



*ESCOLA POLITÉCNICA  
DE PERNAMBUCO*



**POLI**  
**Departamento de  
Sistemas  
Computacionais**

# **Suporte à Decisão em Sistemas Inteligentes de Colheitas Agrícolas**

**Trabalho de Conclusão de Curso  
Engenharia da Computação**

**Diogo Ferreira Pacheco**

**Orientador: Prof. Fernando Buarque de Lima Neto**

**Recife, maio de 2006.**



UNIVERSIDADE  
DE PERNAMBUCO



*ESCOLA POLITÉCNICA  
DE PERNAMBUCO*

**POLI**  
ESCOLA POLITÉCNICA  
DE PERNAMBUCO



**Departamento de  
Sistemas  
Computacionais**

# **Suporte à Decisão em Sistemas Inteligentes de Colheitas Agrícolas**

## **Trabalho de Conclusão de Curso Engenharia da Computação**

Este Projeto é apresentado como requisito parcial para obtenção do diploma de Bacharel em Engenharia da Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

**Diogo Ferreira Pacheco**

**Orientador: Prof. Fernando Buarque de Lima Neto**

**Recife, maio de 2006.**



UNIVERSIDADE  
DE PERNAMBUCO

Diogo Ferreira Pacheco

# **Suporte à Decisão em Sistemas Inteligentes de Colheitas Agrícolas**

## Resumo

Aparentemente de fácil execução, a colheita da cana-de-açúcar inclui um processo decisório muito complexo que pode refletir diretamente na produtividade e conseqüentemente, na lucratividade de toda a safra agrícola. Paradoxalmente, o grande volume informações disponibilizadas pelos sistemas de informação nas usinas sucro-alcooleiras agrega pouco valor à decisão de gestores sobre que lotes devem ser colhidos.

Utilizando-se técnicas de redes neurais artificiais para a modelagem do problema da colheita da cana-de-açúcar, este trabalho apresenta um aplicativo computacional inteligente de apoio à decisão que disponibiliza informações gerenciais, com mais precisão e antecedência (e.g. PCC, TCH e Fibra). Essas informações passam a alimentar um sistema de suporte à decisão (SSD) que é capaz de resolver problemas semi-estruturados de maneira interativa e rápida, através de modelos matemáticos e cenários econômicos parametrizados por seus usuários. O sistema sugere decisões de colheita.

A implementação deste sistema seguiu a metodologia proposta por [Buarque98] e com a modelagem aprimorada por [Pacheco05] realizando um desenvolvimento de uma ferramenta pioneira na área.

Espera-se que a adoção deste sistema por usinas de açúcar possa permitir ao agente de decisão conhecer o momento mais apropriado para a colheita de qualquer variedade cultivada em sua unidade.

Os resultados experimentais obtidos podem possibilitar a construção de uma ferramenta que implemente e estenda o modelo utilizado neste trabalho no trato de outras atividades relacionadas com o manejo da cana-de-açúcar ou a exportação do modelo para outras culturas agrícolas.

## Abstract

Despite of apparently easy execution, the sugar cane harvest includes a considerably complex decision process that can directly affect the productivity and consequently, the profitability of all agricultural harvest. Paradoxically the great volume of information disponibilized by information systems in the sugary factories does not help management decision about which lots must be harvested.

Using techniques of artificial neural nets to model the problem of sugar cane's harvest, this work presents an intelligent software of decision support that offers management information, with more precision and antecedence (e.g. PCC, TCH and fiber). These information are sent to a decision support system (DSS) that can resolves semi-structured problems in an interactive and fast way, through mathematical models and economic scenes, filled by its users. The system suggests the harvest.

The system implementation followed the proposal methodology by [Buarque98] and with a modeling improvement by [Pacheco05] carrying through a development of a pioneering tool in this area.

It is expected that the adoption of this system by sugar factories can allow the decision agent to know the most appropriate moment for the harvest date of any variety cultivated in its unit.

Experimental results suggests construction of a tool that implements and extends the model used in this work in the treatment of other activities related with the sugar cane handling or the model application in others agricultural cultures.

# Sumário

<b>Índice de Figuras</b>	<b>viii</b>
<b>Índice de Tabelas e Equações</b>	<b>ix</b>
<b>Agradecimentos</b>	<b>x</b>
<b>1 Introdução</b>	<b>11</b>
<b>2 Sistemas de Informação</b>	<b>14</b>
2.1 Valor da Informação - Decisão	15
2.2 Tipos de SI	16
2.3 Sistemas de Suporte à Decisão	19
2.3.1 Componentes de um SSD – Banco de Dados	20
2.3.2 Componentes de um SSD – Sistema de Software	20
2.3.3 Componentes de um SSD – Interface	21
2.4 Características desejáveis de SSD para o problema alvo deste trabalho	21
2.5 Aspectos Web relevantes ao sistema	22
<b>3 Redes Neurais Artificiais</b>	<b>23</b>
3.1 Definição	24
3.2 Arquiteturas de Rede	25
3.3 Processos de Aprendizagem	26
3.4 Multi-Layer Perceptron	28
3.4.1 Parametrização	28
3.4.2 Treinamento	28
<b>4 O Sistema Desenvolvido</b>	<b>33</b>
4.1 Características	33
4.2 Funcionalidades	34
4.3 Arquitetura	42
4.4 Suporte à Decisão Baseado em Tecnologia Neural	45
4.5 Ferramentas e Tecnologias Utilizadas	46
<b>5 Experimentos e Resultados Obtidos</b>	<b>47</b>
5.1 Manipulação dos dados	47
5.2 Validação do simulador neural	49
5.3 Validação da aplicação de decisão	52
<b>6 Conclusões e Trabalhos Futuros</b>	<b>55</b>
6.1 Contribuições	55
6.2 Discussão	56
6.3 Trabalhos Futuros	56

<b>Bibliografia</b>	<b>58</b>
<b>Anexo A</b>	<b>61</b>
<b>Anexo B</b>	<b>69</b>
<b>Anexo C</b>	<b>73</b>
▪ Eclipse	73
▪ Tomcat	73
▪ Dreamweaver	73
▪ Struts	74
▪ Log4J	75
<b>Apêndice A</b>	<b>76</b>

# Índice de Figuras

Fig. 1 - Detalhes da produção de cana, açúcar e álcool no Brasil nas safras 1994-2002 .....	12
Fig. 2 - Processo genérico de tomada de decisão (adaptado de [Laudon00]) .....	16
Fig. 3 – Tipo de decisão x Características da Informação .....	17
Fig. 4 - Classificação de SI de proposta por O'Brien [O'Brien01] .....	18
Fig. 5 – Componentes de um SSD (adaptado de [Laudon00]) .....	20
Fig. 6 – Neurônio: célula do sistema nervoso [Tatibana06] .....	23
Fig. 7 – Neurônio artificial projetado por McCulloch e Pitts [MCP43] .....	24
Fig. 8 – Rede alimentada adiante ou acíclica com uma única camada de neurônios.....	25
Fig. 9 – Rede alimentada adiante ou acíclica totalmente conectada com uma camada oculta e uma camada de saída .....	26
Fig. 10 – Rede recorrente com neurônios ocultos.....	26
Fig. 11 – Curvas de aprendizado.....	30
Fig. 12 – Relação entre taxa de aprendizado e convergência a partir do ajuste de pesos.....	30
Fig. 13 – Gráfico mostrando parada antecipada do treinamento usando conjunto de validação.....	31
Fig. 14 – Diagrama de casos de uso para o sistema desenvolvido.....	35
Fig. 15 – Tela de login onde é necessária a autenticação do usuário. ....	36
Fig. 16 – Aba de funcionalidades do sistema .....	36
Fig. 17 – Tela de predição .....	37
Fig. 18 – Tela de simulação .....	38
Fig. 19 – Tela de resultados.....	39
Fig. 20 – Tela de processamento .....	40
Fig. 21 – Tela de logs .....	40
Fig. 22 – Tela sobre o sistema.....	41
Fig. 23 – Diagrama de seqüência da funcionalidade <i>colheita inteligente</i> .....	42
Fig. 24 - Representação da arquitetura <i>Model-View-Controller</i> . Como os componentes se relacionam .....	43
Fig. 25 - Diagrama de classes simplificado do projeto .....	44
Fig. 26 – Amostra dos dados originais, antes de serem pré-processados.....	48
Fig. 27 – Amostra de dados pré processados e prontos para serem apresentados à RNA....	49
Fig. 28 – Tela de resultados com <i>zoom</i> nos lotes indicados para colheita.....	54
Fig. 29 – Diagrama de classes completo (parte 1/5).....	69
Fig. 30 – Diagrama de classes completo (parte 2/5).....	70
Fig. 31 – Diagrama de classes completo (parte 3/5).....	70
Fig. 32 – Diagrama de classes completo (parte 4/5).....	71
Fig. 33 – Diagrama de classes completo (parte 5/5).....	72
Fig. 34 – Relacionamentos em Struts.....	74



# Índice de Tabelas e Equações

Tabela 1 – Classificação de SI por sua estrutura [Lucas97] .....	17
Tabela 1 – Definição dos dados pré-processados .....	49
Tabela 2 – Resumo dos enfoques testados no projeto .....	50
Tabela 3 – Detalhes do treinamento da RNA para o <i>enfoque 0</i> em [Pacheco05] .....	50
Tabela 4 - Detalhes do novo treinamento da RNA para o <i>enfoque 0</i> .....	50
Tabela 5 - Detalhes do treinamento da RNA para o <i>enfoque 1</i> em [Pacheco05] .....	50
Tabela 6 – Detalhes do novo treinamento da RNA para o <i>enfoque 1</i> .....	51
Tabela 7 - Detalhes do treinamento da RNA para o <i>enfoque 2</i> em [Pacheco05] .....	51
Tabela 8 – Detalhes do novo treinamento da RNA para o <i>enfoque 2</i> .....	51
Tabela 9 – Resumo dos enfoques testados em [Pacheco05] e neste trabalho, comparados em relação ao EMQ .....	51
Tabela 10 – Parte do conjunto de lotes com valores de predição gerados pelo sistema.....	52
Tabela 11 – Solução apresentada por um especialista agrícola a partir de um arquivo de lotes preditos e da definição de um cenário de colheita .....	53
Tabela 12 - Brasil: Cana de açúcar – produção, área colhida e rendimento médio – 1990 a 2004 [MinistérioAgricultura06].....	76
Tabela 13 - Brasil: Cana-de-açúcar – exportações, importações e principais países, produtos selecionados – 1996 a 2004 [MinistérioAgricultura06] .....	77
Equação 1 – Fórmula de calculo do erro médio .....	30

# Agradecimentos

Agradeço a Deus e aos anjos que me deram forças durante toda minha vida. Nunca me abandonaram, principalmente, nestes últimos cinco anos em que deixei casa, família, amigos e minha amada cidade natal, Maceió, em busca de um sonho, uma formação profissional mais qualificada.

Que no lugar da casa - me arrumou um novo lar; no lugar da família e dos amigos – me apresentou vários novos amigos e alguns destes agora serão para sempre minha família postiça; no lugar da cidade – Recife que durante algum tempo foi causa de minhas reclamações, mas hoje, me acolhe, me trouxe o primeiro emprego e está possibilitando a tão almejada formação superior.

Agradeço a Deus pela minha família – Pacheco (painho), Geyne (mainha) e Igor (irmão) – os quais são responsáveis diretos pela minha educação e princípios. Por sempre terem sido a base da minha vida e minha fonte de forças para eu nunca desistir. Tios (as), primos (as), minhas avós Déu e Lila, e meu avô Di (*in memoriam*) que sempre (aos seus modos) souberam me aconselhar, repreender, parabenizar e incentivar.

Agradeço a Deus pelo meu professor, supervisor, orientador e amigo Fernando Buarque que muito me ensinou ao longo destes anos (iniciação científica, estágio supervisionado, salas de aula, orientação do TCC) e principalmente, pela “injeção” de ânimo nesta reta final. Sem o seu apoio, estas palavras demorariam, pelo menos, mais seis meses para serem lidas.

Agradeço a Deus pelos amigos do CESAR que compartilharam seus conhecimentos e me ensinaram a trabalhar. Em especial a Rodrigo Mendes, Daniel Melo, Ivan Leite e Bruna Buzen que me “chacoalharam” quando eu estava decidido a “morrer na praia”. A minha gerente Neildes Pedrosa que me ouviu, entendeu e ajudou, antecipando 15 dias de férias – sem esse tempo, eu não teria conseguido!

Agradeço a Deus pelos *amigosPoli*, amigos, irmãos, companheiros e cúmplices. Em especial a dupla Filipe Regueira e Herbert Menezes – maiores parceiros de estudo e farras.

Agradeço a Deus pelo AP901 e agora AP601 que possui moradores iluminados e são testemunhas oculares de toda minha trajetória em Recife.

Agradeço a Deus pela minha namorada que sempre apoiou minhas decisões e foi responsável pela minha sanidade neste último mês de intensa pressão e trabalho.

Por fim peço desculpas a Deus pelas pessoas que esqueci de mencionar e que com simples palavras (ditas ou ouvidas) traçaram meu caminho.

A todos vocês, meu MUITO OBRIGADO!

# Capítulo 1

## Introdução

A cultura da cana-de-açúcar gera por ano milhões de dólares em todo o mundo [AlcoolBrás04]. No nordeste brasileiro, se destaca como uma das principais atividades econômicas, notadamente como insumo fundamental para a produção de açúcar, álcool e combustível.

Dado que o agronegócio nacional é responsável por 33% do Produto Interno Bruto (PIB), 42% das exportações totais e 37% dos empregos [MinistérioAgricultura06], o país é o maior produtor mundial de cana, com uma área plantada de 5,4 milhões de hectares e uma safra anual de cerca de 354 milhões de toneladas [MinistérioAgricultura06]; que o cultivo da cana-de-açúcar deve gerar nos próximos seis anos investimentos de cinco bilhões de reais e 300 mil empregos [JornalDaGlobo05]; por essa evidente relevância sócio-econômica para a nosso país e região (ver Fig. 1), decidimos dedicar o nosso trabalho de conclusão de curso ao setor sucro-alcooleiro. Especificamente, à colheita de cana-de-açúcar. Mais informações sobre essa atividade econômica, ver tabelas no Apêndice A.

A cultura da cana-de-açúcar não é uma atividade simples. As usinas sucro-alcooleira necessitam de bom desempenho e produtividade em todas as etapas da cultura do vegetal: plantio, tratos culturais e colheita. Uma fase importante, e não menos complexa, é o período de colheita onde gerentes têm que tomar uma difícil decisão – quais lotes devem ser colhidos. Diariamente esta decisão tem de ser tomada durante a safra de maneira a satisfazer as regras de negócio, particulares a cada usina. Contribuem para a referida complexidade os inúmeros fatores ligados à maturação da cana-de-açúcar [Alexander73].

Ao longo dos anos, o volume e a velocidade com que novos dados são gerados é diretamente proporcional à importância e risco das decisões tomadas por gestores em organizações empresariais. Sistemas de informação têm ajudado gerentes a comunicarem e distribuírem informação, mas pouco contribuíram no processo de tomada de decisão [Laudon00].

Apesar da informatização que é praticamente regra em todas as grandes usinas que produzem açúcar e álcool no Brasil, os atuais gestores dispõem de poucas ferramentas que os auxiliem mais efetivamente na decisão de colheita da matéria prima [Buarque98]. Na prática, os gerentes apenas dispõem de uma miríade de linhas de informações em relatórios e consultas em tela sobre os lotes cultivados em suas propriedades. Não raro as informações disponíveis são apenas: variedade da cana, topografia, época do plantio, tipo do solo, PCC<sup>1</sup> de safras anteriores. Assomam-se a essas informações de resultados laboratoriais de pré-colheita e outras informações industriais esparsas.

---

<sup>1</sup> PCC – teor de sacarose da cana-de-açúcar a ser colhida.

Da maneira como atualmente são apresentadas as informações para decisão de colheita, elas agregam pouco valor ao processo gerencial que é praticamente empírico. Isto, pois, não são consideradas todas as possíveis interconexões existentes nos dados atuais. Em atendimento a essa necessidade, técnicas de neurocomputação [Haykin94] aplicadas ao problema da colheita da cana-de-açúcar foram propostas e demonstradas eficazes em domínios agrícolas [Buarque98] [Pacheco05]. Estes resultados, naturalmente, encorajam a implementação de um sistema de suporte à decisão usando os resultados de predição advindos de algoritmos baseados em inteligência artificial. Já em 1998 isto havia sido proposto: os sistemas de suporte à decisão neurais (SSDn, do inglês, *neural decision support systems*, nDSS) [Buarque98].

Safra Crop Cosecha	Anidro (m³) Dehydrated alcohol Anidro	Álcool Hidratado (m³) Hidrated Alcohol Alcohol hidratado	Álcool Total (m³) Total Alcohol Alcohol Total	Açúcar (ton.) Sugar Azúcar	Cana-de-Açúcar (ton.) Sugar Cane Caña de Azúcar
94/95	2.873.470	9.892.440	12.765.910	11.700.464	240.867.791
95/96	3.057.557	9.659.202	12.716.759	12.651.084	249.876.575
96/97	4.629.340	9.801.729	14.431.069	13.631.88	289.529.382
97/98	5.697.083	9.726.026	15.423.109	14.848.345	302.193.415
98/99	5.691.872	8.236.059	13.927.931	17.960.587	315.640.797
99/00	6.140.769	6.936.996	13.077.765	19.380.196	310.122.784
00/01	5.584.730	4.932.805	10.517.535	16.020.340	254.921.721
01/02	6.466.835	4.976.801	11.443.936	18.889.321	291.209.635

Fonte/Source/Fuente: MAPA/SPC/DAA

Fig. 1 - Detalhes da produção de cana, açúcar e álcool no Brasil nas safras 1994-2002

O maior objetivo deste trabalho é o desenvolvimento de um SSDn que venha agregar valor à escolha gerencial, ou seja, uma ferramenta que sirva de subsídio, um balizador que indique as melhores configurações de colheita baseado na predição de três importantes fatores de produtividade agrícola: PCC, TCH<sup>2</sup> e Fibra<sup>3</sup>. Espera-se também, validar o motor neural usado como simulador na predição destes valores.

Neste cenário (colheita da cana-de-açúcar), onde o problema é intrinsecamente complexo e trata com grande volume de dados, o gestor decide a colheita quase que empiricamente. Com a ferramenta a ser desenvolvida neste trabalho de conclusão de curso (TCC), pretende-se extrair e correlacionar as informações agrícolas existentes nos sistemas e gerar informações que apoiem decisões de gestão. As conseqüências prováveis serão: melhor gestão do negócio e aumento do desempenho agrícola.

O sistema a ser desenvolvido será parametrizado pelos gestores, agilizando as respostas gerenciais e tornando as mesmas mais fundamentadas em fatos e não em heurísticas. Neste contexto, duas grandes áreas serão abordadas: computação neural inteligente, como motor no

<sup>2</sup> TCH – toneladas de cana por hectare da área a ser colhida.

<sup>3</sup> Fibra – qualidade do bagaço da cana-de-açúcar após a moagem.

processamento e geração de soluções a partir da infinidade de dados disponíveis nas usinas; e sistemas de suporte à decisão, como interface final com o usuário, de uma interação amigável e completa.

Esta monografia está organizada em seis capítulos. Eles são dispostos da seguinte forma:

No Capítulo 2, descrevem-se os sistemas de informação e apresentam-se algumas outras definições essenciais nesta área. Discute-se a importância da informação nos dias atuais e seu valor dentro de uma empresa. Classificam-se os tipos de sistema enfatizando-se os sistemas de suporte à decisão, estes de interesse central para este trabalho.

No Capítulo 3, apresentam-se os fundamentos de redes neurais artificiais (RNA). Como foram concebidas, seus propósitos, e algumas aplicações. Detalham-se como a arquitetura de uma RNA é definida, o trato das topologias e dos parâmetros ajustáveis. Ressaltam-se os processos de aprendizagem, vitais para uma RNA. Dos vários tipos de rede, enfatiza-se as redes *multi-layer perceptron* (MLP) usadas diretamente neste projeto. Por fim, comenta-se detalhadamente as fases de parametrização e de treinamento de uma rede MLP, apresentando os desafios e potencialidades a serem observados.

No Capítulo 4, o sistema desenvolvido é apresentado. São apresentadas suas características, as funcionalidades e a arquitetura usada. Depois é descrito como o suporte à decisão é construído a ser baseado em tecnologia neural. Finalmente, as ferramentas e tecnologias empregadas durante todo o desenvolvimento do sistema são apresentadas. Diversos diagramas e figuras<sup>4</sup> ilustram o capítulo auxiliando o leitor quanto a detalhes de arquitetura e projeto.

No Capítulo 5, são apresentados os experimentos e resultados obtidos a partir da implementação da ferramenta. São mostradas como foram procedidas as validações necessárias ao longo deste projeto.

Finalmente, no Capítulo 6 é feita a conclusão desta monografia com a apresentação das contribuições, algumas discussões e a proposta de trabalhos futuros.

---

<sup>4</sup> A arte utilizada nas telas do sistema foi elaborada pelo designer Victor Ximenes.

## Capítulo 2

# Sistemas de Informação

No início da computação eletrônica as aplicações eram isoladas e independentes, sem uma real preocupação com duplicação de dados e racionalização do processamento. A partir de década de 60, informação e tecnologia passaram a andar juntas. O avanço tecnológico evidenciou a necessidade de uniformidade e integração das informações como forma de organização e alcance dos objetivos - era o nascimento dos sistemas de informação [Rezende03].

Os sistemas de informação gerencial (SIG, do inglês *management information system*, *MIS*) surgiram como produtos de informação que suportavam as decisões estruturadas. Destinados ao controle operacional, exibiam, basicamente, relatórios pré-definidos.

No início da década de 70, começaram a surgir outros tipos de sistemas, os primeiros sistemas de suporte à decisão (SSD). Diferente dos SIGs os SSDs eram cercados por uma economia instável, pelo desejo de melhores informações e por uma competição cada vez mais acirrada. Assim muitas empresas e centros de pesquisas passaram a desenvolver SSDs [Sprague91].

No entanto, o processo decisório continua, ainda hoje, sendo um dos mais desafiadores papéis de um gerente.

Iniciamos este capítulo oferecendo algumas definições básicas, importantes para este trabalho. São elas:

- a. Dados – fatos brutos que representam eventos ocorrendo na organização ou no ambiente físico antes de serem organizados e arrumados numa forma que as pessoas possam entender e usar [Laudon00].
- b. Informação – dados que foram modelados de forma que façam sentido e sejam úteis para os seres humanos [Laudon00].
- c. Tecnologia da Informação (TI) – é o termo que serve para designar o conjunto de recursos tecnológicos e computacionais para geração e uso da informação [Rezende00]. Também é conhecido como tecnologia da informação e comunicações (TIC).
- d. Sistema – do grego *sýstema*, reunião, grupo – é uma combinação de partes coordenadas entre si e que concorrem para um resultado ou para formarem um conjunto [Priberam06]. Convencionou-se que um sistema é, principalmente, composto por três componentes: entrada, processamento e saída.
- e. Sistema de informação (SI) – pode ser definido tecnicamente como um conjunto de componentes interrelacionados que coletam, processam, e distribuem informação para apoiar a tomada de decisão e o controle em uma organização. Adicionalmente a tomada de decisão, coordenação e controle, sistemas de informação podem também ajudar

gerentes e funcionários analisarem problemas, visualizarem assuntos complexos e criarem novos produtos [Laudon00].

Um SI, por sua vez, utiliza recursos de TI para realizar as operações de entrada e saída, e provê o processamento necessário para a transformação dos dados em produtos de informação. Outros componentes de relevância dentro de um SI são:

- a) o *feedback* que é os resultados das interações e informações de desempenho que auxiliam na avaliação e correção da fase de entrada;
- b) o controle, responsável pela captação e manutenção do *feedback*, possibilitando avaliações ou ajustes dos objetivos a serem alcançados.

Nas seções seguintes detalharemos a importância da informação dentro de uma empresa, bem como descreveremos os tipos de SI, suas peculiaridades, enfatizando os SSD que são o propósito deste trabalho. Este capítulo encerra com um resumo das características dos SIG e SSD, a serem incorporados na solução computacional produzida neste trabalho, bem como os aspectos web relevantes nesta implementação.

## 2.1 Valor da Informação - Decisão

Heráclito (540 – 480 AC), filósofo grego, autor da famosa frase - “*Ninguém pode tomar banho duas vezes nas águas do mesmo rio, porque o rio está em constante mudança, mas mais ainda, porque também o está quem nele mergulha.*” – defendia que a característica mais permanente da natureza era a constante mudança [Guimarães89]. Tudo muda, o tempo todo. E assim sendo, os dados de hoje, amanhã, serão ultrapassados.

No contexto empresarial, a revolução tecnológica gerou uma mudança de valores em todo mundo. Os fatores determinantes de sucesso deixaram de ser capital e trabalho e assumiram a forma de informação e conhecimento, e estes, são a chave da produtividade e da competitividade [Braga96]. As informações surgem internamente ou de sistemas externos e com o advento da Internet, a disseminação é quase que uma regra. Com isso, surge o problema da gerência do conhecimento. O que fazer para estar sempre atualizado, compartilhar essas informações internamente e, principalmente, isolá-las dos concorrentes são alguns dos desafios que os SIs tentam contornar.

As estratégias necessárias que são capazes de reduzir os custos, criar inovações e serem diferenciais diante das oponentes no mercado são frutos da informação. Apesar de ser um bem intangível, a informação ocupa o primeiro lugar na lista de ativos dentro de uma empresa.

Ao longo dos anos, o volume e a velocidade com que novos dados são gerados é diretamente proporcional à importância e o risco das decisões tomadas por gestores em organizações empresariais. Os SIs têm ajudado gerentes a comunicarem e distribuírem informação mas pouco contribuíram no processo de tomada de decisão [Laudon00].

Este processo consiste de diferentes atividades realizadas em momentos distintos. O agente decisor tem que perceber e entender problemas. Uma vez percebido, soluções devem ser projetadas; uma vez projetadas, escolhas têm que serem feitas sobre uma solução particular; finalmente, a solução tem que ser implementada. Esquemáticamente existem quatro diferentes estágios no processo de decisão [Simon60] (ver Fig. 2):

- I. Inteligência;
- II. Projeto;
- III. Escolha, e;
- IV. Implementação.

*Inteligência*, o primeiro dos estágios, consiste em coletar informações e identificar os problemas que ocorrem na organização. Indica o porquê, onde e com que efeitos as mudanças

ocorrem. No segundo estágio, *projeto*, o indivíduo planeja possíveis soluções para o problema. Esta atividade pode requerer mais inteligência pelo fato do decisor definir se uma solução é apropriada ou não. O estágio de *escolha* consiste em eleger entre as alternativas. Aqui o gerente pode usar ferramentas de informação que calculem e rastreiem as conseqüências, custos e oportunidades providas por cada uma das alternativas propostas no segundo estágio. Finalmente, na *implementação*, o indivíduo coloca a decisão em prática e reporta o progresso da solução [Laudon00].

Geralmente, chega-se a decisões depois de uma série de iterações e avaliações em cada estágio do processo. O agente decisor pode voltar em um ou mais estágios antes de completar o processo. Assim, a tomada de decisão necessita do maior número de informações possíveis para que venha acarretar em sucesso. Decisões puramente empíricas são fadadas ao fracasso.

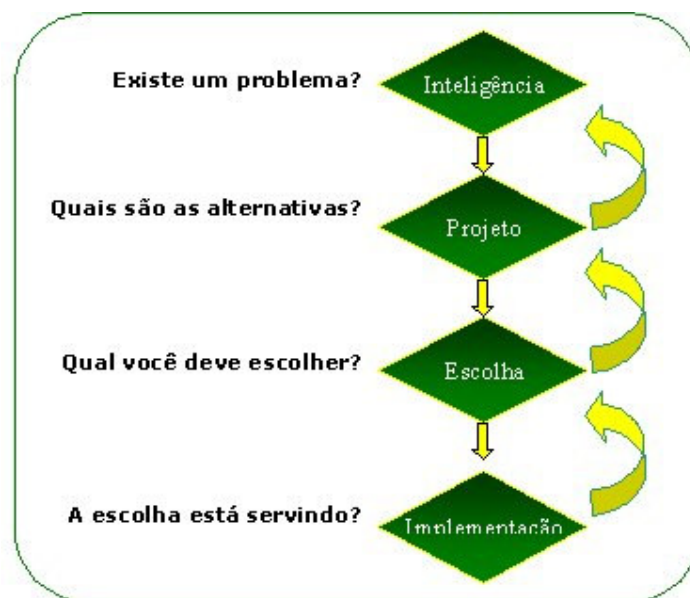


Fig. 2 - Processo genérico de tomada de decisão (adaptado de [Laudon00])

## 2.2 Tipos de SI

Os SIs podem ser classificados de várias formas. Não existe uma regra que defina se um sistema é de um tipo ou de outro, até porque um SI pode ser rotulado em mais de uma categoria. Para facilitar a classificação, ela pode ser feita a partir de algumas perspectivas:

- Com relação ao seu escopo [Zaneti03];
- Através do tipo de decisão que apóiam e o nível hierárquico a que se destinam [Lucas97];
- Através das funções que desempenham [O'Brien01].

Com relação ao escopo, eles podem ser interorganizacionais ou não, caso envolvam mais de uma organização. Por exemplo, sistemas que envolvem clientes e fornecedores [Zaneti03].

Combinando os tipos de decisão e o nível hierárquico a que se destinam, os SIs ainda podem ser classificados de acordo com a Tabela 1. Nela podemos verificar alguns exemplos de SI (em todos estes níveis) para problemas estruturados, semi-estruturados e não estruturados. Problemas bem estruturados são repetitivos, rotineiros e têm um procedimento definitivo para tratá-los. Por outro lado, problemas não-estruturados não são rotineiros e o agente decisor deve prover julgamento, avaliação e visões dentro da definição da questão - não existe fórmula na



resolução destes problemas [Lucas97]. Pode ser visto a seguir (vide a Fig. 3) as informações necessárias em cada nível de decisão.

Tabela 1 – Classificação de SI por sua estrutura [Lucas97]

Classificação	Controle Operacional	Controle Gerencial	Planejamento Estratégico
Estruturada	Processamento de pedidos; contas a pagar	Orçamentos; relatórios pessoais	Localização de plantas; <i>mix</i> de modos de transporte
Semi-estruturada	Controle de estoque; planejamento da produção	Análise de variância	Introdução de novo produto
Não estruturada	Gerenciamento de caixa	Gerenciamento de pessoal	Planejamento para P&D



Fig. 3 – Tipo de decisão x Características da Informação

De acordo com a funcionalidade dentro das organizações existe outra classificação que é ilustrada na Fig. 4 [O'Brien01]. Podem-se dividir os SIs em dois grandes grupos: os de apoio às operações e os de apoio gerencial.

Os sistemas que apóiam às operações são responsáveis pelo dia-a-dia de uma empresa. Devem ser bastante precisos e rápidos. São eles que processam os dados gerados nas operações, processam as transações efetivamente, controlam os processos industriais, apóiam comunicações e colaboração, e sempre estão atualizando o banco de dados. Realizam atividades das mais simples às mais complexas e, geralmente, são vitais para o funcionamento da empresa. Estes SI dividem-se em três categorias: sistemas de processamento de transações (SPT) que têm como principal objetivo o registro acurado das operações e fatos relevantes das áreas de negócio. A ênfase nesses sistemas é com a validação dos dados, visando maior qualidade e depuração das

bases de dados; sistemas de controle de processos (SCP) que monitoram e controlam processos industriais; e sistemas colaborativos que apóiam equipes e grupos de trabalho, bem como a comunicações e colaboração entre e nas empresas.

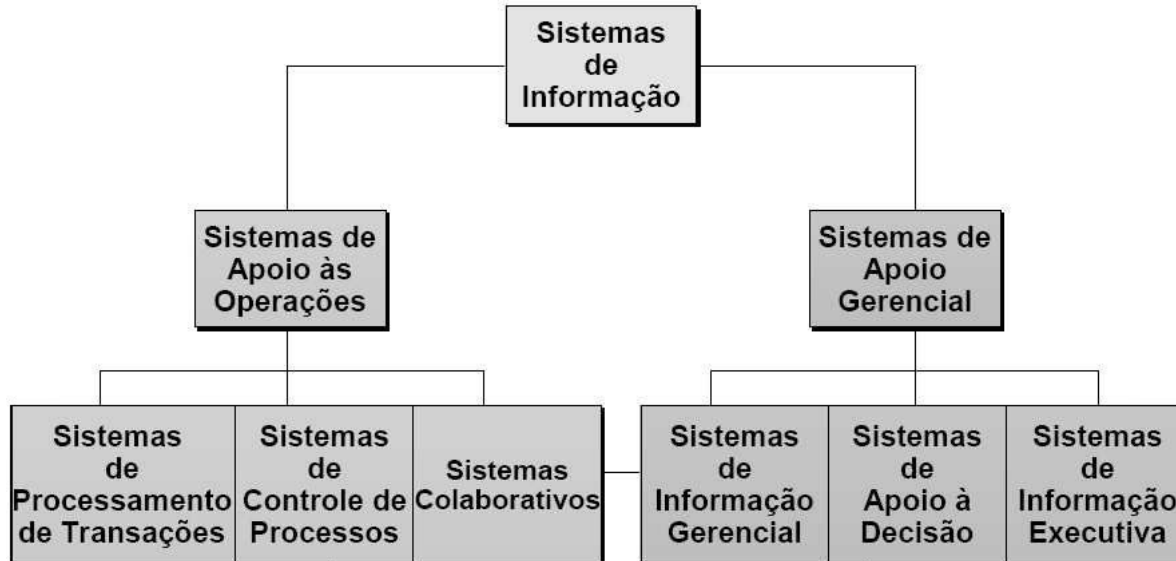


Fig. 4 - Classificação de SI de proposta por O'Brien [O'Brien01]

Sistemas de informação processam dados de maneira a produzir para seus usuários informações relevantes aos problemas para os quais foram propostos. Estes quando se propõem a interagir de forma a proceder escolhas e posicionamentos ligados ao fluxo de negócio, são considerados sistemas de apoio gerencial ou sistemas de decisão. Notadamente, sistemas de decisão são identificados numa posição superior nas hierarquias empresariais, em um âmbito de escolhas mais críticas.

Utilizam os SIs sob eles (os de apoio à operação) como fonte de dados e, por isso, também são considerados sistemas verticais. As informações provindas de diversos setores da empresa (um sistema para cada operação) capacitam os sistemas de decisão no apoio gerencial. Também podem ser subdivididos em três categorias:

- Sistemas de Informações Gerenciais (SIG, do inglês *management information systems*, MIS) - a contínua evolução e geração de dados exigiu uma nova classe de sistemas que estruturasse a informação que seria gerada. Suas principais características são: foco na informação direcionada a gerentes, informação estruturada e dividida por funcionalidades. Apesar de proverem auxílio em decisões gerenciais, estes são pouco significativos nesta função, já que informam decisões tomadas no passado. Desta maneira sua aplicabilidade real são os controles operacionais;
- Sistemas de Apoio à Decisão ou de suporte à decisão (SSD, do inglês *decision support systems*, DSS) que fornecem apoio a decisões semi-estruturadas e serão mais detalhados na seção seguinte (ver 2.3), pois como foi o tipo de sistema adotado neste projeto, é necessário um aprofundamento teórico;
- Sistemas de Informações Executivas (SIE, do inglês *executive information systems*, EIS) – surgem, aparentemente, de um refinamento dos SIGs. Os SIE propõem mais que o manuseio de informações operacionais. Com o foco na conjuntura e indicadores estratégicos, atende os executivos que se encontram no topo da hierarquia empresarial, tendo como objetivo um planejamento estratégico em decisões críticas. Agregam dados internos e externos, junto com indicadores de desempenho e fatores críticos de sucesso.

Podemos ainda citar outras categorias de SI. Sistemas especialistas (SE, do inglês *expert systems*, ES) que são baseados em inteligência artificial e fornecem conselho especializado em tarefas, funcionando como tutores específicos em determinados domínios. Sistemas de administração do conhecimento (do inglês *knowledge work systems*, KWS) que auxiliam a gerência do conhecimento apoiando a criação, a organização e a disseminação do conhecimento empresarial.

## 2.3 Sistemas de Suporte à Decisão

Sistemas de suporte à decisão são desenvolvidos para servir no nível gerencial de uma organização e são encontrados nos estágios de projeto e de escolha do processo decisório (vide Fig. 2). SSDs ajudam os gerentes a tomar decisões que são semi-estruturadas, únicas ou que mudam constantemente, e não são facilmente especificadas a priori. SSDs têm que ser rápidos o suficiente para executarem diversas vezes durante um dia e devem atender às mudanças de condições dos problemas. Embora os SSDs usem informação interna dos SPTs e dos SIGs, eles podem buscar informações de fontes externas, tal como bolsas de valores, preços de concorrentes ou como neste trabalho, preditos e gerados por uma rede neural.

Por definição, SSDs têm mais poder analíticos que outros sistemas de informação; eles são explicitamente construídos contemplando uma variedade de modelos para análise de dados ou eles condensam grandes quantidades de dados de forma que podem ser analisados pelos tomadores de decisão [Laudon00]. São projetados para que os usuários os usem diretamente, desta forma precisam dispor de uma interface amigável. Também devem ser interativos; os usuários podem mudar as suposições, fazer novas perguntas e incluir novos dados.

Os SSDs surgiram como uma especialização dos SIGs, na tentativa de apoio ao processo de gestão empresarial, bem como na resolução de problemas semi ou não estruturados. É a combinação de recursos intelectuais com capacidades computacionais apoiando as tomadas de decisões.

A Fig. 5 mostra os componentes usuais de um SSD e como eles se comunicam entre si. Iremos analisar os componentes que incluem um banco de dados usado para consultas e análises, um sistema de software com modelos, *datamining*, e outras ferramentas analíticas e uma interface para o usuário que serão descritos nas subseções a seguir.

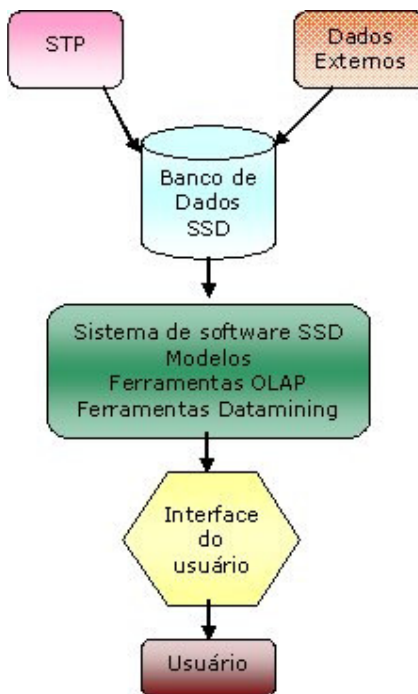


Fig. 5 – Componentes de um SSD (adaptado de [Laudon00])

### 2.3.1 Componentes de um SSD – Banco de Dados

O banco de dados SSD é uma coleção de dados correntes ou históricos de vários outros sistemas da empresa. Pode ser pequeno, uma cópia de parte dos dados corporativos combinados com alguns dados externos e residir na própria máquina onde se encontra o SSD. Pode ser um *data warehouse* massivo, constantemente atualizado pelos principais SPTs da organização. Ou dados gerados de sistemas distribuídos e web. Estes bancos geralmente são cópias (visões) dos bancos de dados de produção, assegurando que os SSDs não interferem nas atividades críticas da empresa [Laudon00].

### 2.3.2 Componentes de um SSD – Sistema de Software

O sistema de software SSD contém ferramentas utilizadas para a análise dos dados. Podem conter ferramentas de OLAP (*Online Analytical Processing*), ferramentas de *datamining*, ou uma coleção de modelos matemáticos e analíticos que são facilmente acessados pelos usuários dos SSDs.

Um modelo é uma abstração que ilustra componentes e relacionamentos de um fenômeno ou problema [Laudon00]. Modelos podem ser físicos, matemáticos ou verbais, e cada SSD disponibiliza diferentes coleções destes modelos, de acordo com o propósito a que foram construídos. Um modelo bastante comum é o estatístico. Com ele os usuários têm acesso a funções como: média, desvio padrão, moda, gráficos, entre outros, que auxiliam em previsões, no estabelecimento de relações e outros fatores atrelados aos dados. Modelos de otimização facilitam a maximização ou minimização de funções de erros ou variáveis como custo ou tempo.

A partir destes modelos podem-se definir os tipos de sistemas de suporte à decisão. Cada tipo associado aos propósitos e complexidade do sistema. Destacam-se três deles:

- a. Simulações - a categoria mais simples e comumente encontrada é dos sistemas baseados numa análise de sensibilidade, os quais possuem modelos que perguntam questões “o que

– se” (“*what – if*”) repetidamente para determinar o impacto de mudanças em um ou mais fatores de saída. Interação com o usuário na forma de modificação de variáveis possibilitando a visualização dos efeitos nas outras variáveis do sistema. Ex.: num sistema agrícola - se eu colher minha plantação amanhã, qual será minha produtividade;

- b. Atingir Metas - um pouco mais complexo, já que a partir da definição de uma meta o sistema define possíveis modificações para a realização deste objetivo. Ex.: num sistema agrícola – preciso de 10 mil toneladas de cana (entrada do usuário) e a resposta seria: corte os lotes ‘a’, ‘b’, ‘d’ e ‘j’;
- c. Otimizações - que busca metas realizando uma otimização das variáveis do sistema. Estes são os mais avançados e complexos sistemas de suporte à decisão. Ex.: num sistema agrícola – preciso de 10 mil toneladas de cana com um melhor aproveitamento de sacarose possível (entrada do usuário) e a resposta seria: corte os lotes ‘k’, ‘b’, ‘d’, ‘h’, ‘o’ e ‘m’.

O sistema proposto pode ser classificado nestas três categorias já que realiza simulações, atinge metas e otimizações de variáveis, como citado nos exemplos acima. Em suma, os SSDs necessitam de informação interativa para fazer simulações de acordo com a necessidade de cada gestor, aplicando os modelos analíticos e/ou banco de dados especializados, usando relações matemáticas entre as variáveis no apoio a soluções semi-estruturadas ou não-estruturadas.

### **2.3.3 Componentes de um SSD – Interface**

A interface de SSD com o usuário permite fácil interação entre os utilizadores do sistema e as ferramentas de software. Requer uma interface gráfica fácil de usar, flexível e capaz de suportar diálogos entre o usuário e o SSD. Vale ressaltar que estes usuários, na sua maioria, são gerentes ou executivos corporativos, pessoas com estilo de trabalho bem definido e preferências individuais, os quais possivelmente não possuem experiência com o computador ou não dispõem de tempo para aprender a manusear ferramentas complexas. Com isso, um SSD de sucesso demanda de um bom planejamento e projeto com criações de protótipos e validações com o usuário [Laudon00].

## **2.4 Características desejáveis de SSD para o problema alvo deste trabalho**

Após toda a teoria de sistemas de informação e na tentativa de solucionar o difícil problema da colheita da cana-de-açúcar, podemos destacar as principais características relevantes a este domínio que o sistema de suporte à decisão proposto precisa apresentar:

- Ajudar na tomada de decisão de problemas semi-estruturados;
- Ser rápidos;
- Ser flexíveis;
- Buscar informações externas;
- Apresentar interface com o usuário bastante amigável e intuitiva;
- Capacidade de simular cenários, atingir metas e otimizar variáveis.

## 2.5 Aspectos Web relevantes ao sistema

A necessidade de disponibilidade e acesso dos sistemas de informação criou uma nova categoria de SI – os sistemas Web. No entanto, a *World Wide Web* (WWW) percorreu um longo caminho desde quando os primeiros bytes foram enviados pela Internet, em 1989. Quando os físicos do laboratório CERN [CERN05] propuseram a idéia de compartilhar informações sobre pesquisas entre os cientistas, usando documentos hipertextos, eles não tinham noção de quanto a Web cresceria ou como se tornaria tão essencial no dia-a-dia de um mundo completamente globalizado. A Web é agora parte do nosso cotidiano [Cavaness02].

Demorou pouco até que os benefícios do uso da Web se tornassem claros para os outros fora do CERN e seu uso fosse expandido, levando à consolidação de uma grande rede mundial, a Internet. No início, a Web foi desenvolvida para distribuição de documentos estáticos, mas foi uma progressão natural querer a geração de documentos com conteúdos dinâmicos. O *Common Gateway Interface* (CGI) foi criado para tornar isto possível, tornando-se um padrão que permitia que *web servers* interagissem com aplicações externas, permitindo que os hipertextos fossem dinâmicos [CGI05]. Esta tecnologia revelou uma infinidade de possibilidades e, de fato, proporcionou a explosão da Internet dos anos 90 até o presente.

No entanto, à medida que as aplicações cresciam, CGI revelava sérias limitações e dificuldades. Entre as principais estão: performance, reusabilidade e manutenção. Na tentativa de minimizar estes problemas surge, em 1997, a tecnologia *Java Servlets* [Sun05]. Prosseguindo na escala evolutiva com *JavaServer Pages* (JSP) e as arquiteturas *Model-View-Controller* (MVC) modelo 1 e 2 [Cavaness02] [BluePrints05].

Apesar do padrão MVC não estar diretamente relacionado a aplicações Web, a separação de responsabilidades proposta tem-se mostrado bastante efetiva. A complexidade inserida na modelagem do problema permite, em última instância, um desenvolvimento modularizado, uma manutenção facilitada e o reuso de componentes em novas aplicações.

Todas estas facilidades difundem o uso de sistemas web para as mais variadas áreas de aplicação. Na área agrícola, aplicações web podem facilmente ser aceitas por seus usuários devido à similaridade de navegação com a Internet e o fato da ferramenta ser intrinsecamente distribuída, une instalações físicas espalhadas pela região.

## Capítulo 3

# Redes Neurais Artificiais

Redes neurais artificiais (RNA) foram inspiradas, no funcionamento sistema nervoso. Ou seja, tenta-se reproduzir no computador o modo como o cérebro processa informações. O objetivo com isto é incluir todas as características necessárias para a resolução de problemas complexos e com alto volume de informações existentes no mundo real. Essas qualidades advindas da inspiração biológica, acrescidas à computação tradicional, transformam o computador numa plataforma que se adapta facilmente aos problemas, com alto grau de paralelismo, generalização e associatividade. E, a mais importante de todas, aprender usando acertos e erros passados.

Mesmo cérebros primitivos e pequenos são capazes de realizar atividades complexas em fração de segundos, por exemplo, a visão de um pássaro ou o olfato de um cão. Estas mesmas atividades levariam horas nos atuais computadores digitais utilizando computação tradicional. Mas como isso é possível?

O conhecimento adquirido pelo cérebro é feito através da vivência, da experimentação. Toda essa experiência vai sendo acumulada ao longo do tempo por estruturas celulares chamadas neurônios (ver Fig. 6) em forma de ligações sinápticas. Esses podem acumular conhecimento sob forma de pesos sinápticos e o ajuste desses valores é conhecido como processo de aprendizagem.

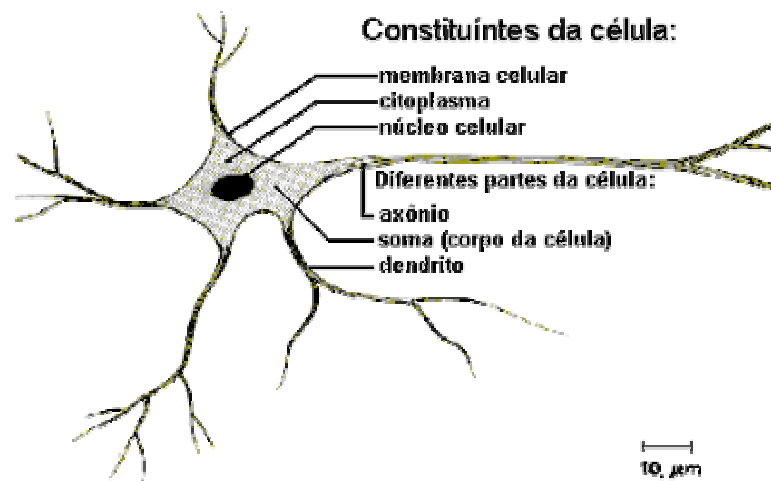


Fig. 6 – Neurônio: célula do sistema nervoso [Tatibana06]

O modelo matemático de uma RNA tenta emular a atividade cerebral, e desta forma, também possui abstrações de neurônios que processam – somatório - entradas ( $x^1, x^2, \dots, x^p$ )

multiplicadas pelos pesos sinápticos ( $w^1, w^2, \dots, w^p$ ). Na seqüência este valor passa por uma função de ativação antes que a resposta ( $y$ ) seja gerada (ver Fig. 7).

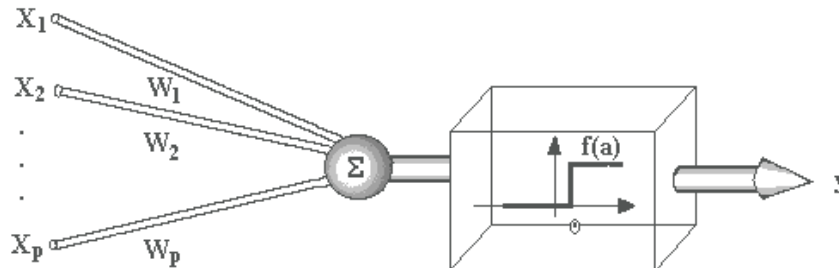


Fig. 7 – Neurônio artificial projetado por McCulloch e Pitts [MCP43]

### 3.1 Definição

A organização destes neurônios artificiais conectados por ligações sinápticas em forma de funções matemáticas permite ao computador a criação de uma estrutura primitiva capaz de armazenar informações e gerar novos conhecimentos.

Haykin define uma rede neural vista como uma máquina adaptativa [Haykin94]:

*“Uma rede neural é um processador maciçamente paralelamente distribuído constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso. Ela se assemelha ao cérebro em dois aspectos:*

1. *O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem.*
2. *Forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.”* [Haykin94, pág. 28].

A unidade básica de processamento de uma RNA é o neurônio artificial que é composto pelos seguintes elementos:

- Um conjunto de sinapses ou elos de conexão, representando as entradas do sistema com pesos próprios.
- Um somador, que processa as entradas ponderadas com os respectivos pesos realizando uma combinação linear;
- E, uma função de ativação ou função restritiva, que limita a amplitude de saída de um neurônio. Existem vários tipos de função de ativação, os mais conhecidos são: função de limiar, função linear por partes e função sigmóide (logística ou tangente hiperbólica).

O restante desse capítulo é organizado da seguinte forma. Na seção 3.2 são abordados os tipos de arquitetura de uma RNA. A 3.3 descreve os processos de aprendizagem mais usuais. E por fim, na seção 3.4, comenta-se em detalhe as redes *Multi-Layer Perceptron* (MLP) [Rumelhart86]. Muito embora este não seja o único tipo de RNA, nos concentraremos nele, pois além de ser o mais usado, foi o tipo empregado neste projeto.



## 3.2 Arquiteturas de Rede

Antes de serem abordados os paradigmas de aprendizado, deve-se analisar as estruturas mais usuais de uma rede neural, já que os dois conceitos estão intimamente ligados. Chamamos de arquitetura de rede neural a estruturação dos neurônios. Existem três classes mais usuais de arquiteturas:

1. Redes alimentadas adiante com camada única, a Fig. 8 ilustra bem este tipo de rede. Como só possui uma camada (a camada de entrada não é contabilizada já que não ocorre processamento computacional), está restrita a resolução de problemas linearmente separáveis<sup>5</sup> [Cybenko89];

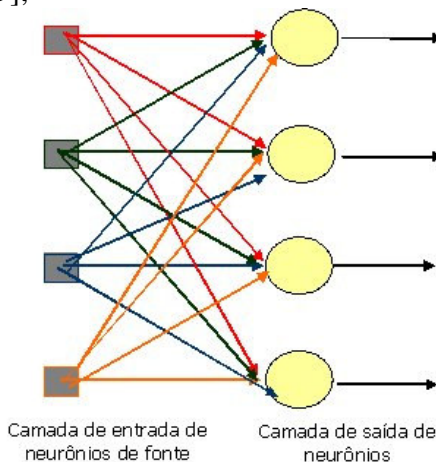


Fig. 8 – Rede alimentada adiante ou acíclica com uma única camada de neurônios

2. Redes alimentadas diretamente com múltiplas camadas – a diferença entre a classe anterior é que estas possuem uma ou mais camadas escondidas (ocultas). Esse acréscimo no número de camadas eleva bastante o poder computacional das RNAs, possibilitando a computação de problemas complexos. Assim, estas redes conseguem resolver qualquer função matemática [Cybenko89]. Observando a Fig. 9, que ilustra a classe, percebe-se também que esta é uma rede dita totalmente conectada, pois todos os neurônios de uma camada  $x$  estão ligados a todos os neurônios da camada  $x+1$ . Se isto não fosse verdade, a rede neural seria chamada de parcialmente conectada;

<sup>5</sup> Problemas que podem ser separados por uma reta em um hiperplano; em 1969 na monografia intitulada Perceptrons por Marvin Minsky e por Seymour Papert mostrou-se que era impossível para uma rede de camada única aprender uma função de XOR

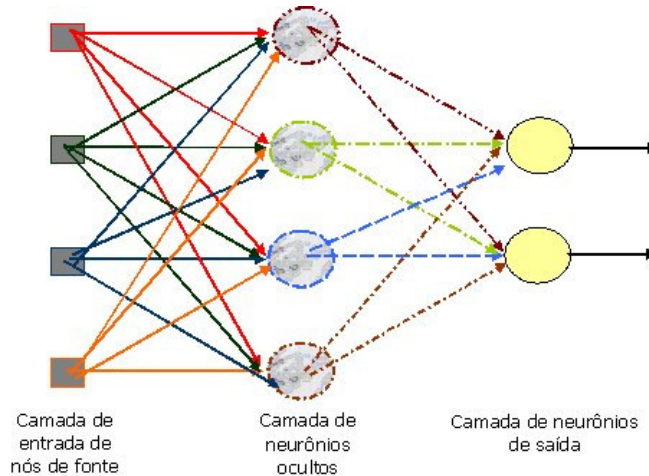


Fig. 9 – Rede alimentada adiante ou acíclica totalmente conectada com uma camada oculta e uma camada de saída

3. Redes recorrentes – essa classe difere das outras duas pelo fato de conter pelo menos um laço de realimentação. As redes recorrentes podem ou não ter neurônios ocultos, mas a presença de realimentação tem um impacto direto no processo de aprendizagem e no desempenho da rede. A Fig. 10 mostra um exemplo de redes recorrentes.

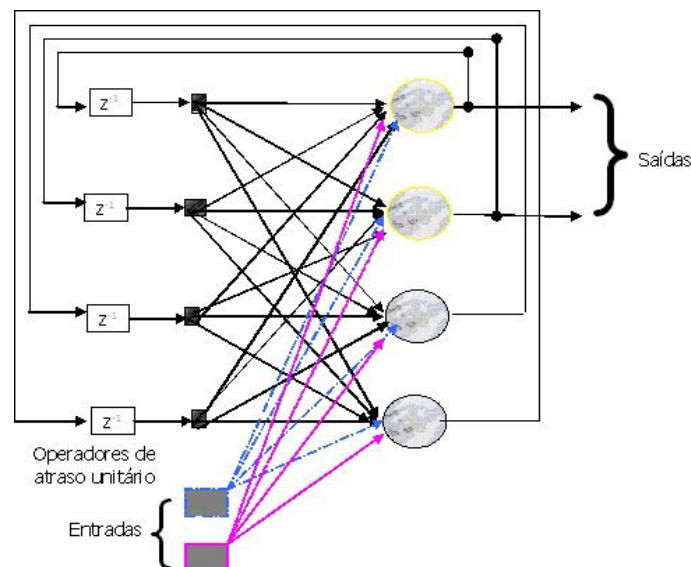


Fig. 10 – Rede recorrente com neurônios ocultos

### 3.3 Processos de Aprendizagem

Capacidade de aprendizagem é uma das mais importantes características de uma RNA. Ela é responsável pela adaptação, dinamicidade e aplicabilidade das redes neurais aos problemas do mundo real. Sem ela, a rede não aprenderia aspectos relevantes do ambiente e tão pouco seria capaz de melhorar seu desempenho ao longo do tempo.

Como o termo aprendizagem é utilizado em várias áreas, com diversas interpretações, usaremos a seguinte definição:

*“Aprendizagem é um processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um processo de estimulação pelo ambiente no qual está inserida. O tipo de aprendizagem é determinado pela maneira como a modificação dos parâmetros ocorre.”* [Haykin94, pág. 75].

Existem algumas regras básicas de aprendizagem:

- a) aprendizagem por correção de erro;
- b) aprendizagem baseada em memória;
- c) aprendizagem hebbiana;
- d) aprendizagem competitiva;
- e) aprendizagem de Boltzmann [Haykin94], dentre outras.

Basicamente, a aprendizagem de correção de erro se fundamenta em filtragem ótima. A aprendizagem baseada em memória, como o nome sugere, memoriza os dados de treinamento e conseqüentemente não é capaz de generalizar. As aprendizagens hebbiana e competitiva possuem inspirações neurobiológicas. E a, de Boltzmann é guiada por postulados da mecânica estatística.

Com relação aos paradigmas de aprendizagem, estes podem variar de acordo com o tipo de RNA e, principalmente, de acordo com a natureza dos dados a serem processados. Eles se dividem em duas grandes categorias:

- a. Supervisionado – ocorre quando se tem disponível um conjunto de entradas e saídas dos dados. A partir daí, a saída produzida pela RNA é comparada com a saída desejada. Em seguida, é calculado o erro desta alimentação e, finalmente, é chamado um processo de reajuste dos pesos sinápticos de maneira a reduzir esse erro de saída. A vantagem é que a rede pode conseguir convergir (se estabilizar em relação aos pesos) mais rapidamente. Por outro lado, obter estes dados (que servem como seu “professor”) pode ser uma tarefa complicada dependendo do domínio do problema. Além do que a “qualidade” desse conjunto de saídas desejadas pode impactar na capacidade de generalização (obter uma resposta correta a partir de um padrão não apresentado durante o treinamento) da rede. Entende-se por qualidade um conjunto que seja representativo sobre o problema.
- b. Não supervisionado – nesta segunda categoria não há presença de um “professor” que indique a saída desejada e possibilite o cálculo do erro. A RNA tem que capturar do ambiente as informações necessárias para que os ajustes dos pesos levem à solução. Este tipo de aprendizado é subdividido em dois: aprendizado por reforço, onde existe a presença de um *crítico* que serve como balizador do treinamento. Apesar de não informar o erro precisamente, pode indicar um caminho a ser seguido; e aprendizado não-supervisionado, onde são dadas condições para realizar uma medida independente da tarefa [Haykin94]. Pode-se utilizar a regra de aprendizagem competitiva na implementação do aprendizado não-supervisionado. Os neurônios de uma camada escondida competem entre si na tentativa de solucionar o problema. As principais redes baseadas neste paradigma são: os mapas SOM (*self organized maps*) de Kohonen e os modelos ART de Grossberg & Carpenter [Buarque98].

Finalizamos acrescentando que não existe paradigma ideal, tudo depende da natureza dos dados e da abordagem escolhida na arquitetura e classe da rede neural.

## 3.4 Multi-Layer Perceptron

O euforismo, a cerca de redes neurais artificiais, que permeava a comunidade acadêmica na década de 50 e 60, após a publicação de Rosenblatt sobre aplicações de computação com aprendizado [Rosenblatt61], era grande. Ela comprovava o teorema da convergência do *Perceptron* e prometia uma revolução nos sistemas computacionais. No entanto, no final da década de 60, Minsky *et* Pappert demonstraram que apesar da convergência, o *Perceptron* não era capaz de resolver problemas relativamente simples. Ele só tratava de problemas linearmente separáveis. Por isso muitos passaram a não apostar mais nas RNAs como uma ferramenta efetiva para resolver problemas reais.

Nos anos seguintes poucas pesquisas foram realizadas. O *Perceptron* só teria poder se fosse encadeado em forma de camadas e com isso a limitação estava no aprendizado. Não existia algoritmo para isto. Felizmente, na década de 80, Rumelhart [Rumelhart86] descreve um algoritmo capaz de “ensinar” as redes perceptron de múltiplas camadas (do inglês *Multi-Layer Perceptron*, MLP). Na verdade, desde as décadas de 60 e 70 esta solução já estava concebida, mas pouco divulgada [Werbos74].

Esse algoritmo foi chamado de *Error Backpropagation* (retropropagação do erro) pois seu princípio era ir propagando o erro desde a camada saída até a camada de entrada compensando os pesos sinápticos de cada neurônio de acordo com sua relevância no processamento realizado [Jain96]. A partir de então, o entusiasmo retornou e as pesquisas estão sempre aumentando na área.

Comprovado o poder computacional das redes, como MLP uma técnica revolucionária de resolver problemas complexos, passou-se a utilizar MLP nas mais diversas áreas de conhecimento: [Fu-Liu04] na medicina, [Sinha98] no reconhecimento de imagens, [Park04] na agricultura e em particular, no problema da colheita da cana-de-açúcar [Buarque98].

### 3.4.1 Parametrização

Apesar das constantes pesquisas, MLPs revelam certa complexidade em seu uso. A quantidade de parâmetros configuráveis exige do usuário uma certa experiência, provavelmente adquirida com muita experimentação. Não existem muitos estudos que comprovem a definição precisa destas variáveis.

Dentre os desafios a contornar na parametrização das RNA destacamos: o domínio do problema e características das entradas; a própria definição da topologia da rede; o número de camadas escondidas; o número de neurônios nessas camadas; e o número de conexões entre essas camadas. Outro procedimento que influencia o treinamento de uma RNA é a inicialização dos pesos da rede. Geralmente são atribuídos valores aleatórios entre -1 e 1. Simulações procedidas por redes inicializadas com valores distintos produzem saídas distintas também.

Com a rede configurada e os dados preparados para o processamento (divididos entre entradas e saídas desejadas), inicia-se a fase de treinamento.

### 3.4.2 Treinamento

Esta é a fase mais importante e mais prolongada no uso de MLPs. Pois, treinar uma rede neural é deixá-la apta para processar e resolver o problema para o qual foi planejada. Nas redes MLP o treinamento em si é composto por duas etapas: propagação dos dados de entrada (*forward*) e retropropagação dos erros de saída (*backward*). No entanto a fase como um todo ressalta outras observações.

Primeiramente, deve-se decidir quais padrões farão parte do conjunto de treinamento e quais serão para testes. Esse é um *trade-off* importante, pois quanto maior for o conjunto de treinamento, mais chances sua rede neural terá de generalizar bem os aspectos do problema. Por outro lado, se os padrões não conseguem contemplar um conjunto de testes, não há como validar o treinamento e, conseqüentemente, não se mede a precisão da rede MLP treinada. Com isso, cabe bom senso na divisão dos dados de treinamento e teste.

Em seguida, deve-se determinar alguns parâmetros de treinamento:

- Tipo de treinamento – pode ser *on-line* ou *batch* (por lote). No aprendizado *on-line*, a fase *backward* ocorre a cada apresentação de um vetor de entrada (um padrão). Por outro lado, no treinamento por lote, os pesos sinápticos são ajustados apenas a cada nova época<sup>6</sup>. Ou seja, são guardados os erros gerados por cada padrão e no final de um ciclo é calculada a média do erro. Só então são procedidos os ajustes necessários para a convergência da rede. Usualmente o treinamento *batch* é mais aplicado, pois além de possuir um desempenho melhor, é mais estável a padrões anormais;
- Número de ciclos ou épocas – define a quantidade de vezes que o conjunto de padrões (uma época) será apresentado à rede neural durante o treinamento. Este número está intimamente ligado à taxa de aprendizado. Dois problemas atrelados à escolha do parâmetro são *underfitting* e *overfitting*. O primeiro cuja tradução aproximada é “sub-ajuste” ocorre quando a rede não é treinada suficientemente para se tornar capaz de produzir resultados satisfatórios, convergir; e o segundo cuja tradução aproximada é “super-ajuste” ocorre quando a rede é demasiadamente treinada e passa a decorar os dados apresentados. Com isso, incorpora ao modelo ruídos e / ou inconsistências atreladas ao conjunto de treinamento;
- Taxa de aprendizado ou  $\eta$  (*eta*) – define a velocidade com que os erros tentarão ser corrigidos, como pode ser visto na Fig. 11. Uma taxa de aprendizado alta representa um grande ajuste nos pesos sinápticos, ou seja, rapidamente a função caminhará na direção do menor erro. No entanto, devido a esse grande ajuste, é possível que a função nunca encontre um erro aceitável e convirja. A Fig. 12 ilustra esse fenômeno. Se relacionarmos a taxa de aprendizado com o número de ciclos em um treinamento, pode-se dizer que estes são grandezas inversamente proporcionais, pois se for usado uma taxa de aprendizado baixa, a rede neural convergirá lentamente na direção do erro mínimo e por isso necessitará de um número maior de ciclos - ou vice-versa;

---

<sup>6</sup> Uma época é todo o conjunto de dados de treinamento, ou seja, uma época são todos os padrões de dados de entrada.

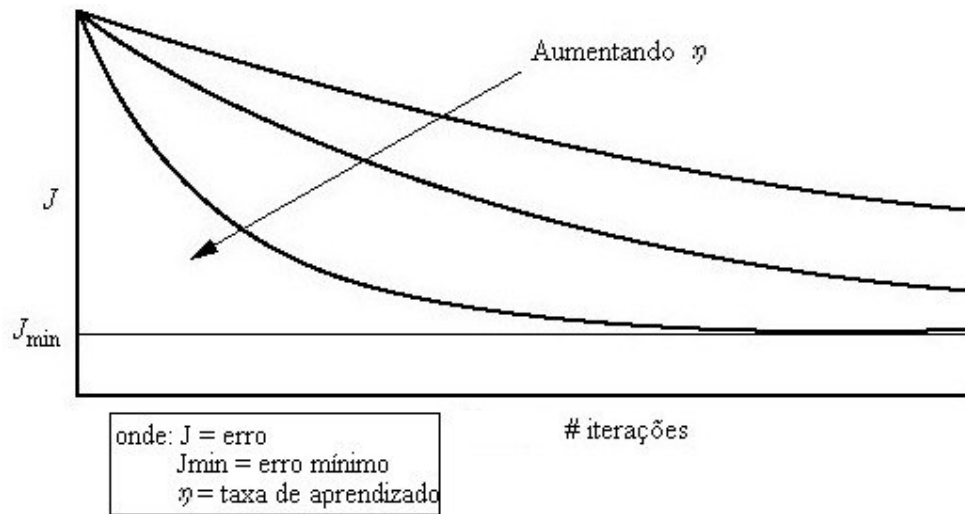


Fig. 11 – Curvas de aprendizado

A Fig. 12 mostra claramente a variação da taxa de aprendizado. Nela podemos observar que quando é escolhida uma taxa pequena, lentamente a rede caminhará na direção do erro mínimo. Quando é escolhida uma taxa um pouco maior a rede consegue encontrar um erro aceitável, mais rapidamente, mas sempre alternando em torno do erro mínimo. No entanto, se a taxa de aprendizado for muito grande, a rede neural pode nunca conseguir convergir.

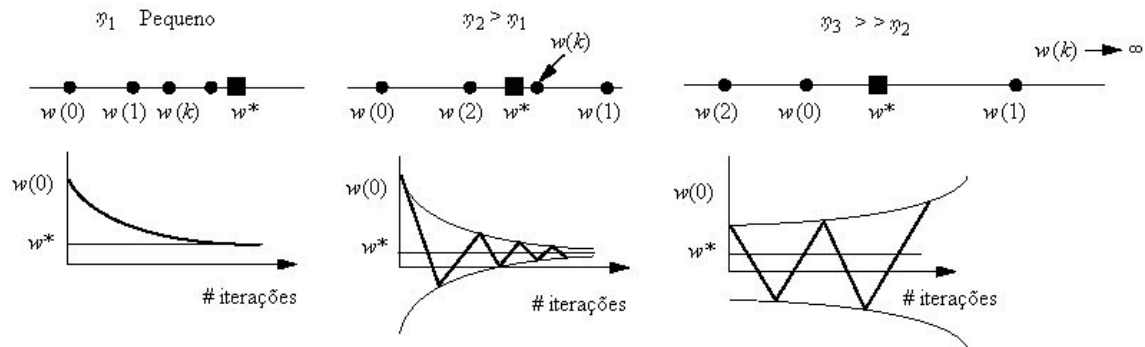


Fig. 12 – Relação entre taxa de aprendizado e convergência a partir do ajuste de pesos

Durante o processo de propagação dos dados, a rede processa as entradas ponderando com os pesos sinápticos a cada camada, gerando uma resposta para o vetor de entradas apresentado. De posse dessa saída gerada, compara-se com a saída desejada apresentada pelo padrão. Com isso, o erro pode ser calculado (ver Equação 1).

$$\bar{E} = \left( \frac{1}{n} \cdot \sum_{i=1}^n (y_i - d_i)^2 \right) \cdot 100$$

onde: n = número de padrões  
y = saída calculada  
d = saída desejada  
E = erro médio

Equação 1 – Fórmula de cálculo do erro médio

A fase de retropropagação do erro é iniciada em seguida (a depender da escolha *on-line* / *batch*), com a chamada do algoritmo *backpropagation*. Este implementa a correção dos pesos sinápticos de todos os neurônios através da regra do gradiente descendente que computa a derivada parcial das saídas de cada neurônio em relação aos pesos sinápticos atuais. Este ciclo de entrada de dados e correção dos pesos é justamente o processo de aprendizagem, que mencionamos anteriormente.

Aparentemente finalizada, restam dois fatores cruciais no treinamento: a definição dos critérios de parada do treinamento e a escolha do critério de erro para medir seu desempenho.

Como dito anteriormente, se o número de ciclos for o único indicador de término de um treinamento, corre-se dois grande riscos. A rede neural sofrer, novamente, *underfitting* ou *overfitting*.

Existem algumas abordagens de critérios de parada. Um primeiro poderia simplesmente ignorar o número de ciclos e definir uma meta de erro, i.e. não termine o treinamento enquanto o erro for mais que 3%, por exemplo. De certo, esta também não é uma boa abordagem, pois impor uma meta fixa, pode levar a rede a sofrer de *overfitting*.

Outra abordagem bastante popular para critério de parada é a GL5 [Prechelt94]. Nele dividimos os vetores de entrada não apenas em treinamento e testes, mas em validação também. O tamanho deste novo conjunto deve ser uma fração do conjunto de treinamento. Ele funciona da seguinte maneira:

1. A cada fração de padrões de treinamento apresentados à MLP, insere-se um vetor de entrada de validação;
2. Calcula-se o erro obtido na propagação do padrão de validação;
3. Guarda-se o menor erro juntamente com o estado (valores dos pesos sinápticos) respectivo para a rede neural em questão;
4. Compara-se o erro calculado com o menor erro que foi guardado.
5. Se o novo erro for 5% maior que o menor erro – o treinamento é parado, pois a rede neural já estaria perdendo generalização. E a configuração da melhor rede é restaurada. A Fig. 13 mostra o melhor momento para parar o treinamento.
6. Se o novo erro for menor que o melhor erro, realiza-se o passo 3 e continua-se o treinamento repetindo os passos acima.
7. Senão (passo 5 e 6 falsos) nada é modificado e o treinamento continua até quando um dos critérios (número de épocas ou passo 5) seja satisfeito.

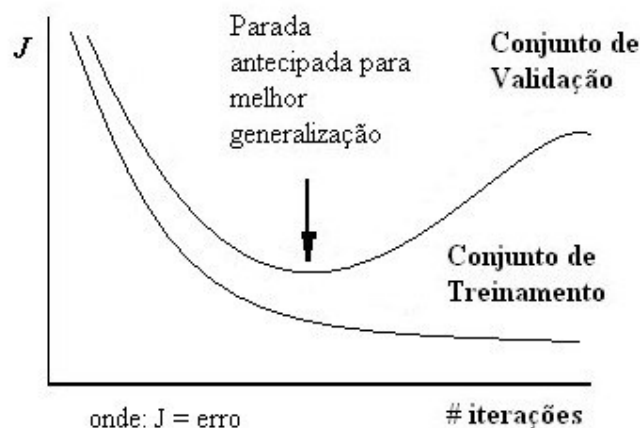


Fig. 13 – Gráfico mostrando parada antecipada do treinamento usando conjunto de validação

Terminado o treinamento, resta apresentar para a RNA o conjunto de testes (sem alterar nenhum peso), definir os critérios de avaliação sobre ele e, finalmente, validar a configuração (topologia, arquitetura, algoritmo de aprendizagem, parâmetros configuráveis, entre outros) testada. É importante frisar que o erro válido para esta rede MLP é o calculado a partir do conjunto de testes. A avaliação deve ser feita em cima das entradas que não foram vistas pela RNA durante o treinamento e validação.

A análise dos resultados está intimamente ligada ao problema. Por exemplo, um sistema baseado em redes neurais artificiais que propõe o controle de acesso a contas bancárias através do reconhecimento de voz exige um erro mínimo (talvez, abaixo de 1%), já um outro que pode fazer previsões sobre melhores rotas de trânsito admite um erro de 20%.

Na Engenharia (às vezes confundida com arte) das redes neurais artificiais – em especial no uso de redes *Multi-Layer Perceptron*, poucas regras apóiam os engenheiros nas suas decisões de projetos. Portanto o experimentalismo contextualizado é uma das poucas maneiras de se conceber e implementar soluções com redes neurais artificiais. Mesmo assim, com muitas decisões heurísticas, RNA são ferramentas poderosas para sistemas de decisão.



## Capítulo 4

# O Sistema Desenvolvido

O atual cenário das usinas sucro-alcooleiras, onde o gestor agrícola tem poucos subsídios computacionais no processo de tomada de decisão, sugere a necessidade de implementação de um sistema que o apóie na escolha dos lotes de cana-de-açúcar a serem selecionados para moagem.

Este suporte pode ser dado a partir da predição de pelo menos 3 (três) fatores de produtividade: PCC, TCH e fibra. Estes valores serão gerados a partir de uma rede neural artificial, previamente treinada com os dados históricos da respectiva usina. O agente decisor interagirá com a ferramenta na definição de cenários de colheita. Sabe-se que o cenário de decisão de uma colheita de cana-de-açúcar é definido a partir da escolha de algumas variáveis. As que foram incorporadas neste sistema foram: tonelagem desejada, PCC mínimo, fibra mínima e uma ordem de relevância entre estes fatores. A partir da definição de um cenário e a escolha de uma data de colheita são apresentadas as melhores possíveis combinações que satisfazem os critérios exigidos pelo gerente agrícola.

De posse dessas novas informações, as decisões a serem tomadas adquirem menos empirismo e reforçam a racionalidade do processo, já que os modelos propostos são baseados em conhecimentos anteriores, analisados logicamente. Isso produz soluções otimizadas reduzindo a necessidade de um experimentalismo prático.

Neste capítulo descreveremos o sistema proposto, suas características, a arquitetura aplicada, algumas funcionalidades, e as ferramentas e tecnologias utilizadas.

### 4.1 Características

O sistema proposto é uma aplicação web de suporte à decisão inteligente e desta forma, facilmente disponível para seus usuários. Para tanto, implementa autenticação exigindo login e senha. Isto é necessário, pois as decisões tomadas podem necessitar de um número maior de consultores e assim, a partir de qualquer computador ligado à Internet, com um navegador instalado, os gestores poderão utilizar a ferramenta restringindo o acesso de pessoas não autorizadas. É caracterizado pela simplicidade de uso. Os requisitos são bastante claros e intuitivos já que os usuários finais (gerentes) não, necessariamente, têm experiência com computadores nem possuem tempo disponível para aprenderem ferramentas complexas.

A linguagem de programação utilizada no desenvolvimento deste projeto foi *Java*, pois suas características atendem às necessidades deste projeto [Cornell00] [Deitel01]:

- O sistema é distribuído: sistemas web são por natureza distribuídos;

- O sistema deve ser independente de arquitetura: uma das características de Java que a tornou ideal para uso na elaboração de aplicativos distribuídos foi a sua independência de plataforma (usinas diferentes implicam, recursos diferentes). Java possui esta característica porque o compilador Java não gera instruções específicas para uma plataforma, mas sim um programa em um código intermediário, denominado *bytecode*, que pode ser descrito como uma linguagem de máquina destinada a um processador virtual. O código Java compilado pode então ser executado por um interpretador de *bytecodes*, a máquina virtual Java (do inglês, *Java Virtual Machine*, JVM), que é um emulador de processador.
- O sistema é portátil: a característica de neutralidade da arquitetura Java é o grande motivo pelo qual os programas em Java são portáteis. Outro aspecto da portabilidade envolve a estrutura ou os tipos de dados inerentes da linguagem, como inteiro, string e ponto flutuante.
- O sistema exige multitarefa: a maior parte dos sistemas operacionais hoje no mercado suportam à multitarefa, como o Windows, OS/2 e Unix, ou seja, o computador é capaz e executar diversas tarefas ao mesmo tempo. Cada um desses sistemas tem o seu tipo de multitarefa, preemptiva ou não, com ou sem multi-processamento. Java tem o suporte a multitarefa embutido na linguagem: um programa Java pode possuir mais de uma linha de execução (thread). Por exemplo, cálculos extensos (computação neural), que em geral demandam muito tempo do processador, podem ser escritos em uma thread, e a parte de interface com o usuário, que depende mais dos periféricos de I/O que do processador, pode ser executada em outra thread. Programação concorrente em geral é uma tarefa difícil, mas Java fornece diversos recursos de sincronização de processos que tornam a programação mais simples.
- O sistema requer alto desempenho: há muitas situações em que a interpretação de objetos de códigos de bytes proporciona desempenho aceitável. Mas outras circunstâncias exigem desempenhos mais elevados. Java concilia tudo isso oferecendo a tradução dos códigos de bytes para o código de máquina nativo em tempo de execução. O alto desempenho permite a implementação de aplicativos WEB em Java, na forma de programas pequenos e rápidos, que podem ampliar significativamente os recursos tanto do cliente quanto do servidor.

A partir de então, definida a LP, procedeu-se ao projeto e o desenvolvimento do sistema. Toda engenharia de software [Sommerville03] realizada será detalhada nas seções seguintes. Os requisitos necessários para rodar a ferramenta são:

- Cliente – qualquer dispositivo com um navegador *Internet Explorer 6.0* [IEexplorer05] ou superior;
- Servidor – deverá ser instalado em qualquer computador com o servidor de aplicação Tomcat 5.0.x [Tomcat06]. Mais detalhes na seção 4.5.

## 4.2 Funcionalidades

A partir de estudos, de pesquisas por ferramentas similares e reuniões com especialistas da área, foi produzido um documento de requisitos para este sistema (ver Anexo A). Nele está apresentada uma breve descrição da ferramenta, os requisitos funcionais e não funcionais e o escopo negativo. A Fig. 14 ilustra o diagrama de casos de uso (do inglês, *user case*, UC) com todas as funcionalidades do sistema e suas dependências entre si.

O agente decisor, único ator sistema, é responsável por: autenticar, exibir tela de apresentação, realizar previsão dos dados, realizar simulação, exibir tela de créditos e logout. Os UC realizar previsão dos dados e realizar simulação podem requerer um tempo maior de processamento e, por isso, incluem o UC exibir tela de processamento. O UC exibir tela de resultados, por sua vez, é chamado pela funcionalidade realizar simulação e por isso, o ator do sistema não tem ligação direta com ele. Existe uma dependência entre alguns casos de uso. O UC gerar logs só pode ser executado após a execução do UC realizar simulação, que por sua vez depende do UC realizar previsão dos dados. Conforme apresentado pelas setas pontilhadas entre estes balões

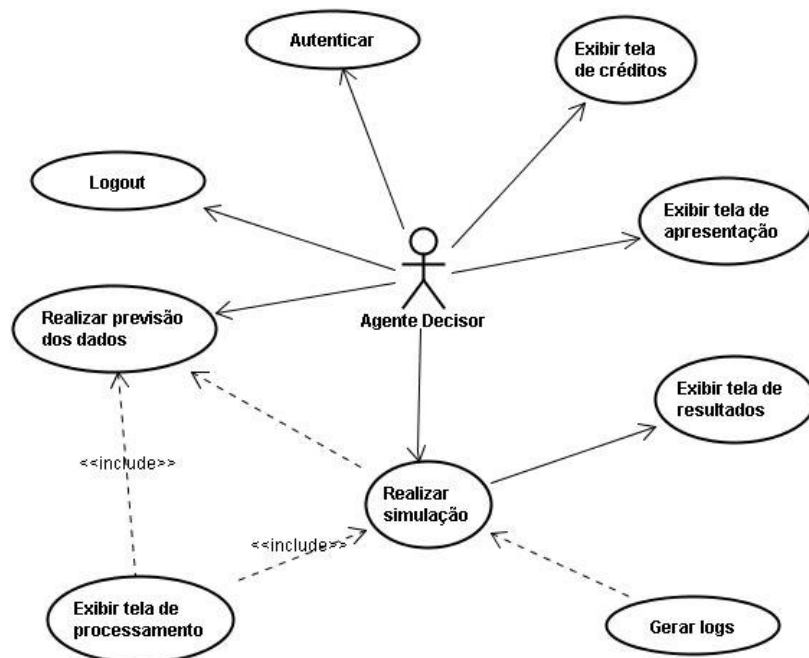


Fig. 14 – Diagrama de casos de uso para o sistema desenvolvido

O sistema é apresentado ao usuário a partir da tela de login (ver Fig. 15). Nela ele deve fornecer um nome de usuário válido e a senha correta para este. Em caso de problemas, existe um botão *Dúvidas* que fornecerá o apoio necessário para que a operação ocorra com sucesso.

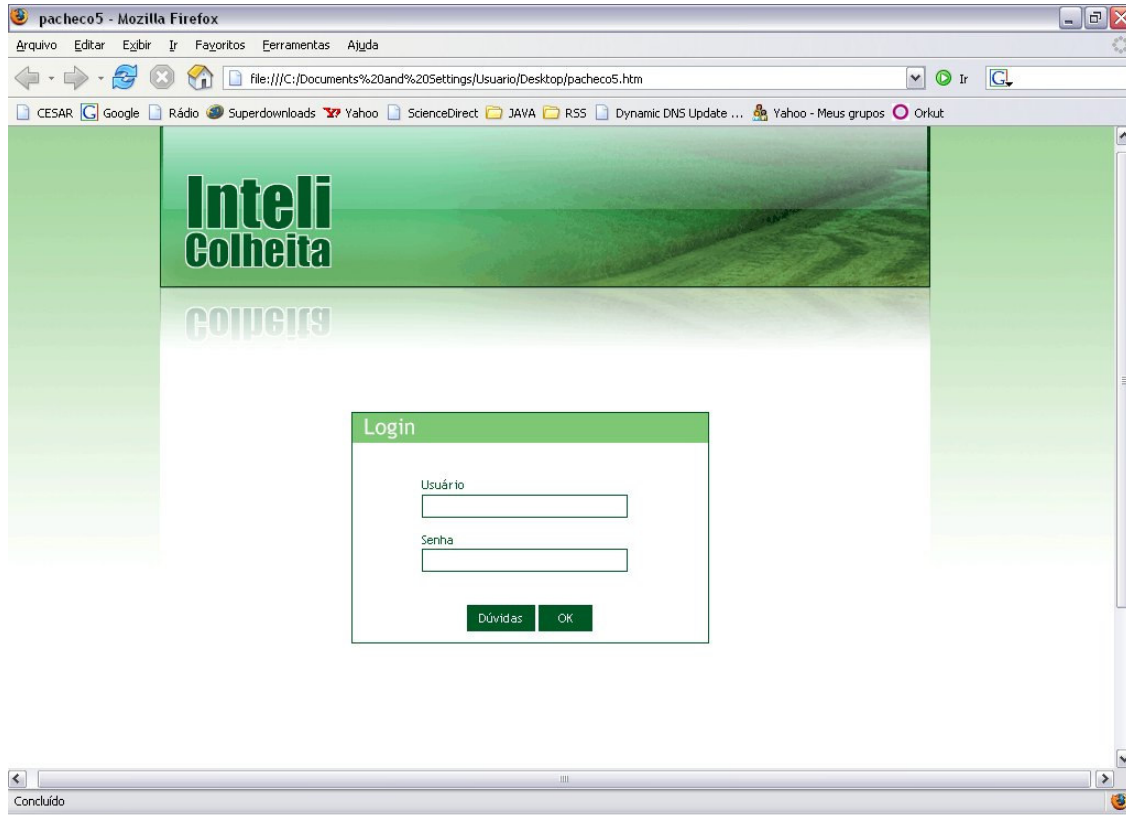


Fig. 15 – Tela de login onde é necessária a autenticação do usuário.

Após a autenticação, ao usuário é apresentada a uma tela de tutorial (*Início*) que descreve a aplicação e seu propósito, explica como utilizar as funcionalidades e oferece algumas instruções de uso da ferramenta. Com isto, ele estará apto a usar as funcionalidades (divididas em abas, ver Fig. 16) disponíveis a todo o momento no sistema.



Fig. 16 – Aba de funcionalidades do sistema

- Na aba *Colheita Inteligente*, o usuário realizará três passos:
  1. *Predição* – inicialmente o usuário deve informar à ferramenta a data que será usada como parâmetro para o dia da colheita. Ela será usada pelo motor neural para realizar a previsão dos fatores de produtividade PCC, TCH e fibra, que serão usados no passo seguinte. O agente decisor também informa a quantidade de cenários desejados (definidos no início deste capítulo) para esta colheita. Após os campos serem preenchidos seleciona-se o botão *Prever*. A Fig. 17 mostra um exemplo desta tela. Com o processamento neural realizado, os dados previstos são armazenados, temporariamente, em memória para a futura manipulação pelo modelo matemático.

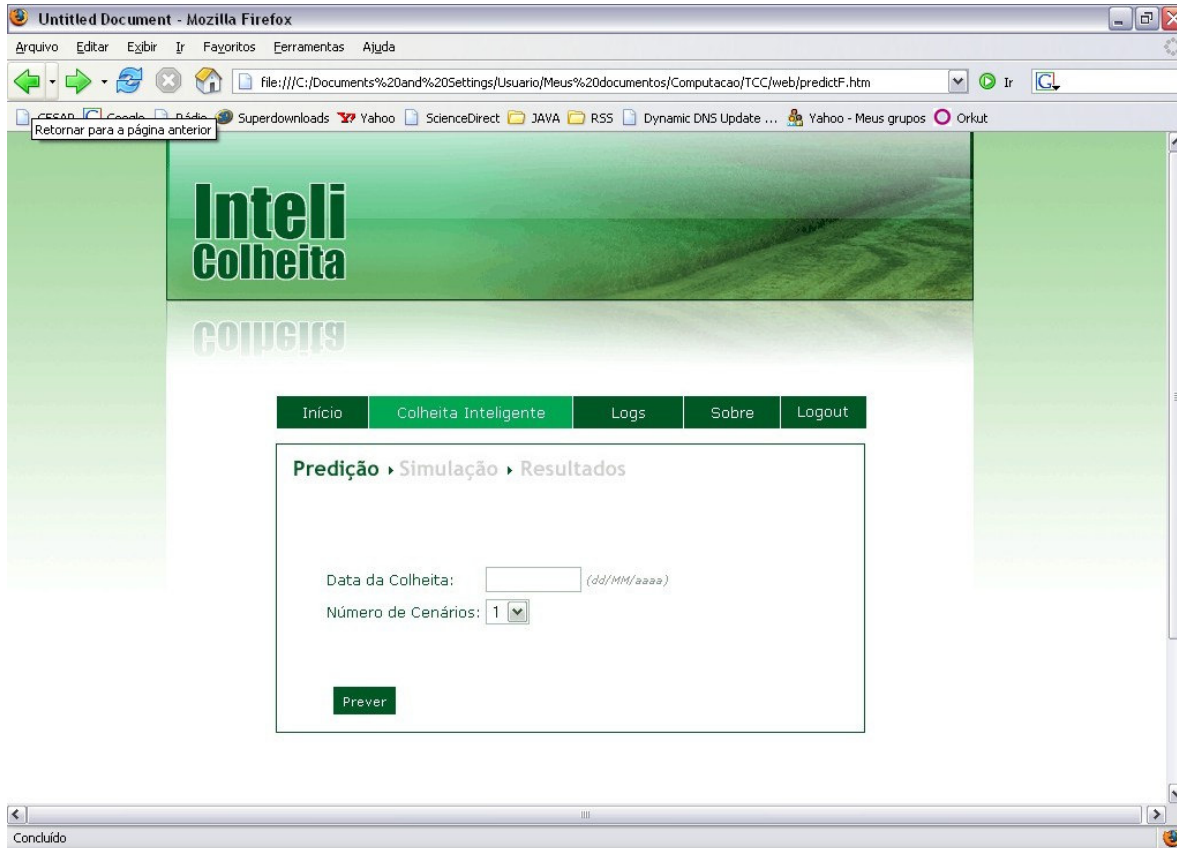


Fig. 17 – Tela de predição

2. *Simulação* – de acordo com o número selecionado na opção *número de cenários* na fase de *predição*, o usuário preencherá (a seu total critério) os campos:
- Tonelagem desejada;
  - PCC mínimo;
  - Fibra mínima;
  - E a relevância (prioridade) que cada um dos campos acima terá para o modelo de decisão.

Cada campo preenchido representa maior restrição ao modelo de decisão. Se algum campo não é completado, este não influenciará o modelo matemático e conseqüentemente, na escolhas dos lotes a serem colhidos. Definidos os cenários, o usuário prossegue clicando no botão *Simular*. A Fig. 18 exemplifica esta tela para um cenário.

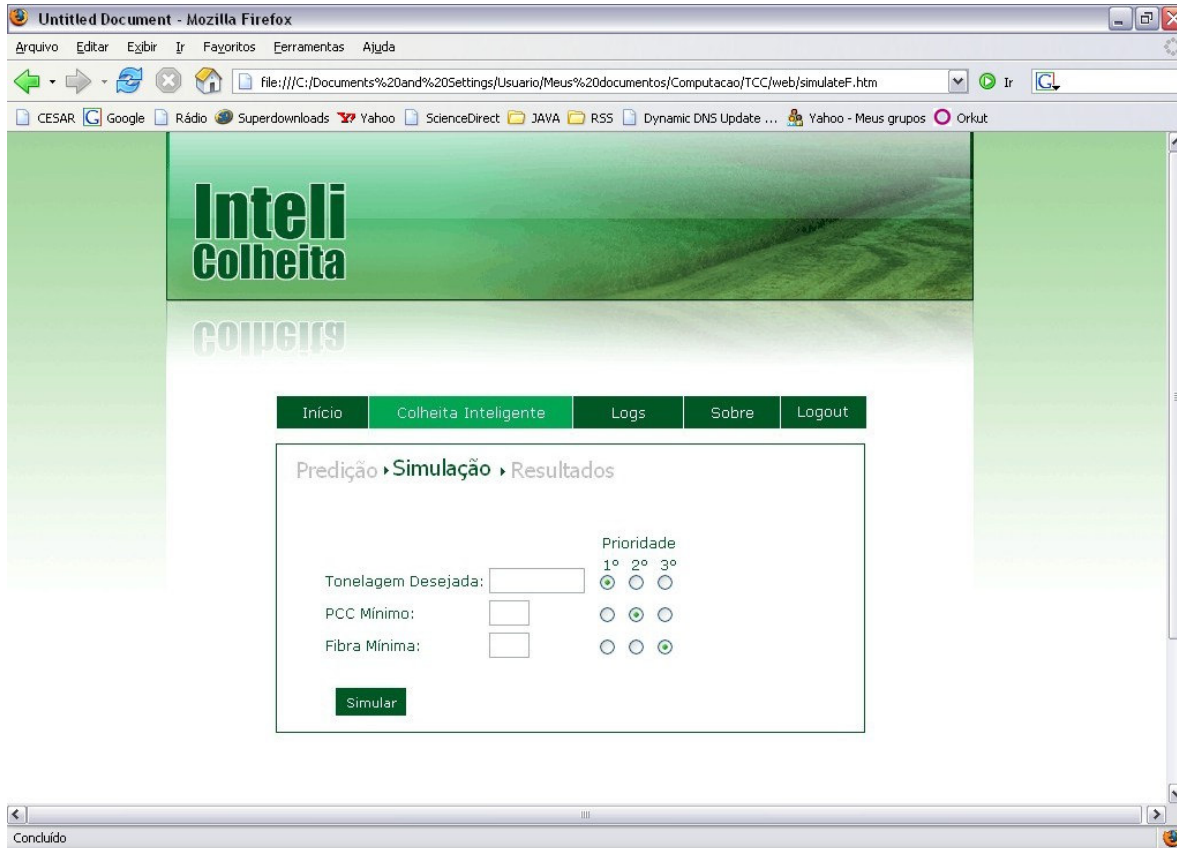


Fig. 18 – Tela de simulação

3. *Resultados* – após o processamento dos dados da RNA com os parâmetros de simulação pelo modelo matemático é exibida a tela de resultados. Nela consta o resumo das seguintes informações:

- a. Data da colheita;
- b. PCC médio dos lotes selecionados;
- c. Fibra média dos lotes selecionados;
- d. Tonelagem liberada;
- e. Uma lista com os números dos lotes sugeridos para a colheita.

Existe também um botão *iniciar nova colheita inteligente* que inicia novamente o processo a partir da funcionalidade *previsão*. A Fig. 19 ilustra esta tela.

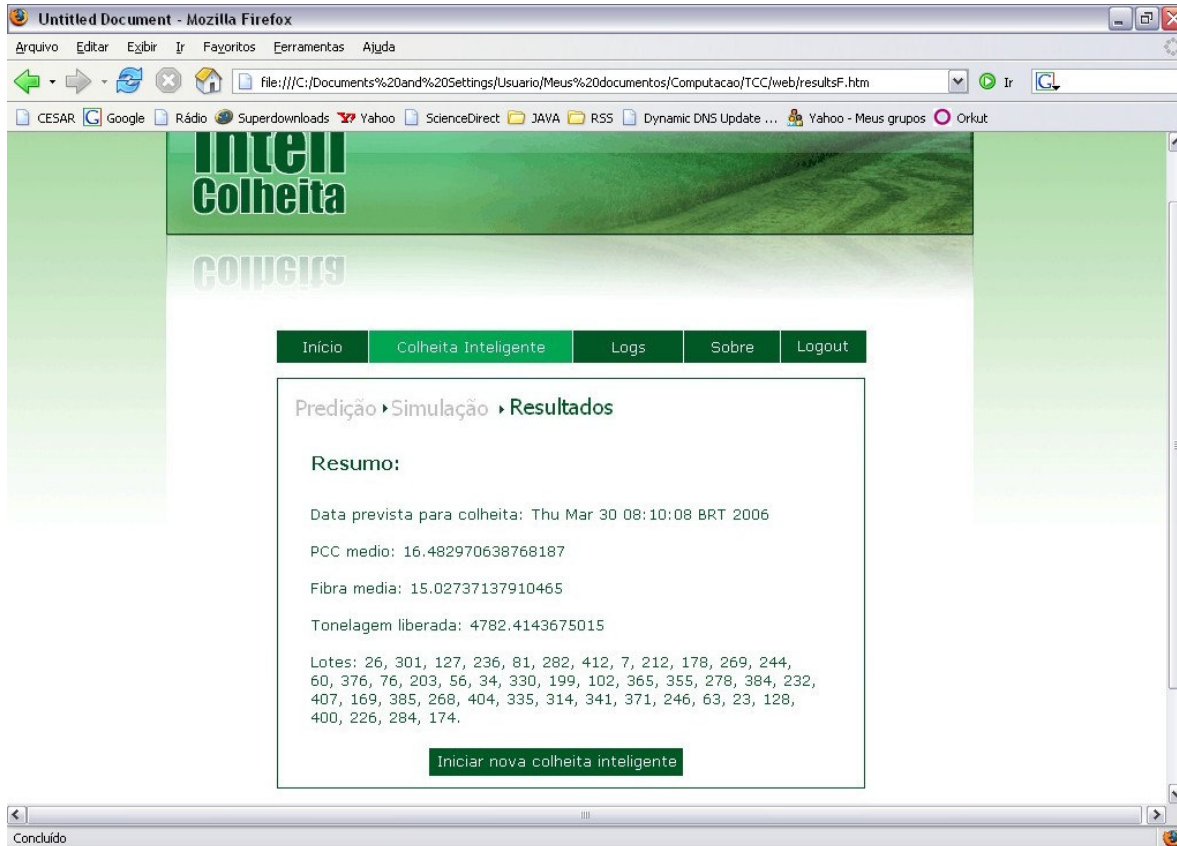


Fig. 19 – Tela de resultados

- *Processamento* – existe uma tela de processamento que será exibida sempre que alguma requisição for feita ou quando uma operação demande do sistema mais que 2 (dois) segundos de resposta. Isso é feito para evitar que um usuário faça a mesma requisição várias vezes, sobrecarregando desnecessariamente o sistema. A Fig. 20 mostra um exemplo desta tela. Ela será utilizada na previsão dos dados e na simulação de corte.
- *Logs* – na aba de logs o usuário tem acesso a todos os registros de simulação feitos por ele. Os arquivos (links) são identificados por data e hora da requisição e contém o resumo das mesmas informações apresentadas nos *resultados*. A Fig. 21 ilustra um exemplo desta tela.
- *Sobre* – nesta aba são apresentadas informações sobre o sistema, seu desenvolvedor e o trabalho que o originou. São exibidos algumas considerações finais, os créditos e alguns agradecimentos. A Fig. 22 exemplifica a tela.
- *Logout* – realiza o logout do sistema, invalidando a sessão do usuário e redirecionando para a página de login.

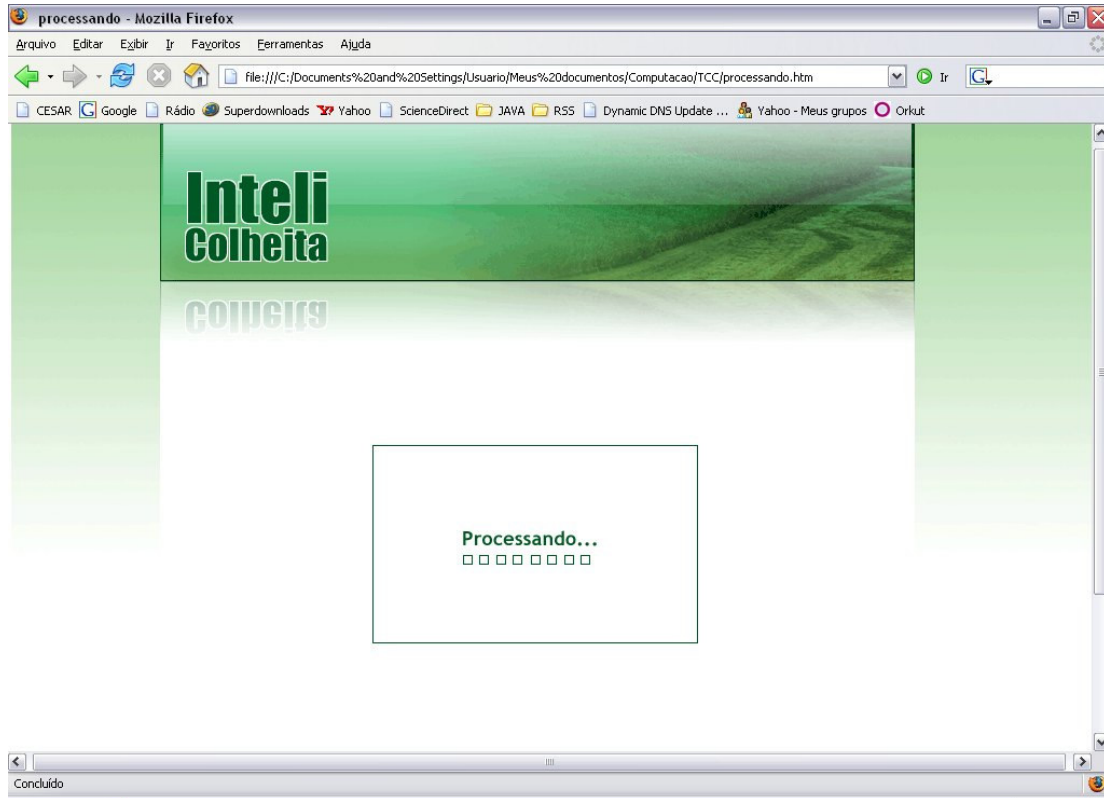


Fig. 20 – Tela de processamento

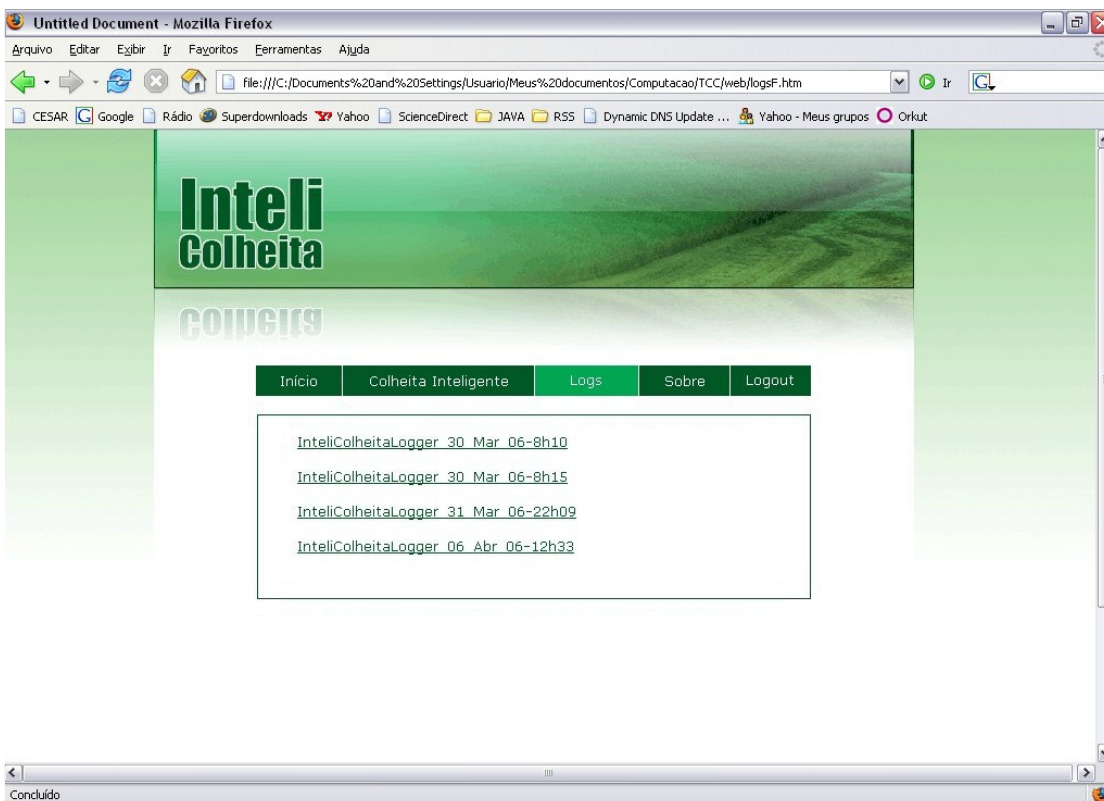


Fig. 21 – Tela de logs



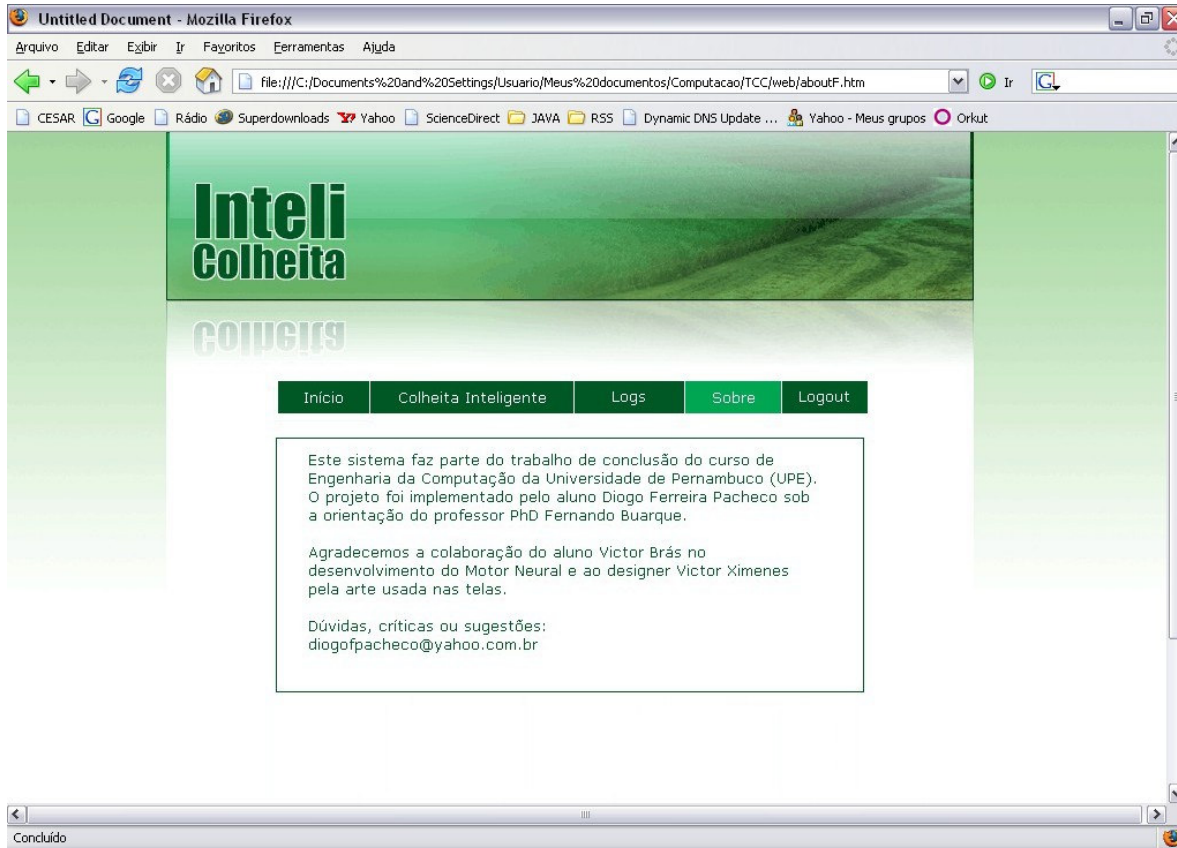


Fig. 22 – Tela sobre o sistema

Para um melhor entendimento do fluxo do sistema é apresentado o diagrama de seqüência para a principal funcionalidade do sistema (ver Fig. 23). O usuário inicia o fluxo clicando no botão *Prever* na aba *Colheita Inteligente*; a requisição é recebida pela classe *ProcessAction* que herda da classe *DispatchAction* do struts. A partir deste *Action* é criada uma instância da classe *InteliController* que é responsável pela interação entre as camadas e por todo controle do sistema. O controlador instancia a classe *InputLots* que representa os dados de entrada sobre os lotes da usina. É chamado o método *setHarvestDate* que modifica estes dados de modo que a data de colheita escolhida pelo usuário seja acrescida nestas informações. Na seqüência, o motor neural é instanciado a partir da classe *Backpropagation* que calcula os fatores de produtividade de cada lote através do método *executeTest*.

A tela é atualizada dentro da própria aba *Colheita Inteligente* para o passo seguinte. Ao clicar no botão *Simular* uma nova requisição é enviada ao *ProcessAction* que cria novamente um *InteliController*. No entanto, desta vez, o método chamado é o *simulate* que irá criar um modelo de decisão a partir da classe *DecisionModel* que aplica as condições estabelecidas pelo cenário criado (detalhes na seção 4.4). Após a chamada dos métodos *removeUndisiredValues*, *soEligibleLots* e *getMinimumEligibleLots* o modelo de decisão retorna para o controlador um vetor com os lotes selecionados para colheita. Com isso, o *InteliController* pode calcular as variáveis de saída (método *calculateOutputVariables*) e criar o registro de log (método *createInfoLogger*). Finalmente, a tela será mudada para exibir os resultados desta colheita inteligente.

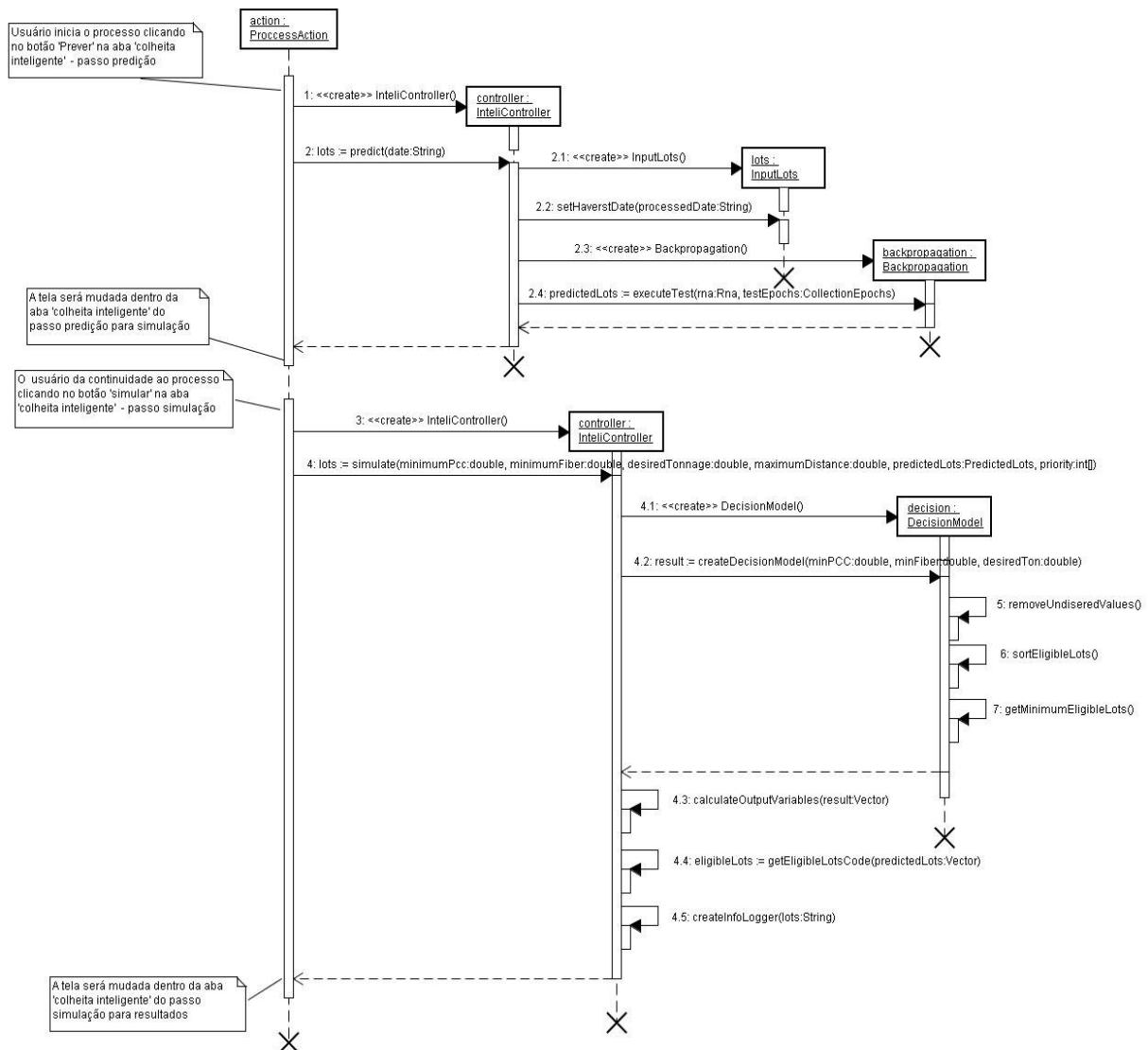


Fig. 23 – Diagrama de seqüência da funcionalidade *colheita inteligente*

### 4.3 Arquitetura

É um sistema cliente-servidor onde se utiliza a arquitetura MVC. Esta técnica fornece uma maneira de dividir a funcionalidade envolvida na manutenção e apresentação dos dados de uma aplicação. Este modelo não é novo e foi, originalmente, desenvolvida para mapear as tarefas tradicionais de entrada, processamento e saída para o modelo de interação com o usuário. Usando o padrão MVC fica fácil mapear esses conceitos no domínio de aplicações Web multicamadas.

Na arquitetura MVC o modelo representa os dados da aplicação e as regras do negócio que governam o acesso e a modificação dos dados. O modelo mantém o estado persistente do negócio e fornece ao controlador a capacidade de acessar as funcionalidades da aplicação encapsuladas pelo próprio modelo.

O componente de visualização renderiza o conteúdo de uma porção particular do modelo e a encaminha para o controlador as ações do usuário; acessa também os dados do modelo via controlador e define como esses dados devem ser apresentados.

O controlador define o comportamento da aplicação, é ele que interpreta as ações do usuário e as mapeia para chamadas do modelo. Em um cliente de aplicações Web essas ações do usuário poderiam ser cliques de botões ou seleções de menus. As ações realizadas pelo modelo incluem ativar processos de negócio ou alterar o estado do modelo. Com base na ação do usuário e no resultado do processamento do modelo, o controlador seleciona uma visualização a ser exibida como parte da resposta à solicitação do usuário. Há normalmente um controlador para cada conjunto de funcionalidades relacionadas. A Fig. 24 ilustra como os componentes *model*, *view* e *controller* se relacionam.

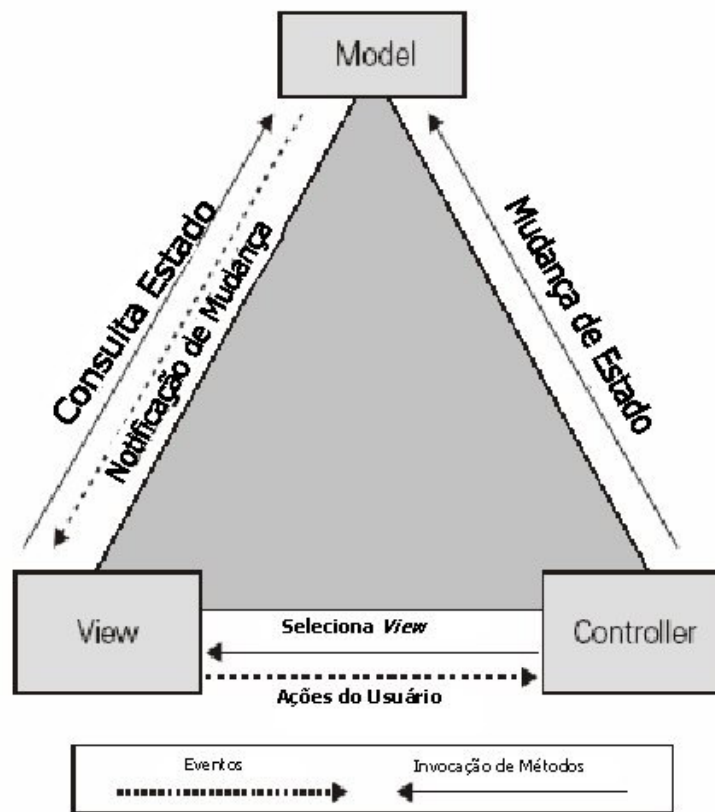


Fig. 24 - Representação da arquitetura *Model-View-Controller*. Como os componentes se relacionam

As vantagens do modelo MVC são:

- Facilidade de manutenção, teste e atualização de sistemas múltiplos, pois o modelo MVC gerencia múltiplos visualizadores usando o mesmo modelo;
- É muito simples incluir novos clientes apenas incluindo seus visualizadores e controles;
- Torna a aplicação escalável, pois pode se adaptar, de maneira eficiente, para trabalhar com redes maiores ou menores, executando tarefas de complexidade variável;

- É possível ter desenvolvimento em paralelo para o modelo, visualizador e controle, pois são independentes.

Algumas desvantagens importantes do modelo MVC são:

- Requer uma quantidade maior de tempo para analisar e modelar o sistema;
- Requer pessoal com maior especialização em engenharia de software;
- Não é aconselhável para pequenas aplicações.

Para o uso real desta arquitetura, o projeto foi dividido em pacotes, o que além de facilitar a organização e visualização, permite a divisão de responsabilidades criando uma estrutura de camadas no código programado. A Fig. 25 mostra o diagrama de classes simplificado do sistema.

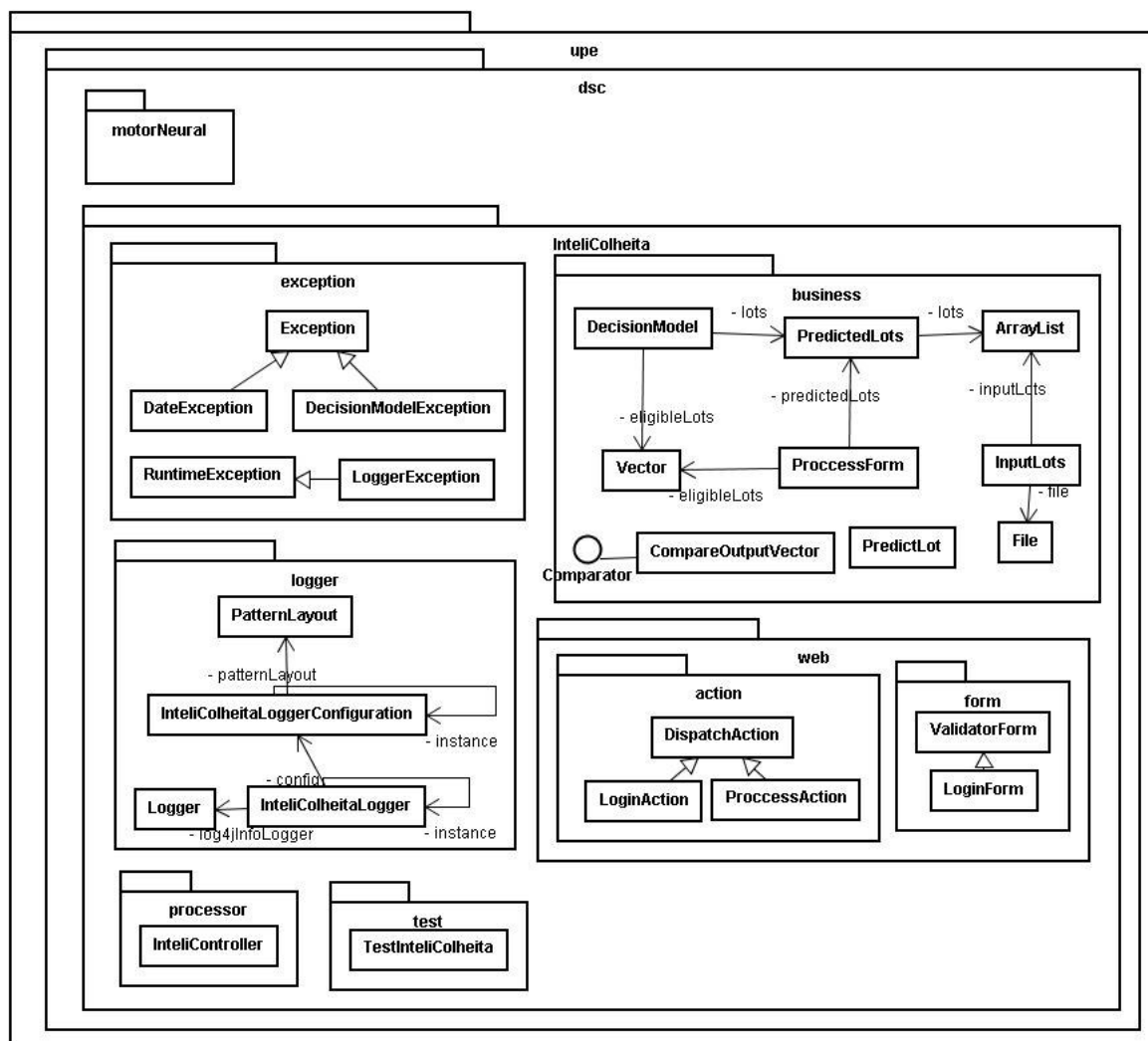


Fig. 25 - Diagrama de classes simplificado do projeto

A primeira camada do MVC é composta pelos seguintes pacotes:

- *Business* - onde se encontram as classes responsáveis pelas regras de negócio;
- *Exception* - pacote que reporta as possíveis exceções levantadas pelo sistema;

- *Logger* – contém as classes do *log4j* utilizadas para geração dos arquivos de log da aplicação.

A segunda camada, *View*, é formada pelo pacote *web* que contém as classes do *struts* nos pacotes *action* e *form*. Finalmente, a camada *Controller* é composta pelo pacote *processor* que contém a única classe de controlador da aplicação. O diagrama completo pode ser visto no Anexo B.

## 4.4 Suporte à Decisão Baseado em Tecnologia Neural

Já discutimos as peculiaridades a cerca de suporte à decisão na seção 2.3 e como a tecnologia neural trabalha, no capítulo 3. Falaremos agora como essas duas grandes áreas de conhecimento podem trabalhar juntas na resolução de problemas complexos.

As principais funcionalidades do sistema de decisão proposto são:

1. Predição dos fatores de produtividade para a cana-de-açúcar;
2. Definição de modelos de escolha de colheita.

Antes que estes fatores pudessem ser antecipados, foi preciso fazer toda a modelagem do problema agrícola. Suas particularidades e informações foram modeladas do mundo real para alguma representação significativa em redes neurais artificiais, no caso uma rede MLP. Os detalhes desta modelagem estão descritos na dissertação de mestrado de Fernando Buarque [Buarque98].

Alinhado com o foco principal deste trabalho, uma ferramenta computacional foi projetada, modelada e desenvolvida. Esta ferramenta utilizou as pesquisas anteriores na área de [Buarque98] estendidas por [Pacheco05], as quais produziram melhores resultados na predição da tríade PCC, fibra e principalmente TCH. A modelagem neural utilizada para o problema continuou igual à proposta original, anteriormente utilizada em todas as pesquisas mencionadas acima. Ela consistiu do pré-processamento dos dados brutos (i.e. normalizados e randomizados) obtidos de uma unidade agro-industrial do interior do Estado de São Paulo.

Após a modelagem neural do problema e o treinamento das RNA, o motor neural<sup>7</sup> estava preparado para gerar predições e, finalmente, ser validado. Estes valores gerados são as chaves no apoio à tomada de decisão. É neste ponto onde se encontra a inteligência do sistema. A possibilidade de prever o futuro é estar sempre um passo a frente da concorrência e o valor desta informação passa a ser incomensurável. A partir deles, os gerentes simulam ambientes reais de colheitas, definindo cenários econômicos.

No domínio agrícola de cana-de-açúcar pode-se dividir um ano em apenas duas fases: colheita e entressafra. Essa divisão ocorre devido à regra de negócio aplicadas nas usinas. Uma vez iniciada a moagem do vegetal, qualquer paralisação do sistema gera prejuízos, pois a cana moída retorna ao sistema, em forma de bagaço (fibra) na geração de energia. Desta maneira, evitando prejuízos e na tentativa de melhorar a produtividade da safra os gestores definem estes cenários de colheita exigindo dos lotes uma maior capacidade calorífica (fibra mais alta) ou quantidade de açúcares (PCC mais alto).

As previsões e as definições dos gestores agora passam por um modelo matemático que definirá quais lotes devem ser colhidos, respeitando alguns princípios e uma ordem lógica:

1. Cria-se um modelo de decisão passando os parâmetros definidos pelo usuário: tonelagem desejada, PCC mínimo e / ou fibra mínima;
2. Removem-se os lotes que não satisfazem aos requisitos mínimos de PCC ou fibra;

---

<sup>7</sup> O motor neural utilizado neste projeto é fruto de pesquisa de iniciação científica na UPE / Poli – DSC. Foi desenvolvido pelo aluno de Engenharia da Computação Victor Moura Brás, orientado pelo professor PhD Fernando Buarque. Pequenas adaptações foram procedidas para ajustá-lo a todo o sistema.

3. Ordena-se a lista restante de acordo com as prioridades ajustadas para cada fator de produtividade;
4. Selecionam-se os lotes, a partir dos mais importantes, até quando a característica restritiva (tonelagem desejada) do sistema seja satisfeita;
5. Com os lotes definidos, calculam-se as variáveis de saída para este cenário. As respostas são: PCC médio, fibra média, tonelagem liberada e a lista dos lotes a serem colhidos;
6. Gera-se um log de relatório contendo as informações do passo anterior para futuras consultas.

Com isso, os gerentes agrícolas podem tomar decisões muito mais embasadas. A partir dos próprios cenários criados, saberão de antemão as respostas que a natureza só relevaria no futuro. No entanto, o personagem que toma a decisão dificilmente será substituído por sistemas. A inteligência usada só gera informação legível e utilizável quando combinada com a interação e experiência de seus usuários.

Assim, obstáculos são superados contando com a colaboração das redes neurais na geração de novas informações, e dos sistemas de suporte à decisão na manipulação destas informações e no apoio no processo decisivo.

## 4.5 Ferramentas e Tecnologias Utilizadas

Várias ferramentas e tecnologias facilmente encontradas na web foram utilizadas neste projeto. Podemos destacar as que mais contribuíram para a execução em tempo deste sistema:

- Eclipse;
- Tomcat;
- Dreamweaver;
- Struts;
- Log4J.

No Anexo C detalhamos estas principais ferramentas e tecnologias utilizadas.

## Capítulo 5

# Experimentos e Resultados Obtidos

Os experimentos realizados neste trabalho foram necessários para realizar a validação do motor neural e do modelo de decisão. A validação do modelo de decisão foi possível a partir da comparação dos lotes selecionados pelo sistema diante de uma solução manual. O motor neural foi constituído de uma parte reusada de um simulador neural não validado e parte adaptada para integração com o sistema desenvolvido. Desta forma, como o objetivo deste trabalho não foi a busca de melhores valores de predição por parte do modelo neural, as simulações não foram o alvo maior nesta monografia.

Não houve necessidade de iniciarmos as simulações com uma pesquisa exploratória dos parâmetros da rede neural (e.g. n° de camadas escondidas, n° de neurônios nessas camadas, função de ativação, taxa de aprendizado ou n° de ciclos), de maneira que as simulações realizadas tiveram apenas o intuito de validar o motor neural como meio de geração de cenários de decisões.

Para tal, usamos os mesmos padrões de dados que foram utilizados em [Pacheco05] e desta forma foi possível realizar uma comparação satisfatória dos novos resultados produzidos aqui com outros publicados anteriormente.

### 5.1 Manipulação dos dados

Como explicado no Capítulo 3, uma das formas para que possa ocorrer o processo de aprendizagem e a geração de conhecimento dentro de uma rede neural é a partir da apresentação de dados históricos. Estes dados agem como um “professor” no paradigma supervisionado.

O volume de dados usado neste projeto representa valores reais fornecidos por uma usina sucro-alcooleira do interior de São Paulo. De acordo com a modelagem do problema alguns dados não puderam ser fornecidos ou foram descartados e, portanto, não compõem o conjunto de treinamento. A seguir, são mostrados os dados que foram considerados / desconsiderados nas diversas simulações procedidas.

#### Entradas

- Safra (código)..... 1ª **considerada**
- Engenho (código).....desconsiderada
- Lote (código).....desconsiderada
- Variedade da cana (código)..... *escolhida a mais **significante***

- Estágio de maturação (código corte)..... 2<sup>a</sup> **considerada**
- Idade do corte (em dias)..... 3<sup>a</sup> **considerada**
- Época do plantio (dia/mês/ano)..... 4<sup>a</sup>, 5<sup>a</sup> e 6<sup>a</sup> **consideradas**
- Época do corte (dia/mês/ano)..... 7<sup>a</sup>, 8<sup>a</sup> e 9<sup>a</sup> **consideradas**
- Topografia (código).....desconsiderada
- Tipo do solo (código)..... 10<sup>a</sup> **considerada**
- Compactação do solo (código).....não fornecida / não considerada
- Tempo médio de transporte (horas)..... não fornecida / não considerada
- Área (há).....desconsiderada

### Saídas

- PCC (médio)..... 11<sup>a</sup> **considerada**
- TCH (médio)..... 12<sup>a</sup> **considerada**
- Fibra (médio)..... 13<sup>a</sup> **considerada**

Com esta definição, os dados foram organizados em tabelas onde cada linha representa os valores de um único lote. Nas treze (13) colunas são dispostas as variáveis consideradas na modelagem neural e descritas acima, respectivamente. A Fig. 26 oferece uma visão exemplo da organização do volume de dados.

1	10150020001	12	5	830	951208	941201	1	1	3,69	12,7586	83,65	15,55
1	10150020002	12	5	830	951208	941201	1	1	6,81	13,0873	27,51	14,75
1	10150020003	12	5	831	951208	941202	1	1	2,54	11,8357	71,36	15,70
1	10150020004	12	5	394	950227	940816	1	1	7,54	12,6365	65,76	11,09
1	10150020005	12	5	394	950224	940815	1	1	3,88	13,6226	61,49	11,25
1	10150020006	12	5	394	0	940816	1	1	8,82	13,6766	54,48	11,02
1	10150020007	12	5	394	950301	940816	1	1	9,78	13,2525	54,62	10,46
1	10150020008	12	5	831	951208	941202	1	1	5,41	11,4570	61,47	16,15
1	10150020009	12	5	830	951208	941201	1	1	6,42	13,5820	106,64	16,69
1	10150020010	12	5	830	951208	941201	1	1	2,80	13,0333	69,54	14,46
1	10150020011	12	3	394	960425	940630	1	1	4,95	12,7339	93,41	11,43
1	10150030001	27	1	571	930225	940919	1	1	2,98	15,1545	167,14	11,00
1	10150030002	27	1	570	930225	940918	1	1	2,47	14,6998	176,18	12,63

... Até 5186 registros.

Fig. 26 – Amostra dos dados originais, antes de serem pré-processados

Para eliminar disparidades entre escalas nos dados originais, antes de serem apresentados para as RNA eles devem ser pré-processados, de modo que suas cardinalidades não influenciem negativamente o processamento neural. Pode-se observar a nova massa de dados na Fig. 27.



```
# Input pattern 1:
0 1 0 0.2308 0.4114 0.9072 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0.9896 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0
# Output pattern 1:
0.7105 0.2858 0.6603
# Input pattern 2:
0 0 1 0.3846 0.408 0.9381 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0
# Output pattern 2:
0.7039 0.2669 0.5737
# Input pattern 3:
0 1 0 0.0769 0.4949 0.9691 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0.9896 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0
# Output pattern 3:
0.7604 0.3321 0.5754
# Input pattern 4:
0 1 0 0.1538 0.4354 0.9588 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0.9896 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0
# Output pattern 4:
0.7503 0.3019 0.6157
# Input pattern 5:
0 0 1 0.2308 0.3851 0.9588 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0
# Output pattern 5:
0.6747 0.1787 0.8301
# Input pattern 6:
0 0 1 0.2308 0.4914 0.9175 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0.9896 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0
# Output pattern 6:
0.7049 0.168 0.6229
# Input pattern 7:
1 0 0 0.1538 0.608 0.9485 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0.9792 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0
# Output pattern 7:
0.844 0.265 0.5537
# Input pattern 8:
0 1 0 0.3077 0.536 0.9381 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0.9896 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0
# Output pattern 8:
0.6877 0.3011 0.5941
... Até 418 registros
```

Fig. 27 – Amostra de dados pré processados e prontos para serem apresentados à RNA

Comparando-se as duas figuras anteriores, percebe-se claramente, um aumento no número de dados. As 13 variáveis originais agora são representadas por 41. Isto ocorre devido às técnicas usadas no pré-processamento. Esta nova estrutura é apresentada na Tabela 1, onde safra e solo são representados com 3 entradas; estágio, idade, ano, PCC, TCH e fibra com 1 entrada cada um; mês são 12 entradas; e dia 2 entradas (indica a primeira ou segunda quinzena do mês).

Tabela 1 – Definição dos dados pré-processados

# Input	SAFRA			ESTÁGIO	IDADE	PLANTIO												CORTE																						
	0	1	0			ANO	MÊS						DIA	ANO	MÊS				DIASOLO																					
0	1	0	0,2308	0,4114	0,9072	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0,9896	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0
# Output	PCC			TCH	FIBRA																																			
	0,7105	0,2858	0,6603																																					

### 5.2 Validação do simulador neural

Os dados de treinamento foram divididos em três grupos: 50% para treinamento, 25% para validação e 25% para teste. Como desempenho não foi o objetivo deste projeto, simulamos apenas três dos cinco enfoques abordados em [Pacheco05]. Foram eles:

- a) Enfoque 0 – a predição considera diferentes safras e anos como fatores relevantes;
- b) Enfoque 1 – a predição não considera safra e ano relevantes;
- c) Enfoque 2 – a predição não considera safra, ano e dia como relevantes;

A seguir, serão detalhados os enfoques abordados e serão apresentadas tabelas com os erros obtidos para os três fatores de produtividade utilizados neste projeto: PCC, TCH e Fibra. A Tabela 2 mostra um resumo dos enfoques abordados nestes experimentos e a Tabela 9 apresenta um resumo comparativo entre os enfoques testados e as saídas calculadas por cada um deles.

Tabela 2 – Resumo dos enfoques testados no projeto

Configuração das Redes Neurais Artificiais					
Experimentos	Entrada	Escondidas - Neurônios	Taxa de aprendizado	Função Ativação	Ciclos
<i>Enfoque 0</i>	38	1 -- 16	0,1	Logística	15000
<i>Enfoque 1</i>	33	1 -- 9	0,2	Logística	3000
<i>Enfoque 2</i>	29	1 -- 8	0,2	Logística	5000

No *enfoque 0* o erro médio quadrado (EMQ, do inglês *mean square error*, MSE) calculado anteriormente foi de *0,012676*. A configuração da RNA era composta pela camada de entrada com 38 neurônios, por uma camada escondida com 16 neurônios, taxa de aprendizado igual a *0,1*, a função de ativação era a *logística* e o número de ciclos foi *15000*. São mostrados mais detalhes na Tabela 3.

Tabela 3 – Detalhes do treinamento da RNA para o *enfoque 0* em [Pacheco05]

SNNS		PCC		TCH		Fibra	
		<i>esperado</i>	<i>calculado</i>	<i>esperado</i>	<i>calculado</i>	<i>esperado</i>	<i>calculado</i>
MSE: 0,012676	<b>Desv Pad(%)</b>	9,83%	<b>8,35%</b>	35,86%	<b>24,70%</b>	14,94%	<b>12,33%</b>
SSE: 5,298743	<b>Desv Pad</b>	0,072989	<b>0,062121</b>	0,107142	<b>0,077478</b>	0,090735	<b>0,072194</b>
	<b>Erro(%)<sup>8</sup></b>	4,384529434		23,94003577		7,480063294	

O mesmo *enfoque 0* abordado no motor neural utilizado neste projeto produziu os seguintes resultados (ver Tabela 4):

Tabela 4 - Detalhes do novo treinamento da RNA para o *enfoque 0*

Motor Neural		PCC		TCH		Fibra	
		<i>esperado</i>	<i>calculado</i>	<i>esperado</i>	<i>calculado</i>	<i>esperado</i>	<i>calculado</i>
MSE: 0,013009	<b>Desv Pad(%)</b>	7,85%	<b>9,83%</b>	25,25%	<b>35,86%</b>	11,71%	<b>14,94%</b>
SSE: 5,437696	<b>Desv Pad</b>	0,058675	<b>0,072989</b>	0,077804	<b>0,107142</b>	0,071519	<b>0,090735</b>
	<b>Erro(%)<sup>8</sup></b>	4,709976767		20,14190303		7,572280824	

No *enfoque 1* o EMQ calculado anteriormente foi de *0,013807*. A configuração da RNA era composta pela camada de entrada com 33 neurônios, por uma camada escondida com 9 neurônios, taxa de aprendizado igual a *0,2*, a função de ativação era a *logística* e o número de ciclos foi *3000*. São mostrados mais detalhes na Tabela 5.

Tabela 5 - Detalhes do treinamento da RNA para o *enfoque 1* em [Pacheco05]

SNNS		PCC		TCH		Fibra	
		<i>esperado</i>	<i>calculado</i>	<i>esperado</i>	<i>calculado</i>	<i>esperado</i>	<i>calculado</i>
MSE: 0,013807	<b>Desv Pad(%)</b>	9,83%	<b>6,86%</b>	35,86%	<b>24,98%</b>	14,94%	<b>11,00%</b>
SSE: 5,771358	<b>Desv Pad</b>	0,072989	<b>0,050688</b>	0,107142	<b>0,073589</b>	0,090735	<b>0,06578</b>
	<b>Erro(%)<sup>8</sup></b>	5,560124536		22,76091397		8,114029771	

O mesmo *enfoque 1* abordado no atual motor neural alcançou os seguintes resultados (ver Tabela 6):

<sup>8</sup> Erro baseado na média dos erros. Calculado a partir da Equação 1.

Tabela 6 – Detalhes do novo treinamento da RNA para o *enfoque 1*

		PCC		TCH		Fibra	
		<i>esperado</i>	<i>calculado</i>	<i>esperado</i>	<i>calculado</i>	<i>esperado</i>	<i>calculado</i>
<b>Motor Neural</b>	<b>Desv Pad(%)</b>	6,97%	<b>9,83%</b>	21,92%	<b>35,86%</b>	11,21%	<b>14,94%</b>
MSE: 0,014525	<b>Desv Pad</b>	0,052215	<b>0,072989</b>	0,071156	<b>0,107142</b>	0,0676	<b>0,090735</b>
SSE: 6,07158	<b>Erro(%)<sup>s</sup></b>	5,431184496		20,78644696		8,140109293	

Finalmente, no *enfoque 2* o EMQ calculado previamente foi de *0,013941*. A configuração da RNA era composta por pela camada de entrada com 29 neurônios, uma camada escondida com 8 neurônios, taxa de aprendizado igual a *0,2*, a função de ativação era a *logística* e o número de ciclos foi *5000*. Maiores informações são exibidas na Tabela 7.

Tabela 7 - Detalhes do treinamento da RNA para o *enfoque 2* em [Pacheco05]

		PCC		TCH		Fibra	
		<i>esperado</i>	<i>calculado</i>	<i>esperado</i>	<i>calculado</i>	<i>esperado</i>	<i>calculado</i>
<b>SNNS</b>	<b>Desv Pad(%)</b>	9,83%	<b>6,98%</b>	35,86%	<b>23,77%</b>	14,94%	<b>10,91%</b>
MSE: 0,012876	<b>Desv Pad</b>	0,072989	<b>0,051698</b>	0,107142	<b>0,069748</b>	0,090735	<b>0,065444</b>
SSE: 5,382159	<b>Erro(%)<sup>s</sup></b>	5,413837255		21,92734347		7,863321149	

E, os resultados alcançados para o *enfoque 2* neste projeto produziu os seguintes valores (ver Tabela 8):

Tabela 8 – Detalhes do novo treinamento da RNA para o *enfoque 2*

		PCC		TCH		Fibra	
		<i>esperado</i>	<i>calculado</i>	<i>esperado</i>	<i>calculado</i>	<i>esperado</i>	<i>calculado</i>
<b>Motor Neural</b>	<b>Desv Pad(%)</b>	6,54%	<b>9,83%</b>	21,68%	<b>35,86%</b>	10,40%	<b>14,94%</b>
MSE: 0,013941	<b>Desv Pad</b>	0,048792	<b>0,072989</b>	0,06476	<b>0,107142</b>	0,06287	<b>0,090735</b>
SSE: 5,827261	<b>Erro(%)<sup>s</sup></b>	5,518558626		19,99227841		8,240514724	

Confrontando-se os enfoques calculados usando o simulador *Stuttgart Neural Networks Simulator (SNNS)* em [Pacheco05] e o **Motor Neural** usado neste sistema, pode-se assumir que ambos produziram aproximadamente a mesma resposta e, naturalmente, o motor neural utilizado está válido. Logo abaixo mostramos um resumo das simulações onde constam as configurações de rede utilizadas em cada *enfoque* e os resultados calculados por cada um destes; bem como as diferenças entre os valores obtidos com estes dois simuladores as quais se mostraram sempre menores que 10% (ver Tabela 9).

Tabela 9 – Resumo dos enfoques testados em [Pacheco05] e neste trabalho, comparados em relação ao EMQ

Configuração das Redes Neurais Artificiais						Resultados das Simulações (EMQ)		
Experimentos	Entrada	Escondidas - Neurônios	Taxa de aprendizado	Função Ativação	Ciclos	SNNS	Motor Neural	Diferença %
<i>Enfoque 0</i>	38	1 -- 16	0,1	Logística	15000	0,012676	0,013009	<b>2,627</b>
<i>Enfoque 1</i>	33	1 -- 9	0,2	Logística	3000	0,013807	0,014525	<b>5,200</b>
<i>Enfoque 2</i>	29	1 -- 8	0,2	Logística	5000	0,012876	0,013941	<b>8,271</b>

### 5.3 Validação da aplicação de decisão

Com a aprovação do modelo neural inteligente (de predições), uma segunda fase se faz necessária para a validação do modelo matemático de decisão do projeto. Estes novos testes iniciam-se a partir do arquivo de lotes com predições geradas pela RNA (motor neural).

Imagine uma situação hipotética onde um profissional especializado em colheitas agrícolas recebe este arquivo composto de informações futuras a respeito dos lotes cultivados em sua usina. O arquivo é composto de quatro colunas, a primeira traz a identificação do lote e as três seguintes contém valores de PCC, TCH e fibra, respectivamente. Parte deste arquivo é mostrado a seguir (vide Tabela 10).

Tabela 10 – Parte do conjunto de lotes com valores de predição gerados pelo sistema

Lotes	PCC	TCH	Fibra
22	16,52172	92,5419	14,71501
26	16,48193	50,16509	15,64129
34	16,55628	73,19184	15,44009
56	16,55598	73,21408	15,44208
102	16,56073	72,86365	15,41072
127	16,80078	73,95839	14,79606
131	16,47562	60,70406	16,53652
169	16,58517	63,38117	16,84698
176	17,01822	71,73766	14,48083
199	16,56073	72,86365	15,41072
232	16,78821	74,34596	14,12165
238	16,45649	92,82518	16,36692
245	16,69994	86,01509	14,33346
268	16,75444	66,09908	14,21014
270	16,84736	91,99919	13,08598
278	16,78171	73,39334	14,14978
285	16,51976	92,59422	14,71955
298	16,45959	115,8839	11,71695
301	16,73833	55,70862	14,34332
314	16,45598	120,9602	14,97912
316	16,84736	91,99919	13,08598
355	16,79005	74,52086	14,12466
365	16,56103	72,84186	15,40877
371	16,44631	89,26981	16,23406
384	16,78869	74,39159	14,12243
385	16,58517	63,38117	16,84698
399	16,51976	92,59422	14,71955
404	16,50928	64,4401	15,36563
407	16,78821	74,34596	14,12165
408	17,03971	85,65791	13,0575

A depender da necessidade do gestor, pode ser escolhida uma quantidade de fibra alta, por exemplo, para garantir o fornecimento de matéria seca para produção de energia; outra hipótese seria um PCC alto para garantir bons níveis de açúcares na produção de álcool.

Definindo, por exemplo, um cenário de colheita apresentando as seguintes condições:

- PCC mínimo = 16;
- Fibra mínima = 15;
- Tonelagem desejada = 650;
- Prioridade dos parâmetros (da maior para menor) = PCC → TCH → Fibra.

Finalmente, solicita ao especialista que selecione os melhores lotes desse arquivo que satisfaçam ao cenário econômico exemplificado. Passados vários minutos ele apresenta uma solução (ver Tabela 11) depois de ter:

- ordenado os dados decrescentemente, seguindo a fila de prioridades;
- marcado de vermelho os valores que não satisfazem aos requisitos mínimos exigidos;
- assinalado de negrito os demais lotes até que somem a tonelagem desejada;
- finalmente, estes (em negrito) serão os lotes indicados.

Tabela 11 – Solução apresentada por um especialista agrícola a partir de um arquivo de lotes preditos e da definição de um cenário de colheita

Lotes	PCC	TCH	Fibra
408	17,03971	85,65791	13,0575
176	17,01822	71,73766	14,48083
316	16,84736	91,99919	13,08598
270	16,84736	91,99919	13,08598
127	16,80078	73,95839	14,79606
355	16,79005	74,52086	14,12466
384	16,78869	74,39159	14,12243
407	16,78821	74,34596	14,12165
232	16,78821	74,34596	14,12165
278	16,78171	73,39334	14,14978
268	16,75444	66,09908	14,21014
301	16,73833	55,70862	14,34332
245	16,69994	86,01509	14,33346
<b>385</b>	<b>16,58517</b>	<b>63,38117</b>	<b>16,84698</b>
<b>169</b>	<b>16,58517</b>	<b>63,38117</b>	<b>16,84698</b>
<b>365</b>	<b>16,56103</b>	<b>72,84186</b>	<b>15,40877</b>
<b>102</b>	<b>16,56073</b>	<b>72,86365</b>	<b>15,41072</b>
<b>199</b>	<b>16,56073</b>	<b>72,86365</b>	<b>15,41072</b>
<b>34</b>	<b>16,55628</b>	<b>73,19184</b>	<b>15,44009</b>
<b>56</b>	<b>16,55598</b>	<b>73,21408</b>	<b>15,44208</b>
22	16,52172	92,5419	14,71501
399	16,51976	92,59422	14,71955
285	16,51976	92,59422	14,71955
<b>404</b>	<b>16,50928</b>	<b>64,4401</b>	<b>15,36563</b>
<b>26</b>	<b>16,48193</b>	<b>50,16509</b>	<b>15,64129</b>
<b>131</b>	<b>16,47562</b>	<b>60,70406</b>	<b>16,53652</b>
298	16,45959	115,8839	11,71695
238	16,45649	92,82518	16,36692
314	16,45598	120,9602	14,97912
371	16,44631	89,26981	16,23406

Fica claro que esta abordagem manual é inviável, pois na prática, os arquivos de previsão representam milhares de lotes o que demandaria horas até quando o agente decisor pudesse fornecer uma resposta satisfatória.

Com uma alternativa a esse procedimento manual (e muito laborioso), na sequência executamos a ferramenta desenvolvida, e ora apresentada, configurando as mesmas restrições mencionadas acima. Obtivemos os mesmos resultados, mas com uma óbvia praticidade para o tomador de decisão, o processamento ocorreu em questões de segundos.

Comparando-se os lotes indicados pelo sistema com os que foram selecionados manualmente pelo gerente agrícola, verifica-se que o modelo de decisão automatizado utilizado [Buarque98] – agora instrumentalizado por este trabalho – está efetivamente operando como desejado. Ou seja, além de sugerir lotes com quantidades de matéria prima necessária a satisfazer todas as restrições (parametrizáveis) do negócio, estas escolhas se fazem sobre valores de previsão validados e em um tempo extremamente rápido; a Fig. 28 mostra um exemplo da saída com resultados sugeridos ao usuário-gerente para um dado cenário de decisão.

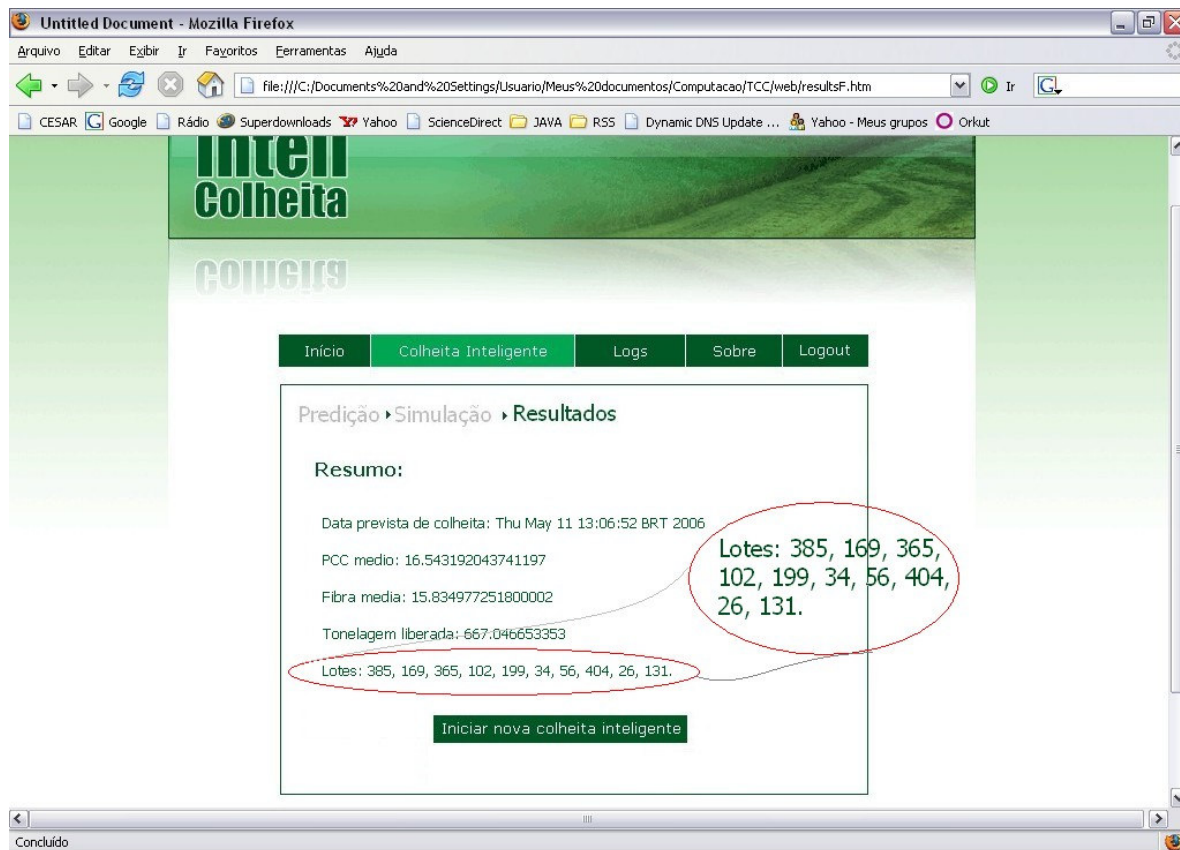


Fig. 28 – Tela de resultados com zoom nos lotes indicados para colheita

## Capítulo 6

# Conclusões e Trabalhos Futuros

### 6.1 Contribuições

Devido ao forte empirismo usado atualmente nas decisões de colheita de cana-de-açúcar existiram alguns trabalhos que aplicaram com sucesso técnicas de neurocomputação na predição de fatores de produtividade agrícola. Este trabalho realiza o passo seguinte destes estudos - a implementação da metodologia proposta por [Buarque98] no desenvolvimento de um sistema de suporte a decisão neural (SSDn), que utiliza os resultados de predição advindos de uma rede neural artificial (RNA). Esta metodologia mostrou-se bastante pertinente e aplicável na modelagem do problema de colheitas agrícolas de cana-de-açúcar já que este problema necessita de soluções semi-estruturadas. Esta metodologia combinada com a complexidade do problema em lide e atrelada à multitudine de informações fracamente correlacionadas existentes nas empresas, proporcionou uma aplicabilidade perfeita de computação inteligente e sistemas de informação.

A validação do motor neural utilizado na geração de dados preditivos para o sistema comprova mais uma vez, o uso de neurocomputação em forma de RNA na modelagem de problemas temporais. A capacidade de predição das redes MLP, quando bem treinadas com dados de históricos de safras agrícolas, continua produzindo (assim como em trabalhos anteriores) resultados com precisão acima dos 90%.

O desenvolvimento de um sistema de suporte à decisão neste cenário possibilitou a manipulação de um grande volume de dados de maneira rápida, precisa e intuitiva. Poupano aos tomadores de decisões um trabalho prolongado e minucioso. Desta maneira, o sistema produzido neste trabalho de conclusão de curso é uma ferramenta poderosa nas mãos de gerentes agrícolas experientes os quais poderão desfrutar, no momento da decisão, de informações presentes e futuras sobre os lotes das suas usinas. Estas, combinadas com parametrizações do cenário econômico desejado, tais como: tonelagem desejada, PCC mínimo e fibra mínima, são capazes de produzir sugestões de colheita bastante plausíveis, i.e. conjuntos de lotes a serem selecionados para colheita.

## 6.2 Discussão

Acredito que a utilização desta ferramenta pode determinar uma melhor qualidade da matéria-prima e, conseqüentemente, aumentar a produtividade desta atividade econômica. Certamente, esta melhoria pode aumentar os lucros nas usinas ao melhorar as decisões dos gestores. Adicionalmente, isto ainda pode reduzir o preço final dos produtos industrializados decorrentes de matéria prima agrícola, aumentando as margens de lucro que, por sua vez, podem contribuir para o desenvolvimento da atividade como um todo.

A concepção modularizada do sistema permite o desenvolvimento de extensões do modelo, utilizado neste trabalho, para gerar curvas de maturação das variedades de cana-de-açúcar desejadas em vários cenários diferentes. Isto possibilitaria o agente de decisão conhecer o momento exato para a colheita de qualquer variedade cultivada em sua unidade, inclusive considerando todos os efeitos extra-temporâneos, tais como seca, tratamentos culturais, pragas etc.

A solução ora apresentada neste trabalho, desde que treinada com dados histórico de produção local, pode ser aplicada às mais diversas regiões agrícolas do país. Isto oferece uma flexibilidade interessante e necessária para unidades de produção instaladas em mais de uma região. A receptividade da universidade dentro de setores econômicos como este - importantes e carentes em tecnologia de ponta, contribui ainda mais para o desenvolvimento do agronegócio brasileiro e firma novas possibilidades de pesquisas e aplicações. Assim sendo, a universidade poderá, cada vez mais, gerar ciência em cima de problemas reais, desenvolvendo nossa região com aplicação direta de novos conhecimentos.

## 6.3 Trabalhos Futuros

Esta foi a primeira implementação de um SSDn no domínio de colheitas agrícolas. Existem outras atividades do setor sucro-alcooleiro, onde outras possíveis abordagens e implementações podem ser exploradas. Aqui, a computação neural foi utilizada na geração de dados preditivos. O motor inteligente faz previsões que alimentam o modelo de decisão. Outra abordagem seria também a utilização de inteligência artificial, mas no modelo decisor propriamente. A busca de metas ou a otimização de variáveis do sistema poderia ser auxiliada por um sistema especialista (brevemente citado na seção 2.2) ou talvez uma RNA.

Novas modelagens ou testes em cima do problema da colheita da cana-de-açúcar podem revelar resultados ainda melhores que em [Pacheco05] – principalmente no parâmetro TCH. A utilização de outros tipos de rede, por exemplo: Elman [Elman90] ou Jordan [Jordan86] pode trazer novas perspectivas do problema, já que incorporam aspectos temporais. Ainda relacionado com modelagem e treinamentos, sugere-se algum paradigma de otimização de redes neurais para previsão de séries, como o proposto em [Barros05].

A partir da disponibilidade de novos dados históricos, o sistema poderá incorporar considerações à cerca de alguma de logística de transporte e armazenamento imediatamente após o corte (distância física entre os lotes indicados para o corte, proximidade com as unidades de moagem, entre outros).

O sistema pode também apresentar interfaces com o modelo neural, o que possibilitaria mudança nas configurações, retreinamentos, entre outros – em tempo real, sem a necessidade de paralisação do servidor.

A disponibilidade da ferramenta poderá ser aumentada, a partir de interfaces com outros dispositivos móveis (celulares ou *palmtops*).



Finalmente, a abordagem utilizada, certamente, poderá ser estendida para outras atividades do manejo da cana-de-açúcar – plantio, irrigação, adubação, entre outros. Bem como a extrapolação da aplicação para outras culturas agrícolas também importantes no país – soja, algodão, feijão, etc.

## Bibliografia

- [AlcoolBrás04] *Revista ALCOOLBrás Especial*. Editora Valete, Edição 87 de Set/Out de 2004.
- [Alexander73] ALEXANDER, A.G. *Sugarcane physiology: a comprehensive study of the Saccharum source-to-sink system*. Amsterdam: Elsevier, 1973.
- [Barros05] BARROS, Adélia C. de A. *Otimização de redes neurais para previsão de séries temporais*. Trabalho de conclusão de curso – Escola Politécnica, Departamento de Sistemas de Informação, Universidade de Pernambuco, Recife, 2005.
- [BluePrints05] *BluePrints – Patterns*. Disponível em <http://java.sun.com/blueprints/patterns/index.html>. Acesso em agosto de 2005.
- [Braga96] BRAGA, Ascensão. *A Gestão da Informação*. Tese de Mestrado Universidade da Beira Interior, com supervisão da Professora-Doutora Maria Joaquina Barrulas, 1996.
- [Buarque98] LIMA NETO, Fernando B. de. *Suporte à decisão gerencial baseado em redes neurais artificiais - nDSS*. Dissertação de Mestrado – Departamento de Informática, Universidade Federal de Pernambuco, Recife, 1998.
- [Cavaness02] CAVANESS, Chuck. *Programming Jakarta Struts*. Primeira edição. Editora O'Reilly & Associates, Inc, 2002.
- [CERN05] CERN. Disponível em <http://public.web.cern.ch/Public/Welcome.html>. Acesso em agosto de 2005.
- [CGI05] *The CGI*. Disponível em <http://hoohoo.ncsa.uiuc.edu/cgi/overview.html>. Acesso em agosto de 2005.
- [Cornell00] CORNELL, Gary e HORSTMAN, Cay S. *Core Java 2: Fundamentos*. Primeira edição. Volume 1. Editora Makron Books, 2000.
- [Cybenko89] CYBENKO, G., *Approximation by superpositions of a sigmoid function*. Mathematics of Control, Signals and Systems, 1989.
- [Deitel01] DEITEL, Harvey M. e DEITEL, Paul J. *Java: Como Programar*. Terceira edição. Editora Bookman, 2001.
- [Dreamweaver06] *O que é Dreamweaver?*. Disponível em <http://www.forumweb.com.br/artigos/artigos.php?action=file&id=367>. Acesso em maio de 2005.
- [Eclipse05] *Eclipse.org*. Disponível em <http://www.eclipse.org/>. Acesso em agosto de 2005.
- [Elman90] ELMAN, J.L. *Finding Structure in time*. Cognitive science, nº14, pages 179-211, 1990.
- [Fu-Liu04] Li M. Fu e Casey S. Fu-Liu. *Multi-class cancer subtype classification based on gene expression signatures with reliability analysis*, FEBS Letters, Volume 561, Issues 1-3, pages 186-190, March 2004.
- [Guimarães89] GUIMARÃES, Orsely. *Heráclito, o Pensador do Logos*. Universidade Federal Fluminense, Cadernos do ICHF, Rio de Janeiro, 1989.
- [Haykin94] HAYKIN, S., *Neural Networks – A Comprehensive Foundation*. Editora: Prentice-Hall International Editions, NJ, USA, 1994.

- [IEplorer05] *Microsoft Internet Explorer*. Disponível em [http://www.microsoft.com/windows/ie\\_intl/br/default.msp](http://www.microsoft.com/windows/ie_intl/br/default.msp). Acesso em dezembro de 2005.
- [Jain96] JAIN, A. K.; MAO, J.; MOHIUDDIN, K. M. *Artificial Neural Networks: A Tutorial*. IEEE Computer, pages 31-44, Março 1996.
- [Jakarta05] The Jakarta Project. Disponível em <http://jakarta.apache.org/>. Acesso em agosto de 2005.
- [Jordan86] JORDAN, M. I. *Attractor dynamics and parallelism in a connectionist sequential machine*. In Proceedings of The Eight annual Conference of The Cognitive Science Society, Amherst, Hillsdale: Erlbaum, pages 531-546, 1986.
- [JornalDaGlobo05] *Eles apostam na cana*. Disponível em <http://jg.globo.com/JGlobo/0,19125,VTJ0-2742-20041130-69060,00.html>. Acesso em Outubro de 2005.
- [Laudon00] LAUDON, Kenneth C. e LAUDON, Jane P. *Management information systems: organization and technology in the networked enterprise*. Sexta edição, editora: Prentice-Hall, New Jersey, 2000.
- [Log4j06] *Log4j*. Disponível em <http://www.developer.com/java/ent/article.php/3097221>. Acesso em maio de 2006.
- [Lucas97] LUCAS, B. M. *Information Technology for Management*. 6th ed. New York: McGraw-Hill, 1997.
- [MCP43] MCCULLOCH, W. S. and PITTS, W. *A logical calculus of the ideas immanent in nervous activity*. Bulletin of Mathematical Biophysics, 1943.
- [MinistérioAgricultura06] *Agronegócio Brasileiro: Uma Oportunidade de Investimentos*. Ministério da Agricultura, Pecuária e Abastecimento. Disponível em <http://www.agricultura.gov.br/>. Acesso em maio de 2006.
- [O'Brien01] O'BRIEN, J. A. *Sistemas de Informação e as Decisões Gerenciais na Era da Internet*. Título original: Introduction to information system. São Paulo: Saraiva, 2001.
- [Pacheco05] PACHECO, Diogo F., REGUEIRA, Filipe S. e NETO, Fernando B. de L. *Utilização de Redes Neurais Artificiais em colheitas de cana-de-açúcar para predição de PCC, TCH e fibra*. Revista Alcoolbrás, São Paulo, Março de 2005.
- [Park04] S.J. Park, C.S. Hwang and P.L.G. Vlek. *Comparison of adaptive techniques to predict crop yield response under varying soil and land management conditions*. Agricultural Systems December 2004.
- [Prechelt94] PRECHELT, L. *Proben1 – a set of neural networks benchmark problems and benchmarking rules*. Technical Report 21/94, Karlsruhe University, Germany, 1994.
- [Priberam06] *Dicionário Universal da Língua Portuguesa*. Disponível em <http://www.priberam.pt/dlpo/dlpo.aspx>. Acesso em abril de 2006.
- [Rezende00] REZENDE, Denis A., ABREU, Aline F. *Tecnologia da Informação Aplicada à Sistemas de Informação Empresariais*. São Paulo: Atlas, 2000.
- [Rezende03] REZENDE, Cátia Gontijo. *Conceitos e Perspectivas em Sistemas de Informação e de Apoio a Tomada de Decisão*, Portal do SERPRO, Tematec, ano IX – n°68, 2003. Disponível em <http://www.serpro.gov.br/publicacao/tematec/publicacao/tematec/2003/ttec68>. Acesso em Dezembro de 2005.
- [Rosenblatt61] ROSENBLATT, Frank. *Principles of neurodynamics*. Spartan Books, Washington-DC, USA, 1961.
- [Rumelhart86] RUMELHART, D. E.; HINTON, G. E. e WILLIAMS, R. J. *Learning internal representations by error propagation*. In: D.E. RUMELHART and J.L. MC-CLELLAND (Eds.): Parallel Distributed Processing Vol 1. MIT Press, Cambridge, pages 318-362, 1986.

- [Simon60] SIMON, H. A. *The new science of management decisions*, Editora Harper & ROW, Nova Iorque, 1960.
- [Sinha98] P. Sinha. *A symmetry perceiving adaptive neural network and facial image recognition*, Forensic Science International, Volume 98, Issues 1-2, pages 67-89, 1998.
- [Sommerville03] SOMMERVILLE, Ian. *Engenharia de Software*. Sexta edição. Editora: Prentice Hall Brasil, 2003.
- [Struts05] *The Apache Struts site*. Disponível em <http://struts.apache.org/>. Acesso em agosto de 2005.
- [Sun05] *Sun Developer Network*. Disponível em <http://java.sun.com/>. Acesso em agosto de 2005.
- [Sprague91] SPRAGUE, Ralph H. e WATSON, Hugh J. *Sistemas de apoio à decisão: colocando a teoria em prática*. Editora Campus, Rio de Janeiro, 1991.
- [Sysdeo06] *SYSDEO groupe sqli – eclipse Tomcat plugins*. Disponível em <http://www.sysdeo.com/eclipse/tomcatplugin>. Acesso em maio de 2006.
- [Tatibana06] *Uma introdução às redes neurais*. Disponível em <http://www.din.uem.br/ia/neurais/>. Acesso em maio de 2006.
- [Tomcat06] *Projeto Apache Tomcat*. Disponível em <http://tomcat.apache.org/>. Acesso em janeiro de 2006.
- [Werbos74] WERBOS, Paul J. *Beyond regression: new tools for predicting and analysis in the behavioral sciences*. Doctoral Dissertation, Applied Mathematics, Harvard University, USA, 1974.
- [Zaneti03] ZANETI JR., Luiz Antonio. *Sistemas de Informação Baseados na tecnologia Web: Um Estudo Sobre Seu Desenvolvimento*. Dissertação (Mestrado) – Faculdade de Economia, Administração e Contabilidade, Departamento de Administração, Universidade de São Paulo, São Paulo, 2003.

# Anexo A

## Documento de Requisitos

### InteliColheita

Cliente: Universidade de Pernambuco



**Versão 0.1** | 05/03/2006

Responsável: Diogo Pacheco

## Conteúdo

<b><u>1.</u></b>	<b><u>Introdução</u></b> .....	<b>3</b>
<b><u>1.1</u></b>	<b><u>Visão geral deste documento</u></b> .....	<b>3</b>
<b><u>1.2</u></b>	<b><u>Convenções, termos e abreviações</u></b> .....	<b>3</b>
<b><u>2.</u></b>	<b><u>Requisitos Funcionais</u></b> .....	<b>4</b>
	[RF001] Autenticação .....	4
	[RF002] Tela de Apresentação .....	4
	[RF003] Previsão dos Dados .....	4
	[RF004] Parametriação da Simulação .....	4
	[RF005] Simulação.....	4
	[RF006] Apresentação dos Resultados.....	4
	[RF007] Mapa Simulado .....	5
	[RF008] Realizar Logout.....	5
	[RF009] Geração de Logs .....	5
<b><u>3.</u></b>	<b><u>Requisitos Não Funcionais</u></b> .....	<b>6</b>
	[NF001] Tela de Processamento .....	6
	[NF002] Servidor de Aplicação.....	6
	[NF003] Navegador.....	6
<b><u>4.</u></b>	<b><u>Escopo Negativo</u></b> .....	<b>7</b>
<b><u>5.</u></b>	<b><u>Referências</u></b> .....	<b>8</b>

## 1. Introdução

Este documento especifica o sistema InteliColheita de acordo com o projeto de TCC[1] já aprovado pela banca examinadora. O sistema será desenvolvido pelo aluno Diogo Ferreira Pacheco como trabalho de conclusão de curso de Engenharia da Computação na Universidade de Pernambuco (UPE). Seu propósito é prover ao gerente agrícola uma ferramenta que o auxilie na decisão de colheita dos lotes de cana-de-açúcar em usinas sucro-alcooleira.

Este suporte será dado a partir da predição de pelo menos 3 (três) fatores de produtividade: PCC, TCH e fibra. Esses valores serão gerados a partir de uma rede neural artificial, previamente treinada com dados históricos da respectiva usina.

### 1.1 Visão geral deste documento

Esta introdução fornece as informações necessárias para fazer um bom uso deste documento, explicitando seus objetivos e as convenções que foram adotadas no texto. As demais seções apresentam a especificação do sistema InteliColheita e estão organizadas como descrito abaixo.

- **Seção 2 - Requisitos funcionais:** lista os requisitos funcionais do sistema, especificando seus objetivos e prioridades.
- **Seção 3 - Requisitos não funcionais:** especifica todos os requisitos não funcionais do sistema, divididos em requisitos de usabilidade, confiabilidade, desempenho, segurança, distribuição, adequação a padrões e requisitos de hardware e software.
- **Seção 4 - Escopo negativo:** especifica as funcionalidades que estão relacionadas com o sistema, mas que não fazem parte do escopo do projeto e, portanto, não serão implementadas.
- **Seção 5 - Referências:** contém uma lista de referências para outros documentos relacionados.

### 1.2 Convenções, termos e abreviações

Esta seção explica o conceito de alguns termos importantes que serão mencionados no decorrer deste documento. Estes termos são descritos na tabela a seguir, estando apresentados por ordem alfabética.

Termo	Descrição
Requisitos funcionais (RF)	Requisitos técnicos do software que compõe o sistema, que descrevem ações que o sistema deve estar apto a executar, ou seja, o que o sistema deve fazer.
Requisitos não funcionais (NF)	Requisitos técnicos do software que compõe o sistema, que descrevem atributos que o sistema deve possuir ou restrições sob as quais ele deve operar.
PCC	Quantidade de sacarose na cana-de-açúcar.
TCH	Tonelada de cana por hectare.
Fibra	Material seco resultante da moagem da cana.

## 2. Requisitos Funcionais

---

### **[RF001] Autenticação**

---

O sistema deverá possuir uma tela de autenticação onde o usuário digitará o login e a senha. Desta forma ele será identificado como gerente (usuário normal do sistema) ou administrador (usuário que altera as configurações internas do InteliColheita).

---

### **[RF002] Tela de Apresentação**

---

Depois de autenticado no sistema o gerente deverá acessar uma tela de apresentação a qual deverá uma pequena descrição do InteliColheita bem como lhe instruirá de algumas funcionalidades básicas.

---

### **[RF003] Previsão dos Dados**

---

O sistema deverá ser capaz de fazer previsão dos fatores PCC, TCH e fibra de cada lote. A previsão deve ser ajustada a partir de uma data selecionada pelo gerente.

---

### **[RF004] Parametriação da Simulação**

---

Antes da realização das simulações o gerente poderá ajustar os seguintes parâmetros:

- Tonelagem mínima;
- PCC mínimo;
- Fibra mínima.

Quanto mais parâmetros forem preenchidos, mais refinada será a simulação.

---

### **[RF005] Simulação**

---

Depois de feita a previsão dos dados e a parametrização da simulação o sistema deverá combinar os valores calculados de maneira a sugerir ao gerente possíveis lotes que se encaixam na configuração pré-estabelecida.

---

### **[RF006] Apresentação dos Resultados**

---

A tela de apresentação dos resultados deverá conter um resumo com as seguintes informações:

- Data prevista para colheita;
- PCC médio dos lotes a serem colhidos;
- Fibra média dos lotes a serem colhidos;
- Tonelagem liberada com o corte dos lotes sugeridos;
- Uma lista com os lotes indicados para corte.

A tela também deverá apresentar um link que exibe um mapa simulado com os lotes a serem colhidos ou não.

---

### **[RF007] Mapa Simulado**

---

O sistema deverá ser capaz de apresentar um mapa simulado com todos os lotes usados na previsão dos dados. Cada lote poderá ser representado por 2 (duas) cores - vermelho, caso não tenha sido escolhido; ou verde, caso tenha sido indicado ao corte. Os lotes deverão exibir também os respectivos PCCs médio.



---

**[RF008] Realizar Logout**

---

O sistema deverá ser capaz de invalidar a sessão do usuário através de uma opção de logout.

---

**[RF009] Geração de Logs**

---

O sistema deve gerar, armazenar e possibilitar registros de logs das simulações dos cenários de colheita..

### 3. Requisitos Não Funcionais

---

**[NF001] Tela de Processamento**

---

O sistema deverá exibir quando necessário uma tela de processamento para evitar que um usuário faça a mesma requisição várias vezes. Isto será exigido na previsão dos dados e na simulação de corte.

---

**[NF002] Servidor de Aplicação**

---

O sistema deverá ser instalado no servidor de aplicação Tomcat 5.0.x [2] ou superior.

---

**[NF003] Navegador**

---

O InteliColheita deverá rodar no navegador Internet Explorer 6.0 [3] ou superior.

#### 4. Escopo Negativo

1. O sistema não implementará a parte do administrador do sistema. Qualquer configuração será feita através de mudança de arquivos de configuração.
2. Não existe limitação de performance.

## Referências

- [1] Projeto de TCC.
- [2] Tomcat - <http://tomcat.apache.org/>
- [3] IExplorer - [http://www.microsoft.com/windows/ie\\_intl/br/default.mspx](http://www.microsoft.com/windows/ie_intl/br/default.mspx)

## Anexo B

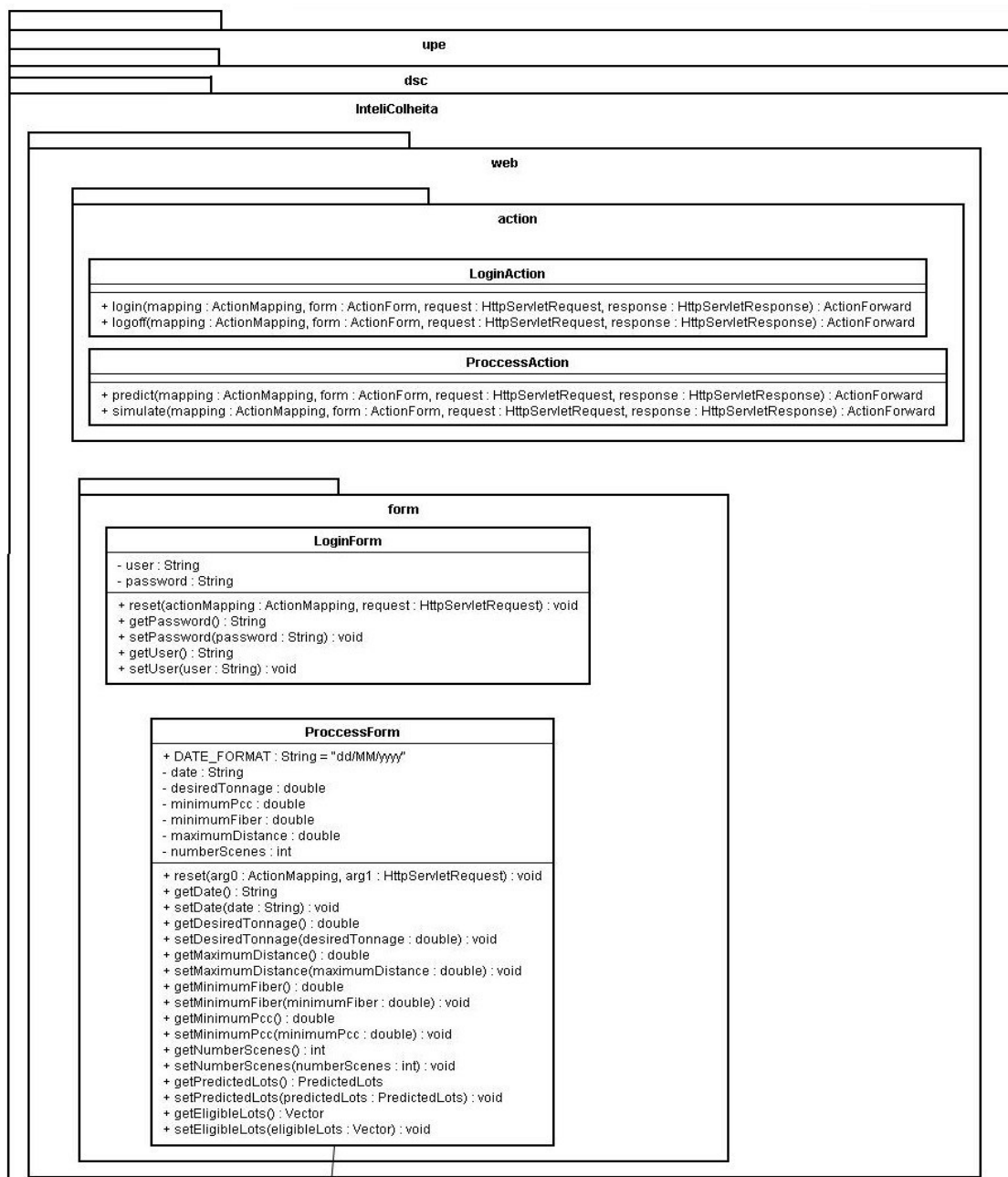


Fig. 29 – Diagrama de classes completo (parte 1/5)

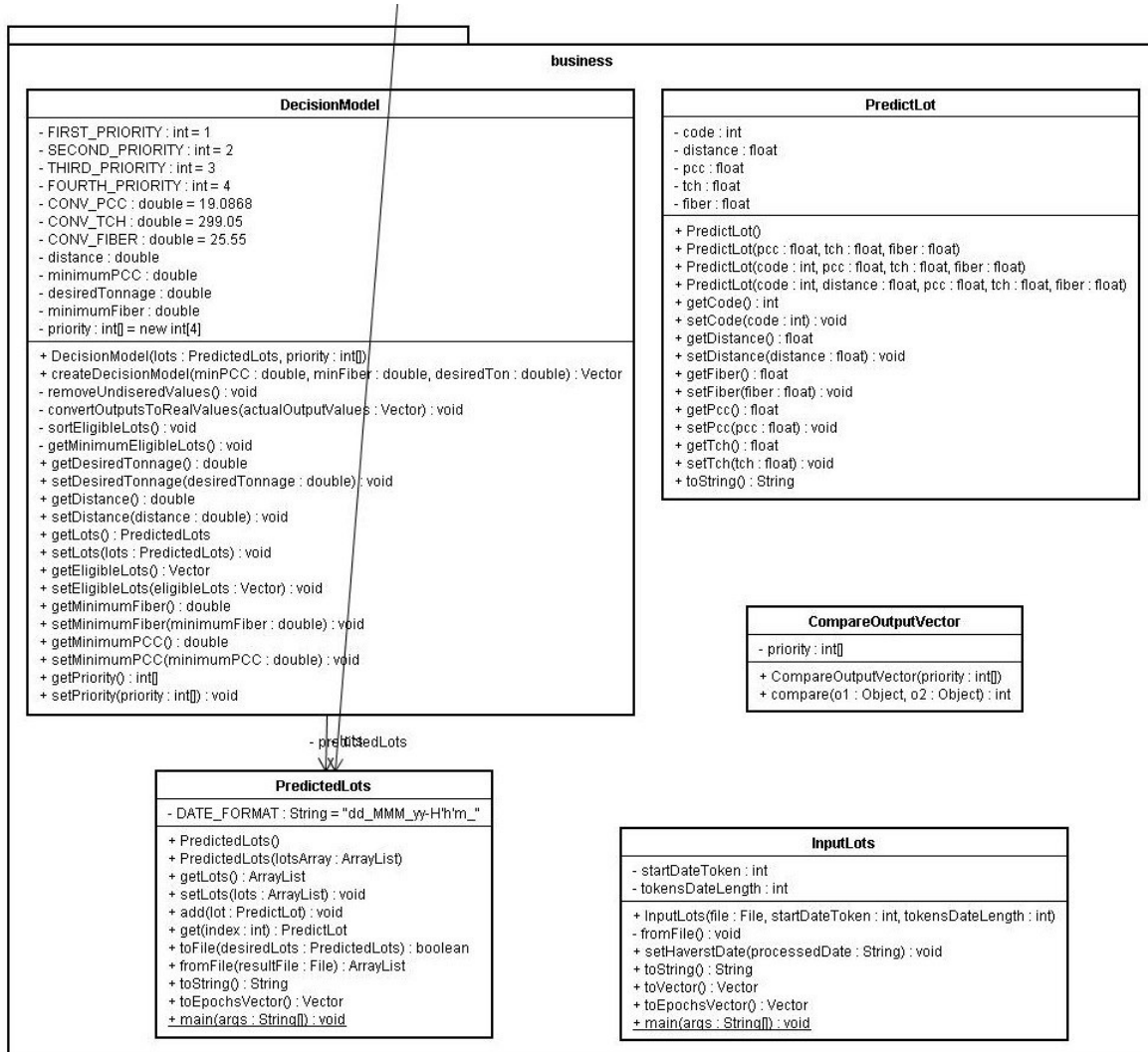


Fig. 30 – Diagrama de classes completo (parte 2/5)

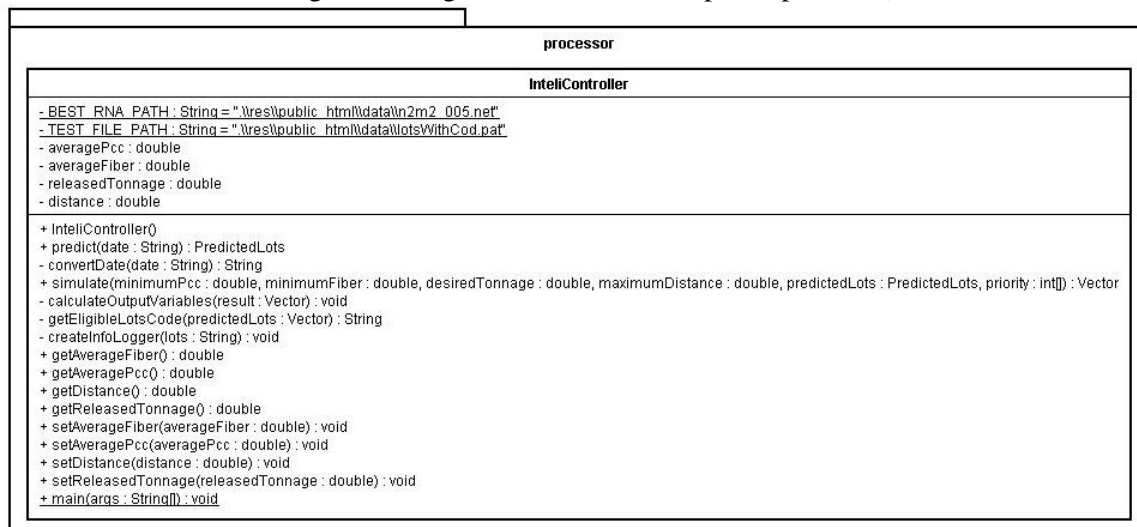


Fig. 31 – Diagrama de classes completo (parte 3/5)

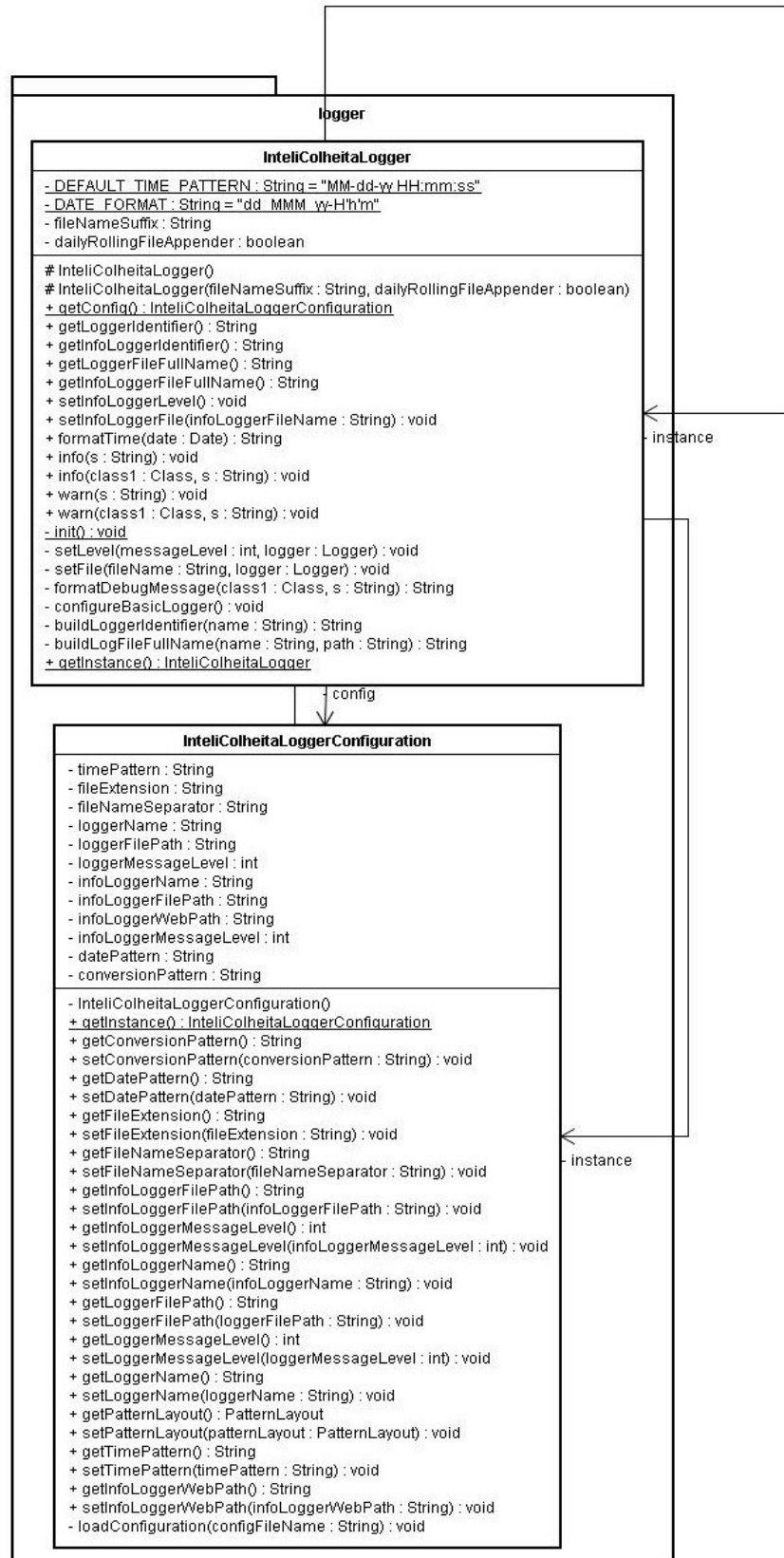


Fig. 32 – Diagrama de classes completo (parte 4/5)

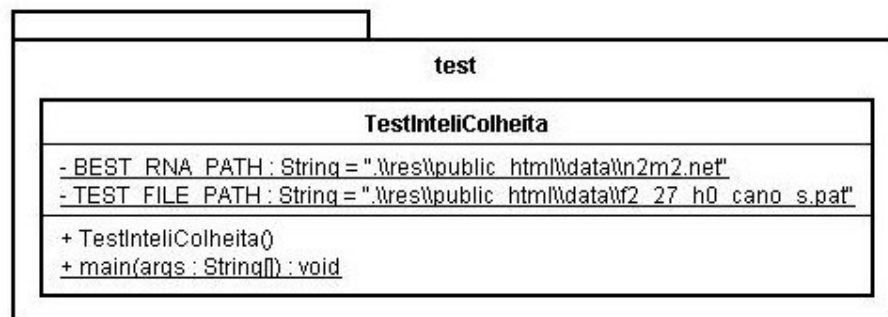
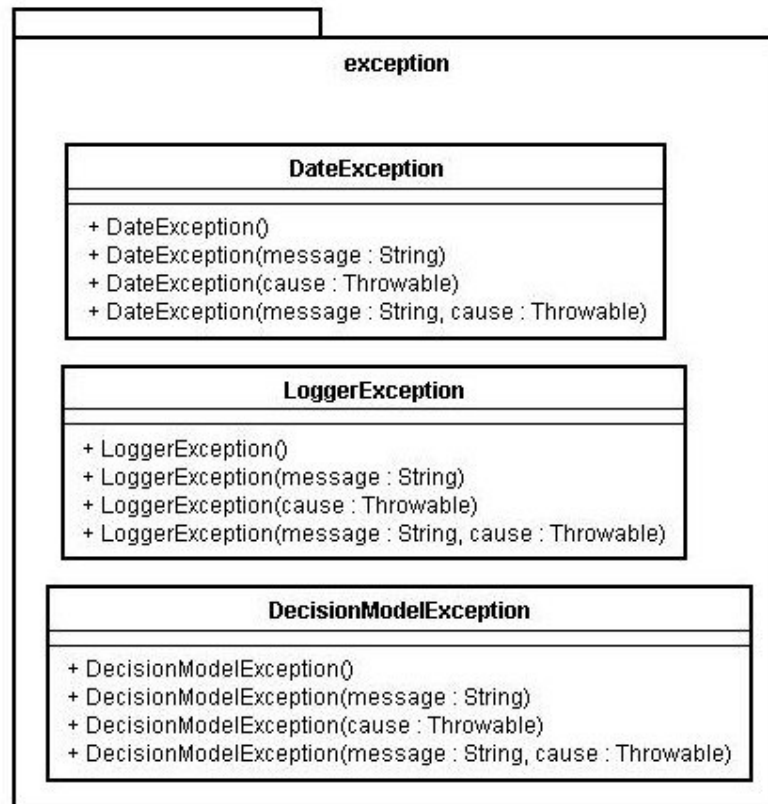


Fig. 33 – Diagrama de classes completo (parte 5/5)



# Anexo C

## Ferramentas e Tecnologias Utilizadas

- **Eclipse**

O ambiente de desenvolvimento utilizado foi o Eclipse [Eclipse05], uma ferramenta gratuita cuja funcionalidade é bastante diversificada. Abrange desde o desenvolvimento de códigos Java, mas também para visualização de base de dados, edição de arquivos XML, JSP, HTML, JavaScript, arquivos de propriedade, entre outras funcionalidades.

Essa ferramenta permite a total personalização do ambiente, de acordo com o projeto que está sendo desenvolvido, seja ele desenvolvimento para plataforma WEB ou Desktop (Standard). Além disto, permite a instalação de plugins que trabalham integrados com a ferramenta. Podemos destacar a integração com os plugins do Ant e com o do Tomcat [Sysdeo06].

- **Tomcat**

O Tomcat [Tomcat06] é um servidor de aplicações Java para web, software livre e de código aberto. Desenvolvido dentro do conceituado projeto Apache Jakarta e oficialmente endossado pela Sun como a Implementação de Referência (RI) para as tecnologias Java Servlet e JavaServer Pages (JSP). É robusto e eficiente o suficiente para ser utilizado mesmo em um ambiente de produção.

Tecnicamente, o Tomcat é um *Servlet Container*, ou seja, é um servidor onde são instalados Servlets para tratar as requisições que o servidor receber. Ele tem a capacidade de atuar também como servidor web/HTTP, ou pode funcionar integrado a um servidor web dedicado como o Apache httpd ou o Microsoft IIS.

- **Dreamweaver**

É um programa gráfico de edição de páginas da web que permite trabalhar com textos, imagens e outros elementos de páginas web em um ambiente de trabalho intuitivo. Permite aos *Webmasters* criar, editar e gerir aplicações e páginas de web de forma muito similar a de programas de publicação que são usados para a criação de *layouts* de publicações impressas. Não é necessário manipular diretamente o código html da página e conseqüentemente, ajuda a acelerar o tempo de produção de *web sites*.

É um poderoso ambiente de desenvolvimento para a construção de sites dinâmicos que fazem o uso de bancos de dados, scripts executados no cliente e tecnologia de servidor, como Asp, (*active dinamic server pages*), *ColdFusion* e JSP. Ainda suporta os

mais recentes padrões adotados na web, tais como xhtml e css. Mais informações em [Dreamweaver06].

## ▪ Struts

O *Struts* [Cavaness02][Struts05] é um *framework open source* da Jakarta [Jakarta05]. Esse *framework* utiliza as mesmas tecnologias de aplicações Java, porém organiza-se de maneira diferente. A idéia principal do Struts é separar os códigos responsáveis pelo processamento, daqueles códigos voltados para a apresentação de dados. Essa separação pode ser feita sem a utilização desse *framework*, porém fica sob responsabilidade do desenvolvedor. O *Struts* implementa a arquitetura *Model – View – Controller* (MVC – Modelo – Visualização - Controle).

A arquitetura MVC define a total separação do Modelo (objetos pertencentes à camada de negócios), da Visualização (interface com o usuário ou com outro sistema qualquer) e do Controlador, que controla o fluxo do sistema. É de responsabilidade do controlador interceptar as requisições HTTP vindas do cliente.

O *Struts* possui a classe Java denominada Action, onde está a lógica para obter cada requisição. Um objeto Action é o responsável pelo processamento dos dados que são enviados através de uma determinada requisição. Para escrever um Action é necessário herdar da classe *org.apache.struts.action.Action* e deve-se sobrescrever o método *execute*. A classe *ActionServlet* é o controle do *Struts* e deve ser configurada para interceptar todas as requisições e então direcioná-las para os Actions correspondentes. A Fig. 34 ilustra melhor essa fluxo em relacionamentos.

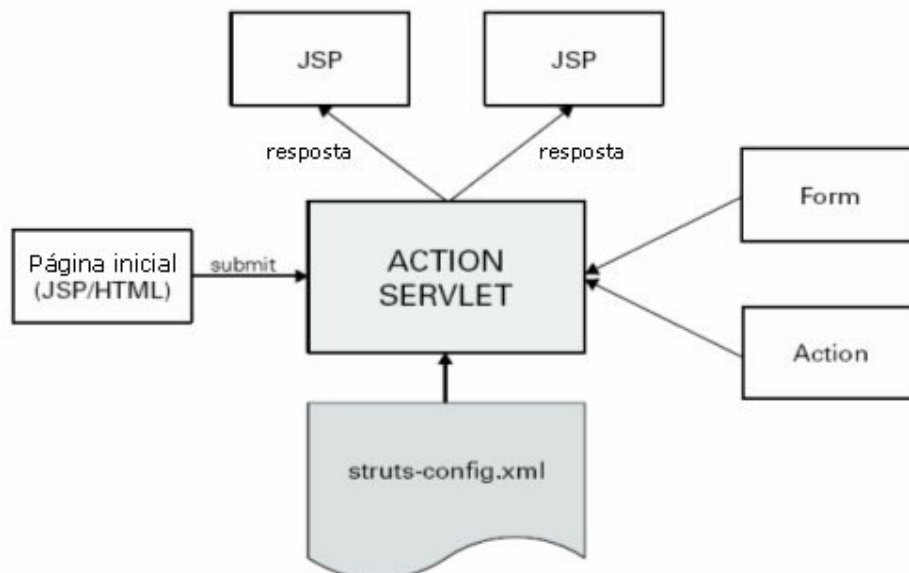


Fig. 34 – Relacionamentos em Struts

Para que o *ActionServlet* consiga fazer o direcionamento de requisições corretamente, é necessário configurar os Actions existentes em um arquivo de configuração chamado *struts-config.xml* e é este arquivo que deve conter todas as informações necessárias para o funcionamento desse *framework*.

- **Log4J**

Log4j é uma API de código aberto desenvolvida pelo projeto Jakarta da Apache. Provê um framework robusto, confiável, totalmente configurável, extensível e fácil de implementar para gerar logs de aplicações Java para diversos propósitos; por exemplo: *debugging* e monitoramento. Log4j permite que os desenvolvedores insiram declarações de log no código e as configurem fora deles.

Logicamente, log4j pode ser visto como sendo composto por três principais componentes: *logger*, *appender*, e *layout*. As funcionalidades de cada um destes componentes são acessíveis através das classes Java de mesmo nome. Usuários podem estender essas classes básicas para criar seus próprios *loggers*, *appenders*, e *layouts* [Log4j06].

## Apêndice A

Tabela 1 - Brasil: Cana de açúcar – produção, área colhida e rendimento médio – 1990 a 2004  
[MinistérioAgricultura06]

Mil toneladas e mil hectares

Ano	Brasil			Maiores Estados Produtores									
	Produção	Área colhida	Rendimento médio (t/ha)	São Paulo		Paraná		Alagoas		Minas Gerais		Pernambuco	
				Produção	Área colhida	Produção	Área colhida	Produção	Área colhida	Produção	Área colhida	Produção	Área colhida
1990	262.674	4.273	61,5	137.835	1.812	11.736	159	26.151	559	17.533	298	22.818	467
1991	260.888	4.211	62,0	136.200	1.852	12.219	172	22.214	484	17.583	276	23.505	467
1992	271.475	4.203	64,6	145.500	1.890	13.571	186	22.669	448	17.354	272	25.199	488
1993	244.531	3.864	63,3	148.647	1.896	13.694	190	12.922	323	15.743	261	14.347	363
1994	292.102	4.345	67,2	174.100	2.173	15.946	216	21.740	439	16.212	262	19.259	400
1995	303.699	4.559	66,6	174.960	2.259	20.430	256	21.573	450	16.726	268	20.665	418
1996	317.106	4.750	66,8	192.320	2.493	23.468	285	20.754	432	13.331	247	18.784	401
1997	331.613	4.814	68,9	194.025	2.446	24.564	300	24.850	450	16.262	279	20.765	421
1998	345.255	4.986	69,2	199.783	2.565	26.642	310	28.524	461	16.918	279	19.622	402
1999	333.848	4.899	68,1	197.144	2.555	27.106	338	26.860	451	17.557	280	12.253	323
2000	326.121	4.805	67,9	189.040	2.485	23.192	327	27.798	448	18.706	291	15.167	304
2001	344.293	4.958	69,4	198.932	2.567	27.424	338	28.693	456	18.975	294	15.977	339
2002	364.389	5.100	71,4	212.707	2.661	28.083	359	25.171	438	18.231	278	17.626	348
2003	396.012	5.371	73,7	227.981	2.818	31.926	374	27.221	416	20.787	303	18.522	359
2004 <sup>1</sup>	416.256	5.635	73,9	239.528	2.952	33.553	399	26.284	423	24.332	335	19.015	364

Nota: <sup>1</sup> Estimativa.

Fonte: Produção, área e rendimento médio: IBGE – Produção Agrícola Municipal (PAM - 1990 a 2003) e Levantamento Sistemático da Produção Agrícola (LSPA - maio/2005).

Elaboração: Secretaria de Política Agrícola-MAPA.

Tabela 2 - Brasil: Cana-de-açúcar – exportações, importações e principais países, produtos selecionados – 1996 a 2004 [Ministério Agricultura06]

(toneladas)

Item / destino / origem	1996	1997	1998	1999	2000	2001	2002	2003	2004
<b>Exportações</b>									
<i>Açúcar de cana, em estado bruto</i> <sup>1</sup>	4.090.398	3.844.224	4.792.247	7.826.980	4.344.076	7.089.873	7.630.323	8.353.676	9.565.748
Federação da Rússia	473.126	1.071.007	1.644.306	4.087.807	1.704.427	3.579.835	3.512.213	4.383.538	3.266.854
Índia	-	25.800	300	33.600	-	-	82.714	36.898	853.367
Egito	296.434	351.143	348.897	444.088	-	302.045	439.737	241.696	826.507
Canadá	110.085	114.464	115.050	416.023	173.062	307.712	599.400	760.428	646.986
<i>Melaço de cana</i> <sup>2</sup>	19	0	14	33	49	68	114.828	130.373	56.217
Estados Unidos	-	-	0	17	-	-	57.701	27.624	30.635
Porto Rico	-	-	-	-	-	-	32.812	30.911	25.400
Dinamarca	-	-	-	-	-	68	85	43	128
Portugal	-	-	-	-	-	-	-	23	50
<b>Importações</b>									
<i>Açúcar de cana, em estado bruto</i> <sup>1</sup>	4	1	2	2	3	2	3	0	0
Estados Unidos	0	1	2	1	1	1	1	0	0
Rep. Árabe da Síria	-	-	-	-	-	-	-	-	0
Maurício	-	-	-	-	-	-	-	-	0
Líbano	-	-	-	-	-	-	-	0	-
<i>Melaço de cana</i> <sup>2</sup>	83.730	28.849	-	12	-	6.431	1	10	-
Uruguai	-	-	-	-	-	-	-	10	-
Itália	-	-	-	-	-	-	1	0	-
Moçambique	-	-	-	-	-	6.431	-	-	-
Estados Unidos	0	-	-	12	-	-	-	-	-

Fonte: MDIC.

Notas:

<sup>1</sup> Posição 1701.11.00 da NCM.

<sup>2</sup> Posição 1703.10.00 da NCM.

Elaboração: Secretaria de Política Agrícola-MAPA.