitura da porta COM7. Desta forma, o que foi lido pela tag chega à interface do usuário, e esse foi o principal motivo da escolha da linguagem Java para construir esta interface.

CAPÍTULO 4 – PROTÓTIPO DO SISTEMA DE CONTROLE DE ATIVIDADES – PETSHOP

4.1. APRESENTAÇÃO GERAL DO PROJETO PROPOSTO

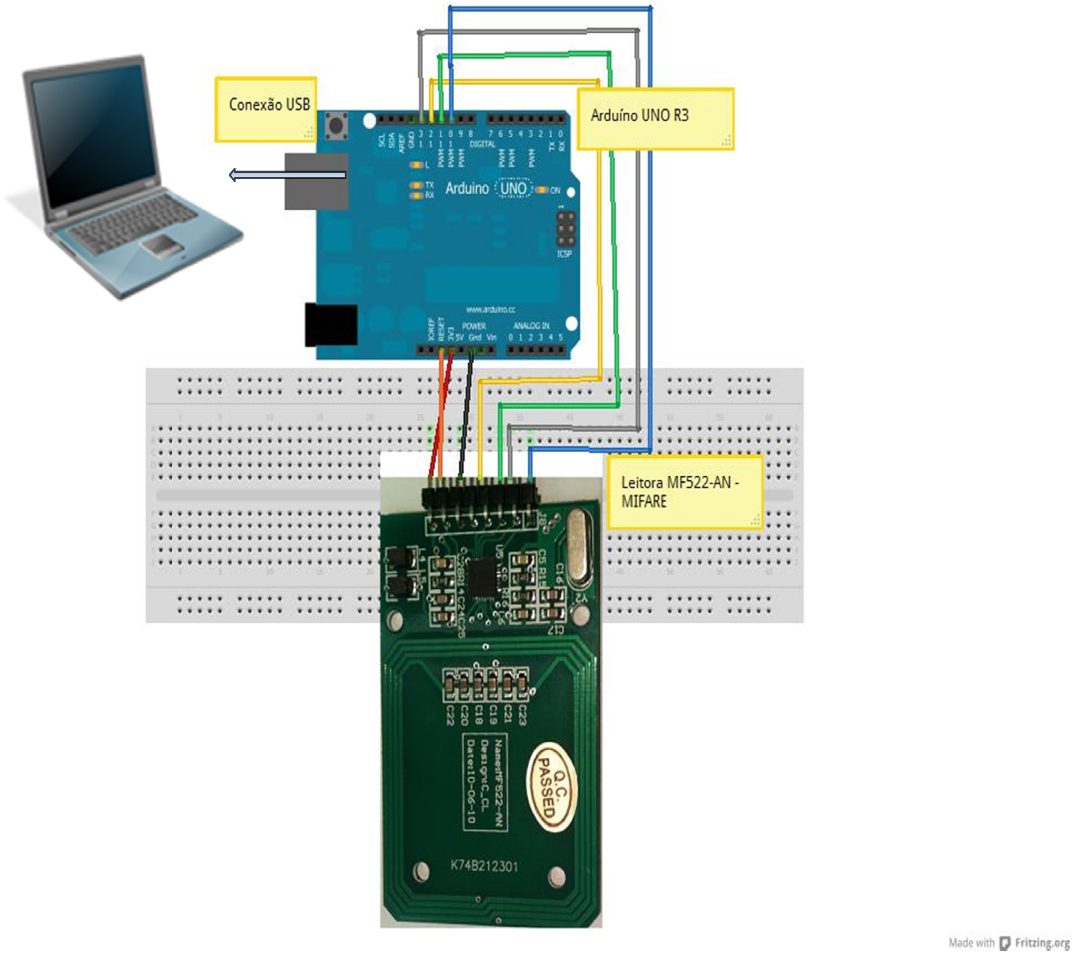


Figura 4.1 - Diagrama geral do Sistema de controle de Atividades de um Pet Shop
Fonte: (Autor)

O sistema é composto de uma placa arduíno uno R3; de uma leitora de RFID – MF522-AN e um notebook contendo a interface com o usuário e um banco de dados.

4.2. DESCRIÇÃO DOS HARDWARES ENVOLVIDOS

4.2.1. ARDUÍNO UNO R3

A placa arduíno uno R3 foi utilizada neste projeto para fazer a comunicação entre a placa MF522-AN e o computador utilizando a interface de comunicação SPI.

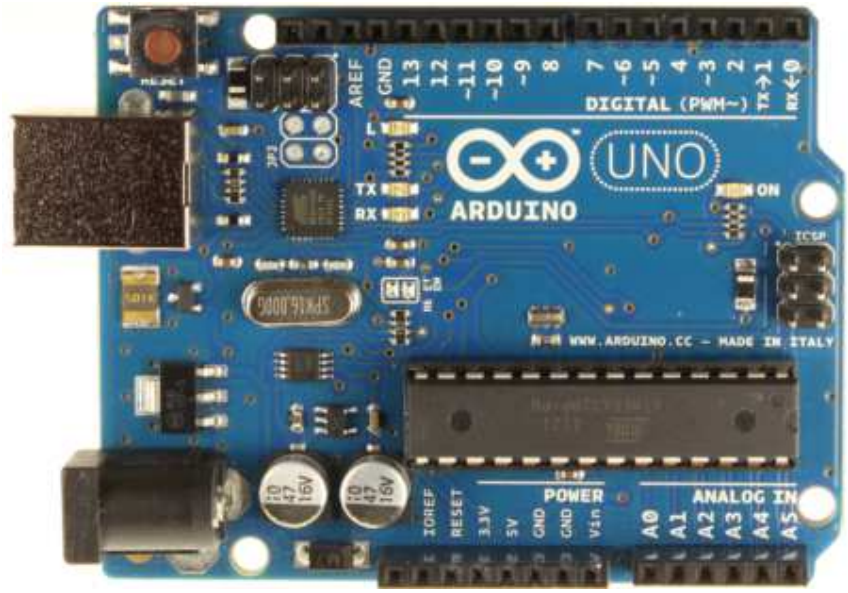


Figura 4.2 – Arduíno Uno R3
Fonte: (ARDUINO2, 2013)

O microcontrolador apresentado na figura 4.2 contém como *microship* ATMEGA328 com sua pinagem apresentada na figura 4.3:

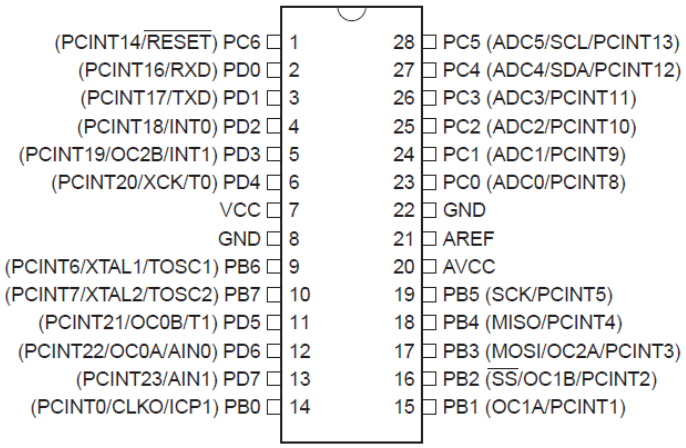


Figura 4.3 - ATmega328 – Microship
Fonte: (ATMEGA, 2009)

O arduíno uno R3 possui quatorze entradas/saídas digitais, das quais seis podem ser usadas como saídas PWM, seis entradas analógicas, um cristal oscilador de 16MHz e um botão de reset. A placa está equipada para acoplar o circuito intergrado e basta conectá-la a um computador com um cabo USB, com a voltagem de 5V ou ligá-la a um adaptador AC/DC ou bateria cujo o fabricante recomende que a tensão fique entre 7V e 12V. O arduíno uno também possui pinos que fornecem energia de 3V a 5V e possui uma memória RAM de 32KB. (ARDUINO, 2013)

4.2.2. LEITORA RFID – MF522-AN

A leitora RFID utilizada neste trabalho foi a MF522-AN produzida pela NPX Semicondutores. É um leitor altamente integrado para comunição sem-fio, utiliza a frequência de 13.56MHz e suporta a interface SPI.

Figura 4.4 – Placa MF522-AN – MIFARE
Fonte: (Autor)

4.3. O SOFTWARE DE CONTROLE LEITURA – ARDUÍNO

Para a utilização da placa arduíno é necessário programá-la. O arduíno possui sua própria linguagem e os seus programas são chamados de *sketches*. Como qualquer linguagem de programação, contém sua estrutura mínima e suas estruturas de controles, variáveis, constantes nativas, funções, tipos de dados e sua sintaxe própria. Ele possui também, sua própria interface de desenvolvimento onde é possível fazer a compilação e *upload* do código para o circuito integrado. A linguagem do arduíno com a finalidade de estender suas funcionalidades também tem suas próprias bibliotecas padrões, entre elas podemos destacar a EEPROM, Ethernet, Firmdata, SPI, Stepper, RFID, entre outras.

Neste trabalho, além da linguagem natural do arduíno, foram utilizadas as bibliotecas SPI e RFID. O código-fonte utilizado para a leitura da tag foi fornecido pela própria biblioteca RFID que contém um exemplo de utilização o qual foi adaptado para a necessidade deste trabalho.

Figura 4.5 – Código-fonte utilizado para fazer a comunicação entre o leitor e o computador utilizando o arduíno.
Fonte: (Autor)

Na figura 4.5 é mostrado a linguagem arduíno sendo desenvolvida no ambiente de desenvolvimento, e possui uma estrutura mínima de configuração – *setup* e de repetições – *loop*. Na configuração, são inicializadas taxa de transmissão dos dados 9600 bits por segundo e a inicialização da biblioteca SPI e RFID. Logo após a configuração, a estrutura de repetição é chamada verificando se é um cartão válido, se válido, imprimindo na porta COM7 os valores lidos.

O código completo pode ser visto no apêndice A.

Na tabela 4.1 é mostrada a configuração de pinos entre a leitora RFID e o arduíno uno R3.

Tabela 4.1 – Configuração dos Pinos MF522-AN – Arduíno Uno R3

Ligação dos Pinos	
PINO MF522-AN	**Pino Arduíno**
1	3,3V
2	Reset
3	GND
4	-
5	12
6	11
7	13
8	10

Fonte: Adptado de (ARDUINO3, 2013) e (LEITOR1, 2007)

4.4. O SOFTWARE DE INTERFACE COM O USUÁRIO

A modelagem deste software é projetado para um atendimento realizado com um usuário e o cliente. Para isso, operações básicas como cadastramento de cliente, criação e tramitação das ordens de serviço foram implementadas.

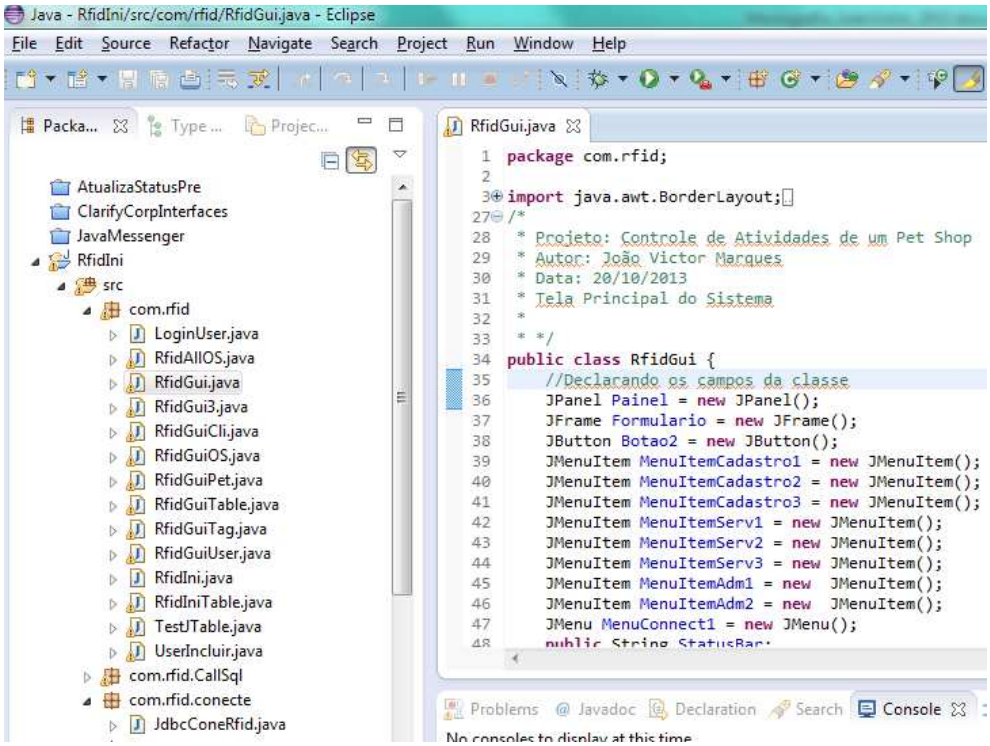


Figura 4.6 – IDE Eclipse Kepler – Projeto: Controle de Atividades de um Pet Shop
Fonte: (Autor)

A interface com o usuário, construída em Java através do ambiente de desenvolvimento Eclipse Kepler, é um processo interativo. O estilo de interação adotada neste projeto foi por seleção de menus e preenchimento de formulários seguindo um fluxo lógico. A entrada de dados é simples, podendo usar *JComb* para a escolha de certas informações evitando problemas quanto às regras de preenchimentos dos campos que, em muitos casos, não são atendidas pelo usuário e *TextFields* para digitação dos dados, além de botão para gerar os eventos necessários. Essa interface ensina o usuário a manipular o sistema através das informações geradas no rodapé do *JFrame.*

A seguir, são apresentadas as possíveis ações do usuário de acordo com o perfil:

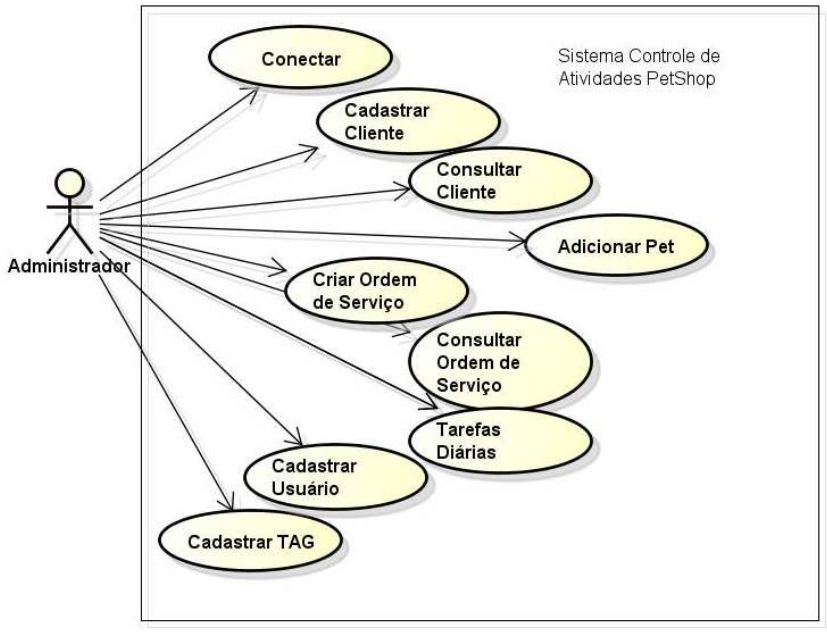


Figura 4.7 – Ações do administrador
Fonte: (Autor)

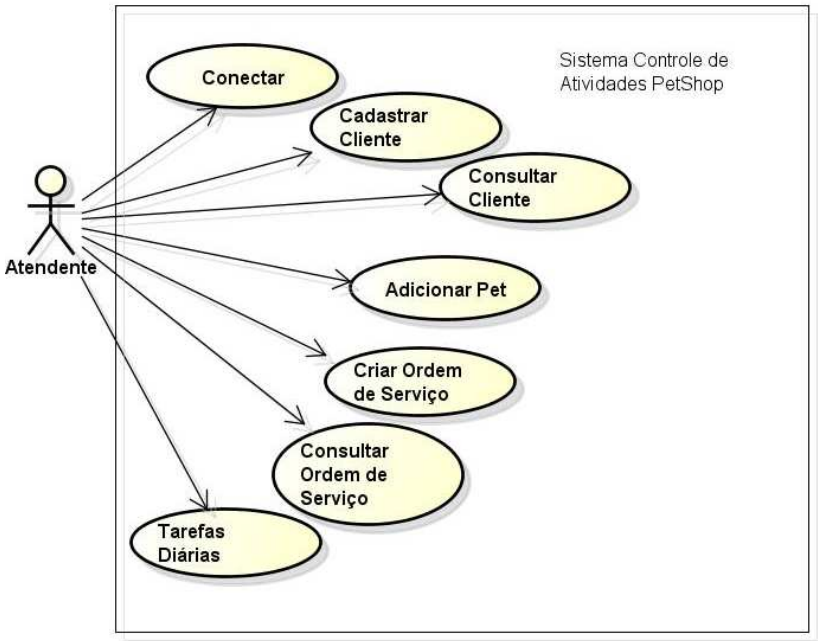


Figura 4.8 – Ações do Atendente
Fonte: (Autor)

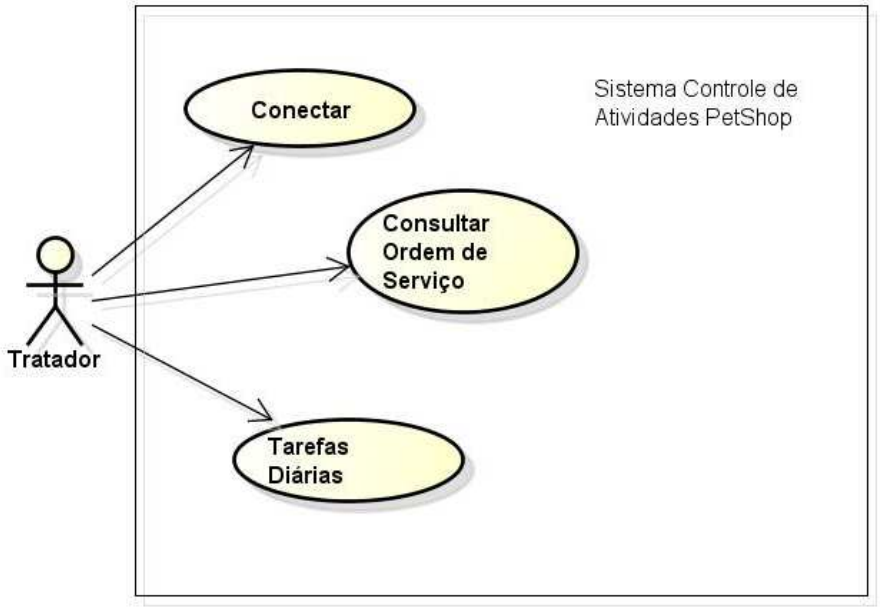


Figura 4.9 - Ações do Tratador
Fonte: (Autor)

4.5. CRIAÇÃO DO BANCO DE DADOS

Entre estas formas de criar/instalar o banco de dados, há duas: via comando, "*create database*", e via assistente de configuração de banco de dados. A forma escolhida foi criação do banco de dados via assistente de configuração, dada a sua facilidade e eficiência.

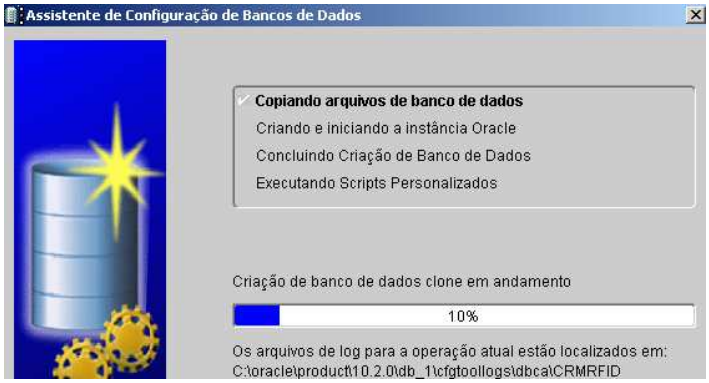


Figura 4.10 – Início da criação do banco de dados
Fonte: (Autor)

Figura 4.11 – Fim da criação do banco de dados
Fonte: (Autor)

Com o banco de dados Oracle criado, iniciou-se o processo de criação do schema, que foi nomeado de SA e posteriormente os objetos do banco. Entre os objetos criados, encontram-se as *tables* e seus respectivos *indexes* e *sequences*, as *procedures* e *functions*, onde foi implementada a maioria das regras de negócios[1].

Tabela 4.2 – Objetos do banco de dados – CRMRFID

Objeto - Tabelas	**Descrição**
TABLE_CLIENTE	tabelas de clientes
TABLE_CONTROL_USER	tabela de controle de Id's
TABLE_ENDERECO	tabela de endereço
TABLE_OSRFID	tabela de Ordem de Serviços
TABLE_PET	tabela de animais de estimação
TABLE_PRIVILEGIO	tabela de perfis
TABLE_TAGRFID	tabela de tags
TABLE_USUARIO	tabela de usuários do sistema
Objeto - Indexes	**Descrição**
IDX_ID	Coluna ID da tabela de usuários
IDX_ID_CLI	Coluna ID da tabela de Clientes
IDX_ID_END	Coluna ID da tabela de endereços
IDX_ID_OS	Coluna ID da tabela de Ordem de Serviços
IDX_ID_PET	Coluna ID da tabela de animais de estimação
IDX_ID_PRIV	Coluna ID da tabela de perfis
IDX_ID_TAG	Coluna ID da tabela de tags
Objeto - Procedures	**Descrição**
INTP_ATUALIZA_OS	Responsável pela atualização de status das Ordens de Serviços
INTP_CADASTRA_TAG	Responsável pelo cadastramento das novas tags
INTP_CONSULTA_OS	Responsável pelas consultas de uma Ordem de Serviço
INTP_GERA_OSRFID	Responsável pela Geração das Ordens de Serviços
INTP_INSERE_CLIENTE	Responsável pela criação de novos clientes
INTP_INSERE_PET	Responsável pela associação de mais pet ao dono
INTP_INSERE_USUARIO	Responsável pela criação de novos usuários do sistema

1 Regras de Negócio – São quaisquer condições lógicas que devem ser verdadeiras no final de uma transação.(MULLER, 2002)

Objeto - Functions	**Descrição**
INTF_NEXT_ID	Controla os IDs que nomeiam as ordens de serviços e usuários
Objeto - Sequences	
SEQ_CLIENTE	Retornam números sequencias para registros na tabela de clientes
SEQ_END	Retornam números sequencias para registros na tabela de endereços
SEQ_OSRFID	Retornam números sequencias para registros na tabela de O.S's
SEQ_PET	Retornam números sequencias para registros na tabela de Pets
SEQ_PRIVILEGIO	Retornam números sequencias para registros na tabela de perfis
SEQ_TAGRFID	Retornam números sequencias para registros na tabela de tags
SEQ_USUARIO	Retornam números sequencias para registros na tabela de usuários

Na tabela 4.2, são mostrados todos os objetos criados para o funcionamento do banco de dados CRMRFID. Esses objetos foram criados utilizando o ambiente de desenvolvimento *SQL Navigator*, versão 6.5, produzido pela *Quest Software*. O *SQL Navigator* foi configurado para acesssar o banco CRMRFID, através de uma configuração no serviço de escuta chamado *TNSListener*. Este serviço é *started* na inicialização do banco de dados e ele pode ser visto no Gerenciador de Serviços do Windows conforme pode ver na figura 4.12.

O código completo pode ser visto no apêndice B.

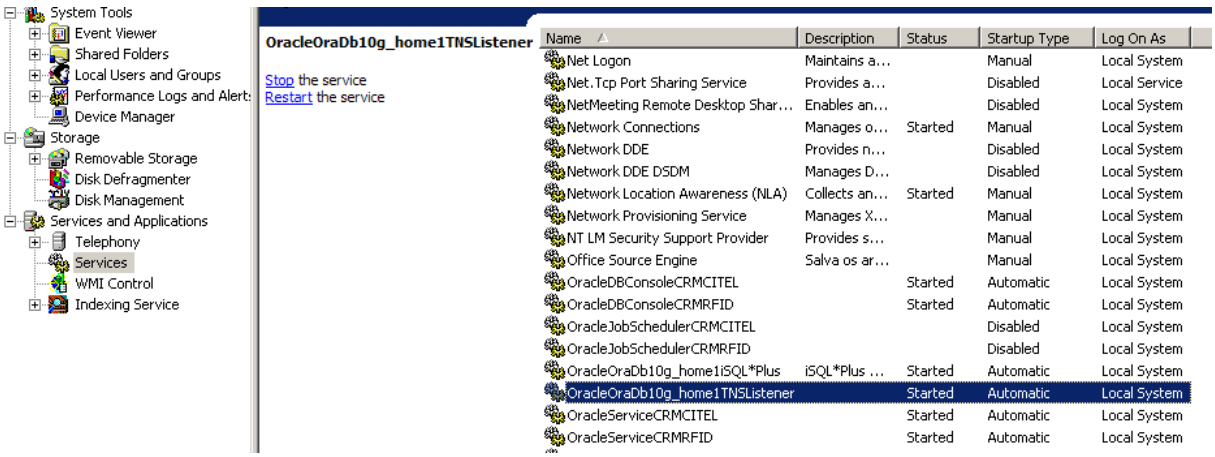


Figura 4.12 – Serviço *TNSListener* iniciado
Fonte: (Autor)

Para fazer requisições entrantes no banco de dados é necessária uma configuração em um arquivo chamado *TNSnames* no computador cliente. Nesse arquivo são configurados o endereço, a porta e o nome do serviço.

A seguir a configuração do *TNSnames*:

```
CRMRFID =
 (DESCRIPTION =
   (ADDRESS_LIST =
     (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.56.10)(PORT = 1521))
      )
     (CONNECT_DATA =
       (SERVICE_NAME = CRMRFID)
     (SERVER = DEDICATED)
    )
 )
```

Na figura 4.13 é mostrado o ambiente de desenvolvimento do *SQL Navigator*.

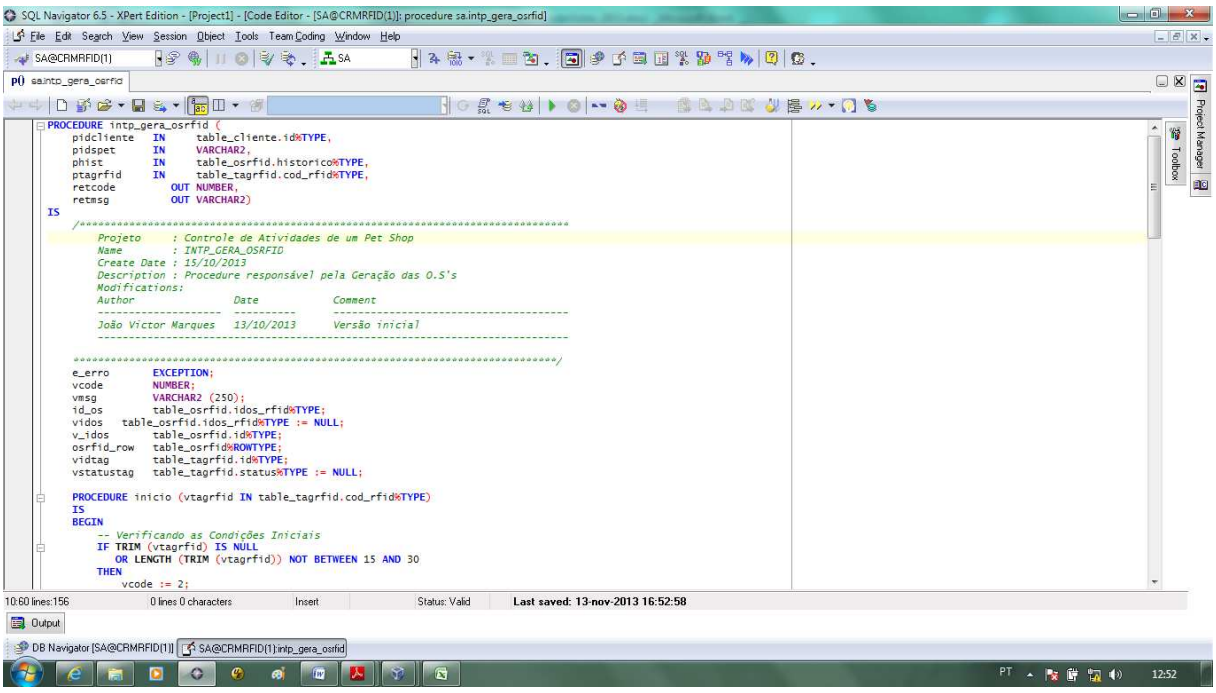


Figura 4.13 – IDE SQL Navigator – Procedure intp_gera_osrfid sendo desenvolvida
Fonte: (Autor)

4.6. CONEXÃO COM O BANCO DE DADOS

Com o banco de dados criado e desenvolvidos seus objetos, iniciou-se a fase de conexão do sistema com o banco de dados. Nesse contexto, a linguagem Java, utilizada na construção

do sistema, possui uma API chamada JDBC[1]. Esta API fornece um conjunto de interfaces que criam um ponto comum, no qual aplicações e serviços de um banco de dados podem ser encontrados. (MULLER, 2002). Quem faz o papel da comunicação ser efetivada é chamado de *Driver JDBC*[2]. Este precisa ser especificado, pois cada fornecedor de banco de dados tem o seu *driver* específico. Para este sistema foi utilizado o ojdbc14.

Figura 4.14 – Cada *DriverManager* possui seu *driver* JDBC específico.
Fonte: (Autor)

1 JDBC – *Java Database Connector*
2 *Driver* JDBC – É o conjunto das classes que implementam as interfaces JDBC para um determinado mecanismo de banco de dados .(MULLER, 2002, p. 30)

CAPÍTULO 5 – A IMPLEMENTAÇÃO

5.1. MODELAGEM DA INTERFACE COM O USUÁRIO

A autenticação é o primeiro processo possível a ser feito quando o sistema é iniciado. Após a autenticação, o usuário, de acordo com o seu perfil, tem acesso as seguintes funcionalidades:

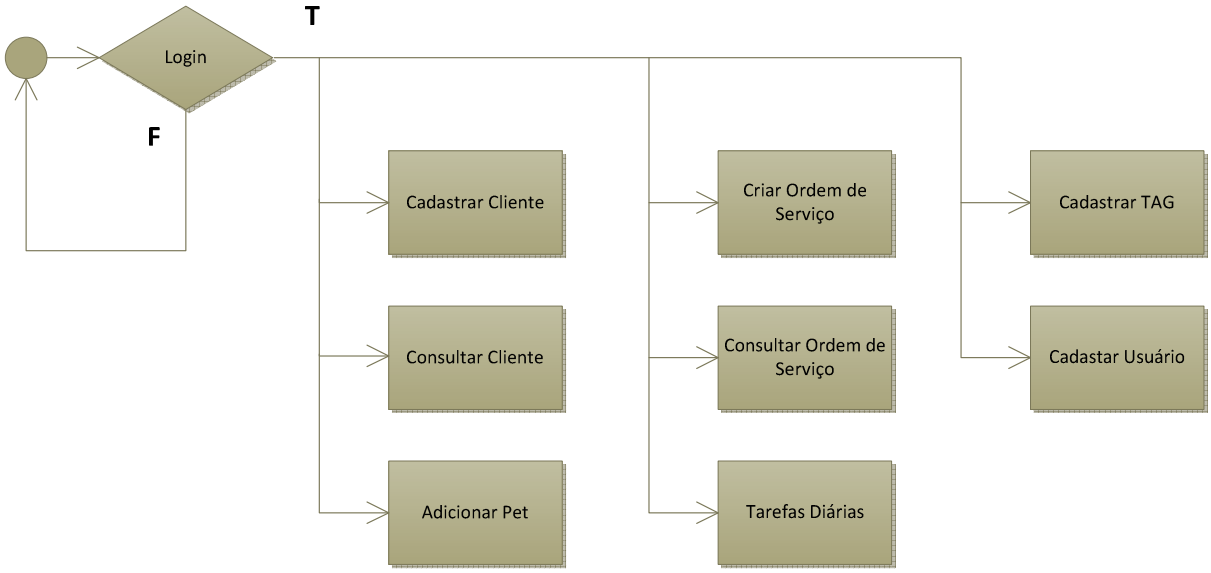


Figura 5.1 – Fluxograma geral do Sistema de Controle de Atividades
Fonte: (Autor)

5.2. AUTENTICAÇÃO

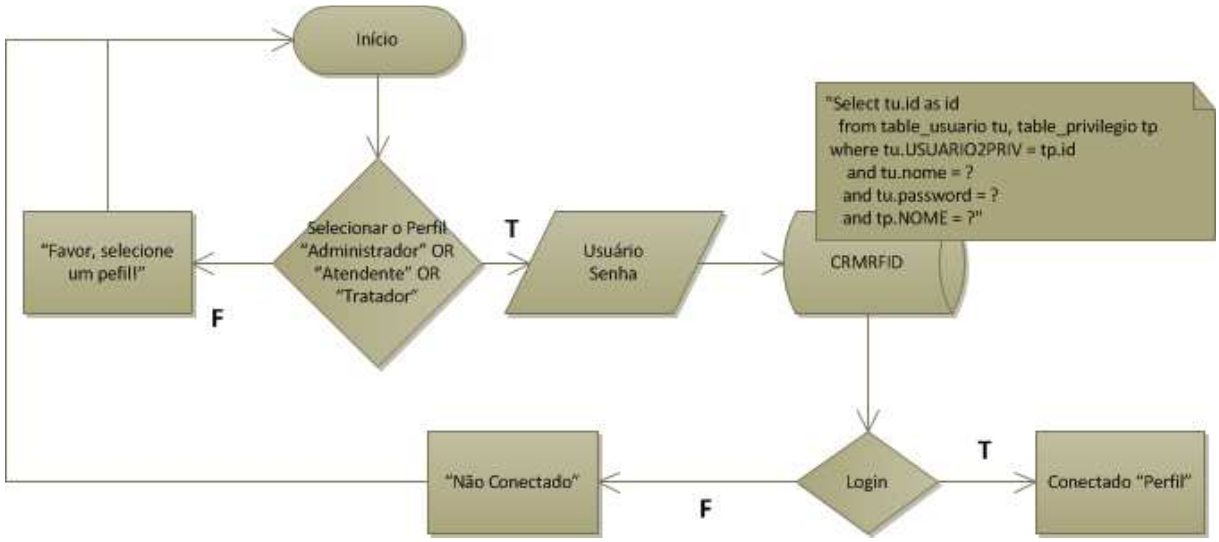


Figura 5.2 – Autenticação
Fonte: (Autor)

Na figura 5.2 vemos através do fluxograma todas as etapas para que um usuário se conecte ao sistema.

O sistema verifica se os dados de entrada conferem com os dados cadastrados no banco de dados. Estando correto, o acesso é liberado. Caso haja alguma divergência irá apresentar a mensagem “Não conectado”.

5.2.1. CADASTRAR CLIENTE

O fluxo de cadastrar um cliente exige uma verificação se o mesmo já não está cadastrado. Segue o fluxo de cadastrar um cliente:

- Preencher o formulário;
- Verificar se já não está cadastrado;

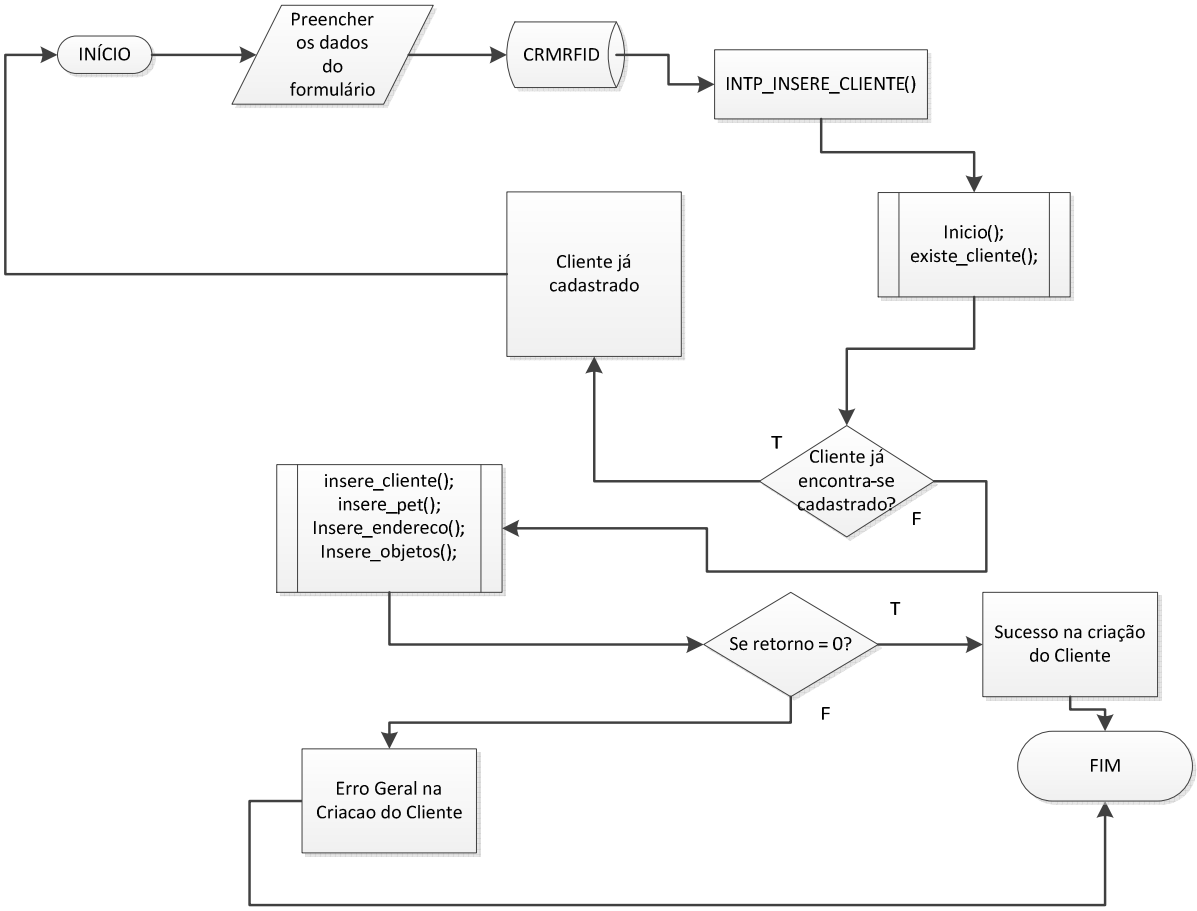


Figura 5.3 – Cadastro de um novo cliente
Fonte: (Autor)

5.2.2. CONSULTAR CLIENTE

Ao consultar um cliente, o usuário está interessado em saber o nome, endereço, os pets associados a ele, assim como os serviços que já foram realizados com o cliente.

Para tanto. Seguem os fluxos de Consulta do Cliente:

- Procurar o cliente;
- Procurar os animais de estimação vinculados ao cliente;
- Procurar todas as ordens de serviços realizados;
- Visualizar o conteúdo das ordens de serviços.

Na figura 5.4 é mostrado o fluxo da procura por um cliente ao clicar no botão “Procurar Cliente”.

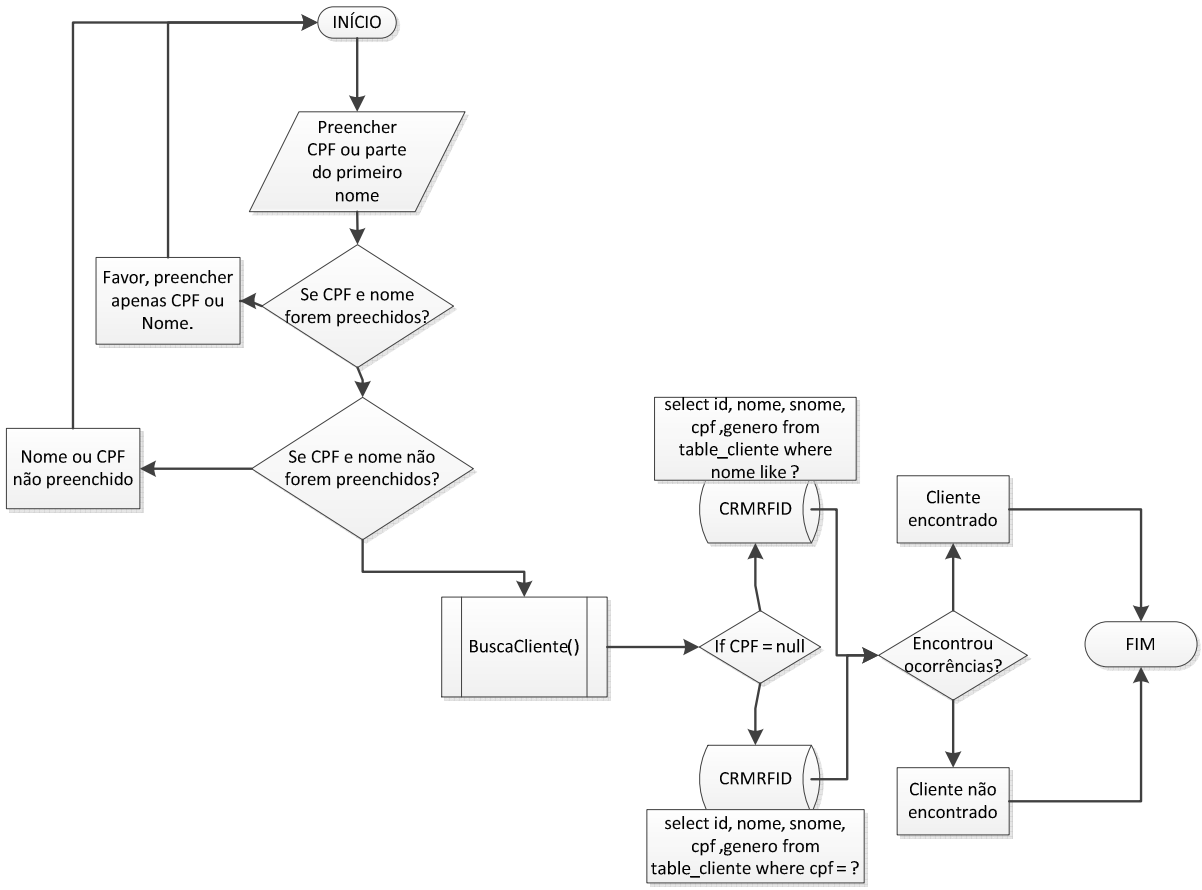


Figura 5.4 – Consultar Cliente - Procurar Cliente
Fonte: (Autor)

Na figura 5.5 é mostrado o fluxo da procura pelos animais de estimação vinculados ao cliente selecionado.

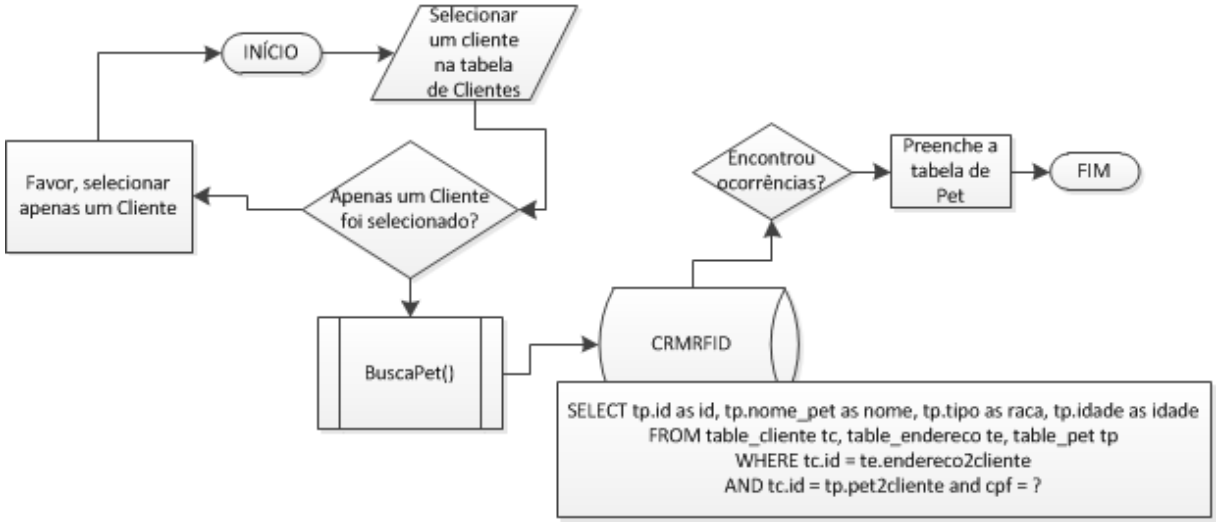


Figura 5.5 – Consultar Cliente - Procurar Pets
Fonte: (Autor)

Na figura 5.6 é mostrado o fluxo da busca de todas as Ordens de Serviço já realizadas, vinculadas ao cliente.

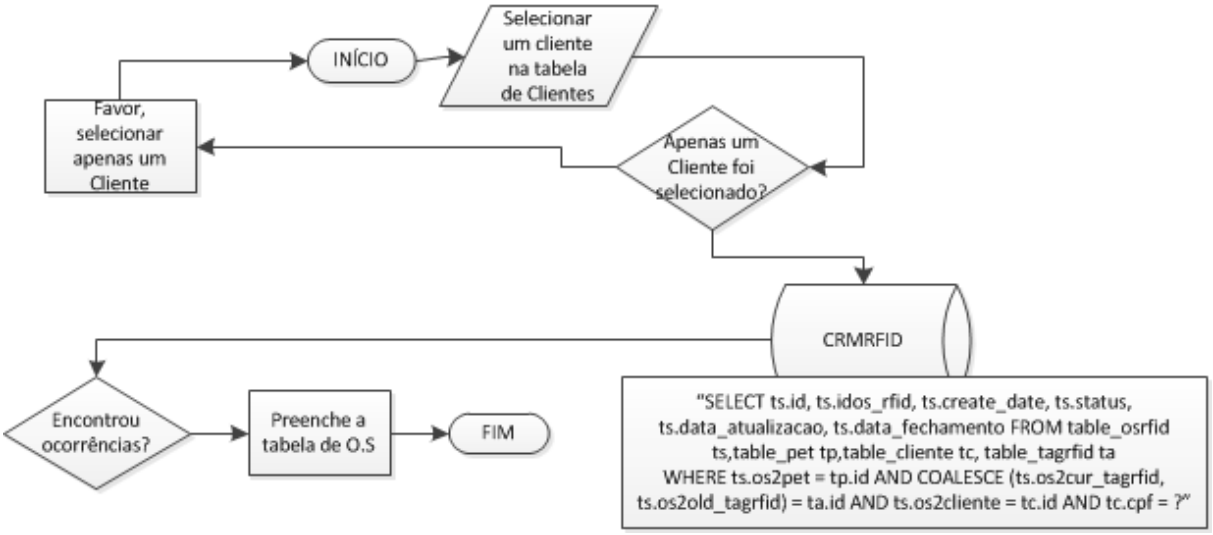


Figura 5.6 – Consultar Cliente – Procurar O.S
Fonte: (Autor)

Com as ordens de serviços já listadas é possível saber os detalhes de cada serviço clicando no botão “Detalhes da O.S”. Na figura 5.7 é mostrado o fluxo dessa busca.

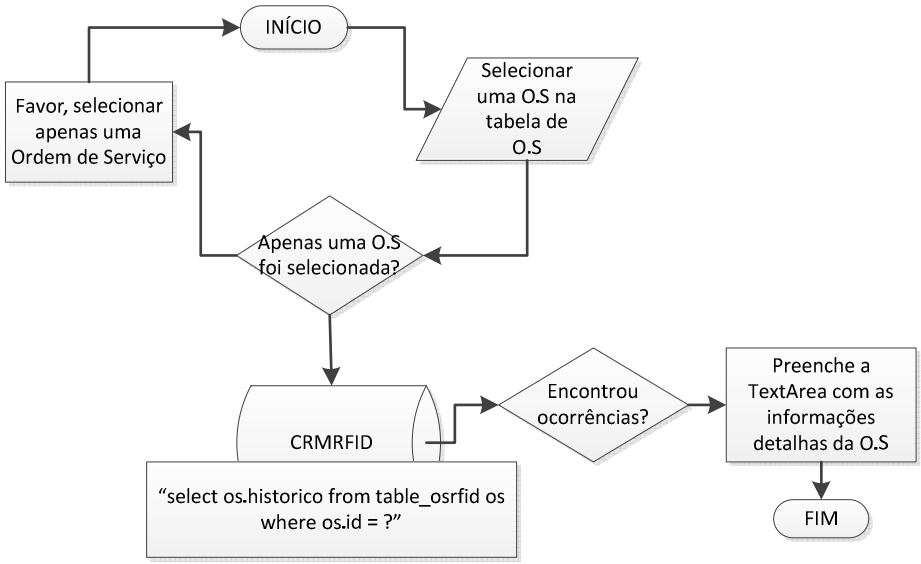


Figura 5.7 – Consultar Cliente – Detalhes da O.S
Fonte: (Autor)

5.2.3. ADICIONAR PET

No modelo relacional que será apresentado mais adiante, está especificado que um cliente pode ter mais de um animal de estimação. Logo, o fluxo abaixo foi pensado nisso, pois no fluxo de cadastramento do usuário só há espaço para inserir um único animal. Segue o fluxo de adicionar Pet.

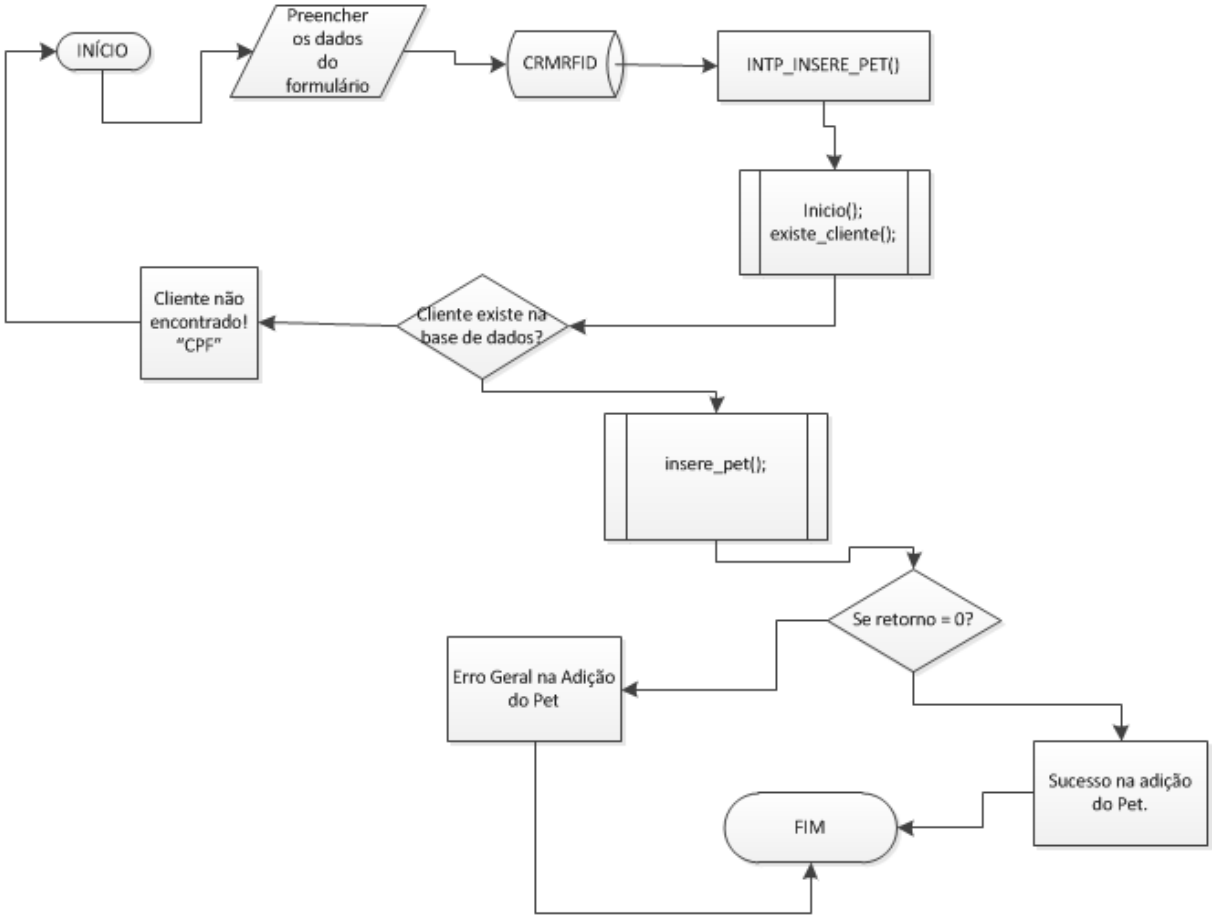


Figura 5.8 – Fluxo adicionar animais de estimação associado a um cliente
Fonte: (Autor)

5.2.4. CRIAR ORDEM DE SERVIÇO

O fluxo de criar uma ordem de serviço exige a leitura da tag, e somente assim poderá ser criado a Ordem de Serviço.

Segue o fluxo de Criar Ordem de Serviço:

- Procurar o Cliente
- Procurar os animais de estimação vinculados ao cliente;
- Selecionar os serviços a serem executados;
- Executar a leitura da tag;
- Gerar a O.S;

A seguir mostra o fluxo da procura por um cliente ao clicar no botão “Procurar”.

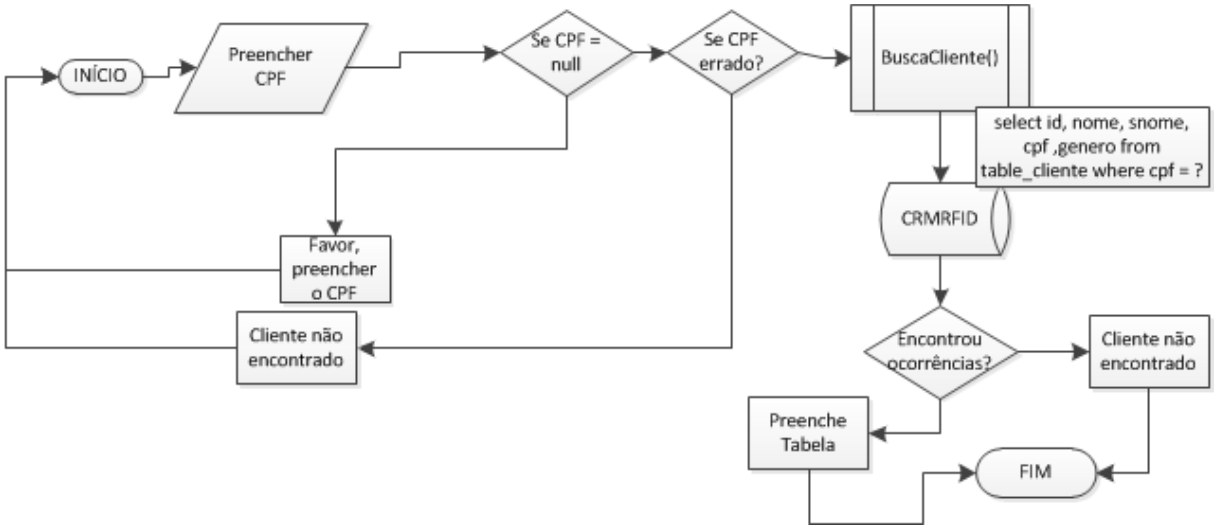


Figura 5.9 – Criar Ordem de Serviço – Procurar
Fonte: (Autor)

Após o Cliente encontrado, procurar pelos Pets vinculados ao Cliente.

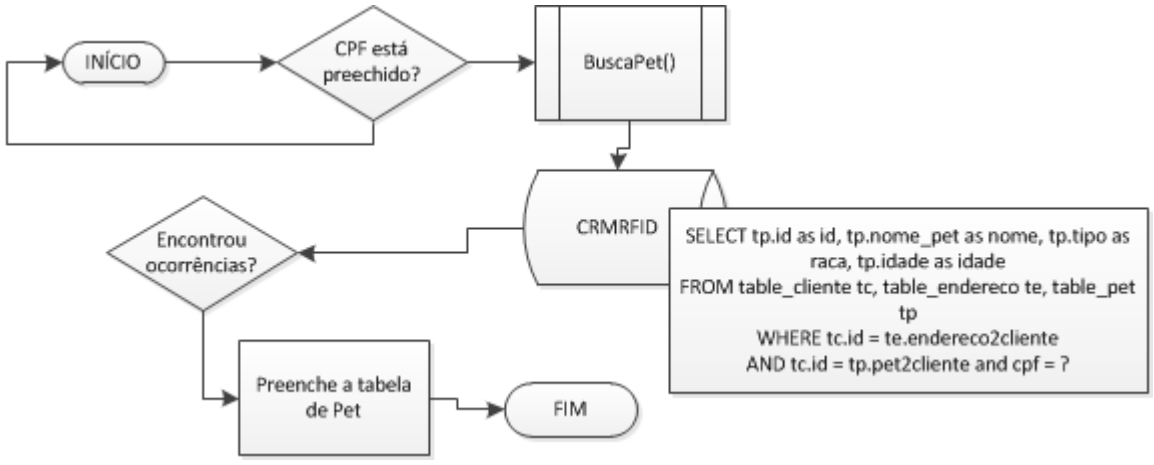


Figura 5.10 - Criar Ordem de Serviço – Procurar Pets
Fonte: (Autor)

No *JCombList* seleciona os serviços a serem executados com o animal e realiza a leitura da TAG.

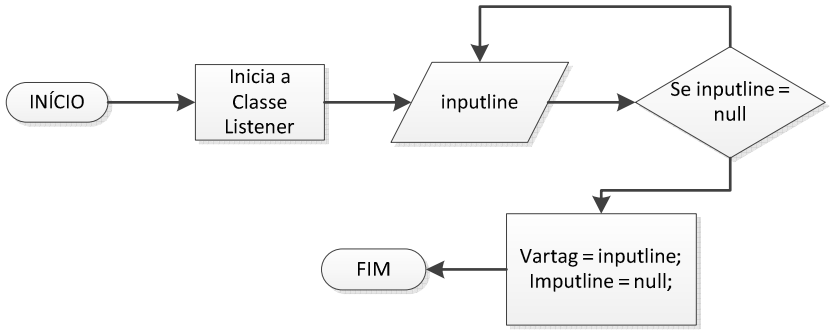


Figura 5.11 - Criar Ordem de Serviço – Selecionar Serviços
Fonte: (Autor)

Somente após a leitura da tag que o sistema deixar gerar uma Ordem de Serviço, clicando no botão “GerarOS”.

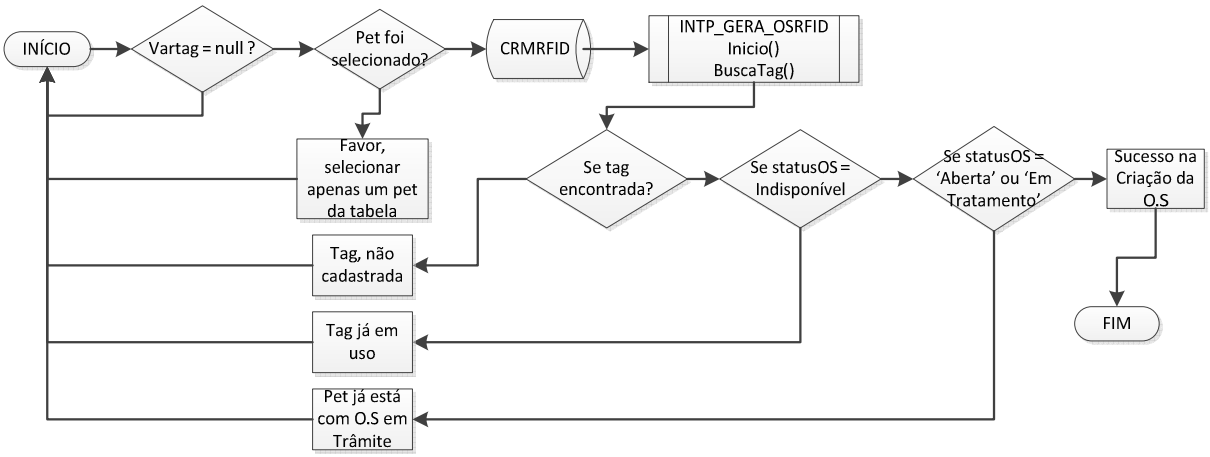


Figura 5.12 – Criar Ordem de Serviço – Gerar Ordem de Serviço
Fonte: (Autor)

5.2.5. CONSULTAR ORDEM DE SERVIÇO

O fluxo de consultar uma ordem de serviço pode ser realizado tanto pelo ID da Ordem de Serviço, ou pela tag associada.

Para tanto, segue o fluxo de Consultar Ordem de Serviço:

- Procurar por tag;

- Procurar por Id da O.S;

A seguir o fluxo da procura por O.S utilizando a tag, clicando no botão “Ler Tag”.

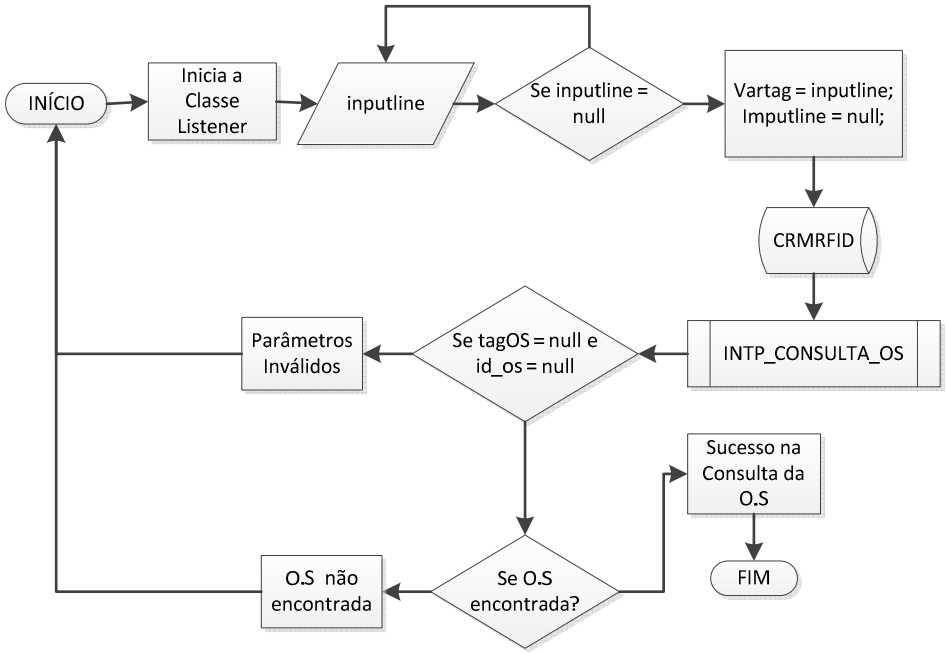


Figura 5.13 – Consultar Ordem de Serviço – Procurar por tag
Fonte: (Autor)

A outra consulta por O.S pode ser feito preenchendo o *JTextField* “Ordem de Serviço”.

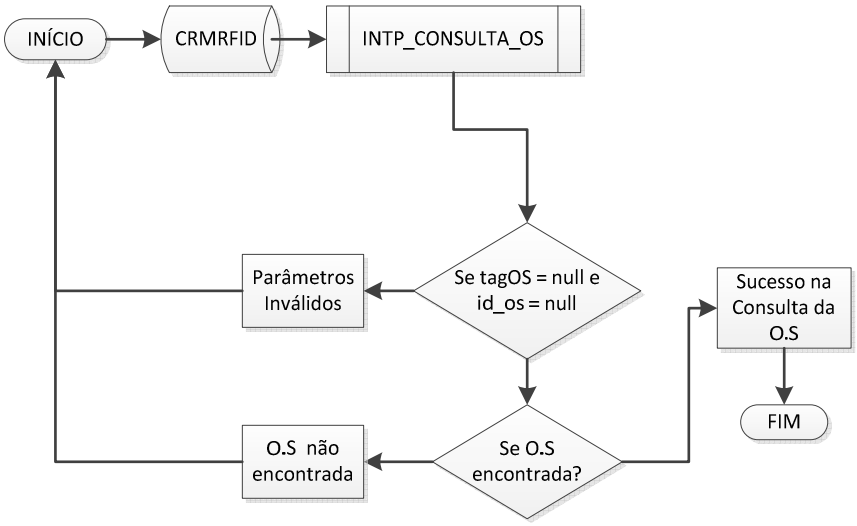


Figura 5.14 – Consultar Ordem de Serviço – Procurar por id O.S
Fonte: (Autor)

5.2.6. TELA DIÁRIA

Esta tela tem uma particularidade, pois com ela é possível verificar todas as O.S's do sistema ou filtrando por status: 'Aberta', 'Em Tratamento' ou 'Fechada'.

A seguir, o fluxo para realizar essa busca.

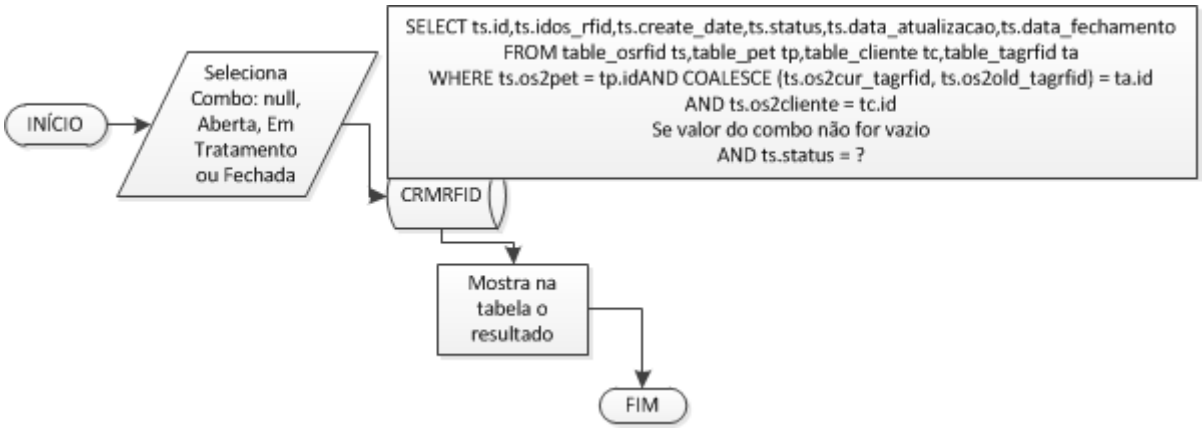


Figura 5.15 – Tela Diária
Fonte: (Autor)

5.2.7. CADASTRAR TAG

Esta tela tem a funcionalidade de cadastrar as novas tags adquiridas para serem utilizadas no estabelecimento de Pet Shop.

Seguem os fluxos de Cadastrar Tags:

- Leitura da Tag;
- Inserção do valor lido no banco de dados;

Primeiramente, temos a leitura da Tag. Iniciando a classe *Listener* que fica escutando o que chegar na porta COM7. Quando o valor chega, a variável *inputline* é preenchida e a leitura é efetuada, e posteriormente é preenchido o *label* para a visualização do usuário.

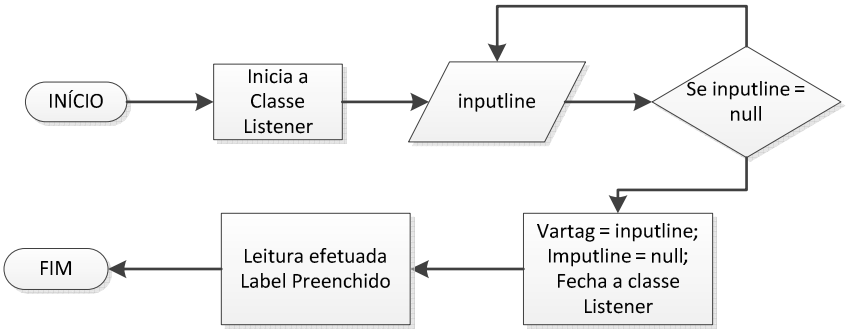


Figura 5.16 – Cadastrar tag – Leitura
Fonte: (Autor)

Após o preenchimento do *label* é possível cadastrar a tag no banco de dados, desde que já não esteja cadastrada.

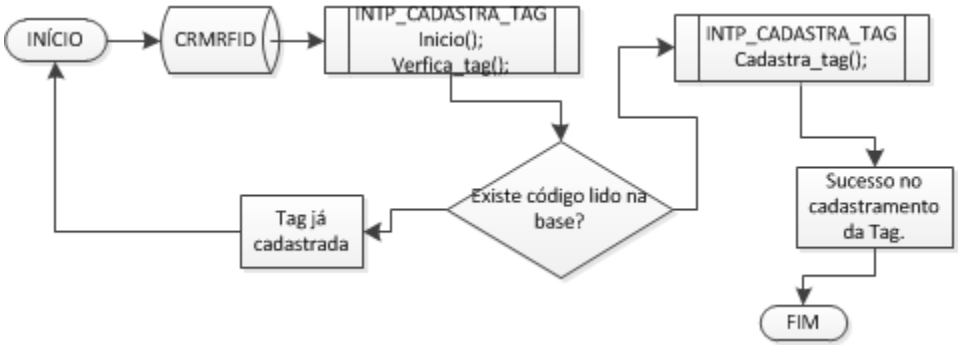


Figura 5.17 – Cadastrar tag – Inserção
Fonte: (Autor)

5.2.8. CADASTRAR USUÁRIO

Esta tela tem a funcionalidade de cadastrar os novos usuários do sistema. Os perfis: Administrador, Atendente e Tratador.

Segue o fluxo do cadastramento de usuário.

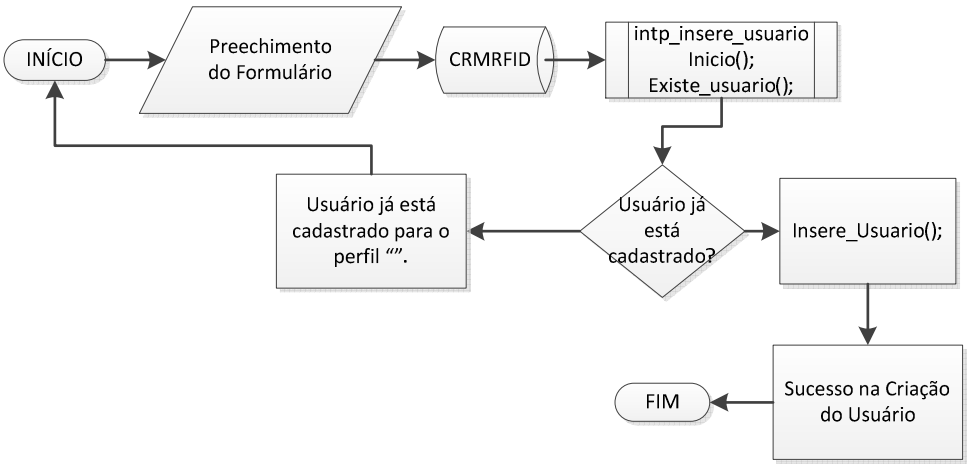


Figura 5.18 – Cadastrar Usuário
Fonte: (Autor)

5.3. M ER – BANCO DE DADOS ORACLE

O modelo de entidade de relacionamento foi utilizado neste projeto para descrever de maneira conceitual e lógica a representação das entidades, que são as tabelas do banco de dados responsáveis por guardarem os dados referentes ao Cliente, Animais de Estimação, Ordens de serviço e tags.

Na Figura 5.19 é mostrado o MER, entre as tabelas do banco de dados.

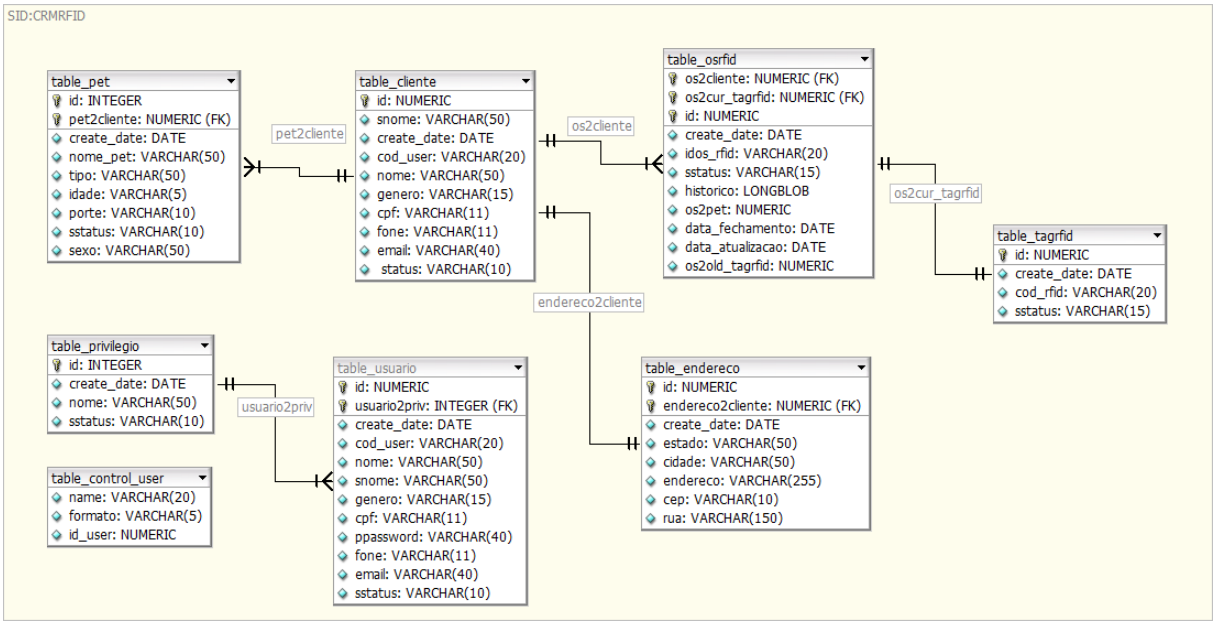


Figura 5.19 – MER – CRMRFID
Fonte: (Autor)

Neste modelo relacional, as entidades interagem umas com as outras, onde um cliente pode ter vários pets, um endereço e várias ordens de serviços. Uma ordem de serviço só pode estar associada a uma tagrfid, e um privilégio pode estar associado a vários usuários.

5.4. TESTES DO SISTEMA DE CONTROLE DE ATIVIDADE DE UM PET SHOP

Os testes realizados englobaram todas as telas, realizando as ações que o usuário fará para o controle de atividades.

Os testes foram constituídos na funcionalidade de cada ação. Para tanto, foi criado: usuário com diferentes perfis, criado cliente, associado animal de estimação a cliente, criado ordem de serviço, consultado ordem de serviço via tagrfid, cadastro de tag. Em todos os testes, houve sucesso no que lhe era proposto.

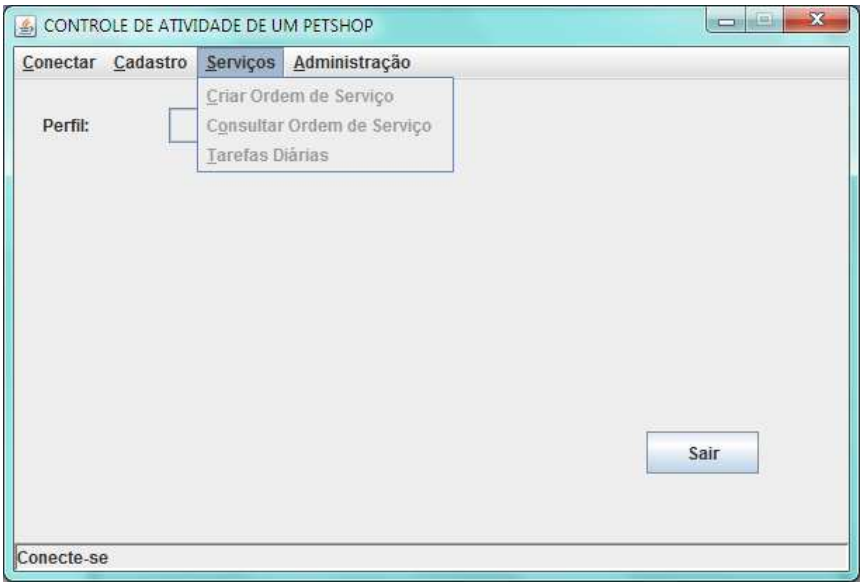


Figura 5.20 – Tela Inicial
Fonte: (Autor)

Na figura 5.20 é mostrada a tela inicial do software. Como ainda não houve uma autenticação, os menus que dão acesso às funcionalidades do sistema não estão habilitados. O único menu que fica habilitado é o “Conectar”. Escolhendo o perfil e

preenchendo o nome e senha corretamente, tem-se a autenticação concluída com sucesso.

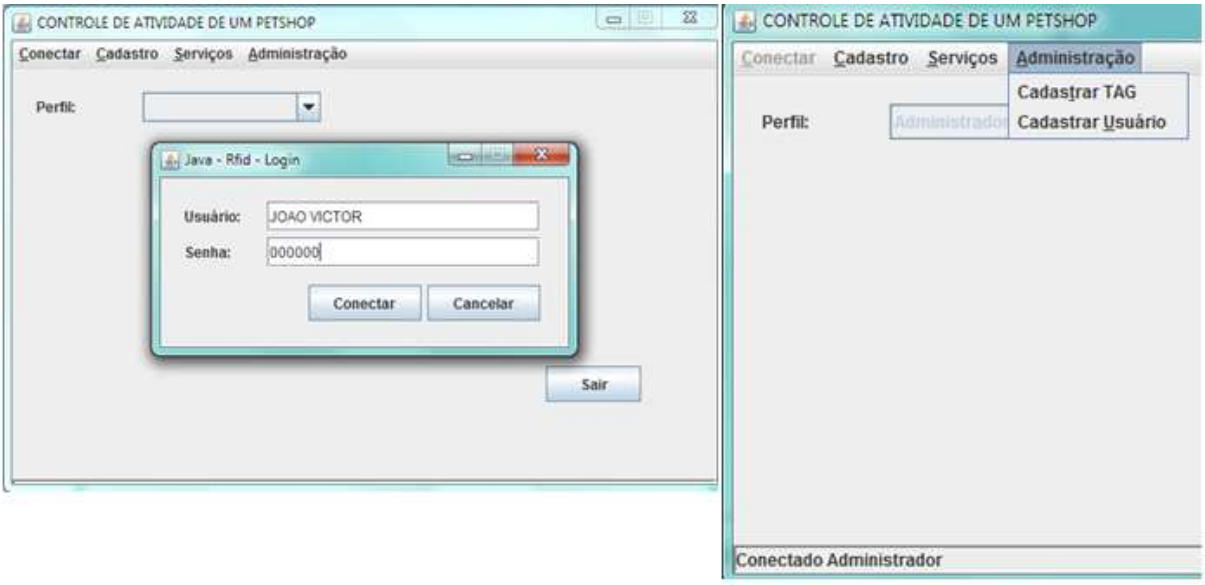


Figura 5.21 – Autenticação
Fonte: (Autor)

Caso não escolha o perfil, quando for tentar acessar o submenu “Conecte” para preencher o nome e senha, uma mensagem irá aparecer, solicitando a seleção de um perfil – ver figura 5.22.

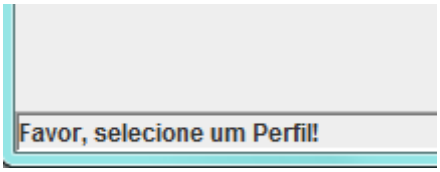


Figura 5.22 – Autenticação – aviso
Fonte: (Autor)

Na figura 5.23 é mostrado o cadastramento de um novo cliente, acessando o menu “Cadastro”, submenu “Cadastrar Cliente” e obtendo a mensagem que o cliente foi cadastrado com sucesso.

Figura 5.23 – Incluir Cliente
Fonte: (Autor)

Na figura 5.24 é mostrada a tentativa de gerar uma Ordem de Serviço para este novo cliente não lendo a tag para ser associada. O Sistema lança uma mensagem avisando que a tag não foi lida.

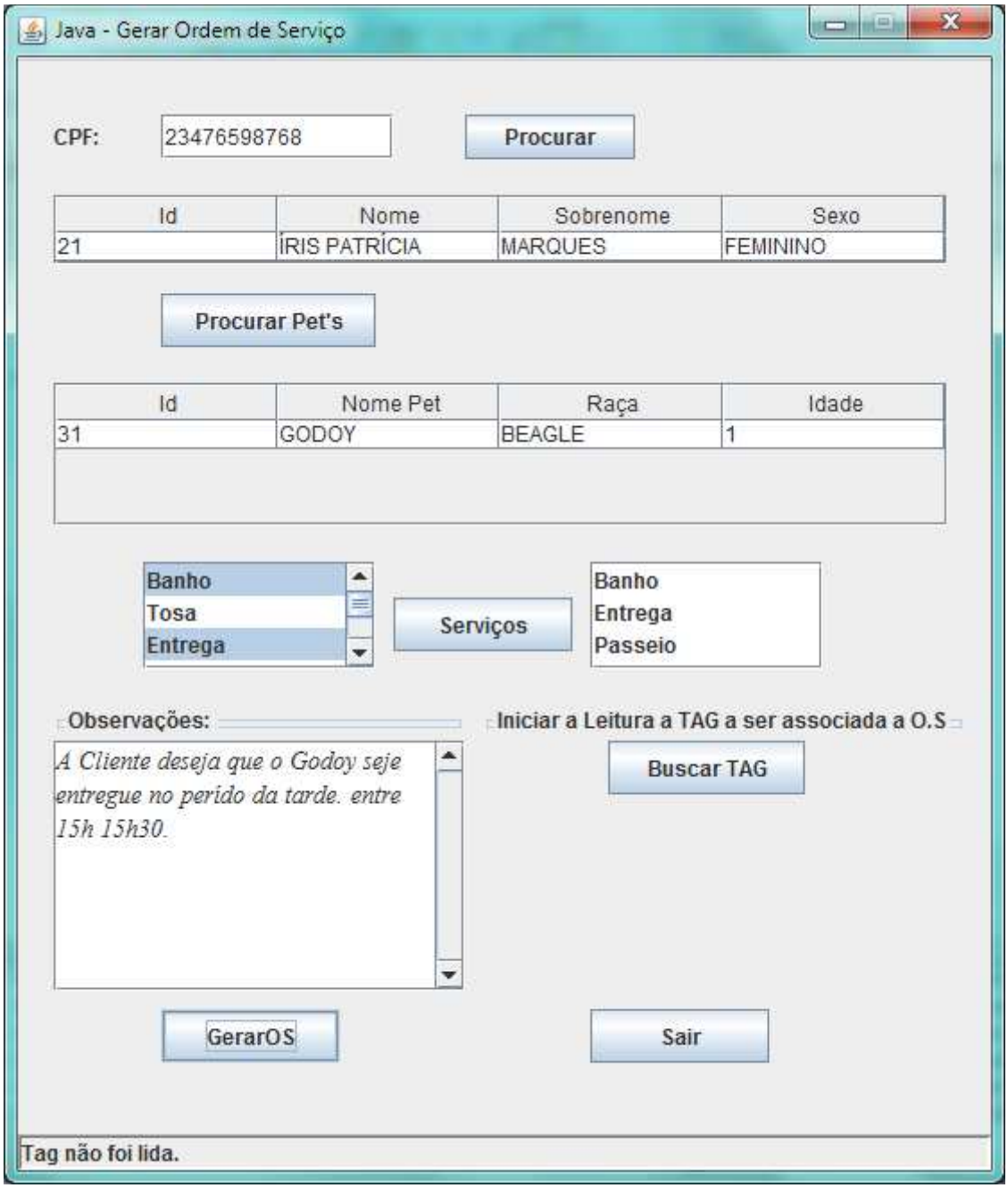


Figura 5.24 – Gerar Ordem de Serviço – Aviso
Fonte: (Autor)

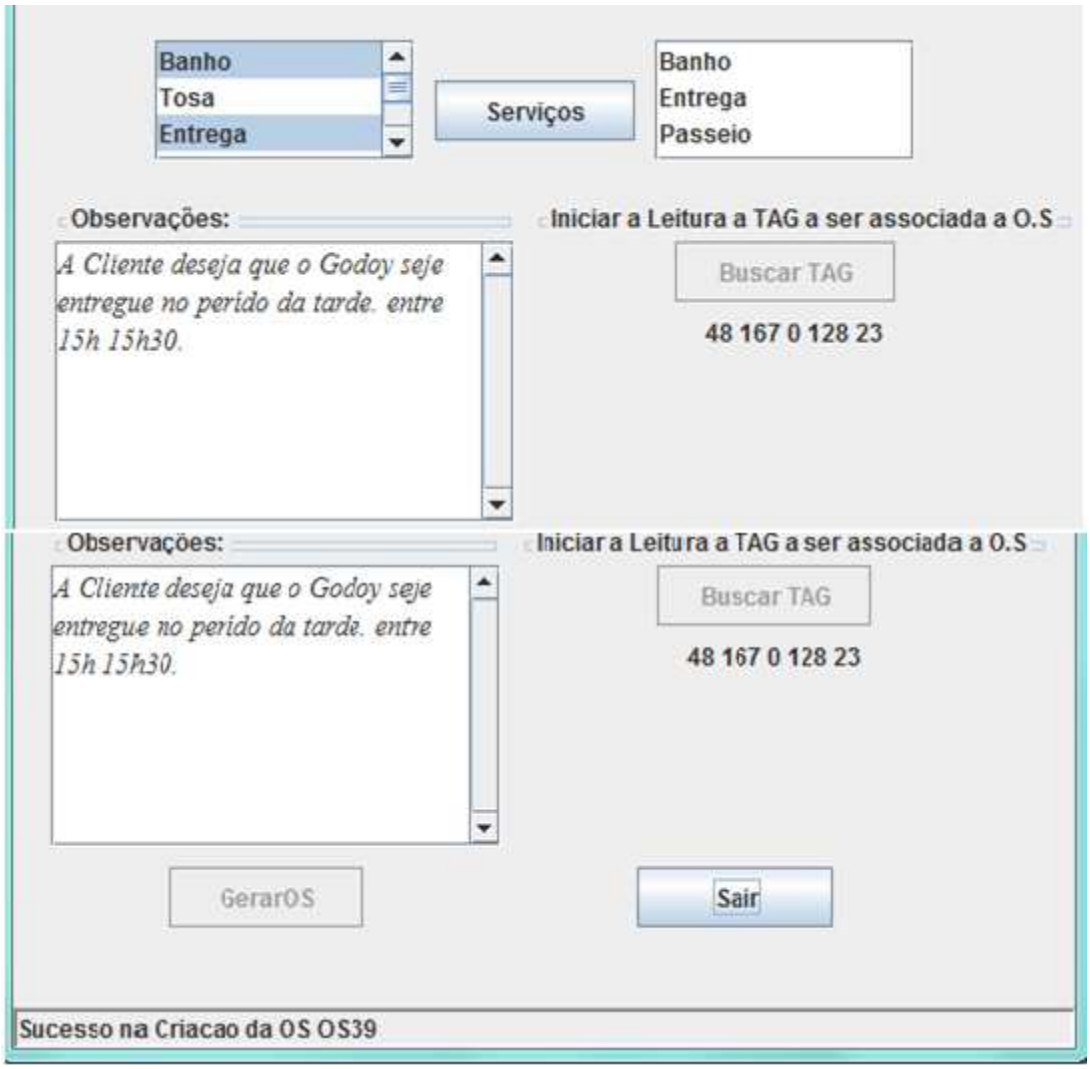


Figura 5.25 – Geração de ordem de serviço com sucesso
Fonte: (Autor)

Após a leitura da tag e clicando no botão “GerarOS”, o ordem de serviço foi criada com sucesso, conforme indicado na figura 5.25.

Com a OS39 criada, podemos através da tag, que agora está na coleira do cachorro, pesquisá-la na tela de “Consultar Ordens de Serviço” e verificar os serviços a serem feitos com o animal. Como a OS39 ainda está aberta, a tag ainda está associada a este serviço e podemos fazer essa pesquisa da ordem de serviço, utilizando a tag. A figura 5.26 ilustra essa situação.

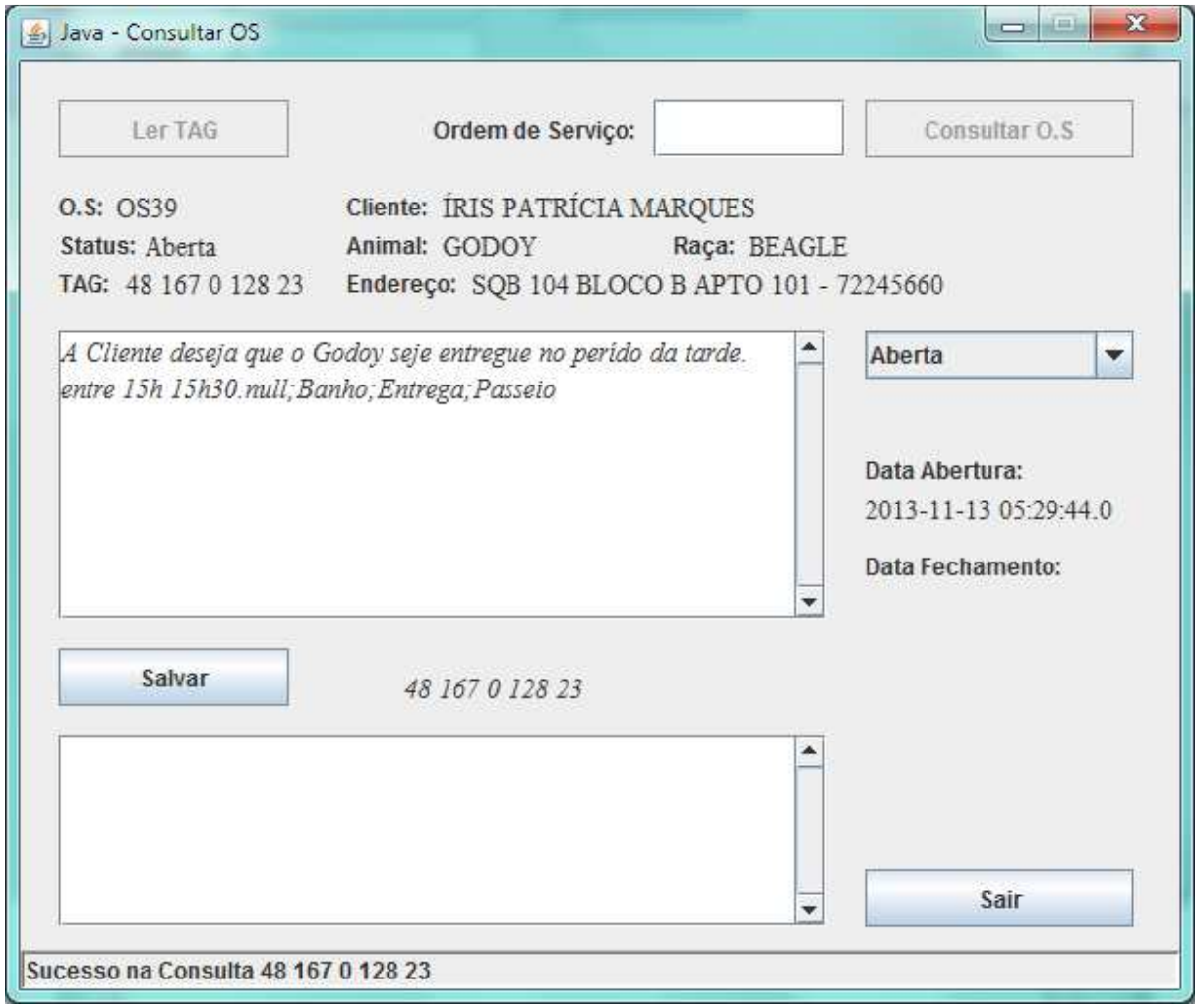


Figura 5.26 – Constar O.S por tag
Fonte: (Autor)

Na figura 5.27, ilustra a pesquisa do cliente que foi criado, qual animal está associado a ele, suas ordens de serviços e detalhes dos serviços. Esta ação pode ser feita, acessando o menu “Consultar Cliente”.

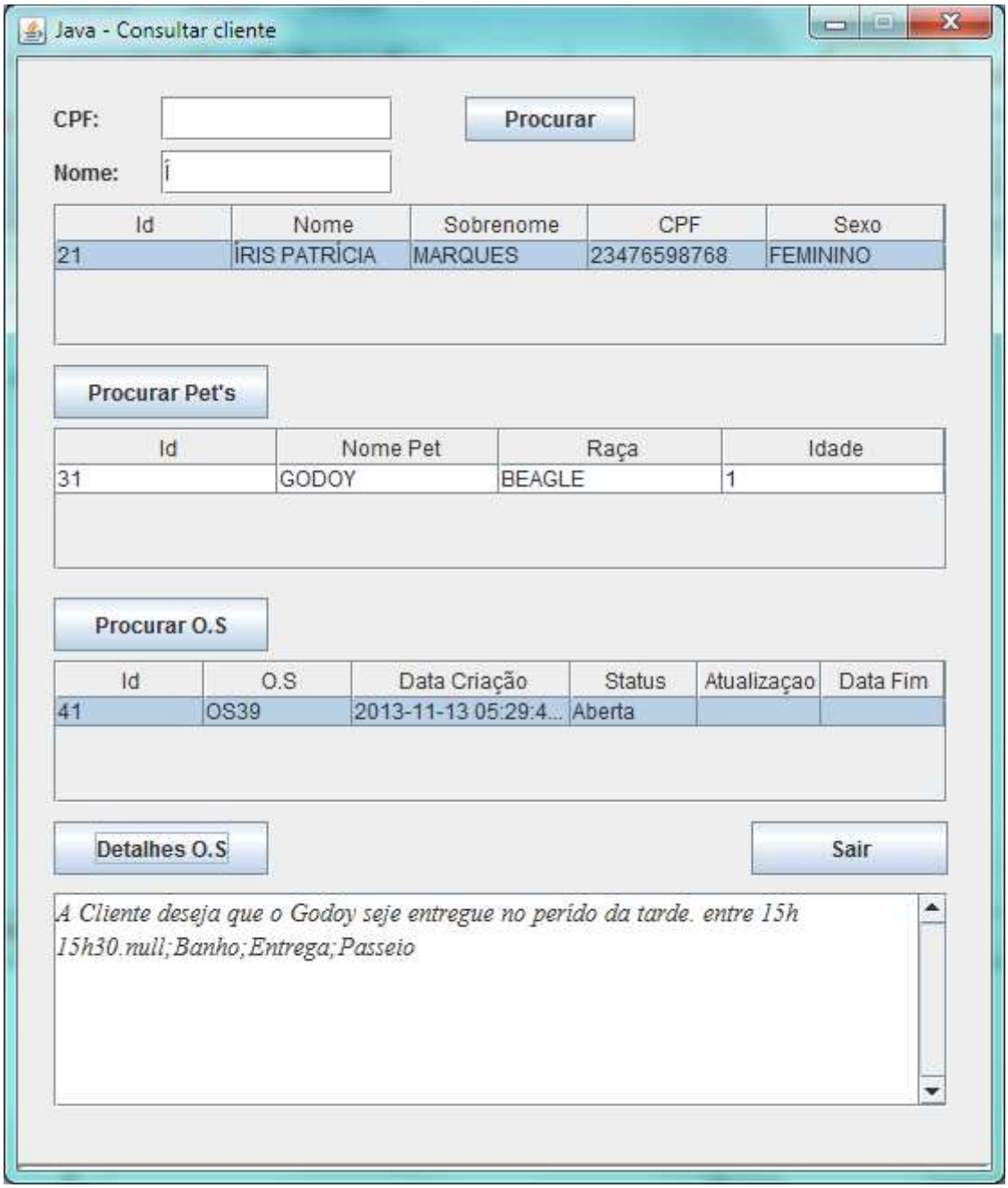


Figura 5.27 – Consultar Cliente
Fonte: (Autor)

A ordem de serviço OS39 foi tramitada na tela de “Consultar Ordens de Serviço”, e como o esperado, o status refletiu nesta tela de “Consultar Cliente”.

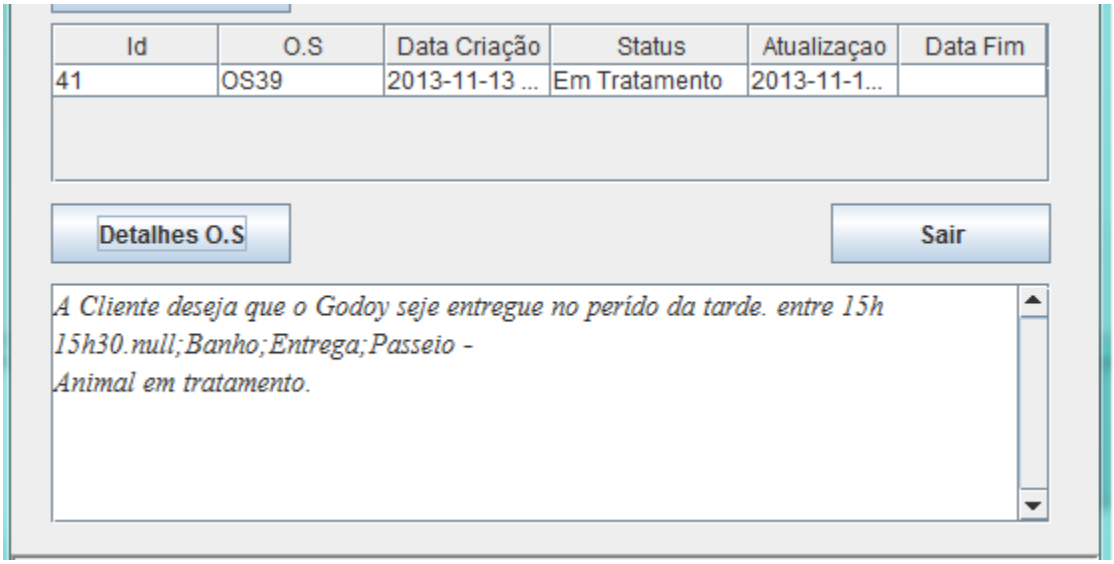


Figura 5.28 – Consultar Cliente, tramitação de O.S
Fonte: (Autor)

Após a execução do serviço, esta OS39 foi fechada, a tag desvinculada do cachorro que foi entregue ao seu dono no endereço cadastrado, dentro do prazo estabelecido quando na criação da ordem de serviço. Na tela diária podemos verificar essa OS39 fechada.

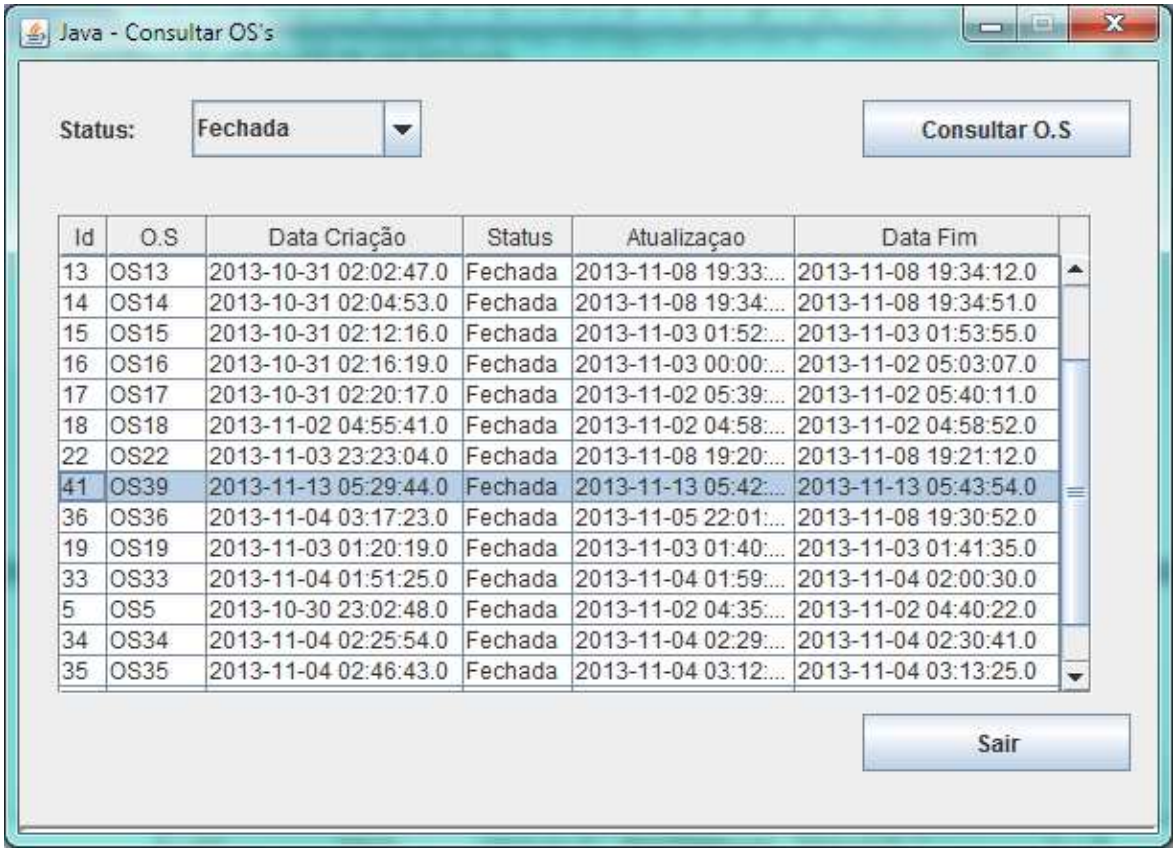


Figura 5.29 – Tela Diária
Fonte: (Autor)

5.5. DIFICULDADES ENCONTRADAS

Houve dificuldade de fazer funcionar a interface leitora – microcontrolador – dentro desta dificuldade o primeiro ponto foi encontrar a pinagem correta do leitor RFID para com o microcontrolador. Depois, outro ponto dentro desta dificuldade foi encontrar a biblioteca para realizar a leitura da tag. Dificuldade sanada, após a leitura de instruções no próprio site do fabricante do microcontrolador seguindo o padrão da biblioteca SPI.

Outra dificuldade foi ler a porta COM7 - Dificuldade de encontrar a API do java que faz leitura da porta COM7. Verificado que no site da Oracle, é fornecida a API Javacomm apenas para as plataformas Solaris SPARC, Solaris x86, and Linux x86. Como a plataforma usada neste trabalho é o Windows 64bits, recorreu-se a pesquisas e foi descoberta uma API chamada de RxTxcomm da GNU, que trabalha em plataforma windows 32bits e é baseada na API JavaComm, e para que ela funcione é necessário copiar uma .dll fornecida dentro da própria biblioteca, para a pasta do sistema windows system32. Feito isso é montado o código com base nos exemplos dos métodos fornecidos pela própria GNU, e obtendo-se a leitura da porta COM7, que até o momento era lido e mostrado na saída padrão do ambiente de desenvolvimento.

Identificada a dificuldade em capturar os dados lidos - em virtude do dado lido, ser mostrado apenas na saída padrão, outra dificuldade foi capturá-lo dentro do software. Ao acionar o botão de "*Ler tag*" um evento é disparado por uma *thread*. Uma *thread* possui um clico de vida independente de quem a iniciou e quando ela terminava, o valor lido era perdido, sendo que a classe principal também já tinha terminado ação. Pensando nisso, foi declarada uma variável do tipo *static* (que possui o mesmo valor em todo o programa), na classe que é instanciada. No programa principal foi desenvolvida uma estrutura de controle utilizando o *while*, sendo assim, enquanto a variável não fosse preenchida, o programa principal não terminaria. Após esse desenvolvimento foi possível adquirir os dados lidos para dentro do domínio do software.

5.6. RESULTADOS OBTIDOS

Foi desenvolvido neste trabalho um sistema de controle de atividades em um Pet Shop tendo como apoio a tecnologia de identificação via rádio frequência – RFID. Com os testes realizados, o projeto funcionou conforme o previsto, e desta forma pode ser usado para o devido fim de controlar as atividades em um estabelecimento de Pet Shop.

Houve sucesso na leitura da tag pelo leitor. A escolha do leitor MF522-AN para realizar a captação dos dados da tag se mostrou eficaz, utilizando as bibliotecas SPI e RFID disponíveis ao microcontrolador arduíno R3. E ainda nesta parte de leitura, afim de evitar que ao emparelhar a tag com o leitor, sejam feitas várias leituras por segundo, foi inserido um atraso programado de 0,5 segundos.

A utilização do bando de dados Oracle 10G, mostrou-se eficaz gravando os dados com integridade o que comprova que o modelo de entidade relacional foi bem projetado.

A codificação do projeto, gerou no sistema funcionalidades no qual podemos citar:

- a autenticação dos usuários por perfil;
- o cadastro dos dados dos clientes e pets no sistema;
- a consulta de clientes e seus dados;
- a inserção de mais de um animal de estimação para um mesmo cliente;
- a geração da ordem de serviço associada ao valor lido da tag, bem como as observações inerentes ao serviço;
- a consulta da ordem de serviço, realizada tanto pelo código gerado, quanto pela tag que está associada a ordem de serviço, (enquanto a ordem de serviço estiver com status diferente de fechada);
- a tramitação da ordem de serviço e suas anotações;
- a consulta de todas as ordens de serviços através do status(aberta, em tratamento ou fechada);
- o cadastramento da tag no banco de dados para que possa ser utilizada no sistema associando-se a ordem de serviço enquanto estiver com status diferente de fechada;
- a criação de usuário de acordo com o perfil.

Nos testes realizados, houveram sucesso em todas as funcionalidades geradas pelo software.

5.7. PRODUTO GERADO

O produto gerado foi um sistema capaz de gerenciar o cadastro do cliente e dos seus animais de estimação, bem como os serviços prestados.

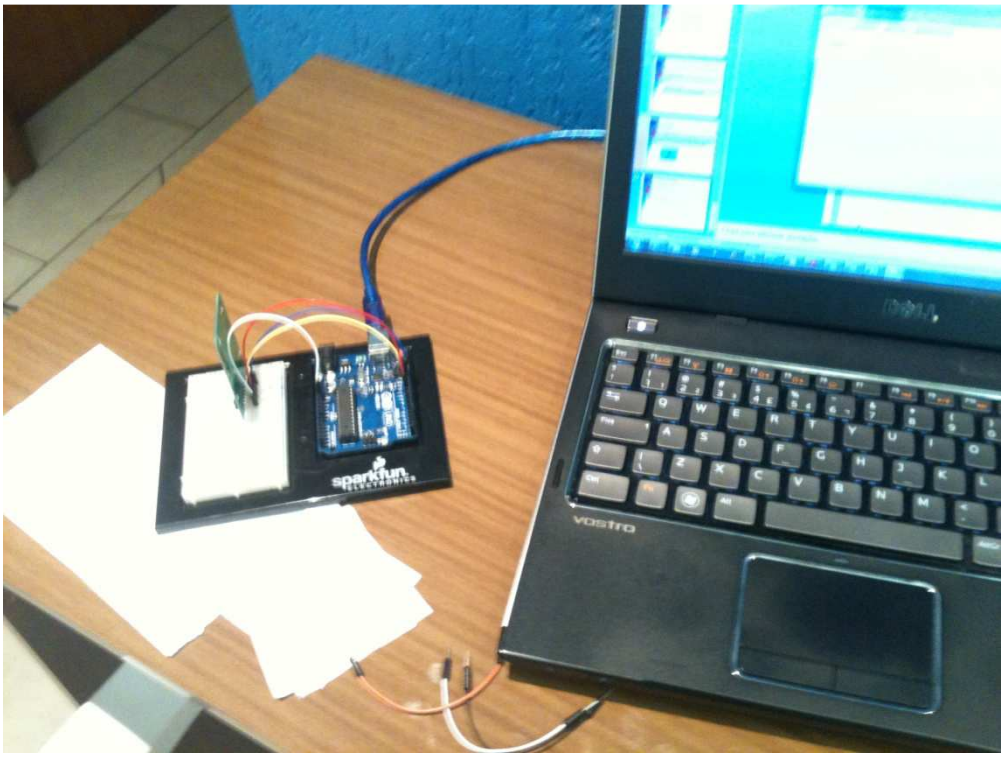


Figura 5.30 – Montagem do Arduíno, Placa leitora RFID e o computador
Fonte: (Autor)

Na figura 5.30 é mostrado a montagem física do sistema constando de um notebook estando em execução um sistema *frontend* e banco de dados, um leitor rfid conectado ao arduíno uno R3.

CAPÍTULO 6 – CONSIDERAÇÕES FINAIS

Este capítulo trata das considerações finais a respeito deste do projeto desenvolvido ao longo deste trabalho, bem como apresenta propostas para futuros trabalhos nessa área.

6.1. CONCLUSÃO

Todo o empenho e trabalho dos empresários brasileiros no ramo da indústria pet, gerou como resultado uma elevada movimentação financeira, geração de empregos, desenvolvimento especializados de serviços nesta área (logística, veterinária, comercial, alimentação). A ideia de chamar a tecnologia da informação para este ramo, juntamente com a engenharia da computação converge para uma receita de muito sucesso.

O RFID é uma realidade e está sendo utilizada em diversas áreas como logística, indústria e varejo. O objetivo deste projeto de identificar o animal de estimação dentro do Pet Shop, bem como seu dono, e os seus serviços a serem realizados com o auxílio do RFID e um sistema foi alcançado.

O microcontrolador arduíno uno R3 foi o fator decisivo na conquista desta solução, pois integrou com sucesso o leitor ao computador utilizando as bibliotecas SPI e RFID do arduíno. As interfaces foram construídas e os diferentes elementos deste projeto (JVM, ORACLE, Rxtxcomm, leitor, tag, porta serial) foram integrados na obtenção desta solução.

6.2. PROPOSTAS PARA FUTUROS PROJETOS

Para a melhor utilização desta ideia indica-se:

- Implementação de um módulo WEB para o cliente acompanhar suas ordens de serviços;
- Neste módulo WEB, implementar o envio de relatórios ao e-mail do cliente.
- Implementar o módulo wirelless de comunicação entre o microcontrolador arduíno e o computador;

REFERÊNCIAS BIBLIOGRÁFICAS

ABINPET. Disponível em < Fonte: http://abinpet.org.br/imprensa/abinpet-faz-balanco-do-mercado-pet-brasileiro/>. Acesso em: 25 Out. 2013.

AULETE. Disponível em < Fonte: http://aulete.uol.com.br/controle#ixzz2kRM40RXU”>. Acesso em: 01 Nov. 2013.

ARDUINO. Disponível em < Fonte: www.arduino.cc/>. Acesso em: 2 Out. 2013.

ARDUINO1. Disponível em < Fonte: www.arduino.cc/en/Main/ArduinoBoardUno>. Acesso em: 10 Out. 2013.

ARDUINO2, Team, Arduino Uno R3 Front, 2013. Disponível em: <http://arduino.cc/en/Main/ArduinoBoardUno> Acesso em: 30 Set. 2013.

ARDUINO3, Team, Arduino Uno R3 Front, 2013. Disponível em: <http://arduino.cc/en/Reference/SPI> Acesso em: 30 Set. 2013.

LEITOR1, MFRC522 Contactless Reader IC, 3.2Rev. 2007. Disponível em: <http://www.npx.com>

ATMEGA, Datasheet ATmega48PA/88PA/168PA/328P, 35.4Rev. San Jose, CA-USA: Atmel Corporation, 2009.

BRANCO, Mariana, Mercado de animais domésticos cresce; segmento pede estatuto federal e incentivos fiscais, 2013. Disponível em: < Fonte: agenciabrasil.ebc.com.br/noticia/2013-03-16/mercado-de-animais-domesticos-cresce-segmento-pede-estatuto-federal-e-incentivos-fiscais> Acesso em: 01 Out. 2013.

CONSUMIDOR, Disponível em <Fonte: www.reclameaqui.com.br/812490/pet-center-marginal/erro-na-vacinacao/>. Acesso em: 15 Set. 2013

CRESTANI, Pablo Augusto. A lealdade entre clientes de pet shop e agroveterinárias de santa rosa. Monografia, departamento de ciências Administrativas, Contábeis, Econômicas e da Comunicação, Universidade Regional do Noroeste do Estado do Rio Grande do Sul, Ijuí, RS, 2012 Orientado por: Zamberlan, Luciano.

DEITEL, H.M. Java: como programar, 6ª ed. São Paulo: Pearson, 2005.

GIMENEZ, S. P. Microcontroladores 8051. 1ª ed. São Paulo: Pearson, 2005.

GENNICK, Jonathan; LUERS, Tom. Aprenda em 21 dias PL/SQL. Tradução da segunda edição, Kátia Roque. Rio de Janeiro: Editora Campus, 2000.

MECENAS, Ivan; OLIVEIRA, Viviane de. Banco de Dados: Do Modelo Conceitual à Implementação Física. Rio de Janeiro: Editora Alta Books, 2005.

MOURA, Reinaldo A; RFID: 10 anos depois, IntraLogística movimentação e armazenagem de materiais, São Paulo, número 269, 8 - 10, mar. 2013

MULLER, Robert J. Projeto de banco de dados: usando UML para modelagem de dados. Tradução Bazan Tecnologia e Linguística. São Paulo: Berkeley Brasil, 2002.

PINHEIRO, José Mauricio Santo. RFID - Identificação por Rádio frequência, 2004. Disponível em: < Fonte: http://www.projetoderedes.com.br/artigos/ artigo_identificacao_por_rádio_frequencia.php >. Acesso em: 2 Mar. 2013.

SEBRAE, SEBRAE/MS; Ficha Técnica Pet Shop – Clínica Veterinária, 2013. Disponível em: < Fonte: http://www2.ms.sebrae.com.br/uploads/UAI/fichastecnicas/petshop.pdf> Aceso em: 13 Set. 2013.

TAUFENBACH, Sérgio Luiz Dalcastagne; RFID: Identficação por Radiofrequeência a Etiqueta Inteligente, Revista de divulgação técnico-científica do ICPG, Santa Catarina, volume 2, número 7, 71 - 77, dez. 2004.

TRIBUNA ANIMAL. Polícia investiga morte de cão em pet shop de Orlândia (SP). Publicado em 26 de jan 2012. Disponível em <Fonte: http://tribunaanimal.org/index.php?/Noticias/ANIMAIS-BRASIL/Policia-investiga-morte-de-cao-em-pet-shop-de-Orlandia-SP.html >. Acesso em 15 de Out. 2012.

APÊNDICE A

```
/*####################################################################*/
PROJETO FINAL - Engenharia da Computação - UniCEUB
2o. Semestre de 2013
JOAO VICTOR MARQUES DOS SANTOS
RA: 2022768/0
CONTROLE DE ATIVIDADES DE UM PET SHOP
/*####################################################################*/
//Código Utilizado pelo Arduíno

// Incluindo as bibliotecas necessarias para a leitura do cartao

#include <SPI.h>

#include <RFID.h>

// Configurando os pinos de leitura do Arduino

// Seta o valor do pino 10 e 5 como saida e seta o pina 10 para 0 volts e o pino 5 para 3,3

//volts

RFID rfid(10,5);

void setup()

{       Serial.begin(9600); //Taxa de transmissao

        SPI.begin();        //Inicializando a biblioteca SPI

        rfid.init();        //Inicializando a biblioteca RFID

}

void loop()

{  if (rfid.isCard()) {

       if (rfid.readCardSerial()) {

                Serial.print(rfid.serNum[0],DEC);

                Serial.print(" ");

                Serial.print(rfid.serNum[1],DEC);

                Serial.print(" ");

                Serial.print(rfid.serNum[2],DEC);

                Serial.print(" ");

                Serial.print(rfid.serNum[3],DEC);

                Serial.print(" ");

                Serial.print(rfid.serNum[4],DEC);

                Serial.println(" ");

        }

   }
```

```
    // implementar uma pausa apos a leitura do cartao.
    rfid.halt();    // Finaliza o usa da biblioteca RFID
    delay(500);
}
```

APÊNDICE B

```
--/*################################################################*/
--PROJETO FINAL - Engenharia da Computação - UniCEUB
--2o. Semestre de 2013
--JOAO VICTOR MARQUES DOS SANTOS
--RA: 2022768/0
--CONTROLE DE ATIVIDADES DE UM PET SHOP
--Criação das tabelas e valores iniciais
--/*################################################################*/

CREATE TABLE sa.table_usuario
(
   id           NUMBER,
   create_date   DATE,
   cod_user      VARCHAR2 (20),
   nome          VARCHAR2 (50),
   snome         VARCHAR2 (50),
   genero        VARCHAR2 (15),
   cpf          varchar2 (11),
   password      VARCHAR2 (40),
   fone          VARCHAR2 (11),
   email         VARCHAR2 (40),
   status        VARCHAR2 (10),
   usuario2priv  NUMBER
)
TABLESPACE users
STORAGE (INITIAL 1 M NEXT 2 M);

CREATE INDEX sa.idx_id
   ON sa.table_usuario (id)
   TABLESPACE users
   STORAGE (INITIAL 1 M NEXT 2 M);
```

```
CREATE SEQUENCE sa.seq_usuario
   MINVALUE 1
   MAXVALUE 99999999
   INCREMENT BY 1
   NOCYCLE
   NOORDER
   CACHE 5;

GRANT SELECT ON sa.seq_usuario TO dba;

INSERT INTO sa.table_usuario (id,
                        create_date,
                        cod_user,
                        nome,
                        snome,
                        password,
                        genero,
                        cpf,
                        fone,
                        email,
                        status,
                        usuario2priv )
VALUES (sa.seq_usuario.NEXTVAL,
      SYSDATE,
      'BR001',
      UPPER ('joao victor'),
      UPPER ('marques'),
      NULL,
      'MASCULINO',
      '72396377191',
      '6199682695',
      LOWER('JOAOVICTOR2@GMAIL.COM'), 'Ativo', NULL);
```

```
--##################################################

CREATE TABLE sa.table_privilegio
(
   id           NUMBER,
   create_date   DATE,
   nome         VARCHAR2 (50),
   status       VARCHAR2 (10)
)
TABLESPACE users
STORAGE (INITIAL 1 M NEXT 2 M);

CREATE INDEX sa.idx_id_priv
   ON sa.table_privilegio (id)
   TABLESPACE users
   STORAGE (INITIAL 1 M NEXT 2 M);

CREATE SEQUENCE sa.seq_privilegio
   MINVALUE 1
   MAXVALUE 99999999
   INCREMENT BY 1
   NOCYCLE
   NOORDER
   CACHE 5;


GRANT SELECT ON sa.seq_privilegio TO dba;

INSERT INTO sa.table_privilegio (id,
                        create_date,
                        nome,
                        status)
VALUES (sa.seq_privilegio.NEXTVAL,
      SYSDATE,
```

```
      UPPER ('Administrador'),
      'Ativo');

INSERT INTO sa.table_privilegio (id,
                        create_date,
                        nome,
                        status)
VALUES (sa.seq_privilegio.NEXTVAL,
      SYSDATE,
      UPPER ('Atendente'),
      'Ativo');

INSERT INTO sa.table_privilegio (id,
                        create_date,
                        nome,
                        status)
VALUES (sa.seq_privilegio.NEXTVAL,
      SYSDATE,
      UPPER ('Tratador'),
      'Ativo');
COMMIT;

--#########################################################

CREATE TABLE sa.table_pet
(
   id           NUMBER,
   create_date   DATE,
   nome_pet      VARCHAR2 (50),
   tipo          VARCHAR2 (50),
   idade         VARCHAR2 (5),
   porte         VARCHAR2 (10),
   sexo          VARCHAR2 (10),
   status        VARCHAR2 (10),
```

```
   pet2cliente   NUMBER
)
TABLESPACE users
STORAGE (INITIAL 1 M NEXT 2 M);

CREATE INDEX sa.idx_id_pet
   ON sa.table_pet (id)
   TABLESPACE users
   STORAGE (INITIAL 1 M NEXT 2 M);

CREATE SEQUENCE sa.seq_pet
   MINVALUE 1
   MAXVALUE 99999999
   INCREMENT BY 1
   NOCYCLE
   NOORDER
   CACHE 5;


GRANT SELECT ON sa.seq_pet TO dba;

--#######################################################

CREATE TABLE sa.table_endereco
(
   id              NUMBER,
   create_date      DATE,
   estado          VARCHAR2 (50),
   cidade          VARCHAR2 (50),
   endereco         VARCHAR2 (255),
   rua             VARCHAR2 (155),
   cep             VARCHAR2 (10),
   endereco2cliente NUMBER
)
```

```
TABLESPACE users
STORAGE (INITIAL 1 M NEXT 2 M);

CREATE INDEX sa.idx_id_end
   ON sa.table_endereco (id)
   TABLESPACE users
   STORAGE (INITIAL 1 M NEXT 2 M);

CREATE SEQUENCE sa.seq_end
   MINVALUE 1
   MAXVALUE 99999999
   INCREMENT BY 1
   NOCYCLE
   NOORDER
   CACHE 5;

GRANT SELECT ON sa.seq_end TO dba;

--#########################################################

CREATE TABLE sa.table_control_user
(
   name          VARCHAR2(20),
   formato        VARCHAR2(5),
   id_user        NUMBER

)
TABLESPACE users
STORAGE (INITIAL 1 M NEXT 2 M);

CREATE SEQUENCE sa.seq_user
   MINVALUE 1
   MAXVALUE 99999999
   INCREMENT BY 1
```

```
   NOCYCLE
   NOORDER
   CACHE 5;

GRANT SELECT ON sa.seq_user TO dba;

INSERT INTO sa.table_control_user (name, formato, id_user)
VALUES ('OSID', 'OS', 0);
COMMIT;

INSERT INTO sa.table_control_user (nome, formato, id_user)
VALUES ('ControlID', 'US', 0);
UPDATE TABLE_CONTROL_USER SET ID_USER = 0 WHERE NAME =  'ControlID';
COMMIT;

--#########################################################

CREATE TABLE sa.table_cliente
(
   id           NUMBER,
   create_date   DATE,
   cod_user      VARCHAR2 (20),
   nome          VARCHAR2 (50),
   snome         VARCHAR2 (50),
   genero        VARCHAR2 (15),
   cpf          varchar2 (11),
   fone          VARCHAR2 (11),
   email         VARCHAR2 (40),
   status        VARCHAR2 (10),
)
TABLESPACE users
STORAGE (INITIAL 1 M NEXT 2 M);

CREATE INDEX sa.idx_id_cli
```

```
   ON sa.table_cliente (id)
   TABLESPACE users
   STORAGE (INITIAL 1 M NEXT 2 M);

CREATE SEQUENCE sa.seq_cliente
   MINVALUE 1
   MAXVALUE 99999999
   INCREMENT BY 1
   NOCYCLE
   NOORDER
   CACHE 5;


GRANT SELECT ON sa.seq_cliente TO dba;

--#########################################################

CREATE TABLE sa.table_tagrfid
(
   id           NUMBER,
   create_date   DATE,
   cod_rfid      VARCHAR2 (20),
   status        VARCHAR2 (15)
)
TABLESPACE users
STORAGE (INITIAL 1 M NEXT 2 M);

CREATE INDEX sa.idx_id_tag
   ON sa.table_tagrfid (id)
   TABLESPACE users
   STORAGE (INITIAL 1 M NEXT 2 M);

CREATE SEQUENCE sa.seq_tagrfid
   MINVALUE 1
```

```
   MAXVALUE 99999999
   INCREMENT BY 1
   NOCYCLE
   NOORDER
   CACHE 5;


GRANT SELECT ON sa.seq_tagrfid TO dba;

--#########################################################

CREATE TABLE sa.table_osrfid
(
   id              NUMBER,
   create_date     DATE,
   idos_rfid       VARCHAR2 (20),
   status          VARCHAR2 (15),
   data_fechamento  DATE,
   data_atualizacao DATE,
   historico       CLOB,
   os2tratador     NUMBER,
   os2pet          NUMBER,
   os2cur_tagrfid   NUMBER,
   os2old_tagrfid   NUMBER,
   os2cliente      NUMBER
)
TABLESPACE users
STORAGE (INITIAL 1 M NEXT 2 M);

CREATE INDEX sa.idx_id_os
   ON sa.table_osrfid (id)
   TABLESPACE users
   STORAGE (INITIAL 1 M NEXT 2 M);
```

```
CREATE SEQUENCE sa.seq_osrfid
   MINVALUE 1
   MAXVALUE 99999999
   INCREMENT BY 1
   NOCYCLE
   NOORDER
   CACHE 5;

GRANT SELECT ON sa.seq_osrfid TO dba;

--#########################################################
```