CENTRO UNIVERSITÁRIO DE BRASÍLIA -UniCEUB

CURSO DE ENGENHARIA DE COMPUTAÇÃO

SÉRGIO MARTINS DA SILVA

Ajuda Eletrônica à Identificação de Falsificação de Notas de Dinheiro

Orientador: M.C.Prof. Maria Marony Sousa Farias

Brasília

Novembro/2010

SÉRGIO MARTINS DA SILVA

Ajuda Eletrônica à Identificação de Falsificação de Notas de Dinheiro

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB) como pré-requisito para a obtenção de Certificado de Conclusão de Curso de Engenharia de Computação.
Orientador:M.C. Prof.
Maria Marony Sousa Farias

Brasília
Novembro/2010

SÉRGIO MARTINS DA SILVA

Ajuda Eletrônica à Identificação de Falsificação de Notas de Dinheiro

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB) como pré-requisito para a obtenção de Certificado de Conclusão de Curso de Engenharia de Computação.
Orientador:M.C.Prof.Maria Maron Sous a Farias

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

_____

Prof. Abiezer Amarilia Fernandez
Coordenador do Curso

Banca Examinadora:

_____

M.C.Prof. Maria Marony Sousa Farias
Orientador

_____

M.C.Prof. Francisco Javier
Uniceub

_____

M.C.Prof. . Vera Farini
Uniceub

DEDICATÓRIA

Dedico este trabalho a minha esposa Daiane Vaz e a minha sogra Maria Aparecida, com todo amor, pelo esforço que fizeram para realização deste sonho.

AGRADECIMENTOS

A minha esposa Daiane Vaz, meus agradecimentos por ter aceito se privar de minha companhia pelos estudos, concedendo a mim a oportunidade de realizar este sonho.

A minha mãe Elidia pelo seu apoio moral e financeiro quando necessário.

A DEUS, por ter me iluminado nessa jornada, e, a todos que me ajudaram a chegar até aqui. Obrigado.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE SIGLAS

SUID: Sistema Único de Identificação de Dinheiro.

RFID: Radio-Frequency IDentification.

RIFIDI: suite de ferramentas para desenvolvimento integrado de soluções RFID.

EPC: Eletronic Product Code.

UHF: Ultra High Frequency.

ONS:Object Naming Service.

DoS: Denial Of Service.

JCP: Java Community Process.

JSE: Java Standard Edition.

JDK: Java Development Kit.

JEE: Java Enterprise Edition.

RESUMO

O estudo do RFID associado aos meios de controle de falsificação de notas de dinheiro busca validar um modelo de implementação que explore ao máximo a tecnologia somatizando alternativas de mitigação no seu uso, especialmente em instituições financeiras. Atenuar a falsificação de cédulas tem sido um constante desafio das autoridades em se tratando de uma sociedade que caminha rumo a tecnologia de ponta, porém a sua realização ainda é uma tarefa árdua e gera desconfortos na população, desviando o agente falsificador do alvo, em virtude das várias técnicas encontradas para burlar as autoridades. O desenvolvimento desse projeto está baseado nos conceitos de controle de notas através do uso de etiquetas RFID. A proposta é o desenvolvimento de um sistema que utilize alguns aspectos computacionais para garantir o controle da veracidade das moedas circulantes de forma mais confiável aos usuários, causando o mínimo impacto na sociedade e disponibilizando maior segurança ao acesso das mesmas. O hardware necessário para o funcionamento do sistema é composto pelas etiquetas, ou *transponders* e os leitores de etiquetas, chamados de *transceivers.* O controle desses dispositivos é realizado pelo *middleware*, um conjunto de aplicações que controla os atributos de dados entre a etiqueta e a aplicação externa. Estes atributos são armazenados em um banco de dados, dentro de um raio de leitura predefinido e as etapas do sistema são realizadas seqüencialmente, acessando o banco de dados e a configuração do emulador para realizar a leitura das *tags*, por meio de representação hexadecimal e validar as notas de dinheiro.

Palavras Chave: RFID, Falsificação, Java

ABSTRACT

The study of Radio-Frequency Identification (RFID) associated with the control of counterfeit money bills seeks to validate an implementation model to explore the most of technology creating alternatives to mitigate its use, especially in financial institutions. Mitigate the falsification of ballots has been a constant challenge for authorities in a society that moves towards the technology, but its realization is still an arduous task and generates discomfort in the population, skewing the forger agent of the target, because of the various techniques found to circumvent the authorities. The development of this project is based on the concepts of money ballots control through the use of RFID tags. The proposal is to develop a system that uses computational aspects to ensure a veracity control of the coins circulating more reliably to users, resulting in minimal impact on society and providing greater security to access them. The hardware needed to operate the system consists of tags called transponders and readers of labels called transceivers. The control of these devices will be performed by the middleware, a set of applications that control the attributes of data between the label and external application. These attributes are stored in a database within a default radius of reading and system's steps are performed sequentially accessing the database and the emulator settings to read the tags using hexadecimal representation and to validate the banknotes.

Key words: RFID, Counterfeit, Java

1 – INTRODUÇÃO

1.1 – Apresentação do Problema

Com o grande aumento no interesse mundial sobre a tecnologia RFID, é necessário que sejam estudadas as formas como estas aplicações estão sendo desenvolvidas e quais são os principais desafios que devem ser superados para implementações de sucesso com o uso desta tecnologia.

Um dos setores que merece destaque e ainda carece de amadurecimento de soluções RFID é o setor bancário, no qual os ganhos com as implantações desta tecnologia vão muito além da substituição da tecnologia de automação de captura de dados (do código de barras pelas etiquetas de radio freqüência) tendo como principal objetivo aumentar a segurança de rede bancária e seus sistemas, de forma a fornecer ferramentas que possibilitem total rastreabilidade e controle das notas de dinheiro que circulam em toda rede.

Com esta visão, vários projetos estão sendo desenvolvidos nos últimos anos, com o intuito de promover a tecnologia e fazer medições da redução da falsificação de notas de dinheiro que podem ser obtidos com a inclusão do RFID para estas operações.

Muitas instituições financeiras vêm apresentando dificuldades em lidar com a proliferação de nota falsas em suas transações, sendo imprescindível a aplicação de uma investigação muito mais aprofundada e precisa da sistemática de avaliação das falsificações, para permitir que ocorra uma apreensão do grande quantitativo informado pela polícia federal.

Muitas vezes as cédulas ficam armazenadas em equipamentos da própria instituição de forma não integrada, ou seja, espalhadas em lotes de diferentes pontos geográficos, dificultando sua rastreabilidade e conseqüentemente, o combate a esse tipo de crime.

Devido à proliferação dessas notas falsas produzidas em laboratórios altamente equipados, sem o intuito de compartilhamento de informações, e de várias aplicações, desenvolvidas ao longo do tempo para atender a demandas específicas, ocorrem muitas vezes, a essas instituições, a perda da visão global do que ela realmente possui no que tange a dados e informações confiávies. A idéia principal deste projeto surgiu após o conhecimento de que cerca de um terço da população já havia tido contato com uma nota falsa de real, índice considerado muito elevado e que chamou a atenção da autoridade monetária.

Aproximadamente 1% das unidades da autoridade monetária espalhadas no país têm criado agora laboratórios com o objetivo de investigar esse tipo de crime.

Segundo dados informados pelo Banco Central do Brasil, as medidas de prevenção adotadas para combater o crime eletrônico deverá permitir tanto o reconhecimento de uma caligrafia falsa, como também seu formato, a pressão feita sobre o papel, o tipo de máquina e tintas utilizadas.

Se existem hoje, no mundo, esses tipos de problemas que crescem em grande escala, porque então não se criar um método que auxilie a Polícia Federal a implantar elementos de segurança, com melhor eficiência e agilidade no uso de suas aplicações do dia a dia?

1.2 – Objetivos do Trabalho

O objetivo geral é o desenvolvimento de um sistema de validação de cédulas de dinheiro utilizando a tecnologia de RFID, unificando informações constantes em banco de dados e comparando-as aos atributos configurados nas etiquetas inteligentes, por meio de um emulador que simula um leitor RFID em execução. A forma de acesso a este banco se dará pelo uso de um *middleware*.

Os objetivos específicos do trabalho são:

a) Identificar os elementos de segurança analisados na aplicação RIFIDI;
b) Armazenar esses elementos em um banco de dados;
c) Definir o raio de leitura das etiquetas;
d) Configurar os parâmetros específicos do leitor;
e) Desenvolver um sistema de gerenciamento das informações para acessar esse banco de dados;
f) Configurar o emulador que irá fazer a leitura das tags e validar as notas de dinheiro (representação hexadecimal);

d) disponibilizar, através do leitor baseado em RFID, o acesso ao conteúdo que fica armazenado na base de dados.

1.3 – Justificativa e Importância do Trabalho

O projeto foi motivado pela vontade de resolver um problema real que necessitasse de uma solução tecnológica de engenharia: a segurança de cédulas de dinheiro. Segundo uma pesquisa do Laboratório *Thomson*, em 2009 a relação de cédulas falsas por verdadeiras apresentava níveis preocupantes e muito acima do que acontece com o euro, cuja relação é de três cédulas falsas por milhão. Verificou-se um aumento de 16% ao ano (desde 2004). A falta de investimentos em infra-estrutura e segurança em tecnologia da informação aliados ao crescimento do número de falsificadores são as principais causas desses dados caóticos.

A falsificação do real foi estimulada pela estabilidade monetária desde a implantação do real, em 1994. Sobretudo nos anos 2000, a quantidade de notas falsas cresceu bastante, atingindo seu ápice em 2007, quando o BC tirou de circulação quase 667 mil delas. É o equivalente a 170 notas falsas por cada milhão de cédulas verdadeiras em circulação.

Algumas reportagens foram essenciais para a motivação e para a definição da tecnologia a ser usada. Só no ano de 2009 a Polícia Federal prendeu quatro quadrilhas de grande porte que operavam nos estados de Minas Gerais, Rio Grande do Sul, Paraná e São Paulo, mas que despachavam as cédulas falsas para diversos outros estados, inclusive o Rio. Ao todo, foram apreendidas quase 63 mil notas de R$ 10, R$ 20, R$ 50 e R$ 100. Mas, pelo porte dos criminosos, a PF calcula que o potencial era de mais de 800 mil cédulas falsas. Esse volume representa quase 20% das 4,981 milhões de notas falsas tiradas de circulação na última década.

A instalação de etiquetas identificadoras em todas as notas de real talvez leve alguns anos para atingir usuários e comerciantes no Brasil, já que a técnica não é tão simples nem barata. Segundo o Instituto os equipamentos portáteis com capacidade de análise rápida e com preço reduzido ainda estão em desenvolvimento. No entanto, os pesquisadores ponderam que esse é um assunto que deve ser resolvido pelo mercado. Esses identificadores são etiquetas RFID similares às etiquetas utilizadas em veículos nos pedágios de algumas rodovias brasileiras.

Esses informativos trouxeram a oportunidade de realizar um projeto que traria um grande impacto positivo na sociedade aproveitando um conceito já bastante difundido e que ainda teríamos alguns desafios para resolver, tais como a identificação de notas falsas através da técnica *Easy Ambient Sonicspray Ionization* (EASI), que, por meio de análise química,

identifica as características dos compostos presentes nas cédulas, obtendo a resposta em dez segundos. A EASI defende que a cédula verdadeira tem como presença constante quatro íons, chamados de íons diagnósticos e se esses íons não estiverem presentes, a cédula é considerada falsa.

1.4 – Escopo do Trabalho

Quanto ao escopo, este trabalho se limita aos aspectos interdisciplinares da identificação de notas falsas, centrados nas relações entre o conceito de Identificação por Radio freqüência e atributos intrínsecos nas cédulas, e não busca esgotar suas especificidades técnicas. Analisam-se neste trabalho as possíveis maneiras de falsificar nota de dinheiro e novas técnicas de inibição que surgem a partir deste fenômeno e, portanto, não será aprofundado tecnicamente sua utilização em aplicações comerciais e industriais.

Para solucionar o problema de identificação de notas falsas a proposta se limita ao desenvolvimento de um módulo utilizando a tecnologia de identificação por rádio freqüência (RFID), tal módulo seria responsável por identificar corretamente os atributos quando os mesmos estiverem dentro do raio de detecção do leitor RFID. Para que a identificação ocorra, é necessária a presença de uma antena de rádio freqüência junto, e as notas deverão simular a utilização de etiquetas RFID embutidas, por meio do emulador RIFIDI. Estas etiquetas são consideradas uma solução adequada, pois elas podem ser normalmente utilizadas na vida útil das cédulas.

O módulo de identificação também poderá ser utilizado para autenticar os elementos de segurança contidos nas notas a fim de garantir que as leituras sejam válidas apenas quando este estiver presente no banco de dados. A função de verificação da presença desses elementos pode ser chamada para autenticar transações e operações financeiras. É de extrema importância que o processo de autenticação através da verificação desses elementos de segurança não cause lentidão no sistema, pois ele será útil em muitas operações e pode impactar no desempenho de todo o sistema de identificação caso não seja bem projetado.

Portanto este trabalho concentra-se no uso adequado de métodos capazes de gerar uma matriz de veracidade, que, a partir da comparação das tabelas do banco de dados com as etiquetas configuradas no emulador, reproduza padrões previamente selecionados e que garantam equidade de dados. Modelos lógicos também servirão de base para completude dos

métodos utilizados, especificamente a modelagem baseada em cenários para a estimação de parâmetros e geração de padrões previamente configurados.

1.5 – Resultados Esperados

O projeto apresenta uma proposta que visa minimizar a falsificação de notas de dinheiro, através de uma solução que atue no monitoramento e, conseqüentemente, gerenciamento, planejamento e fiscalização.

Um dos principais resultados que este projeto inicial mostra é que devido à grande inovação trazida por este tipo de tecnologia, para se atingir o sucesso nestas implantações a avaliação precisa dos elementos de segurança contidos nas cédulas e a formatação da solução são fatores decisivos para casos de sucesso.

1.6 – Estrutura do Trabalho

No projeto são mostrados os principais desafios e dificuldades deste tipo de implantação, apresentando soluções que tornem viáveis este tipo de solução para a realidade das operações bancárias brasileiras. Para atingir o objetivo deste trabalho a divisão dos capítulos ficou desta forma:

Capítulo 2 - A Tecnologia de Identificação por Rádio Freqüência (RFID): mostra todo aspecto técnico da tecnologia de **RFID**, mostrando as freqüências de operação e funcionalidades que cada uma delas oferece;

Capítulo 3 – Arquitetura proposta - emulando com RIFIDI: estuda o comportamento do simulador, detalhando toda sua arquitetura desenhada para atender o problema descrito em uma rede de Instituições Financeiras - IFs;

Capítulo 4 – Projeto e Implementação: destaca as principais fases da aplicação de **RFID** para o setor financeiro, mostrando a implementação da solução proposta;

Capítulo 5 – Resultados obtidos: traz um estudo detalhado acerca da resposta obtida com a implementação da aplicação, visando inibir a proliferação de notas falsas, focando principalmente na implantação desta tecnologia com o uso do RIFIDI;

Capítulo 6 – Conclusão: apresenta os resultados obtidos no trabalho mostrando alternativas que facilitem a implantação de soluções similares, principalmente para o mercado financeiro.

2 – APRESENTAÇÃO DO PROBLEMA

O presente capítulo apresenta conceitos necessários para a compreensão de um dos problemas mais recorrentes do sistema financeiro nos últimos tempos: a Moeda Falsa. Além disso, busca-se o entendimento do modelo atual de controle desta prática e outros tópicos relacionados à tecnologia que visa extinguí-la.

O artigo 289 do Código Penal expressa a definição de Moeda Falsa e destaca também que uma simples alteração nas cédulas de dinheiro representa uma fraude contra a fé pública no tocante à moeda como instrumento de poder de compra.(BITTENCOURT,2007)

A falsificação de moeda, hoje, já é o terceiro crime federal que mais gera inquéritos policiais, perdendo apenas para o estelionato e o contrabando.(ABOSO,2008) Com o intuito de combater este tipo de crime, recentemente foi desenvolvida uma sistemática científica de avaliação das falsificações. Isso permite que ela ocorra não só do ponto de vista de uma apreensão, mas de uma classe completa de falsificação.

Com a adoção dessa sistemática, o Banco Central e a Policia Federal tem a possibilidade de fazer uma investigação muito mais aprofundada e precisa sob o ponto de vista geográfico, tecnológico e gráfico. Para isso, as duas instituições implantaram um novo sistema tecnológico de investigação que consegue mapear as notas falsas por “famílias” — mesmo tipo de papel e/ou tinta, por exemplo — e que permite chegar aos infratores em cada região do país. Os primeiros resultados já começaram a surgir e, neste ano, já foram retiradas ou impedidas de entrar em circulação o equivalente a quase 20% do total das cédulas falsas confiscadas nos últimos dez anos.

Outro método, ainda em estudo, de proteção contra falsificações é o uso de moedas bimetálicas, que são feitas de dois metais diferentes com cores semelhantes, o que pode tornar a falsificação muito difícil. A forma mais simples de forjar este tipo de moeda é alterar a área que deve apresentar cor diferente da pintura.

2.1 Moeda Falsa

Moeda é a medida do valor das coisas. Surgiu quando o homem sentiu necessidade de abandonar os instrumentos de troca.

Segundo BONFIM (2005), em Roma a cunhagem de moedas data de três séculos a.C., porém, foi somente mais tarde, que se tratou de reprimir severamente a falsificação delas.

2.2 A tecnologia de Identificação por Rádio Freqüência (RFID)

2.2.1 Histórico do RFID

A tecnologia de RFID (Radio *Frenquency Identification*) tem sua origem nos sistemas de radares utilizados na Segunda Guerra Mundial para avisá-los com antecedência de aviões enquanto eles estavam bem distantes. O problema era identificar dentre esses aviões qual era inimigo e qual era aliado. Os alemães então descobriram que se os seus pilotos girassem seus aviões quando estivessem retornando à base iriam modificar o sinal de radio que seria refletido de volta ao radar. Esse método simples alertava os técnicos responsáveis pelo radar que se tratava de aviões alemães. (PINHEIRO, 2004).

Mas foram os Ingleses que desenvolveram o primeiro sistema de RFID ativo o IFF (*Identify Friend or Foe*). Foi colocado um transmissor em cada avião Britânico sendo que quando esses transmissores recebiam sinais das estações de radares no solo, começavam a transmitir um sinal de resposta, que identificava o avião como *Friendly* (aliado).

Os RFID funcionam no mesmo principio básico. Um sinal é enviado a um transponder, o qual é ativado e reflete de volta o sinal (sistema passivo) ou transmite seu próprio sinal (sistemas ativos).

Nas décadas de 50 e 60 cientistas e acadêmicos dos Estados Unidos, Europa e Japão realizaram pesquisas e apresentaram estudos explicando como a energia RF poderia ser utilizada para identificar objetos remotamente. Companhias começaram a comercializar sistemas antifurto que utilizavam ondas de rádio para determinar se um item havia sido roubado ou pago normalmente. (BRAUN, 2007). Era o advento das Tags (etiquetas) denominadas de “etiquetas de vigilância eletrônica” as quais ainda são utilizadas até hoje.

Cada etiqueta utiliza um bit. Se a pessoa paga pela mercadoria, o bit é posto em off ou 0. E os sensores não dispararão o alarme.

Caso o contrário, o bit continua em on ou 1, e caso a mercadoria saia através dos sensores, um alarme será disparado.

Em 1973, Charles Walton, um empreendedor da Califórnia desenvolve um *transponder* passivo usado para destravar uma porta sem a utilização de uma chave. Um cartão com um *transponder* embutido comunicava com um leitor/receptor localizado perto da porta. Quando o receptor detectava um número de identificação válido armazenado na etiqueta RFID, a porta era destravada através de um mecanismo.(BRAUN, 2007)

Na década de 1970, o laboratório nacional de Los Alamos nos EUA teve um pedido do departamento de energia para desenvolver um sistema para rastrear materiais nucleares. Um grupo de cientistas idealizou um projeto onde seria colocado um *transponder* em cada caminhão transportador, o qual corresponderia com uma identificação e potencialmente outro tipo de informação, como, por exemplo, a identificação do motorista.

No começo da década de 90, engenheiros da IBM desenvolveram e patentearam um sistema de **RFID** baseado na tecnologia **UHF** (*Ultra High Frequency*). O **UHF** oferece um alcance de leitura muito maior (aproximadamente 10 metros sobre condições boas) e transferência de dados mais velozes. Mas a tecnologia era muito custosa comparada ao pequeno volume de vendas, e a falta de interesse internacional. (LOES, 2006)

O **RFID** utilizando UHF teve uma melhora na sua visibilidade em 1999, quando o ***Uniform Code Concil***, o ***EAN internacional***, a ***Procter & Gamble*** e a ***Gillette*** se uniram e estabeleceram o Auto-ID Center, no instituto de tecnologia de Massachusetts. (LOES, 2006)

O Objetivo do Auto-ID Center era desenvolver uma rede de comunicação dentro da cadeia logística baseada em sistemas de **RFID** com a tecnologia de UHF. Com os sucessos nas pesquisas e desenvolvimento do Auto-ID Center várias outras grandes empresas se associaram ao Auto-ID center, dentre elas *Wal-Mart*, *Metro*, *Target* , *HP*, *Unilever*, e em 2002 o Auto-ID finalizou a especificação do **EPC** ("*Eletronic Product Code*" – Código Eletrônico de Produto) e de toda rede de informação que deve ser formada para que as informações gravadas nos tags de **RFID** estejam disponíveis a todos.

2.2.2 Componentes do Sistema de RFID

Os três componentes envolvidos em um sistema de **RFID** possuem características próprias e que devem ser selecionadas de acordo com as necessidades e desafios impostos pela aplicação na qual se deseja utilizar a tecnologia de **RFID**.(SANTANA, 2005)

Cada um dos componentes contribui para o desempenho total do sistema. Desta forma, para se atingir um sistema otimizado, é necessário conhecer todas as características e comportamento dos componentes de forma a compor as melhores combinações entre estes, proporcionando assim o melhor resultado global do sistema. São eles:

a) Leitores

A antena é a responsável por ativar a *Tag* para que possa haver a troca de informações, e para isso ela envia um sinal de rádio. Existem em diversos formatos e tamanhos, cada uma com sua configuração e característica distinta para um determinado tipo de aplicação. Algumas antenas são acopladas ao *transceiver* e ao decodificador em um mesmo invólucro: neste caso recebem o nome de leitor. (SANTANA, 2005)

Na figura 2.1 é mostrada um leitor, que nada mais é do que uma antena, por intermédio do *transponder*, emite as freqüências de rádio, podendo atingir alguns centímetros ou até metros, dependendo da potência de saída e da freqüência de rádio que está sendo utilizada. Este sinal enviado pela antena ativa o *Tag* realizando assim, a leitura ou a gravação dos dados na mesma.

Figura 2.1 – Leitor RFID

Quanto à conexão e função em relação aos outros itens do sistema, o RFID apresenta semelhanças ao leitor de código de barras. A principal diferença está no fato de o leitor de RFID não precisar estar na frente do produto para realizar a leitura, pois pode ser feita através de vários tipos de materiais e tamanho, uma vez que as ondas emitidas pela antena se propagam em diversas direções e distâncias. Quando uma *Tag* passa pelo campo de cobertura da antena, o seu campo magnético é detectado e os dados que estão na *Tag* são decodificados e enviados para o computador fazer o processamento.(SANTANA, 2005)

O tempo desta execução é inferior a um décimo de segundo, portanto o tempo de exposição necessário da Tag é bem pequeno. Resumidamente, as únicas funções da leitora são ler, gravar e decodificar os dados que estão em uma Tag que passa pela sua região de "abrangência"

b) Tag

As *Tag*s, (Figura 2.2), estão disponíveis em diversas formas: cartões, pastilhas, argolas. Além disso podem ser encapsuladas em materiais como o plástico e vidro.

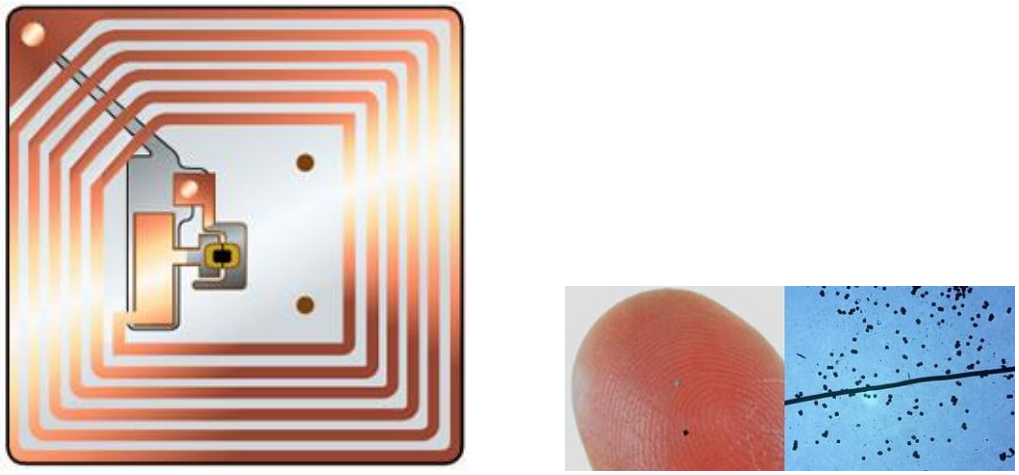

Figura 2.2 – Etiquetas (Tags) RFID

São hardwares que possuem uma antena e um chip, e respondem a sinais remotos de um leitor.

As *Tags* podem ser:

a) ativas: alimentados por bateria interna, e permitem o processo de leitura e gravação;

b) passivas: operam sem bateria e sua alimentação é feita pelas próprias ondas eletromagnéticas, usadas, geralmente, para leitura com curtas distâncias.

A faixa de freqüência que as tags atuam pode variar de acordo com a classificação abaixo:

a) sistema de baixa freqüência (30 a 500 KHz): para curta distância de leitura e baixos custos. Utilizada para controle de acesso, rastreabilidade e identificação de animais.

b) sistema de alta freqüência (850 a 950 MHz e 2.4 a 2.5 GHz): para leitura a médias e longas distâncias e a alta velocidade.

c) Computador Host

O computador host é considerado um componente a ser adaptado para ser conectado à unidade de controle para receber o software mediador. Como mediador (*middleware*) entende-se um termo genérico usado para descrever um software que se encontra entre o leitor RFID e as aplicações que a utiliza. É um componente crítico de qualquer sistema RFID, porque os *middlewares* recebem os dados originais do leitor – um leitor tem capacidade de ler até 100 vezes por segundo uma mesma etiqueta – filtrar esses dados e reenviar somente os dados úteis para os terminais. (PINHEIRO, 2004)

O *middleware* exerce um papel indispensável em levar a informação correta para a aplicação associada dentro do tempo estipulado.

.

2.2.3 Formas de Comunicação

Existem três tipos de comunicação entre o leitor e as tags. Na comunicação FDX (Full Duplex) a Tag e o leitor falam simultaneamente. Já na comunicação HDX (Half Duplex) a conexão é alternada. Nos dois casos, é o leitor que fornece toda energia necessária para a comunicação. (PINHEIRO, 2004).

Na comunicação seqüencial (SEQ), para que a comunicação seja realizada, um capacitor tem que armazenar a energia que será utilizada quando a transmissão do leitor terminar, pois a Tag transmite os dados em intervalos de tempos, também chamadas de pulsos.

2.3 Segurança com RFID

Há muito tempo, a segurança da informação tem sido prioridade em temas relacionados a investimentos na área de tecnologia, e segundo o Instituto de Desenvolvimento Cultural (IDC), este cenário não tende a mudar tão cedo. De acordo com pesquisa efetuada pelo Instituto e publicada pelo site IDG Now, a segurança foi a segunda colocada em questão de prioridade de investimento nas empresas. (COMPUTERWORLD,2008).

Informações valiosas devem ser guardadas no mais absoluto sigilo. Para evitar que dados como estes caiam em mãos perigosas, as grandes empresas vêm investindo fortemente em sistema de segurança. Servidores equipados com antivírus são utilizados para evitar a entrada de qualquer tipo de espião. Além disso treinamentos são ministrados para os funcionários, a fim de ajudar a empresa a não cair em "cilada" da segurança. (COMPUTERWORLD,2008).

Infelizmente, além dessas ameaças a que as empresas estão expostas, a proliferação de vírus também preocupa e muito, pois uma vez que uma máquina da rede está infectada, toda a rede da empresa esta ameaçada. Aqui entra um disfunção do RFID, pois apesar da tecnologia apresentar um grande avanço, ela pode trazer grandes problemas aos seus usuários.

O grande problema das etiquetas de RFID, é que elas não possuem nenhuma rotina ou dispositivo para proteger seus dados, mesmo as *tags* passivas que tem raio de atuação muito menor podem sofrer intervenções e extravio de suas informações, já com as ativas, o problema é bem maior. (COMPUTERWORLD,2008). Se o RFID tomar as proporções que estão sendo almejadas, muitas pessoas terão t*ags* em seus objetos pessoais, assim, dados pessoais poderão ser obtidos por qualquer um que possuir uma leitora de RFID.

Como medida preventiva, várias alternativas estão sendo estudadas, entre elas, a mais aceitável é a de não armazenar informações nas tags, mas sim em bancos de dados ou nas leitoras. Os sistemas de RFID são divididos em três zonas de segurança, que precisam ser analisadas e protegidas:

> a) a primeira zona compreende as próprias *tags* RFID. A possibilidade de vulnerabilidade existente consiste em como os dados são armazenados na *tag*. A grande maioria das empresas não tem por hábito encriptar os dados que estão gravados na *tag* pelo simples motivo de espaço. Gravar os dados de forma simples, pode em alguns casos permitir que sejam gravadas mais informações na *tag* que em alguns casos pode ser realmente útil;

b) a segunda zona está nas leitoras que geralmente estão conectadas a uma rede local, através de redes cabeadas ou *wireless*. Existem duas possibilidades, uma que o tráfego de dados entre a leitora e a *tag* não é criptografado ou então os leitores que não possuem um sistema de autenticação das *tags*.

Neste caso, podem ocorrer ataques como o *spoofing* (*Tag* falsa) ou então por *Denial Of Service* (DoS, Negação de Serviço). Este DoS consiste em a *tag* receber um sinal, considerá-lo válido, e quando tentar decodificar, verificar que o sinal é inválido, assim ela se reinicia para um estado de erro, o mesmo estado que ela se encontra quando é ligada (LOES, 2006).

A grande ameaça neste caso consiste em qualquer pessoa que esteja por perto, equipada de um *sniffer* (ferramenta que busca por dispositivos conectados a rede) podem capturar as informações que estão sendo transmitidas;

c) Por último, são os serviços como ONS1, Gerenciador de Eventos, EPCIS2 e o servidor de integração.(ONS: *Object Naming Service*, Serviço de Nomeação de Objetos. É um serviço da EPC Global Inc que faz a tradução de um código para a informação de um produto. EPCIS: EPC *Information Service* é um repositório de eventos de RFID EPC. O *Security Working Group* da EPC é um grupo que trabalha na segurança dos sistemas RFID com a ajuda da VeriSign e ConnecTerra.)

Diante dos fatos apresentados, conclui-se que para que possa haver uma maior confiança na utilização da tecnologia, seja usada criptografia, assim como é usada no envio e recebimento de e-mail que contenham informações confidenciais.

2.4 Padronização Internacional

A tecnologia de RFID tem se disseminado rapidamente por meio dos mais variados mercados. A necessidade de compartilhar informações relacionadas a objetos ou a pessoas torna necessário o desenvolvimento de padrões internacionais. Os padrões são importantes pois especificam as exigências de produtos, serviços, processos, materiais, sistemas, e práticas gerenciais e organizacionais, sendo projetados para serem implementados em nível mundial.

Além disso os padrões reduzem as barreiras à difusão da inovação tecnológica e facilitam a interação entre equipamentos.

2.5 O Padrão EPC

2.5.1 Os padrões da tecnologia de RFID

Os sistemas de RFID possuem vários padrões que foram desenvolvidos no decorrer das décadas para normatizar a estruturação de dados, de protocolos de comunicação entre leitor e tag, de definições de memória e de tipo de tag. (LOES, 2006)

Os padrões de RFID**,** no passado, eram quase como padrões proprietários onde o consumidor assumia fidelidade ao aderir a tecnologia criando uma relação de dependência com o fabricante da tecnologia. Os primeiros sistemas de RFID eram totalmente proprietários e devido à complexidade envolvida neste desenvolvimento, era importante se reter informações sobre o funcionamento e composição do sistema.

Com o avanço e popularização do RFID para o mercado consumidor começou a surgir à necessidade de se padronizar as interfaces de comunicação com os tags e a estrutura de memória dos mesmos, para que um cartão de acesso com RFID que fosse feito por um fabricante europeu pudesse ser utilizado com travas de portas feitas nos Estados Unidos.

Desta forma a divisão de RFID da *Texas Instruments* criou um dos primeiros padrões de RFID do mercado, o sistema de 125/134kHz. Este sistema é muito utilizado até hoje para controle de gado, controle de veículos e cartões de acesso. (LOES, 2006)

Com o avanço do RFID dentro da área de captura automática de dados, alguns outros padrões foram surgindo e nasceu a necessidade de criar um órgão normatizador que controlasse estes padrões. Esta tarefa foi agregada pela ISO que passou a regulamentar os padrões de sistema de RFID. Os primeiros sistemas de RFID regulamentados pela ISO foram os sistemas de RFID HF como o ISO15693 e ISO15443 padrões atualmente bem definidos e muito utilizados.

Observando que a tecnologia de RFID operando na faixa de UHF poderia atender as necessidades da cadeia logística, vários desenvolvedores de tecnologia começaram a desenvolver padrões de RFID que operassem nesta faixa de freqüência de forma a atender as exigências deste nicho de mercado e foram surgindo a partir do ano 2000 vários “padrões” de RFID UHF que aspiravam ser o padrão que iria normatizar toda a cadeia logística. (LOES, 2006)

Para agilizar este processo e unir os esforços de forma a se atingir um objetivo em comum de forma rápida e direta, foi criado o Auto ID center, centro de pesquisa e desenvolvimento da tecnologia de RFID que tinha como objetivo criar um padrão para as

operações de RFID, mas não só no que se diz respeito ao protocolo de comunicação entre leitores e tags ou na forma de armazenagem de dados dentro dos tags, mas sim à criação de uma nova rede de troca de informações entre indústrias – distribuição - varejo - consumidor de forma a disponibilizar informações dos produtos em qualquer ponto da cadeia. Nascia ai o EPC (*Eletronic Product Code*). (GS1Brasil)

A criação do Auto ID Center em 2000 foi o primeiro passo para a estruturação da maior revolução na identificação de produtos na cadeia logística desde o surgimento do código de barras em 1970. O Auto ID Center foi criado por fabricantes de tecnologia, o MIT (Instituto de Tecnologia de Massachussets nos Estados Unidos) e grandes usuários (empresas de bem de consumo, governo, exército e o varejo).

Deste centro de pesquisas foram realizados diversos estudos com as principais tecnologias disponíveis no momento, aspectos técnicos e comerciais foram avaliados por parte dos usuários e assim foram levantadas as principais necessidades e recursos que foram julgados pertinentes à identificação dos produtos e, depois de dois anos de estudos e esforços, foi definido que o padrão que seria utilizado nas operações logísticas do mundo seria *o EPC – Eletronic Product Code* ou código eletrônico de produto. (GS1BRASIL, 2007)

Em 2002 a GS1 se junta ao grupo do Auto ID Center para dar origem ao órgão que hoje controla toda implantação da Rede EPC e de toda padronização dos sistemas de RFID/EPC, a EPCglobal, organização subsidiária da GS1 que trata da Rede EPC.

2.5.2 Estrutura do padrão EPC

O padrão EPC não define apenas a interface entre tags e leitores, mas também uma nova estrutura de comunicação entre a cadeia logística mundial e a tecnologia de RFID, toda a conversão do sistema de código de barras GS1 (antigo Sistema EAN.UCC) para EPC e todos os ajustes da tecnologia para a criação de um padrão único.

Durante a definição do padrão EPC existiam duas tecnologias que apresentavam resultados muito satisfatórios para as necessidades levantadas pelo Auto ID center. A primeira delas foi denominada como EPC Classe 0 baseada na tecnologia de RFID da empresa Matrics (hoje *Symbol*). A segunda era o EPC Classe 1 baseada na tecnologia de RFID da empresa Alien Technology. Tendo cada uma destas tecnologias suas vantagens e desvantagens a solução mais rápida para colocar o padrão em vigência foi a divisão do padrão em classes. (GS1BRASIL, 2007)

Além das classes existiam outros componentes que formavam o que é conhecido como Rede EPC. Foi desenhada toda uma estrutura de comunicação que iria levar e disponibilizar esta informação para cada elo da cadeia e todas as trocas de informações seriam feitas de forma imediata e simultânea.

2.6 RIFIDI

O RIFIDI é um emulador destinado a desenvolvedores que desejam simular sua solução RFID em um cenário controlado, sem precisar adquirir hardware - como leitores e tags RFID. Para realizar testes, experimentos ou otimização de processos empresariais, este software simula um leitor RFID em execução.

Quando configurado, este simulador possui a capacidade de reproduzir as funcionalidades de um leitor RFID. Algumas das principais ferramentas para simulação de leitores RFID são: rifidi emulator, rifidi designer, rifidi tag, streamer e leitores.

2.7 Middleware

O *middleware* nada mais é que um software com função de gerir os dados capturados pelos leitores e também de integrar outros sistemas externos ao sistema de RFID. Uma implementação de *middleware* para um sistema RFID, de um modo geral e básico, visa a utilização de três camadas:

1. Camada de Transferência de Dados: Esta camada contém os diversos modelos de tags e leitores.
2. Camada Operacional: Esta camada intermédia contém o *middleware* que faz a integração entre os diversos leitores com os diversos sistemas existentes.
3. Camada de Negócio: Nesta camada encontra-se toda a infra-estrutura de uma empresa que utiliza o sistema RFID. Esta camada caracteriza-se por uma grande heterogeneidade devido aos diversos tipos de plataformas suportados.

O *middleware*, numa definição geral, é uma camada de software de distribuição, ou plataforma de programação, que abstrai a complexidade e a heterogeneidade do ambiente

distribuído sobre uma rede com diversas tecnologias, arquiteturas de máquinas, sistemas operacionais e linguagens de programação. (AUTOCOM, 2007). No contexto de televisão digital, esta camada está localizada entre a rede de transmissão (hardware) e as aplicações. Seu principal objetivo é fornecer um conjunto de ferramentas que possibilite a interoperabilidade entre os sistemas de transmissão de vídeo para os vários tipos de mídias de transmissão, incluindo redes terrestres, microondas, cabos e satélites.

No *middleware* são encontrados os componentes responsáveis por gerenciar os eventos captados, bem como controlar todas as fases do ciclo de vida de uma aplicação. Através dele, é possível acessar ao fluxo de vídeo, áudio e dados.

A arquitetura básica da organização dos elementos do *middleware* está dividida em camadas. A camada mais baixa da arquitetura, onde estão localizados os recursos de hardware e os de software da plataforma, onde seus elementos (placas-mãe, microprocessadores, sistemas e subsistemas operacionais em tempo real) variam de acordo com o fabricante do equipamento. (AUTOCOM, 2007).

2.8 Tecnologia Java

A tecnologia *Java* há algum tempo, tem sido a principal escolha do mercado de TI para o desenvolvimento de sistemas distribuídos. Java ainda é a linguagem mais popular para o desenvolvimento de sistemas. (SOUZA, 1999)

É importante observar que a linguagem apesar de ser a mais utilizada não se refere a melhor linguagem de programação, ou a linguagem com a qual se escreveu a maior quantidade de linhas de código até o momento. (SOUZA, 1999). Esta convergência para a tecnologia *Java* deve, principalmente, ao reconhecimento por parte do mercado de TI das qualidades inerentes da linguagem e da plataforma *Java* como um todo.

A tecnologia Java surgiu de uma pesquisa corporativa interna, financiada pela Sun *Microsystems*, em 1991. O objetivo era o desenvolvimento de uma linguagem que atendesse ao mercado de dispositivos eletrônicos inteligentes. (SOUZA, 1999).

O resultado foi a criação de uma linguagem baseada em C e C++, que teve o nome inicial de *Oak*. Pouco tempo depois, rebatizada de Java, a linguagem se mostrou adequada para o desenvolvimento de páginas web dinâmicas, e em 1995 foi apresentada formalmente pela *Sun Microsystems* em uma conferência. Nascia aí a plataforma Java.

Hoje, muito mais do que uma linguagem, Java é uma plataforma rica, que permite o desenvolvimento de aplicações para dispositivos móveis, como celulares e PDAs, até aplicações corporativas complexas, baseadas em *web services*, passando ainda por aplicações *desktop* e discos *blue-ray*. (SOUZA, 1999).

2.8.1 A linguagem de programação *Java*

Ao criar a linguagem de programação Java, os engenheiros da Sun se basearam em duas das linguagens de implementação mais utilizadas até então, C e C++. Isso permitiu que Java estivesse facilmente acessível a uma enorme base de desenvolvedores ao redor do mundo, a maioria dos quais envolvidos no desenvolvimento de sistemas operacionais, sistemas de bancos de dados, telecomunicações e aplicativos para computadores pessoais. (YUAN, 2005).

Foram removidos da linguagem os recursos mais confusos, complexos, e propensos a erros encontrados em C e C++, mantendo assim a linguagem concisa, com aquilo que havia de melhor das duas linguagens.

Foram incluídos na linguagem recursos realmente necessários à grande maioria dos desenvolvedores, como strings, imagens gráficas, componentes de interface gráfica com o usuário, tratamento de exceções, *multithreading*, multimídia, processamento de arquivos, processamento de bancos de dados, redes cliente/servidor baseados na Internet e na *World Wide Web* e em computação distribuída, e estruturas de dados pré-empacotadas. Java foi também a primeira linguagem verdadeiramente multiplataforma, implementando o conceito de "*write once, run everywhere*", graças à sua arquitetura baseada em uma máquina virtual, que interpreta um código intermediário, conhecido como *bytecode*. (YUAN, 2005).

A existência de máquinas virtuais para sistemas operacionais específicos permite que um mesmo *bytecode* gerado por um código compilado de uma classe Java seja executado em qualquer sistema operacional sem a necessidade de recompilação.

Desde sua criação, muitas características vem sendo adicionadas à linguagem Java e à plataforma Java de maneira geral Todas estas características tornam a linguagem Java adequada para o desenvolvimento de uma ampla variedade de sistemas, que incluem aplicações desktop, aplicações multimídia, *applets*, aplicações para Internet, intranet, extranet, portais, aplicações distribuídas baseadas em *web services*, aplicações para dispositivos móveis, discos *blue-ray*, aplicações para tv digital, entre outros.(YUAN, 2005)

2.8.2 A plataforma tecnológica *Java*

A tecnologia Java não está limitada somente à linguagem de programação Java. Antes disso, Java é uma ampla plataforma de desenvolvimento, constituída de várias APIs e ambientes de execução. (YUAN, 2005)

Apesar de ter sido desenvolvida inicialmente pela Sun, hoje a evolução da especificação da plataforma Java é determinada por uma comunidade de empresas e indivíduos denominada JCP (*Java Community Process*) que, com suas experiências, ajudam a definir os rumos da plataforma, contribuindo com o melhor de cada segmento.

Este modelo de evolução da plataforma permite que qualquer empresa possa implementar a especificação da plataforma, na forma de um produto.

A plataforma é dividida em três segmentos, ou edições principais: JSE, JEE e JME. Cada edição engloba e licencia um conjunto de APIs e ambiente de execução da plataforma Java para atender às necessidades específicas dos desenvolvedores de aplicações.

A JSE, ou *Java Standard Edition*, é o segmento base da plataforma Java. É nele que encontramos as principais APIs a plataforma, que servem de base para os outro dois segmentos, além de APIs para o desenvolvimento de aplicações desktop. (YUAN, 2005).

A figura 2.3 apresenta o segmento JSE da plataforma Java.

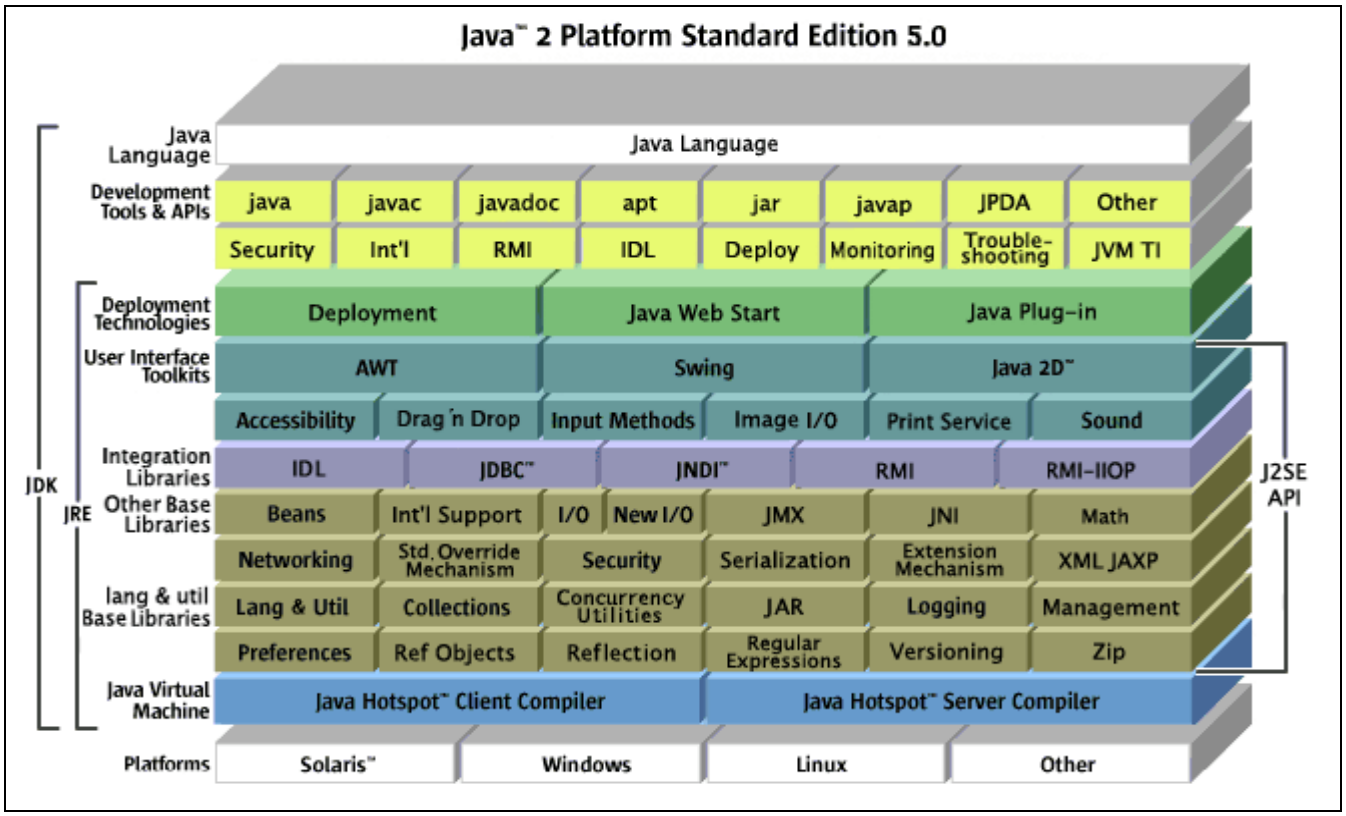

Figura 2.3 – JSE da plataforma Java

Como pode ser visto, a linguagem Java está no topo de uma série de tecnologias que compõem a plataforma JSE.

É possível visualizar também, além das APIs que compõem a plataforma JSE, as ferramentas que acompanham o kit de desenvolvimento (JDK) e as tecnologias de implantação que acompanham ambiente de execução (JRE).

A JEE, ou *Java Enterprise Edition*, é o segmento da plataforma Java que apresenta APIs para o desenvolvimento e aplicações corporativas distribuídas, transacionais, baseadas principalmente em tecnologias web. (YUAN, 2005)

A figura 2.4 ilustra a distribuição das APIs da plataforma JEE de acordo com o tipo de container JEE.

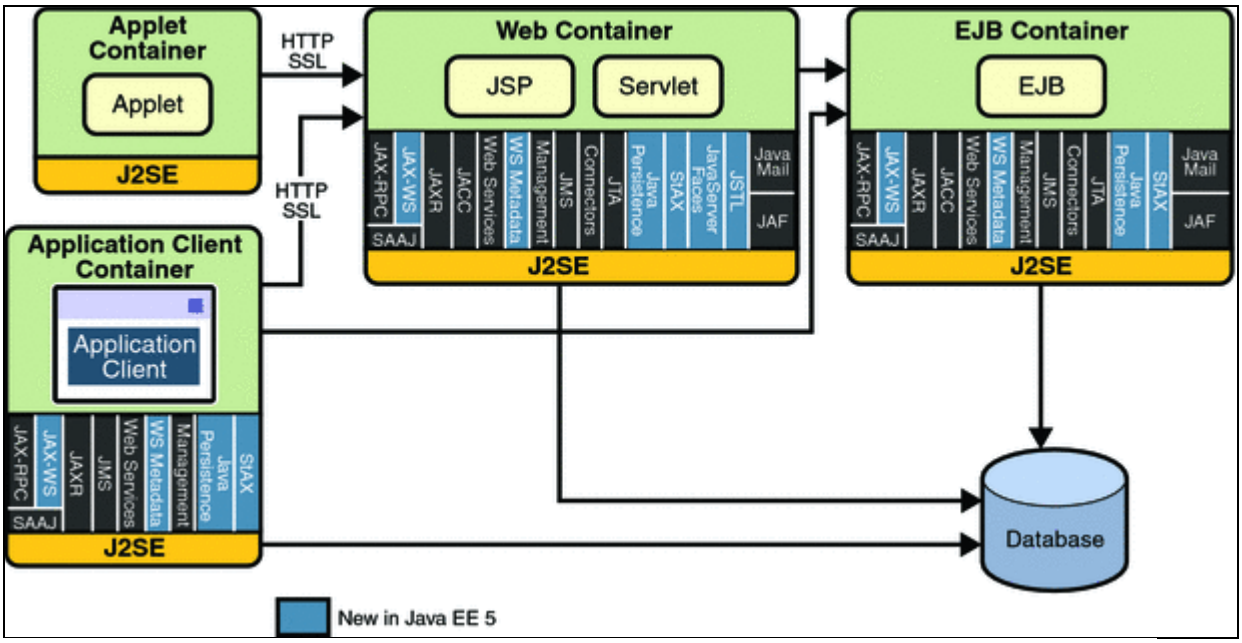

Figura 2.4 – Distribuição das APIs - Plataforma JEE

A plataforma JSE é a base para todas as outras tecnologias presentes na plataforma JEE. A JME, ou *Java Micro Editon*, é o segmento da plataforma Java destinado a dispositivos móveis como celulares, PDAs, e outros dispositivos embarcados que suportam Java, mas não toda a API JSE ou JEE, e de certa forma é porção da tecnologia Java que vai ao encontro das intenções originais ao se criar a linguagem Java.

Atualmente, a plataforma JME evoluiu para uma arquitetura organizada de tecnologias distintas para dispositivos embarcados, e está dividida conforme a figura 2.5.

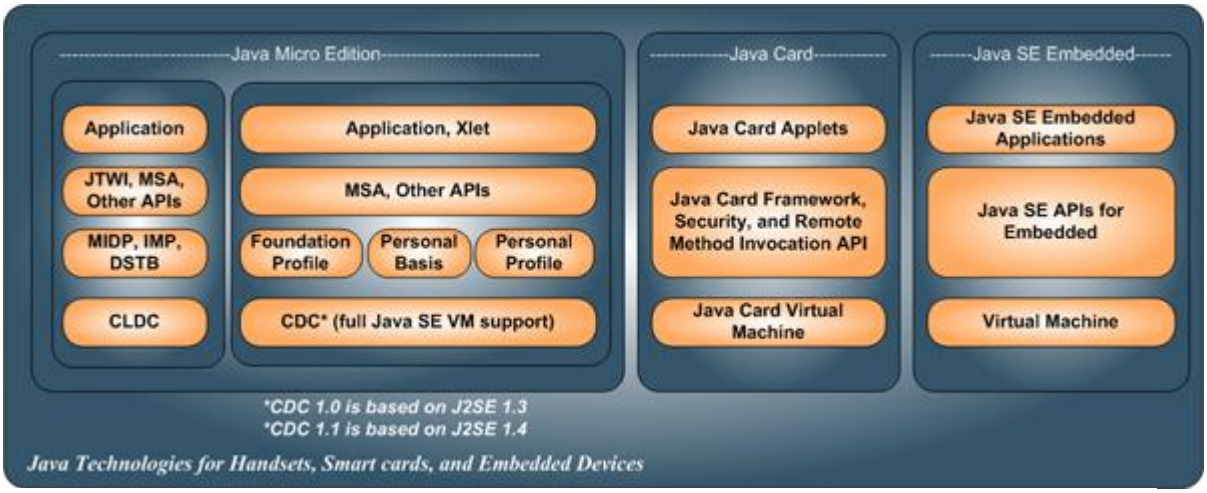

Figura 2.5 – Arquitetura da Plataforma JME

2.9 Banco Oracle 10g Express Edition

O Oracle 10g representa uma das versões mais complexas e sofisticadas desse produto. O Banco de Dados Oracle XE (versão gratuita) oferece mobilidade para desenvolver e implementar aplicativos de várias plataformas e traz suporte para uma variedade extensas de ambientes de desenvolvimento. Muitos programadores *Java*, *.NET*, *PHP* e da *Web*, incluindo estudantes e fornecedores de soluções de terceiros, já baixaram o banco de dados da internet a custo zero. (MORALES, 2009)

O Oracle XE tem recursos de desempenho, confiabilidade e segurança da versão 2 do Oracle 10g, pelo fato de ter sido construído sobre a mesma base de código. Por ser compatível com toda a família de produtos do Oracle, ele permite aos usuários a facilidade de iniciar com uma solução básica e ir alterando para outras versões quando necessário. (LONEY, 2005). Permite ainda que os desenvolvedores explorem o *Oracle Application Express* para rápido desenvolvimento e implementação de aplicativos baseados na *Web*.

Há compatibilidade completa entre todas as versões Oracle 10g, ou seja, uma *package* desenvolvida para ser executada no Oracle XE, será executada da mesma forma no Oracle *Enterprise Edition*. As limitações da versão supracitada são: memória, *storage*, processador e plataforma. (MORALES, 2009).

O Oracle 10g XE oferece suporte integral à linguagem *procedural* PL/SQL. Isso permite ao desenvolvedor criar a lógica de negócios no próprio banco de dados, por meio de procedimentos, funções, pacotes e *triggers*. Além disso, há suporte nativo para manipulação de objetos, expressões regulares, PL/SQL *server pages* e o conjunto de ferramentas de desenvolvimento voltado para o Visual Studio .Net.(LONEY, 2005)

Porém, ainda não há suporte para a construção de *procedures* em Java, no entanto, toda a lógica de negócios pode ser desenvolvida no banco de dados por meio de *procedures* escritas em PL/SQL.

A expressão “SQL:1999” refere-se às notações SQL:1999 dos padrões ANSI e ISO/IEC, oficialmente conhecidos como “ISO/IEC 9075-1:1999”. O Oracle XE está totalmente de acordo com esses padrões. (LONEY, 2005). A conformidade com a sintaxe SQL:1999 é importante pois além dos SGBD’s orientados a objeto, existem também os Objeto relacionais que misturam banco de dados relacional com conceitos de orientação a objetos. Um exemplo é o Oracle 10g que usa SQL no sistema Objeto Relacional.

Existe no Oracle o *Object Type* que um tipo definido pelo usuário na qual equivale ao de classe em POO (Programação Orientada a Objeto). O Object *Type* captura tanto a estrutura como o comportamento de um objeto. (LONEY, 2005)

2.9.1 Vantagens no uso do Oracle

Entre as vantagens do Oracle, destaca-se a capacidade de armazenamento de objetos, poder de processamento de requisições. (LONEY, 2005). Por não possuírem chaves primarias nem estrangeiras, o desempenho de consultas e processos aumenta e os objetos se comunicam entre si através de mensagens.

2.9.2 Desvantagens no uso do Oracle

Como desvantagens, cabe ressaltar a falta de padronização das linguagens de manipulação dos dados, alto custo de aquisição das novas tecnologias além de apresentar a curva de aprendizagem e adaptação ao novo ambiente demorada. (LONEY, 2005).

Como pode ser visto, as junções das tecnologias supracitadas vai muito além de uma plataforma multidisciplinar. É , na verdade, um arcabouço completo, um conjunto de alternativas para solucionar o problema apresentado, através do desenvolvimento de uma aplicação simples e completa.

Apesar das técnicas que já existem, a solução proposta representa uma alternativa que merece destaque no mercado de desenvolvimento de aplicações RIFIDI, e as constantes melhorias devem enriquecer e tendem a consolidar ainda mais a tecnologia sugerida, durante os próximos anos, como uma das principais do mercado.

3 – DESENVOLVIMENTO DO TRABALHO

Neste capítulo são apresentadas as características funcionais do sistema, os diagramas e o detalhamento da estrutura do projeto, seu hardware e seu software.

No primeiro capítulo foram apresentados os objetivos e motivações para a elaboração do projeto, no segundo foi apresentado o embasamento teórico para o entendimento do problema e da solução proposta.

Neste capítulo o problema e a solução são detalhados em maior profundidade.

3.1 Requisitos Principais do Problema a ser Trabalhado

No cenário em que está inserido o projeto, as tags e leitoras estão embutidas no emulador RIFIDI equipado com dois módulos: módulo de controle: trata das questões relativas as configurações e inicialização. E módulo de comunicação: tem o papel de tratar da comunicação com os leitores contidos no RIFIDI, através da integração com o *middleware* e com a aplicação cliente.

A interface do usuário será projetada de forma modular a adaptar-se às habilidades e interesses do usuário. A aplicação está configurada para abstrair detalhes de implementação para recuperação das informações na base de dados.

Como exemplo é possível citar o uso do sistema por bancos, com uma série de cédulas duvidosas, que apresentam dificuldades em entender se são ou não verdadeiras.

Neste caso, uma vez identificado os atributos e suas características, a interface é capaz de fazer uma leitura das tags através de funcionalidades contidas no módulo de comunicação.

Como o sistema lida com instituições financeiras, não se pode tomar por garantida a hipótese de que as notas de dinheiro. Se o executor cometer um engano ao escolher a etiqueta de identificação, o sistema identificará o nota errada e, poderá ficar incapacitado de continuar interagindo com o mesmo. É por este motivo que existe a necessidade de identificar corretamente as cédulas.

Serão descritas a seguir aqui, as principais técnicas e ferramentas desenvolvidas para simulação de leitores RFID.

3.1.1 Requisitos Funcionais

Com base na descrição da aplicação, especialmente na descrição de seus serviços, é possível listar seus requisitos funcionais. Eles guiam os passos conseguintes de desenvolvimento da aplicação e de seus testes.

3.1.1.1 Sistema Único de Identificação de Dinheiro - SUID através de etiquetas RFID

a) Permitir a inicialização da rotina de detecção de tags através da aplicação SUID;

b) Permitir a paralisação da rotina de detecção de tags através da aplicação SUID;

c) Assegurar a correta identificação dos atributos das tags uma vez que suas características forem detectada pelo emulador RIFIDI;

d) Permitir a detecção da tag por meio da aproximação entre ela e a antena que se encontra instalada.

3.1.1.2 Sistema SUID - Middleware

a) Permitir o controle e leitura de uma tag (cédula de dinheiro) conectada ao leitor sob demanda de uma interface feita em Java;

b) Permitir o controle e leitura de um conjunto de tags previamente configuradas e conectadas ao leitor sob o controle configurado no sistema SUID, desenvolvido em Java e conectado ao simulador RIFIDI (antena).

3.1.2 Requisitos não funcionais

Os requisitos não funcionais são de grande importância e servem para garantir que, apesar das restrições de tecnologia, os níveis de serviço da solução sejam garantidos.

Abaixo são informados os requisitos não-funcionais do Sistema Único de Identificação de Dinheiro - SUID.

3.1.2.1 Desempenho

A verificação de aproximação da cédula, deve ocorrer com baixa latência de modo que seja percebida como instantânea quando ligada a um leitor.

3.1.2.2 Confiabilidade

Relacionado a confiabilidade da aplicação é possível citar a importância de que os dados de identificação da tag passados através do emulador RIFIDI até a aplicação sejam verificados antes de processados a fim de garantir a correta identificação. Este requisito é especialmente importante uma vez que a tecnologia por rádio freqüência está sujeita a diversas interferências provenientes do ambiente externo.

3.1.2.3 Manutenibilidade

O sistema deve permitir a inclusão, listagem e remoção de dados das tags sem a necessidade de recompilação do módulo de identificação para facilitar a manutenção da aplicação.

3.2 Especificação

A especificação do aplicativo Sistema Único de Identificação de Dinheiro (SUID) se dará através dos diagramas da *Unified Modeling Language* (UML):

- Diagrama de casos de uso;
- Diagrama de classes;
- Diagrama de atividades.

Para a construção dos diagramas foi utilizada a ferramenta *Violet Uml* Editor.

3.2.1 Diagrama de casos de uso

Diagramas de caso de uso mostram conceitualmente o conjunto de funções que o sistema deve executar para atender aos requisitos do cliente, servindo também, como um

contrato entre o cliente e o desenvolvedor. No caso de uso, o sistema é visto com a perspectiva do usuário.

3.2.1.1 Cadastro de usuários

A figura 3.1 apresenta um detalhamento do caso de uso 01, apresentado na figura 3.2

UC01: Cadastra usuário	
Resumo	Administrador cria diversos usuários.
Seqüência de ações	1. Cadastra usuário. 2. Atribui perfil. 3. Atribui as permissões de acesso. 4. Grava os dados no banco.

Figura 3.1 – Detalhamento do caso de uso 01

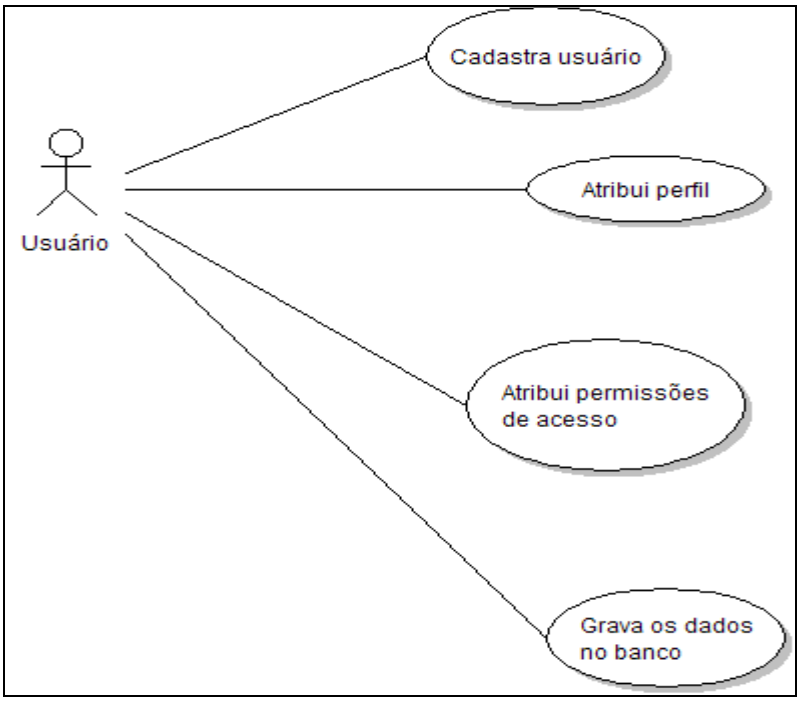

Figura 3.2 – Caso de uso 1 - cadastra usuário

3.2.1.2 Cadastro de tags

A figura 3.3 apresenta um detalhamento do caso de uso 02, apresentado na figura 3.4

UC02: Cadastra tag	
Resumo	Administrador cadastra diversas usuários.
Seqüência de ações	1. Usuário informa o valor. 2. Usuário informa o serial. 3. Usuário grava os dados no banco.

Figura 3.3 – Detalhamento do caso de uso 02

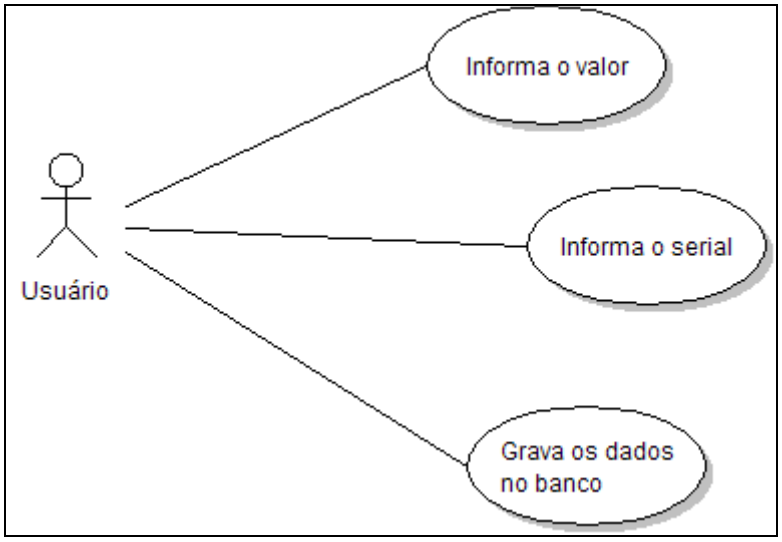

Figura 3.4 – Caso de uso 2 - Cadastra tag

3.2.1.2 Leitura da tag (Nota)

A figura 3.5 apresenta um detalhamento do caso de uso 03 apresentado na figura 3.6

UC03: Leitura da tag(Nota)	
Resumo	Usuário aproxima o cartão do leitor para realizar a leitura dos dados.
Seqüência de ações	1.Usuário aproxima um grupo de tag`s ao leitor 2. Usuário tem a opção de informar as notas que serão lidas 3. Usuário solicita a leitura da tag`s 4. Usuário verifica os resultados, valores e notas lidas
Exceção	1.No passo 3 quando for lido uma tag que não esteja no banco será apresentado uma mensagem de alerta. 2.No passo 4 ao analisar e verificar diferença no resultados o usuário deverá analisar manualmente a tag

Figura 3.5 – Detalhamento do caso de uso 03

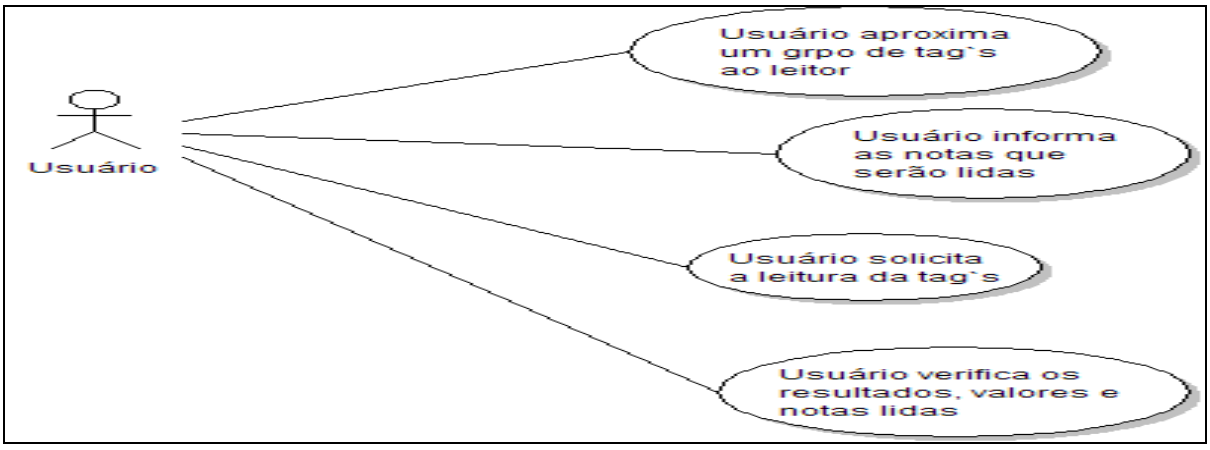

Figura 3.6– Caso de uso 3 - Leitura da tag(Nota)

3.2.1.2 Login

A figura 3.7 apresenta um detalhamento do caso de uso 04 apresentado na figura 3.8

UC04: Efetua login	
Resumo	Usuário se conecta no sistema.
Seqüência de ações	1.Usuário informa seu usuário e senha. 2. O sistema faz a verificação para saber quais as permissões deste usuário, habilitando assim somente aquele cujo cadastro foi encontrado.
Exceção	No passo 1 o usuário não está cadastrado no sistema, ou o apelido e senha foram digitados de forma incorreta. O sistema apresenta uma mensagem “Login não aceito!”“

Figura 3.7 – Detalhamento do caso de uso 04

Figura 3.8 – Caso de uso 4 – Efetua Login

3.2.2 Diagrama de classes

Um diagrama de classes mostra a estrutura estática do modelo, em que os elementos são representados por classes, com sua estrutura interna e seus relacionamentos.

A figura 3.9 apresenta as classes componentes da aplicação SUID, proposta neste estudo.

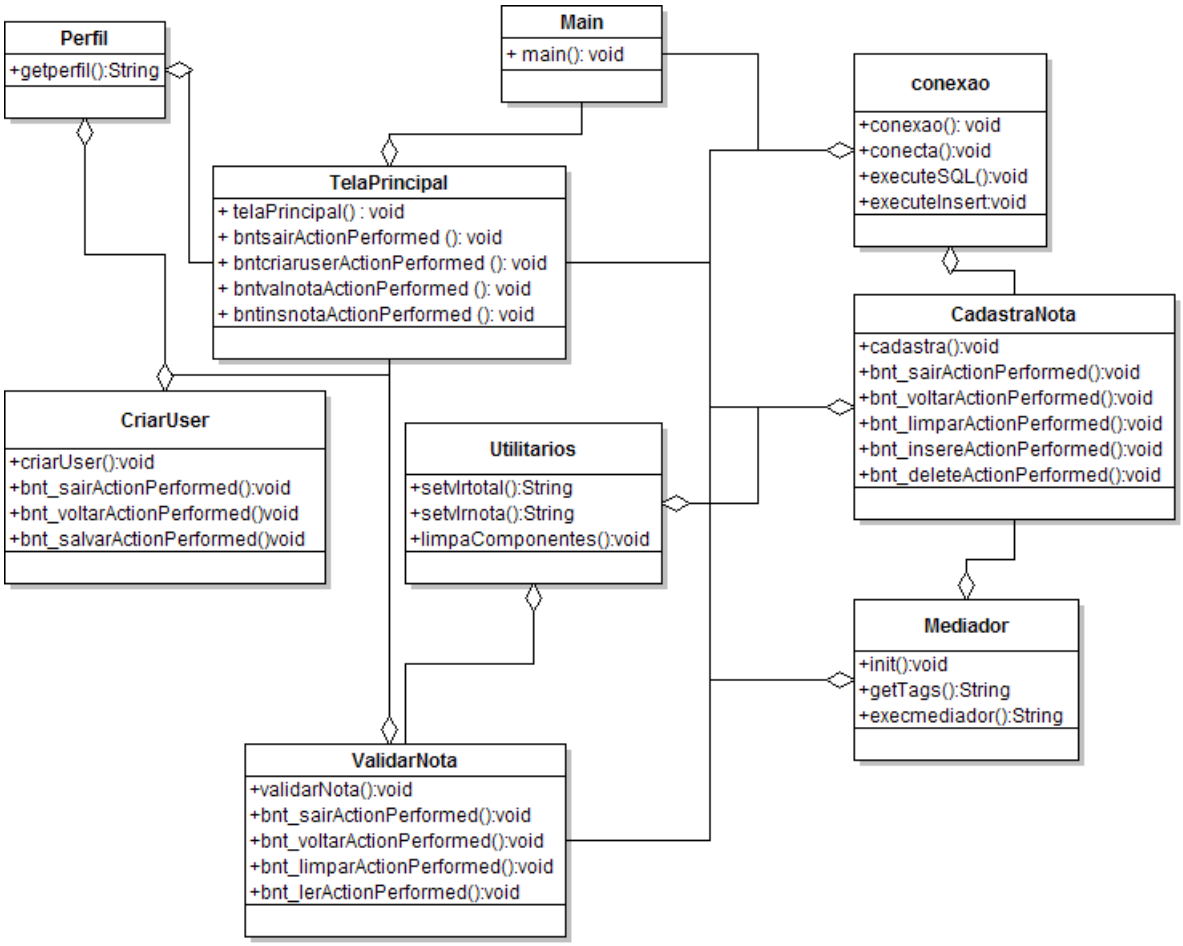

Figura 3.9 – classes componentes da aplicação SUID

3.2.3 Diagrama de atividades

O diagrama de atividades apresenta um tipo especial de diagrama de estados, ou seja, os estados de uma atividade. Estes diagramas são orientados a fluxos de controle. Um diagrama de atividades permite modelar o comportamento do sistema, demonstrando os caminhos lógicos que um processo pode seguir.

3.2.3.1 Login

Na Figura 3.10 são demonstrados através de um diagrama de atividades todas as etapas para que um usuário se conecte ao sistema.

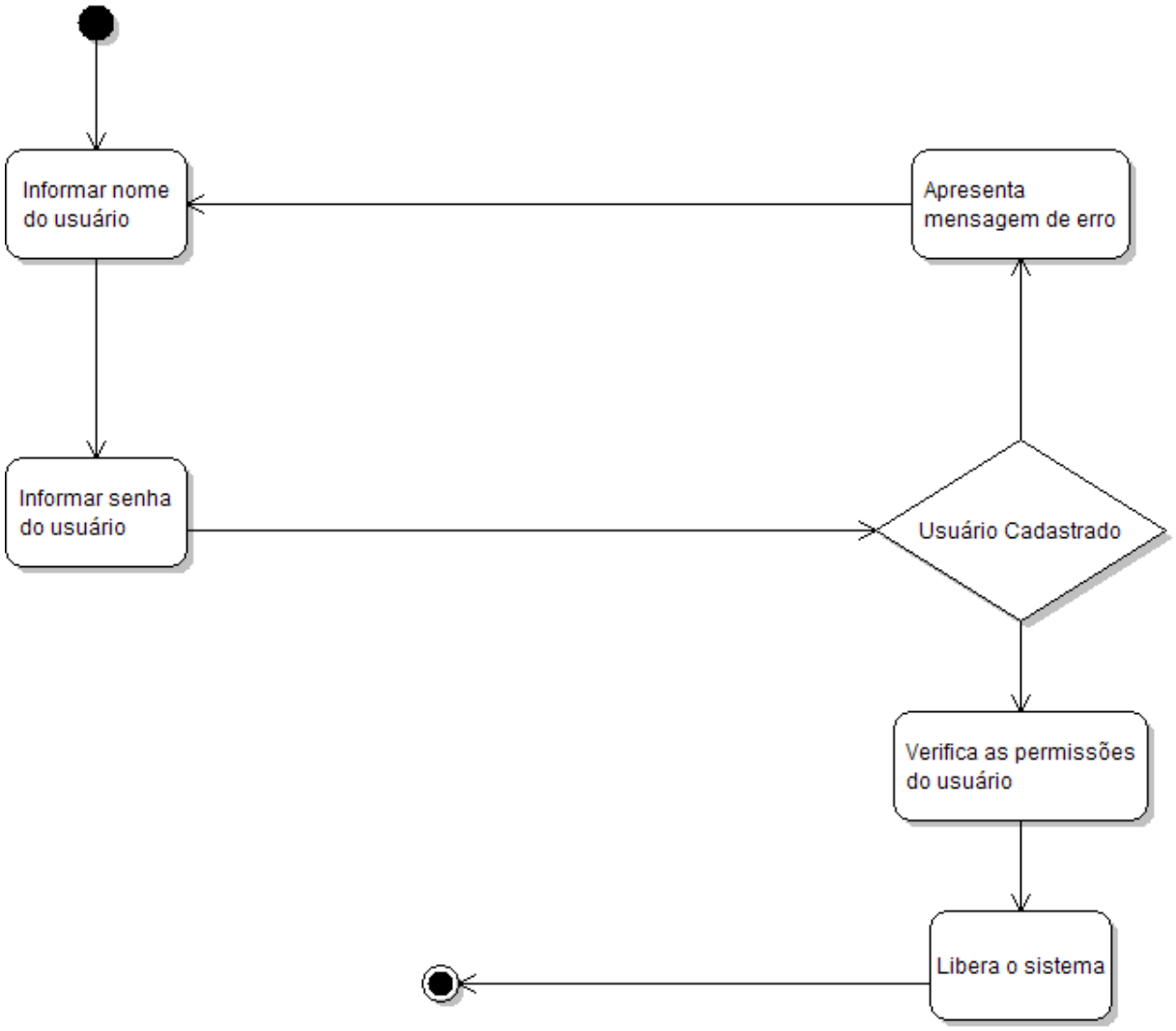

Figura 3.10 - Diagrama de atividades – Login

Através da figura é possível observar que após a leitura do nome e da senha, o sistema verifica se o usuário está mesmo cadastrado, e se a senha digitada confere com a que está no cadastro. Se todos os dados estiverem corretos, o cadastro é lido e as informações são validadas e finalmente o acesso é liberado. Caso a senha ou o apelido tenham sido digitados erroneamente, o sistema assume como usuário não cadastrado e irá apresentar a mensagem "Usuário não cadastrado!".

3.2.3.2 Leitura da Tag

Na Figura 3.11 são demonstradas através de um diagrama de atividades todas as etapas necessárias para efetuar a leitura de dados a partir da leitura da tag.

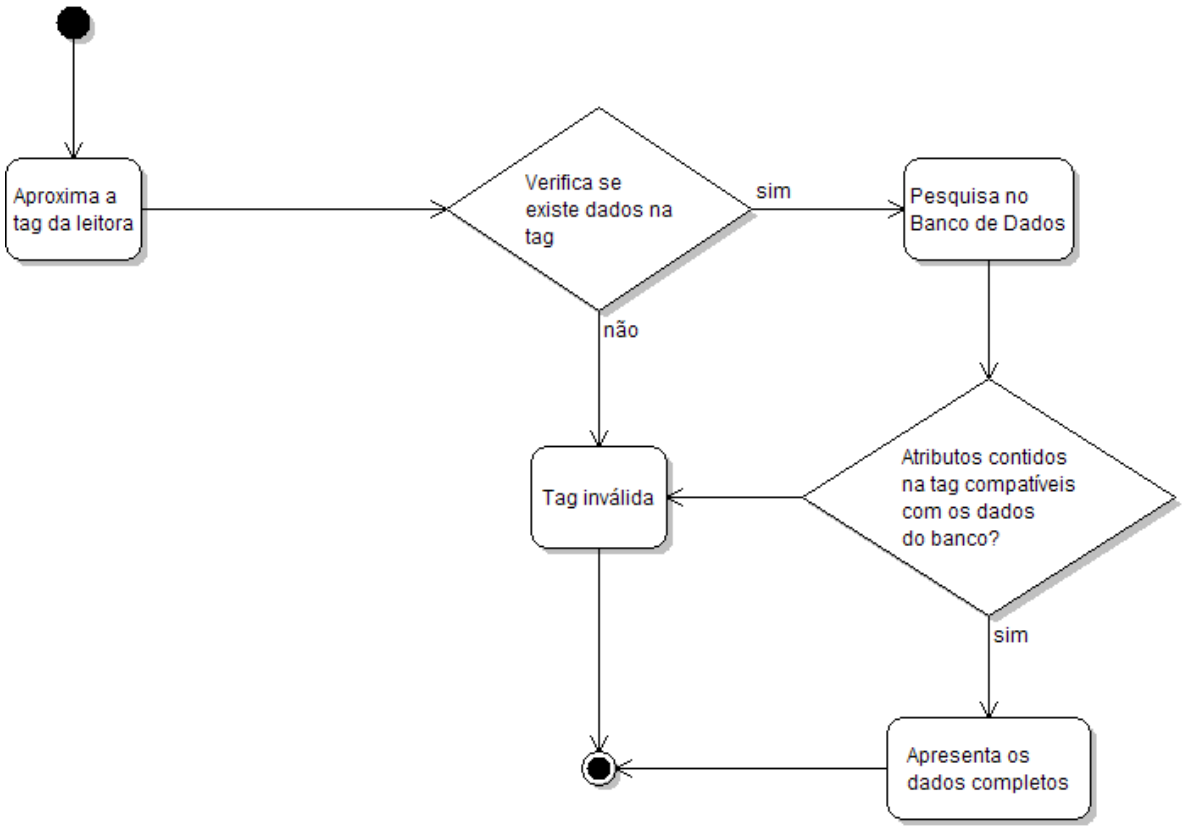

Figura 3.11 – Diagrama de atividades

Observa-se que o primeiro passo é aproximar a tag da leitora para que ela possa capturar seu valor (código). Com esta identificação, o sistema irá checar o cadastro da na base de dados, se encontrar irá verificar a compatibilidade das informações contida na etiqueta e apresenta as mesmas em tela.

Caso os dados sejam divergentes, a tag é considerada inválida, ou seja, a nota é falsa. Os dados apresentados variam de acordo com as permissões do usuário e caso não existam dados gravados na base com estes códigos, o sistema apresenta uma mensagem de erro.

3.3 Modelo de banco de dados

Na figura 3.12, está apresentado o modelo de entidade relacionamento do banco de dados utilizado neste trabalho. Existem duas tabelas, que são usadas para o controle de acesso as

informações no sistema, mas elas não estão interligadas a nenhuma outra tabela. Adicionalmente existe outra tabela para validação da nota, mas dentro de um só modelo.

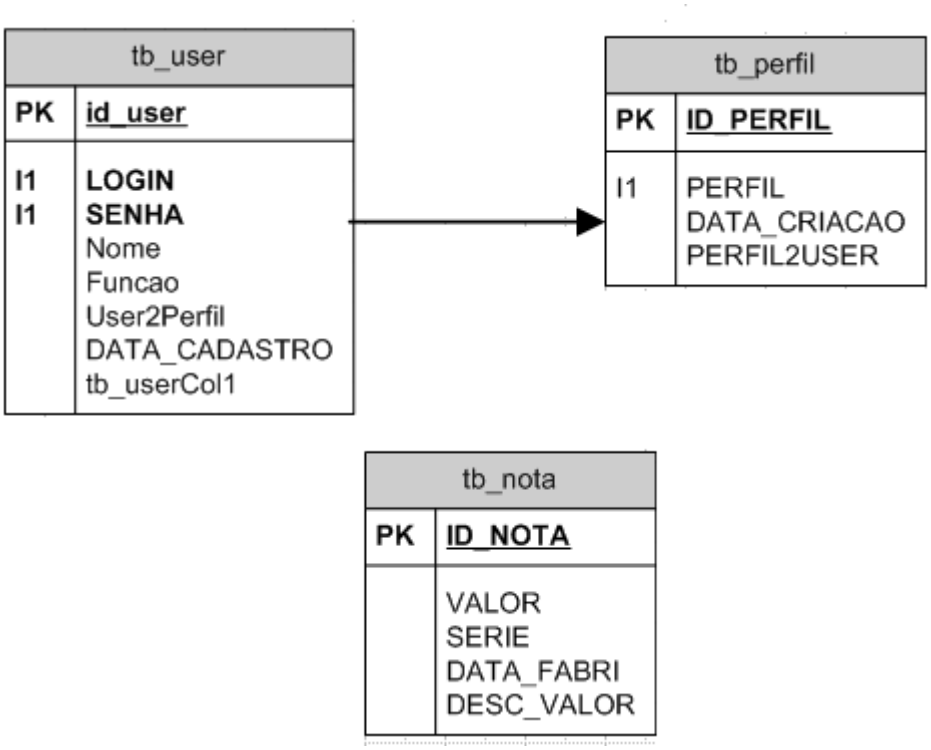

Figura 3.12 – Modelo de Banco de Dados

Neste capítulo, foi apresentado uma visão geral do projeto, suas especificações, descrição funcional, ferramentas utilizadas, dimensionamento e a utilização dos módulos de controle e validação do emulador.

Quanto à implementação do sistema, a conectividade ao simulador e a operacionalidade do protótipo desenvolvido será tratado no próximo capítulo a fim de mostrar de forma mais interativa o funcionamento da aplicação.

4 – MODELO PROPOSTO

4.1 Apresentação Geral do Modelo Proposto

Este capítulo destina-se a apresentar de forma detalhada a aplicação renomeada de SUID – Sistema Único de Identificação de Dinheiro -, demonstrando como este foi construído, bem como sua interação com outros elementos do emulador, com base na solução proposta no presente trabalho.

Também e feita, neste capítulo, uma descrição detalhada do *Emulator* RIFID, utilizado no desenvolvimento deste trabalho, demonstrando seus componentes, recursos, e especificações, além de um breve descritivo de outras iniciativas RFID de mercado, de forma superficial, apenas para demonstração.

E, como prova do funcionamento da solução foi desenvolvida uma aplicação de identificação e controle de notas de dinheiro, representadas por tags, onde o processo de comunicação entre este e a biblioteca Java se estabelece através de seu uso.

4.2 Descrição das Etapas do Modelo

4.2.1 Funcionamento da Aplicação

Esta seção tem o objetivo de demonstrar o funcionamento da aplicação desenvolvida. O SUID é uma *interface* Java a qual está intimamente relacionada ao simulador RIFIDI. Este funciona como um elemento de "ligação" entre o código da aplicação e a interface usuário, ou seja, uma espécie de interlocutor entre esses elementos. Na figura 4.1 é possível visualizar o relacionamento do *middleware*, a interface e a aplicação propriamente dita.

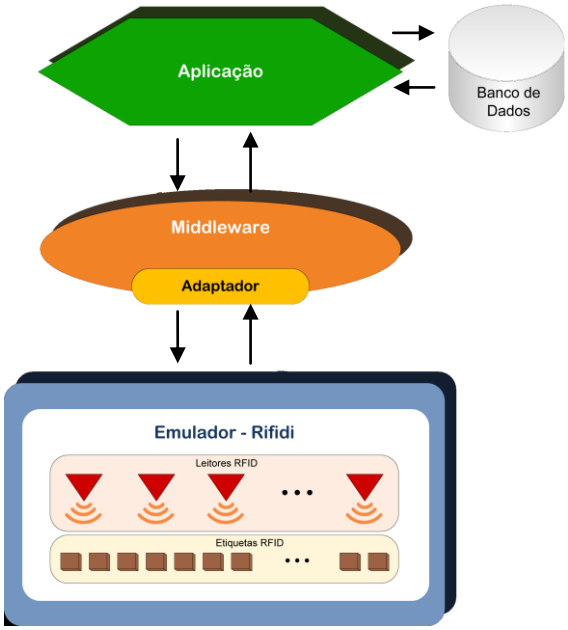

Figura 4.1 – Arquitetura SUID

Em sua estrutura estão instanciadas classes do Java constantes em sua biblioteca, útil nos retornos dos resultados da maioria dos métodos através do mapeamento realizado por meio do módulo de controle.

Logo, o objetivo da criação do sistema é estender a capacidade de desenvolvimento de aplicações RFID por meio do RIFID visando com isso a sua disponibilização para a comunidade, seja ela acadêmica ou profissional.

Para demonstrar o funcionamento do SUID será utilizada "*snapshots*", ou seja, instantâneos, das telas do sistema em suas principais situações, como, por exemplo, cadastro e alterações de dados dos usuários, cadastros e leitura das tags, situações, estas, onde se necessita do uso da linguagem Java para obtenção dos atributos armazenados nas tags RFID e que serão captados através do transceptor contido no RIFIDI.

4.3 - Descrição da Implementação

4.3.1 Operacionalidade da implementação

Nesta seção é apresentada a seqüência de telas e operações que um usuário deve fazer para que consiga utilizar o SUID. Inicialmente, para que o sistema consiga localizar a base de dados, é preciso estar com a ferramenta de SGBD instalada na máquina, e a base de dados dentro da pasta criada e escolhida para instalação

Posteriormente faz-se necessário que ela seja registrada com o seguinte nome “SUID – Base de Dados” em fonte de dados (SGBD), para isso deve ser instalado na máquina o *driver* do Oracle XP.

4.3.2 Usando o SUID

A seguir serão apresentados alguns procedimentos comuns ao uso do sistema.

4.3.2.1 Logar no sistema

Na tela apresentada na Figura 4.2, é onde o usuário do sistema informa seu nome de usuário e senha.

Figura 4.2 – Logar no sistema

Informado o nome de usuário e senha, o usuário clica em OK. O sistema irá verificar se o usuário existe, caso seu cadastro não seja encontrado irá apresentar a mensagem como mostra a Figura 4.3. Caso a senha informada seja diferente da que consta no cadastro do usuário, será apresentada a mensagem de “Senha inválida!”.

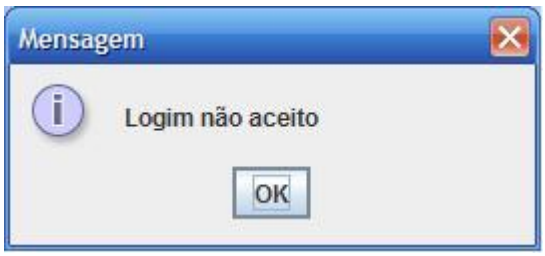

Figura 4.3– Usuário não cadastrado.

Caso esteja cadastrado e a senha estiver correta, o sistema é liberado para o uso.

4.3.2.2 Tela Interação

Ao carregar a aplicação será mostrada a tela interação como visto da figura 4.4, esta tela é o ponto inicial da aplicação. Onde o usuário pode escolher qual processo executar (Vincular Nota, Criar Usuário e Validar Nota).

Logo após o usuário apertar algum dos botões, será apresentada a tela correspondente.

Figura 4.4 – Tela Interação

4.3.2.3 Vincular Nota

A tela de Vincular Nota permite que o usuário possa adicionar etiquetas a as notas correspondentes. A aplicação permite que o usuário prossiga após ter inserido a etiqueta ou cancele a operação neste momento, voltando assim para a tela interação, como mostra a figura 4.5.

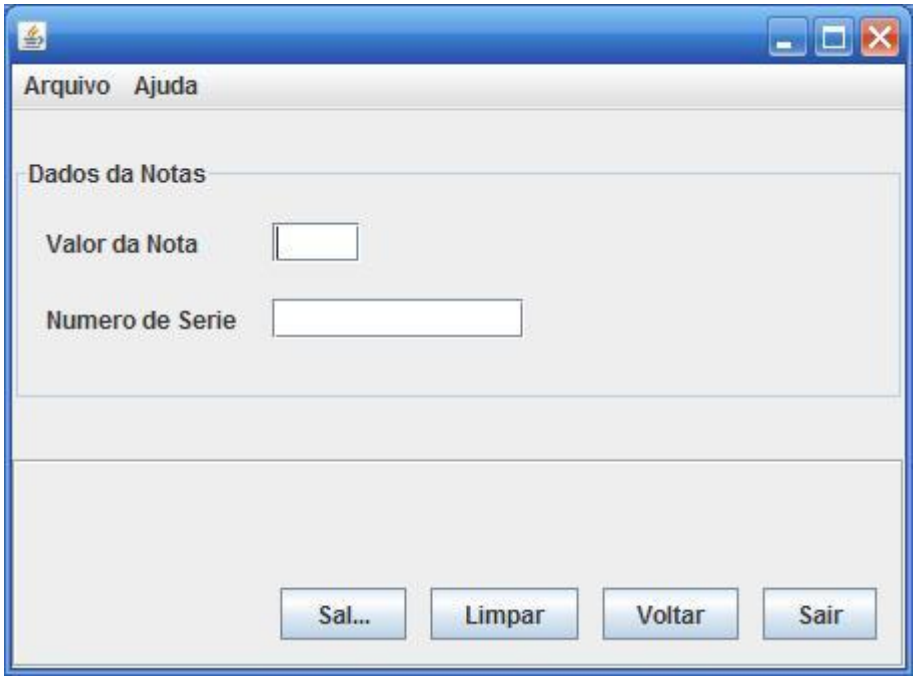

Figura 4.5 – Vincular Nota

4.3.2.4 Validar Notas

A tela Validar Notas Permite que o usuário confirme se as notas analisadas são verdadeiras, ao clicar o botão “Ler”.

O usuário tem a opção de inserir a quantidade de nota por valor, facilitando, assim, a identificação de notas falsas ou com defeito de fabricação. Será possível, ainda, confrontá-las com as informações relatadas pelo sistema, que após ter realizado a leitura, o sistema retorna informações das notas em uma *grid*, informando a quantidade de nota lidas, o valor total lido e matriz quantas notas lidas x por valor.

Caso a nota lida que tenha uma tag, não seja encontrada no banco de dados, uma mensagem de alerta será emitida indicando que no montante pesquisado existem notas que possivelmente serão falsas, como trás a figura 4.6.

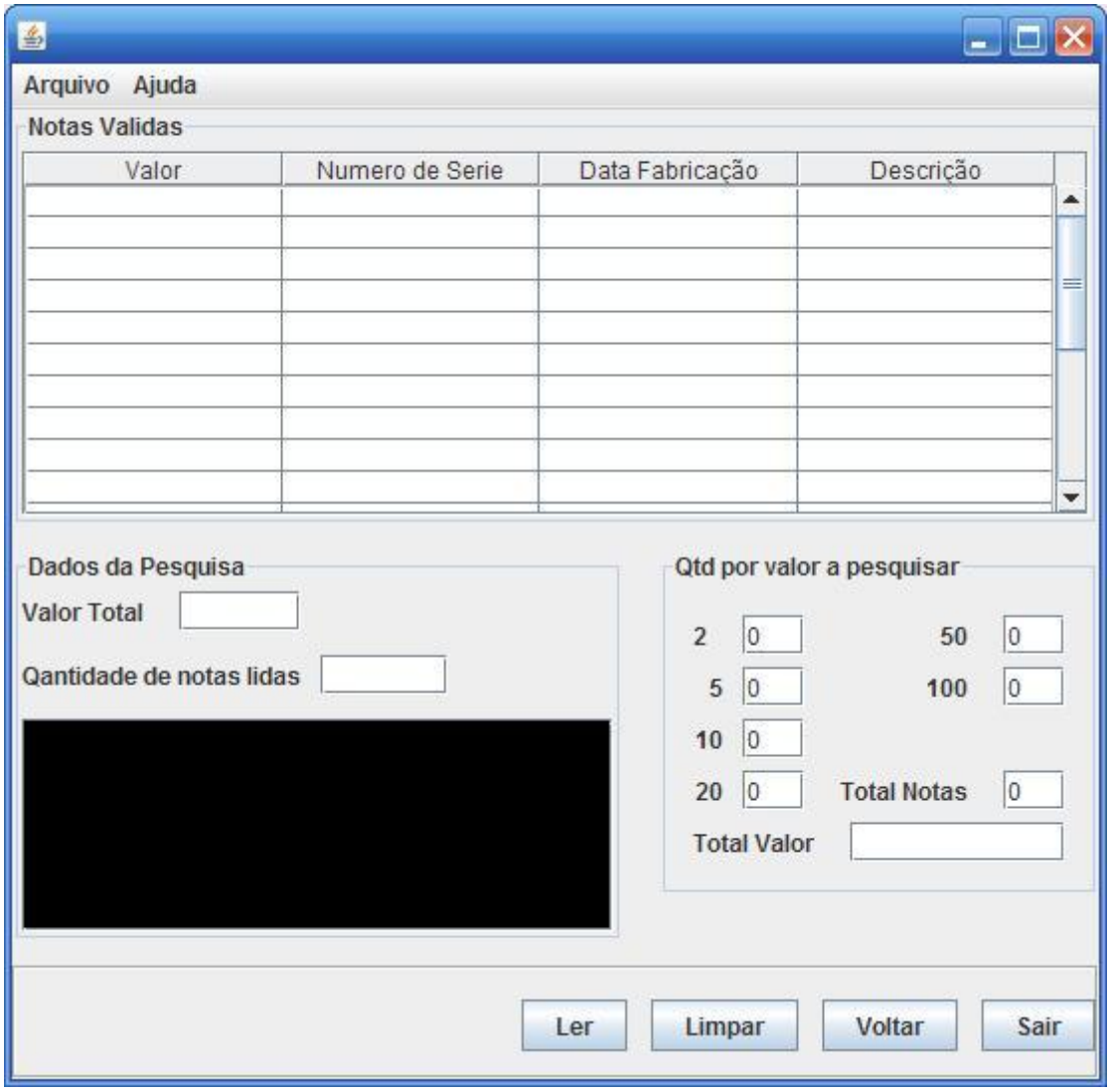

Figura 4.6 – Validar Notas

4.3.2.5 Criar Usuários

O cadastro dos usuários é realizado, conforme mostra a Figura 4.7 na tela Criar Usuário. O cadastro é feito através da indicação das informações do usuário com CPF e NOME, LOGIN e SENHA, perfil e função de um usuário e conseqüentemente seu registro no banco de dados.

Figura 4.7 – Criar Usuários

4.3.2.6 Características da Aplicação

Para que o sistema possa funcionar corretamente, é necessário ativar a leitora e cartões de RFID. E para isso também se faz necessária à comunicação através da SUID que se interconecta com a leitora através da classe mediador para obter as informações que são enviadas pela leitora e depois acessa o banco de dados para validar a tags (Notas).

O código fonte da comunicação está demonstrado no Anexo 01.

4.4 Considerações Finais

Neste capítulo foi apresentada a solução proposta pelo trabalho com o objetivo de automatizar o processo de falsificação de notas de dinheiro, através da tecnologia RFID. Essa proposta de solução para o problema apresentado, especificamente neste capítulo, mostra de forma sistematizada e holística como implementar um sistema utilizando a identificação por rádio freqüência através do *middleware*.

5 – RESULTADOS E DISCUSSÃO

Esta seção tem o objetivo de apresentar e verificar as funcionalidades funcionais do projeto – Aplicação, Emulador e *Middleware* –, apresentando as entradas esperadas e a verificação dos resultados obtidos.

Os testes foram efetuados quando da conclusão do sistema. O primeiro teste efetuado foi de conexão com o banco de dados, pois, no momento de informar usuário e senha, já se observa a primeira conexão.

A segunda etapa do teste cuidou da verificação dos dados que estavam sendo gravados na tabela de usuários do sistema e tags que também estavam sendo gravadas na base de dados. Foi considerada, como etapa mais importante a verificação da gravação das informações e dos seus dados e por fim, os dados estavam gravados corretamente no banco de dados.

Com os dados gravados, a última etapa do teste foi verificar se passando a etiqueta (tag) próximo a leitora, se os dados seriam apresentados corretamente em tela.

Conclusão: Testes efetuados e informações apresentadas conforme foram gravadas no banco.

5.1 Apresentação da área de Aplicação do modelo

A aplicação foi desenvolvida para subsidiar as instituições financeiras, nas atividades de análise, verificação e estudos sobre falsificação de notas de dinheiro. O SUID poderá ser utilizado como um sistema de combate à falsificação de cédulas.

Importante ressaltar que ao se realizar uma leitura da etiqueta, faz-se necessário o envio de uma autenticação, para que o conteúdo do cartão possa ser lido somente por pessoas autorizadas.

5.2 Descrição da Aplicação do Modelo

A aplicação conta com um ambiente RFID simulado com a ferramenta RIFIDI, que será integrada ao Sistema Único de Identificação de Dinheiro - SUID com o *middleware*, conforme ilustra a figura 5.1:

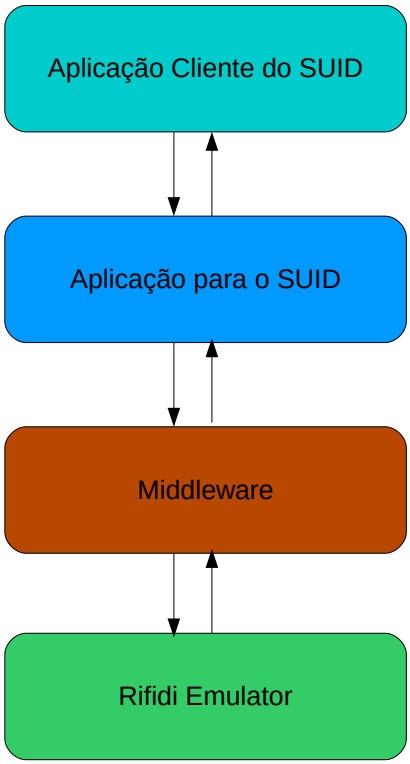

Figura 5.1 – Esquemático do SUID

5.3 Resultados da Aplicação do Modelo

Esta etapa do trabalho consiste em estudar a tecnologia de identificação por radiofreqüência através da utilização de simulação. Seu objetivo é analisar o comportamento de um sistema RFID em um cenário virtual. A utilização da simulação, ao invés da montagem de um experimento real com leitores e tags reais, traz vantagens em termos de custo e flexibilidade.

O objetivo da simulação é analisar o comportamento das etiquetas inteligentes, quando estas fazem o papel de cédulas de dinheiro. Essa análise é apresentada em um grid virtual. Em cada conjunto de simulações, um número fixo de tags é disposto de modo randômico neste grid. A técnica procura associar cada uma das tags a um único leitor, evitando o fenômeno de colisão, onde uma tag permanece no raio de ação de mais de um

leitor. O objetivo final é verificar em uma amostra de notas quantas tags são lidas e validadas no banco e mostrada para o usuário.

As simulações de cada cenário foram processadas através da análise das tags RFID em seu raio de cobertura, retornando a média das notas verdadeiras pelo leitor. Cada cenário apresenta uma característica, conforme ilustrado na figura 5.2:

Cenário	Componente	Condição
Cenário 1	1 cadastro Tags(Notas)	Vincular 1 tag a 1 nota
Cenário 2	10 Tags(Notas)	Todas Verdadeiras
Cenário 3	10 Tags(Notas)	7 Verdadeiras e 3 falsas com tags
Cenário 4	10 Tags(Notas)	8 Verdadeiras e 2 falsas sem tags
Cenário 5	10 Tags(Notas)	5 Verdadeiras e 3 Falsas com tags e 2 sem tags
Cenário 6	Usuário	Criação um Usuário

Figura 5.2 – Cenários da Simulação

5.4 – Descrição dos Cenários

5.4.1 Cenário 1 – 1 cadastro Tags (Notas)

No cenário 1 foi realizado o cadastro das tags e configurado leitor do simulador que é fixo, O tempo médio de cada simulação foi de 10 segundos e todas as simulações foram executadas no mesmo hardware e com a mesma configuração de software.

As evidências mostraram o sucesso dos testes deste cenário, conforme mostra a figura 5.3 - usuário cadastrando a nota. A figura 5.4 mostra a nota inserida na tabela correspondente no banco.

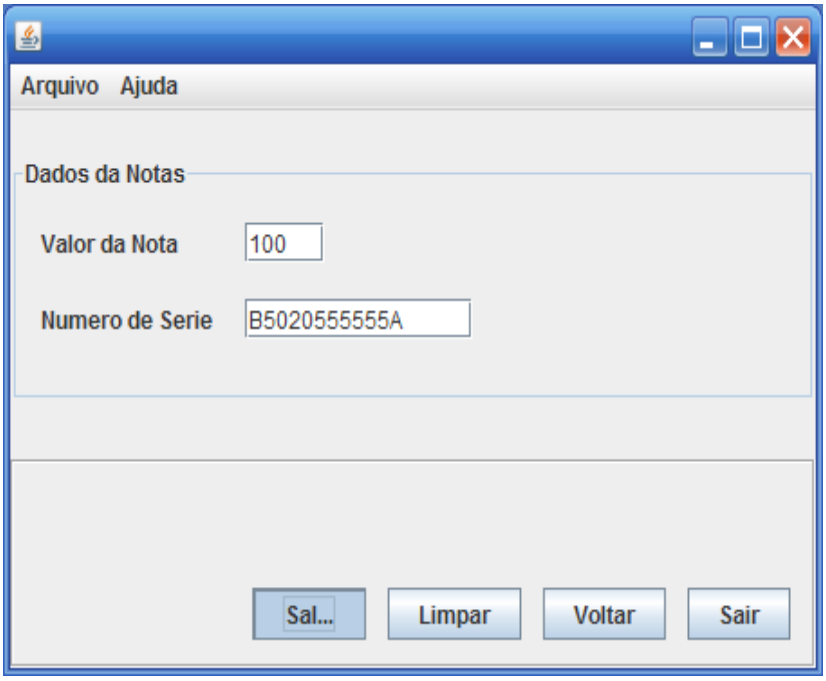

Figura 5.3 – Cadastro das Notas

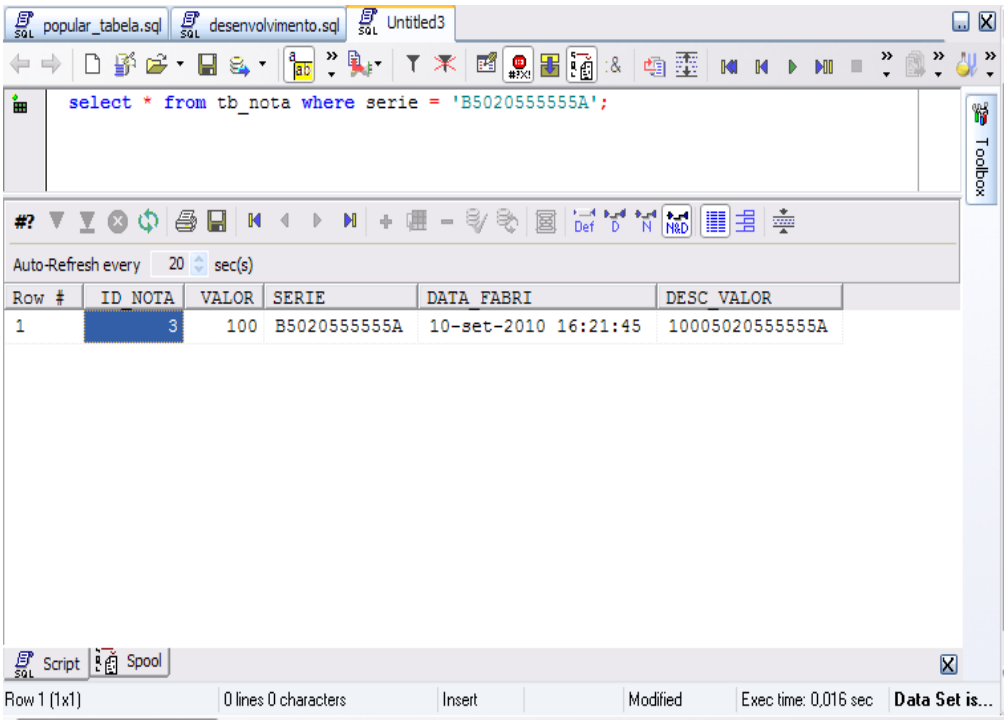

Figura 5.4 – Nota Cadastrado no Banco

5.4.2 Cenário 2 – 10 Tags (Notas)

No cenário 2 foi informado 10 tags, sendo o leitor um elemento fixo, inserido varios valores e distribuída randomicamente no grid, observando a condição de verdadeira para todas as etiquetas. O tempo médio de cada simulação foi de 10 segundos e todas as simulações foram executadas no mesmo hardware e com a mesma configuração de software.

As evidencias mostraram o sucesso dos testes do cenário 2 A figura 5.5 mostra o usuário pela aplicação solicitando a leitura da tags, a figura 5.6 é aproximando as tags à leitora pelo simulador, figura 5.7 mostra o resultado na tela.

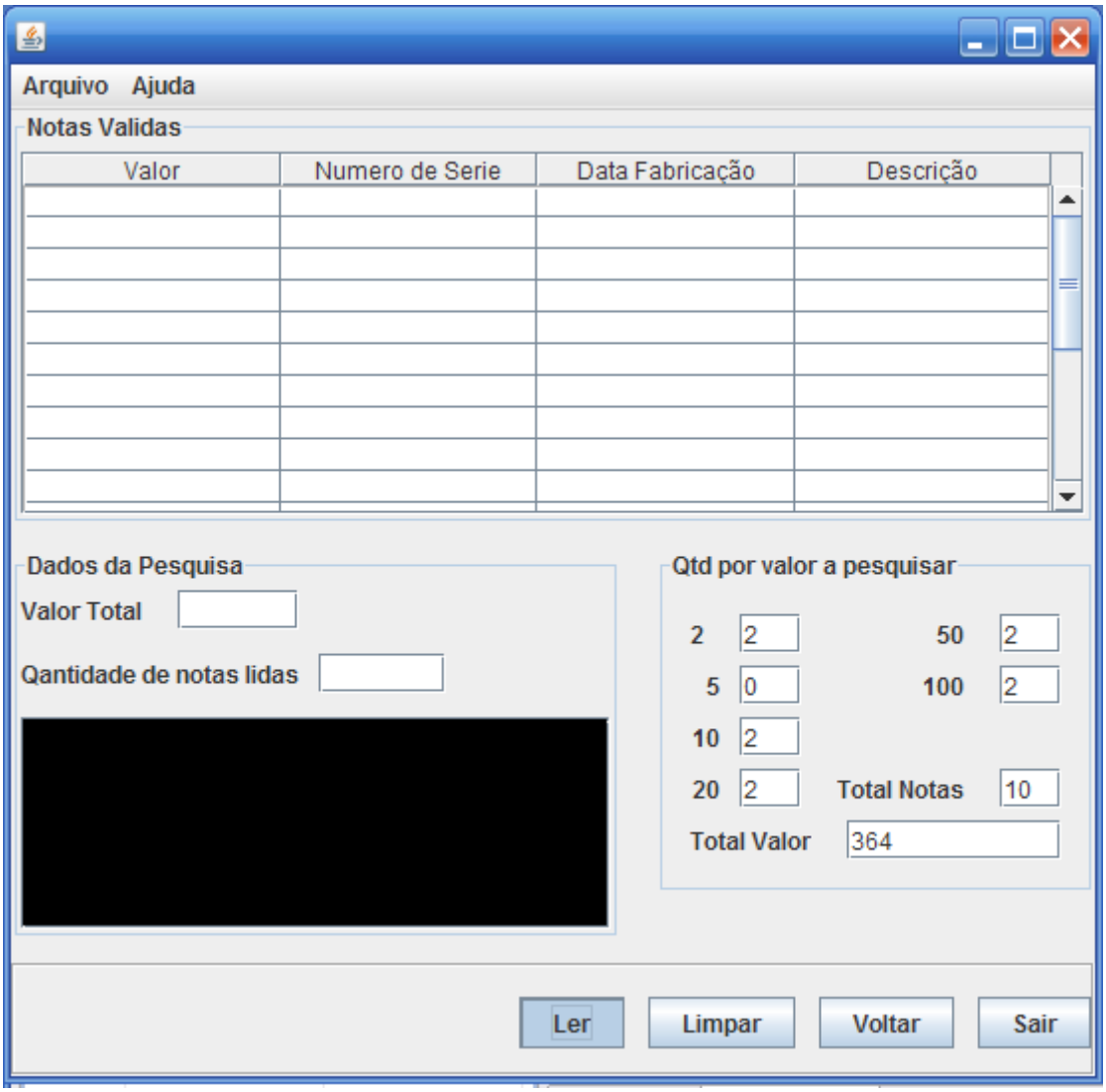

Figura 5.5 – Validar Notas

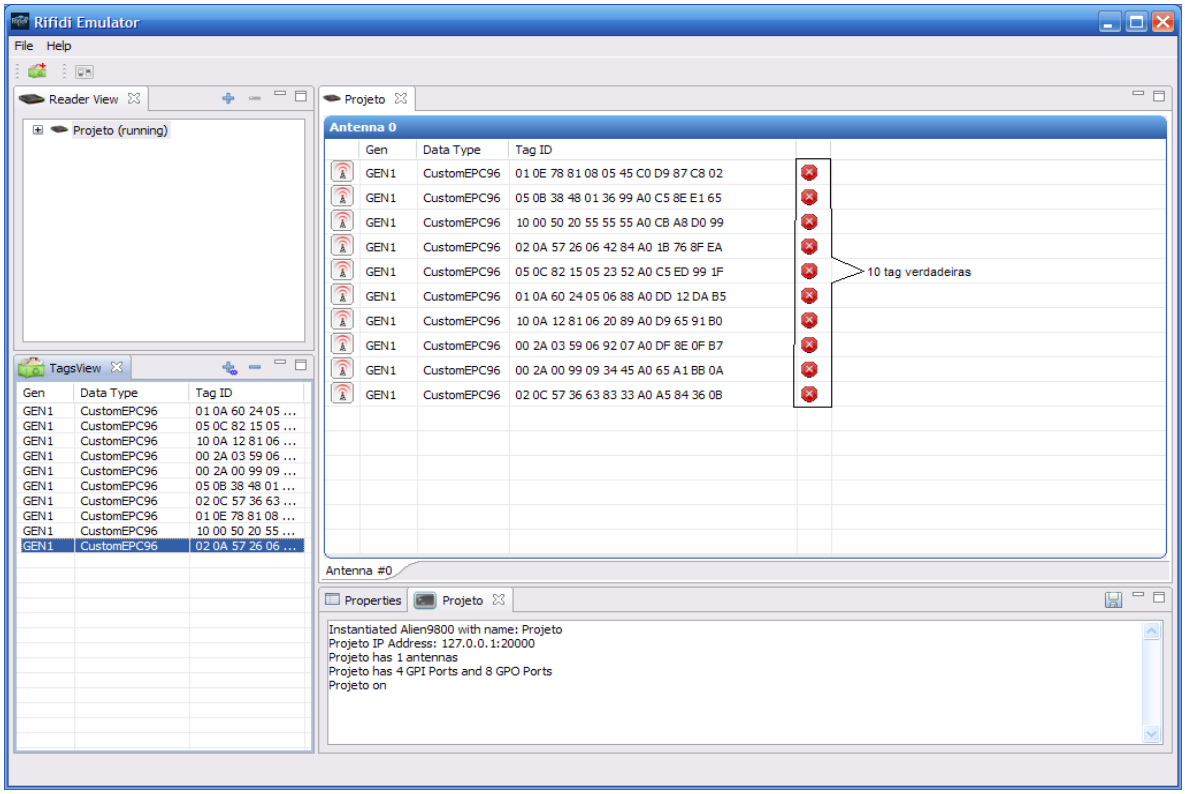

Figura 5.6 – Aproximação das Tags a Leitora

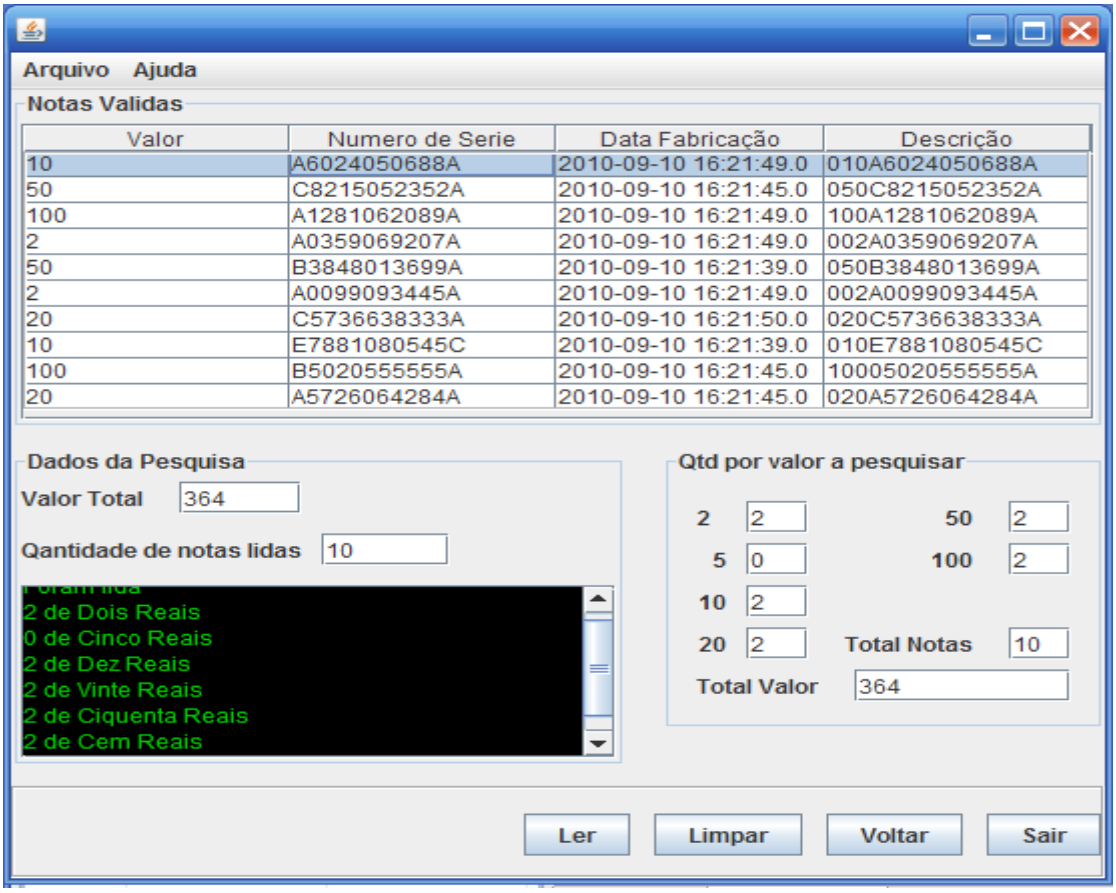

Figura 5.7 – Resultado das Notas Impresso na Telas

5.4.3 Cenário 3 – 10 Tags (Notas)

No cenário 3 foi mantido o número e valor das tags, sendo o leitor um elemento fixo. A condição estabelecida para garantir a veracidade das notas foi na ordem de : 7 verdadeiras e 3 falsas . O tempo médio de cada simulação foi de 10 segundos e todas as simulações foram executadas no mesmo hardware e com a mesma configuração de software.

As evidencias mostraram o sucesso dos teste do cenário 3, a figura 5.8 mostra o usuário pela aplicação solicitando a leitura da tags, a figura 5.9 é aproximando as tags à leitora pelo simulador, figura 5.10 mostra uma mensagem indicando que existe uma tag na nota mas a mesma não consta no banco e a figura 5.11 mostra o resultado com algumas divergências entre os valores a pesquisar com os já consultados.

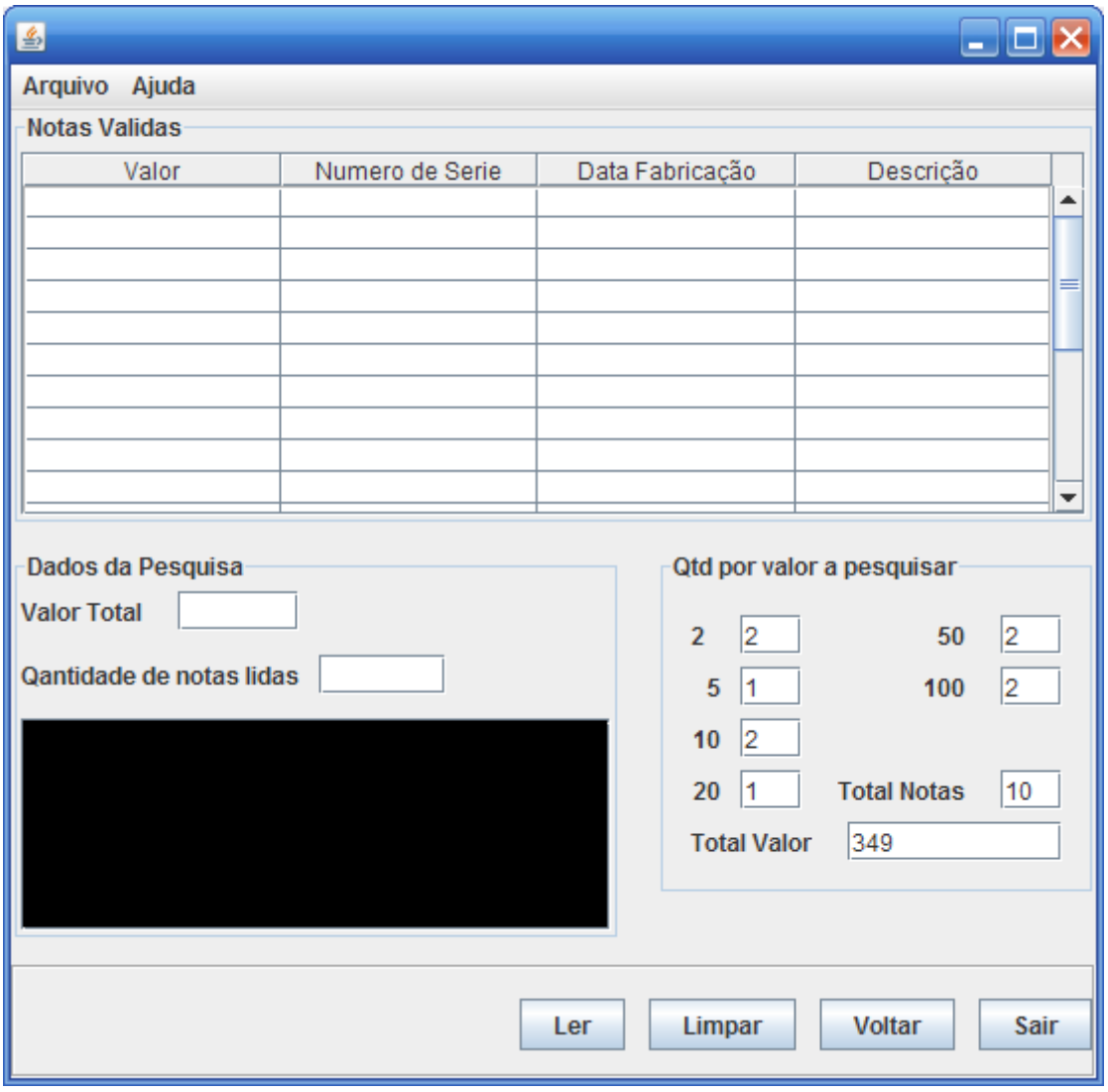

Figura 5.8 – Validar Notas

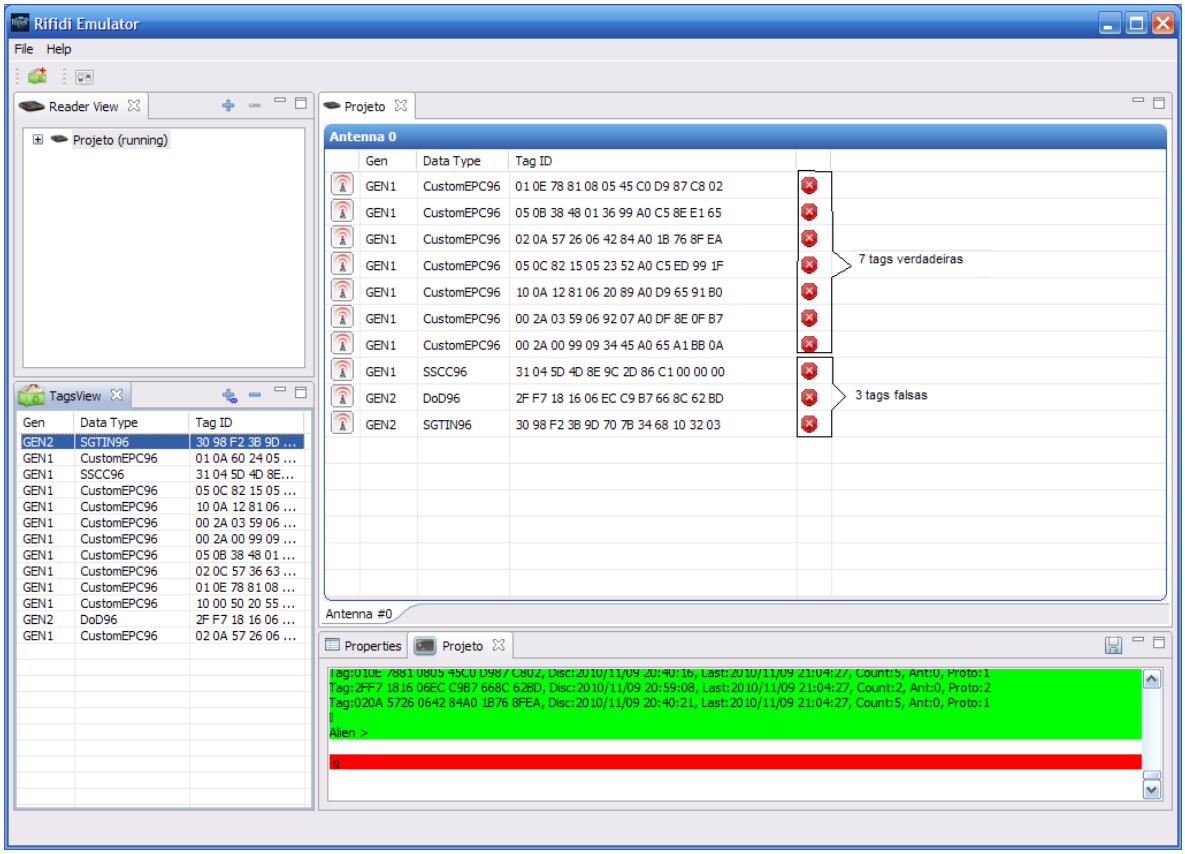

Figura 5.9 – Aproximação das Tags a Leitora

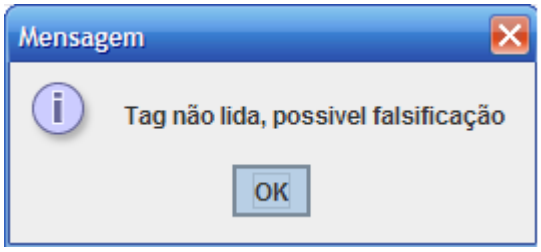

Figura 5.10 – Tag não lida

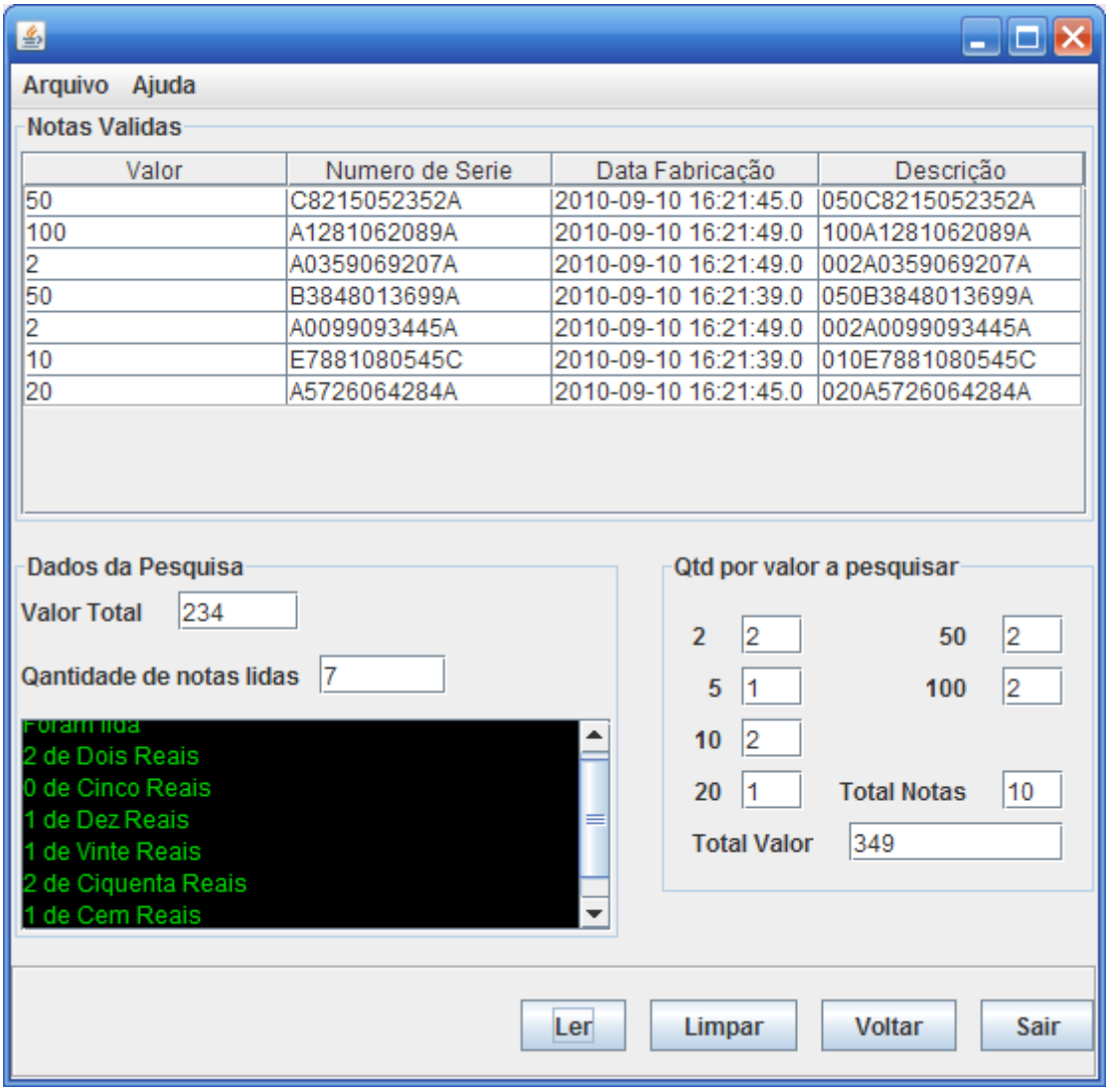

Figura 5.11 – Resultado das Notas Impresso na Telas

5.4.4 Cenário 4 – 10 Tags (Notas)

No cenário 4 foi mantido o número e valor das tags, sendo o leitor um elemento fixo e, estabelecida como condição, a premissa de 8 tags verdadeiras e 2 tags (sem leitura das tags) . O tempo médio de cada simulação foi de 10 segundos e todas as simulações foram executadas no mesmo hardware e com a mesma configuração de software.

As evidencias mostraram o sucesso dos testes do cenário 4, a figura 5.12 mostra o usuário pela aplicação, solicitando a leitura da tags, a figura 5.13 é aproximando as tags à leitora pelo simulador, sendo que as três notas falsas tem que abstrair a aproximação pois o emulador não permite esse teste, pois não tem como

colocar no simulador uma nota sem tag e a figura 5.14 mostra o resultado com algumas divergências entre os valores a pesquisar com àqueles já consultados, mostrando que possivelmente existem notas falsas, que são as notas sem tags.

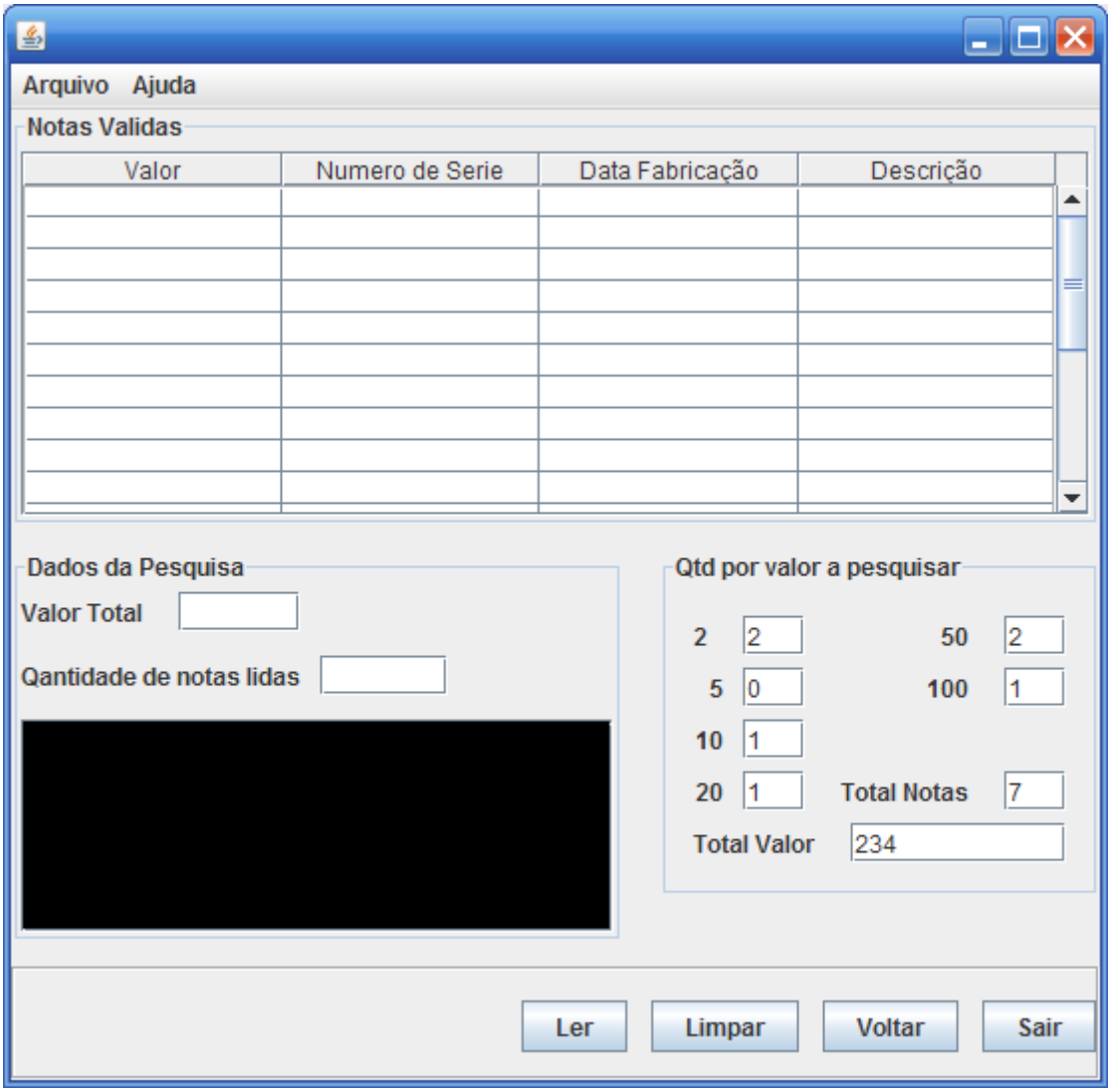

Figura 5.12 – Validar Notas

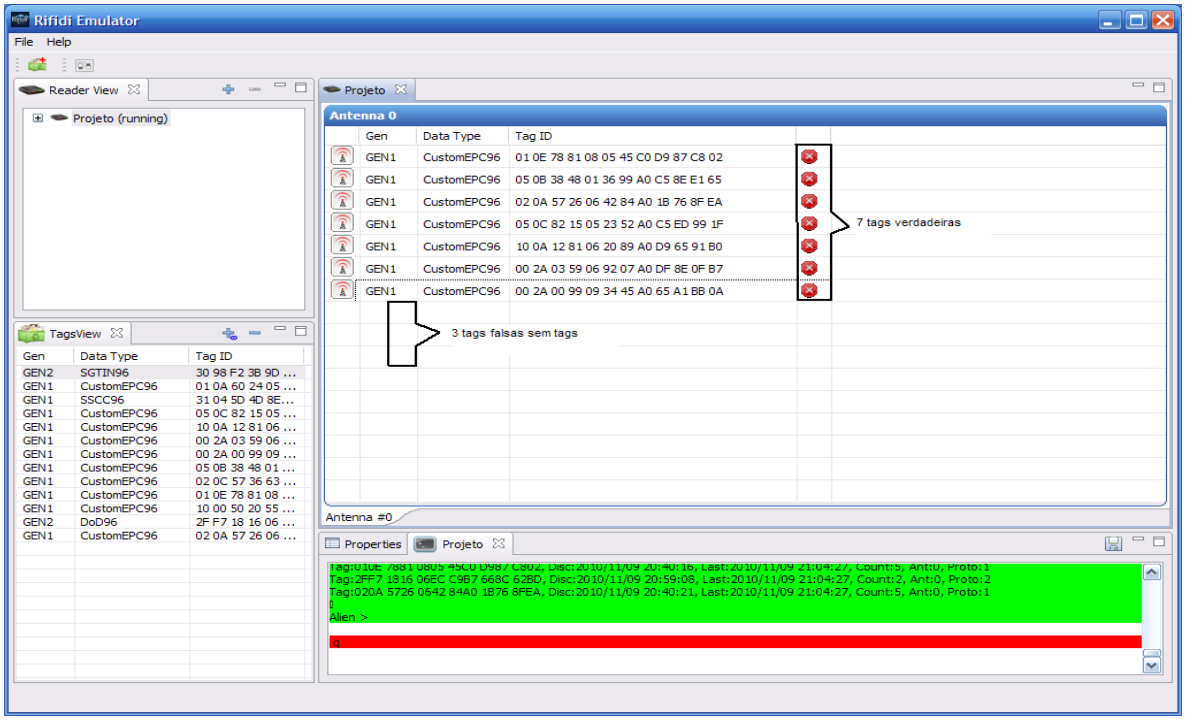

Figura 5.13 – Aproximação das Tags a Leitora

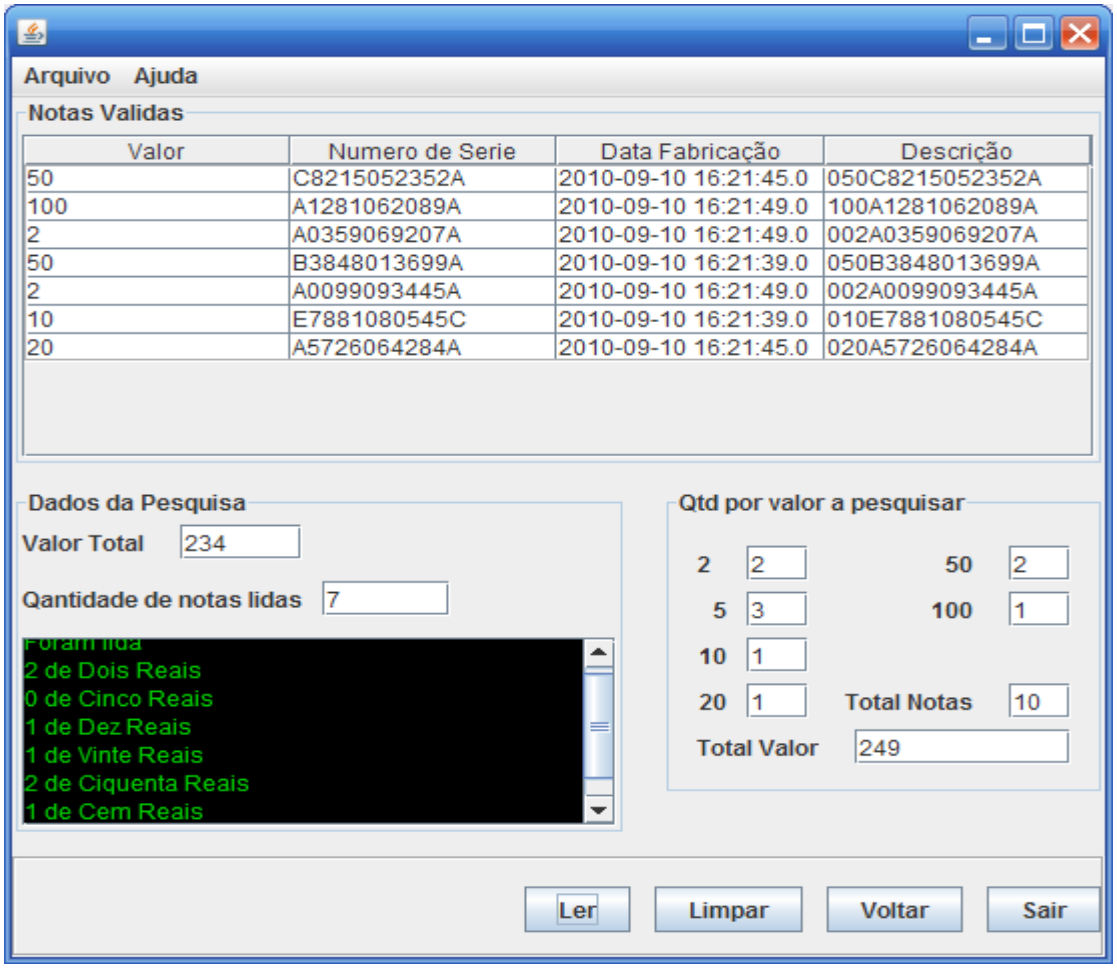

Figura 5.14 – Resultado das Notas Impresso na Telas

5.4.5 Cenário 5 – 10 Tags (Notas)

No cenário 5 o número, o valor e a configuração do *tranceiver* que lê as tags, foram mantidos. A análise considera 5 tags verdadeiras e 3 tags falsas, sendo que para 2 delas não ocorrerá leitura. O tempo médio de cada simulação foi de 10 segundos e todas as simulações foram executadas no mesmo hardware e com a mesma configuração de software.

As evidencias mostraram o sucesso dos teste do cenário 5, a figura 5.15 mostra o usuário pela aplicação, solicitando a leitura da tags, a figura 5.16 é aproximando as tags à leitora pelo simulador, mesclando o cenário 3 e 4 e a figura 5.17 mostra o resultado com algumas divergências entre os valores a pesquisar com os pesquisados, mostrando que possivelmente existem notas falsas.

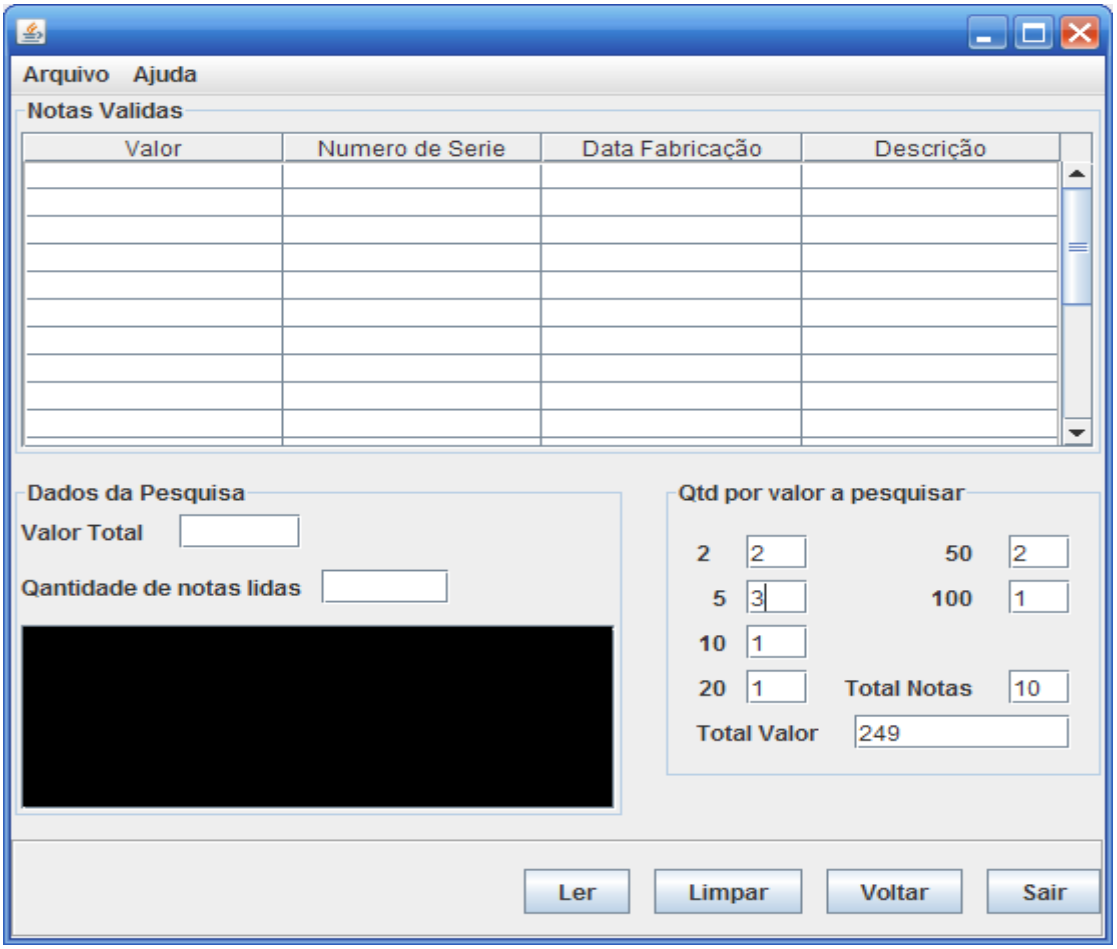

Figura 5.15 – Validar Notas

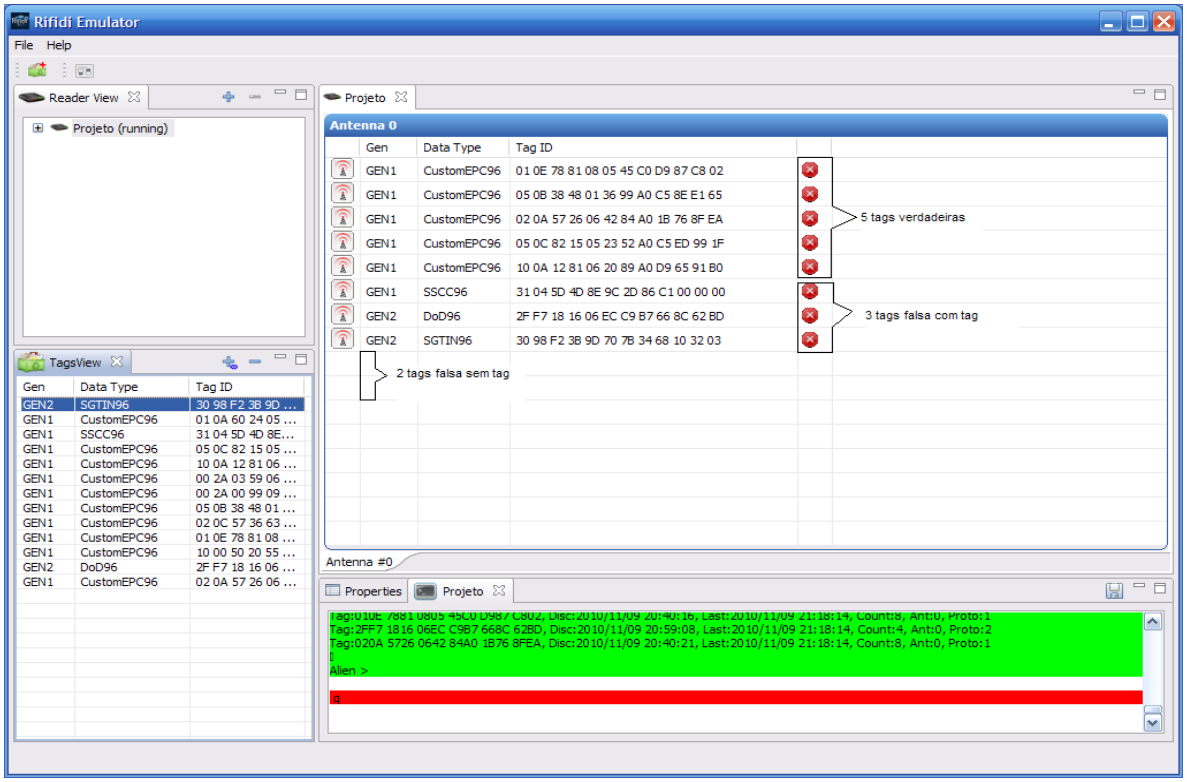

Figura 5.16 – Aproximação das Tags a Leitora

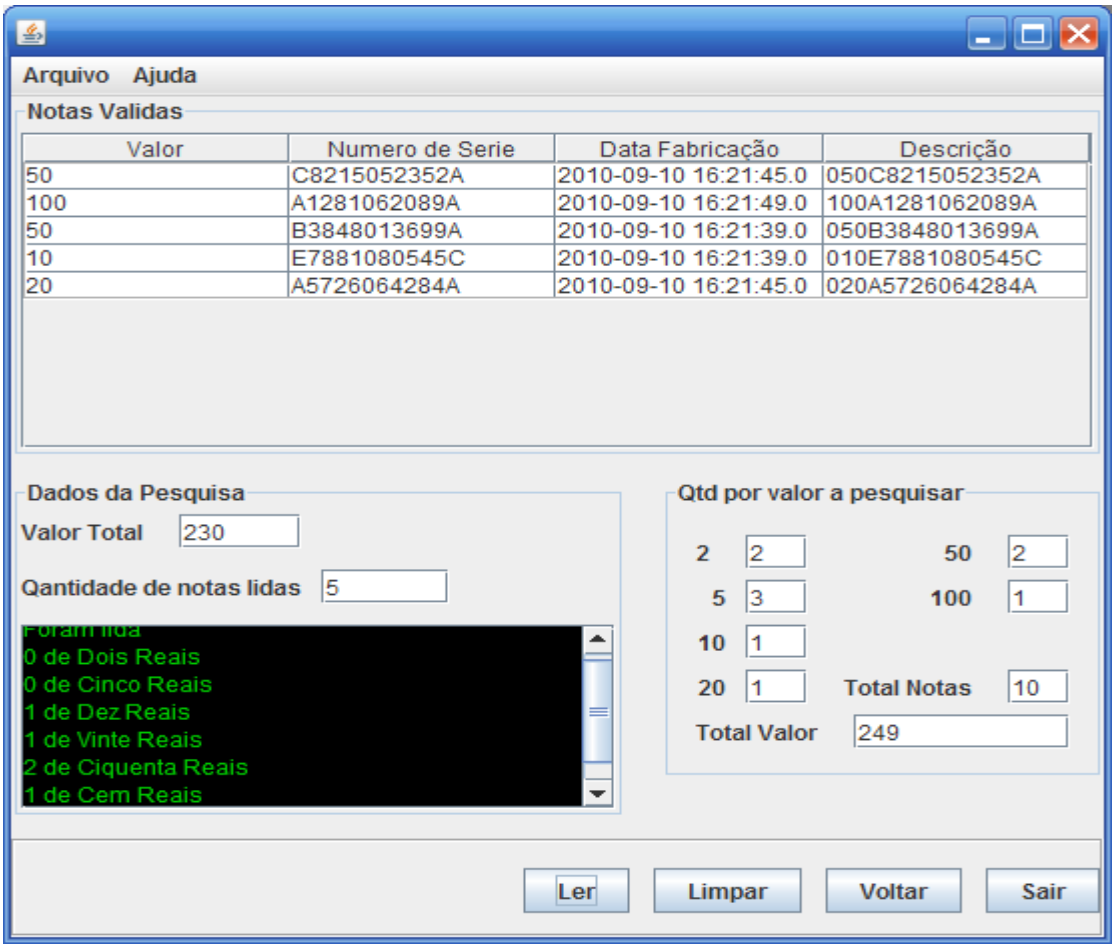

Figura 5.17 – Resultado das Notas Impresso na Telas

5.4.6 Cenário 6 – Usuário

No cenário 6 foi realizado o cadastro de usuários e configurado seu perfil de acesso à leitura das tags. Todas as simulações foram executadas no mesmo hardware e com a mesma configuração de software e o tempo médio de cada simulação foi de 10 segundos.

As evidencia mostraram o sucesso dos teste do cenário 1, a figura 5.18 mostra o cadastramento de um usuário e a figura 5.19 mostra o novo usuário inserido na tabela correspondente no banco.

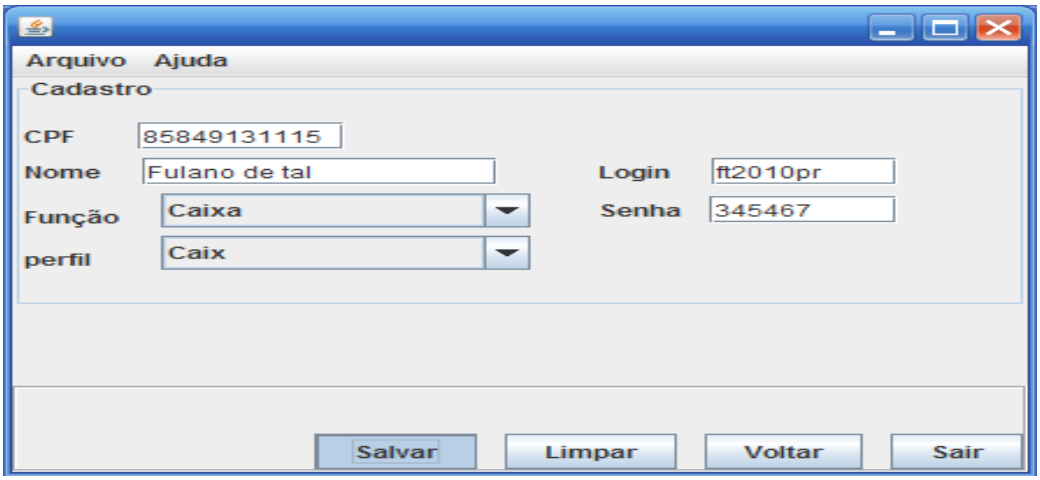

Figura 5.18 – Cadastro de Usuário

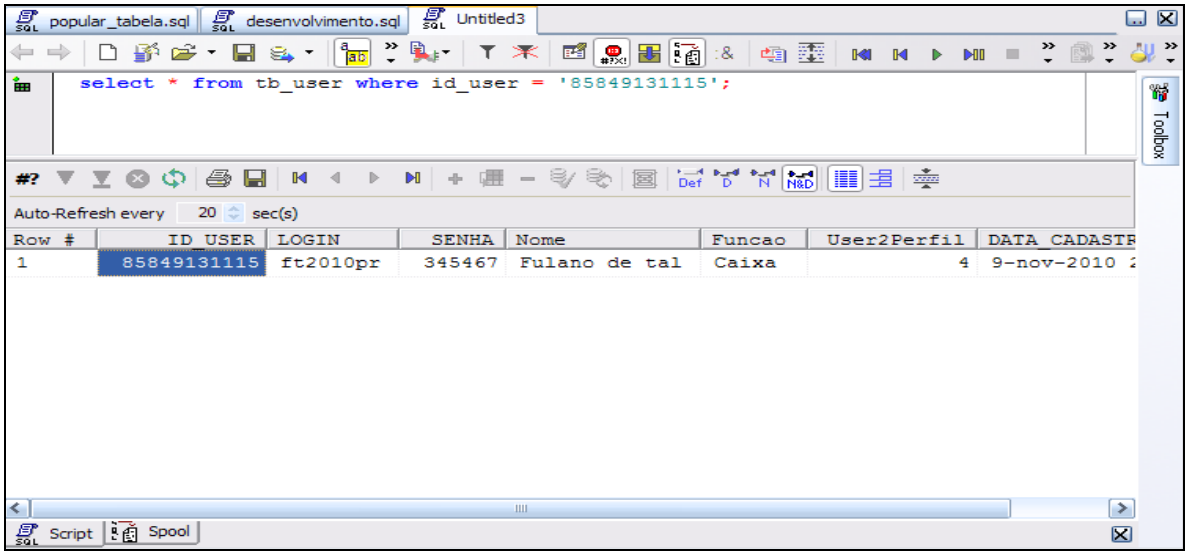

Figura 5.19 – Usuário Cadastrado no Banco

5.4 Discussão dos resultados

Depois de obtidos os resultados foi possível estabelecer algumas conclusões sobre os tópicos principais referente a proposta de simulação do trabalho. Eles demonstram que é viável a utilização de simulação para a análise de sistemas RFID. No entanto, em virtude da existência de poucas opções de simuladores e de trabalhos científicos na área, é aconselhável a utilização da técnica de simulação apenas para fins acadêmicos. Além disso, os modelos de tags e leitores presentes nos simuladores são bastante limitados em termos de funcionalidades e características. Na análise detalhada dos resultados obtidos, observou-se que as associações entre tags e leitores são eficazes. Isso prova a veracidade das notas de dinheiro, ora representadas pelas etiquetas lidas no emulador escolhido.

Outra curiosidade que merece destaque foi que o número de tags identificadas pelo leitor apresentou uma média de tags verdadeiras por tags lidas. A hipótese mais provável, para o número inferior de tags detectadas por esse leitor, pode ser tanto sua localização fora do grid, ou seja, fora do raio de cobertura, como também a ausência de atributos coincidentes com àqueles cadastrados no banco de dados, classificando-as como falsificadas.

5.5 Custos do modelo proposto

Devido à escassez de etiquetas e leitores apropriado, ao alto custo destas, em caso de importação, e da pouca flexibilidade para realizar experimentos com os equipamentos específicos, a alternativa mais viável foi utilizar ferramentas de simulação que possibilitem a criação de diversos cenários onde a tecnologia possa ser estudada.

A técnica de simulação é infinitamente útil para prever o desempenho de diversas alternativas e efetuar comparações entre uma maior variedade de ambientes. Dentre suas principais vantagens vale ressaltar que o custo total da realização deste projeto foi muito baixo, já que o emulador é um Software livre e os resultados ficaram muito próximos ou até mesmo iguais aos reais.

5.6 Avaliação Global do Modelo

Foi constatado que a implementação é eficiente e eficaz no que tange à identificação de notas falsas em relação as tecnologias hoje existentes. A tecnologia RIFD implantada na

nota apresenta, segundo os testes, uma menor probabilidade de incidência e disseminação de notas falsas no mercado circulante.

Através dos resultados obtidos foi possível identificar as notas falsificadas, sob várias perspectivas, pois existem vários atributos que podem ser alvo de falsificação (imagem latente, registro coincidente, relevo e textura do papel). A solução aqui proposta visa reduzir o crescente número de notas falsas em circulação e com previsão de crescimento e circulação de ordem exponencial.

6 - CONCLUSÃO

6.1 Conclusões

O objetivo deste trabalho foi abordar de uma forma geral a área de RFID, desde o entendimento da tecnologia até a utilização de identificação por radiofreqüência propriamente dita. Neste cenário também foi explorada a utilização de simulador para verificar a veracidade de cédulas de dinheiro, através da manipulação e operação das ferramentas de RFID, contidas neste emulador, em um ambiente flexível e de baixo custo.

A principal abordagem desta monografia trata o problema de falsificação de dinheiro identificado no mercado, incluindo o histórico, conceitos, princípios, relatos de falsificação, os diferentes projetos iniciados para tratar o problema e, por fim, sua evolução, chegando até o conceito de identificação por rádio freqüência e sua aplicação na área financeira. Além desta área apresentar um escasso material de pesquisa, outra grande dificuldade foi em selecionar qual material seria mais relevante de acordo com a proposta inicial do trabalho, partindo do tema "falsificação" até a identificação por radiofreqüência.

A conceituação da história do RFID, a especificação dos seus componentes e suas operações, além da questão da segurança, vantagens e desvantagens e exemplos de aplicações na área de RFID foram temas também explorados. O ponto mais relevante no contexto geral não foi apenas o fato de que a maior parte da bibliografia foi encontrada na língua inglesa, o que já era esperado, mas sim em encontrar uma utilização ainda muito discreta de RFID no Brasil, comparada a outros países.

A maior criticidade encontrada na perspectiva geral do projeto foi a simulação, pois o intuito de utilizá-la era justamente para evitar um custo alto na realização do trabalho, pois, apesar dos equipamentos de RFID estarem com valores cada vez menores, ainda representariam um custo considerável para a realização do projeto. O desenvolvimento de um script de simulação de RFID também foi cogitado, mas devido ao tempo escasso até a conclusão do trabalho e pela dificuldade no desenvolvimento de scripts utilizando ferramentas de simulação existentes, esta alternativa acabou sendo desconsiderada. A melhor opção então foi utilizar a ferramenta de simulação RFID existente e mais difundida no meio acadêmico. Porém raros foram os trabalhos encontrados e alguns, dentre os poucos considerados, não foram executados com sucesso. Problemas nos scripts de simulação, ausência de

documentação e de suporte por parte dos próprios desenvolvedores foram os problemas mais encontrados.

Após intensa pesquisa e estudo de diversas simulações com diferentes cenários, foi decidido a utilização daqueles mais importantes para apresentar a solução proposta, cujo principal objetivo foi mostrar a viabilidade em utilizar a simulação para analisar o comportamento de um sistema de identificação por radiofreqüência. As dúvidas que surgiram no decorrer da simulação poderão ser esclarecidas em novas pesquisas, mas não são consideradas comuns pelo fato de ser um trabalho novo e com pouco material de apoio.

A crescente utilização e o custo cada vez menor dos componentes levam a crer que os sistemas RFID se tornarão um importante diferencial competitivo, dentre algumas características que apontam para essa tendência.

Como conclusão final, do estudo efetuado sobre a falsificação fazendo uso de simulação, percebeu-se claramente que esta técnica poderá ser bastante útil no mercado financeiro e poderá ter resultados ainda mais eficientes, inclusive para distâncias mais longas.

6.2 - Sugestões para Trabalhos Futuros

Neste item são apresentadas algumas idéias e sugestões que podem ser utilizadas em trabalhos futuros, que seguindo a linha deste trabalho têm o seu foco voltado para a utilização da RFID no processo de identificação de falsificação de qualquer natureza.

Uma sugestão interessante seria criar um protótipo de um sistema que controle a emissão das cédulas de dinheiro, rastreando-as desde o início da produção até a saída da Casa da Moeda.

Outra sugestão é a utilização da tecnologia RFID no desenvolvimento de um framework, isto é, uma ferramenta de apoio ao desenvolvimento de aplicações em Java para a tecnologia RFID, onde estejam incorporadas funcionalidades de “Identificar e enviar –” onde os objetos serão identificados e essa informação enviada a um local predeterminado. Adicionado a isto, sugere-se a integração deste framework com outras ferramentas e bibliotecas que sejam relevantes para tal finalidade, de forma a permitir que estudantes, desenvolvedores, cientistas e utilizadores da tecnologia possam utilizá-lo e usufruir da melhor forma possível.

REFERÊNCIAS BIBLIGRÀFICAS

ABOSO, Gustavo Eduardo; ZAPATA, María Florencia. **Cibercriminalidae y derecho penal**. Buenos Aires: B de F, 2008.

AUTOCOM. **Soluções completas para RFID**. Disponível em: <http://www.autocom.com.br/produtos-RFID.asp>. Acesso em: 04 jun. 2008.

BITTENCOURT, Cezar Roberto. **Tratado de Direito Penal – Parte geral**. 10. ed. São Paulo: Saraiva, 2007. v.2.

BONAVIDES, Paulo. **Curso de direito constitucional**. 6. ed. Rio de Janeiro: Forense, 1996..

BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML.** Rio de Janeiro: Campus, 2002.

BIELETRO, Automação Industrial. **RFID.** 2005. Disponível em: <http://www.bieletro.com.br/rfid.htm>. Acesso em: 02 mai. 2009.

BONFIM, Marcia Bonassi Mougenot e BONFIM, Edilson Mougenot. **Lavagem de dinheiro**. São Paulo: Malheiros, 2005.

BRAUN, Daniela. RFID: **Etiquetas inteligentes conquista território do código de barras.** São Paulo, 2007. Disponível http://idgnow.uol.com.br/computacao_corporativa/2007 Acesso em: 04 abr 2010.

COMPUTERWORLD. **RFID está sujeito a fraudes, afirmam pesquisadores.** Sidney, 2006. Disponível em: < http://computerworld.uol.com.br/seguranca/2006/04/12/idgnoticia.2006-04-12.1739890681/IDGNoticia_view>. Acesso em: 22 out 2008.

CORDEIRO, Carlos. **Simulação de Sistemas Computacionais**. Disponível em: <http://www.cin.ufpe.br/~mac/ADS/aulas/adsaula3.ppt>. Acesso em 03 jan 2009.
ITS Newsletter. **RFID: Melhor, mais rápido, mais eficiente.**

_____. *Oracle* **10g Release 2 Grid Control Installation On Red Hat Enterprise Linux and CentOS**. Online. Available: http:// www.oracle- base.com / articles/10g.

FONTES, Attila. **Nova Geração, a Tecnologia dos BDOO's**.

GS1Brasil: **EPC - CÓDIGO ELETRÔNICO DE PRODUTO**. Disponível em: <http://www.gs1brasil.org.br/main.jsp?lumChannelId=480F89A81A90B196011A917351EE10F2>. Acesso em: 03 nov. 2010.

INOVAÇÃO TECNOLÓGICA. **Liberada utilização de chip de identificação pessoal nos EUA.** Campinas, 2004. Disponível em: <http://www.inovacaotecnologica.com.br/ quem.html>. Acesso em: 20 maio 2010.

LIMA, Adilson da Silva. **UML 2.0 do requisito à solução.** 1 edição. São Paulo: Érica, 2005.

LOES, João. **O RFID vai etiquetar o mundo.** [S.l] Disponível em: <http://wnews.uol. com.br/site/noticias/materia_especial.php?id_secao=17&id_conteudo=255>. Acesso em 16 ago 2010.

LONEY, Kevin, BRYLA, Bob. **Oracle 10g.o manual do dba**.1ª Ed. Trad. Docware Traduções Técnicas. Rio de Janeiro; Campus, 2005.

Monografia - Curso de Tecnólogo em Informática com Ênfase em Gestão de Negócios. Faculdade de Tecnologia da Baixada Santista – Extensão Praia Grande.

MORALES, Diego Francisco de Gastal. **Arquitetura dos Real Application Clusters da Oracle, e Análise do seu Mecanismo de Cache Distribuído***: o Cache Fusio*n.

OLIVEIRA, A. S., PEREIRA, M. F. **Estudo da tecnologia de identificação por radiofreqüência - RFID**. Disponível em http://www.ene.unb.br/antenas/Arquivos/Alessandromilene.pdf>. Acesso em: 13 dez. 2008.

PINHEIRO, José Mauricio Santos. **RFID Identificação por rádio freqüência.** Volta Redonda, 2004. Disponível em: <http://www.projetoderedes.com.br/artigos/artigo_identificacao_por_radiofrequencia.php>. Acesso em: 20 jun 2010.

RAMOS, Ricardo. **Banco de Dados Orientado Objeto**.

SANTANA, Sandra Regina Matias. **RFID – Identificação por radiofreqüência.** 2005.

SOUZA Bruno. **JAVA & JINI**. 1999. SENAC, São José do Rio Preto, SP. Disponível em <http://www.javaman.com.br/apres/java&jini/index.html>. Acesso em: 08 Out. 2010.

WIKIPEDIA. **RFID.** [S.l] 2010. Disponível em: < http://pt.wikipedia.org/wiki/Rfid >. Acesso em: 05 mai 10.

YUAN, Michael. Tecnologia RFID e a Internet de Objetos. **Revista Mundo Java**. Curitiba, n.11, 2005

ANEXOS

Código-Fonte

```
public class Mediador {
    public static final String IP_ADDRESS = new String("127.0.0.1");
   public static final int READER_PORT = 20000;
   private static Socket connection = null;
   private static PrintWriter out = null;
   private static BufferedReader in = null;

   // Initializar o conector e username/password
   private void  init() throws IOException, InterruptedException{
      connection = new Socket("127.0.0.1", READER_PORT);
      in = new BufferedReader(new InputStreamReader(connection.getInputStream()));
      out = new PrintWriter(connection.getOutputStream());
      Thread.sleep(500);
      System.out.println(readFromReader(in));
      out.write("alien\n");
      out.flush();
      System.out.println(readFromReader(in));
      Thread.sleep(500);
      out.write("password\n");
      out.flush();
      System.out.println(readFromReader(in));
   }

   private void tearDown() throws IOException{
      out.write("q");
         out.flush();
         connection.close();
   }
   // Pegar tega pela leitora
```

```
    private String getTags() throws IOException{
        String command = "t";
            out.write(command + "\n");
            out.flush();
            String returnVal = readFromReader(in);
            return returnVal;
    }

    public String execmediador () throws IOException, InterruptedException {
        String dleitura = null;
        Mediador client = new Mediador();
            client.init();
            Thread.sleep(500);
            String tags = client.getTags();
            System.out.println(tags);
        dleitura = tags;
            Thread.sleep(500);
            client.tearDown();
        return dleitura;
    }

    public static String readFromReader(BufferedReader inBuf) throws IOException{
        StringBuffer buf=new StringBuffer();
            int ch=inBuf.read();
            while((char)ch!='\0'){
            buf.append((char)ch);
            ch=inBuf.read();
            }
            return buf.toString();
    }
}
```

```
package projetofinal;

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

import conexao.conexao;
/**
 *
 * @author sergio
 */
public class Main {



    /**
     * @ram args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here

        conexao con_projeto;
        con_projeto = new conexao();
        con_projeto.conecta();
        new Login().show();
    }

}



/*
 * To change this template, choose Tools | Templates
```

```
 * and open the template in the editor.
 */

/*
 * Logim.java
 *
 * Created on 25/08/2010, 14:33:21
 */

package projetofinal;

import conexao.conexao;
import java.sql.*;
import javax.swing.*;

/**
 *
 * @author ianuck
 */

public class Login extends javax.swing.JFrame {
    conexao con_login;
    /** Creates new form Logim */
    public Login() {
        initComponents();

        con_login = new conexao();
        con_login.conecta();

    }

    /** This method is called from within the constructor to
```

```
    * initialize the form.
    * WARNING: Do NOT modify this code. The content of this method is
    * always regenerated by the Form Editor.
    */
   @SuppressWarnings("unchecked")
   // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
   private void initComponents() {

      jPanel1 = new javax.swing.JPanel();
      bnt_entra = new javax.swing.JButton();
      bnt_sair = new javax.swing.JButton();
      jLabel1 = new javax.swing.JLabel();
      jLabel2 = new javax.swing.JLabel();
      txt_login = new javax.swing.JTextField();
      txt_senha = new javax.swing.JTextField();

      setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

      jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder("Login"));

      bnt_entra.setText("Entrar");
      bnt_entra.addActionListener(new java.awt.event.ActionListener() {
         public void actionPerformed(java.awt.event.ActionEvent evt) {
            bnt_entraActionPerformed(evt);
         }
      });

      bnt_sair.setText("Sair");
      bnt_sair.addActionListener(new java.awt.event.ActionListener() {
         public void actionPerformed(java.awt.event.ActionEvent evt) {
            bnt_sairActionPerformed(evt);
```

```
        }
    });

    jLabel1.setText("Login");

    jLabel2.setText("Senha");

    txt_login.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txt_loginActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
            .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addComponent(bnt_entra)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(bnt_sair))
                .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(jLabel1)
                        .addComponent(jLabel2))
                    .addGap(18, 18, 18)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(txt_login, javax.swing.GroupLayout.DEFAULT_SIZE, 104, Short.MAX_VALUE)
                    .addComponent(txt_senha, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT_SIZE, 104, Short.MAX_VALUE))))
            .addContainerGap())
      );
      jPanel1Layout.setVerticalGroup(
         jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
         .addGroup(jPanel1Layout.createSequentialGroup()
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
               .addComponent(jLabel1)
               .addComponent(txt_login, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
               .addComponent(jLabel2)
               .addComponent(txt_senha, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
               .addComponent(bnt_sair)
               .addComponent(bnt_entra)))
      );

      javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
      getContentPane().setLayout(layout);
```

```
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addContainerGap())
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addContainerGap(23, Short.MAX_VALUE)
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(21, 21, 21))
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents

    private void txt_loginActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_txt_loginActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_txt_loginActionPerformed

    private void bnt_sairActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_bnt_sairActionPerformed
        // TODO add your handling code here:
        System.exit(0);
    }//GEN-LAST:event_bnt_sairActionPerformed
```

```
  private void bnt_entraActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_bnt_entraActionPerformed
      // TODO add your handling code here:
      try
      {
         String sql = "select * from tb_user where  login like'"+
               txt_login.getText()+"' and senha like'" +txt_senha.getText()+"'";
         con_login.executeSQL(sql);
         if (con_login.resultset.next())
         {
            JOptionPane.showMessageDialog(null, "Logim aceito");
            new TelaPrincipal().show();
            this.dispose();
         }
         else
         {
           JOptionPane.showMessageDialog(null, "Logim não aceito");
         }
      }
      catch(SQLException erro)
      {
         JOptionPane.showMessageDialog(null,"Erro = " +erro);
      }
  }//GEN-LAST:event_bnt_entraActionPerformed

  /**
  * @param args the command line arguments
  */
  public static void main(String args[]) {
     java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
```

```
            new Login().setVisible(true);
        }
    });
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton bnt_entra;
private javax.swing.JButton bnt_sair;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JPanel jPanel1;
private javax.swing.JTextField txt_login;
private javax.swing.JTextField txt_senha;
// End of variables declaration//GEN-END:variables

}



/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/*
 * CriarUser.java
 *
 * Created on 24/08/2010, 10:35:07
 */

package criaruser;

import conexao.conexao;
```

```
import java.awt.Component;
import java.sql.*;
import javax.swing.*;
import projetofinal.TelaPrincipal;
import utilitarios.Perfil;
import utilitarios.Utilitarios;

/**
 *
 * @author ianuck
 */

public class CriarUser extends javax.swing.JFrame {

    conexao con_user;

    /** Creates new form CriarUser */
    public CriarUser() {
        initComponents();



        con_user = new conexao();
        con_user.conecta();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
```

```
BEGIN:initComponents
   private void initComponents() {

      jFrame1 = new javax.swing.JFrame();
      jPanel1 = new javax.swing.JPanel();
      jLabel1 = new javax.swing.JLabel();
      jLabel2 = new javax.swing.JLabel();
      jLabel3 = new javax.swing.JLabel();
      jLabel4 = new javax.swing.JLabel();
      txt_cpf = new javax.swing.JTextField();
      txt_nome = new javax.swing.JTextField();
      cb_funcao = new javax.swing.JComboBox();
      cb_perfil = new javax.swing.JComboBox();
      jLabel5 = new javax.swing.JLabel();
      jLabel6 = new javax.swing.JLabel();
      txt_login = new javax.swing.JTextField();
      txt_senha = new javax.swing.JTextField();
      jPanel2 = new javax.swing.JPanel();
      bnt_sair = new javax.swing.JButton();
      bnt_limpar = new javax.swing.JButton();
      bnt_voltar = new javax.swing.JButton();
      bnt_salvar = new javax.swing.JButton();
      jMenuBar1 = new javax.swing.JMenuBar();
      m_arquivo = new javax.swing.JMenu();
      im_salvar = new javax.swing.JMenuItem();
      im_sair = new javax.swing.JMenuItem();
      m_ajuda = new javax.swing.JMenu();
      jMenuItem1 = new javax.swing.JMenuItem();

      javax.swing.GroupLayout jFrame1Layout = new
javax.swing.GroupLayout(jFrame1.getContentPane());
      jFrame1.getContentPane().setLayout(jFrame1Layout);
```

```
        jFrame1Layout.setHorizontalGroup(
            jFrame1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 400, Short.MAX_VALUE)
        );
        jFrame1Layout.setVerticalGroup(
            jFrame1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 300, Short.MAX_VALUE)
        );

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder("Cadastro"));

        jLabel1.setText("CPF");

        jLabel2.setText("Nome");

        jLabel3.setText("Função");

        jLabel4.setText("perfil");

        txt_cpf.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                txt_cpfActionPerformed(evt);
            }
        });

        cb_funcao.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Analista I",
"Analista II", "Administrador Sistema", "Caixa", "Gerente Ag.", " " }));

        cb_perfil.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Anal",
"GenA", "SA", "Caix" }));
```

```
        jLabel5.setText("Login");

        jLabel6.setText("Senha");

        txt_login.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                txt_loginActionPerformed(evt);
            }
        });

        txt_senha.setCursor(new java.awt.Cursor(java.awt.Cursor.TEXT_CURSOR));

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addComponent(jLabel1)
                        .addGap(27, 27, 27)
                        .addComponent(txt_cpf,      javax.swing.GroupLayout.PREFERRED_SIZE,      89,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addComponent(jLabel2)
                        .addGap(18, 18, 18)
                        .addComponent(txt_nome,    javax.swing.GroupLayout.PREFERRED_SIZE,    153,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(jPanel1Layout.createSequentialGroup()
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabel3)
                    .addComponent(jLabel4))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(cb_perfil, 0, 153, Short.MAX_VALUE)
                    .addComponent(cb_funcao,  0,  javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))))
            .addGap(29, 29, 29)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addComponent(jLabel6)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(txt_senha,  javax.swing.GroupLayout.DEFAULT_SIZE,  72,
Short.MAX_VALUE))
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addComponent(jLabel5)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addComponent(txt_login,  javax.swing.GroupLayout.DEFAULT_SIZE,  71,
Short.MAX_VALUE)))
            .addContainerGap(51, javax.swing.GroupLayout.PREFERRED_SIZE))
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addGroup(jPanel1Layout.createSequentialGroup()
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel1)
                .addComponent(txt_cpf,          javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel2)
                .addComponent(txt_nome,         javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel5)
            .addComponent(txt_login,            javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(jLabel3)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(cb_funcao,        javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel6)
            .addComponent(txt_senha,        javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(jLabel4)
            .addComponent(cb_perfil,        javax.swing.GroupLayout.PREFERRED_SIZE,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap(22, Short.MAX_VALUE))
        );

        jPanel1Layout.linkSize(javax.swing.SwingConstants.VERTICAL, new
java.awt.Component[] {jLabel3, jLabel6});

        jPanel2.setBorder(javax.swing.BorderFactory.createEtchedBorder());

        bnt_sair.setText("Sair");
        bnt_sair.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                bnt_sairActionPerformed(evt);
            }
        });

        bnt_limpar.setText("Limpar");
        bnt_limpar.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                bnt_limparActionPerformed(evt);
            }
        });

        bnt_voltar.setText("Voltar");
        bnt_voltar.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                bnt_voltarActionPerformed(evt);
            }
        });

        bnt_salvar.setText("Salvar");
        bnt_salvar.addActionListener(new java.awt.event.ActionListener() {
```

```
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            bnt_salvarActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
    jPanel2.setLayout(jPanel2Layout);
    jPanel2Layout.setHorizontalGroup(
        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
            .addContainerGap(128, Short.MAX_VALUE)
            .addComponent(bnt_salvar)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(bnt_limpar)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(bnt_voltar)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(bnt_sair))
    );
    jPanel2Layout.setVerticalGroup(
        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
            .addContainerGap(33, Short.MAX_VALUE)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(bnt_sair)
                .addComponent(bnt_voltar)
                .addComponent(bnt_limpar)
                .addComponent(bnt_salvar)))
    );
```

```
        m_arquivo.setText("Arquivo");


im_salvar.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_S,
java.awt.event.InputEvent.CTRL_MASK));
        im_salvar.setText("Salvar");
        m_arquivo.add(im_salvar);


im_sair.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_F4,
java.awt.event.InputEvent.ALT_MASK));
        im_sair.setText("Sair");
        im_sair.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                im_sairActionPerformed(evt);
            }
        });
        m_arquivo.add(im_sair);

        jMenuBar1.add(m_arquivo);

        m_ajuda.setText("Ajuda");

        jMenuItem1.setText("Sobre");
        m_ajuda.add(jMenuItem1);

        jMenuBar1.add(m_ajuda);

        setJMenuBar(jMenuBar1);

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
```

```
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel1, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 58,
Short.MAX_VALUE)
                .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents

    private void txt_loginActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_txt_loginActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_txt_loginActionPerformed

    private void txt_cpfActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_txt_cpfActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_txt_cpfActionPerformed
```

```
    private void bnt_limparActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_bnt_limparActionPerformed
        // TODO add your handling code here:
        Utilitarios util = new Utilitarios();
        JPanel obj = jPanel1;
        util.limpaComponentes(obj);
    }//GEN-LAST:event_bnt_limparActionPerformed

    private void im_sairActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_im_sairActionPerformed
        // TODO add your handling code here:
        System.exit(0);
    }//GEN-LAST:event_im_sairActionPerformed

    private void bnt_sairActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_bnt_sairActionPerformed
        // TODO add your handling code here:
        System.exit(0);
    }//GEN-LAST:event_bnt_sairActionPerformed

    private void bnt_voltarActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_bnt_voltarActionPerformed
        // TODO add your handling code here:
        new TelaPrincipal().show();
        this.dispose();
    }//GEN-LAST:event_bnt_voltarActionPerformed

    private void bnt_salvarActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_bnt_salvarActionPerformed
        // TODO add your handling code here:
        Perfil perfil = new Perfil();
```

```
    String passaperfil = (String) this.cb_perfil.getSelectedItem();
    String funcao = (String) this.cb_funcao.getSelectedItem();
    String resperfil = perfil.getperfil(passaperfil);



     String  sqlinsert = "Insert into tb_user Values ('"+txt_cpf.getText()+
       "','"+txt_login.getText()+"','"+txt_senha.getText()+
       "','"+txt_nome.getText()+"','"+funcao+"','"+resperfil+"',sysdate)";



       con_user.executeInsert(sqlinsert);

}//GEN-LAST:event_bnt_salvarActionPerformed

/**
* @param args the command line arguments
*/
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new CriarUser().setVisible(true);
        }
    });
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton bnt_limpar;
private javax.swing.JButton bnt_sair;
private javax.swing.JButton bnt_salvar;
private javax.swing.JButton bnt_voltar;
private javax.swing.JComboBox cb_funcao;
```

```
    private javax.swing.JComboBox cb_perfil;
    private javax.swing.JMenuItem im_sair;
    private javax.swing.JMenuItem im_salvar;
    private javax.swing.JFrame jFrame1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JMenuBar jMenuBar1;
    private javax.swing.JMenuItem jMenuItem1;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JMenu m_ajuda;
    private javax.swing.JMenu m_arquivo;
    private javax.swing.JTextField txt_cpf;
    private javax.swing.JTextField txt_login;
    private javax.swing.JTextField txt_nome;
    private javax.swing.JTextField txt_senha;
    // End of variables declaration//GEN-END:variables
package conexao;

import java.sql.*;
import javax.swing.*;

public class conexao
{
    final private String driver = "oracle.jdbc.driver.OracleDriver";
    final private String url = "jdbc:oracle:thin:@127.0.0.1:1521:XE";
    final private String usuario = "system";
    final private String senha = "system";
```

```
private Connection conexao;
public Statement statement;
public ResultSet resultset;

public boolean conecta()
{
   boolean result = true;
   try
   {
      Class.forName(driver);
      conexao = DriverManager.getConnection(url,usuario,senha);
      JOptionPane.showMessageDialog(null, "Sistema conectado ao banco Oracle EX");
   }
   catch(ClassNotFoundException Driver)
   {
      JOptionPane.showMessageDialog(null,"Driver não localizado" +Driver);
      result = false;
   }
   catch(SQLException Fonte)
   {
      JOptionPane.showMessageDialog(null, "Deu erro na conexao" +
                        "com a fonte de dados:" + Fonte);
      result = false;
   }

   return result;

}

public void executeSQL(String sql)
{
   try
```

```
        {
            statement = conexao.createStatement();
            resultset = statement.executeQuery(sql);
        }
        catch(SQLException sqlex)
        {
            JOptionPane.showMessageDialog(null,"Não foi possivel "+
                    "executar o comando sql," + sqlex+ "o sql passado");
        }
    }

    public void executeInsert(String sqlinsert)
    {
        try
        {
            statement = conexao.createStatement();
            statement.executeUpdate(sqlinsert);;
             JOptionPane.showMessageDialog(null,"Insert executado corretamente" );
        }
        catch(SQLException sqlex)
        {
            JOptionPane.showMessageDialog(null,"Não foi possivel "+
                    "executar o comando insert," + sqlex);
        }
    }
}



/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
```

```
 */

package utilitarios;

import conexao.conexao;
import java.sql.*;
import javax.swing.*;
import java.lang.String;
/**
 *
 * @author ianuck
 */
public class Perfil {



   conexao con_perfil;

   public String getperfil (String passaperfil)
   {
      con_perfil = new conexao();
      con_perfil.conecta();
     String valperfil = null;


       try
      {
         String sql = "select id_perfil from tb_perfil where perfil = '"+passaperfil+"'";
         con_perfil.executeSQL(sql);
         if (con_perfil.resultset.next()){
         valperfil = con_perfil.resultset.getString("id_perfil");
         }
```

```
        }
        catch(SQLException erro)
        {
            JOptionPane.showMessageDialog(null,"Erro = " +erro);
        }
        return valperfil;
    }
}



package projetofinal;

import criarnota.CriarNotas;
import criaruser.CriarUser;
import validarnota.ValidarNota;

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/*
 * TelaPrincipal.java
 *
 * Created on 23/08/2010, 20:20:54
 */

/**
 *
 * @author ianuck
 */
public class TelaPrincipal extends javax.swing.JFrame {
```

```
    /** Creates new form TelaPrincipal */
    public TelaPrincipal() {
        initComponents();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        bntcriaruser = new javax.swing.JButton();
        bntvalnota = new javax.swing.JButton();
        bntsair = new javax.swing.JButton();
        bnt_criarnota = new javax.swing.JButton();
        jMenuBar1 = new javax.swing.JMenuBar();
        m_arquivo = new javax.swing.JMenu();
        im_sair = new javax.swing.JMenuItem();
        m_ajuda = new javax.swing.JMenu();
        im_sobre = new javax.swing.JMenuItem();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder("Interações Possiveis"));

        bntcriaruser.setText("Criar Usuario");
```

```
bntcriaruser.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        bntcriaruserActionPerformed(evt);
    }
});

bntvalnota.setText("Validar Nota");
bntvalnota.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        bntvalnotaActionPerformed(evt);
    }
});

bntsair.setText("Sair");
bntsair.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        bntsairActionPerformed(evt);
    }
});

bnt_criarnota.setText("Criar Nota");
bnt_criarnota.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        bnt_criarnotaActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
```

```
            .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(bntvalnota, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 95, Short.MAX_VALUE)
                .addComponent(bntcriaruser, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(bntsair, javax.swing.GroupLayout.DEFAULT_SIZE, 95,
Short.MAX_VALUE)
                .addComponent(bnt_criarnota, javax.swing.GroupLayout.DEFAULT_SIZE, 95,
Short.MAX_VALUE))
            .addContainerGap())
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(bntcriaruser)
                .addGap(18, 18, 18)
                .addComponent(bntvalnota)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(bnt_criarnota)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(bntsair)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );

        m_arquivo.setText("Arquvo");
        m_arquivo.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                m_arquivoActionPerformed(evt);
            }
```

```
        });


im_sair.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_F4,
java.awt.event.InputEvent.ALT_MASK));
        im_sair.setText("Sair");
        im_sair.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                im_sairMouseClicked(evt);
            }
        });
        im_sair.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                im_sairActionPerformed(evt);
            }
        });
        m_arquivo.add(im_sair);

        jMenuBar1.add(m_arquivo);

        m_ajuda.setText("Ajuda");

        im_sobre.setText("Sobre");
        m_ajuda.add(im_sobre);

        jMenuBar1.add(m_ajuda);

        setJMenuBar(jMenuBar1);

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
```

```
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(135, 135, 135)
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(134, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(57, 57, 57)
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(38, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents

    private void bntsairActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_bntsairActionPerformed
        // TODO add your handling code here:
        System.exit(0);
    }//GEN-LAST:event_bntsairActionPerformed

    private void bntcriaruserActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_bntcriaruserActionPerformed
        // TODO add your handling code here:
        new CriarUser().show();
        this.dispose();
    }//GEN-LAST:event_bntcriaruserActionPerformed
```

```
    private void bntvalnotaActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_bntvalnotaActionPerformed
        // TODO add your handling code here:
        new ValidarNota().show();
        this.dispose();
    }//GEN-LAST:event_bntvalnotaActionPerformed

    private void im_sairMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event_im_sairMouseClicked
        // TODO add your handling code here:
        System.exit(0);
    }//GEN-LAST:event_im_sairMouseClicked

    private void m_arquivoActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_m_arquivoActionPerformed
        // TODO add your handling code here:
        System.exit(0);
    }//GEN-LAST:event_m_arquivoActionPerformed

    private void im_sairActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_im_sairActionPerformed
        // TODO add your handling code here:
        System.exit(0);
    }//GEN-LAST:event_im_sairActionPerformed

    private void bnt_criarnotaActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_bnt_criarnotaActionPerformed
        // TODO add your handling code here:
        new CriarNotas().show();
        this.dispose();
    }//GEN-LAST:event_bnt_criarnotaActionPerformed
```

```
    /**
    * @param args the command line arguments
    */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new TelaPrincipal().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton bnt_criarnota;
    private javax.swing.JButton bntcriaruser;
    private javax.swing.JButton bntsair;
    private javax.swing.JButton bntvalnota;
    private javax.swing.JMenuItem im_sair;
    private javax.swing.JMenuItem im_sobre;
    private javax.swing.JMenuBar jMenuBar1;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JMenu m_ajuda;
    private javax.swing.JMenu m_arquivo;
    // End of variables declaration//GEN-END:variables

}


/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package utilitarios;

import java.awt.Component;
import javax.swing.*;
import javax.swing.JTextField;

/**
 *
 * @author ianuck
 */
public class Utilitarios {

    public String setvlrtotal (String valor2 , String valor5 , String valor10 ,
            String valor20 , String valor50 , String valor100){

        int vlr1 = Integer.parseInt  (valor2);
        int vlr2 = Integer.parseInt(valor5);
        int vlr3 = Integer.parseInt(valor10);
        int vlr4 = Integer.parseInt(valor20);
        int vlr5 = Integer.parseInt(valor50);
        int vlr6 = Integer.parseInt(valor100);



        String vlr_total =null;

        int vlr_tl = vlr1 + vlr2 + vlr3 +
                vlr4 + vlr5 + vlr6;
        vlr_total = Integer.toString(vlr_tl);
        return vlr_total;

    }
```

```
    public String setvlrnota (String valor2 , String valor5 , String valor10 ,
        String valor20 , String valor50 , String valor100){

      int vlr1 = Integer.parseInt  (valor2);
      int vlr2 = Integer.parseInt(valor5);
      int vlr3 = Integer.parseInt(valor10);
      int vlr4 = Integer.parseInt(valor20);
      int vlr5 = Integer.parseInt(valor50);
      int vlr6 = Integer.parseInt(valor100);



      String vlr_nota =null;

      int vlr_tl = vlr1*2 + vlr2*5 + vlr3*10 +
            vlr4*20 + vlr5*50 + vlr6*100;
      vlr_nota = Integer.toString(vlr_tl);
      return vlr_nota;

   }
    public void limpaComponentes(JPanel obj){

      for (int i=0; i < obj.getComponentCount(); i++) {
         Component c = obj.getComponent(i);
         if (c instanceof JTextField) {
            JTextField field = (JTextField) c;
            field.setText("");
         }
      }
   }
}
```

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/*
 * ValidarNota.java
 *
 * Created on 24/08/2010, 11:11:42
 */

package validarnota;

import conexao.conexao;
import java.io.IOException;
import java.sql.SQLException;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;
import projetofinal.TelaPrincipal;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import mediador.Mediador;
import utilitarios.Utilitarios;

/**
 *
 * @author ianuck
 */
public class ValidarNota extends javax.swing.JFrame {
    conexao con_val_nota;
```

```
    /** Creates new form ValidarNota */
    public ValidarNota() {
        initComponents();
        con_val_nota = new conexao();
        con_val_nota.conecta();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jScrollPane1 = new javax.swing.JScrollPane();
        grid_notasvalidas = new javax.swing.JTable();
        jPanel2 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        txt_valor = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();
        txt_quant = new javax.swing.JTextField();
        jScrollPane2 = new javax.swing.JScrollPane();
        txa_info = new javax.swing.JTextArea();
        jPanel3 = new javax.swing.JPanel();
        bnt_ler = new javax.swing.JButton();
        bnt_limpar = new javax.swing.JButton();
        bnt_sair = new javax.swing.JButton();
        bnt_voltar = new javax.swing.JButton();
```

```
        jPanel4 = new javax.swing.JPanel();
        txt_2 = new javax.swing.JTextField();
        txt_5 = new javax.swing.JTextField();
        txt_10 = new javax.swing.JTextField();
        txt_20 = new javax.swing.JTextField();
        txt_50 = new javax.swing.JTextField();
        txt_100 = new javax.swing.JTextField();
        txt_total_notas = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        jLabel10 = new javax.swing.JLabel();
        jLabel11 = new javax.swing.JLabel();
        txt_total_vlr = new javax.swing.JTextField();
        jMenuBar1 = new javax.swing.JMenuBar();
        m_arquivo = new javax.swing.JMenu();
        mi_sair = new javax.swing.JMenuItem();
        m_ajuda = new javax.swing.JMenu();
        jMenuItem2 = new javax.swing.JMenuItem();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder("Notas Validas"));

        grid_notasvalidas.setBorder(javax.swing.BorderFactory.createEtchedBorder());
        grid_notasvalidas.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null, null},
                {null, null, null, null},
```

```
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null}
        },
        new String [] {
            "Valor", "Numero de Serie", "Data Fabricação", "Descrição"
        }
    ) {
        Class[] types = new Class [] {
            java.lang.String.class, java.lang.String.class, java.lang.String.class,
java.lang.String.class
        };
        boolean[] canEdit = new boolean [] {
            false, false, false, false
        };

        public Class getColumnClass(int columnIndex) {
```

```
            return types [columnIndex];
        }

        public boolean isCellEditable(int rowIndex, int columnIndex) {
            return canEdit [columnIndex];
        }
    });
    jScrollPane1.setViewportView(grid_notasvalidas);

    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 494,
Short.MAX_VALUE)
    );
    jPanel1Layout.setVerticalGroup(
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 183,
Short.MAX_VALUE)
    );

    jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder("Dados da Pesquisa"));

    jLabel1.setText("Valor Total");

    jLabel2.setText("Qantidade de notas lidas");

    txt_quant.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txt_quantActionPerformed(evt);
        }
```

```
        });

        txa_info.setBackground(new java.awt.Color(0, 0, 0));
        txa_info.setColumns(20);
        txa_info.setForeground(new java.awt.Color(0, 204, 0));
        txa_info.setRows(5);
        txa_info.setCaretColor(new java.awt.Color(0, 102, 153));
        jScrollPane2.setViewportView(txa_info);

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addComponent(jLabel1)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(txt_valor, javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addComponent(jLabel2)
                .addGap(10, 10, 10)
                .addComponent(txt_quant, javax.swing.GroupLayout.PREFERRED_SIZE, 64,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT_SIZE, 281,
Short.MAX_VALUE)
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel1)
```

```
                .addComponent(txt_valor, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel2)
                .addComponent(txt_quant, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT_SIZE, 107,
Short.MAX_VALUE))
        );

        jPanel3.setBorder(javax.swing.BorderFactory.createEtchedBorder());

        bnt_ler.setText("Ler");
        bnt_ler.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                bnt_lerActionPerformed(evt);
            }
        });

        bnt_limpar.setText("Limpar");
        bnt_limpar.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                bnt_limparActionPerformed(evt);
            }
        });

        bnt_sair.setText("Sair");
        bnt_sair.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
            bnt_sairActionPerformed(evt);
        }
    });

    bnt_voltar.setText("Voltar");
    bnt_voltar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            bnt_voltarActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
    jPanel3.setLayout(jPanel3Layout);
    jPanel3Layout.setHorizontalGroup(
        jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()
            .addContainerGap(254, Short.MAX_VALUE)
            .addComponent(bnt_ler)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(bnt_limpar)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(bnt_voltar)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(bnt_sair))
    );
    jPanel3Layout.setVerticalGroup(
        jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()
            .addContainerGap(15, Short.MAX_VALUE)
```

```
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(bnt_sair)
                .addComponent(bnt_voltar)
                .addComponent(bnt_limpar)
                .addComponent(bnt_ler))
            .addContainerGap())
        );

        jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder("Qtd por valor a pesquisar"));

        txt_2.setText("0");
        txt_2.addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {
            public void mouseMoved(java.awt.event.MouseEvent evt) {
                txt_2MouseMoved(evt);
            }
        });
        txt_2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                txt_2ActionPerformed(evt);
            }
        });
        txt_2.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyPressed(java.awt.event.KeyEvent evt) {
                txt_2KeyPressed(evt);
            }
            public void keyReleased(java.awt.event.KeyEvent evt) {
                txt_2KeyReleased(evt);
            }
            public void keyTyped(java.awt.event.KeyEvent evt) {
                txt_2KeyTyped(evt);
            }
```

```
});

txt_5.setText("0");
txt_5.addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {
   public void mouseMoved(java.awt.event.MouseEvent evt) {
      txt_5MouseMoved(evt);
   }
});
txt_5.addActionListener(new java.awt.event.ActionListener() {
   public void actionPerformed(java.awt.event.ActionEvent evt) {
      txt_5ActionPerformed(evt);
   }
});
txt_5.addKeyListener(new java.awt.event.KeyAdapter() {
   public void keyReleased(java.awt.event.KeyEvent evt) {
      txt_5KeyReleased(evt);
   }
});

txt_10.setText("0");
txt_10.addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {
   public void mouseMoved(java.awt.event.MouseEvent evt) {
      txt_10MouseMoved(evt);
   }
});
txt_10.addActionListener(new java.awt.event.ActionListener() {
   public void actionPerformed(java.awt.event.ActionEvent evt) {
      txt_10ActionPerformed(evt);
   }
});
txt_10.addKeyListener(new java.awt.event.KeyAdapter() {
   public void keyReleased(java.awt.event.KeyEvent evt) {
```

```
            txt_10KeyReleased(evt);
        }
    });

    txt_20.setText("0");
    txt_20.addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {
        public void mouseMoved(java.awt.event.MouseEvent evt) {
            txt_20MouseMoved(evt);
        }
    });
    txt_20.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txt_20ActionPerformed(evt);
        }
    });
    txt_20.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyReleased(java.awt.event.KeyEvent evt) {
            txt_20KeyReleased(evt);
        }
    });

    txt_50.setText("0");
    txt_50.addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {
        public void mouseMoved(java.awt.event.MouseEvent evt) {
            txt_50MouseMoved(evt);
        }
    });
    txt_50.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txt_50ActionPerformed(evt);
        }
    });
```

```
txt_50.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txt_50KeyReleased(evt);
    }
});

txt_100.setText("0");
txt_100.addMouseMotionListener(new java.awt.event.MouseMotionAdapter() {
    public void mouseMoved(java.awt.event.MouseEvent evt) {
        txt_100MouseMoved(evt);
    }
});
txt_100.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txt_100ActionPerformed(evt);
    }
});
txt_100.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txt_100KeyReleased(evt);
    }
});

txt_total_notas.setText("0");

jLabel3.setText("2");

jLabel4.setText("5");

jLabel5.setText("10");

jLabel6.setText("20");
```

```
        jLabel7.setText("50");

        jLabel8.setText("100");

        jLabel10.setText("Total Notas");

        jLabel11.setText("Total Valor");

        javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);
        jPanel4.setLayout(jPanel4Layout);
        jPanel4Layout.setHorizontalGroup(
            jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel4Layout.createSequentialGroup()
                .addContainerGap()

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel4Layout.createSequentialGroup()

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addGroup(jPanel4Layout.createSequentialGroup()
                                .addComponent(jLabel5)
                                .addGap(10, 10, 10)
                                .addComponent(txt_10,      javax.swing.GroupLayout.PREFERRED_SIZE,
31, javax.swing.GroupLayout.PREFERRED_SIZE))
                            .addGroup(jPanel4Layout.createSequentialGroup()
                                .addComponent(jLabel6)
                                .addGap(10, 10, 10)
                                .addComponent(txt_20,      javax.swing.GroupLayout.PREFERRED_SIZE,
31, javax.swing.GroupLayout.PREFERRED_SIZE)))
```

```
                    .addGroup(jPanel4Layout.createSequentialGroup()
                        .addComponent(jLabel4)
                        .addGap(10, 10, 10)
                        .addComponent(txt_5,    javax.swing.GroupLayout.PREFERRED_SIZE,    31,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(jPanel4Layout.createSequentialGroup()
                        .addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(txt_2,    javax.swing.GroupLayout.PREFERRED_SIZE,    31,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel4Layout.createSequentialGroup()
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
47, Short.MAX_VALUE)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                            .addComponent(jLabel7)
                            .addComponent(jLabel8))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(txt_100,     javax.swing.GroupLayout.PREFERRED_SIZE,
31, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(txt_50,      javax.swing.GroupLayout.PREFERRED_SIZE,
31, javax.swing.GroupLayout.PREFERRED_SIZE)))
                    .addGroup(jPanel4Layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jLabel10)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(txt_total_notas,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    31,
javax.swing.GroupLayout.PREFERRED_SIZE))))
                .addGroup(jPanel4Layout.createSequentialGroup()
                    .addComponent(jLabel11)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addComponent(txt_total_vlr,    javax.swing.GroupLayout.DEFAULT_SIZE,    98,
Short.MAX_VALUE)))
                .addContainerGap())
        );
        jPanel4Layout.setVerticalGroup(
            jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel4Layout.createSequentialGroup()
                .addContainerGap()

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addGroup(jPanel4Layout.createSequentialGroup()

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(txt_2,              javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jLabel3))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(txt_5,              javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jLabel4))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(txt_10,          javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel5))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(txt_20,          javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel6)
                    .addComponent(txt_total_notas, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel10)))
            .addGroup(jPanel4Layout.createSequentialGroup()

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(txt_50,          javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel7))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(txt_100,         javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel8))
                .addGap(52, 52, 52)))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(txt_total_vlr,       javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel11))
```

```
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );

        m_arquivo.setText("Arquivo");


mi_sair.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_F4,
java.awt.event.InputEvent.ALT_MASK));
        mi_sair.setText("Sair");
        mi_sair.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                mi_sairActionPerformed(evt);
            }
        });
        m_arquivo.add(mi_sair);

        jMenuBar1.add(m_arquivo);

        m_ajuda.setText("Ajuda");

        jMenuItem2.setText("Sobre");
        m_ajuda.add(jMenuItem2);

        jMenuBar1.add(m_ajuda);

        setJMenuBar(jMenuBar1);

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel3,                    javax.swing.GroupLayout.DEFAULT_SIZE,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGap(18, 18, 18)
            .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
      );
      layout.setVerticalGroup(
         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
         .addGroup(layout.createSequentialGroup()
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
               .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
               .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
      );

      pack();
   }// </editor-fold>//GEN-END:initComponents
```

```
    private void txt_quantActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_txt_quantActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_txt_quantActionPerformed

    private void bnt_sairActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_bnt_sairActionPerformed
        // TODO add your handling code here:
        System.exit(0);
    }//GEN-LAST:event_bnt_sairActionPerformed

    private void mi_sairActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_mi_sairActionPerformed
        // TODO add your handling code here:
        System.exit(0);
    }//GEN-LAST:event_mi_sairActionPerformed

    private void bnt_voltarActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_bnt_voltarActionPerformed
        // TODO add your handling code here:
        new TelaPrincipal().show();
        this.dispose();
    }//GEN-LAST:event_bnt_voltarActionPerformed

    private void txt_2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_txt_2ActionPerformed
        // TODO add your handling code here:
        Utilitarios util = new Utilitarios();

        String valor2 = txt_2.getText();
        String valor5 = txt_5.getText();
        String valor10 = txt_10.getText();
```

```
        String valor20 = txt_20.getText();
        String valor50 = txt_50.getText();
        String valor100 = txt_100.getText();



        String vlr_total = util.setvlrtotal(valor2,valor5,valor10,
                valor20,valor50,valor100);

        txt_total_notas.setText(vlr_total);

        String vlr_nota = util.setvlrnota(valor2,valor5,valor10,
                valor20,valor50,valor100);
        txt_total_vlr.setText(vlr_nota);



    }//GEN-LAST:event_txt_2ActionPerformed

    private void txt_20ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_txt_20ActionPerformed
        // TODO add your handling code here:
        Utilitarios util = new Utilitarios();

        String valor2 = txt_2.getText();
        String valor5 = txt_5.getText();
        String valor10 = txt_10.getText();
        String valor20 = txt_20.getText();
        String valor50 = txt_50.getText();
        String valor100 = txt_100.getText();



        String vlr_total = util.setvlrtotal(valor2,valor5,valor10,
                valor20,valor50,valor100);
```

```
        txt_total_notas.setText(vlr_total);

         String vlr_nota = util.setvlrnota(valor2,valor5,valor10,
                valor20,valor50,valor100);
        txt_total_vlr.setText(vlr_nota);

    }//GEN-LAST:event_txt_20ActionPerformed

    private void txt_100ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_txt_100ActionPerformed
        // TODO add your handling code here:
        Utilitarios util = new Utilitarios();

        String valor2 = txt_2.getText();
        String valor5 = txt_5.getText();
        String valor10 = txt_10.getText();
        String valor20 = txt_20.getText();
        String valor50 = txt_50.getText();
        String valor100 = txt_100.getText();



        String vlr_total = util.setvlrtotal(valor2,valor5,valor10,
                valor20,valor50,valor100);

        txt_total_notas.setText(vlr_total);

         String vlr_nota = util.setvlrnota(valor2,valor5,valor10,
                valor20,valor50,valor100);
        txt_total_vlr.setText(vlr_nota);

    }//GEN-LAST:event_txt_100ActionPerformed
```

```
  private void txt_5ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_txt_5ActionPerformed
      // TODO add your handling code here:
      Utilitarios util = new Utilitarios();

      String valor2 = txt_2.getText();
      String valor5 = txt_5.getText();
      String valor10 = txt_10.getText();
      String valor20 = txt_20.getText();
      String valor50 = txt_50.getText();
      String valor100 = txt_100.getText();



      String vlr_total = util.setvlrtotal(valor2,valor5,valor10,
            valor20,valor50,valor100);

      txt_total_notas.setText(vlr_total);

       String vlr_nota = util.setvlrnota(valor2,valor5,valor10,
            valor20,valor50,valor100);
      txt_total_vlr.setText(vlr_nota);

  }//GEN-LAST:event_txt_5ActionPerformed

  private void txt_10ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_txt_10ActionPerformed
      // TODO add your handling code here:
      Utilitarios util = new Utilitarios();

      String valor2 = txt_2.getText();
      String valor5 = txt_5.getText();
```

```
        String valor10 = txt_10.getText();
        String valor20 = txt_20.getText();
        String valor50 = txt_50.getText();
        String valor100 = txt_100.getText();



        String vlr_total = util.setvlrtotal(valor2,valor5,valor10,
                valor20,valor50,valor100);

        txt_total_notas.setText(vlr_total);

         String vlr_nota = util.setvlrnota(valor2,valor5,valor10,
                valor20,valor50,valor100);
        txt_total_vlr.setText(vlr_nota);

    }//GEN-LAST:event_txt_10ActionPerformed

    private void txt_50ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_txt_50ActionPerformed
        // TODO add your handling code here:
        Utilitarios util = new Utilitarios();

        String valor2 = txt_2.getText();
        String valor5 = txt_5.getText();
        String valor10 = txt_10.getText();
        String valor20 = txt_20.getText();
        String valor50 = txt_50.getText();
        String valor100 = txt_100.getText();



        String vlr_total = util.setvlrtotal(valor2,valor5,valor10,
                valor20,valor50,valor100);
```

```
        txt_total_notas.setText(vlr_total);

         String vlr_nota = util.setvlrnota(valor2,valor5,valor10,
                valor20,valor50,valor100);
        txt_total_vlr.setText(vlr_nota);

    }//GEN-LAST:event_txt_50ActionPerformed

    private void txt_2MouseMoved(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_txt_2MouseMoved
        // TODO add your handling code here:
        Utilitarios util = new Utilitarios();

        String valor2 = txt_2.getText();
        String valor5 = txt_5.getText();
        String valor10 = txt_10.getText();
        String valor20 = txt_20.getText();
        String valor50 = txt_50.getText();
        String valor100 = txt_100.getText();



        String vlr_total = util.setvlrtotal(valor2,valor5,valor10,
                valor20,valor50,valor100);

        txt_total_notas.setText(vlr_total);

         String vlr_nota = util.setvlrnota(valor2,valor5,valor10,
                valor20,valor50,valor100);
        txt_total_vlr.setText(vlr_nota);

    }//GEN-LAST:event_txt_2MouseMoved
```

```
  private void txt_5MouseMoved(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_txt_5MouseMoved
      // TODO add your handling code here:
      Utilitarios util = new Utilitarios();

      String valor2 = txt_2.getText();
      String valor5 = txt_5.getText();
      String valor10 = txt_10.getText();
      String valor20 = txt_20.getText();
      String valor50 = txt_50.getText();
      String valor100 = txt_100.getText();



      String vlr_total = util.setvlrtotal(valor2,valor5,valor10,
            valor20,valor50,valor100);

      txt_total_notas.setText(vlr_total);

       String vlr_nota = util.setvlrnota(valor2,valor5,valor10,
            valor20,valor50,valor100);
      txt_total_vlr.setText(vlr_nota);

  }//GEN-LAST:event_txt_5MouseMoved

  private void txt_10MouseMoved(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_txt_10MouseMoved
      // TODO add your handling code here:
      Utilitarios util = new Utilitarios();

      String valor2 = txt_2.getText();
      String valor5 = txt_5.getText();
```

```
        String valor10 = txt_10.getText();
        String valor20 = txt_20.getText();
        String valor50 = txt_50.getText();
        String valor100 = txt_100.getText();



        String vlr_total = util.setvlrtotal(valor2,valor5,valor10,
                valor20,valor50,valor100);

        txt_total_notas.setText(vlr_total);

         String vlr_nota = util.setvlrnota(valor2,valor5,valor10,
                valor20,valor50,valor100);
        txt_total_vlr.setText(vlr_nota);

    }//GEN-LAST:event_txt_10MouseMoved

    private void txt_20MouseMoved(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_txt_20MouseMoved
        // TODO add your handling code here:
        Utilitarios util = new Utilitarios();

        String valor2 = txt_2.getText();
        String valor5 = txt_5.getText();
        String valor10 = txt_10.getText();
        String valor20 = txt_20.getText();
        String valor50 = txt_50.getText();
        String valor100 = txt_100.getText();



        String vlr_total = util.setvlrtotal(valor2,valor5,valor10,
                valor20,valor50,valor100);
```

```
        txt_total_notas.setText(vlr_total);

         String vlr_nota = util.setvlrnota(valor2,valor5,valor10,
                valor20,valor50,valor100);
        txt_total_vlr.setText(vlr_nota);

    }//GEN-LAST:event_txt_20MouseMoved

    private void txt_50MouseMoved(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_txt_50MouseMoved
        // TODO add your handling code here:
        Utilitarios util = new Utilitarios();

        String valor2 = txt_2.getText();
        String valor5 = txt_5.getText();
        String valor10 = txt_10.getText();
        String valor20 = txt_20.getText();
        String valor50 = txt_50.getText();
        String valor100 = txt_100.getText();



        String vlr_total = util.setvlrtotal(valor2,valor5,valor10,
                valor20,valor50,valor100);

        txt_total_notas.setText(vlr_total);

         String vlr_nota = util.setvlrnota(valor2,valor5,valor10,
                valor20,valor50,valor100);
        txt_total_vlr.setText(vlr_nota);

    }//GEN-LAST:event_txt_50MouseMoved
```

```
  private void txt_100MouseMoved(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_txt_100MouseMoved
      // TODO add your handling code here:
      Utilitarios util = new Utilitarios();

      String valor2 = txt_2.getText();
      String valor5 = txt_5.getText();
      String valor10 = txt_10.getText();
      String valor20 = txt_20.getText();
      String valor50 = txt_50.getText();
      String valor100 = txt_100.getText();



      String vlr_total = util.setvlrtotal(valor2,valor5,valor10,
            valor20,valor50,valor100);

      txt_total_notas.setText(vlr_total);

       String vlr_nota = util.setvlrnota(valor2,valor5,valor10,
            valor20,valor50,valor100);
      txt_total_vlr.setText(vlr_nota);

  }//GEN-LAST:event_txt_100MouseMoved

  private void txt_2KeyPressed(java.awt.event.KeyEvent evt) {//GEN-
FIRST:event_txt_2KeyPressed
      // TODO add your handling code here:
  }//GEN-LAST:event_txt_2KeyPressed

  private void txt_2KeyTyped(java.awt.event.KeyEvent evt) {//GEN-
FIRST:event_txt_2KeyTyped
```

```
        // TODO add your handling code here:
    }//GEN-LAST:event_txt_2KeyTyped

    private void txt_2KeyReleased(java.awt.event.KeyEvent evt) {//GEN-
FIRST:event_txt_2KeyReleased
        // TODO add your handling code here:
        Utilitarios util = new Utilitarios();

        String valor2 = txt_2.getText();
        String valor5 = txt_5.getText();
        String valor10 = txt_10.getText();
        String valor20 = txt_20.getText();
        String valor50 = txt_50.getText();
        String valor100 = txt_100.getText();



        String vlr_total = util.setvlrtotal(valor2,valor5,valor10,
                valor20,valor50,valor100);

        txt_total_notas.setText(vlr_total);

         String vlr_nota = util.setvlrnota(valor2,valor5,valor10,
                valor20,valor50,valor100);
        txt_total_vlr.setText(vlr_nota);

    }//GEN-LAST:event_txt_2KeyReleased

    private void txt_5KeyReleased(java.awt.event.KeyEvent evt) {//GEN-
FIRST:event_txt_5KeyReleased
        // TODO add your handling code here:
        Utilitarios util = new Utilitarios();
```

```
        String valor2 = txt_2.getText();
        String valor5 = txt_5.getText();
        String valor10 = txt_10.getText();
        String valor20 = txt_20.getText();
        String valor50 = txt_50.getText();
        String valor100 = txt_100.getText();



        String vlr_total = util.setvlrtotal(valor2,valor5,valor10,
                valor20,valor50,valor100);

        txt_total_notas.setText(vlr_total);

         String vlr_nota = util.setvlrnota(valor2,valor5,valor10,
                valor20,valor50,valor100);
        txt_total_vlr.setText(vlr_nota);

    }//GEN-LAST:event_txt_5KeyReleased

    private void txt_10KeyReleased(java.awt.event.KeyEvent evt) {//GEN-
FIRST:event_txt_10KeyReleased
        // TODO add your handling code here:
        Utilitarios util = new Utilitarios();

        String valor2 = txt_2.getText();
        String valor5 = txt_5.getText();
        String valor10 = txt_10.getText();
        String valor20 = txt_20.getText();
        String valor50 = txt_50.getText();
        String valor100 = txt_100.getText();
```

```
        String vlr_total = util.setvlrtotal(valor2,valor5,valor10,
                valor20,valor50,valor100);

        txt_total_notas.setText(vlr_total);

         String vlr_nota = util.setvlrnota(valor2,valor5,valor10,
                valor20,valor50,valor100);
        txt_total_vlr.setText(vlr_nota);

    }//GEN-LAST:event_txt_10KeyReleased

    private void txt_20KeyReleased(java.awt.event.KeyEvent evt) {//GEN-
FIRST:event_txt_20KeyReleased
        // TODO add your handling code here:
        Utilitarios util = new Utilitarios();

        String valor2 = txt_2.getText();
        String valor5 = txt_5.getText();
        String valor10 = txt_10.getText();
        String valor20 = txt_20.getText();
        String valor50 = txt_50.getText();
        String valor100 = txt_100.getText();



        String vlr_total = util.setvlrtotal(valor2,valor5,valor10,
                valor20,valor50,valor100);

        txt_total_notas.setText(vlr_total);

         String vlr_nota = util.setvlrnota(valor2,valor5,valor10,
                valor20,valor50,valor100);
        txt_total_vlr.setText(vlr_nota);
```

```
    }//GEN-LAST:event_txt_20KeyReleased

    private void txt_50KeyReleased(java.awt.event.KeyEvent evt) {//GEN-
FIRST:event_txt_50KeyReleased
        // TODO add your handling code here:
        Utilitarios util = new Utilitarios();

        String valor2 = txt_2.getText();
        String valor5 = txt_5.getText();
        String valor10 = txt_10.getText();
        String valor20 = txt_20.getText();
        String valor50 = txt_50.getText();
        String valor100 = txt_100.getText();



        String vlr_total = util.setvlrtotal(valor2,valor5,valor10,
                valor20,valor50,valor100);

        txt_total_notas.setText(vlr_total);

         String vlr_nota = util.setvlrnota(valor2,valor5,valor10,
                valor20,valor50,valor100);
        txt_total_vlr.setText(vlr_nota);

    }//GEN-LAST:event_txt_50KeyReleased

    private void txt_100KeyReleased(java.awt.event.KeyEvent evt) {//GEN-
FIRST:event_txt_100KeyReleased
        // TODO add your handling code here:
        Utilitarios util = new Utilitarios();
```

```
        String valor2 = txt_2.getText();
        String valor5 = txt_5.getText();
        String valor10 = txt_10.getText();
        String valor20 = txt_20.getText();
        String valor50 = txt_50.getText();
        String valor100 = txt_100.getText();



        String vlr_total = util.setvlrtotal(valor2,valor5,valor10,
                valor20,valor50,valor100);

        txt_total_notas.setText(vlr_total);

         String vlr_nota = util.setvlrnota(valor2,valor5,valor10,
                valor20,valor50,valor100);
        txt_total_vlr.setText(vlr_nota);

    }//GEN-LAST:event_txt_100KeyReleased

    private void bnt_limparActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_bnt_limparActionPerformed
        // TODO add your handling code here:
        Utilitarios util = new Utilitarios();
        JPanel obj = jPanel2;
        util.limpaComponentes(obj);
        obj = jPanel4;
        util.limpaComponentes(obj);
        txa_info.setText("Informações Sobre a Leitura");
        //add(txa_info);

    }//GEN-LAST:event_bnt_limparActionPerformed
```

```
    private void bnt_lerActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_bnt_lerActionPerformed
        // TODO add your handling code here:
        Mediador mediador = new Mediador();
        String dleitura = null;
        try {
            dleitura = mediador.execmediador();
        } catch (IOException ex) {
            Logger.getLogger(ValidarNota.class.getName()).log(Level.SEVERE, null, ex);
        } catch (InterruptedException ex) {
            Logger.getLogger(ValidarNota.class.getName()).log(Level.SEVERE, null, ex);
        }
        try {
            prencher_tabela(dleitura);
        } catch (SQLException ex) {
            Logger.getLogger(ValidarNota.class.getName()).log(Level.SEVERE, null, ex);
        }




    }//GEN-LAST:event_bnt_lerActionPerformed


    public void prencher_tabela (String dleitura) throws SQLException {
       String novaString = null;
            String novaStringII = null;
            Scanner sc = new Scanner(dleitura);
            int ilinhas=0;
        int i = 0;
        int vltt = 0;
        int vl2 = 0 , vl5= 0, vl10= 0, vl20= 0, vl50= 0, vl100 = 0;
```

```
        while (sc.hasNextLine())
        {
        ilinhas++;
        novaString = sc.nextLine();
        System.out.println(novaString);
        if (ilinhas > 1){
           novaStringII = novaString.substring(4,22);
           String sql = ("Select * from TB_NOTA where desc_valor like '"
                 +novaStringII.replaceAll(" ","")+"'");
           con_val_nota.executeSQL(sql);

           if (con_val_nota.resultset.next()){

              DefaultTableModel modelo = (DefaultTableModel)grid_notasvalidas.getModel();
              modelo.setNumRows(i);
              i++;
              try {
                 /*while(*/
              /* con_val_nota.resultset.next();*/
              modelo.addRow(new Object [] {con_val_nota.resultset.getString("valor"),
con_val_nota.resultset.getString("serie"), con_val_nota.resultset.getString("data_fabri"),
con_val_nota.resultset.getString("desc_valor")});

              int vlr = Integer.parseInt(con_val_nota.resultset.getString("valor"));
              vltt = vltt + vlr;

              if ( vlr == 2){
                vl2++;

              }
              if (vlr == 5){
                vl5++;
```

```
            }
            if (vlr == 10){
              vl10++;
            }
            if (vlr == 20){
              vl20++;
            }
            if (vlr == 50){
              vl50++;
            }
            if (vlr == 100){
              vl100++;
            }

            }
            catch (SQLException erro)
            {
               JOptionPane.showMessageDialog(null,"Erro ao listar no JTable" +erro);
            }


        }
        else
        {
           JOptionPane.showMessageDialog(null, "Tag não lida, possivel falsificação");
        }
    }
    }
String vlr_tt = Integer.toString(vltt);
txt_valor.setText(vlr_tt);
String quant = Integer.toString(i);
txt_quant.setText(quant);
```

```
    txa_info.setText("Foram lida  \n"
            +vl2+ " de Dois Reais \n"
            +vl5+ " de Cinco Reais \n"
            +vl10+ " de Dez Reais \n"
            +vl20+ " de Vinte Reais \n"
            +vl50+ " de Ciquenta Reais \n"
            +vl100+ " de Cem Reais \n");

}



/**
* @param args the command line arguments
*/
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new ValidarNota().setVisible(true);
        }
    });
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton bnt_ler;
private javax.swing.JButton bnt_limpar;
private javax.swing.JButton bnt_sair;
private javax.swing.JButton bnt_voltar;
private javax.swing.JTable grid_notasvalidas;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
```

```
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JMenuBar jMenuBar1;
    private javax.swing.JMenuItem jMenuItem2;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JPanel jPanel4;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JMenu m_ajuda;
    private javax.swing.JMenu m_arquivo;
    private javax.swing.JMenuItem mi_sair;
    private javax.swing.JTextArea txa_info;
    private javax.swing.JTextField txt_10;
    private javax.swing.JTextField txt_100;
    private javax.swing.JTextField txt_2;
    private javax.swing.JTextField txt_20;
    private javax.swing.JTextField txt_5;
    private javax.swing.JTextField txt_50;
    private javax.swing.JTextField txt_quant;
    private javax.swing.JTextField txt_total_notas;
    private javax.swing.JTextField txt_total_vlr;
    private javax.swing.JTextField txt_valor;
    // End of variables declaration//GEN-END:variables}
```