

**Centro Universitário Positivo - UnicenP  
Núcleo de Ciências Exatas e Tecnológicas – NCET  
Engenharia da Computação**

**Fernando Cezar Lucktemberg**

## **Sistema de Rastreamento Animal**

**Curitiba  
2005**

**Centro Universitário Positivo - UnicenP**  
**Núcleo de Ciências Exatas e Tecnológicas – NCET**  
**Engenharia da Computação**

**Fernando Cezar Lucktemberg**

## **Sistema de Rastreamento Animal**

Monografia apresentada à disciplina de Projeto Final, como requisito parcial à conclusão do Curso de Engenharia da Computação.

Orientador: Prof. Valfredo Pilla Jr.

**Curitiba**  
**2005**

## **TERMO DE APROVAÇÃO**

Fernando Cezar Lucktemberg

Sistema de Rastreamento Animal

Monografia aprovada como requisito parcial à conclusão do curso de Engenharia da Computação do Centro Universitário Positivo, pela seguinte banca examinadora:

Prof. Valfredo Pilla Jr.

Prof. Edson Pedro Ferlin

Prof. José Carlos da Cunha

Curitiba, 01 de Janeiro de 2005

## **AGRADECIMENTOS**

A meus pais e meu irmão pelo esforço imensurável feito para poder me prover de uma boa educação, todo o apoio prestado em qualquer momento de desânimo, vocês foram sem sombra de dúvidas meus mentores e heróis em todas as conquistas.

A minha namorada por todo carinho, compreensão e ajuda no que foi possível durante a execução do projeto.

A meu orientador pela ajuda prestada.

# Sumário

Lista de figuras.....	iii
Lista de tabelas.....	iv
1. Introdução.....	1
2. Revisão da Teoria.....	3
2.1. Rfid.....	3
2.2. Xml.....	4
2.3. Hibernate.....	5
2.4. Commapi.....	6
3. Especificação Técnica.....	7
3.1. Introdução.....	7
3.2. Especificação do Hardware.....	8
3.2.1. Funções do Hardware.....	8
3.2.2. Diagrama em Blocos com Descrição.....	9
3.2.3. Ambiente de Desenvolvimento do Firmware.....	10
3.3. Especificação do Software.....	11
3.3.1. Descrição geral do sistema.....	11
3.3.2. Diagrama de blocos.....	12
3.3.3. Protótipo de interface.....	12
3.4. Validação.....	14
3.4.1. Validação do Hardware.....	14
3.4.2. Validação do Software.....	14
3.4.3. Validação Integrada.....	14
3.5. Recursos.....	15
3.6. Custos.....	15
3.7. Cronograma.....	16
4. Projeto.....	18
4.1. Hardware.....	18
4.1.1. Sinais de Interface entre Periféricos e Microcontrolador.....	18
4.1.2. Lista de Materiais do Hardware.....	19
4.1.3. Decodificação do Transponder.....	20
4.2. Software.....	21
4.2.1. Atores do Sistema.....	23
4.2.2. Diagramas de caso de uso.....	24
4.2.3. Diagramas de classes.....	29
4.2.4. Diagramas de sequência.....	29
4.2.5. Modelo Entidade-Relacionamento.....	31

4.2.6. Diagrama de estados do Firmware.....	32
4.2.7. Fluxograma do Firmware.....	33
5. Resultados.....	34
5.1. Comunicação com o Computador.....	34
5.2. Decodificação do Transponder de RFID.....	34
5.3. Distância de leitura.....	38
5.4. Taxa de Transmissão Transponder → Transceiver.....	38
5.5. Características de funcionamento.....	39
6. Conclusão.....	40
6.1. Conclusão.....	40
6.2. Discussão.....	42
7. Bibliografia.....	43
8. Anexos.....	44
8.1. Esquemáticos.....	44
8.2. Lista de Materiais.....	44
8.3. Diagrama de Classes.....	44

## Lista de figuras

Figura 1: Diagrama em blocos do Hardware.....	8
Figura 2: Diagrama em blocos do software.....	11
Figura 3: Tela Principal.....	11
Figura 4: Protótipo da Interface de cadastro de produtores.....	12
Figura 5: Protótipo da Interface de cadastro de raças.....	12
Figura 6: Protótipo da Interface de controle de pesagem.....	12
Figura 7: Cronograma do Projeto.....	16
Figura 8: Estrutura do telegrama FDX.....	20
Figura 9: Atores.....	22
Figura 10: Caso de uso Manter Produtor.....	23
Figura 11: Caso de uso Manter Propriedade.....	24
Figura 12: Caso de uso Manter Rebanhos.....	25
Figura 13: Caso de uso Manter Pesagens.....	26
Figura 14: Caso de uso Broker.....	27
Figura 15: Diagrama de sequência do Produtor.....	28
Figura 16: Diagrama de sequência de Pesagem.....	29
Figura 17: Diagrama de sequência de Login.....	29
Figura 18: Diagrama de sequência de Logoff.....	30
Figura 19: Modelo Entidade Relacionamento.....	30
Figura 20: Diagrama de Estados do Firmware.....	31
Figura 21: Fluxograma do Firmware.....	32
Figura 22: Estrutura do sinal de Rádio-Frequência (FDX-B).....	34

## **Lista de tabelas**

Tabela 3.1: Estimativa de custos.....	15
Tabela 3.2: Cronograma do Projeto.....	16
Tabela 4.1: Diferenças entre os tipos de modulação.....	19
Tabela 4.2: Estrutura do código.....	20
Tabela 5.1: Dados de temporização.....	35
Tabela 5.2: Dados decodificados da temporização (Com bit de sinalização do bloco de 8 bits).....	36
Tabela 5.3: Dados binários do transponder na ordem correta.....	36
Tabela 5.4: Dados decodificados - Linha 1 = Transponder 1, Linha 2 = Transponder 236	
Tabela 5.5: Resultado da leitura dos transponders.....	37



## RESUMO

Neste trabalho descreve-se um sistema para automatizar a identificação e catalogação de dados de bovidos. A identificação do animal é tomada de um transponder acoplado ao brinco de identificação, através do transceiver de RFID (Identificação por Rádio-Frequência). Com os dados colhidos pelo transceiver de RFID um software de gestão pecuária possibilitará aos produtores analisar os dados, assim fazendo com que o produtor tenha conhecimento da evolução do peso dos animais da propriedade, bem como a localização dos mesmos, assim possibilitando uma melhor tomada de decisões em relação à mudanças e melhorias no processo de criação tais quais efetuar ou não o processo de cria, recria e engorda, investir na compra de matrizes, e outros.

**Palavras-chave:** RFID, Auto-ID, Identificação por Rádio Frequência, Rastreabilidade, SISBOV.

## **ABSTRACT**

This project describes a system which aims to automate the identification and collection of data from cattle. Also, every animal on the property will have its own SISBOV number. The process of animal identification is done by attaching a RFID transponder to an earring. The RFID transceiver then reads the data in the transponder and sends it to a managing software. This software will allow the farmer to have a better understanding of how animal weight is evolving in the property, as well as their location at each moment of the animal's life. That way the farmer can have a better tool to research about any change that can be done to the production Process.

**Keywords:** RFID, AutoID, Radio Frequency Identification, animal tracking, SISBOV.

## 1. INTRODUÇÃO

A instrução normativa emitida pelo MAPA (Ministério da Agricultura, Pecuária e Abastecimento) [Ministério da Agricultura, Pecuária e Abastecimento, 2002] exige a identificação individual dos bovídeos de uma propriedade rural, sendo a data limite para a identificação de todo o rebanho nacional o dia 31 de Dezembro de 2007. O prazo máximo para identificação de cada animal após essa data é de até 90 dias após o nascimento.

Sabendo-se que uma das maiores dificuldades na criação de bovídeos está vinculada a todo o processo de manutenção, identificação e no gerenciamento de todo este processo foi definido o tema do Projeto Final de Graduação do Curso de Engenharia da Computação do Centro Universitário Positivo.

Definiu-se então a utilização de um sistema de Identificação por Rádio-Frequência (doravante mencionado como RFID – *Radio Frequency Identification*).

Este projeto tem por objetivo assistir as atuais propriedades rurais de produção pecuária facilitando o processo de identificação dos animais, fazendo com que estas tenham uma maior facilidade de se enquadrarem à nova legislação vigente, para tanto foi desenvolvido um sistema computacional composto por um dispositivo de RFID, sendo que este colhe os dados de pesagens, e um software de gestão pecuária que efetua a análise e o processamento destes dados. Este software faz o processamento dos dados colhidos pelo dispositivo de RFID e mostra de uma maneira mais ágil para o produtor alguns indicadores de produtividade relacionados a engorda de cada animal, origem de cada animal e outros dados que são de suma importância para o processo produtivo, facilitando assim a sua tomada de decisão quanto a mudanças nos processos executados na propriedade



## 2. REVISÃO DA TEORIA

### 2.1. RFID

Através da necessidade de várias companhias de haver um processo mais confiável e preciso do que a utilização de código de barras para identificação de seus produtos, nasceu a idéia do uso de um dispositivo aonde fosse possível armazenar os dados a serem coletados diretamente em um chip de silício. Este novo meio permitiu uma transferência de dados aonde não é necessário o contato físico entre o dispositivo a ser lido e o seu interrogador, fazendo assim com que houvesse uma maior agilidade deste processo, pois a energia para o dispositivo a ser lido passa a ser adquirida diretamente das ondas eletromagnéticas transmitidas pelo interrogador de RFID [FINKENZELLER, K., 2003].

RFID aplica os mesmos princípios da física que aqueles usados na transmissão de rádio, onde as ondas eletromagnéticas, transmitem e recebem vários tipos de dados. Para compreender melhor, visualize uma voz sendo transmitida ou uma música da estação de rádio que é emitida através de um transmissor. Estes são dados codificados nas ondas de rádio de uma frequência específica. Sabemos que a distância que uma transmissão de rádio pode cobrir é determinada por condições ambientais e pelo tamanho e poder das antenas em cada ponto de comunicação. Usando estes conhecimentos para a RFID, o transmissor da estação de rádio funciona como um Transponder, conhecido também como Tag, e o rádio como o leitor, ou interrogador. As antenas determinam o alcance. Estes três componentes (antena, tag e transceiver) são os blocos principais de um sistema de RFID [BHUPTANI, 2005].

RFID é um termo genérico utilizado para descrever um sistema que envia a identidade de um objeto ou pessoa através de ondas de rádio. Por isso o RFID é agrupado na categoria de tecnologias de identificação automática.

As tecnologias de Auto-ID (Identificação Automática) incluem, código de barras, leitores óticos de caracteres e algumas tecnologias de biometria. Estas tecnologias são utilizadas para reduzir o tempo necessário para efetuar a entrada de dados manual, melhorar a precisão dos dados e reduzir os erros gerados pela entrada manual.

Das tecnologias citadas, usualmente é necessário a interação do usuário com o

sistema para que o dado seja coletado. O RFID foi projetado para que a partir de leitores sem fio fosse possível captar as informações, não sendo necessária a interação do usuário.

Um sistema de RFID é composto por um Transponder (armazena os dados), um transceiver (lê e escreve os dados no tag) e uma antena. Um Transponder padrão é composto por um microchip ligado a uma antena montada em um substrato. Os chips atuais podem carregar até 2 kilobytes de dados. Para ler as informações deste TAG, é necessário o uso de um transceiver, sendo este tipicamente composto por uma antena que transmite e recebe os sinais de rádio do/para o Transponder, então essas informações são decodificadas e enviadas para um dispositivo capaz de processá-las

Os Transponders podem ser tanto passivos (sem bateria) quanto ativos (com bateria). Referente aos Transponders passivos, estes não contêm fonte de energia própria nem transmissor, por isso são mais baratos que Transponders ativos, além de terem custo de manutenção reduzidos. Por não possuírem uma fonte própria de energia a distância de leitura é geralmente muito pequena (de 3 a 30cm).

Como citado anteriormente, um Transponder de RFID consiste em um microchip ligado a uma antena. Este Transponder pode ser encapsulado de diferentes maneiras. Como exemplo podemos citar etiquetas, cartões, chaveiros, entre outros, devendo apenas notar que diferentes encapsulamentos tem diferentes custos.

Os tags passivos funcionam em diferentes frequências, sendo que as mais comuns são 125 kHz e 135 kHz para baixa frequência, 13.56 MHz para alta frequência e de 860 MHz a 960 MHz para UHF [RFIDJOURNAL].

## **2.2. XML**

Abreviação de *Extensible Markup Language* (Linguagem de marcação extensível), que descreve uma classe de objetos chamados Documentos XML, e parcialmente descreve como um programa de computador deve processar estes documentos. Os documentos XML são constituídos de unidades de armazenamento chamadas entidades, que contêm dados previamente processados ou não. Estes dados são compostos de caracteres e de uma forma de marcação. Estas marcações descrevem o layout de armazenamento de dados e a estrutura do documento.

[SPERBERG-MCQUEEN, 2004].

### 2.3. HIBERNATE

O Hibernate é uma ferramenta de mapeamento objeto/relacional para Java. Ela transforma os dados tabulares de um banco de dados em um grafo de objetos definido pelo desenvolvedor. Usando o Hibernate, o desenvolvedor se livra de escrever muito do código de acesso a banco de dados e de SQL que ele escreveria não usando a ferramenta, acelerando a velocidade do seu desenvolvimento de uma forma fantástica [LINHARES, 2005].

Hibernate é um mecanismo bem simples e poderoso que permite a persistência de objetos em banco de dados relacionais de maneira transparente e para qualquer tipo de aplicação Java (seja ela web-based ou desktop-based). Ou seja, ao invés de perder tempo escrevendo SQL, misturando estas consultas no meio do seu código Java e ficar mapeando o resultado de suas consultas para objetos, com o Hibernate você vai precisar se preocupar somente com seus objetos.

Além disso, o processo de desenvolvimento usando Hibernate é muito mais simples do que usar JDBC (*Java Database Connector*) puro. Com Hibernate, você só tem estes 5 passos:

- criar a tabela no seu banco de dados onde os objetos vão persistir;
- criar o objeto cujo estado vai ser persistido;
- criar um XML, que relaciona as propriedades do objeto aos campos na tabela
- criar um arquivo contendo as propriedades para que o Hibernate se conecte ao banco de dados [OLIVEIRA, 2005]

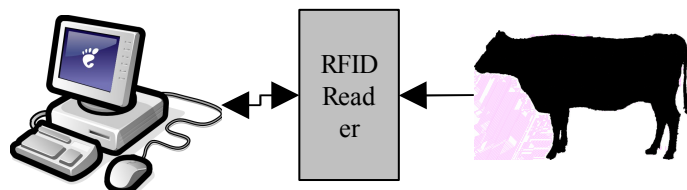
## 2.4. COMMAPI

A COMMAPI, também conhecida como Java Communications API é uma extensão à JDK (*Java Development Kit*) e à JRE (*Java Runtime Environment*) que possibilita a comunicação de aplicações desenvolvidas em JAVA com uma porta paralela ou uma porta serial. A API habilita as aplicações desenvolvidas a comunicarem essencialmente com qualquer dispositivo conectado a um dos dois tipos de porta mencionados anteriormente, como um scanner, um modem, uma unidade de backup externa, entre outros. A COMMAPI opera em nível baixo, ou seja, somente envia e recebe bits através destas portas, e não compreende as informações que estão sendo enviadas através destas portas. Portanto para que possa se fazer algo útil usando a COMMAPI deve se não somente entender como a mesma funciona, mas também ter pleno conhecimento do protocolo utilizado pelo dispositivo que se deseja acoplar ao sistema. [HAROLD, 1999]



### 3. ESPECIFICAÇÃO TÉCNICA

#### 3.1. INTRODUÇÃO



O sistema reconhece o animal através de um link de RFID criado entre o Transponder (dispositivo que é lido pelo interrogador de RFID) e o Interrogador (base de leitura de RFID). Desta forma, quando um Transponder entrar no campo de leitura de um interrogador o animal é automaticamente identificado, fazendo com que não seja necessário estressar o animal utilizando-se de bastões de choque ou outros meios para que este possa ser corretamente identificado. Esta interação possibilita ao utilizador verificar a necessidade ou não de efetuar alguma intervenção junto ao animal.

Um Transponder é anexado ao brinco utilizado pelo animal (brinco de uso obrigatório segundo as novas normas de rastreabilidade). A posição de utilização do Transponder é estratégica, pois no caso da não utilização de um brinco, o transponder deve ser implantado no animal, assim sendo haveria uma maior dificuldade na leitura devido a posição do transponder (normalmente implantado no Bólume Ruminal).

Após finalizada a avaliação do animal em questão, os dados são enviados para o computador e gravados no banco de dados do software de gestão.

Este software possibilita ao utilizador do sistema uma fácil identificação dos animais da propriedade, trazendo vários dados de importância para o negócio efetuado, como dados históricos relativos a pesagem e outros dados de utilidade que possibilitarão efetuar uma verificação de produtividade, dados históricos relativos a manejo, entre outros.

O sistema é dividido em duas partes, hardware e software. Dentre estas partes, o hardware será dividido basicamente em três módulos, sendo estes a aquisição do sinal enviado pelo Transponder que é efetuada por um circuito integrado específico para esta tarefa. Após a aquisição os dados serão enviados para o computador com o software de gestão através do módulo de transmissão Serial do Interrogador, e o

mesmo dado enviado para o computador pode também ser visualizado através do módulo de display gráfico contido no hardware. Todas estas partes são gerenciados por um microcontrolador.

A segunda parte é composta por um software para efetuar a gestão dos dados adquiridos pelo Interrogador. Este software foi dividido em dois módulos distintos. Um middleware para carregar os dados aqisitados pelo Hardware no banco de dados, e o módulo para gerenciamento dos dados da propriedade.

### **3.2. ESPECIFICAÇÃO DO HARDWARE**

#### **3.2.1. FUNÇÕES DO HARDWARE**

O Hardware é composto por um módulo que fará a aquisição do sinal de rádio frequência que será enviado pelo transponder quando este estiver na área de leitura do interrogador, estes dados serão então processados e enviados para o computador através do módulo de comunicação serial. Os dados colhidos pela unidade de aquisição do sinal poderão também ser visualizados pelo módulo de display gráfico.

### 3.2.2. DIAGRAMA EM BLOCOS COM DESCRIÇÃO

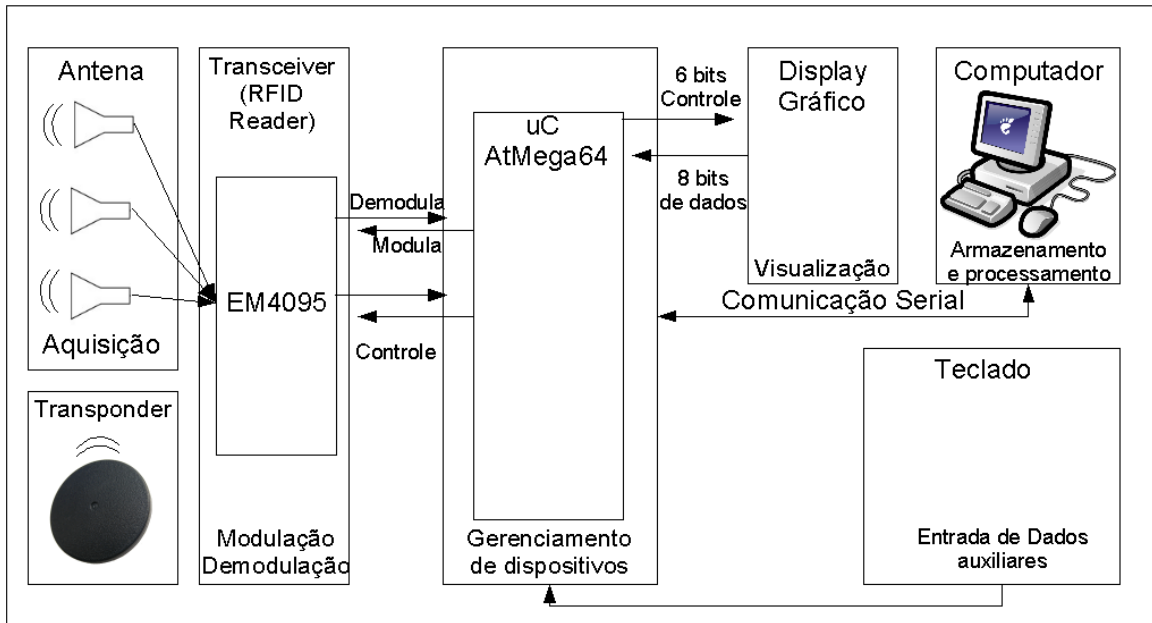


Figura 1: Diagrama em blocos do Hardware

- Gerenciamento de dispositivos: O gerenciamento dos dispositivos é efetuado por um microcontrolador. Neste projeto o microcontrolador é utilizado para gerenciar os dispositivos do sistema. O microcontrolador faz a aquisição de um sinal enviado pelo circuito integrado de RFID e então demodula esta informação. Após efetuado este passo os dados são processados e mostrados na interface de visualização, para que o utilizador possa decidir a ação a ser tomada. Posteriormente estes dados são enviados para o computador por meio de um barramento Serial.
- Computador: Faz o armazenamento e processamento dos dados com o software que é instalado nele. Para que possa se comunicar com o interrogador de RFID necessita ter pelo menos uma interface Serial disponível. Um banco de dados e uma máquina virtual Java 1.5 devem estar instalados neste computador.
- Transponder: É um elemento no qual o dado é realmente armazenado, é do Transponder que o interrogador vai obter os dados necessários. Para o projeto foram utilizados Transponders passivos, que não tem alimentação própria, e

somente passam a emitir sinais quando estão dentro da área de leitura de um interrogador. Usualmente transponders passivos são constituídos de um capacitor, uma antena e um chip, aonde os dados ficam gravados.

- Antena: A aquisição dos sinais a serem modulados / demodulados pelo circuito integrado responsável pela modulação e demodulação é efetuada através de uma antena da Texas, modelo RI-ANT-S02C. Através do uso de princípios físicos, podemos dizer que todos os circuitos nos quais tensão e/ou corrente são percebidos existe a formação de uma radiação de ondas eletromagnéticas, sendo que estes tendem a ser parasitas. A antena é um dispositivo otimizado para que possa captar estas radiações eletromagnéticas, assim como emití-las. Como uma antena pode irradiar ondas eletromagnéticas, este componente será de suma importância para o funcionamento do sistema, já que a mesma é responsável pela alimentação dos transponders.
- Transceiver: A tarefa de modular e demodular os sinais emitidos pelo Transponder é executada por um circuito integrado específico. Este integrado é um transceiver para utilização em uma estação base de RFID. O transceiver decodifica sinais na frequência de 134.2 Khz. Este transceiver deve efetuar as seguintes operações:
  - Alimentação da antena com a frequência base a ser utilizada (134.2 Khz)
  - Modulação AM do campo eletromagnético a ser transmitido pela antena para o Transponder.
  - Demodulação AM do campo eletromagnético recebido pela antena de um Transponder.
- Display Gráfico: Para que os dados adquiridos possam ser visualizados no momento em que o dispositivo está sendo utilizado, foi adicionado ao sistema o display gráfico que possibilita ao usuário uma fácil verificação dos dados lidos pelo interrogador RFID.

### **3.2.3. AMBIENTE DE DESENVOLVIMENTO DO FIRMWARE**

O ambiente para o desenvolvimento de código do microcontrolador, em linguagem C, é o software ICCAVR.

O ICCAVR foi escolhido devido a grande gama de microcontroladores da família ATMEL com as quais é possível trabalhar com este compilador, bem como as otimizações de código em relação a espaço de memória flash que o mesmo é capaz de empenhar.

Para o desenho do esquemático do circuito foi utilizado o software Orcad Capture e para o desenvolvimento do circuito impresso o OrCad Layout.

### **3.3. ESPECIFICAÇÃO DO SOFTWARE**

#### **3.3.1. DESCRIÇÃO GERAL DO SISTEMA**

Sistema de gestão pecuária responsável pelo armazenamento, processamento e demonstração dos dados adquiridos pelo sistema de RFID.

Os dados adquiridos pelo hardware do sistema são descarregados através da interface Serial com o computador que contém o middleware, que automaticamente fará o upload destas informações para o banco de dados.

A filosofia adotada em relação aos softwares de apoio para o desenvolvimento do programa, bem como protocolos e servidores, é o uso de padrões de mercados.

Como linguagem para o desenvolvimento do software do Projeto foi escolhida a linguagem Java, utilizando a plataforma de desenvolvimento Eclipse em conjunto com os plugins para uso do hibernate e de desenvolvimento de interface gráfica.

O banco de dados escolhido para o projeto foi o PostgreSQL, pois além de ter uma boa escalabilidade e suporte é compatível com a linguagem SQL, tem uma ativa comunidade de desenvolvedores e está em conformidade com os padrões ACID (Atomicity Consistency Isolation Durability) . A importância da conformidade com o Padrão ACID vem na confiabilidade dos dados, já que com o princípio da atomicidade, consistência, isolamento e durabilidade é possível garantir que ou a transação será completamente executada no banco, ou não será executada, assim não gerando entradas inválidas no banco de dados se a operação não for completamente terminada devido a algum tipo de falha na comunicação.

### 3.3.2. DIAGRAMA DE BLOCOS

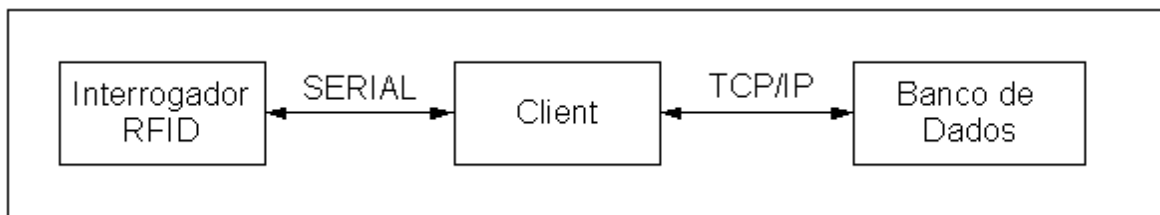


Figura 2: Diagrama em blocos do software

### 3.3.3. PROTÓTIPO DE INTERFACE

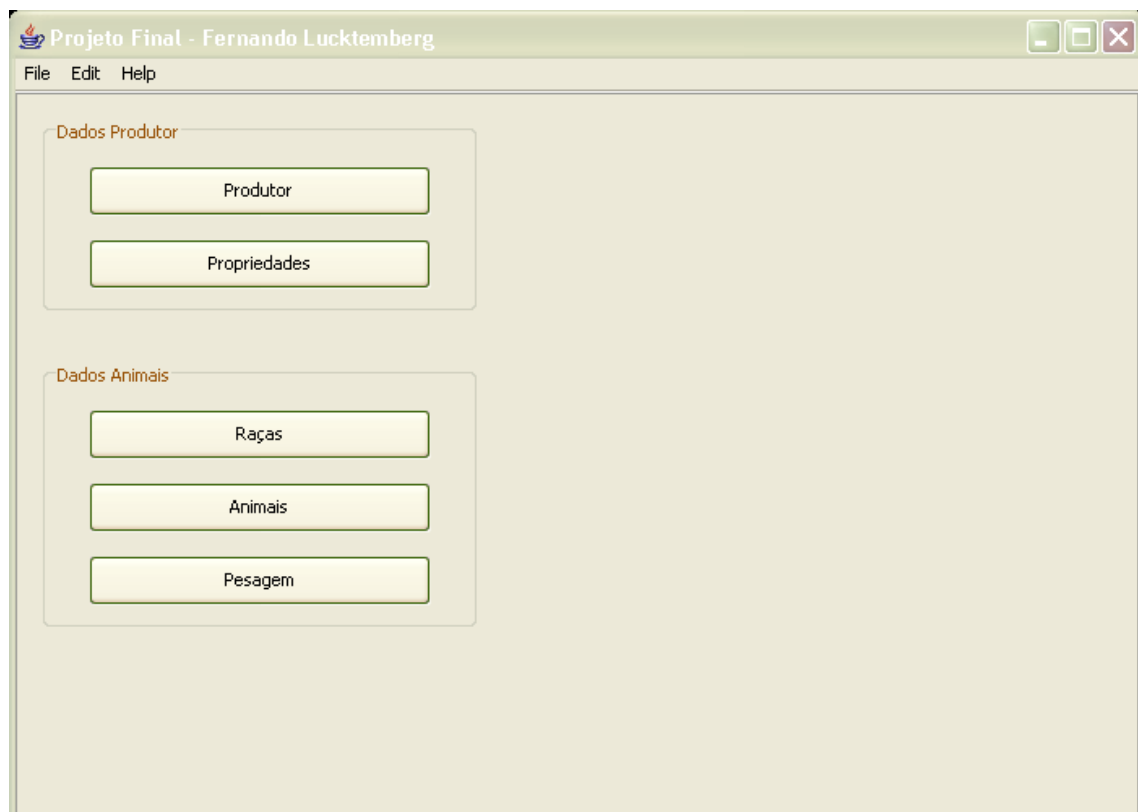


Figura 3: Tela Principal

A	B	C	D	E
0x0	0x1	0x2	0x3	0x4
1x0	1x1	1x2	1x3	1x4
2x0	2x1	2x2	2x3	2x4
3x0	3x1	3x2	3x3	3x4
4x0	4x1	4x2	4x3	4x4

**Adicionar**

Nome da Raça

**Remover**

Figura 4: Protótipo da Interface de cadastro de produtores

**Adicionar Pesagem**

Animal

Peso Aferido

**Remover Pesagem**

**Seleção de parâmetros**

Animal

0x0	0x1	0x2	0x3	0x4
1x0	1x1	1x2	1x3	1x4
2x0	2x1	2x2	2x3	2x4
3x0	3x1	3x2	3x3	3x4
4x0	4x1	4x2	4x3	4x4

Figura 6: Protótipo da Interface de controle de pesagem

### **3.4. VALIDAÇÃO**

#### **3.4.1. VALIDAÇÃO DO HARDWARE**

Os testes dos dispositivos de hardware foram executados da seguinte maneira. Primeiro foi checado o funcionamento do display gráfico, depois foi checado o funcionamento do teclado, após checado o funcionamento do teclado verifica-se o funcionamento da antena e da modulação / demodulação através do uso de um transponder de código conhecido, e finalmente executada-se um teste para aferir o funcionamento da interface Serial.

#### **3.4.2. VALIDAÇÃO DO SOFTWARE**

A validação inicial do software foi efetuada em laboratório. Será executada através de um sistema de Unit Testing, sendo este o mesmo que vem incluso no ambiente de desenvolvimento Eclipse.

Fazendo teste das Units o tempo de teste diminui drasticamente, pois são impostos os parâmetros de entrada e os parâmetros de saída, a unit então é extensivamente testada para verificar a existência de alguma exceção não tratada.

Após o teste individual de cada unit criada, o sistema sofreu então uma validação com a total integração destas units, para então ser executado o teste integrado com o hardware.

#### **3.4.3. VALIDAÇÃO INTEGRADA**

A validação integrada prossegue da seguinte maneira:

- Um Transponder é lido pelo interrogador
- O dado colhido pelo interrogador é descarregado no computador
- Caso não seja cadastrado ainda uma mensagem de aviso será mostrada no middleware



- Os dados colhidos pelo interrogador são gravados no banco de dados
- Uma nova leitura do transponder
- Os dados são novamente descarregados para o sistema
- Uma verificação da consistência dos dados é executada

### **3.5. RECURSOS**

Para o desenvolvimento do sistema serão necessários o uso de alguns recursos, sendo eles:

- Osciloscópio
- Fonte de tensão
- Computadores providos dos seguintes softwares e frameworks:
- PonyProg para gravar o firmware no microcontrolador
- ICCAVR para desenvolver o firmware
- Eclipse
- Eclipse Visual Editor
- COMM API
- Computadores providos dos seguintes dispositivos de hardware:
- Porta serial (RS-232) para gravação do microcontrolador e posterior descarga de informações do sistema.

### **3.6. CUSTOS**

Para o cálculo de custos foi levado em conta o valor dos componentes utilizados e a quantidade de horas de trabalho. Nesta estimativa não está incluso o custo de equipamentos necessários para o desenvolvimento do projeto, bem como o custo dos softwares que foram utilizados no desenvolvimento do sistema. Será desprezado inicialmente o custo de resistores, capacitores e conectores que sejam de utilização necessária.

*Tabela 3.1: Estimativa de custos*

Componente	Quantidade	Custo unitário	Custo
AtMega64	1	R\$ 60,00	R\$ 60,00
Display Gráfico	1	R\$ 90,00	R\$ 90,00
MAX232	1	R\$ 3,00	R\$ 3,00
EM4095	1	R\$ 10,00	R\$ 10,00
Regulador de tensão LM317	1	R\$ 1,00	R\$ 1,00
Botões para teclado	14	R\$ 0,25	R\$ 3,50
Horas de trabalho	550	R\$ 8,25	R\$ 4.537,50
Total			R\$ 4.705,00

### 3.7. CRONOGRAMA

Ver a tabela 2.

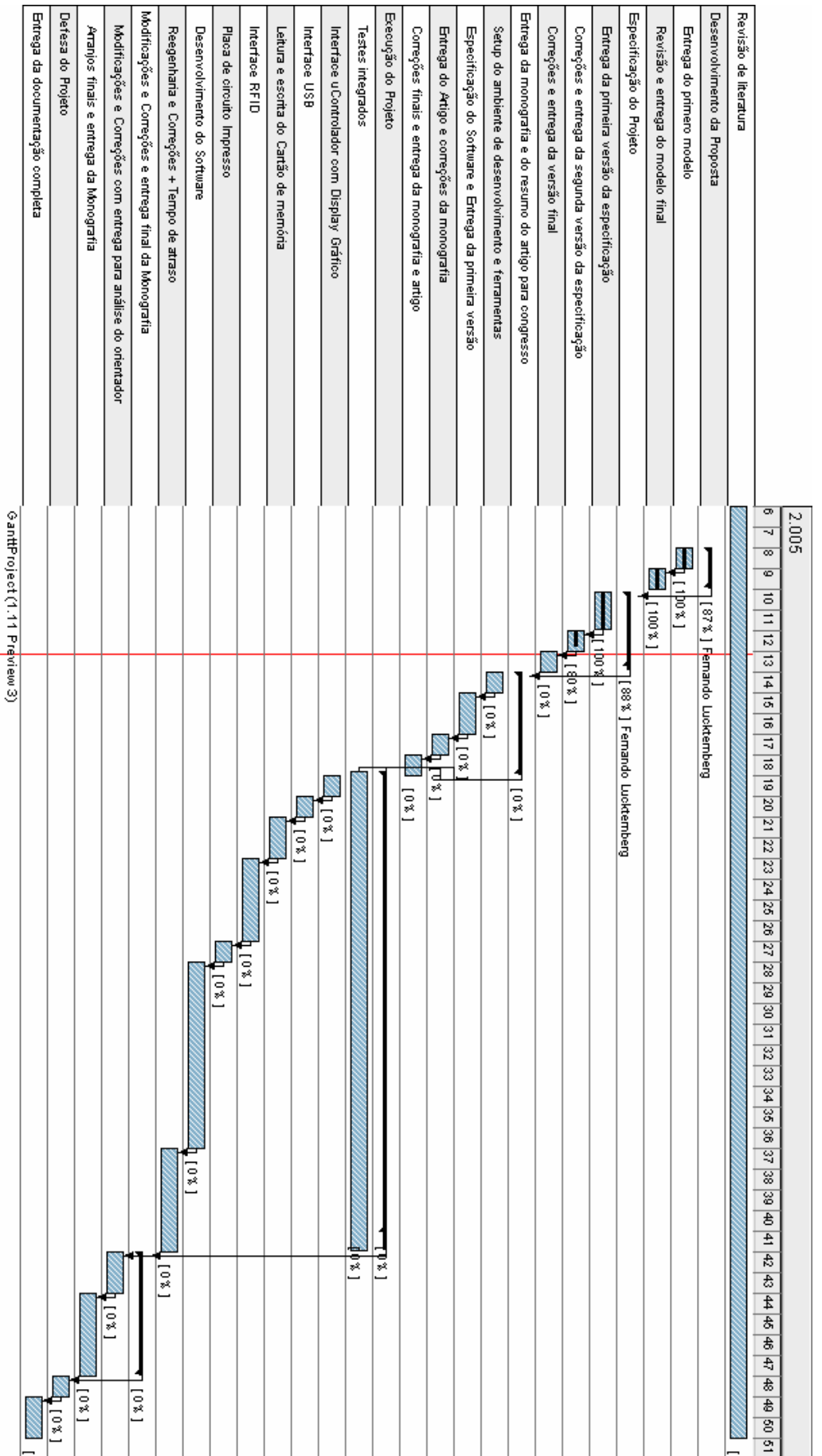


Tabela 3.2: Cronograma do Projeto

## 4. PROJETO

### 4.1. HARDWARE

#### 4.1.1. SINAIS DE INTERFACE ENTRE PERIFÉRICOS E MICROCONTROLADOR

- EM4095 (RFID)
  - SHD
    - Quando a entrada está em nível lógico alto o integrado entra em modo de suspensão. Quando o dispositivo é ligado, o pino SHD deve estar em nível lógico alto para que a inicialização do dispositivo ocorra corretamente. Ao ser aplicado um nível lógico baixo ao pino SHD o integrado passa a emitir um campo de radiofrequência, que demodula sinais que chegam à antena.
  - DEMOD\_OUT
    - O sinal digital demodulado do sinal proveniente da antena é enviado ao microcontrolador através deste pino para posterior processamento e decodificação
  - MOD
    - Nível lógico alto neste pino força tri-state na antena. Enquanto o pino estiver em nível lógico alto o VCO (Voltage Controlled Oscillator) e a demodulação AM são mantidos no estado em que se encontravam quando o pino MOD foi levado ao nível lógico alto.
  - RDY/CLK
    - Provê o sinal de clock para a sincronização da recepção entre o microcontrolador e o integrado.
  - ANT1 / ANT2:
    - Pinos de entrada da antena.
  - FCAP:
    - Pino aonde deve ser ligado o capacitor que será utilizado como filtro pelo integrado.
- TUSB (Transceiver USB)
  - SOUT
    - Saída de dados do Transceiver

- SIN
  - Entrada de dados do Transceiver
- GLCD
  - D0...D7
    - Barramento de dados
    - D/I
    - Nível lógico alto: Entrada de dados
    - Nível lógico baixo: Entrada do código de instrução
  - R/W
    - Nível lógico alto: Lê dados do display para o microcontrolador
    - Nível lógico Baixo: Grava dados do microcontrolador para o display
  - E
    - Sinal de ativação, pode ser configurado para funcionar em nível lógico alto ou borda de descida
  - CS1
    - Chip Select para a primeira metade do glcd (64x32)
  - CS2
    - Chip Select para a segunda metade do glcd (64x32)

#### **4.1.2. LISTA DE MATERIAIS DO HARDWARE**

- Microcontrolador ATMEGA64L 8AI
- Display de cristal líquido gráfico EW13B10GLY
- Controlador Serial MAX232
- Modulador e Demodulador de sinais de RFID EM4095
- Antena TI RI-ANT-S02C
- Teclado de Membrana Comum
- Capacitores
- Resistores
- Reguladores de tensão

### 4.1.3. DECODIFICAÇÃO DO TRANSPONDER

Sabendo que os transponders utilizados para a execução do projeto são do tipo ISO 11784/5, foi necessário efetuar a compra desta norma para entender o funcionamento do transponder. Como para a norma ISO11784/5 existem dois tipos de transponders, FDX-B (Full Duplex) e HDX (Half duplex), e os transponders disponíveis eram somente do tipo FDX-B, apenas este tipo de codificação foi executado no projeto.

As principais diferenças entre os dois tipos de modulação de transponders são as seguintes (Tabela 3):

Tabela 4.1: Diferenças entre os tipos de modulação

Parâmetro	Sistema FDX-B	Sistema HDX
Frequência de ativação	134,2kHz	134,2kHz
Modulação	AM-PSK	FSK
Codificação	DPB Modificada	NRZ

A estrutura do código enviado na modulação FDX-B é a seguinte (Figura 22):

- um header de 11 bits (000000001) usado para identificar o começo do telegrama de identificação
- um código de identificação de 64 bits transmitido em 8 blocos de 8 bits
- dois blocos de 8 bits contendo os 16 bits de detecção de erros de CRC (CRC-CCITT reverso)
- três blocos de 8 bits contendo os bits de *trailer*.
- Notar que para todos os blocos exceto o bloco de header é enviado um bit de nível lógico alto (1) após os 8 bits para que não ocorra um outro header no meio do código.



Figura 8: Estrutura do telegrama FDX

Podemos notar na estrutura do diagrama que os dados são do código de identificação são enviados partindo do bit menos significativo para o bit mais significativo.

Quanto à estrutura do código proveniente do transponder ver tabela 4.

Tabela 4.2: Estrutura do código

Bit	Informação	Descrição
1	Flag de aplicação animal (1) ou não animal (0)	Sinaliza se é ou não utilizado para aplicações animais.
2 – 15	Campo reservado	Reservados para uso futuro.
16	Existência de bloco de dados (1) ou não (0)	Sinaliza que dados adicionais devem ser recebidos.
17 – 26	Código numérico do país conforme ISO3166	Reservado ao código de país.
27 – 64	Código de identificação do animal	Número único dentro de um país.

## 4.2. SOFTWARE

O objetivo do software do projeto de Rastreamento é fazer um cadastro com as pesagens de cada animal, cadastro de produtores, propriedades e cadastro de raças de cada animal.

O que faz:

- Gerenciamento das divisões dos rebanhos das propriedades
- Facilidade na localização do animal
- Controle do sistema de manejo
- Histórico do animal na propriedade

- Controle da produtividade de cada animal
  - Peso
  - Taxa de engorda
- Controle de acessos através de ACL's

O que não faz:

- Gerenciamento de custos da propriedade

Requisitos do sistema

- Visão geral
  - O propósito deste software é o de controlar a vida de cada animal dentro de uma propriedade rural desde a sua entrada na mesma até a saída. Para manter este controle será utilizada a codificação do animal junto ao SISBOV (Sistema de cadastro de Bovídeos Nacional), pois esta é única e obrigatória.
- Objetivos
  - Facilitar o gerenciamento de propriedades rurais de qualquer porte (pequenas, médias e grandes)
  - Verificar a produtividade de rebanhos
  - Acompanhamento da produtividade dos animais individualmente
  - Redução de Roubo no rebanho
- Pré-suposições
  - A propriedade deve estar em conformidade com o SISBOV
  - Deve existir uma balança na propriedade para o acompanhamento do peso



#### 4.2.1. ATORES DO SISTEMA

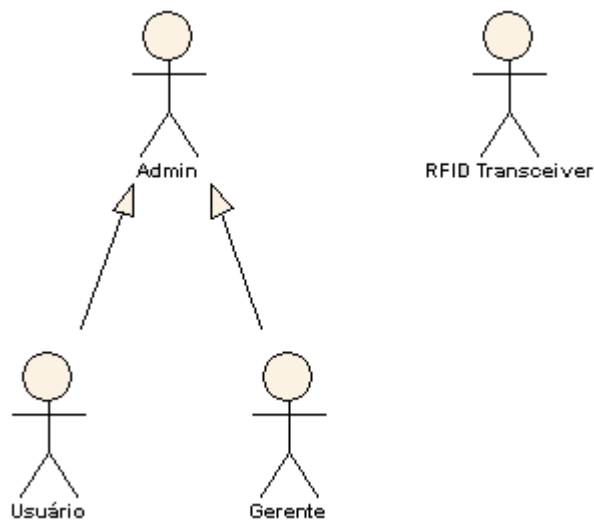


Figura 9: Atores

Os atores são os responsáveis pela entrada de dados junto ao software. Dentre os atores estão listados o administrador do software, o gerente da propriedade bem como o usuário final do software.

Um ator que é independente de todos estes outros citados é o Transceiver de Rádio Frequência, sendo que para o transceiver não existem restrições, somente como já citado anteriormente, para o caso do animal ainda não estar cadastrado no sistema o software exibirá uma mensagem de erro no console do middleware.

#### 4.2.2. DIAGRAMAS DE CASO DE USO

Caso de uso	Manter Proprietário
Atores	Gerente e Admin
Propósito	Cadastrar, remover, editar e visualizar os proprietários da fazenda e os proprietários que fazem negócios com a mesma
Descrição	O Gerente ou o administrador vão ao sistema fazem a manutenção relativa a um proprietário. É feita uma verificação quanto à permissão do usuário de fazer tal mudança, caso seja autorizado os dados são então persistidos.
Tipo	Primário e essencial
Típica sequência de eventos	

Ação do ator	Resposta do sistema
1. Solicita a criação / remoção / edição / visualização de um proprietário.	
	2. Sistema verifica se usuário tem permissão para efetuar a atividade.
	3. Sistema persiste os dados

#### Seqüência alternativa de eventos

- 2 – Usuário não tem permissão para efetuar atividade. Cria log de exceção.
- 3 – Erro ao gravar os dados. Cria log de erro e avisa usuário

#### Diagrama do Caso de uso

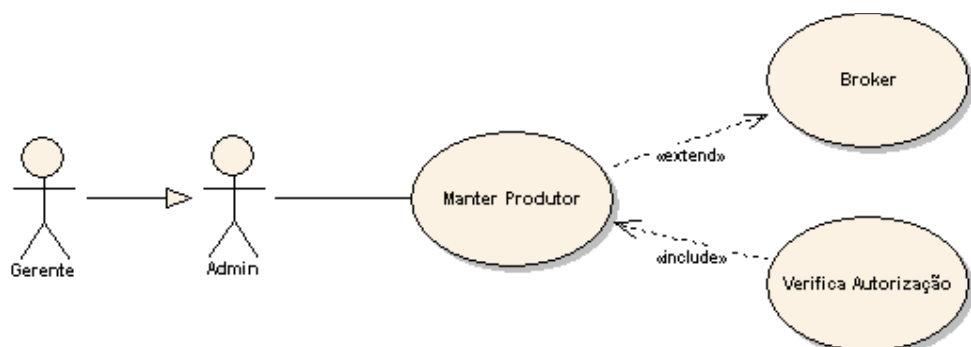


Figura 10: Caso de uso Manter Produtor

Caso de uso Manter Propriedade  
 Atores Gerente e Admin  
 Propósito Cadastrar, remover, editar e visualizar as propriedades  
 Descrição O Gerente ou o administrador vão ao sistema fazem a manutenção relativa a uma propriedade. É feita uma verificação quanto à permissão do usuário de fazer tal mudança, caso seja autorizado os dados são então persistidos.

Tipo Primário e essencial

Típica sequência de eventos

Ação do ator	Resposta do sistema
1. Solicita a criação / remoção / edição / visualização de uma propriedade.	
	2. Sistema verifica se usuário tem permissão para efetuar a atividade.
	3. Sistema persiste os dados

Seqüência alternativa de eventos

- 2 – Usuário não tem permissão para efetuar atividade. Cria log de exceção.
- 3 – Erro ao gravar os dados. Cria log de erro e avisa usuário

Diagrama do

Caso de uso

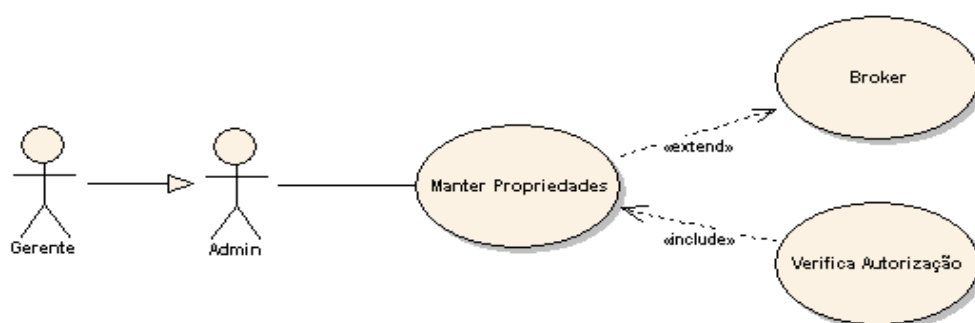


Figura 11: Caso de uso Manter Propriedade

Caso de uso Manter Rebanhos  
 Atores Transceiver RFID, Usuário, Gerente e Admin  
 Propósito Cadastrar, remover, editar e visualizar rebanhos  
 Descrição Um dos usuários vai ao sistema fazer a manutenção relativa a um rebanho. É feita uma verificação quanto à permissão do usuário de fazer tal mudança, caso seja autorizado os dados são então persistidos.

Tipo Primário

Típica sequência de eventos

Ação do ator	Resposta do sistema
1. Solicita a criação / remoção / edição / visualização de um rebanho.	
	2. Sistema verifica se usuário tem permissão para efetuar a atividade.
	3. Sistema persiste os dados

Seqüência alternativa de eventos

- 2 – Usuário não tem permissão para efetuar atividade. Cria log de exceção.
- 3 – Erro ao gravar os dados. Cria log de erro e avisa usuário

Diagrama do

Caso de uso

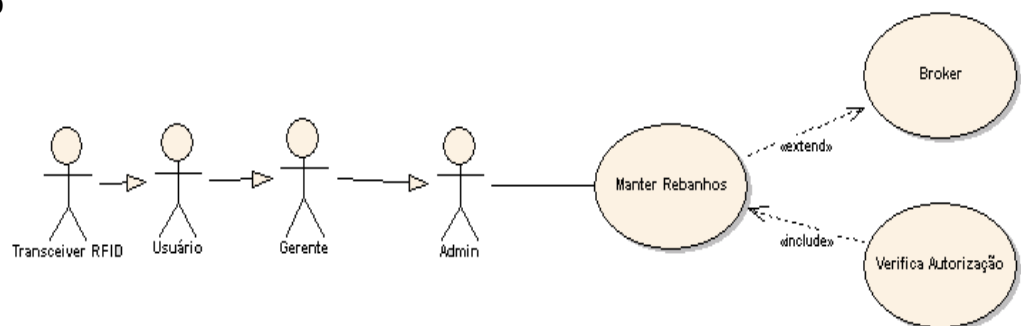


Figura 12: Caso de uso Manter Rebanhos

Caso de uso Manter Pesagens  
 Atores Transceiver RFID, Usuário, Gerente e Admin  
 Propósito Cadastrar, remover, editar e visualizar pesagens  
 Descrição Um dos usuários vai ao sistema fazer a manutenção relativa a uma pesagem. É feita uma verificação quanto à permissão do usuário de fazer tal mudança, caso seja autorizado os dados são então persistidos.

Tipo Primário

Típica sequência de eventos

Ação do ator	Resposta do sistema
1. Solicita a criação / remoção / edição / visualização de uma pesagem.	
	2. Sistema verifica se usuário tem permissão para efetuar a atividade.
	3. Sistema persiste os dados

Seqüência alternativa de eventos

- 2 – Usuário não tem permissão para efetuar atividade. Cria log de exceção.
- 3 – Erro ao gravar os dados. Cria log de erro e avisa usuário

Diagrama do

Caso de uso

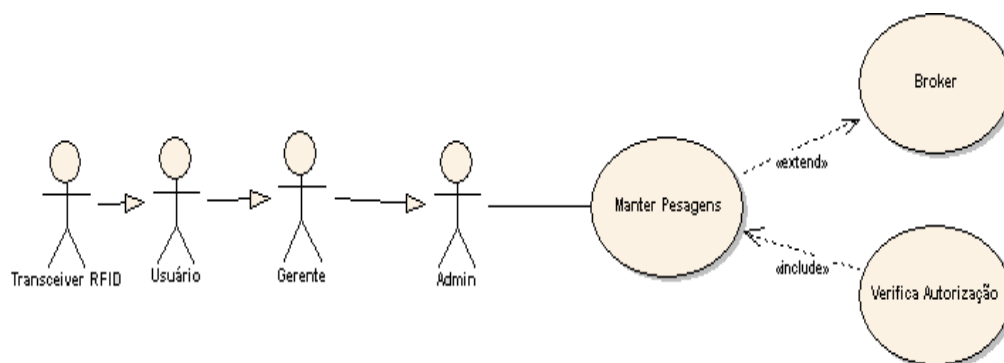


Figura 13: Caso de uso Manter Pesagens

Caso de uso	Broker
Atores	Sistema
Propósito	Gravar, Remover, Editar e Visualizar os dados do banco de dados.
Descrição	Alguma funcionalidade do sistema precisa visualizar, remover, editar ou visualizar os dados no/do banco de dados.
Tipo	Primário
Típica sequência de eventos	

Ação do ator	Resposta do sistema
1. Solicita ação (Criação, Visualização, Edição ou Remoção) Junto ao banco através do Broker	
	2. Sistema consiste os dados e efetua a operação

Seqüência alternativa de eventos

- 2 – Falha na operação. Cria log de exceção.

Diagrama do Caso de uso

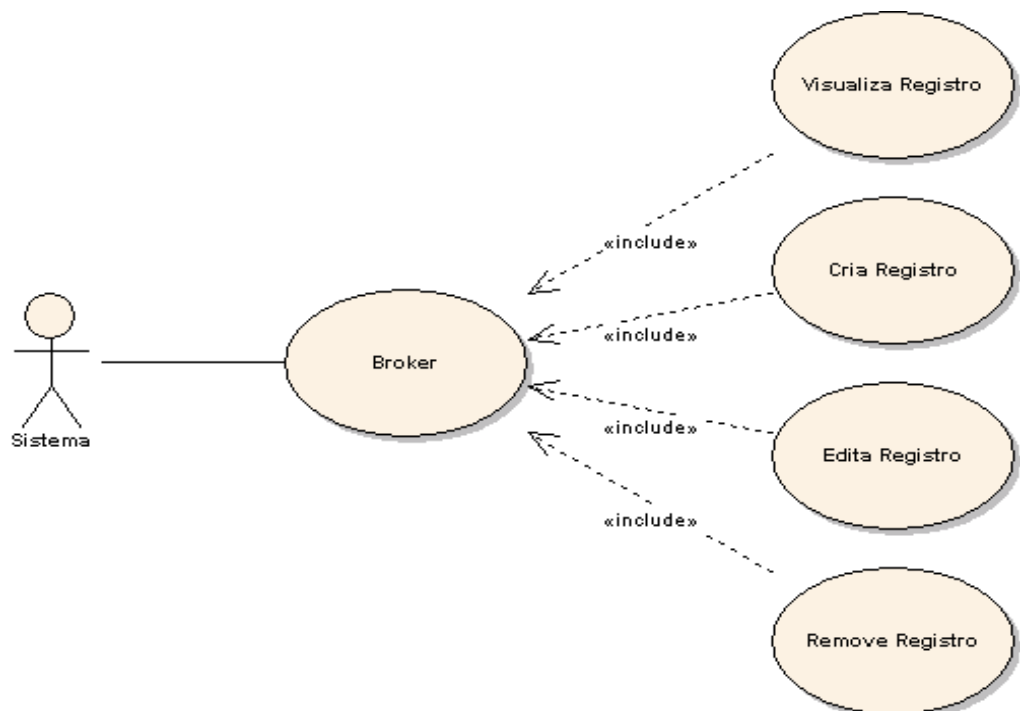


Figura 14: Caso de uso Broker

### 4.2.3. DIAGRAMAS DE CLASSES

Ver Anexos.

### 4.2.4. DIAGRAMAS DE SEQUÊNCIA

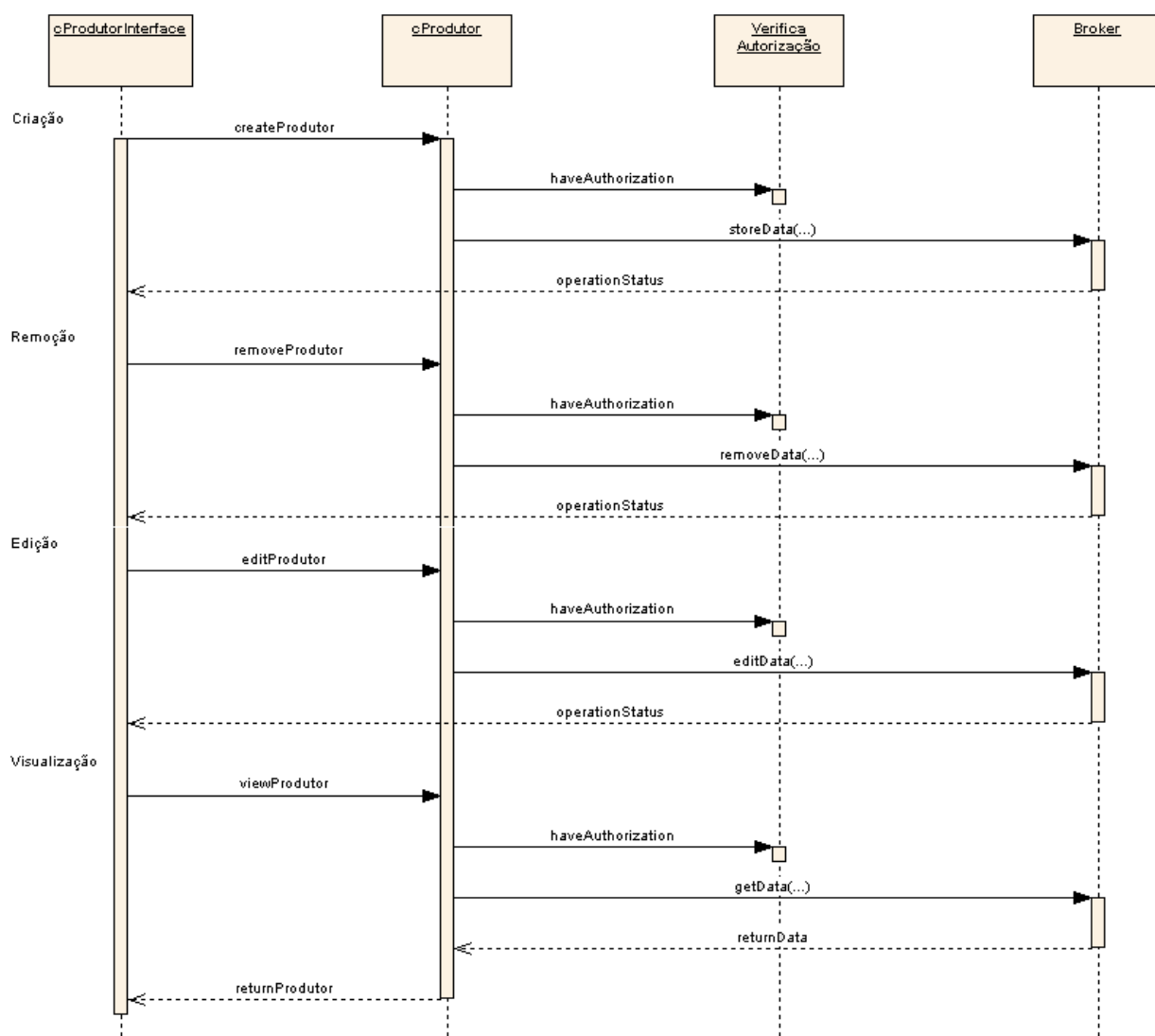


Figura 15: Diagrama de sequência do Produtor

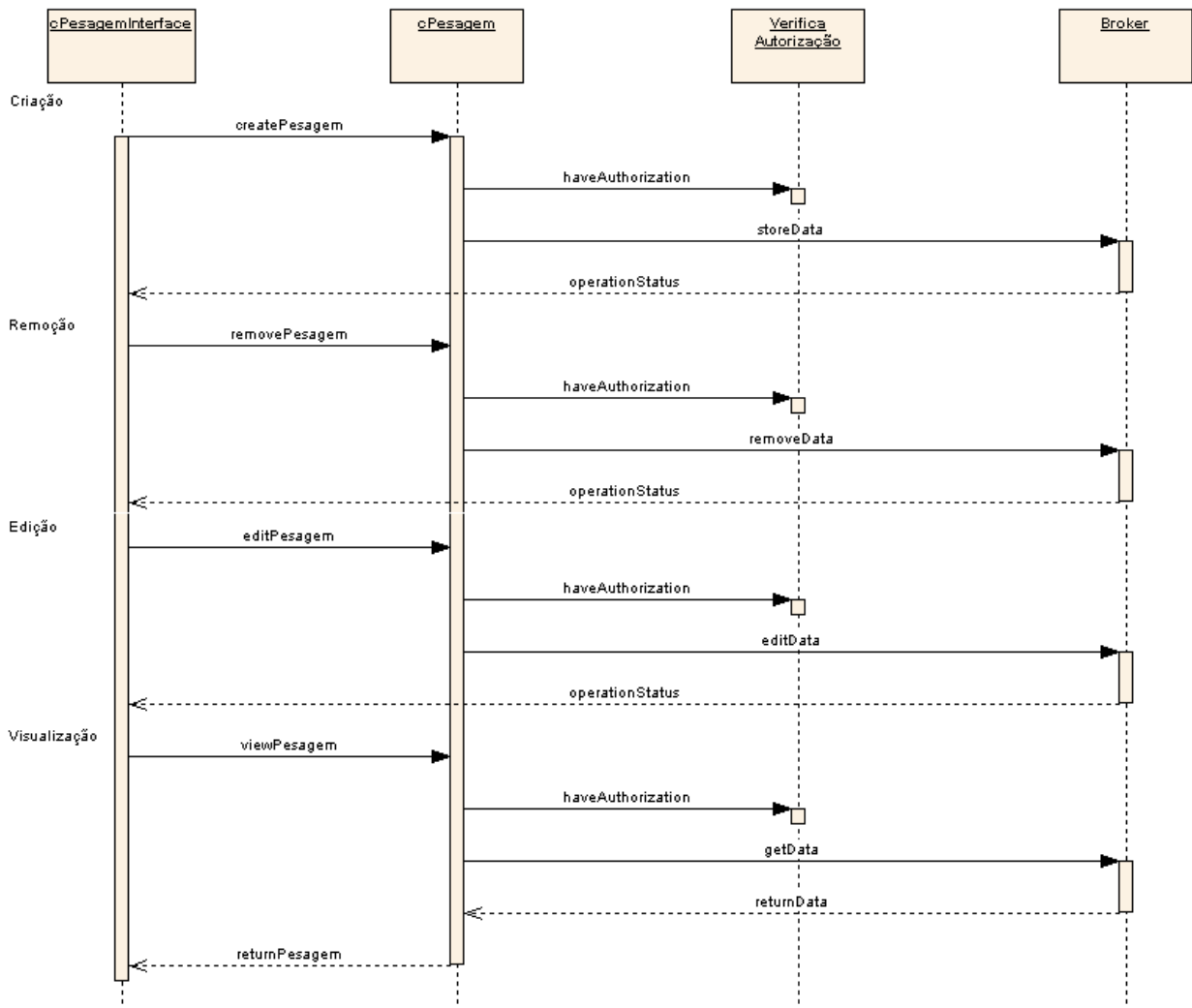


Figura 16: Diagrama de seqüência de Pesagem

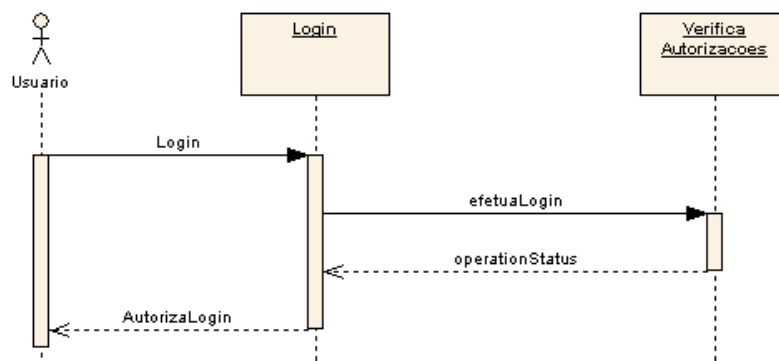


Figura 17: Diagrama de seqüência de Login



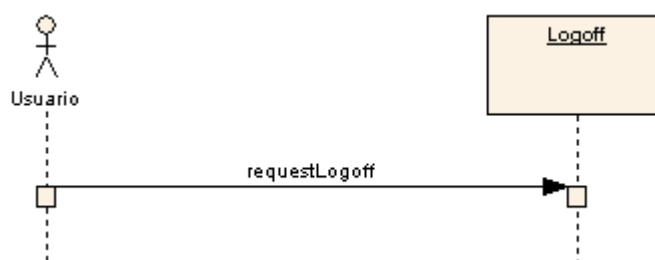


Figura 18: Diagrama de sequência de Logoff

#### 4.2.5. MODELO ENTIDADE-RELACIONAMENTO

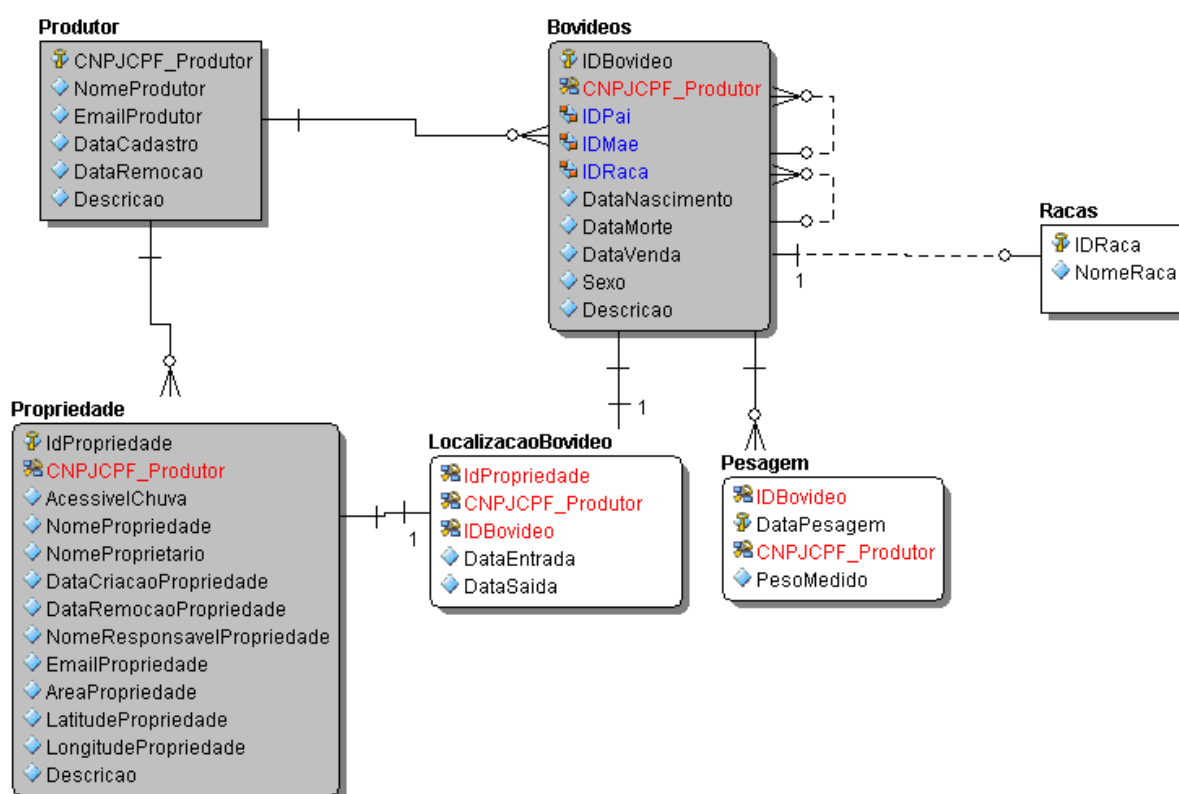
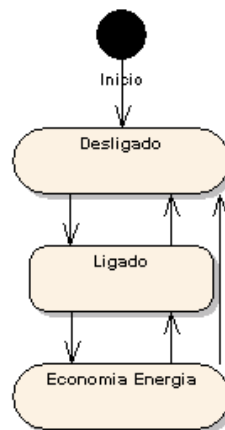


Figura 19: Modelo Entidade Relacionamento

#### 4.2.6. DIAGRAMA DE ESTADOS DO FIRMWARE



*Figura 20: Diagrama de Estados do Firmware*

## 4.2.7. FLUXOGRAMA DO FIRMWARE

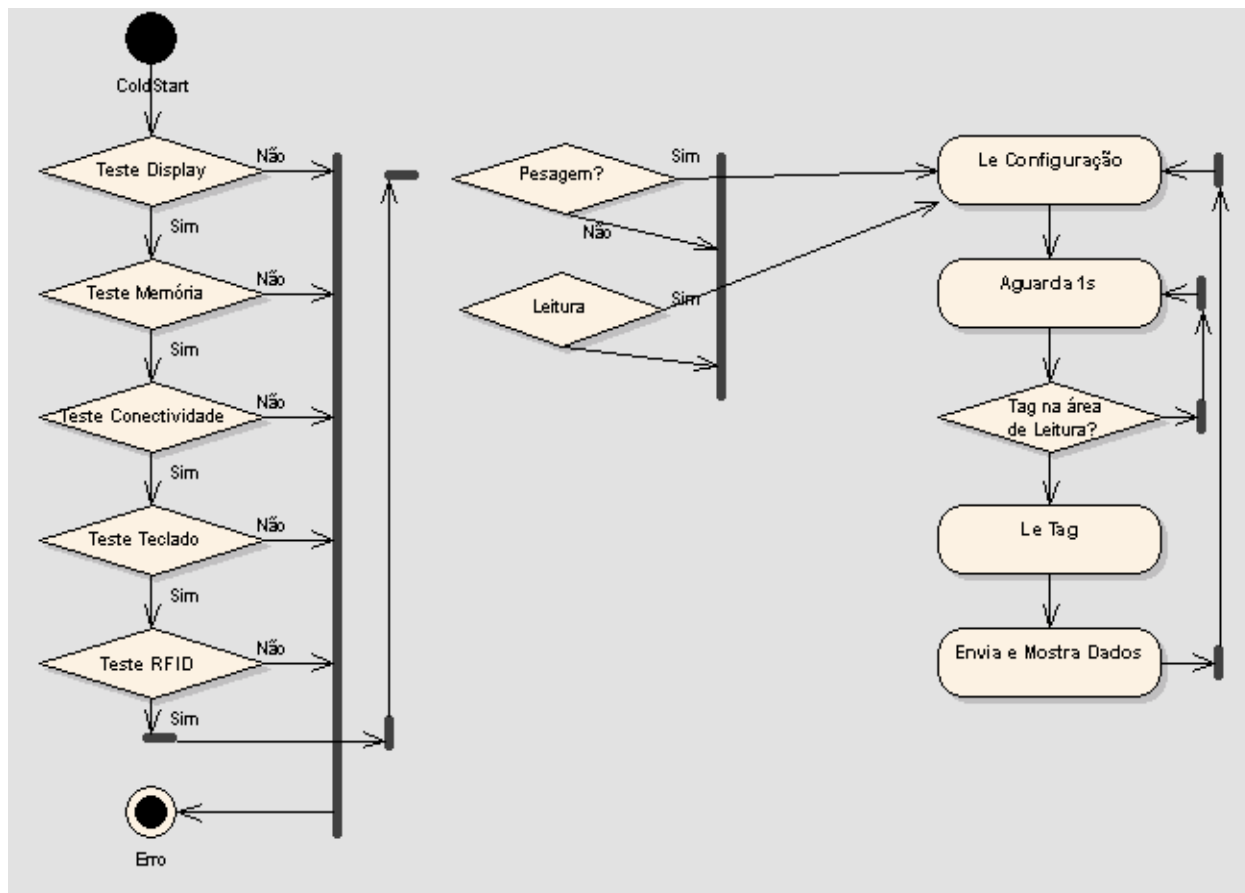


Figura 21: Fluxograma do Firmware

## **5. RESULTADOS**

Neste capítulo são apresentados os resultados obtidos durante as fases de implementação deste projeto.

Para a validação do projeto foram utilizados dois (2) transponders compatíveis com a norma ISO11784/5, sendo que a estrutura de código contida nestes transponders pode ser obtida com base na norma NBR14766, ou na própria norma ISO11784/5. Estes transponders são do tipo FDX-B.

### **5.1. COMUNICAÇÃO COM O COMPUTADOR**

Para a comunicação com o Computador foi criado um protocolo no qual fosse possível transmitir o dado recebido pelo transceiver e algumas informações adicionais. Este protocolo tem a seguinte estrutura.

- 4 dígitos para o código da operação efetuada no animal
- 4 dígitos para o país do transponder
- 12 dígitos para o código de identificação do transponder
- 30 dígitos para o envio de informações adicionais
- 1 dígito de quebra de linha para sinalizar o final da transmissão

### **5.2. DECODIFICAÇÃO DO TRANSPONDER DE RFID**

A maior dificuldade para efetuar a decodificação de um transponder RFID é que os sinais de emitidos pelo transponder não são constantes, sendo que algumas vezes acontecem algumas modificações nos tempos de envio.

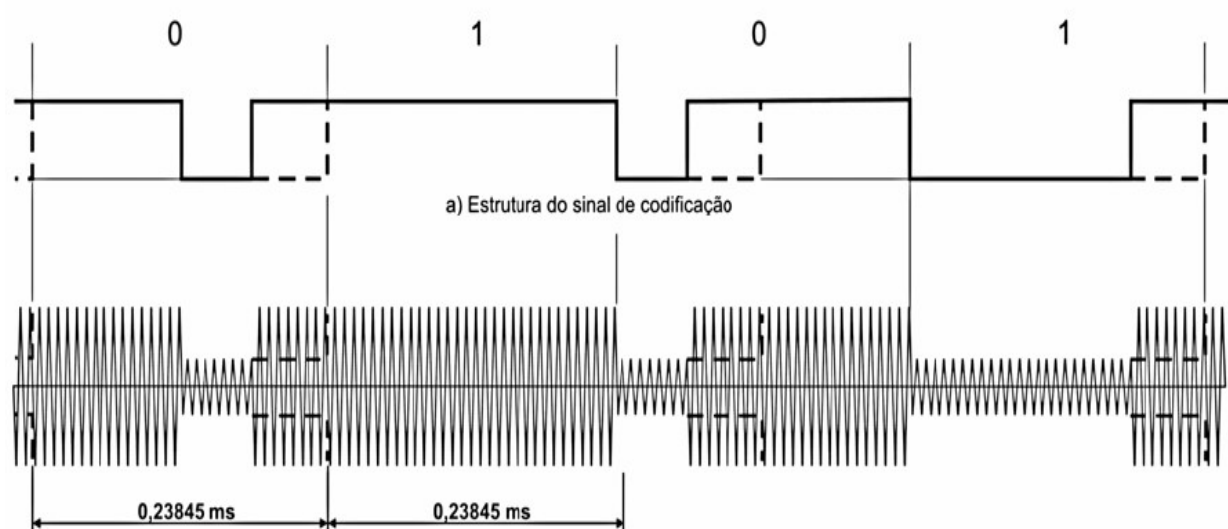


Figura 22: Estrutura do sinal de Rádio-Frequência (FDX-B)

A decodificação do transponder foi baseada nos tempos entre os ciclos de clock disponíveis no pino RDY/CLK do transceiver de rádio frequência. Como podemos notar na figura 23 o clock proveniente do transceiver oscila 32 vezes entre cada bit. Para a decodificação do transponder foi utilizada uma funcionalidade importante dos microcontroladores da família AVR da ATMEL. Esta funcionalidade permite o uso de um clock externo para sincronização, e a cada mudança de nível lógico no pino que funciona como interrupção deste clock externo, uma marcação de tempo é dada. Desta maneira, é calculado o tempo entre cada marcação para decodificar o transponder, caso a última interrupção tenha ocorrido a 28 ou mais ciclos de clock, o bit é 1, caso tenha ocorrido a 14 ou 16 ciclos, verifica-se se esta é uma segunda interrupção de 14 ou 16 ciclos, caso afirmativo o bit é 0, caso não seja uma segunda interrupção, aguarda-se a próxima interrupção. Desta maneira foi feita a decodificação dos dados do transponder.

Sabendo destes detalhes, os dados de temporização adquiridos do transponder foram então analisados (Tabela 5):

Tabela 5.1: Dados de temporização

Temporização Transponder 1																					
16	16	16	17	15	16	16	17	16	15	16	16	17	15	17	16	16	15	17	16	31	Cabecalho
32	32	32	17	16	15	17	15	17	15	16	17	15	32		Código						
32	32	17	15	32	16	16	32	31	18	16	31		Código								
32	16	16	16	17	15	16	16	17	16	15	16	17	15	16	32		Código				
17	15	16	16	16	17	16	15	16	16	17	15	17	15	17	15	32		Código			
16	16	17	16	16	15	16	17	15	17	15	16	32	32	32		Código					
17	15	17	15	16	16	16	16	32	32	32	32	32		Código							
16	16	17	15	16	16	16	17	15	16	16	17	15	17	16	16	31		Código			
17	16	15	17	15	16	16	16	16	17	16	15	17	15	32	32		Código				
Temporização Transponder 2																					
16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	31	Cabecalho
17	16	16	16	32	16	16	16	16	16	16	16	16	16	16	17	30		Código			
17	16	32	31	32	33	16	16	32	16	16	31		Código								
33	16	16	16	16	16	16	16	16	16	16	16	16	16	16	31		Código				
17	16	16	16	17	15	16	16	16	16	16	16	16	16	16	16	32		Código			
16	16	16	16	16	16	16	16	16	16	16	16	31	32	32		Código					
18	15	16	16	16	16	16	16	31	32	33	31	32		Código							
16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	31		Código			
17	16	17	15	16	16	16	16	16	16	16	16	16	16	31	32		Código				

Tendo em mãos os dados de temporização dos transponders o próximo passo foi convertê-los para os respectivos binários. Para cada par de temporizações entre 14 e 18 temos um bit 0, e para cada temporização de 28 a 34 temos um bit 1. Desta maneira o código proveniente da temporização dos transponders pode ser decodificado para (tabela 6):

Tabela 5.2. Dados decodificados da temporização (Com bit de sinalização do bloco de 8 bits)

	<b>Transponder 1</b>	<b>Transponder 2</b>
Cabeçalho	0000000001	0000000001
1° bloco de dados	111000001	001000001
2° bloco de dados	110101101	011110101
3° bloco de dados	100000001	100000001
4° bloco de dados	000000001	000000001
5° bloco de dados	000000111	000000111
6° bloco de dados	000011111	000011111
7° bloco de dados	000000001	000000001
8° bloco de dados	000000011	000000011

Usando a estrutura do código descrita na tabela 4 analisamos os dados do transponder de acordo com o código descrito na ISO11784/5 (NBR14766) temos o seguinte resultado (Tabela 7).

Tabela 5.3: Dados binários do transponder na ordem correta

	<b>Transponder 1</b>	<b>Transponder 2</b>
<b>1° bloco de dados</b>	10000000	10000000
<b>2° bloco de dados</b>	00000000	00000000
<b>3° bloco de dados</b>	11110000	00001111
<b>4° bloco de dados</b>	11000000	00000011
<b>5° bloco de dados</b>	00000000	00000000
<b>6° bloco de dados</b>	00000001	00000001
<b>7° bloco de dados</b>	01101011	01011110
<b>8° bloco de dados</b>	00000111	00000100

Utilizando agora a os dados da tabela 4 chegamos aos seguintes valores:

Tabela 5.4: Dados decodificados - Linha 1 = Transponder 1, Linha 2 = Transponder 2

<b>A</b>	<b>Reservado</b>	<b>D</b>	<b>País</b>	<b>Número</b>
1	0000000000000000	0	1111000011	000000000000000000000000010110101100000111
1	0000000000000000	0	1111000011	000000000000000000000000010101111000000100

O próximo passo para decodificar o transponder é uma simples conversão de binário para decimal dos dados contidos no bloco do País, e da mesma maneira para o bloco de Número, sendo o resultado os números lidos dos dois transponders testados.

*Tabela 5.5: Resultado da leitura dos transponders*

	<b>País</b>	<b>Número</b>
<b>Transponder 1</b>	963	92935
<b>Transponder 2</b>	963	89604

### **5.3. DISTÂNCIA DE LEITURA**

A distância de leitura do transponder em relação a antena é diminuta (em torno de 2mm). Isto ocorre devido a uma série de fatos introduzidos no projeto pela antena escolhida para desenvolvimento. O problema ocorre devido ao fato desta antena ter um altíssimo fator de qualidade e uma baixíssima resistência.

O transceiver de rádio frequência suporta uma corrente máxima de 250mA nos pinos de entrada da antena, devido ao fator de qualidade muito alto desta antena foi necessário adicionar uma resistência externa ao circuito para reduzir a corrente máxima da antena, o que diminuiu também a tensão de pico e a tensão normal na antena. Os valores das tensões respectivamente são: 12,51V para a tensão de pico e 7,61V para a tensão normal. Em um produto comercial as antenas alcançam facilmente tensões acima de 200V em pico na antena, e tensões normais acima de 150V, desta maneira emitindo um campo eletromagnético mais forte do que o emitido pela antena utilizada no circuito.

### **5.4. TAXA DE TRANSMISSÃO TRANSPONDER → TRANSCEIVER**

A taxa de transmissão de dados do transponder com o transceiver é de 4,194kbits por segundo.





## 6. CONCLUSÃO

### 6.1. CONCLUSÃO

Durante a fase de documentação não foram previstos alguns problemas que haveriam durante a execução deste projeto, sendo que o maior desafio a ser transpassado foi fazer com que o transceiver de rádio frequência viesse a funcionar, esta fase tomou do total da execução do projeto aproximadamente 2 meses, fazendo com que não fosse possível fazer um projeto mais próximo de um produto. Para a finalização do mesmo é necessário o desenvolvimento e testes de uma placa de circuito impresso bem como o desenvolvimento de uma caixa para o produto.

Em relação aos resultados obtidos, a distância de leitura é muito pequena, para que a antena passasse a perceber o transponder em sua proximidade distância foi de 1cm, e para a correta interpretação dos dados contidos no transponder o mesmo deve estar a aproximadamente 2mm da antena. Isso se deve ao fato da antena não ser a ideal para o uso com o transceiver de rádio frequência utilizado neste projeto.

Se compararmos a distância de leitura obtida com o sistema desenvolvido neste projeto, a diferença entre as distâncias de leitura é bem grande, sendo que um leitor comercial consegue chegar facilmente entre 10 a 20 cm de distância para perceber e ler o dado dos transponders na área de leitura.

A aplicabilidade do produto fica inviável com esta distância de leitura, sendo necessário o desenvolvimento de uma nova antena para que as distâncias de leitura superem ao mínimo 5cm.

Quanto ao desenvolvimento do software o ganho em tempo de desenvolvimento foi grande devido principalmente à mudança da linguagem de programação de C/C++ para Java. Um outro fator que agilizou muito o tempo de desenvolvimento foi o uso da ferramenta Hibernate. Esta ferramenta faz toda a camada de abstração entre banco e aplicação, fazendo com que os dados provenientes do banco de dados sejam todos mapeados em Objetos, facilitando o trabalho e manuseio dos dados. Como exemplo para simplificar o entendimento, para a classe Propriedade, ao pesquisar um dado no banco referente a alguma propriedade, os mapeamentos com as outras tabelas como

as de Bovídeos, e Produtores eram todos executados automaticamente, sendo que os dados referentes a estes mapeamentos são automaticamente mapeados como listas adjacentes do objeto gerado inicialmente pela pesquisa da classe propriedade.

Quanto ao banco de dados, a partir da premissa de que foi utilizado o framework hibernate, deixa de ocorrer a necessidade de uso de um banco de dados específico, pois existe uma grande flexibilidade no framework para a troca transparente de banco de dados. Foram executados alguns testes com outros bancos de dados como o HSQLDB, MySQL, Derby entre outros, mas optei por manter o PostgreSQL pois a performance obtida foi melhor que a de todos os outros bancos de dados citados.

## **6.2. Discussão**

Dentre as melhorias possíveis estão o desenvolvimento de uma nova antena, para que o sistema possa efetuar leituras a uma distância mais efetiva.

Como sugestões de outros projetos um deles pode ser o uso deste sistema para o desenvolvimento de um cadastro para pacientes de um hospital por exemplo, sendo que quando os pacientes estiverem dentro do hospital estes portem dispositivos de RFID em algum tipo de pulseira ou colar e estes tragam todos os dados relativos a este paciente, facilitando o acesso do médico aos diagnósticos.

## 7. BIBLIOGRAFIA

FINKENZELLER, K., RFID handbook: fundamentals and applications in contactless smart cards and identification; 2nd ed., John Wiley & Sons, 2003.

BHUPTANI, M.; MORADPOUR, S.; RFID Field Guide: Deploying Radio Frequency Identification Systems; Prentice Hall, 2005.

HAROLD, ELLIOTE R.; JAVA I/O; 1<sup>st</sup> ed.; O'Reilly; 1999;

SPERBERG-MCQUEEN, C. M.; Extensible Markup Language (XML) 1.1. W3C. (<http://www.w3.org/TR/2004/REC-xml11-20040204/>), Fevereiro 2004, (24/03/05).

LINHARES, MAURÍCIO; Introdução ao Hibernate 3; Junho 2005.

OLIVEIRA, DANIEL Q.; Livre-se do SQL: Uma introdução ao hibernate; (<http://www.guj.com.br/java/tutorial/artigo.125.1.guj>) (11/10/2005)

VASCONCELOS, GOMES; Usando a COMM API; Junho, 2004d

BRASIL, Ministério da Agricultura, Pecuária e Abastecimento.; Instrução Normativa N° 1, Brasília, 9 de Janeiro de 2002.

RFIDJOURNAL. What is RFID? (<http://www.rfidjournal.com/article/articleview/1339/1/129/>) (24/03/05).

## **8. ANEXOS**

**8.1. ESQUEMÁTICOS**

**8.2. LISTA DE MATERIAIS**

**8.3. DIAGRAMA DE CLASSES**