

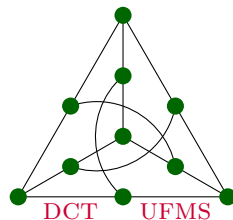
# *Middleware* **Cerberus** usando RFID para Rastreabilidade Bovina

Márcio Roberto Silva

Dissertação de Mestrado

Orientação: Prof. Dr. Edson Norberto Cáceres

Dissertação apresentada ao Departamento de Computação e Estatística da Universidade Federal de Mato Grosso do Sul como parte dos requisitos para obtenção do título de  
**Mestre em Ciência da Computação.**



Departamento de Computação e Estatística  
Centro de Ciências Exatas e Tecnologia  
Universidade Federal de Mato Grosso do Sul  
21 de agosto de 2009

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

# Conteúdo

<b>Agradecimentos</b>	<b>4</b>
<b>Resumo</b>	<b>5</b>
<b>Abstract</b>	<b>6</b>
<b>Lista de Figuras</b>	<b>7</b>
<b>Lista de Tabelas</b>	<b>9</b>
<b>1 Introdução</b>	<b>10</b>
1.1 Rastreabilidade . . . . .	10
1.2 Identificação Automática . . . . .	11
<b>2 Conceitos Básicos e Tecnologias</b>	<b>13</b>
2.1 Identificação por Rádio Frequência . . . . .	13
2.2 Rastreabilidade Bovina . . . . .	14
2.3 <i>Middleware</i> RFID . . . . .	15
2.4 Balança de Passagem . . . . .	17
<b>3 Análise do Problema</b>	<b>18</b>
3.1 Definição do problema . . . . .	18
3.2 Soluções apontadas . . . . .	19
3.3 O problema decorrente do uso da tecnologia RFID . . . . .	20
<b>4 <i>Middleware</i> Cerberus</b>	<b>22</b>
4.1 Descrição do <i>Middleware</i> . . . . .	23

4.2	Arquitetura . . . . .	24
4.2.1	Camada Física . . . . .	25
4.2.2	Camada Gerenciamento . . . . .	26
4.2.3	Camada Integração . . . . .	30
4.3	Configuração do <i>middleware</i> . . . . .	31
<b>5</b>	<b>Estudo de Caso</b>	<b>37</b>
5.1	Sistema de Balança de Passagem . . . . .	37
5.1.1	Desafios . . . . .	38
5.1.2	Solução Proposta . . . . .	38
5.1.3	Resultados . . . . .	40
<b>6</b>	<b>Trabalhos Relacionados</b>	<b>42</b>
<b>7</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>44</b>
7.1	Contribuições da Dissertação . . . . .	44
7.2	Conclusões . . . . .	45
7.3	Extensões . . . . .	46
	<b>Referências Bibliográficas</b>	<b>47</b>
	<b>Anexo I</b>	<b>49</b>
	<b>Anexo II</b>	<b>51</b>
	<b>Anexo III</b>	<b>53</b>
	<b>Anexo IV</b>	<b>55</b>

# Agradecimentos

À Deus pela sua infinita bondade.

À minha amada família por seu apoio e presença em todos os momentos de minha vida. Em especial, aos meus pais, Romeu e Maria Francisca, que sempre estiveram do meu lado e aos meus irmãos Marcos, Cláudia e Mauro.

Um agradecimento especial a minha esposa, Mara, pelo seu apoio e carinho e estímulo durante todo o meu mestrado e aos meus filhos, Luan e Nathan.

A todos os professores e funcionários do Departamento de Computação e Estatística (DCT) e do Núcleo de Informática (NIN) da Universidade Federal de Mato Grosso do Sul (UFMS), que me sempre me acolheram bem.

A todos os funcionários, estagiários e colaboradores da Empresa Brasileira de Pesquisa Agropecuária de Gado de Corte (EMBRAPA) que me receberam muito bem e onde tive a oportunidade de aprender muitas coisas.

É um agradecimento especial ao professor Dr. Edson Norberto Cáceres pela sua ajuda e sua extrema paciência para comigo e ao Dr. Pedro Paulo Pires a quem eu aprendi a admirar como pesquisador e como pessoa.

E a todos os meus amigos de Campo Grande,MS e de Bambuí,MG que tenho certeza que estão sempre torcendo por mim.

Este trabalho foi financiado pela Fundect e teve o apoio dos projetos e-SAPI bovis e OTAG.

*Se pude enxergar mais longe foi porque estava sobre os ombros de gigantes.*  
Isaac Newton

# Resumo

Silva, M.R. *Middleware **Cerberus** usando RFID para Rastreabilidade Bovina*. Campo Grande, 2009. Dissertação de Mestrado – Universidade Federal de Mato Grosso do Sul.

A gerência na zootecnia de precisão é cada vez mais necessária para garantia de lucros e a conquista de novos mercados os quais exigem qualidade e certificação. Nesta dissertação é abordado um sistema de rastreabilidade para o controle na bovinocultura utilizando *chips* RFID. Sendo que a identificação segura dos animais é a base para a rastreabilidade bovina e bubalina.

Em vista as exigências do mercado consumidor o governo brasileiro instituiu o Sistema Brasileiro de Identificação e Certificação de Origem Bovina e Bubalina (SISBOV) que permite a utilização de diferentes métodos para a identificação dupla dos animais, entre eles: brinco e *botton*, brinco e marca a fogo, brinco e tatuagem e brinco com dispositivos eletrônicos. A utilização de dispositivos eletrônicos é sem dúvida a forma mais segura e eficiente para identificação dos animais, no entanto ao se identificar animais com dispositivos eletrônicos, os custos aumentam significativamente.

Diante deste cenário justifica-se o desenvolvimento de ferramentas computacionais que incentivem os produtores a utilizarem a identificação eletrônica em seus rebanhos, como uma forma segura e efetiva de rastreabilidade e produtividade. É fundamental que estas ferramentas possibilitem aos produtores, além da rastreabilidade, a obtenção de informações gerencias para uma melhor administração do rebanho.

O objetivo deste trabalho é propor uma plataforma de *software* que promova o desenvolvimento de novas aplicações na área do agronegócio, esta plataforma deverá abstrair a conexão com os dispositivos utilizados no agronegócio, principalmente as conexões com estrutura de Rádio Frequência (RFID), afim de gerenciar e formatar os dados coletados por estes dispositivos. Com a utilização desta plataforma de *software* poderão ser desenvolvidas novos aplicativos para a administração de rebanhos, independentes da parte física representada pelos dispositivos.

## Palavras-chave

*Middleware* RFID, RFID, EPC, Identificação por Rádio Frequência, Sistema de Balança de Passagem, rastreabilidade bovina.

# Abstract

Silva, M.R. *Middleware **Cerberus** using RFID for Traceability Bovine*. Campo Grande, 2009. Dissertation of Master's degree—Federal University of Mato Grosso do Sul.

The management in zootechnics is increasingly necessary to ensure the profits and conquest of new markets that require a quality and certification. This monograph a traceability system is approached for the control in the bovinocultura using chips RFID. The identification holds of animals is the basic for trace cattle and buffaloes.

Given the demands of the consumer market the Brazilian government institutes the Brazilian System of Identification and Certification of Origin Cattle and Buffaloes (SISBOV) allowing the use of different methods for the animal identification, among them: eartags together BOTTON, eartags together mark of fire, eartags together tattoos, and eartags together electronic devices. The use of electronic devices is undoubtedly the way safest and more efficient for animal identification, however is to identify animals with electronic devices, the costs increase significantly.

Facing this scenario is warranted the development of computational tools that encourage producers to use electronic identification in their herds, as a base to a safe and effective traceability and productivity. It is fundamental that these tools make possible to the producers, besides the traceability, the obtaining of information manages for a best administration of the herd.

The objective of this work is to provide a platform of software that promote the development of new applications in the area of the agribusiness, this platform should abstract the connection with the devices used in the agribusiness, mainly the connections with structure of Radio Frequency (RFID), similar of to manage and to format the data collected by these devices. With the use of this platform of software new applications can be developed for the administration of herd, independent of the physical part acted by the devices.

## Keywords

*Middleware* RFID, RFID, EPC, Radio Frequency Identification, System Balance of Passage, bovine traceability.

# Lista de Figuras

2.1	Típico sistema RFID. Extraído de Prabhu[21]. . . . .	14
2.2	Balança de Passagem. . . . .	17
3.1	Elementos de Identificação: brinco, marca a fogo, dispositivo eletrônico, tatuagem. . . . .	19
4.1	Visão funcional dos componentes. A transmissão de dados entre aplicação e os equipamentos é feita por meio do <i>middleware</i> . . . . .	23
4.2	Arquitetura do <i>middleware</i> <b>Cerberus</b> . . . . .	24
4.3	Diagrama de Classes que representa a camada física. Composto pelas classes Connect e SerialReader. . . . .	26
4.4	Diagrama de Classes que representa a camada de gerenciamento. Composto pelas classes: ManagerEventData, ConnectPort. . . . .	27
4.5	Diagrama de Classes do núcleo do <i>middleware</i> <b>Cerberus</b> . . . . .	34
4.6	Diagrama de sequência das classes que compõem o núcleo do <i>middleware</i> <b>Cerberus</b> . . . . .	35
4.7	Tela de configuração do <i>middleware</i> <b>Cerberus</b> . . . . .	36
5.1	Gráfico obtido da tabela[9.1]: variação dos pesos, obtido pelo Sistema de Balança de Passagem. . . . .	39
7.1	Contextualização do <i>middleware</i> <b>Cerberus</b> frente ao portal <i>e-Sapi bovis</i> . .	45



# Lista de Tabelas

3.1	Custo da identificação e rastreabilidade animal utilizando diferentes métodos em função da quantidade de animais, em reais (R\$).Fonte: Lopes[13]	20
3.2	Custo da identificação e rastreabilidade animal utilizando diferentes métodos em função da quantidade de animais, em reais (R\$).Fonte: Lopes[13]	21
4.1	Camadas que compõem o núcleo do <i>middleware</i> .	23
4.2	Métodos da classe <i>Connect</i> .	25
4.3	Métodos da classe <i>SerialReader</i> .	26
4.4	Métodos da classe <i>ManagerEventData</i> .	28
4.5	Métodos da classe <i>ConnectPort</i> .	30
4.6	Métodos da classe <i>WriteFile</i> .	31
4.7	<i>Tags</i> de configuração do <i>middleware</i> .	33
5.1	Tabela das médias dos pesos diários, calculado das tabelas (9.3,9.4 ,9.5,9.6,9.7,9.8). Peso em kg.	40
6.1	Tabela comparativa entre os <i>middlewares</i> pesquisados e o <b>Cerberus</b> .	43
9.1	Tabela de pesagem do animais, passando sobre a balança. Tempo em segundo e o peso em kg.	53
9.2	Tabela de pesagem do animais, passando sobre a balança. Tempo em segundo e o peso em kg, continuação9.1.	54
9.3	Tabela de coleta do peso dos animais parado. Peso em kg.	55
9.4	Tabela de coleta do peso dos animais em marcha. Peso em kg.	55
9.5	Tabela de coleta do peso dos animais em marcha. Peso em kg.	56
9.6	Tabela de coleta do peso dos animais em marcha. Peso em kg.	56
9.7	Tabela de coleta do peso dos animais em marcha. Peso em kg.	56

9.8	Tabela de coleta do peso dos animais em marcha. Peso em kg. . . . .	56
-----	---	----

# Capítulo 1

## Introdução

Uma cadeia produtiva consiste de um conjunto de componentes do qual fazem parte sistemas produtivos, fornecedores de insumos e serviços, indústria de processamento e transformação, agentes de distribuição e comercialização, além dos consumidores finais. O interesse pelas cadeias produtivas está aumentando gradativamente devido as necessidades do mercado e das exigências dos consumidores. Com isso, para se determinar o critério de qualidade de um produto já são incluídos alguns itens como: se a técnica de produção de uma de suas matérias-primas é potencialmente prejudicial ao meio ambiente, se utiliza mão-de-obra infantil, se o produto contém ingredientes geneticamente modificados, entre outros.

Para que estes itens possam ser verificados, é necessário um arcabouço que permite a rastreabilidade<sup>1</sup> dos produtos, e para isso é necessário a identificação dos mesmos.

### 1.1 Rastreabilidade

Rastreabilidade é um conceito que permeia vários tipos de atividades e refere-se à habilidade de descrever e seguir a vida de um elemento conceitual ou físico. A rastreabilidade poder ser realizada a partir da origem, desenvolvimento até a utilização dos produtos (chamada de rastreabilidade para frente ou *forward*) e também pode ocorrer em ordem inversa (rastreabilidade para trás, *backward*[16]. Em sistemas de banco de dados, a noção de rastreabilidade está associada à execução de transações e é baseada em *log*. Em engenharia de software, está relacionada à etapas da construção de um software desde a parte de requisitos até o produto final. Para um domínio alvo específico, rastreabilidade consiste na possibilidade de acompanhar as atividades e processos executados entre dois estágios no tempo.

As questões de rastreabilidade vêm tomando um papel importante em todos os setores, como por exemplo, em produtos agrícolas, ganhando significativa importância e destaque com os eventos ligados à segurança alimentar, como os ocorridos nas décadas

---

<sup>1</sup>Termo extraído do Glossário de Termos Usados em Atividades Agropecuárias, Florestais e Ciências Ambientais [18].

de 80 e 90, em especial na Europa, um dos principais mercados consumidores de carne bovina do Brasil. Embora o mais emblemático tenha sido o mal da vaca louca, houve também diversos casos de contaminações de produtos, como leite e carne de frango, entre outros[17]. Estes eventos trouxeram à tona, as crescentes preocupações dos consumidores com a origem e a forma de produção dos produtos, tais como:

- A origem genética do produto (geneticamente modificados ou não);
- A forma de produção dos pontos de vista: agrônômicos (produtos “orgânicos” ou não);
- Trabalhista e social: uso de mão de obra infantil, condições de trabalho e legislação trabalhista;
- Sanitário: condições de higiene na produção, dos animais e dos produtos;
- Ambiental: impacto ambiental da produção, entre outras.

Atualmente, a rastreabilidade é um instrumento essencial para a comercialização da carne bovina, porém a implantação da rastreabilidade utilizando uma estrutura RFID (leituras e etiquetas), ainda é considerada de alto custo. A proposta deste trabalho é incentivar o desenvolvimento de aplicativos que utilizem esta tecnologia, agregando novos serviços para quem utiliza a identificação eletrônica.

## 1.2 Identificação Automática

Procedimentos de identificação automática de objetos têm se tornado populares para diferentes aplicações da indústria, logística, controle de estoque, distribuição e diversos outros. A identificação automática tem o objetivo de prover informações a respeito de pessoas, animais, ativos e produtos, sem intervenção humana [9], isto é, automaticamente.

Em diferentes tipos de aplicação, a tecnologia de código de barras, antes revolucionária, se tornou inadequada devido ao fato de que o código de barras possui pequena capacidade de armazenamento, não pode ser reprogramado e necessitar de manuseio para que se consiga uma leitura da etiqueta do item em questão, impossibilitando, em alguns casos, a identificação automática. Uma alternativa ao uso de código de barras é o armazenamento da identificação do item em *chips*<sup>2</sup> que possuem maior capacidade de armazenamento. O armazenamento de informações em *chips* acoplados a micro antenas, capazes de transmitir as informações armazenadas via rádio frequência é chamada de tecnologia RFID (*Radio Frequency Identification*)[6, 9].

Existem muitos tipos diferentes de tecnologia RFID[5], isto faz com que sistemas que usam esta tecnologia se tornem dependentes de uma arquitetura específica, desmotivando o investimento em sistemas voltados para tecnologia RFID. Aliada a este fato tem os

---

<sup>2</sup>Os termos *tag*, *transponder*, *chip* e etiqueta RFID são usados como sinônimos.

altos custo da tecnologia RFID ( leitoras e *transponders* ), tornando a implantação deste sistema difícil. Na contramão destes fatores encontra-se a rastreabilidade bovina, que necessita de um sistema que possa identificar cada animal, a identificação unívoca é a base do processo de rastreabilidade.

Chegamos ao seguinte dilema, o custo das *tags* RFID é alto porque ainda é pouco utilizado, e é pouco utilizado porque o seu custo ainda é muito alto. Neste trabalho é apresentado uma plataforma de software que motiva o uso da tecnologia RFID contribuindo de forma direta na solução deste impasse.

Sistemas que utilizam identificação por rádio frequência usam mais do que *tags* e leitoras [3]. Para agregar valores a todas as informações coletadas, usuários da tecnologia RFID necessitam de plataforma de software que manipule os dados coletados, softwares que situam-se entre leitores RFID e aplicativos são comumente referenciados como *middleware* RFID. Este é um componente crítico de qualquer sistema RFID, porque um *middleware* RFID recebe dados de uma leitora capaz de ler 100 vezes por segundo a mesma *tag*. Ficando a cargo do *middleware* ler a informação correta entregá-la ao aplicativo correto e na hora certa.

O objetivo desta dissertação é contribuir com a popularização da tecnologia RFID tornando-a uma tecnologia acessível, para isso apresentamos uma plataforma de software denominada *middleware* **Cerberus**, para estimular o desenvolvimento de sistemas que utilizam RFID. Para avaliar o *middleware*, realizamos um estudo de caso para o qual desenvolvemos um sistema que utiliza o *middleware* para automatizar a pesagem de animais em campo, denominado balança de passagem.

A proposta é que o Cerberus seja uma plataforma de software genérica, abstraindo a conexão dos equipamentos que utilizam portas seriais para comunicar com o computador, e que o sistema de balança de passagem seja um motivador para os pecuarista que deseja utilizar a tecnologia RFID para implantar um sistema de rastreabilidade confiável em suas propriedade. Ao estudo de caso proposto, várias aplicações importantes podem ser adicionadas.

## Capítulo 2

# Conceitos Básicos e Tecnologias

Neste capítulo, são apresentados os principais conceitos necessários para um entendimento adequado do tema da dissertação. Como a proposta desta dissertação envolve a criação de um *middleware* RFID para a instanciação de sistemas capazes de interagir com esta tecnologia, conceitos de *middleware*, tecnologia RFID, e uma visão de rastreabilidade e balança de passagem para aferição do peso dos animais são apresentados a seguir.

### 2.1 Identificação por Rádio Frequência

Identificação por rádio frequência, do inglês ( *Radio frequency identification* (RFID) ) é o termo genérico usado para descrever um sistema que transmite a identidade (em forma de um número serial único) de um objeto ou animal, usando ondas de rádio[5]. Auto-ID é o termo usado para agrupar todas as tecnologias de identificação automática, incluindo código de barras, leitores ópticos e biometria. Tecnologias de auto-ID [2] têm sido usadas para diminuir o tempo e o trabalho necessário para entrada de dados.

Algumas tecnologias de auto-ID, tais como sistemas de códigos de barras, necessitam da intervenção humana no manuseio da leitora para escanear o código de barras. Leitoras RFID capturam os dados das *tags* e transmitem para o computador, sem necessidade de envolver uma pessoa. Esse tipo de identificação é denominado Identificação Automática.

Um típico sistema RFID consiste de *transponders* (*tags*), leitora(s), antenas e um *host* (computador para processar os dados) como mostra a Figura 2.1.

A comunicação em RFID ocorre através de ondas de rádio, onde a informação transita de uma *tag* para o leitor ou vice-versa, passando por uma antena. A identificação única ou dados eletrônicos são armazenados em uma *tag* RFID, os quais podem consistir de um número serial, código de segurança, código do produto e outros dados específicos do objeto. Usando leitoras RFID estes dados podem ser lidos sem a necessidade de um contato direto.

Prabhu [21], aponta as seguintes características da estrutura RFID que estimulam o investimento nesta nova tecnologia.

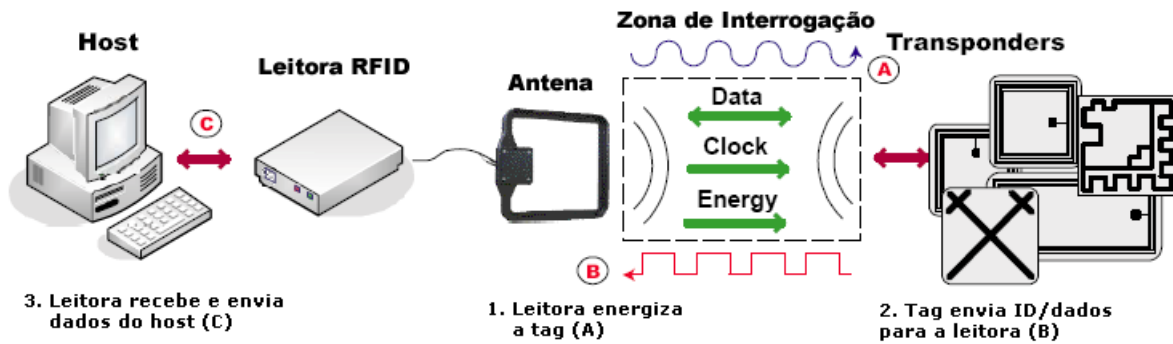


Figura 2.1: Típico sistema RFID. Extraído de Prabhu[21].

- *Transponders* podem armazenar dados que podem ser modificados e atualizados durante vários estágios dos processos.
- Automatização da coleta dos dados, sem a necessidade de manuseio ou contato visual.
- Recuperação dos dados de forma correta, aumentando a precisão na tomada de decisão.
- Diminuição no manuseio das mercadorias, portanto menos trabalho requerido.
- Identificação simultânea de vários itens na mesma área do leitor.

Devido a estas características, acredita-se que a tecnologia de identificação por rádio frequência (RFID) torne possível a automatização do rastreamento de itens através de uma rede de produção.

A tecnologia RFID é um conceito simples que pode ter muitas aplicações e grandes impactos em determinados nichos de mercado. Ao se colocar uma etiqueta, “um microchip com uma antena”, em produtos ou em animais, é criada a possibilidade de computadores localizarem e identificarem cada produto/animal individualmente. Uma das grandes barreiras para adoção desta tecnologia é o seu custo e a integração com sistemas existentes. A proposta deste trabalho é abstrair a integração com uma estrutura RFID e oferecer novas aplicações.

## 2.2 Rastreabilidade Bovina

Rastreabilidade bovina é um sistema de controle de animais que permite sua identificação individual desde o nascimento até o abate, registrando todas as ocorrências relevantes ao longo de sua vida [4]. A rastreabilidade também existe para garantir ao consumidor um produto seguro, por meio do controle de todas as fases de produção, industrialização, transporte, distribuição e comercialização, possibilitando uma correlação entre o produto final e a matéria-prima que lhe deu origem[14].

O Brasil detém o maior rebanho comercial do mundo e é o maior exportador mundial. É ainda o segundo em quantidade de carcaça produzida, atrás somente dos Estados Unidos em volume produzido. A produção de equivalente carcaça aumentou mais de 20% enquanto as exportações quadruplicaram. Aproximadamente 140 países compram hoje a carne bovina brasileira [15]. A pecuária bovina desempenha, portanto, um papel de destaque na economia do país. Cada vez mais os consumidores, que estão se tornando mais exigentes, buscam por produtos de melhor qualidade. Adicionalmente, a preocupação com os aspectos relacionados à saúde e bem estar das pessoas, também tem aumentado consideravelmente. No caso específico das carnes, essa demanda acontece tanto pelos atributos intrínsecos de qualidade como maciez, sabor, quantidade de gordura, como também pelas características de ordem ou natureza voltadas para as formas de produção, processamento, comercialização, segurança sanitária e saúde animal.

O Ministério da Agricultura Pecuária e Abastecimento (MAPA) publicou a Instrução Normativa nº 1, de 9 de janeiro de 2002 [8], a qual institui o Sistema de Identificação e Certificação de Origem Bovina e Bubalina (SISBOV)<sup>1</sup>. Segundo a mesma, o SISBOV será gerenciado pela Secretaria de Defesa Agropecuária (SDA), que expedirá instruções complementares necessárias para a implementação do sistema brasileiro de rastreabilidade.

A definição de um sistema de rastreamento ocorreu em razão da exigência dos países importadores da Europa que requerem uma identificação precisa da procedência dos produtos que têm a carne bovina como componente. Tais medidas tentam restaurar a confiança dos consumidores dos países europeus no produto, depois da crise provocada pela encefalopatia espongiforme bovina (“doença da vaca louca”) e pela febre aftosa que afetaram gravemente o comércio e reafirmaram a necessidade de melhorar os métodos para o rastreamento de animais vivos e seus derivados, especialmente quando são objetos de intercâmbios comerciais de âmbito internacional. Na União Européia, foram promulgados textos legislativos para aplicar tais melhorias.

A identificação precisa da procedência dos produtos derivadas da carne requer métodos para identificar os animais em todos os estágios da cadeia produtiva, um dos métodos de identificação e a utilização da tecnologia RFID, que por sua vez abre um novo leque de serviços relacionados a automatização da cadeia produtiva da carne.

## 2.3 *Middleware* RFID

A adoção generalizada de RFID requer leitoras e *transponders* de baixo custo e também uma plataforma de software dedicada, que gerencie estas leitoras e processe uma grande quantidade de dados capturados.

Segundo Prabhu *et al*[21], um *middleware* RFID é uma nova geração de *software* que situa entre o *hardware* RFID (leitoras) e as aplicações convencionais.

Para Thornton *et al*[11], *middleware* RFID é um *software* que foi designado para

---

<sup>1</sup>Conjunto de procedimentos adotados para caracterizar a origem, o estado sanitário, a produção e a produtividade da pecuária nacional e a segurança dos alimentos provenientes dessa exploração econômica.



processar um fluxo de dados de uma *tag*, vindo de um ou mais dispositivos de leitura. O *middleware* realiza a filtragem, agregação, e contagem dos dados da *tag*, reduzindo, a priori, o volume de dados enviados.

Floerkemeier e Lampe [10] consideram *middleware* RFID uma “aplicação-agnóstica”<sup>2</sup>, que gerencia leitoras, filtra e agrupa os dados RFID capturados e os envia aos clientes apropriados.

Floerkemeier e Lambe [10] identificaram os seguintes requisitos para um *middleware* RFID:

- Disseminação dos dados RFID. As informações capturadas por um leitor são usualmente de interesse, não só de uma única aplicação, mas para um conjunto de aplicações da organização e dos seus parceiros comerciais. Portanto os dados capturados devem ser distribuídos entre as aplicações interessadas.
- Filtragem e agregação dos dados. Todas as aplicações que fazem uso de dados capturados desejam recebê-los agregados e filtrados. Isto porque diferentes aplicações estão interessadas em diferentes subconjuntos de dados capturados pelo leitor e como o volume de dados capturados aumenta se faz necessário agrupa-los de forma resumida para que aplicações legadas possam suportar este volume de dados.
- Leitura e escrita de *transponders*. Alguns tipos de *transponders* utilizam memórias de leitura e escrita para dados adicionais. Um *middleware* RFID deve fornecer meios para leitura/escrita nesta memória adicional.
- Integração dos leitores. Devido a grande proliferação de leitores, há a necessidade de um serviço que realize a configuração e o gerenciamento dos mesmos.
- Privacidade. O uso de *transponders* em aparelhos pessoais e roupas, implica em um monitoramento da vida pessoal dos consumidores, constituindo uma vigilância velada, ferindo desta forma os direitos individuais. Um *middleware* RFID deve possuir ferramentas para garantir a proteção dos dados coletados dos consumidores, como por exemplo o uso de criptografia.

Além dos requisitos apontados acima o *middleware* RFID deve apresentar, escalabilidade e desempenho.

Alinhado com os requisitos apresentados este trabalho visa desenvolver um *middleware* de uso geral para gerencia dados de um conjunto diverso de equipamentos que se comunicam com o computador via porta seriais.

Portanto um *middleware* é uma plataforma de *software* responsável por abstrair a conexão com uma estrutura física de captura de dados, capaz de filtrar, agrupar e formatar os dados lidos, de tal sorte que, a aplicação torne-se o mais independente possível dos equipamentos que compõem a estrutura física.

---

<sup>2</sup>Aplicações que não especifica quais aplicativos está associado, mas no lugar disso provê uma forma de poder se associar.

## 2.4 Balança de Passagem

Com a possibilidade da identificação eletrônica dos bovinos houve a demanda de associar esta informação com outras que permitissem avaliar o desempenho de cada animal. Não bastava que os *softwares*, de gerência de rebanho, recebessem apenas a identificação dos bovinos mas também saber sobre seu ganho de peso e todos os manejos que eles sofriam. Na unidade da Embrapa Gado de Corte de Campo Grande, Mato Grosso do Sul, foram desenvolvidos equipamentos para pesagem dos animais em marcha, os dados de pesagem e identificação coletados dos animais deveriam migrar para o computador<sup>[19]</sup> de forma automática, este sistema foi denominado Sistema de Balança de Passagem.

Na pesagem tradicional os animais são levados até o mangueiro, onde são identificados e pesados. Esta operação de levar os animais até um mangueiro, além de gerar um custo com mão de obra, ser muito trabalhosa ela também estressa os animais. O estresse em que os animais são submetidos pode refletir negativamente no ganho de peso dos mesmos. Por isso muitos produtores efetuam poucas pesagens. Como o acompanhamento do ganho de peso é precário, muitas vezes o produtor só vai saber que teve prejuízo, com alguns animais, na hora do abate.

Para um melhor acompanhamento do peso dos animais, sem o estresse do manejo tradicional, o pesquisador da Embrapa Gado de Corte, Dr. Pedro Paulo Pires<sup>3</sup>, propôs o sistema denominado “balança de passagem” ( figura 2.2 ). Neste sistema, a identificação e a pesagem dos animais são feitas no momento em que o animal vai beber água. Para ter acesso a água o animal deve passar por um corredor onde está instalado: - uma balança eletrônica, - e uma leitora RFID; desta forma toda vez que o animal fosse beber água o mesmo será identificado e pesado automaticamente.

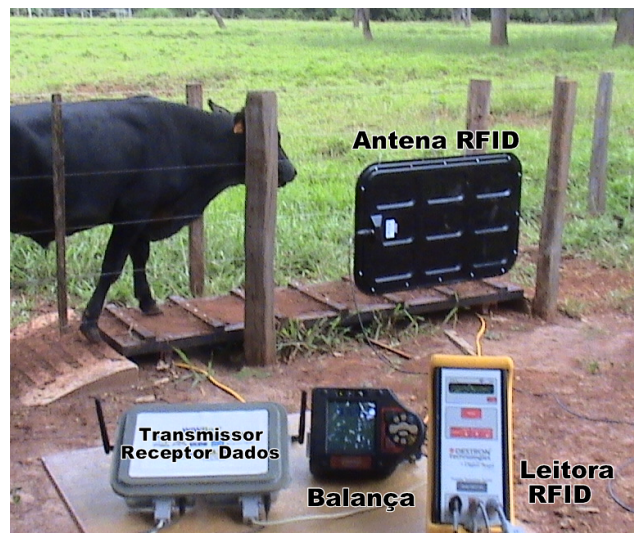


Figura 2.2: Balança de Passagem.

<sup>3</sup>Dr. Pedro Paulo Pires é Médico Veterinário Ph.D. e Pesquisador Embrapa Gado de Corte. Campo Grande, MS, Brasil

# Capítulo 3

## Análise do Problema

A pecuária de precisão é um novo paradigma de gerenciamento para a pecuária de corte, que se baseia na identificação automática dos animais. Ela também corresponde a uma área de pesquisa ampla e fundamentalmente multidisciplinar, requerendo sistemas que viabilizem a automação dessa nova pecuária. Juntamente com este novo paradigma encontra-se a necessidade de rastrear toda a cadeia produtiva da carne, afim de garantir aos consumidores produtos de qualidade. A tecnologia de identificação automática é bastante recente e vem evoluindo muito rapidamente, beneficiada pelos novos equipamentos e dispositivos de identificação.

### 3.1 Definição do problema

A pecuária de corte brasileira destaca-se como detentora do maior rebanho comercial e maior exportadora no cenário mundial[20]. Com aproximadamente 195 milhões de cabeças, o agronegócio da carne bovina tem crescido de forma expressiva, ao mesmo tempo em que tem se estruturado de forma competitiva, contribuindo para a formação do Produto Interno Bruto (PIB) do Brasil. Concomitante ao notório crescimento da pecuária brasileira, o mercado global está cada vez mais exigente, no que se refere à boa qualidade, com inocuidade, do produto final e as conseqüências ao meio ambiente por causa do manejo dos animais e da vegetação. Tais fatos foram desencadeados com a demanda pela sustentabilidade da cadeia produtiva, e foram intensificados com a descoberta da doença da vaca louca (Bovine Spongiform Encephalopathy BSE) em 1996, na Europa, além do constante risco que a febre aftosa apresenta aos rebanhos e, conseqüentemente, aos países produtores de carne.

Em adição e como conseqüência desses fatos e tendências, o mercado consumidor passou a exigir o rastreamento do alimento na cadeia produtiva, requerendo, ainda, que o processo seja transparente. Nesse contexto, a padronização de conceitos e, principalmente, de ações, constitui tema de importância estratégica para o País, que deve disponibilizar alimentos certificados, ou seja, com garantia de origem, de qualidade ambiental e de qualidade de bom produto.

Tais mudanças vêm sendo acompanhadas pela população devido ao maior acesso a programas de educação ambiental que resultaram em alterações importantes na forma de pensar o desenvolvimento econômico e a convivência com o ambiente. Com isto, surgiram novos fatores que interferem no critério de escolha e na diferenciação de produtos baseados no conhecimento da forma como o produto foi elaborado, com preferência crescente para aqueles produtos oriundos de cadeias produtivas ambientalmente corretas, socialmente justas e economicamente viáveis. O trinômio saúde-ambiente-preço passou a ser sempre considerado no momento da escolha de produtos, com tendência clara de favorecimento dos alimentos de boa aparência, de preferência, sem conservantes, produzidos sem agrotóxicos e sem risco para o ambiente e saúde.

Com base no exposto e visando ao desenvolvimento sustentável da cadeia produtiva da carne brasileira, sistemas de identificação, gerenciamento e rastreamento dos rebanhos vêm sendo desenvolvidos com a finalidade de oferecer aos produtores e mercado consumidor ferramentas que garantam a certificação do produto final.

## 3.2 Soluções apontadas

Visando atender a exigência da rastreabilidade da carne, por parte da Comunidade Europeia, o Ministério da Agricultura Pecuária e Abastecimento (MAPA)- instituiu o Sistema Brasileiro de Identificação e Certificação de Origem Bovina e Bubalina (SISBOV)[8]. Para implantar o SISBOV em seu sistema de produção e passe a ter o seu rebanho certificado[12], é exigido do pecuarista a identificação segura dos animais e de quase todas as funções do sistema de manejo do gado. Para identificação dos animais o SISBOV permite utilizar os seguintes dispositivos - brinco e marca de fogo, brinco e tatuagem, brinco e *botton*, brinco e dispositivos eletrônicos, ver figura 3.1.



Figura 3.1: Elementos de Identificação: brinco, marca a fogo, dispositivo eletrônico, tatuagem.

Para Pires *et al*[20], em se tratando dos sistemas de identificação animal, os identificadores clássicos, como tatuagens na face interna do pavilhão auricular, marcas a ferro quente e brincos numerados, têm sido os mais utilizados. No entanto, a diversidade de raças e manejos, além da constante ocorrência de erros na transcrição manual de dados, motivou a procura por métodos mais eficientes de identificação, uma vez que a revisão da numeração marcada com ferro quente na pele ou com tatuagem na orelha dos animais é, muitas vezes, invisível em animais de pelagem escura ou com excesso de pêlos, tornando-a excessivamente trabalhosa e ineficiente. Ademais, a marcação a ferro quente, em muitas

situações, quando mal localizada, danifica o couro do animal, produto de grande valor econômico. No caso dos brincos, tem ocorrido uma perda que varia de acordo com a raça e o tipo de ambiente entre 3% e 15%, anualmente, o que propicia erros no processo de rastreabilidade, além de perdas econômicas para o produtor rural. O brinco, quando mal colocado, além do processo traumático, pode causar solução de continuidade da pele e propiciar a instalação de infecções e miíases (bicheiras).

O sistema eletrônico de identificação animal, além de eliminar estas falhas e dificuldades, permite a automatização do manejo, realizados nos mangueiros.

### 3.3 O problema decorrente do uso da tecnologia RFID

A rastreabilidade usando dispositivos eletrônicos, leitoras e etiquetas, é apontada como a forma mais segura para a efetiva rastreabilidade, contudo a identificação dos animais usando dispositivos eletrônicos, aumentam os custos significativamente, quando comparado aos métodos menos tecnificados (ver Tabelas 3.1 e 3.2), desestimulando desta forma a sua implantação.

Tabela 3.1: Custo da identificação e rastreabilidade animal utilizando diferentes métodos em função da quantidade de animais, em reais (R\$).Fonte: Lopes[13]

Quantidade de animais		brbo	%	brmf	%	brta	%
100	Identificação	3,00	31,50	2,48	27,60	3,12	32,50
	Rastreabilidade	6,50	68,50	6,50	72,40	6,50	65,50
	Total	9,50	100,0	8,98	100,0	9,62	100,0
500	Identificação	2,84	43,50	2,06	35,70	2,22	37,50
	Rastreabilidade	3,70	56,50	3,70	64,30	3,70	62,50
	Total	6,54	100,0	5,76	100,0	5,92	100,0
1000	Identificação	2,82	45,70	2,00	37,50	2,11	38,60
	Rastreabilidade	3,35	54,30	3,35	62,50	3,35	61,40
	Total	6,17	100,0	5,35	100,0	5,46	100,0
5000	Identificação	2,80	47,70	1,96	38,90	2,02	39,70
	Rastreabilidade	3,07	52,30	3,07	61,10	3,07	60,30
	Total	5,87	100,0	5,03	100,0	5,09	100,0

br+bo: brinco e botton; br+mf: brinco e marca a fogo; br+ta: brinco e tatuagem.

Tabela 3.2: Custo da identificação e rastreabilidade animal utilizando diferentes métodos em função da quantidade de animais, em reais (R\$).Fonte: Lopes[13]

Quantidade de animais		<i>chip</i> intra	%	<i>chip</i> sub	%	bri ele	%
100	Identificação	11,00	62,8	10,91	62,6	11,50	63,8
	Rastreabilidade	6,50	37,2	6,50	37,4	6,50	36,2
	Total	17,50	100,0	17,41	100,0	18,00	100,0
500	Identificação	8,56	69,8	8,47	69,5	8,94	70,7
	Rastreabilidade	3,70	30,2	3,70	30,5	3,70	29,3
	Total	12,26	100,0	12,17	100,0	12,64	100,0
1000	Identificação	8,25	71,2	8,16	70,9	8,62	72,1
	Rastreabilidade	3,35	28,8	3,35	29,1	3,35	27,9
	Total	11,60	100,0	11,51	100,0	11,97	100,0
5000	Identificação	8,01	72,3	7,92	72,1	8,37	73,2
	Rastreabilidade	3,07	27,7	3,07	27,9	3,07	26,8
	Total	11,08	100,0	10,99	100,0	11,44	100,0

chip intra: chip intraruminal; chip sub: chip subcutâneo; bri ele: brinco e brinco eletrônico.

Num cenário como este, onde a rastreabilidade é essencial para a comercialização da carne bovina, o uso da tecnologia de RFID promoverá um fator competitivo, no entanto a opção de outras formas de identificação e a relação custo/benefício força os produtores a adotar sistemas de identificação não tecnificados. Por outro lado quando se opta pela adoção da tecnologia RFID, verifica-se que existem poucos sistemas que a utilizam para coleta de dados, principalmente na pecuária, para automatização do manejo dos animais.

# Capítulo 4

## *Middleware* Cerberus

Observando os problemas apresentados na seção anterior, a solução proposta foi desenvolver uma plataforma de *software* que ao mesmo tempo estimule o desenvolvimento de sistemas que utilizam RFID e facilite a integração desta nova tecnologia com os sistemas legados, provendo uma arquitetura que ajuda na criação de novas funcionalidades decorrentes do leque de aplicações que surgirão com a arquitetura RFID. Para fins de nomenclatura e identificação desta solução, o nome **Cerberus** esta sendo atribuído, ao *middleware* proposto.

Neste contexto o **Cerberus** apresenta-se como um *middleware*, cuja função é abstrair a conexão entre sistemas e os equipamentos utilizados no agronegócio, facilitando a integração de diversas tecnologias. O *middleware* proposto promoverá o desenvolvimento de novos sistemas que atendem a pecuária de precisão, possibilitando que novas tecnologias sejam mais facilmente incorporadas na produção de carne.

Para facilitar a visualização do contexto no qual esta solução está inserida, a Figura 4.1 apresenta uma visão de alto nível dos componentes participantes de uma solução genérica do domínio de RFID, na qual o **Cerberus** está inserido.

Na Figura 4.1, o *middleware* apresenta-se como uma camada intermediária entre as aplicações, sejam elas legadas ou não, e os equipamentos desenvolvidos para o agronegócio. Como definimos anteriormente segundo Floerkemeier [10], um *middleware* deve possuir os seguintes requisitos: prover a disseminação, filtragem e a agregação dos dados, ler e escrever no *transponder*, gerenciar leitoras e implementar algum nível de privacidade em relação aos dados capturados.

O *middleware* **Cerberus** apresenta duas partes principais, uma interface permitindo que a configuração do *middleware* possa ser realizada e testada de forma amigável. A segunda parte do *middleware*, que é executada em *background*, gerenciará os equipamentos e os dados de acordo com o arquivo de configuração. Como forma de interagir com outros aplicativos foram implementadas as seguintes formas de comunicação: troca de mensagem, compartilhamento de arquivo e compartilhamento de memória.



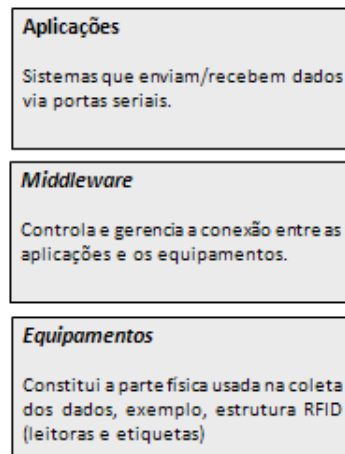


Figura 4.1: Visão funcional dos componentes. A transmissão de dados entre aplicação e os equipamentos é feita por meio do *middleware*.

## 4.1 Descrição do *Middleware*

O *middleware* Cerberus é composto de duas partes. Uma interface, que permite a configuração da plataforma de *software*, e o núcleo composto de três camadas: camada física, camada de gerenciamento, camada de integração (ver Tabela 4.1).

A interface é completamente independente do núcleo, e é através da interface que são realizadas as configurações do *middleware*, e estas configurações podem ser referentes a quaisquer camadas. Na camada física por exemplo, é configurado a conexão com os equipamentos, tais como: taxa de transferência e porta de recebimento dos dados. Na camada de gerenciamento são configurados os tipos de manipulação dos dados como por exemplo: tipo de filtragem dos dados, por leitora ou por número do *transponder*. Na camada de integração é configurado como se dará a integração com os outros aplicativos, seja por troca de mensagem ou compartilhamento de arquivo ou de memória.

A Tabela 4.1, mostra as três camadas que compõem o *middleware* **Cerberus**, abaixo fazemos uma breve descrição das mesmas.

Configuração	Camada de Integração	- troca de mensagem; - compartilhamento de arquivo; - compartilhamento de memória.
	Camada de Gerenciamento	- filtragem do dados; - agregação do dados; - formatação dos dados.
	Camada Física	- comunicação com os equipamentos.

Tabela 4.1: Camadas que compõem o núcleo do *middleware*.

A camada física é a responsável pela conexão com os dispositivos, ou seja, com a parte física, esta camada fica responsável por se conectar com os equipamentos através



das portas seriais RS232 , ou outro tipo de conexão. Para implementar esta camada utilizamos uma API <sup>1</sup> denominada RXTX [1] para comunicação com os dispositivos físicos através das portas seriais.

A camada de gerenciamento é a responsável por receber os dados vindos da camada física gerenciando-os de acordo com a configuração pré-estabelecida. Nesta camada, os dados são filtrados, agregados, e formatados antes de serem disponibilizados.

A camada de integração é a responsável por disponibilizar os dados para as demais aplicações, utilizando para isso o envio de mensagens, compartilhamento de arquivos ou memória.

## 4.2 Arquitetura

Esta seção apresenta a arquitetura do *middleware* **Cerberus**4.2, mostrando as diferentes camadas e suas responsabilidades.



Figura 4.2: Arquitetura do *middleware* **Cerberus**.

O conceito básico é que o *middleware* seja simples e rápido, visto que, o número de equipamentos ligados ao computador pode aumentar consideravelmente e estes podem

<sup>1</sup> Application Programming Interface(API) é um conjunto de funções e sub-rotinas usadas para ativar um determinado dispositivo.

fazer várias requisições ao *middleware*. As subseções seguintes apresentam com mais detalhes cada camada da arquitetura apresentada na Figura 4.2.

A figura 4.5 mostra o diagrama de classe completo do núcleo do *middleware* **Cerberus** e as demais classes que compõem o *middleware*. Para um melhor entendimento da arquitetura do *middleware* é apresentado uma descrição das camadas e o diagrama de sequência 4.6 mostrando o comportamento e as iterações dos objetos que implementam o núcleo do *middleware*.

### 4.2.1 Camada Física

A camada física faz parte do núcleo do *middleware* e é responsável pelo envio/recebimento dos dados para os dispositivos eletrônicos. Para a sua implementação foi utilizada a API denominada RXTX [1] que utiliza as portas seriais RS232 para a transmissão dos dados.

Para curtas distâncias, barramentos seriais estão se tornando cada vez mais comuns devido a sua simplicidade, sendo que uma das grandes dificuldades dos usuários são as configurações que devem ser feitas para que ocorra de forma correta a comunicação entre os dispositivos e os sistemas. Para facilitar a configuração dos equipamentos foi disponibilizado uma interface visual, figura 4.7.

A maioria das mensagens digitais é mais longas que alguns poucos bits. Por não ser prático nem econômico transferir todos os bits de uma mensagem simultaneamente, a mensagem é quebrada em partes menores e transmitida sequencialmente, portanto os dados que chegam devem ser agrupados ou divididos para formar uma mensagem, esta divisão ou agrupamento fica a cargo da classe *Connect*, que por sua vez instancia os objetos da API RXTX. (ver diagrama de classes 4.3). A classe *Connect* utiliza objetos da classe *SerialReader* para monitorar as portas seriais, capturando os dados enviados pelos dispositivos via portas seriais. As tabelas 4.2 e 4.3 mostram os métodos implementados na classe e a descrição dos mesmos.

A classe *SerialReader* implementa a classe *Runnable*, portanto é executada como uma *thread*<sup>2</sup> independente. Esta classe monitora entradas seriais da máquina, sua função é capturar os dados e agrupá-los ou dividi-los para formar uma informação a ser enviada para a classe *ManagerEventData*.

Nome	Parâmetros	Descrição
connect	ManagerEventData managerEventData, Integer index	Método que cria uma conexão entre os objetos e a parte física.
listPorts		Retorna uma lista de todas as portas seriais que estão sendo utilizadas.
stop		Interrompe a leitura/escrita e libera os recursos da máquina.

Tabela 4.2: Métodos da classe *Connect*.

A comunicação de dados digitais para dispositivos externos ao circuitos do computador apresentam via de regra um conjunto de parâmetros a serem configurados, como o nome

<sup>2</sup>Uma thread é uma unidade básica de utilização da CPU[22].

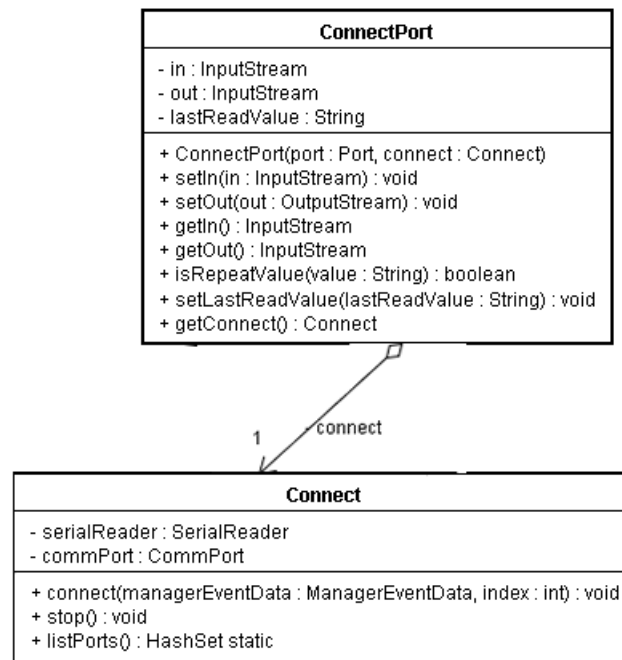


Figura 4.3: Diagrama de Classes que representa a camada física. Composto pelas classes **Connect** e **SerialReader**.

Nome	Parâmetros	Descrição
SerialReader	ManagerEventData managerEventData, ConnectPort connectPort, CommPort commPort	Construtor da classe.
run		Implementação do método abstrato da classe <i>Runnable</i> . Este método monitora as portas seriais, assim que uma informação é capturada esta é enviada para classe <i>ManagerEventData</i>

Tabela 4.3: Métodos da classe *SerialReader*.

da porta que esta sendo usada, a taxa de transmissão dos dados, caracter de finalização de mensagem etc. A seção 4.3 apresenta um resumo destes parâmetros e suas possíveis configurações.

### 4.2.2 Camada Gerenciamento

A camada de gerenciamento faz parte do núcleo do *middleware* e é responsável pela gerência das conexões do sistema com os dispositivos conectados as portas seriais. É a camada de gerenciamento que implementa os requisitos de *middleware* proposto por Floerkemeier e Lampe[10]. A funcionalidades implementadas no *middleware* **Cerberus** foram :

- Formatação dos dados. O *middleware* permite que o usuário defina quais partes da informação é útil para o sistemas, esta funcionalidade é importante pois alguns

dispositivos enviam juntamente com a informação lida a data e a hora de leitura. Um sistema que recebe somente como entrada um número deve ignorar as demais entradas, como data e hora, por exemplo.

- Filtragem dos dados. Para evitar processamento ou consulta a banco de dados desnecessárias o *middleware* permite definir uma lista valores que deverão ser ignorados e portanto não enviadas ao sistemas.
- Disseminação dos dados. A princípio a cada porta serial esta associada um único sistema, neste caso o *middleware* simula a conexão de vários sistemas a uma única porta, portanto a informação lida deve ser enviada a todos os sistemas que estejam ligados a esta porta serial.
- Leitura única. Para não sobrecarregar os sistemas com consultas e processamento desnecessários o *middleware* ignora as leituras repetidas.
- Agregação dos dados. Usando a configuração do *middleware* é possível agrupar as informações lidas por vários dispositivos em um único arquivo de saída.
- Checagem de erro (*Checkout error procedure*). Para leitura de *transponders* que utilizam o padrão EPC[7] é possível definir padrões na informação a serem verificados para validação da informação lida do *transponder*.

Estas funcionalidades estão implementadas na classe *ManagerEventData*, veja o diagrama de classes 4.4 mostrando detalhes da classe *ManagerEventData* juntamente com as classes que compõem a camada de gerenciamento.

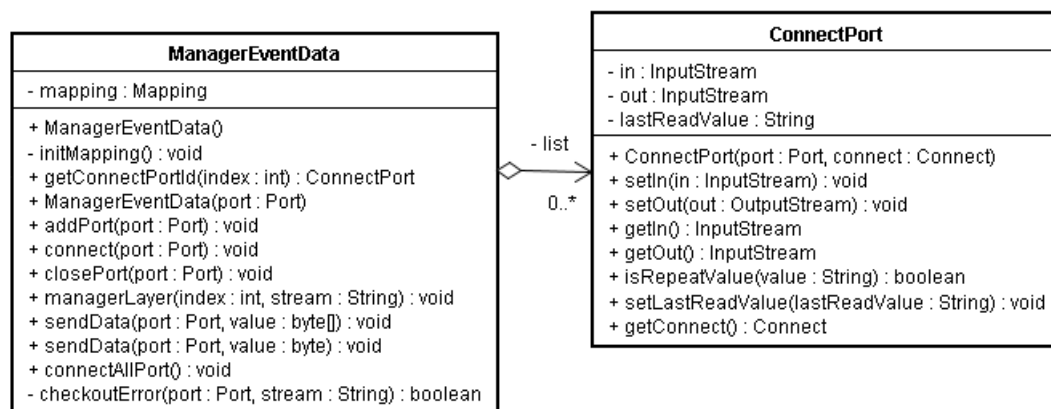


Figura 4.4: Diagrama de Classes que representa a camada de gerenciamento. Composto pelas classes: *ManagerEventData*, *ConnectPort*.

A classe *ManagerEventData* permite instanciar várias conexões ao mesmo tempo, ou seja, receber/enviar dados de vários dispositivos ligados as portas seriais. Assim a classe *ManagerEventData* implementa uma lista de objetos da classe *ConnectPort*, que permite estabelecer a conexão com os equipamentos cabendo a classe *ManagerEventData* gerenciar

as informações. Os métodos implementados pela classe *ManagerEventData* são mostrados na tabela 4.4.

Nome	Parâmetros	Descrição
ManagerEventData		Este construtor carrega o arquivo de configuração das portas e dos dispositivos a serem lidos.
ManagerEventData	Port port	Inicializa a leitura/escrita do objeto Port que foi passado como parâmetro.
addPort	Port port	Adiciona uma nova porta a lista para ser gerenciado.
close	Port port	Interrompe a leitura/escrita na porta que foi passada como parâmetro.
closeAllPort		Interrompe a leitura/escrita de todas as portas gerenciadas pela classe.
connectAllPort		Inicializa a leitura/escrita de todas as portas gerenciadas pela classe.
getConnectPortId	int index	Retorna a porta identificada pelo valor passado como parâmetro.
managerLayer	int index, String stream	A classe <i>SerialReader</i> envia uma mensagem a este método passando o identificador da porta e a mensagem lida para que este método possa manipular e enviar a informação lida para os devidos sistemas.

Tabela 4.4: Métodos da classe *ManagerEventData*.

O método *managerLayer* da classe *ManagerEventData* é invocado pela classe *SerialReader* a toda informação que chega a uma porta serial. É passado ao método *managerLayer* o identificador da porta e a informação lida do dispositivo ligado a porta serial, cabendo ao método *managerLayer* fazer o pré-processamento da informação e passa-la ao sistema interessado. O algoritmo 1 exibe um trecho do código fonte do método *managerLayer* 1.

A classe *ConnectPort* possui os objetos da classe *Port* responsável por armazenar todas as configurações da porta e o objeto da classe *Connect* responsável por armazenar as informações da conexão. A classe *ConnectPort* armazena informações lidas pelos dispositivos para verificar se houve repetição da leitura. A tabela 4.5 apresenta um resumo dos métodos implementados pela classe.

O algoritmo 1 é referente ao método *managerLayer* da classe *ManagerEventData* mostra neste método as informações lidas dos dispositivos via portas seriais e são formatadas (linha 5), extraindo a *substring* da posição *getStartFormat* até a posição *getEndFormat* definidas na configuração. Para *chips* RFID que utilizam o padrão EPC o *middleware* compara com os valores recebidos (linha 7) com um classe de valor esperado, se o valor lido for diferente o *middleware* descarta a leitura solenemente. O comando da (linha 10) verifica se é permitida a repetição da leitura e se a informação já foi lida, o comando da (linha 11) faz a filtragem dos valores, as linhas 21 e 22 fazem uma chamada ao método *setText* passando como argumento a informação lida, e a linha 24 inicializa uma *thread* em baixa prioridade para salvar as informações lidas no arquivo XML. Neste arquivo são agrupados os dados recebidos dos diferentes dispositivos conectados ao *middleware*.

---

**Algoritmo 1** Método managerLayer da classe ManagerEventData

---

**Entrada:** (1) Um *index* identificando a porta; (2) Uma *stream* contendo a informação lida.

**Saída:** Um arquivo contendo as informações lidas e a chamada ao método *setText*.

```

1: ConnectPort connectPort  $\leftarrow$  (ConnectPort)list.get(index);
2: Port port  $\leftarrow$  connectPort.getPort();
3: Method mtd  $\leftarrow$  null;
4: if stream.length()  $\geq$  (port.getEndFormat() – port.getStartFormat()) then
5:   stream  $\leftarrow$  stream.substring(port.getStartFormat(), port.getEndFormat());
6: end if
7: if checkoutError(port, stream) then
8:   return;
9: end if
10: if  $\neg$ (port.isReadUnique()  $\wedge$  connectPort.isRepeatValue(stream)) then
11:   if  $\neg$ port.isFilterValue(stream) then
12:     String time  $\leftarrow$  newSimpleDateFormat("dd/MM/yyyyHH : mm : ss").format(newDate(System.currentTimeMillis()));
13:     List < ConnectPort > listTemp  $\leftarrow$  new ArrayList < ConnectPort > ();
14:     listTemp  $\leftarrow$  listPort(port.getPortName());
15:     for ConnectPort connectTemp : listTemp do
16:       Port temp  $\leftarrow$  connectTemp.getPort();
17:       if port.getPortName().equals(temp.getPortName()) then
18:         Object obj  $\leftarrow$  port.getObjectRead();
19:         Class cls  $\leftarrow$  obj.getClass();
20:         try
21:           mtd  $\leftarrow$  cls.getMethod("setText", new Class[] {String.class});
22:           mtd.invoke(obj, new Object[] {stream});
23:         catch (Exception ex) ex.printStackTrace();
24:         (newWriteFile(newWriteReadValues(
25:           temp.getEquipmentName(), temp.getPortName(), stream, time))).run();
26:       end if
27:     end for
28:   end if
29: end if

```

---

Nome	Parâmetros	Descrição
ConnectPort	Port port, Connect connect	Construtor da classe.
getConnect		Retorna um objeto da classe <i>Connect</i> .
isRepeatValue	String value	Verifica se houve leitura repedida.
setConnect	Connect connect	Atribui um valor ao objeto connect da classe.
getPort		Retorna um objeto da classe <i>Port</i> .
setPort	Port port	Atribui um valor ao objeto port da classe.
setLastReadValue	String lastReadValue	Salva o último valor lido pelo dispositivo.
getIn		Retorna um objeto da classe <i>InputStream</i> .
setIn	InputStream in	Atribui um valor ao objeto in.
getOut		Retorna um objeto da classe <i>OutputStream</i> .
setOut	OutputStream out	Atribui um valor ao objeto out.

Tabela 4.5: Métodos da classe *ConnectPort*.

### 4.2.3 Camada Integração

Esta é a camada que permite que outros sistemas utilizem o *middleware Cerberus*, uma das formas de utilizar o *middleware* e configurando os dispositivos no arquivo de configuração *listports.xml* e iniciar a execução do *middleware*. Os dados serão salvos no arquivo definido na configuração, veja seção Configuração do *middleware*(4.3). Segue abaixo um exemplo de um arquivo de saída do *middleware*.

```
<?xml version="1.0" encoding="UTF-8"?>
<write-read-values>
<equipment-name>Leitora Destron</equipment-name>
<port-name>COM8</port-name>
<value>24/06/2009 16:34:36 985140000201565 23.5C / 74.3F </value>
<time>24/06/2009 16:34:36</time>
</write-read-values>
<write-read-values>
<equipment-name>Leitora Data Mars</equipment-name>
<port-name>COM7</port-name>
<value>985140000201565</value>
<time>24/06/2009 16:34:42</time>
</write-read-values>
```

Observe pela tag *port-name* que são duas portas seriais distintas e a tag *equipment-name* mostra que são dois dispositivos, uma leitora RFID *Data Mars* e uma leitora RFID *Destron*, a primeira lê somente o número do *transponder* enquanto a segunda lê o número do *transponder* a data e a temperatura em graus Celsius e em *Fahrenheit*. Considerando um *software* que foi desenvolvido para trabalhar especificamente com a leitora *Data Mars* a aquisição da leitora RFID *Destron* obrigaria a uma mudança no código fonte do *software*, por outro lado ao se utiliza o *middleware Cerberus* poder-se-ia configurar o *middleware* para extrair somente o número do *transponder* sem a necessidade de modificar o *software* para se adaptar a nova leitora RFID.

A classe *WriteFile* que estende da classe *Thread* salva os dados lidos pela classe *ManagerEventData* utilizando o padrão XML definido no ANEXO I. O objeto desta classe é executado como uma *thread* de baixa prioridade para não concorrer com as *threads* dos objetos *SeriaReader*, a tabela 4.6 resume os métodos implementados na classe.

Nome	Parâmetros	Descrição
WriteFile	WriteReadValues stream, Port port	Construtor da classe, recebe o objeto a ser salvo e o objeto port contendo informações como: nome do arquivo, da porta, do dispositivo etc.
run		Implementação do método abstrato da classe <i>Thread</i> . Este método salva as informações lidas nas portas seriais.

Tabela 4.6: Métodos da classe *WriteFile*.

A segunda opção é utilizar o pacote<sup>3</sup> **Cerberus** e instanciar um objeto da classe *ManagerEventData* passando como parâmetro para o construtor um objeto da classe *Port*, esta deve possuir um objeto que tenha um método com a seguinte assinatura *setText(String text)* (as classes *JTextArea*, *JLabel*, *JButton*, *TextField* são exemplos de classes que possuem o método *setText(String text)*), este método será invocado pela classe *ManagerEventData* passando para o método *setText* a informação recebida do dispositivo já pré-processada. O trecho de código abaixo mostra como instanciar o objeto da classe *ManagerEventData* 2.

---

**Algoritmo 2** Instanciando um objeto da classe *ManagerEventData*


---

```

1: JLabel transponder ← new JLabel();
2: Port port ← new Port("COM9", 9600);
   port.setObjectRead(transponder);
4: ManagerEventData manager ← new ManagerEventData(port);
   manager.connectAllPort();

```

---

Este trecho de código mostra (linha 1) a instancição do objeto *transponder* da classe *JLabel* que irá receber (linha 3) via o método *setText* os dados que foram lidos dos dispositivos ligados as portas seriais e processados pela classe *ManagerEventData*, na linha 4 e instanciado um objeto da classe *ManagerEventData* passando como parâmetro o objeto da classe *Port* e na linha 5 e realizado a conexão com os dispositivos.

## 4.3 Configuração do *middleware*

Esta camada fornece uma interface gráfica (GUI Graphical User Interface) 4.7 para configuração das portas seriais e dos dispositivos ligados a ela, sendo que uma das grandes dificuldades observadas refere-se a configuração dos equipamentos e das portas seriais aos quais estes estão conectados. Ocorrendo em muitos casos do usuário não saber o nome da porta na qual um dispositivo está conectada.

---

<sup>3</sup>Um pacote Java é um conjunto de classes que estão agrupados.



Na tela de configuração o usuário pode configurar as propriedades das portas seriais e as configurações de saída dos dados como: formatação, nome do arquivo de saída, nome do dispositivo, checagem de erro, etc., as configurações utilizadas no *middleware* estão listadas na tabela 4.7.

Ao abrir a tela de configuração é listada todas as portas que estão com algum dispositivo conectado a ela, o usuário deve escolher a porta e modificar os parâmetros de configuração, e nesta mesma tela visualizar os dados que estão sendo lidos de acordo com o arquivo de configuração, estes valores são mostrados na caixa de texto bem com os seus valores em hexadecimal.

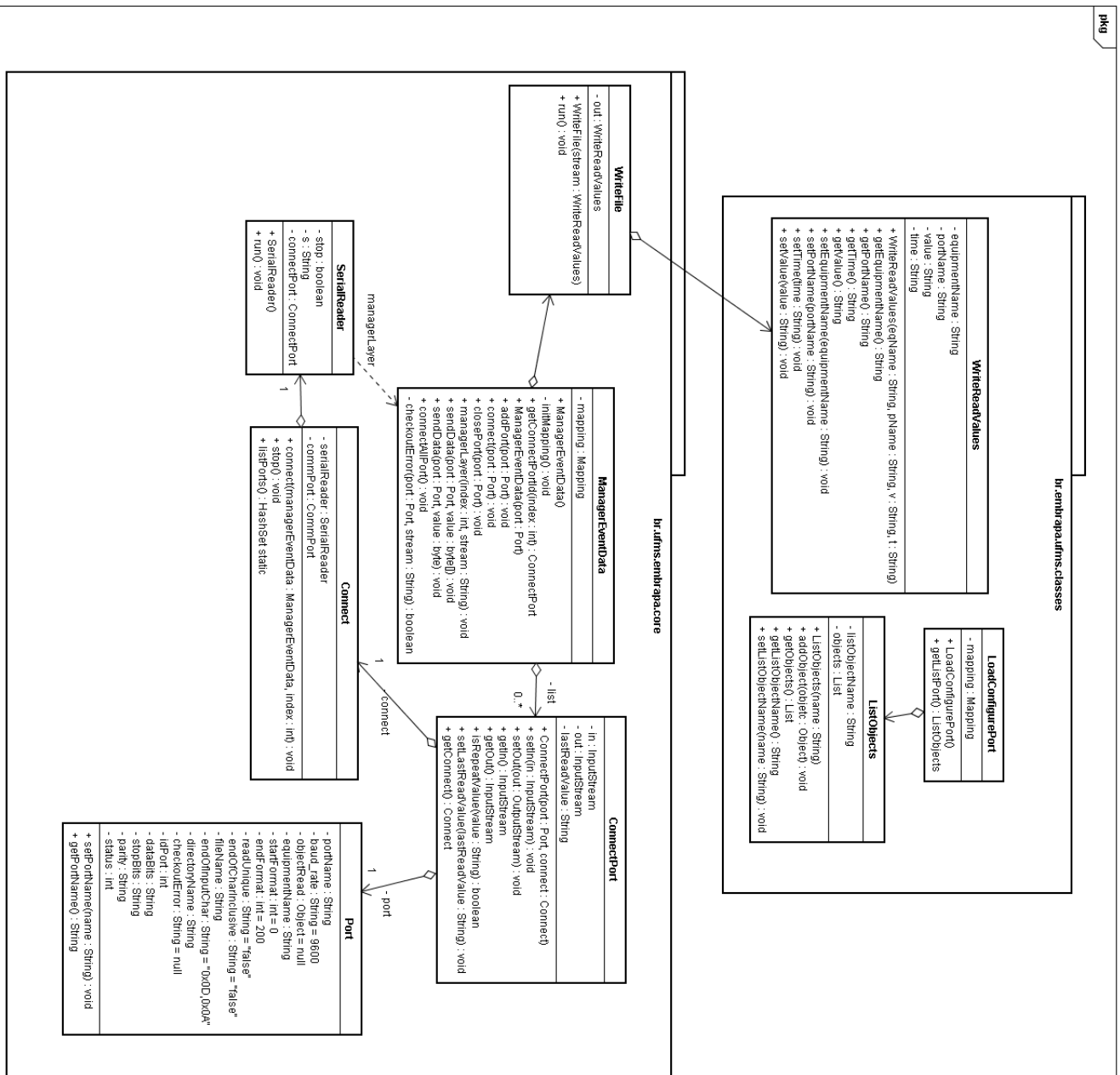
Outra maneira de configurar o *middleware* é editar diretamente o arquivo XML de configuração(ANEXO I), veja o exemplo do arquivo de configuração para uma leitora RFID ligada a porta serial COM7.

```
<?xml version="1.0" encoding="UTF-8"?>
<list-objects>
  <list-object-name>Ports</list-object-name>
  <objects xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="java:br.
    <equipment-name>Leitora RFID</equipment-name>
    <port-name>COM7</port-name>
    <baud_rate>9600</baud_rate>
    <data-bits>8</data-bits>
    <stop-bits>1</stop-bits>
    <parity>PARITY_EVEN</parity>
    <start-stream>0</start-stream>
    <end-stream>100</end-stream>
    <EndOfInputChar>0x0D,0x0A,0x7A</EndOfInputChar>
    <EndOfCharInclusive>false</EndOfCharInclusive>
    <read-unique>false</read-unique>
    <filter-values>99999</filter-values>
    <file-name>inforfid.xml</file-name>
    <directory-name>E:</directory-name>
    <checkout-error>null</checkout-error>
  </objects>
</list-objects>
```

A tabela 4.7 mostra cada uma das *tags* usadas e sua descrição.

<b>Tag</b>	<b>Valor default</b>	<b>Descrição</b>
<i>equipment-name</i>		Nome do equipamento ligado a porta serial.
<i>port-name</i>		Nome da porta serial. (COM1, COM2, COM3 etc).
<i>baud-rate</i>	9600	A taxa de transferência é a velocidade na qual os dados são enviados através de um canal
<i>data-bits</i>	8	Quantidade de <i>bits</i> por dado.
<i>stop-bits</i>	1	Bit que marca o fim de transmissão e tempo para o receptor reiniciar.
<i>parity</i>	PARITYNONE	Bit adicionado ao pacote de dados com o propósito de detecção de erro.
<i>start-stream</i>	0	Valor usado para formatar a informação lida.
<i>end-stream</i>	100	Valor usado para formatar a informação lida.
<i>EndOfInputChar</i>	0x0D,0x0A,0x7A	Caracteres que marca o fim de uma informação válida.
<i>EndOfCharInclusive</i>	false	Marca se o bit de parada faz parte da informação.
<i>read-unique</i>	true	Marca se pode ou não ocorrer a leitura repetida de um mesmo valor.
<i>filter-values</i>	null	Indica um conjunto de valores a ser ignorado pelo <i>middleware</i> .
<i>file-name</i>	info.xml	Nome do arquivo de saída.
<i>directory-name</i>		Nome do diretório de saída.
<i>checkout-error</i>	null	Verifica se houve leitura errada.
<i>flowcontrol-mode</i>	FLOWCONTROLNONE	Controle do fluxo de leitura.

Tabela 4.7: *Tags* de configuração do *middleware*.



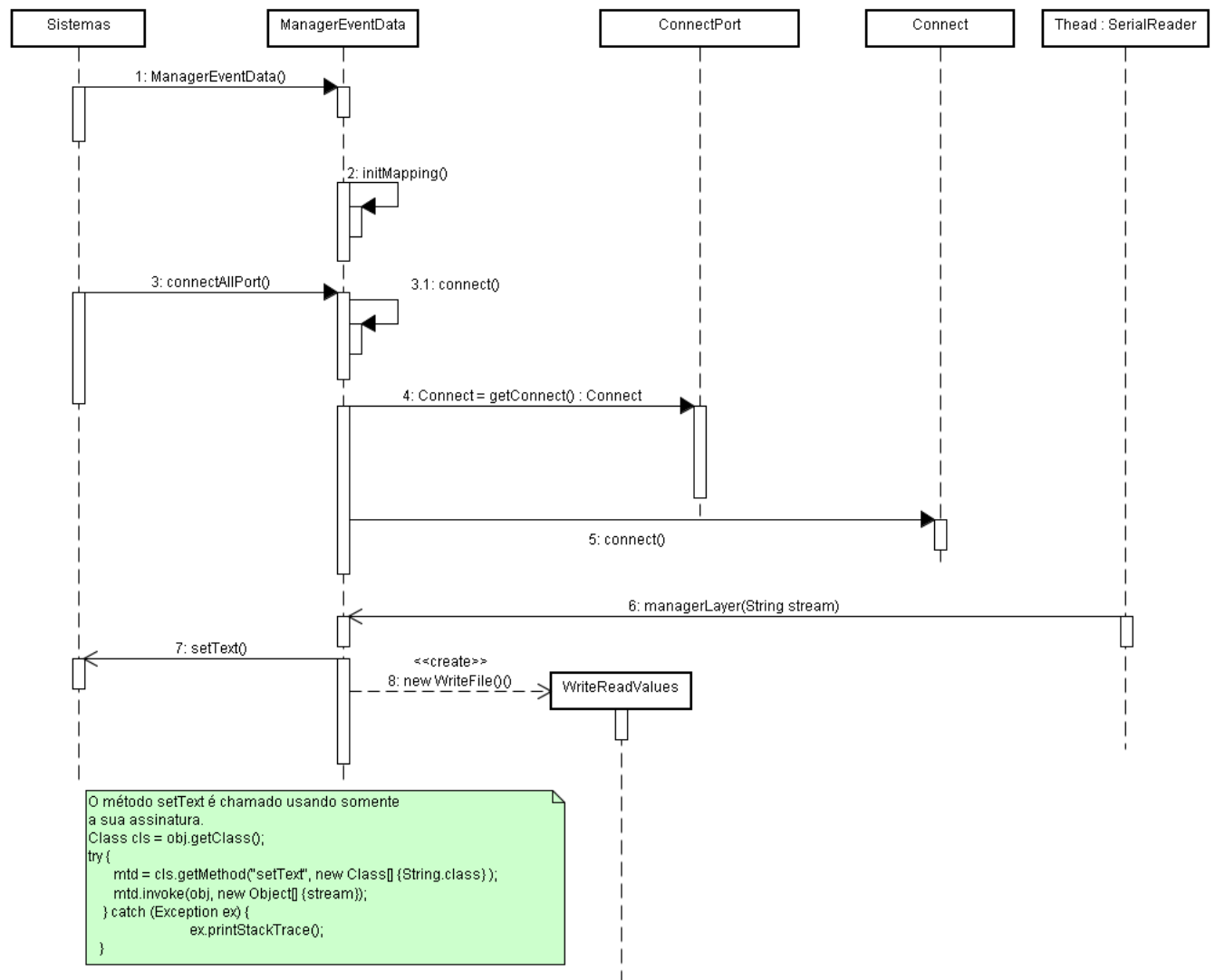


Figura 4.6: Diagrama de sequência das classes que compõem o núcleo do *middleware* Cerberus

**Tela de Configuração**

Wizard de Configuração do Middleware Cerberus

Nome da Porta : COM7

Nome do Arquivo de Saída : info.xml

Taxa de Transferência : 9600

Paridade : None

Caracteres de finalização :

Nome do Dispositivo :

Nome do Diretório : c:\

Data Bit : 8 Bits

Stop Bit : 8 Bits

Start : 0

End : 100

Incluir caracter de inform. ☒ Sim ☐ Não

Leitura Única ☐ Sim ☐ Não

Valores Filtrados :

Checagem de Erro :

Controle de Fluxo :

Saída :

Parar a Leitura da Porta

9	0	0	0	0	4	0	0	0
0x39	0x30	0x30	0x30	0x30	0x34	0x30	0x30	0x30
9	8	7	6	6	\n	\r		
0x39	0x38	0x37	0x36	0x36	0x0D	0x0A		

Figura 4.7: Tela de configuração do *middleware* Cerberus.

# Capítulo 5

## Estudo de Caso

Para validar o *middleware* **Cerberus** e avaliar sua viabilidade na instanciação de sistemas que utilizam as portas seriais para entrada de dados, foi implementado um estudo de caso que visa o registro diário e automático do peso dos animais.

Neste estudo de caso o *middleware* envia/recebe dados da leitora RFID e da balança, mostrando que o *middleware* **Cerberus** é útil para a instanciação de aplicações que utilizam diferentes dispositivos para captura de dados, não centrando somente na estrutura RFID.

Esta é uma das diversas aplicações vislumbradas com a utilização da tecnologia RFID, e tem como objetivo o acompanhamento do ganho de peso dos animais, servindo como uma ferramenta de apoio na tomada de decisão com relação ao manejo dos animais.

### 5.1 Sistema de Balança de Passagem

Sistema de Balança de Passagem é um modelo idealizado pelo Dr. Pedro Paulo Pires <sup>1</sup> para realizar a identificação e a pesagem de animais em marcha, de forma automática, isto é, sem intervenção humana, portanto sem estresse para os animais. Neste modelo é possível a identificação/pesagem dos animais diariamente, sem o trabalho de conduzi-los ao mangueiro.

A Balança de Passagem consiste de uma entrada afunilada por cerca, formando um corredor onde são instalados os seguintes componentes: leitora RFID e uma balança de pesagem de bovinos. O afunilamento serve para conduzir os animais sobre uma plataforma de madeira ( 250x60 cm ), apoiada sobre células de pesagem. A leitora RFID serve para identificar o animal que está passando pelo corredor e as células de pesagem para registrar a variação de peso. Logo, a cada acesso ao bebedouro/ração o animal é pesado e identificado, e estas informações são enviadas para um *software* por meio do *middleware* proposto. Este *software*, denominado Sistema de Balança de Passagem, é responsável por

---

<sup>1</sup>Dr. Pedro Paulo Pires é Médico Veterinário Ph.D. e Pesquisador Embrapa Gado de Corte. Campo Grande, MS, Brasil.

relacionar o animal com seu peso, validar o peso enviado e registrar as informações em um banco de dados.

### 5.1.1 Desafios

Os primeiros experimentos realizados com a balança de passagem, setembro de 2002, mostrou que foi registrado o peso de somente 10% dos animais. Este baixo índice de pesagem deve-se às falhas listadas abaixo:

- Falha na identificação dos animais. Este tipo de ocorrência se dá por falhas de equipamentos, ou por sobreposição da leitura dos *transponders* devido a passagem de mais de um animal, ao mesmo tempo, no corredor da balança;
- Falha na pesagem dos animais. Para pesagem eletrônica é exigido um tempo mínimo para estabilização do peso; assim a falha ocorrerá se a passagem do animal sobre a balança for muito rápida, ou se o animal não subir completamente na balança ou ainda dois animais subirem ao mesmo tempo na balança;
- Falha na associação do animal com seu peso. Esta ocorre quando o animal identificado e o peso registrado não pertencem ao mesmo animal;
- Os animais levam um período de quatro dias de adaptação para passarem espontaneamente sobre a balança;
- O custo de implantação do sistema ( leitora/*tags* RFID e balança eletrônica ) ainda é alto para os padrões da pecuária nacional.

Sendo que estes dois últimos pontos são solucionados com o tempo, sendo que o primeiro requer um tempo necessário para adaptação do animal e o segundo para que as novas tecnologias e a competitividade leve a redução dos custos da tecnologia RFID.

### 5.1.2 Solução Proposta

A solução proposta visa melhorar o índice de pesagem dos animais, tornando o sistema viável, para isso foi utilizado o *middleware* proposto para implementar o Sistema de Balança de Passagem, nessa nova implementação os pesos são registrados independente estabilização da balança.

O Sistema de Balança de Passagem segue o protocolo da balança de pesagem Coimma KT40 (ANEXO II), seguindo este protocolo o sistema envia/recebe dados da balança. Teste com animais passando sobre a balança, mostraram os resultados apresentados na Tabela 5.1.

Neste teste o Sistema de Balança de Passagem ficou fazendo requisições intermitentemente durante uma hora, observamos que os picos que aparecem no gráfico correspondem ao peso máximo atingido pela balança no momento da passagem do animal, e este peso

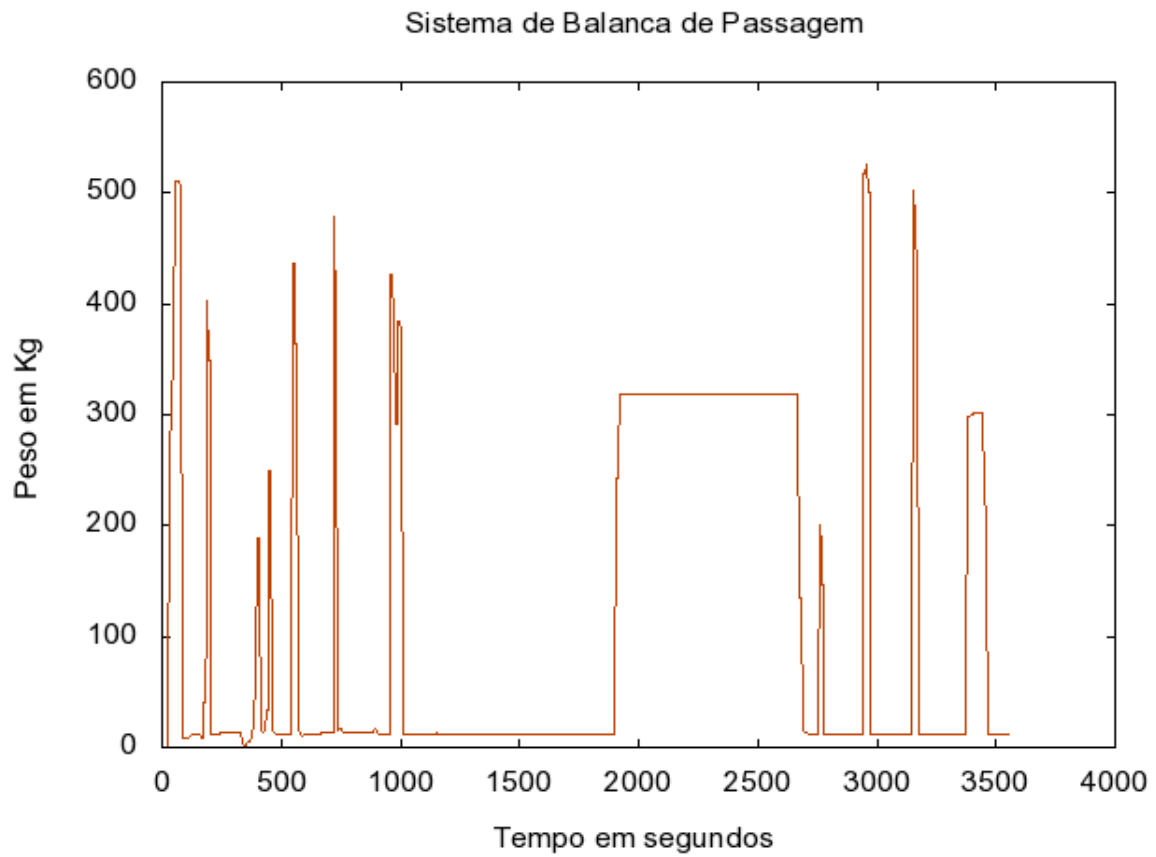


Figura 5.1: Gráfico obtido da tabela[9.1]: variação dos pesos, obtido pelo Sistema de Balança de Passagem.

corresponde aproximadamente ao peso do animal. Analisando os dados do gráfico observamos que no momento em que o tempo atinge  $t=500s$ , ocorreram três picos próximos. Esse fato ocorre em situações em que o animal não sobe totalmente na balança, ou dois animais sobem na balança, para o intervalo de tempo entre  $t=2000$  e  $t=2500$  temos uma situação ideal, pois o animal permaneceu parado na balança o tempo suficiente para a mesma se estabilizasse.

Neste teste observamos que a contínua requisição do peso que o sistema faz ao dispositivo da balança, acarreta os seguintes problemas: sobrecarga no *hardware* da balança, impedindo o seu perfeito funcionamento e diminuição da autonomia de energia da balança.

Para não sobrecarregar o *hardware* da balança foi adotado o seguinte procedimento, a requisição de pesos da balança é ativada pela leitora RFID. Portanto, o Sistema de Balança de Passagem recebe dados vindos da leitora RFID e recebe/envia dados para a balança, no momento em que a leitora RFID envia um dado para o software (significa que tem um animal passando sobre a balança), o software envia um pedido para a balança solicitando o peso (a princípio o *software* foi configurado para fazer dez requisições de peso para a balança), dentre os vários pesos enviado para o *software* e escolhido o maior. Este peso será registrado, somente se, este corresponder ao peso do animal, levando em



conta o desvio padrão do peso real do animal, e a expectativa do ganho de peso diário.

### 5.1.3 Resultados

Os resultado obtidos foram excelentes, considerando que a identificação e a coleta de peso foi de cem por cento, não levando em conta que nem toda a pesagem corresponde a um peso válido, cabendo ao *software* fazer a validação do valor obtido.

Comparado ao sistema tradicional de pesagem dos animais, a desconsideração de um grande número de pesagens, seria muito maior, visto que no sistema tradicional o número de pesagens variam em torno de duas ao ano. Em um sistema de pesagens diárias teríamos dados suficientes para estimar o peso real do rebanho com uma margem mínima de erro. A tabela abaixo mostra a média dos pesos coletados no intervalo de nove dias.

Chip intra-ruminal	07/05	12/05	13/05	14/05	15/05	19/05	20/05	21/05	28/05
76000000007578	495,0	576,0	467,3	499,0	482,0	462,0	502,8	473,5	0,0
76000000007612	317,5	597,5	517,0	0,0	0,0	502,0	524,0	0,0	0,0
76000000007606	0,0	369,0	541,8	544,5	488,3	517,3	467,3	484,5	534,0
76000000007775	0,0	449,0	500,4	518,3	520,5	441,8	495,2	498,5	504,0
76000000007644	494,5	449,5	492,0	0,0	493,5	487,0	0,0	0,0	0,0
76000000007714	0,0	544,0	575,8	469,0	539,5	545,0	546,0	542,0	534,0
76000000007324	393,0	427,0	420,0	0,0	0,0	420,3	388,0	0,0	0,0
76000000007072	0,0	425,0	368,0	0,0	0,0	496,0	426,5	534,0	0,0
76000000007132	0,0	553,7	549,0	528,0	547,5	547,0	505,0	541,5	493,5

Tabela 5.1: Tabela das médias dos pesos diários, calculado das tabelas (9.3,9.4,9.5,9.6,9.7,9.8). Peso em kg.

A tabela 5.1 mostra a média dos pesos brutos, isto é, não foi realizado nenhuma filtragem em relação ao peso coletado pela balança, a única manipulação dos dados feita foi: pegar somente os picos da balança, como foi dito anteriormente e não computar na média os animais que não foram pesados naquele dia, ou seja, desconsiderar os pesos iguais a zero.

Para facilitar a compreensão dos dados da tabela 5.1, foi calculado o desvio padrão das médias diárias, usando a fórmula 5.1.3 com as seguintes modificações. A média  $\bar{x}$  é a média diária da tabela 5.1, o  $n$  da fórmula representa o número de médias maior que 0 e para a variável *peso* da fórmula foi utilizado o maior peso registrado com o animal parado 9.3.

Fórmula do desvio padrão, adaptada.

$$\sigma = \frac{\sum_{i=1}^n \sqrt{(peso - \bar{x}_i)^2}}{n} \quad (5.1)$$

Abaixo segue o resultado obtido com a aplicação da formula 5.1.3, usando a média das pesagens diárias da tabela 5.1 e dos pesos do dia 19/05 [9.3]. Observamos que a maior diferença de peso foi de 28,2kg, e se calcularmos a média do desvio padrão das médias diárias obteremos o valor de 19,2kg,  $\bar{\bar{x}} = 19,2$ ;

$$\sigma = [12,6; 44,1; 28,7; 19,0; 9,8; 10,2; 9,4; 28,2; 11,0]$$

$$\bar{\bar{x}} = \frac{12,6 + 44,1 + 28,7 + 19,0 + 9,8 + 10,2 + 9,4 + 28,2 + 11,0}{9};$$

Como estes dados foram calculados em cima de valores brutos, sem nenhum pré-processamento, pode se dizer que é possível diminuir a diferença entre o peso coletado pelo Sistema de Balança de Passagem e o peso real do animal.

Para diminuir esta diferença o *software* deve levar em consideração o histórico de pesagem de cada animal individualmente, e satisfazer os pré-requisitos abaixo, a fim de aumentar o grau de confiabilidade, nos resultados obtidos.

Diante destes desafios o sistema de balança de passagem, proposto neste trabalho, deve satisfazer alguns pré-requisitos para se obter um alto grau de confiabilidade, nos resultados obtidos.

1. O sistema deve ser alimentado inicialmente com o peso real dos animais.
2. Deve ser colocado um ganho médio de peso diário esperado por animal.
3. Cadastrar uma margem de erro inicial para registro de pesos, considerando os cálculos anteriores um valor possível seria 20kg inicialmente.
4. Regularmente deve ser feita uma pesagem assistida, para a calibração do sistema.

A partir destes pré-requisitos o sistema deve, automaticamente registrar os pesos dos animais, descartando as pesagens que estão fora da margem de erro, emitir alertas para animais que não estejam ganhando peso, e a toda nova pesagem assistida recalculer os índices de pesagem: ganho de peso diário e margem de erro.

# Capítulo 6

## Trabalhos Relacionados

Esta Seção apresenta três trabalhos com a mesma linha de pesquisa do nosso trabalho (ver quadro comparativo<sup>1</sup> 6.1). Estes trabalhos serviram como ponto de referência para o desenvolvimento do **Cerberus**.

O conceito de infraestrutura de rede distribuída RFID foi inicialmente proposto por Auto-ID Center[7], uma indústria patrocinadora do programa de pesquisa para estimular a adoção de RFID, além de ser a responsável pelo termo EPC Network.

A visão do Auto-ID Center é a identificação única e universal de itens. O número único, chamado EPC<sup>2</sup> (*electronic product cod*) será codificado em RFID. EPC Network também tornará pública (via internet e por autorizações), outras informações referentes a um dado item. O EPC Network desenvolveu um componente de *middleware* chamada de Savant.

Savant é um *software* designado para processar o fluxo de dados da *tag* vindo de um ou mais leitores. Savant realiza filtragem, agregação, e contagem dos dados, reduzindo o volume de dados enviado para a aplicação. Este é definido em termos de “Módulos de Processamento”, ou “Serviços” e cada qual providência um conjunto específico de características, e pode combinar outras dependendo das necessidades dos usuários e/ou aplicações.

Prabhu [21], apresenta um *middleware* denominado WinRFID. Ele é um *middleware* RFID desenvolvido em multi-camadas, usando a tecnologia .NET.<sup>3</sup>

Há cinco camadas principais. A primeira camada lida com o *hardware* de leitores, *tags* e outros sensores. A segunda camada abstrai os protocolos de leitura. Acima destas camadas está a camada de processamento de dados, que lida com o processamento do fluxo de dados gerado pelos leitores. A quarta camada constitui um *framework* XML para representação de dados e informações. No topo encontra-se a camada que lida com

---

<sup>1</sup>As células em branco significa que o item em questão não foi comentado pelo autor do texto pesquisado.

<sup>2</sup>Atualmente dominado pela tecnologia de código de barras.

<sup>3</sup>Tecnologia ponto Net do inglês *dot Net*, é uma plataforma de *software* desenvolvida pela Microsoft para executar aplicações *Web*.

a representação dos pré-requisitos dos usuários finais.

RFIDStack é um outro *middleware* RFID baseado nos conceitos apresentado no trabalho de Floerkemeier e Lampe [10], este está focando não só as necessidades das aplicações mas também os obstáculos impostos pela tecnologia RFID. Como forma de diminuir a quantidade de informação recebida pelos leitores, RFIDStack implementa dois tipos de manipulação de dados, filtragem e agrupamento.

Middleware	Estrutura do software	Manipulação dos Dados	Interface física	Interface aplicativos	Configuração
WinRFID[21]	Estruturado em 5 camadas: - física; - protocolo; - processamento de dados; - <i>framework</i> XML; - representação de dados.	- Filtra; - Limpa; - Agrupa.	Utiliza os protocolos RS232, RS484, TTL e Ethernet.	Uso de serviços Web, como: <i>Windows Service</i> , <i>Web Service</i> e objetos remotos.	Possui um <i>wizard</i> para gerenciar as configurações.
RFIDStack[10]		- Filtra; - Agrupa; - implementa algum nível de privacidade.	Utiliza os protocolos RS232, RS484, TTL e Ethernet.	Utiliza um sistema de mensagem, responsável por disponibilizar as informações.	
Savant[7]	Definido em "módulos de processamento" - Módulo de processamento definido pelo usuário; - Módulo de processamento padrão.	- Filtra; - Agrupa; - Contagem dos dados.	Permite a conexão com um conjunto de leitoras de acordo com o ReaderProtocol 1.0.	Implementa um sistema de mensagem denominado MTB ( <i>Messaging/Transport Binding</i> ), responsável por disponibilizar as informações para as aplicações.	Usuários definem novas funcionalidades através de APIs definidas no módulo de processamento.
Cerberus	Definido em 3 camadas: - física; - gerenciamento; - integração.	- Formatação; - Filtragem; - Disseminação; - Leitura única; - Agregação; - Checagem de erro.	Utiliza o protocolo RS232.	Utiliza um sistema mensagem, responsável por disponibilizar a informação. E arquivo XML.	Possui um <i>wizard</i> para auxiliar o usuário nas configurações.

Tabela 6.1: Tabela comparativa entre os *middlewares* pesquisados e o **Cerberus**.

A interface de comunicação com outros aplicativos é comum a todos estes *middleware*, ela ocorre através de envio/recebimento de mensagens utilizando serviços *web*, como: *Web Service*, *Windows Service* e objetos remotos.

O trabalho apresentado difere dos demais por não centrar somente em uma infraestrutura RFID, mas em um conjunto de dispositivos físicos que envia/recebe dados do computador através das portas seriais. O desenvolvimento de um *middleware* capaz de receber e gerenciar dados de diferentes dispositivos permitirá o desenvolvimento de novos aplicativos que utilizam as portas seriais para enviar e receber dados.

# Capítulo 7

## Conclusões e Trabalhos Futuros

### 7.1 Contribuições da Dissertação

Com o objetivo de integrar a cadeia produtiva da carne bovina, a Universidade Federal do Mato Grosso do Sul juntamente com a Embrapa Gado de Corte desenvolveram uma plataforma Web denominado e-SAPI bovis. O e-SAPI bovis é um sistema de planejamento, gestão e avaliação de informações relativas à produção pecuária bovina nacional, com a proposta de assegurar confiabilidade e agilidade no acompanhamento de informações relativas à produção por parte de todos os atores envolvidos na cadeia produtiva da carne bovina.

Para compreender a importância da automatização da coleta de informações que tange a cadeia produtiva da carne bovina vale ressaltar que esta vem tornando cada vez mais estruturada e consolidada devido a motivação dada pela economia de escala ao intensivo uso e aplicação das Tecnologias de Informação e Comunicação (TIC) na rastreabilidade de alimentos. Para permitir a integração das informações de diferentes parceiros da cadeia de carne bovina (produtor, frigorífico, agência de vigilância animal, entre outros) foi desenvolvida uma plataforma Web de alta usabilidade intitulada “e-SAPI bovis”, que tem como objetivo viabilizar a integração nacional das informações de bovinos e bubalinos e prover a gestão de qualidade e rastreabilidade destes animais.

Num contexto mais amplo o *middleware* **Cerberus** apresenta-se como uma ferramenta que integra os elos desta cadeia ao prover mecanismo de automatização de coleta de dados e a padronização destas informações usando o padrão XML para alimentar o portal “e-SAPI bovis”, a figura 7.1, contextualiza o *middleware* frente ao portal e-Sapi, desenvolvido para agregar as informações da cadeia produtiva da carne.

O Sistema de Balança de Passagem desenvolvido no caso de uso servirá não só para tomada de decisão dos pecuaristas mas, também como um ferramenta de auxílio para os pesquisadores que precisam medir o ganho de peso dos animais utilizado nos experimentos sendo que a constante pesagem dos animais na forma tradicional acaba por influenciar nos resultados dos experimentos, com a utilização do sistema de balança de passagem a pesagem dos animais será realizada de forma automática e sem influenciar na variação do

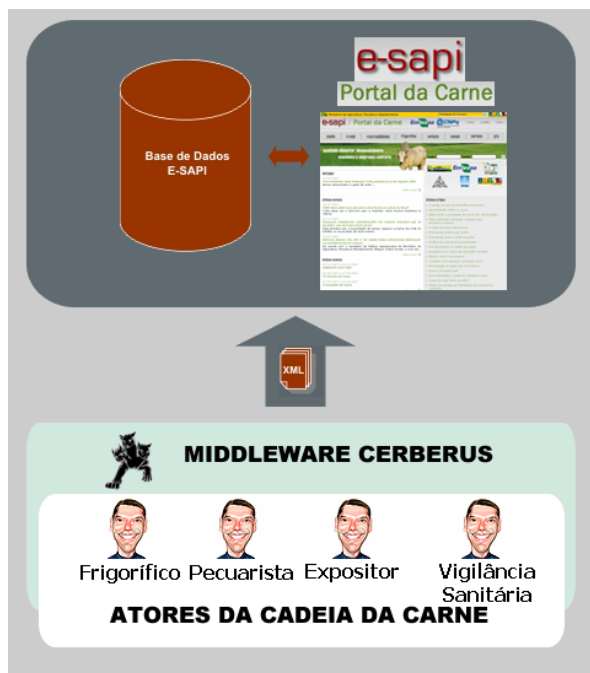


Figura 7.1: Contextualização do *middleware* **Cerberus** frente ao portal *e-Sapi bovis*.

ganho de peso dos animais.

## 7.2 Conclusões

A utilização da tecnologia de RFID é uma realidade que será vivida cada dia mais por todos os setores da cadeia produtiva. Motivada principalmente pela definição de um padrão eficiente de rastreabilidade, e a adoção de tecnologias de *softwares*, que darão suporte a infraestrutura RFID, será praticamente uma exigência do mercado para os próximos anos. Desta forma, cada vez mais será necessário habilitar e disponibilizar soluções que atendam as necessidades atuais e reais do agropecuária nacional, para que a adoção da tecnologia seja feita de forma consistente e que não traga incertezas e mitos sobre a eficiência desta tecnologia.

Outro fator que funcionará como catalisador da tecnologia RFID será a disponibilização de serviços que justifiquem o seu uso, como e o caso por exemplo da balança de passagem, estes serviços devem contribuir com a qualidade e a facilidade no manejo dos animais.

Hoje existem vários projetos que utilizam a tecnologia RFID, no entanto estes estão focados na distribuição e compartilhamento das informações oriundas das leitoras RFID, não se preocupando diretamente em facilitar o seu uso, principalmente em um dos elos mais importantes da cadeia produtiva da carne, o produtor rural, onde o grau de instrução ainda é baixo, para preencher esta lacuna o *middleware* **Cerberus** apresenta uma tela de configuração que auxilie aos usuários desta tecnologia.

O custo da tecnologia esta decrescendo, mas ainda é uma tecnologia que possui custo elevado, principalmente para os patamares brasileiros. Desta forma para que novos projetos possam ser viabilizados foi apresentado uma plataforma de *software* que incentive o desenvolvimento de novos sistemas que utilizem a tecnologia RFID, e uma aplicação que justifique os seus custos.

## 7.3 Extensões

As extensões sugeridas neste trabalho seguem duas linhas, uma centrada no *middleware* e a outra na balança de passagem. As duas primeiras sugestões são relativas ao *middleware* e a última relativa ao Sistema de Balança de Passagem.

### Uso do padrão USB<sup>1</sup>

A conexão com aparelhos que utilizam as portas USB ampliaria a quantidade de dispositivos que podem utilizar o *middleware* para gerir os dados coletados. Os computadores atuais não apresentam mais as portas seriais, dispositivos que possuem usam as portas seriais para enviar/receber dados do computado precisam utilizar equipamentos que emule portas seriais utilizando portas USB via *software* e *hardware*. A leitura direta nas portas USB eliminaria a utilização destes equipamentos que emulam as portas seriais.

### Utilização de *Web-Services*<sup>2</sup>

Para tornar transparente o envio de informações para o portal e-SAPI é sugerido a implementação de serviços Web que alimentam o portal com os dados coletados dos diferentes dispositivos.

### Explorar o uso de Inteligência Artificial

Visto que a grande quantidade de variáveis que influenciam no ganho e na perda de peso dos animais, como: tempo, qualidade do pasto, tipo de manejo, sexo, idade dos animais e etc. A utilização de uma rede neural para facilitar a precisão e assegurar a validade do peso captado no Sistema de Balança de Passagem diminuindo a margem de erro do peso captado em relação ao peso real, traria mais segurança na tomada de decisão por parte do pecuarista.

---

<sup>1</sup>USB é a sigla para Universal Serial Bus.Trata-se de uma tecnologia para conexão de diversos tipos de aparelhos ao computador.

<sup>2</sup>Web Service ou serviços web são soluções utilizadas para integrar a comunicação entre sistemas diferentes.

# Referências Bibliográficas

- [1] Api rxtx para manipulação das portas seriais. Relatório técnico, rxtx.org, <http://www.rxtx.org>. Data de Acesso: 20 de agosto de 2007.
- [2] Auto-ID Labs. <http://autoidlabs.org/page.html>. Data de Acesso: 18 março de 2009.
- [3] RFID System Components and Costs. <http://www.rfidjournal.com/article/articleview/1336/1/129/>. Data de Acesso: 10 março de 2009.
- [4] SISTEMA Integrado de Rastreabilidade Bovina. <http://www.sirb.com.br/pg/rastreabilidade.php>. Data de Acesso: 12 junho 2007.
- [5] The Basics of RFID Technology. <http://www.rfidjournal.com/article/articleview/1337/1/129/>. Data de Acesso: 08 março de 2009.
- [6] Chiesa, M.; Genz, R.; Heubler, F. Mingo, K.; Noessel, C.; Sopieva, N.; Slocombe, D.; Tester, J. RFID: a week long survey on the technology and its potential, as part of the Harnessing Technology project for Interaction Design Institute Ivrea. <http://people.interaction-ivrea.it/c.noessel/RFID/RFID/research.pdf>, Março 2002. Data de Acesso: 20 de julho de 2007.
- [7] Clark, Sean; Traub, Ken; Anarkat, Dipan; Osinski, Ted. Auto-id savant specification 1.0. Relatório técnico, Auto-ID Center, <http://develop.autoidcenter.org/archives/sagsavant/att0034/WDsavant1020030724.doc>, setembro 2003. Versão 1.0.
- [8] BRASIL. Ministério da Agricultura Pecuária e Abastecimento, editor. *Instrução Normativa nº 1 de 9 de janeiro de 2002*, 1. Brasília, DF, 2006. Publicado no D.O.U em 10/01/2002.
- [9] Finkenzeller, K. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*. John Wiley & Sons, 2º edição, 2003.
- [10] Christian Floerkemeier e Matthias Lampe. Rfid middleware design: addressing application requirements and rfid constraints. In *sOc-EUSAI '05: Proceedings of the 2005 joint conference on Smart objects and ambient intelligence*, páginas 219–224. ACM, New York, NY, USA, 2005. ISBN 1-59593-304-2.



- [11] John Kleinschmidt Frank Thornton, Brad Haines. *RFID Security*, página 264. Syn-  
gress, Abril 2006.
- [12] LOPES, M. A. *Informática Aplicada à Bovinocultura*, página 82. FUNEP: Jaboti-  
cabal, 1997.
- [13] Lopes, Marcos Aurélio; Santos, Glauber dos. *Custo da Implantação da Rastreabili-  
dade Bovina e Bubalina Utilizando os Diferentes Métodos de Identificação Permitidos  
pelo SISBOV*, páginas 76–80. Anais do 6º Congresso Brasileiro de Agroinformática,  
SBIAgro, outubro 2007. São Pedro, Brasil.
- [14] LUCHIARI, A. F. Beefpoint: por que rastreabilidade?  
<http://www.beefpoint.com.br/secoes/radar/printpage.asp>. Data de Acesso: 22  
março 2007.
- [15] LUCHIARI FILHO, A. Produção de carne bovina no brasil: Qualidade, quantidade  
ou ambas? in: II simpósio sobre desafios e novas tecnologias na bovinocultura de  
corte. Relatório técnico, SIMBOI'06. Brasília-DF, 2006.
- [16] K. Mohan e B. Ramesh. Managing variability with traceability in product and ser-  
vice families. In *HICSS '02: Proceedings of the 35th Annual Hawaii International  
Conference on System Sciences (HICSS'02)-Volume 3*, página 76. IEEE Computer  
Society, Washington, DC, USA, 2002. ISBN 0-7695-1435-9.
- [17] Murakami, Edson; Saraiva, Antônio Mauro. *Rastreabilidade da Informação nas  
Cadeias Produtivas: Padrões de Troca de Dados*, volume 7, páginas 58–66. Revista  
Brasileira de Agroinformática, 2005.
- [18] José Geraldo Pacheco Ormond. *Glossário de Termos Usados em Atividades  
Agropecuárias, Florestais e Ciências Ambientais*. Banco Nacional de Desenvolvi-  
mento Econômico e Social, Rio de Janeiro, 2º edição, 2004.
- [19] Pires, P. P. ; Almeida, Nalvo F ; Hanashiro, Erik J ; Dias, Waldemir M ; Goiozo, P. F.  
I. Interface de hardware e software para conexão de balança, identificador eletrônico  
e teclado do peão com pc. *Zootec*, 2005. Produção Animal e Responsabilidade.
- [20] PIRES, P. P.; AMARAL, T. B.; GOIOZO, P. F. I.; FERREIRA, V. P. C.; CORRÊA,  
E. S.; ALMEIDA, J. G. de. Identificadores eletrônicos em bovinos: uma nova fer-  
ramenta para o gerenciamento de rebanhos de corte. Relatório técnico, Embrapa  
Gado de Corte, <http://www.cnpqc.embrapa.br/publicacoes/cot/cot97/COT97.pdf>,  
2006. Campo Grande, MS.
- [21] Prabhu, B. S.; Su, Xiaoyong; Ramamurthy, Harish; Chu, Chi-Cheng; Gadh, Rajit.  
Winrfid-a middleware for the enablement of radio frequency identification (rfid) based  
applications, 2005.
- [22] SILBERSCHATZ, ABRAHAM; GALVIN, PETER BAER; GAGNE, GREG . *Sis-  
temas Operacionais com Java*. 3 edição, 2007.

# Anexo I: Arquivo de Configuração (XML Schema da configuração do *Middleware*)

Arquivo de configuração *mapping.xml* que modela a saída dos dados lidos.

```
<?xml version='1.0'?>
<mapping>
<class name='br.ufms.embrapa.classes.WriteReadValues' />
<field name='equipmentName' type='string' />
  <field name='portName' type='string' />
  <field name='value' type='string' />
  <field name='time' type='string' />
</class>
```

Modelo do arquivo de configuração *listports.xml* dos dispositivos e das portas serias.

```
<class name='br.ufms.embrapa.core.Port'>
  <field name='equipmentName' type='string' />
  <field name='portName' type='string' />
  <field name='baud_rate' type='string' />
  <field name='dataBits' type='string' />
  <field name='stopBits' type='string' />
  <field name='parity' type='string' />
<field name='FlowControlMode' type='string' />
  <field name='startStream' type='string' />
  <field name='endStream' type='string' />
  <field name='EndOfInputChar' type='string' />
  <field name='readUnique' type='string' />
  <field name='filterValues' type='string' />
```

```
<field name='fileName'          type='string' />
<field name='directoryName'     type='string' />
<field name='endOfCharInclusive' type='string' />
<field name='checkoutError'     type='string' />
</class>

<class name='br.ufms.embrapa.classes.ListObjects'>
  <map-to xml='list-objects' />
  <field name='listObjectName' type='string' />
  <field name='objects'
    type='br.ufms.embrapa.core.Port'
    collection='collection'>
    <bind-xml name='objects' />
  </field>
</class>
</mapping>
```

# Anexo II: Protocolo Balança KT40

**1. Formato geral** O protocolo de comunicação funciona no sistema PEDIDO e RESPOSTA, ou seja, cada informação requerida do KT40 deve ser precedida de um pedido. Para sincronizar esse pedido, existe a cabeça. O valor da cabeça é \$85. Após enviar a cabeça o KT40 responde com \$7A.

**Atenção:**

\$ = número em hexadecimal.

' ' = caracter ASCii

Após receber o \$7A, o PC deve imediatamente enviar os bytes do pedido. O tempo de timeout é de 3 mS.

Pedido: PC envia para a KT40 um \$85

Resposta: KT40 envia para o PC um \$7A

Pedido: PC envia para a KT40 os jDADOS do pedidoj

Resposta: KT40 envia para o PC os jDADOS da respostaj

Os dados do pedido ou da resposta seguem o seguinte formato:

1o byte: TAMANHO

2o byte: COMANDO

onde:

TAMANHO = Número de bytes total do pedido.

Exemplo:

No pedido: \$02,'M'

\$02 é o tamanho, ou número de bytes total, 'M' é o comando. A resposta é: dd/mm hh:mm +pppp.p 00

## 2. Configuração da comunicação serial

Baudrate: 19200 bps

Bits de dado: 8

Bits de stop: 1

Bits de start: 1

Bits de paridade: nenhuma

3. Comandos a) Transmite peso durante a pesagem Pedido: \$85 Resposta: \$7A  
Pedido: \$02,'M' Resposta: dd/mm hh:mm +pppp.p 00 A resposta é em ASCii. onde:  
dd/mm = dia / mês hh:mm = hora : minuto da pesagem + = peso positivo (+) ou

negativo (-) pppp.p = valor do peso em kg 00 = status, pesando (00), estabilizado (10) ou zero (01).

b) Transmite peso logo após estabilizar A KT40 transmite o peso após a estabilização da mesma forma que no comando anterior, porém não é necessário nenhum pedido. O peso é transmitido uma única vez. Pedido: nenhum Resposta: dd/mm hh:mm +pppp.p 10 onde: dd/mm = dia / mês hh:mm = hora : minuto da pesagem + = peso positivo (+) ou negativo (-) pppp.p = valor do peso em kg 10 = status estabilizado Essa opção deve ser a preferida pelos integradores por ser mais simples. Para implementar essa opção, não é necessário ler o item 1. Formato geral, basta seguir o item 2. Configuração da comunicação serial.

# Anexo III: Tabela de Dados

Tempo	Peso	Tempo	Peso	Tempo	Peso	Tempo	Peso	Tempo	Peso	Tempo	Peso	Tempo	Peso
0	000.0	250	013.0	500	011.0	750	016.5	1000	376.0	1250	012.0		
5	000.0	255	013.0	505	011.0	755	015.0	1005	376.0	1255	012.0		
10	000.0	260	013.0	510	011.0	760	014.0	1010	011.5	1260	012.0		
15	000.0	265	013.0	515	011.0	765	014.0	1015	011.5	1265	012.0		
20	000.0	270	013.0	520	011.0	770	013.5	1020	011.5	1270	012.0		
25	000.5	275	013.0	525	011.0	775	013.0	1025	011.5	1275	012.0		
30	259.0	280	013.0	530	011.0	780	013.0	1030	012.0	1280	012.0		
35	267.0	285	013.0	535	011.0	785	013.0	1035	012.0	1285	012.0		
40	303.0	290	013.0	540	011.0	790	013.0	1040	012.5	1290	012.0		
45	357.0	295	013.0	545	011.0	795	013.0	1045	012.5	1295	012.0		
50	357.0	300	013.0	550	437.0	800	013.0	1050	012.5	1300	012.0		
55	510.0	305	013.0	555	437.0	805	013.0	1055	012.5	1305	012.0		
60	510.0	310	013.0	560	364.0	810	013.0	1060	012.5	1310	012.0		
65	511.0	315	013.0	565	364.0	815	013.0	1065	012.5	1315	012.0		
70	511.0	320	013.0	570	017.5	820	013.0	1070	012.5	1320	012.0		
75	506.0	325	013.0	575	015.0	825	013.0	1075	012.5	1325	012.0		
80	484.0	330	013.0	580	012.5	830	013.0	1080	012.5	1330	012.0		
85	008.5	335	004.0	585	010.5	835	013.0	1085	012.5	1335	012.0		
90	008.5	340	004.0	590	010.0	840	013.0	1090	012.5	1340	012.0		
95	009.0	345	000.0	595	011.0	845	013.0	1095	012.5	1345	012.0		
100	009.0	350	001.5	600	011.5	850	013.0	1100	012.5	1350	012.0		
105	009.0	355	001.5	605	012.0	855	013.0	1105	012.5	1355	012.0		
110	009.0	360	005.0	610	012.0	860	013.0	1110	012.5	1360	012.0		
115	009.5	365	005.0	615	012.0	865	013.0	1115	012.5	1365	012.0		
120	010.0	370	006.5	620	012.0	870	013.0	1120	012.5	1370	012.0		
125	011.0	375	008.5	625	012.0	875	013.0	1125	012.5	1375	012.0		
130	011.0	380	009.5	630	012.0	880	013.0	1130	012.5	1380	012.0		
135	011.0	385	021.0	635	012.0	885	013.5	1135	012.5	1385	012.0		
140	011.0	390	063.5	640	012.0	890	013.5	1140	012.5	1390	012.0		
145	011.0	395	063.5	645	012.0	895	017.5	1145	012.5	1395	012.0		
150	011.0	400	189.0	650	012.0	900	015.5	1150	012.5	1400	012.0		
155	011.0	405	189.0	655	012.0	905	013.5	1155	013.0	1405	012.0		
160	011.0	410	149.0	660	012.0	910	012.5	1160	012.0	1410	012.0		
165	008.5	415	014.5	665	012.0	915	012.5	1165	012.0	1415	012.0		
170	008.5	420	014.0	670	013.5	920	012.0	1170	012.5	1420	012.0		
175	040.5	425	014.0	675	013.5	925	012.0	1175	012.5	1425	012.0		
180	040.5	430	015.0	680	013.5	930	011.5	1180	012.0	1430	012.0		
185	120.0	435	015.0	685	013.5	935	011.5	1185	012.0	1435	012.0		
190	402.0	440	034.0	690	013.5	940	011.5	1190	012.0	1440	012.0		
195	349.0	445	034.0	695	013.5	945	011.5	1195	012.0	1445	012.0		
200	349.0	450	250.0	700	013.5	950	011.5	1200	012.0	1450	012.0		
205	011.5	455	250.0	705	013.5	955	011.5	1205	012.0	1455	012.0		
210	011.5	460	017.0	710	013.5	960	427.0	1210	012.0	1460	012.0		
215	012.0	465	015.5	715	013.5	965	427.0	1215	012.0	1465	012.0		
220	011.5	470	014.0	720	013.5	970	382.0	1220	012.0	1470	012.0		
225	012.0	475	012.0	725	478.0	975	382.0	1225	012.0	1475	012.0		
230	012.5	480	011.5	730	381.0	980	292.0	1230	012.0	1480	012.0		
235	012.5	485	011.5	735	016.0	985	292.0	1235	012.0	1485	012.0		
240	012.5	490	011.0	740	016.0	990	385.0	1240	012.0	1490	012.0		
245	013.0	495	011.0	745	016.5	995	385.0	1245	012.0	1495	012.0		

Tabela 9.1: Tabela de pesagem do animais, passando sobre a balança. Tempo em segundo e o peso em kg.

Tempo	Peso	Tempo	Peso	Tempo	Peso	Tempo	Peso	Tempo	Peso	Tempo	Peso
1500	012.0	1840	012.0	2180	318.0	2520	318.0	2860	012.5	3200	012.0
1505	012.0	1845	012.0	2185	318.0	2525	318.0	2865	012.5	3205	012.0
1510	012.0	1850	012.0	2190	318.0	2530	318.0	2870	012.5	3210	012.0
1515	012.0	1855	012.0	2195	318.0	2535	318.0	2875	012.5	3215	012.0
1520	012.0	1860	012.0	2200	318.0	2540	318.0	2880	012.5	3220	012.0
1525	012.0	1865	012.0	2205	318.0	2545	318.0	2885	012.5	3225	012.0
1530	012.0	1870	012.0	2210	318.0	2550	318.0	2890	012.5	3230	012.0
1535	012.0	1875	012.0	2215	318.0	2555	318.0	2895	012.5	3235	012.0
1540	012.0	1880	012.0	2220	318.0	2560	318.0	2900	012.5	3240	012.0
1545	012.0	1885	012.0	2225	318.0	2565	318.0	2905	012.5	3245	012.0
1550	012.0	1890	012.0	2230	318.0	2570	318.0	2910	012.5	3250	012.0
1555	012.0	1895	012.0	2235	318.0	2575	318.0	2915	012.5	3255	012.0
1560	012.0	1900	012.0	2240	318.0	2580	318.0	2920	012.5	3260	012.0
1565	012.0	1905	243.0	2245	318.0	2585	318.0	2925	012.5	3265	012.0
1570	012.0	1910	243.0	2250	318.0	2590	318.0	2930	012.5	3270	012.0
1575	012.0	1915	262.0	2255	318.0	2595	318.0	2935	012.5	3275	012.0
1580	012.0	1920	318.0	2260	318.0	2600	318.0	2940	012.5	3280	012.0
1585	012.0	1925	318.0	2265	318.0	2605	318.0	2945	518.0	3285	012.0
1590	012.0	1930	318.0	2270	318.0	2610	318.0	2950	518.0	3290	012.0
1595	012.0	1935	318.0	2275	318.0	2615	318.0	2955	525.0	3295	012.0
1600	012.0	1940	318.0	2280	318.0	2620	318.0	2960	525.0	3300	012.0
1605	012.0	1945	319.0	2285	318.0	2625	318.0	2965	501.0	3305	012.0
1610	012.0	1950	319.0	2290	318.0	2630	318.0	2970	501.0	3310	012.0
1615	012.0	1955	318.0	2295	318.0	2635	318.0	2975	012.5	3315	012.0
1620	012.0	1960	318.0	2300	318.0	2640	318.0	2980	012.5	3320	012.0
1625	012.0	1965	318.0	2305	318.0	2645	318.0	2985	012.0	3325	012.0
1630	012.0	1970	318.0	2310	318.0	2650	318.0	2990	012.0	3330	012.0
1635	012.0	1975	318.0	2315	318.0	2655	318.0	2995	012.0	3335	012.0
1640	012.0	1980	318.0	2320	318.0	2660	318.0	3000	012.0	3340	012.0
1645	012.0	1985	318.0	2325	318.0	2665	318.0	3005	012.5	3345	012.0
1650	012.0	1990	318.0	2330	318.0	2670	318.0	3010	012.5	3350	012.0
1655	012.0	1995	318.0	2335	318.0	2675	135.0	3015	012.5	3355	012.0
1660	012.0	2000	318.0	2340	318.0	2680	135.0	3020	012.5	3360	012.0
1665	012.0	2005	318.0	2345	318.0	2685	135.0	3025	012.5	3365	012.0
1670	012.0	2010	318.0	2350	318.0	2690	014.5	3030	012.5	3370	012.0
1675	012.0	2015	318.0	2355	318.0	2695	014.0	3035	012.5	3375	012.0
1680	012.0	2020	318.0	2360	318.0	2700	013.5	3040	012.5	3380	299.0
1685	012.0	2025	318.0	2365	318.0	2705	013.0	3045	012.5	3385	299.0
1690	012.0	2030	318.0	2370	318.0	2710	012.5	3050	012.5	3390	299.0
1695	012.0	2035	318.0	2375	318.0	2715	012.5	3055	012.5	3395	300.0
1700	012.0	2040	318.0	2380	318.0	2720	012.5	3060	012.5	3400	300.0
1705	012.0	2045	318.0	2385	318.0	2725	012.5	3065	012.5	3405	301.0
1710	012.0	2050	318.0	2390	318.0	2730	012.5	3070	012.5	3410	302.0
1715	012.0	2055	318.0	2395	318.0	2735	012.5	3075	012.5	3415	302.0
1720	012.0	2060	318.0	2400	318.0	2740	012.5	3080	012.5	3420	302.0
1725	012.0	2065	318.0	2405	318.0	2745	012.5	3085	012.5	3425	302.0
1730	012.0	2070	318.0	2410	318.0	2750	012.5	3090	012.5	3430	302.0
1735	012.0	2075	318.0	2415	318.0	2755	012.5	3095	012.5	3435	302.0
1740	012.0	2080	318.0	2420	318.0	2760	200.0	3100	012.5	3440	302.0
1745	012.0	2085	318.0	2425	318.0	2765	181.0	3105	012.5	3445	302.0
1750	012.0	2090	318.0	2430	318.0	2770	147.0	3110	012.5	3450	231.0
1755	012.0	2095	318.0	2435	318.0	2775	147.0	3115	012.5	3455	204.0
1760	012.0	2100	318.0	2440	318.0	2780	012.0	3120	012.5	3460	161.0
1765	012.0	2105	318.0	2445	318.0	2785	012.0	3125	012.5	3465	011.5
1770	012.0	2110	318.0	2450	318.0	2790	012.0	3130	012.5	3470	011.5
1775	012.0	2115	318.0	2455	318.0	2795	012.5	3135	012.5	3475	012.0
1780	012.0	2120	318.0	2460	318.0	2800	012.5	3140	012.5	3480	012.0
1785	012.0	2125	318.0	2465	318.0	2805	012.5	3145	012.5	3485	012.0
1790	012.0	2130	318.0	2470	318.0	2810	012.5	3150	501.0	3490	012.0
1795	012.0	2135	318.0	2475	318.0	2815	012.5	3155	503.0	3495	012.5
1800	012.0	2140	318.0	2480	318.0	2820	012.5	3160	476.0	3500	012.5
1805	012.0	2145	318.0	2485	318.0	2825	012.5	3165	421.0	3505	012.5
1810	012.0	2150	318.0	2490	318.0	2830	012.5	3170	421.0	3510	012.5
1815	012.0	2155	318.0	2495	318.0	2835	012.5	3175	012.5	3515	012.5
1820	012.0	2160	318.0	2500	318.0	2840	012.5	3180	012.5	3520	012.5
1825	012.0	2165	318.0	2505	318.0	2845	012.5	3185	012.0	3525	012.5
1830	012.0	2170	318.0	2510	318.0	2850	012.5	3190	012.0	3530	012.5
1835	012.0	2175	318.0	2515	318.0	2855	012.5	3195	012.0	3535	012.5

Tabela 9.2: Tabela de pesagem do animais, passando sobre a balança. Tempo em segundo e o peso em kg, continuação 9.1.

## Anexo IV: Coleta dos Pesos dos Animais

<i>Chip</i> intra-ruminal	Pesagem 06/03	Pesagem 04/05	Pesagem 19/05
76000000007578	458,0	494,0	506,0
76000000007612	493,0	516,0	524,0
76000000007606	498,0	542,0	554,0
76000000007775	497,0	536,0	537,0
76000000007644	458,0	498,0	497,0
76000000007714	516,0	550,0	543,0
76000000007324	357,0	407,0	396,0
76000000007072	405,0	430,0	426,0
76000000007132	498,0	550,0	556,0

Tabela 9.3: Tabela de coleta do peso dos animais parado. Peso em kg.

<i>Chip</i> intra-ruminal	Pesagem 07/05				<i>Chip</i> intra-ruminal	Pesagem 12/05		
76000000007578	489,0	501,0	-	-	76000000007578	576,0	-	-
76000000007612	115,0	520,0	-	-	76000000007612	516,0	679,0	-
76000000007606	-	-	-	-	76000000007606	369,0	-	-
76000000007775	-	-	-	-	76000000007775	449,0	-	-
76000000007644	484,0	505,0	-	-	76000000007644	491,0	408,0	-
76000000007714	-	-	-	-	76000000007714	544,0	-	-
76000000007324	399,0	399,0	365,0	409,0	76000000007324	460,0	394,0	-
76000000007072	-	-	-	-	76000000007072	425,0	-	-
76000000007132	-	-	-	-	76000000007132	552,0	545,0	564,0

Tabela 9.4: Tabela de coleta do peso dos animais em marcha. Peso em kg.



<i>Chip</i> intra-ruminal	Pesagem 13/05					<i>Chip</i> intra-ruminal	Pesagem 14/05		
76000000007578	520,0	371,0	484,0	494,0	-	76000000007578	502,0	495,0	500,0
76000000007612	560,0	474,0	-	-	-	76000000007612	-	-	-
76000000007606	504,0	592,0	540,0	531,0	-	76000000007606	541,0	548,0	-
76000000007775	486,0	455,0	533,0	493,0	535,0	76000000007775	527,0	499,0	529,0
76000000007644	492,0	-	-	-	-	76000000007644	-	-	-
76000000007714	526,0	704,0	529,0	544,0	-	76000000007714	393,0	545,0	-
76000000007324	561,0	279,0	-	-	-	76000000007324	390,0	-	-
76000000007072	416,0	300,0	388,0	-	-	76000000007072	-	-	-
76000000007132	562,0	546,0	531,0	557,0	-	76000000007132	510,0	546,0	-

Tabela 9.5: Tabela de coleta do peso dos animais em marcha. Peso em kg.

<i>Chip</i> intra-ruminal	Pesagem 15/05					<i>Chip</i> intra-ruminal	Pesagem 19/05			
76000000007578	503,0	504,0	399,0	502,0	502,0	76000000007578	422,0	502,0	-	-
76000000007612	-	-	-	-	-	76000000007612	516,0	488,0	-	-
76000000007606	540,0	545,0	329,0	539,0	-	76000000007606	556,0	447,0	549,0	-
76000000007775	515,0	531,0	506,0	530,0	-	76000000007775	534,0	366,0	526,0	341,0
76000000007644	481,0	506,0	-	-	-	76000000007644	487,0	-	-	-
76000000007714	539,0	540,0	-	-	-	76000000007714	545,0	-	-	-
76000000007324	-	-	-	-	-	76000000007324	483,0	377,0	401,0	-
76000000007072	-	-	-	-	-	76000000007072	431,0	417,0	640,0	-
76000000007132	552,0	553,0	537,0	548,0	-	76000000007132	535,0	559,0	-	-

Tabela 9.6: Tabela de coleta do peso dos animais em marcha. Peso em kg.

<i>Chip</i> intra-ruminal	Pesagem 20/05				<i>Chip</i> intra-ruminal	Pesagem 21/05			
76000000007578	422,0	502,0	-	-	76000000007578	463,0	508,0	507,0	416,0
76000000007612	516,0	488,0	-	-	76000000007612	-	-	-	-
76000000007606	556,0	447,0	549,0	-	76000000007606	432,0	537,0	-	-
76000000007775	534,0	366,0	526,0	341,0	76000000007775	534,0	463,0	-	-
76000000007644	487,0	-	-	-	76000000007644	-	-	-	-
76000000007714	545,0	-	-	-	76000000007714	542,0	-	-	-
76000000007324	483,0	377,0	401,0	-	76000000007324	-	-	-	-
76000000007072	431,0	417,0	640,0	-	76000000007072	534,0	-	-	-
76000000007132	535,0	559,0	-	-	76000000007132	534,0	549,0	-	-

Tabela 9.7: Tabela de coleta do peso dos animais em marcha. Peso em kg.

<i>Chip</i> intra-ruminal	Pesagem 28/05	
76000000007578	-	-
76000000007612	-	-
76000000007606	534,0	-
76000000007775	529,0	479,0
76000000007644	-	-
76000000007714	534,0	534,0
76000000007324	-	-
76000000007072	-	-
76000000007132	475,0	512,0

Tabela 9.8: Tabela de coleta do peso dos animais em marcha. Peso em kg.

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)