

**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**UM ESTUDO SOBRE APLICAÇÃO DE TÉCNICAS DE  
INTELIGÊNCIA ARTIFICIAL E ENGENHARIA DE  
SOFTWARE À CONSTRUÇÃO DE UM SISTEMA DE  
SUPERVISÃO E CONTROLE**

**LETICIA TOLEDO MAIA**

**ORIENTADOR: JOSÉ CAMARGO DA COSTA**

**CO-ORIENTADOR: ALEXANDRE RICARDO SOARES ROMARIZ**

**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA**

**PUBLICAÇÃO: PPGENE.DM – 322/07**

**BRASÍLIA/DF: DEZEMBRO - 2007**

**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**UM ESTUDO SOBRE APLICAÇÃO DE TÉCNICAS DE  
INTELIGÊNCIA ARTIFICIAL E ENGENHARIA DE  
SOFTWARE À CONSTRUÇÃO DE UM SISTEMA DE  
SUPERVISÃO E CONTROLE**

**LETICIA TOLEDO MAIA**

**DISSERTAÇÃO SUBMETIDA AO DEPARTAMENTO DE  
ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA  
UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS  
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE.**

**APROVADO POR:**

---

**Prof. José Camargo da Costa,  
(Orientador)**

---

**Profa Janaina Gonçalves Guimarães  
(Examinador Interno)**

---

**Prof. Jorge Henrique Cabral Fernandes  
(Examinador Externo)**

**BRASÍLIA/DF, 19 DE DEZEMBRO DE 2007**

## **FICHA CATALOGRÁFICA**

MAIA, LETICIA TOLEDO	
SISTEMA DE SUPERVISÃO INTELIGENTE [Distrito Federal] 2007. xvii, 99p., 210 x 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2007). Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia. Departamento de Engenharia Elétrica.	
1. Automação Predial	2. Sistema inteligente
3. Sistema de Supervisão	
I. ENE/FT/UnB	II. Título (série)

## **REFERÊNCIA BIBLIOGRÁFICA**

MAIA, L. T. (2007). Sistema de Supervisão Inteligente, Publicação PPGENE.DM-322/07, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 99p.

## ***CESSÃO DE DIREITOS***

AUTOR: Leticia Toledo Maia

TÍTULO: Um Estudo Sobre Aplicação de Técnicas de Inteligência Artificial e Engenharia de Software à Construção de um Sistema de Supervisão e Controle

GRAU: Mestre

ANO: 2007

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

---

Leticia Toledo Maia  
CLN 408 Bl B EM 50 Ap 107, Asa Norte.  
70.856-500 Brasília – DF – Brasil.

## AGRADECIMENTOS

À Deus por ter me permitido trilhar este caminho, mostrando as soluções para todos os problemas e por enviar as pessoas certas para caminharem comigo.

Aos meus pais pelo amor, apoio, pelo exemplo de honestidade, coragem e confiança que sempre me dedicaram, pelo estímulo nestes estudos e por toda a minha formação.

Às minhas irmãs, Lidia e Natália, pela amizade, carinho, companheirismo e força por uma conquista.

À minha grande amiga e irmã Anna Paula pelas acolhidas, pelo carinho, respeito e amor. Pelo apoio nas horas mais difíceis, pelas farras e risadas sem as quais esse trabalho teria sido mais árduo.

Ao Professor Camargo, por acreditar neste trabalho e dedicar-se pacientemente à sua realização. Por transmitir com amor suas idéias e oferecer a todos a sua volta não só seu conhecimento mas também seu carinho e amizade tornado-se um exemplo a ser seguido.

Ao Professor Romariz pelo apoio, incentivo e sua orientação neste trabalho.

À professora Janaina pela amizade, carinho e toda ajuda desde quando cheguei na cidade.

Aos meus tios, Rosana e Roberto, obrigado pelo apoio, acolhida, amor, incentivo aos meus estudos.

Ao meu lindo e querido Zé por sua generosidade, paciência, amor e apoio. Pelo companheirismo, em tantos finais de semana no laboratório, pela força e inspiração.

Ao meu amigo e irmão Cláudio por suas contribuições preciosas, tirando minhas duvidas, pela amizade e pelo carinho.

Ao meu amigo Gilmar, que sempre me apoiou e teve muita paciência. Pelo ensinamento e toda solidariedade, principalmente ao chegar ao laboratório.

À mainha, Dona Maria, pelo amparo carinhoso durante todo esse tempo que estive em Brasília.

À Maria Regina e ao José Luiz pela afeição, cuidados, conselhos e deliciosos papos.

Ao professor Marcelo Ladeira, do Departamento de Ciência da Computação da UnB, sou profundamente grata pela luz que ele me irradiou na minha reta final.

Ao professor Jorge por suas preciosas contribuições.

Aos colegas Fernando Autran, Mauricio Panisset, Adriano Assunção e Telmo, do Tribunal de Justiça do Distrito Federal, pelo apoio e toda ajuda disponibilizada para conclusão deste trabalho.

Aos amigos do laboratório LDCI e do Bloco G e todos os outros que encontrei ao longo dos cursos, pelo convívio e pela troca de figurinhas.

E todas as pessoas que me ajudaram a chegar até aqui. Seria impossível numa única folha de papel agradecer por este momento.

"Aquele que recebe de mim uma idéia tem aumentada a sua instrução sem que eu tenha diminuído a minha. Como aquele que acende sua vela na minha recebe luz sem apagar a minha vela. Que as idéias passem livremente de uns aos outros no planeta, para a instrução moral e mútua dos homens e a melhoria de sua condição..."

(Thomas Jefferson)

## **RESUMO**

UM ESTUDO SOBRE APLICAÇÃO DE TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL E ENGENHARIA DE SOFTWARE À CONSTRUÇÃO DE UM SISTEMA DE SUPERVISÃO E CONTROLE

**Autor: Leticia Toledo Maia**

**Orientador: José Camargo da Costa**

**Programa de Pós-graduação em Engenharia Elétrica**

**Brasília, dezembro de 2007.**

Este trabalho apresenta a concepção de um sistema inteligente de supervisão (SIS) com atuação sobre sistema de automação predial. O SIS provê tomada de decisão fornecendo possíveis soluções para falhas, emergências e otimização do desempenho geral das tarefas de manutenção predial, utilizando técnicas de inteligência artificial (Sistemas Especialistas e Lógica Difusa). O SIS foi validado através da implementação de simuladores para os subsistemas de ar condicionado, iluminação e elevador. Foi utilizada uma metodologia orientada a objeto, fazendo uso dos recursos disponíveis na linguagem Java voltada à *web*.

## **ABSTRACT**

A STUDY ABOUT THE APPLICATION OF ARTIFICIAL INTELLIGENCE AND SOFTWARE ENGINEERING TECHNIQUES TO THE IMPLEMENTATION OF A SUPERVISION AND CONTROL SYSTEM

**Author: Leticia Toledo Maia**

**Supervisor: José Camargo da Costa**

**Programa de Pós-graduação em Engenharia Elétrica**

**Brasília, december 2007.**

This work presents the conception of an Intelligent Supervision System (ISS) that acts on a building automation system. The ISS provides decision support for possible failure solutions, emergencies and general optimization of building maintenance using artificial intelligence (Specialist System and Fuzzy Logic). The ISS is validated through the implementation of simulators of the air conditioning, lighting and elevator subsystems. An object oriented methodology was applied using Java language tools for web application.

# SUMÁRIO

<b>1 - INTRODUÇÃO .....</b>	<b>1</b>
1.1. OBJETIVOS .....	1
1.2. CONTEÚDO E ORGANIZAÇÃO.....	2
<b>2 - REVISÃO BIBLIOGRÁFICA .....</b>	<b>3</b>
2.1 PLANO DE DESENVOLVIMENTO DE SOFTWARE .....	3
2.1.1 Engenharia de Software .....	4
2.1.2 Modelagem Orientada a Objeto .....	4
2.1.3 Metodologia Orientada a Objeto.....	7
2.1.4 Linguagem de Programação.....	11
2.2 AUTOMAÇÃO DE PROCESSO EM SISTEMAS PREDIAIS .....	13
2.2.1 Sistema de Supervisão.....	14
2.2.1.1 Subsistema Predial de Ar Condicionado .....	18
2.2.1.2 Subsistema Predial de Iluminação .....	20
2.2.1.3 Subsistema Predial de Elevador.....	21
2.3 INTELIGÊNCIA ARTIFICIAL .....	22
2.3.1 Lógica Fuzzy.....	23
2.3.2 Sistema Especialista .....	26
<b>3 - SISTEMA SUPERVISÃO PREDIAL: APLICAÇÃO DE INTELIGÊNCIA ARTIFICIAL E ENGENHARIA DE SOFTWARE .....</b>	<b>30</b>
3.1 DESENVOLVIMENTO DO SOFTWARE DO SIS E DO SIMULADOR DO SIS .....	37
3.1.1 Software do SIS .....	38
3.1.2 Descrição do Simulador do SIS .....	44
<b>4 – APLICAÇÃO DE INTELIGÊNCIA ARTIFICIAL EM UM SUBSISTEMA PREDIAL .....</b>	<b>46</b>
4.1 SIMULADOR DO ELEVADOR.....	46
<b>5 - RESULTADOS E DISCUSSÃO .....</b>	<b>48</b>
5.1 DESENVOLVIMENTO DE UMA ARQUITETURA DE SISTEMA SUPERVISÃO .....	48
5.1.1 Validação .....	48
5.2 OTIMIZAÇÃO DO SISTEMA DO ELEVADOR.....	54
5.2.1 Validação .....	54
<b>6 - CONCLUSÕES E RECOMENDAÇÕES .....</b>	<b>59</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>61</b>
<b>APÊNDICES .....</b>	<b>70</b>



## LISTA DE FIGURAS

Figura 2.1: Iterações .....	9
Figura 2.2: Organização.....	10
Figura 2.3: Pirâmide da automação.....	14
Figura 2.4: Esquema de um sistema de ar condicionado central .....	19
Figura 2.5: Módulos de um Sistema Fuzzy .....	25
Figura 2.6: Arquitetura de um SE .....	28
Figura 3.1: Diagrama de classe da concepção de um SIS .....	35
Figura 3.2: Diagrama de atividade do acesso ao <i>software</i> do SIS.....	35
Figura 3.3: Futura implantação do SIS nas dependências do TJDF. Apenas a comarca de Brazlândia apresenta subsistemas prediais com automação .....	36
Figura 3.4: Desenvolvimento do <i>Software</i> e Simulador do SIS.....	38
Figura 3.5: Tela do <i>software</i> do SIS com tipos de usuários e permissões .....	39
Figura 3.6: Tela de saída do <i>software</i> do SIS - Menu .....	41
Figura 3.7: Tela de relatório do <i>software</i> do SIS - Menu .....	41
Figura 3.8: Exemplo de regra de produção do subsistema ar condicionado.....	42
Figura 3.9: Regras de Produção – Expert SINTA .....	42
Figura 3.10: Diagrama de classe.....	43

<b>Figura 3.11: Diagrama de caso de uso .....</b>	<b>44</b>
<b>Figura 4.1: Tela de saída do simulador.....</b>	<b>47</b>
<b>Figura 5.1: Monitoramento de equipamentos do Simulador do SIS .....</b>	<b>50</b>
<b>Figura 5.2: Menu Monitor do Simulador do SIS.....</b>	<b>50</b>
<b>Figura 5.3: Listagem dos possíveis sintomas de um determinado equipamento .....</b>	<b>50</b>
<b>Figura 5.4: Exemplo de regra estabelecida no SE (Expert SINTA).....</b>	<b>50</b>
<b>Figura 5.5: Características estabelecida pelas Regras de Produção .....</b>	<b>50</b>
<b>Figura 5.6: Resultado de uma consulta ao (a) Expert SINTA e (b) Simulador do SIS</b>	<b>52</b>
<b>Figura 5.7: Características estabelecidas pelas Regras de Produção - Erro .....</b>	<b>53</b>
<b>Figura 5.6: Resultado de uma consulta ao <i>software</i> do SIS - Erro .....</b>	<b>53</b>
<b>Figura 5.9: Funções de Pertinência: (a) número de passageiros; (b) tempo de viagem; (c) distância da viagem; e (d) tempo de espera. ....</b>	<b>55</b>
<b>Figura 5.10: Regras <i>Fuzzy</i> .....</b>	<b>55</b>

## LISTA DE TABELAS

<b>Tabela 2.1: Comparação entre enfoque estruturado e orientado a objetos .....</b>	<b>5</b>
<b>Tabela 2.2: Comparação entre diversas linguagens de programação [25].....</b>	<b>13</b>
<b>Tabela 3.1: Descrição dos subsistemas.....</b>	<b>31</b>
<b>Tabela 3.2: Características da linguagem Java.....</b>	<b>34</b>
<b>Tabela 3.3: Tipos de permissões do <i>software</i> do SIS.....</b>	<b>39</b>
<b>Tabela 5.1: Três elevadores e dez andares .....</b>	<b>56</b>
<b>Tabela 5.2: Três elevadores e quinze andares .....</b>	<b>57</b>
<b>Tabela 5.3: Três elevadores e trinta andares. ....</b>	<b>57</b>

## LISTA DE SÍMBOLOS, NOMENCLATURA E ABREVIACÕES

API	<i>Application Programming Interface</i>
BMS	<i>Building Management System</i>
HTML	<i>Hyper Text Markup Language</i>
http	<i>HyperText Transfer Protocol</i>
HVAC	<i>Heating, Ventilation and Air Conditioning</i>
IA	Inteligência Artificial
IAS	<i>Industrial Automation System</i>
IDE	<i>Integrated Development Environment</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IHM	Interfaces Homem-Máquina
JIT	<i>Just InTime</i>
JSP	<i>Java Server Pages</i>
JVM	<i>Java Virtual Machine</i>
LHS	<i>Left-Handed Side</i>
OLE	<i>Object Linking and Embedding</i>
OMG	<i>Object Management Group</i>
OMT	<i>Object Modeling Technique</i>
OO	Orientação a Objetos
OOSE	<i>Object-Oriented Software Engineering</i>
OPC	<i>OLE for Process Control</i>
PLC	<i>Power Line Communications</i>
RHS	<i>Right-Handed Side</i>
RUP	<i>Rational Unified Process</i>
TJDF	Tribunal de Justiça do Distrito Federal
SCADA	<i>Supervisory Control and Data Acquisition</i>
SE	Sistema Especialista
SGBD	Sistema de Gerenciamento de Banco de Dados
SIS	Sistema Inteligente de Supervisão
SQL	<i>Structured Query Language</i>
UML	<i>Unified Modeling Language</i>

# 1 - INTRODUÇÃO

A construção de edifícios inteligentes é uma tendência mundial que vem sendo incorporada mais recentemente no país, em função das crescentes preocupações ambientais e dos ganhos em segurança e conforto para os usuários. A automação tem sido aplicada no controle de diversos subsistemas destes edifícios, tais como iluminação, climatização e operação de elevadores.

Os edifícios projetados e construídos com automação possuem algum tipo de monitoramento, supervisão e controle dos subsistemas que o compõem. E são chamados de sistemas de supervisão. Estes possuem *softwares* que monitoram o estado dos equipamentos permitindo a visualização e a sua operação através de interfaces gráficas.

Os sistemas de supervisão desenvolvidos com tecnologia nacional são escassos e não apresentam protocolos abertos que permitam a interligação entre os serviços. Por isso, existem amplas possibilidades de desenvolvimento na construção e programação destes sistemas. E o ambiente para validação é o Tribunal de Justiça do Distrito Federal (TJDF), que possui automação em dois subsistemas que são o ar condicionado e a iluminação, e adotariam uma estratégia de reforma: a implantação do conceito de automação a outros subsistemas e sua integração.

E visando essa estratégia optou-se por aplicação de inteligência artificial que apresenta-se como uma área promissora para aumento do desempenho dos sistemas supervisórios. Por isso, foi desenvolvido um simulador de um sistema predial de supervisão inteligente visando a detecção de falhas de equipamentos.

## 1.1. OBJETIVOS

O principal objetivo deste trabalho foi oferecer uma contribuição ao gerenciamento inteligente e ao comando de supervisão e atuação que se inserem na estrutura de automação existente e(ou) prevista para uma edificação do Tribunal de Justiça do Distrito Federal (TJDF), incluindo a implementação física de um demonstrador conceitual. Entretanto devido às freqüentes alterações nos projetos de reforma do TJDF ficou inviável a realização deste estudo no tempo disponível para uma dissertação de mestrado. Em

função disso, decidiu-se validar a proposta de sistema inteligente de supervisão usando recursos de simulação. A estratégia de supervisão a ser implementada compreende os seguintes aspectos: obtenção e identificação de falhas e por consequência agilidade na solução dos problemas; a gestão predial mais racional e econômica; e provimento de uma supervisão integrada - equipamentos e serviços - nas áreas operacionais (processos de automação).

## **1.2. CONTEÚDO E ORGANIZAÇÃO**

Este trabalho está dividido em cinco capítulos. No capítulo 2, serão expostos conceitos de desenvolvimento de um software, de automação predial e de inteligência artificial. No capítulo 3 será detalhado o sistema inteligente de supervisão, como sua construção e simulação. No capítulo 4, serão relatados os procedimentos de simulação e os resultados. Por fim, no capítulo 5, serão apresentadas as conclusões e recomendações para trabalhos futuros.

## 2 - REVISÃO BIBLIOGRÁFICA

Neste capítulo são apresentados conceitos fundamentais para a elaboração de planos de desenvolvimento de software, bem como conceitos de automação de processos em sistemas prediais e de inteligência artificial utilizados na elaboração deste trabalho.

### 2.1 PLANO DE DESENVOLVIMENTO DE SOFTWARE

O processo de desenvolvimento de um *software* tem como objetivo a criação de um *software* que seja confiável e funcione eficientemente em máquinas reais. O *software* deve utilizar: métodos, ferramentas e procedimentos para o entendimento de como é possível projetar, implementar, avaliar e melhorar processos de seu desenvolvimento.

Os métodos estão relacionados em “como fazer”. Deve haver planejamento, estimativa do projeto, arquitetura, análise dos requisitos, testes e possível manutenção.

As ferramentas tornam possível a automação dos métodos. Adicionalmente, as ferramentas podem dar suporte ao desenvolvimento de um *software*.

Os procedimentos definem as seqüências de métodos, ferramentas a serem utilizadas e a qualidade das atividades a serem desenvolvidas.

No método de desenvolvimento de um *software*, um conjunto de etapas deve ser definido, denominado de Paradigmas da Engenharia de *Software*, também conhecido como Modelos de Ciclo de Vida. Independentemente do paradigma utilizado, três fases genéricas dividem o processo de desenvolvimento [1]:

- Definição: análise do sistema, planejamento do projeto de *software* e análise de requisitos (focaliza-se o "o quê");
- Desenvolvimento: projeto de *software*, codificação e realização de testes do *software* (focaliza-se o "como");
- Manutenção: correção, adaptação e melhoramento funcional (concentra-se nas "mudanças").

A seguir são apresentados conceitos básicos e necessários para o desenvolvimento de *software*.

### **2.1.1 Engenharia de Software**

Engenharia de *software* é um procedimento para a produção organizada que utiliza uma coleção de técnicas predefinidas e convenções notacionais. Uma metodologia costuma ser apresentada por uma série de etapas com técnicas e notações associadas a cada uma delas [2].

Existem conjuntos estruturados de práticas que resultam em aumento da qualidade e da produtividade no processo de desenvolvimento de *software*. Os mais conhecidos são os modelos de ciclos de vida em Cascata, Iterativo e Incremental, e Espiral, que apresentam as seguintes características:

- Cascata: define-se como seqüência unidirecional de atividades que transforma requisitos em um sistema. É estruturado de forma a parecer uma cascata de fases onde o final de uma fase implica no início de outra. Os resultados de cada uma destas fases são produtos intermediários que se agregam ao final do processo gerando um produto final [3];
- Iterativo e Incremental: um modelo incremental envolve a integração contínua da arquitetura do sistema para a produção de novas versões. E o modelo iterativo envolve o gerenciamento de uma seqüência de versões executáveis [1];
- Espiral: Também conhecido como modelo de Boehm, é um processo de modo cíclico e incremental.. Aprimora o grau de definição e implementação de um sistema, enquanto diminui o grau de risco do projeto [3].

Nas técnicas atuais de engenharia de *software* uma das ênfases está na encapsulação e reutilização de componentes. Reutilizar módulos prontos e testados diminui consideravelmente o tempo e custo na composição de um *software* [4]. A modelagem orientada a objeto permite a aplicação destes conceitos.

### **2.1.2 Modelagem Orientada a Objeto**

A programação orientada a objetos surgiu no final da década de 60, quando dois cientistas dinamarqueses criaram a linguagem *Simulation Language*. Em 1967, foi criada a



linguagem de programação Simula-67, envolvendo conceitos de classe e herança. Já o termo programação orientada a objetos é introduzido com a Linguagem *Smalltalk* em 1983 [5].

A idéia do desenvolvimento orientado a objetos (OO) é a de aproximar o mundo computacional do real. Nessa análise orientada a objetos há uma mudança no enfoque das funções para os dados. É necessário modelar de forma mais detalhada o sistema que será desenvolvido.

A OO é uma maneira de pensar os problemas utilizando modelos organizados a partir de conceitos do mundo real [2]. Os objetos aparecem como uma única entidade que combina a representação da informação (estrutura de dados) e sua manipulação (procedimentos). A Tabela 2.1 apresenta uma comparação entre o enfoque de modelagem estruturada e o enfoque orientado a objetos.

Tabela 2.1: Comparação entre enfoque estruturado e orientado a objetos

<b>Técnica</b>	<b>Ênfase</b>	<b>Iniciação</b>	<b>Foco</b>
Enfoque da Linguagem Estruturada	Com comunicação entre procedimentos passagem de dados; Processos e Dados.	<i>Top-Down</i> (Decomposição por Funcional).	Sistema
Enfoque da Linguagem Orientada a Objeto	Dados e procedimentos sendo fragmentos de um só elemento básico: objeto. E esses comunicam-se entre si, caracterizando a execução do programa; Dados, Controles e Processos	Definição de Objetos; Objeto sendo uma variável onde são armazenados dados e possui atributos (dados) e comportamentos a(métodos, procedimentos, funções, que atuam sobre eele).	Objeto

A OO trouxe vários novos conceitos ao desenvolvimento de *software*, descritos a seguir:

- Abstração: consiste em ignorar aspectos irrelevantes e concentrar-se nos principais. [6];

- Encapsulamento: consiste na separação dos aspectos externos de um objeto, acessíveis por outros objetos, de seus detalhes internos de implementação, que ficam ocultos aos demais [2];
- Objeto: elemento computacional que representa, no domínio da solução, alguma entidade (abstrata ou concreta) do domínio de interesse do problema sob análise [7];
- Classe: descreve um grupo de objetos com propriedades semelhantes (atributos), mesmo comportamento (operações) e relacionamento com outros objetos e mesma semântica [2];
- Atributo: conjunto de propriedades da classe e para cada uma delas especifica-se um nome, um tipo, um valor e sua visibilidade (privado, público ou protegido) [7];
- Método: conjunto de funcionalidades da classe e para cada método, especifica-se sua assinatura, composta por nome, tipo, lista de argumento e visibilidade [7];
- Mensagem: para solicitar um serviço um objeto (cliente) envia uma mensagem a outro [6];
- Herança: mecanismo existente que permite a reutilização da estrutura e do comportamento de uma classe ao se definir novas classes [8];
- Polimorfismo: habilidade de variáveis terem “mais de um tipo”. Funções são ditas polimórficas, quando seus operandos podem ter mais de um tipo [8].

Uma das mais importantes etapas de desenvolvimento de um *software* se encontra no desenvolvimento de dados. A escolha por uma modelagem orientada a objetos apresenta vantagens como confiabilidade, facilidade de interoperabilidade, reusabilidade, adaptabilidade e extensibilidade [1] [5]. Esses conceitos de engenharia de *software* são usuais na modelagem OO durante o ciclo de análise, concepção e implementação.

É nesse contexto de mudança do paradigma tradicional para o orientado a objetos que está inserida a Linguagem Unificada de Modelagem (UML – *Unified Modeling Language*), padrão de fato adotado pelo Grupo de Gerenciamento de Objetos (OMG – *Object Management Group*) em janeiro de 1997, em resposta à sua própria solicitação de propostas para uma linguagem-padrão de modelagem [9].

A UML é, na verdade, resultado da unificação de notação empregada em três metodologia OO bastante utilizadas na década de 90: Booch (de Grady Booch da *Rational*

*Software Corporation*), OOSE - *Object-Oriented Software Engineering* (de Ivar Jacobson da *Objectory*) e OMT - *Object Modeling Technique* (de James Rumbaugh da *General Electric*) [9] [10]. Posteriormente, foi desenvolvida uma metodologia orientada a objetos chamada *Rational Unified Process* (RUP - Processo Unificado da Rational), que abrange todo o processo de desenvolvimento de um *software*.

A UML foi projetada para o desenvolvimento de sistemas OO, mas é independente da linguagem de programação a utilizar. Ela permite especificar, visualizar e documentar artefatos e assim construir uma estrutura consistente de um sistema *software*. Além disso, através dessa linguagem é possível modelar elementos, relacionamentos e diagramas.

Os elementos especificados com UML podem ser estruturais, como classes, interfaces, comportamentais, grupos de elementos dentre outros. Os relacionamentos são a relação entre os blocos de elementos e podem ser subdivididos em dependência, associação, generalização e realização. Os diagramas são as representações gráficas dos elementos e sua relação. São disponibilizados nove diagramas: de classes, de objetos, de caso de uso, de seqüência, de colaboração, de gráficos de estados, de atividades, de componentes, de implantação [10].

### **2.1.3 Metodologia Orientada a Objeto**

O RUP é uma metodologia de engenharia de *software* desenvolvido pela *Rational Software Corporation*<sup>1</sup>, cujas principais características são um desenvolvimento iterativo e incremental, orientado a objetos, com foco na criação de uma arquitetura robusta, análise de riscos e utilização de casos de uso para o desenvolvimento [11] [12] [13]. Ele oferece uma abordagem baseada em disciplinas para atribuir tarefas e responsabilidades dentro de uma organização de desenvolvimento. Sua meta é garantir a produção de *software* de alta qualidade que atenda às necessidades dos usuários dentro de um cronograma e de um orçamento previsível [14].

Atualmente, RUP é uma metodologia para engenharia de *software* em evidência no mercado, principalmente no que tange às grandes empresas desenvolvedoras. Entretanto,

---

<sup>1</sup> <http://www-306.ibm.com/software/rational/>

por ser um processo longo e complexo, nem sempre é ágil para pequenas empresas e pequenas implementações [15].

Um grande problema nos projetos atuais é o grande dinamismo e complexidade dos negócios. Cada vez mais os sistemas são complexos e precisam estar prontos em menos tempo. Visando reduzir o risco e tornar o desenvolvimento mais eficiente, o RUP adota as seguintes premissas básicas: uso de iterações para evitar o impacto de mudanças no projeto; gerenciamento de mudanças e abordagens dos pontos de maior risco o mais cedo possível [16] [17].

Analisando os aspectos gerais do RUP pode-se considerar aspectos dinâmicos e estáticos descritos a seguir.

a) Aspectos Dinâmicos – Ciclos, Fase e Iteração

O RUP é composto por quatro fases: concepção (ênfase no escopo do sistema), elaboração (ênfase na arquitetura), construção (ênfase no desenvolvimento) e transição (ênfase na implantação).

Cada fase é concluída com um marco (*milestone*) bem definido, no qual os resultados são comparados com os objetivos principais, onde deverão ser tomadas algumas decisões sobre a continuidade do projeto [18].

A Iteração é um ciclo completo de desenvolvimento e finaliza com uma versão (*release*) de um produto executável, que é uma parte incrementada no produto final em desenvolvimento [19].

### **Concepção**

Nessa fase são estabelecidos o escopo e viabilidade econômica do projeto. É na concepção inicial do sistema que se faz uma discussão sobre o problema, definição do escopo do projeto e estimativa de recursos necessários para a execução do projeto, entre outros. É nessa fase que se apresenta o plano de projeto, caso de uso inicial e o glossário, entre outros [16].

## Elaboração

O propósito é analisar o domínio do problema, desenvolver o plano de projeto, estabelecer uma arquitetura estável a partir da qual o sistema poderá evoluir e eliminar os elementos de alto risco [7] [16]. Esta fase é considerada como uma das mais críticas, pois ao final da elaboração do projeto podem ocorrer mudanças que representam aumentos de custos.

## Construção

É a fase de desenvolvimento do produto até que ele esteja pronto para a versão de testes. Compreende a modelagem e o desenvolvimento em si, no qual o *software* será efetivamente programado, utilizando notação definida pela UML [16].

## Transição

A partir dessa fase, o sistema já está pronto. Inicia-se, assim, a sua implantação, compreendendo desde o período de ajustes até a transferência de propriedade do sistema.

Em cada uma das fases que compõem o aspecto dinâmico da metodologia OO existem iterações, conforme apresentado na Figura 2.1.

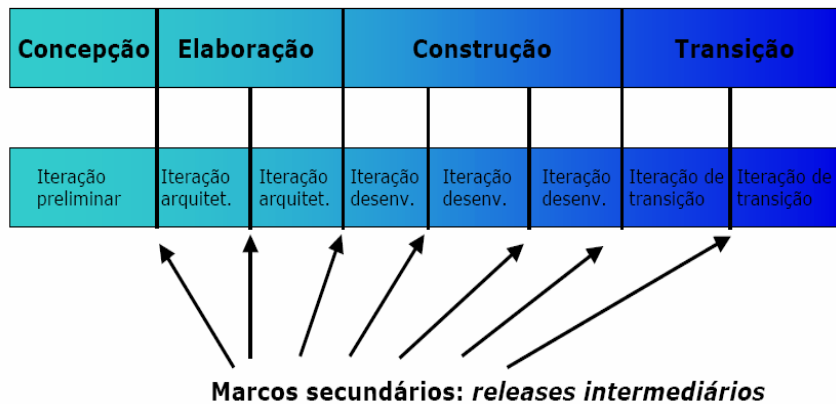


Figura 2.1: Iterações [13]

Várias iterações podem ocorrer em uma fase. Cada uma destas, por sua vez, está organizada em *workflows* (tarefas), que descrevem o que deve ser feito em termos de atividades, responsáveis e artefatos, como mostra a Figura 2.2. O RUP fornece modelos

(*templates*) para cada artefato e diretrizes (*guidelines*) para a execução de suas atividades [13].

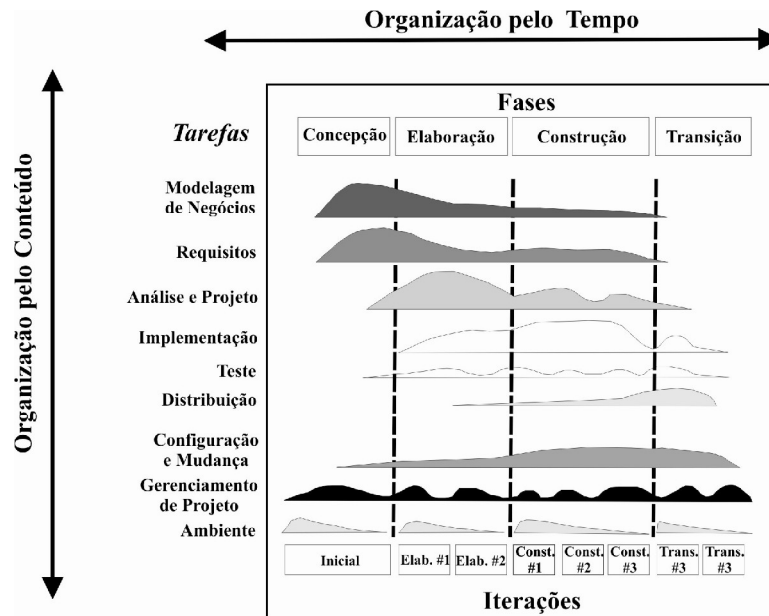


Figura 2.2: Organização [ 1]

b) Aspectos Estáticos – Tarefas, Atividades e Artefatos

O aspecto estático do RUP inclui os *workflows*, as atividades realizadas e os artefatos produzidos. Cada fluxo do projeto determina um conjunto de atividades que serão realizadas por trabalhadores e resultarão em artefatos.

Uma atividade descreve as tarefas executadas pelos trabalhadores para criar ou modificar artefatos. Um artefato é algum documento, relatório, modelo ou código que é produzido, manipulado ou consumido. Um trabalhador é alguém que desempenha um papel e é responsável pela realização de atividades [19].

Os *workflows* são divididos em [18]:

- Modelagem de Negócios (*Business Modeling*): envolve o entendimento da estrutura e dinâmica da organização cliente, garantindo que clientes, usuários e desenvolvedores tenham a mesma visão da organização para a qual será feito o desenvolvimento;
- Requisitos (*Requirements*): envolve a definição dos requisitos do sistema e de como gerenciar escopo e mudanças de requisitos;

- *Análise e Projeto (Analysis and Design)*: envolve a tradução dos requisitos numa especificação que descreve como implementar o sistema. A UML é utilizada para modelar o sistema;
- *Implementação (Implementation)*: envolve o desenvolvimento de código: classes, objetos, teste de unidades e integração de subsistemas;
- *Teste (Test)*: envolve a verificação do sistema como um todo com testes de integração e conformidade com os requisitos especificados;
- *Distribuição (Deployment)*: envolve o empacotamento, distribuição, instalação e treinamento de usuários, assim como o planejamento e condução de beta-testes;
- *Gerenciamento de Projeto (Project Management)*: envolve o gerenciamento de riscos, planejamento e acompanhamento do projeto;
- *Ambiente (Environment)*: envolve a organização do ambiente de trabalho para a equipe do projeto e a configuração do RUP para o projeto;
- *Configuração e Mudanças (Configuration and Change Management)*: envolve o gerenciamento dos artefatos gerados durante o desenvolvimento.

#### 2.1.4 Linguagem de Programação

Os microprocessadores têm um impacto significativo em dispositivos inteligentes eletrônicos voltados para o consumidor. Por isso, a *Sun Microsystems*<sup>2</sup>, em 1991, financiou um projeto de pesquisa corporativa interna com o codinome *Green*, que resultou no desenvolvimento de uma linguagem baseada em C++, posteriormente apresentada em 1995, como Java. Esta linguagem despertou enorme interesse na área da *World Wide Web* pelo seu conteúdo dinâmico e interativo. Atualmente é utilizada para desenvolver aplicativos corporativos de grande porte e aprimorar a funcionalidade de servidores *Web* [20] [21] [22].

Java foi iniciado com o objetivo de fornecer um produto simples e portátil para diferentes plataformas e sistemas operacionais. Os programas assim podem ser executados em qualquer computador que tenha instalado a Máquina Virtual Java (JVM – *Java Virtual Machine*), sendo código-fonte ou código binário.

---

<sup>2</sup> <http://www.sunmicrosystems.com.br/>

A linguagem Java exibe importantes características que, em conjunto, diferenciam-na de outras linguagens de programação [23] [24]:

- Orientada a Objetos: uma linguagem puramente orientada a objetos, pois, com exceção de seus tipos primitivos de dados, tudo em Java são classes ou instância de uma classe;
- Independente de Plataforma: são compilados para uma forma intermediária de código denominada *bytecodes* que utiliza instruções e tipos primitivos de tamanho fixo, ordenação *big-endian* e uma biblioteca de classes padronizada. Os *bytecodes* são como uma linguagem de máquina destinada a uma única plataforma, JVM, um interpretador de *bytecodes*;
- Sem ponteiro: não permite a manipulação direta de endereços de memória, tampouco exige que os objetos criados sejam destruídos, eliminando os programadores de uma tarefa complexa. Além disso, a JVM possui um mecanismo automático de gerenciamento conhecido como *garbage collector*, que recupera a memória alocada para objetos não mais referenciados pelo programa;
- Desempenho: como uma linguagem interpretada, o desempenho é razoável, não podendo ser comparada à velocidade de execução de código nativo. Para superar essa limitação, várias JVM dispõem de compiladores *just in time* (JIT) que compilam os *bytecodes* para código nativo durante a execução, otimizando-a e melhorando significativamente o desempenho de programas Java;
- Segurança: considerando a possibilidade de aplicações obtidas através de uma rede, a linguagem Java possui mecanismos de segurança que podem, no caso de *applets*, evitar qualquer operação no sistema de arquivos da máquina-alvo, minimizando problemas de segurança. Adicionalmente, com a coleta automática de lixo, evita erros comuns que os programadores cometem, quando são obrigados a gerenciar diretamente a memória;
- A Linguagem *Multithreading*: oferece recursos para o desenvolvimento de aplicações capazes de executar múltiplas rotinas concorrentemente, bem como dispõe de elementos para a sincronização dessas várias rotinas. Cada um desses fluxos de execução é o que se denomina *thread*, um importante recurso de programação de aplicações mais sofisticadas.

Java é uma linguagem bastante robusta, oferece tipos inteiros e ponto flutuante compatíveis com as especificações do IEEE (*Institute of Electrical and Electronics*



*Engineers*), suporte para caracteres *unicode*, é extensível dinamicamente, além de ser naturalmente voltada para o desenvolvimento de aplicações em rede ou aplicações distribuídas. A Tabela 2.2 mostra as vantagens do Java em relação a outras linguagens.

Tabela 2.2: Comparação entre diversas linguagens de programação [25]

	<b>Java</b>	<b>C</b>	<b>ASP</b>	<b>PHP</b>
<b>Sintaxe</b>	Semelhante a C	C	VB Script	Semelhante a C
<b>Simplicidade</b>	Intermediário	Difícil	Fácil	Fácil
<b>Tamanho</b>	Muito pequeno	Pequeno	Pequeno	Pequeno
<b>Suporte a orientação a objetos</b>	Sim	Não	Não	Sim
<b>Portabilidade</b>	Sim	Sim*	Sim*	Sim
<b>Suporte a redes (TCP/IP, Sockets, etc)</b>	Sim	Sim	Não	Sim
<b>Robusto</b>	Sim	Não	Sim	Sim
<b>Segurança</b>	Sim	Sim	Sim	Sim
<b>Interpretado</b>	Interpretado/ Compilado	Compilado	Interpretado	Interpretado
<b>Performance</b>	Muito boa	Ótima	Mediana	Mediana
<b>Multithreads</b>	Sim	Sim	Não	Sim
<b>Suporte a comunicação Serial</b>	Sim	Sim	Não	Não

\*Dependendo da plataforma pode requerer algumas mudanças no código.

## 2.2 AUTOMAÇÃO DE PROCESSO EM SISTEMAS PREDIAIS

A necessidade de conforto, padrões de qualidade, redução de custos de operação, confiabilidade, entre um extenso conjuntos de funções, estão cada vez mais presentes nas edificações. Por isso, a automação, que nasceu na indústria, acabou alcançando espaços em prédios e hoje chega a residências.

A automação envolve todas as atividades de transformação de trabalho natural cumpridos pelo homem em tarefas executadas por sistemas computacionais, visando o aumento de produtividade, eficiência, segurança e redução de custos.

Para melhor representar uma arquitetura de um sistema de automação, pode-se dividi-lo em níveis, conforme mostra a figura 2.3. No primeiro nível, encontram-se os sensores, atuadores e os controladores lógicos programáveis, como sensores de nível, pressão, temperatura, válvulas, inversores entre outros componentes. No nível seguinte,

encontram-se os sistemas que possuem algum tipo de supervisão associada ao primeiro nível e as Interfaces Homem Máquina (IHM). No Nível 3 há controle do processo produtivo da planta, constituído por banco de dados e relatórios. O nível 4 é constituído pela programação e planeamento da produção realizando o controle e a logística dos suprimentos. No nível mais alto da pirâmide encontra-se a administração dos recursos da empresa Também possui softwares para gestão de vendas, por exemplo, e o gerenciamento de todo o sistema envolvido.

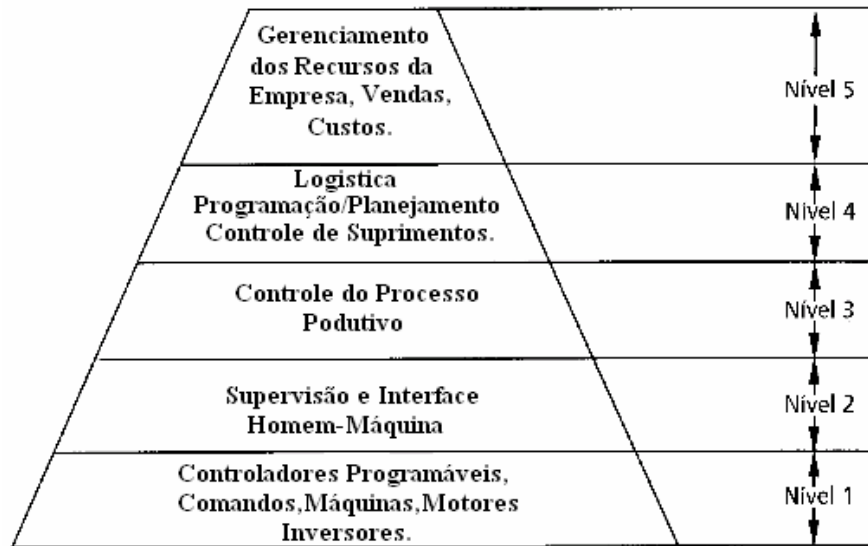


Figura 2.3: Pirâmide da automação [24]

Inicialmente a automação predial era restrita ao controle dos subsistemas de ar condicionado, elétrico e hidráulico. Ao longo do tempo foi incorporando outros, como detecção e alarme de incêndio e segurança. Com o grande desenvolvimento na área de computação, a automação predial passa a integrar as áreas de telecomunicações e informática para melhores aplicações nas atividades cotidianas e voltadas para a internet.

Para o gerenciamento dos diversos subsistemas prediais foram desenvolvidos sistemas de supervisão.

### 2.2.1 Sistema de Supervisão

O sistema predial permite, através da integração, a monitoração e controle dos subsistemas que compõem um edifício.

O desenvolvimento de um sistema de supervisão pode ser caracterizado pela possibilidade de configuração por parte do usuário. As variáveis de processo podem ser visualizadas de forma gráfica, em tempo real, permitindo fácil e rápida compreensão do que acontece no processo controlado [25].

O sistema de supervisão proporciona o gerenciamento das instalações através de computadores, oferecendo flexibilidade, ou seja, alterações das características operacionais do sistema, confiabilidade, precisão, segurança e *feedback*, que permite confirmações das ações.

Pode permitir ainda o monitoramento e reastreamento de dados de um processo produtivo ou instalação física. Tais informações são coletadas através de equipamentos de aquisição de dados e, em seguida, manipuladas, analisadas, armazenadas e, posteriormente, apresentadas ao usuário. Estes sistemas também são chamados de SCADA (*Supervisory Control and Data Acquisition*). O objetivo principal do SCADA é propiciar uma interface de alto nível do operador com o processo, informando-o em tempo real de todos os eventos de importância da planta [27] [28] [29].

#### a) Funcionamento do Sistema de Supervisão

São necessários alguns componentes físicos, como sensores, atuadores e rede de comunicação dentre outros, e componentes lógicos, divisão interna do sistema, como, por exemplo, as tarefas.

Os dados obtidos pelos sensores (pressão, temperatura, umidade, vazão, nível de água e nível de CO<sub>2</sub>, entre outros) são convertidos para sinais digitais e/ou analógicos e processados por microcontroladores, os quais oferecem as informações para o SCADA.

#### b) Protocolo de Comunicação

Um Protocolo de Comunicação é um conjunto de regras necessárias para a comunicação de dados ou mensagens entre equipamentos.

No mercado há dois tipos de protocolos: abertos e fechados. Os abertos não estão vinculados a uma única empresa desenvolvedora. Os fechados, também conhecidos como

protocolos proprietários ou privados, apresentam utilização exclusiva da empresa desenvolvedora.

Existem diversos protocolos de comunicação para diferentes tipos de aplicações. No mercado de automação, o ideal tanto para os fornecedores quanto para os usuários é a modularidade, além da padronização dos programas e equipamentos. Como isso é dificilmente alcançável em um livre mercado de fabricantes, passa a haver um grande interesse em que os dispositivos inteligentes “conversem” com todos os outros, ou seja, que existam normas técnicas de aceitação universal [31].

No nível 0 do sistema de supervisão (camada onde se encontram os computadores com os *softwares* de supervisão), o acesso às informações do sistema pode ser realizado, no caso do protocolo aberto, através de [30]:

- Conexão interna (Intranet) via rede Ethernet, protocolo TCP/IP e
- Conexão externa (Intranet) via rede Ethernet, protocolo TCP/IP ou conexão criptografada via comunicação digital (ex: modem).

No caso do protocolo fechado ou proprietário, o acesso é realizado apenas pela empresa implementadora.

Já o acesso aos níveis 1, IHM, e 2 (Controladores e outras integrações) do sistema de supervisão é feito através de uma rede com protocolo de comunicação específico para essas camadas de controle. Dentre os protocolos abertos, destacam-se o BACnet<sup>3</sup>, LonTalk<sup>4</sup> e MODBUS<sup>5</sup>. O que mais se popularizou entre os fabricantes de diversos tipos de dispositivos para automação, tais como *Industrial Automation System (IAS)* e *Building Management System (BMS)*, foi o protocolo MODBUS, por fatores como sua alta flexibilidade e, ao mesmo tempo, facilidade de implementação, não só em dispositivos como microcontroladores e PLCs (*Power Line Communications*), mas também em sensores inteligentes, que podem enviar seus dados para sistema de gerenciamento [30].

---

<sup>3</sup> <http://www.bacnet.org>

<sup>4</sup> <http://www.echelon.com/developers/lonworks/protocol/default.htm>

<sup>5</sup> <http://www.modbus.org>

### c) Funcionalidades e Recursos

Há maneiras de verificar o funcionamento de um determinado equipamento ou o valor de alguma variável no sistema de supervisão. Também podem ser analisadas, controladas, exibidas e comandadas tais informações. Algumas funcionalidades básicas de um sistema de supervisão podem ser destacadas:

- Aquisição de dados: processo que envolve a coleta e transmissão de dados desde a planta (estações remotas) até as estações centrais de monitoração. O processo é concluído com o armazenamento da informação na base de dados [32];
- Visualização de dados: consiste na apresentação das informações através de IHM, com o intuito de exibir o status atual ou passado dos dados ou permitir a configuração dos parâmetros da planta [30] [32];
- IHM: em qualquer *software* desenvolvido deve haver uma interface amigável, na qual os recursos de tela sejam de fácil manipulação e interativos, onde o usuário não sinta nenhuma dificuldade de manipulação;
- Monitoração: deve permitir apresentar, ao operador, os valores provenientes das medições realizadas, além das indicações de estado dos disjuntores, chaves seccionadoras e demais equipamentos de interesse. As medições podem ser obtidas por meio de transdutores conectados às entradas analógicas de unidades terminais remotas ou controladores programáveis, ou ainda através de equipamentos dedicados que promovam a transdução analógico/digital [33];
- Alarmes e Eventos: devem notificar o operador da ocorrência de alterações espontâneas da configuração da malha elétrica, a irregularidade funcional de algum equipamento ou do sistema digital ou ainda a ocorrência de transgressões de limites operativos de medições. Há identificação visual e sonora das ocorrências e registro para análises posteriores [30] [33];
- Histórico: todas as medições, indicações de estado, alarmes e ações do operador devem ser armazenadas em bancos de dados relacionais. Essas informações são de grande importância, quando utilizadas na manutenção preditiva [30] [33];
- Acesso Remoto: devem permitir a monitoração (e alteração de parâmetros através de proteção com senha) à distância via conexão digital [30] [33];
- Segurança: deve permitir o acesso aos sistemas de gerenciamento e pode ser protegido através de senhas subdivididas em vários níveis de acesso [30];
- Relatório: deve permitir a visualização e impressão dos dados [30].

A seguir são descritos os subsistemas prediais estudados e incorporados ao sistema de supervisão desenvolvido neste trabalho.

### **2.2.1.1 Subsistema Predial de Ar Condicionado**

Este subsistema apresenta possibilidade de aquecimento e resfriamento dos ambientes. Água gelada e quente são produzidas centralmente e distribuídas no edifício para onde resfriam ou aquecem o ar em unidades de condicionamento. O controle de temperatura em ambientes é realizado através da modificação da posição de válvulas que variam a vazão de água quente ou fria que passa pelas serpentinas. O ar a ser condicionado é renovado em caixas de mistura, de acordo com a necessidade da zona [34].

O condicionamento de ar é o processo de tratamento do ar que controla sua temperatura, umidade e qualidade para a manutenção de condições ambientais, a fim de produzir o conforto e a produtividade de seus usuários [35]. Um exemplo de subsistema de ar condicionado é apresentado na Figura 2.4 e seu modo de operação é apresentado a seguir.

Existem dois tipos de condicionamento de ar: aquecimento e refrigeração. O primeiro envolve a simples adição de energia térmica para sobrepor as perdas na edificação, enquanto que o segundo utiliza métodos mecânicos e químicos para a produção de ar frio. O princípio de operação destes sistemas de condicionamento (resfriamento ou aquecimento) envolve quatro seqüências de transferência de frio (ou calor) [36]: sistema de distribuição do ar; sistema de distribuição do fluido refrigerador ou aquecedor; sistema de refrigeração; e sistema de expulsão do calor gerado durante o processo.

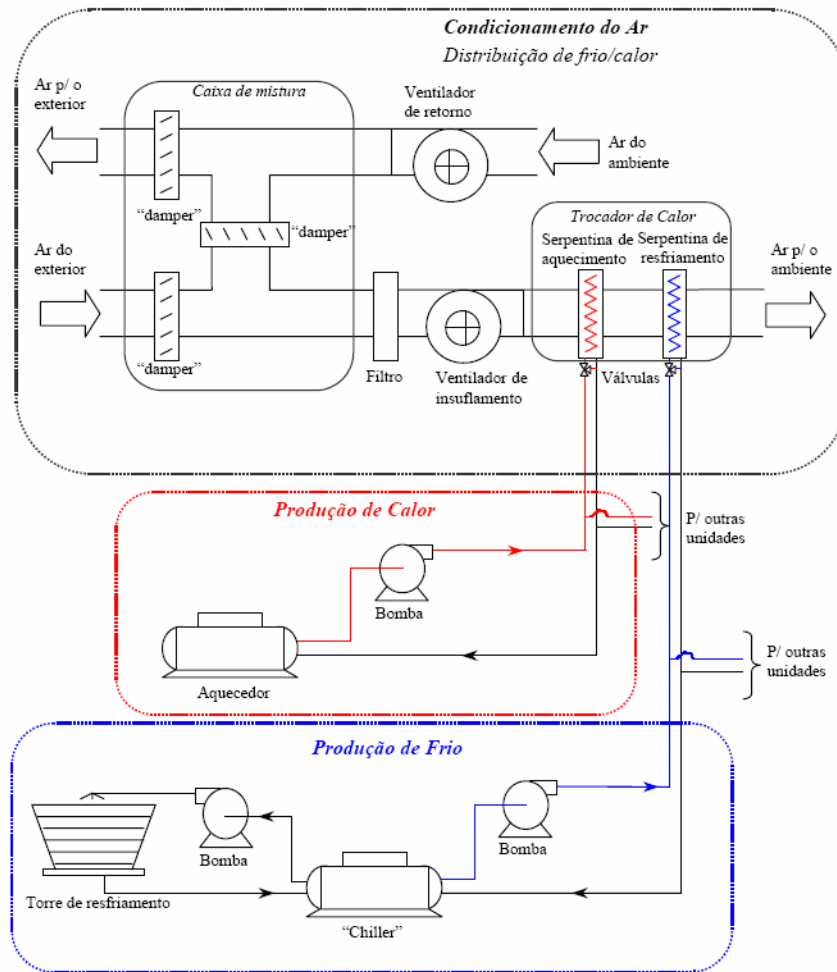


Figura 2.4: Esquema de um sistema de ar condicionado central [37]

A “produção de frio” é geralmente realizada em *chillers*, onde um fluido é resfriado. Este fluido pode ser um líquido ou um gás e é comumente chamado de refrigerante. O fluido refrigerante é bombeado até as unidades de condicionamento do ar onde, através de um trocador de calor, recebe o calor do ar a ser resfriado. No *chiller* ocorre a retirada de calor do refrigerante aquecido nas unidades de condicionamento. Este calor deve então ser devolvido à atmosfera. Alguns *chillers* liberam calor diretamente para o ar, outros o transmitem para a água que é bombeada até as torres de resfriamento, onde é liberado para a atmosfera exterior [37].

A produção de calor é realizada de modo semelhante. No *boiler*, ou aquecedor, ocorre o aquecimento do fluido, geralmente água, que transmitirá o calor. A água quente é

bombeada até as unidades de condicionamento, onde trocará calor com o ar do ambiente a ser condicionado [37].

O condicionamento do ar é realizado em diversas unidades que podem estar espalhadas pelo edifício. Neste subsistema, o ar é retirado do ambiente condicionado por um ventilador. Uma parcela deste ar é renovada através da sua substituição por ar da atmosfera exterior. A quantidade de ar a ser renovada e a quantidade de ar exterior a ser enviada ao ambiente é determinada na caixa de mistura. O ar a ser condicionado é então conduzido para a unidade de condicionamento onde é filtrado e, posteriormente, resfriado ou aquecido, retornando então ao ambiente condicionado [37]. Estes elementos, que servirão para esfriar ou aquecer o ar a ser enviado ao ambiente, são conhecidos por *fancoils*.

O sistema básico descrito pode ainda ser acrescido de outros subsistemas, de acordo com a necessidade, como, por exemplo, umidificação e/ou desumidificação do ar, controle entálpico, ventilação e filtragem do ar [38].

### **2.2.1.2 Subsistema Predial de Iluminação**

Em um projeto de automação predial, a luz é um dos principais itens, pois tem por finalidade garantir condições ideais de visibilidade para os usuários, complementar a decoração de ambientes e ainda é capaz de influenciar o humor das pessoas. Trata-se não somente de necessidade, mas também de conforto [39].

O conforto luminoso resulta em boa distribuição de luz no ambiente (quantidade e qualidade), que está associada à visibilidade e segurança, e na ausência de contrastes excessivos.

Para a aplicação da iluminação na automação predial é necessário saber os principais tipos de lâmpadas [40].

As lâmpadas incandescentes são constituídas por um filamento de tungstênio alojado no interior de um bulbo de vidro preenchido com gás inerte. Quando da passagem da corrente elétrica pelo filamento, os elétrons se chocam com os átomos de tungstênio, liberando uma energia que se transforma em luz e calor.



As lâmpadas halógenas possuem luz mais branca e brilhante, que possibilita realçar as cores e os objetos com eficiência energética maior do que a das lâmpadas incandescentes comuns. Oferecem mais luz com potência menor ou igual à das incandescentes comuns, além de possuírem vida útil mais longa, variando entre 2.000 e 4.000 horas.

Por fim, as lâmpadas fluorescentes possuem princípio de funcionamento similar ao das fluorescentes, mas com dimensões bastante reduzidas. Oferecem excelente qualidade de luz, alta eficiência energética, longa durabilidade (até 20 vezes maior que a das lâmpadas incandescentes comuns), excelente distribuição de luz, um consumo energético 80% menor e aquecem menos o ambiente, representando forte redução na carga térmica das grandes instalações.

Os equipamentos auxiliares das lâmpadas fluorescentes são o reator e o *starter*. Os reatores, que geralmente são bobinas enroladas em um núcleo de ferro, têm duas funções: a de produzir a sobretensão no momento do desligamento do *starter* e limitar corrente. O *starter* baseia-se no princípio do bimetal, isto é, um contato que no início do funcionamento está fechado, mas que, com o aquecimento da lâmina, abre o circuito, produzindo a sobretensão necessária para dar partida à lâmpada. É portanto, uma chave temporizadora, que permite que a corrente passe pelos filamentos nas extremidades do tubo.

A iluminação apresenta atualmente aplicação ampla com funcionalidades que ultrapassam os conceitos de apenas ligar e desligar a luz, resultando em um controle eficaz dos materiais e equipamentos que se traduz em uma boa solução para a obtenção de economias substanciais. Estas economias podem ser obtidas através da otimização na operação dos sistemas de iluminação. Assim, lâmpadas e luminárias conectadas a sistema inteligentes podem ligar e desligar automaticamente e regular a intensidade da luz, vantagens que proporcionam economia de eletricidade.

### **2.2.1.3 Subsistema Predial de Elevador**

Elevadores primitivos eram acionados por energia humana ou animal e, como resultado, a segurança, o conforto e a eficiência do transporte eram comprometidos.

Durante séculos, as comunidades foram agrárias e de vida horizontal e foi apenas após a revolução industrial que os modernos elevadores para transporte de passageiros foram desenvolvidos [41] [42] [43].

Os sistemas de elevadores são instalados nos edifícios para satisfazer as necessidades de transporte vertical de seus usuários e são necessários não apenas para a conveniência e o conforto humano, mas também para assegurar condições de trabalho, comerciais e residenciais [44].

No funcionamento de um sistema convencional de elevador, existem essencialmente dois estágios: no primeiro, um painel controla a chamada em cada andar do edifício e, no segundo momento, já dentro do elevador, um painel indica o andar selecionado.

Para exemplificar o funcionamento de um elevador, considera-se que ele inicialmente esteja parado em algum andar do prédio. Quando um passageiro, através de um painel, seleciona o botão indicando se deseja subir ou descer do andar aonde se encontra, o sistema do elevador identifica, entre os de elevadores disponíveis, o mais próximo do andar solicitado, que é deslocado até o andar do passageiro. Uma vez dentro do elevador, o passageiro escolhe o andar desejado, iniciando assim o trajeto para o destino.

### **2.3 INTELIGÊNCIA ARTIFICIAL**

O desejo de resolver problemas cada vez mais complexos tem impulsionado o desenvolvimento da Inteligência Artificial (IA) [45], que é um ramo da ciência que estuda o conjunto de paradigmas que pretendem justificar como um comportamento inteligente pode emergir de implementações artificiais em computadores [46].

Os trabalhos desenvolvidos pelas máquinas agora buscam integrar, aos sistemas de processamento, características típicas do comportamento inteligente, otimizando-os.

Para se obter um projeto otimizado, utilizam-se usualmente soluções já existentes. A primeira tarefa é identificar os elementos do projeto que podem ser melhorados, de

forma a aumentar seu desempenho, reduzir seu custo, ou ambos. Estes elementos a serem melhorados são as variáveis de projeto a serem otimizadas [47].

O projeto ótimo difere do processo tradicional pela introdução de técnicas numéricas, de forma que a alteração do valor de suas variáveis, à medida que se tenta obter melhores soluções, é feita automaticamente, seguindo um procedimento pré-estabelecido, definido pelo método de otimização utilizado. Nos últimos 40 anos, foram desenvolvidas diversas técnicas numéricas para tratar o problema da busca pela otimização [48] [49] [50] [51] [52] [53] [54] [55].

A existência de uma grande variedade de técnicas é resultante de uma constatação prática e teórica: a eficiência de um método de otimização é dependente do tipo de problema que está sendo resolvido, pois não existe uma técnica melhor que todas as outras, mas aquela mais adequada para um dado problema [53] [56].

De fato, os métodos tradicionais para otimização são bastante eficientes, quando aplicados em problemas que apresentam um espaço de projeto convexo com variáveis contínuas e onde a função objeto e suas restrições não possuem características altamente não-lineares. Todavia, muitos problemas em engenharia apresentam espaços de projetos complexos que podem ser não-convexos ou mesmo não-contínuos, com a presença de variáveis de diversos tipos [57]. Essas características reduzem bastante a eficiência dos métodos tradicionais, que tendem a fornecer soluções subótimas.

No presente trabalho foram utilizadas duas técnicas de IA para otimização do sistema predial: Lógica *Fuzzy* e Sistema Especialista, que serão apresentadas a seguir.

### **2.3.1 Lógica *Fuzzy***

A lógica sempre foi estudada em IA como um formalismo para representação do conhecimento. A representação obtida é, no entanto, inflexível demais para lidar com conceitos do mundo real sem exigir um nível elevado de simplificação. Na lógica clássica existe um mundo de clareza, rigor e precisão em que conceitos reais não se encaixam com facilidade [58].

A teoria dos conjuntos *Fuzzy* foi proposta por Lotfi Zadeh em 1965. Por muito tempo permaneceu incompreendida. Em meados dos anos 80, Mamdani utilizou a para projetar controladores *Fuzzy*. A partir daí houve um grande progresso da área, em especial com muitas aplicações reportadas do Japão [46].

A Lógica *Fuzzy*<sup>6</sup> provê um método de traduzir expressões verbais, vagas, imprecisas e qualitativas, comuns na comunicação humana em valores numéricos. Trata matematicamente de informações imprecisas usualmente empregadas na comunicação humana [58]. A implementação da Lógica *Fuzzy* cresce a cada dia e nas mais diversas áreas, como automação predial, robótica e processamento de imagens.

Os sistemas *Fuzzy* são sistemas especialistas baseados em conhecimento que criam a base de conhecimento na forma de um banco de regras.

O princípio fundamental da Lógica *Fuzzy*, o da dualidade, estabelece que dois eventos opostos coexistem. Isto é, um elemento pode pertencer, em certo grau, a um conjunto e, em um outro grau, a um conjunto. Em Lógica *Fuzzy*, os paradoxos podem ser reduzidos a “meias verdades” ou “meias mentiras”, através de uma lógica multivalorada. O verdadeiro (1) e o falso (0) são substituídos por graus de pertinência que podem assumir qualquer valor entre 0 e 1, também conhecido como Conjuntos Ordinários ou *Crisp*.

A idéia de grau de pertinência da Lógica *Fuzzy* possibilita agrupar os elementos de maneira diferente da aplicada na lógica clássica, o que permite re-interpretar antigos conceitos, elaborados segundo esta lógica. As duas se diferem na seguinte maneira [60]:

- Função característica:

$$A : X \rightarrow \{ 0, 1 \}$$

elementos do conjunto universo X pertencem totalmente ao conjunto A:  $A(x) = 1$

ou não pertencem ao conjunto:  $A(x) = 0$

- Função de Pertinência

$$A : X \rightarrow [ 0, 1 ]$$

---

<sup>6</sup> Lógica *Fuzzy* também é conhecida como Lógica Difusa ou Nebulosa, como tradução da palavra inglesa *fuzzy*.

X - conjunto base

A - conjunto fuzzy

elementos do conjunto base pertencem ao conjunto com um certo grau, que usualmente varia entre 0 e 1.

Sendo que a função característica representa a teoria clássica dos conjuntos onde um dado elemento pertence ou não pertence ao conjunto determinado. Já no conjunto nebuloso existe um grau de pertinência de cada elemento a um determinado conjunto.

A Lógica *Fuzzy* é mais usada nos sistemas de controle [61]. O sistema controla uma máquina para conseguir uma resposta desejada, dado um número de entradas do ambiente, usando um processo de “fuzzificação”, gerador de entradas para um mecanismo de inferência, que é uma base de conhecimento das ações a serem tomadas. O processo inverso, “defuzzificação”, gera os valores reais, numéricos que serão aplicados na máquina ou no processo sobre controle.

A Fuzzificação, Inferência e Defuzzificação são etapas do raciocínio *Fuzzy*, que são apresentadas na Figura 2.5.

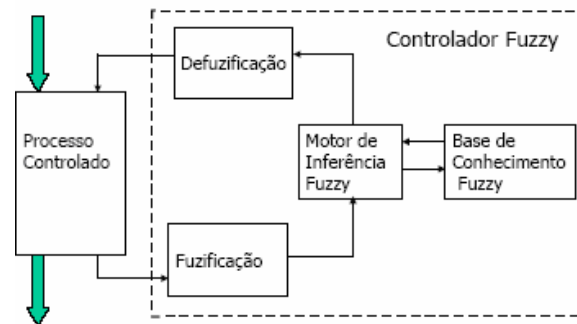


Figura 2.5: Módulos de um Sistema Fuzzy [60]

Cada uma destas etapas é descrita a seguir:

- Fuzzificação: as variáveis linguísticas são definidas de forma subjetiva, bem como as funções de pertinência. Os conceitos, análise do problema e definições das variáveis, entre outras, são englobadas nesse processo;

- Inferência: nesse estágio são definidas as regras e depois analisadas. A etapa abrange definição das proposições, análise das regras e criação de uma ou mais regiões;
- Defuzzificação: etapa nas quais as regiões resultantes são convertidas em valores para a variável de saída do sistema. Corresponde à ligação funcional entre as regiões *fuzzy* e o valor esperado. Há várias técnicas de defuzzificação; podem ser exemplificadas por: centróide, usada nesse trabalho, *first-of-maxima*, *middle-of-maxima*, critério máximo.

As características da lógica *fuzzy* são:

- Modelagem de problemas complexos, capazes de lidar com problemas, com propriedades não-lineares, por exemplo;
- Modelagem cognitiva, que permite codificar o conhecimento de forma similar ao modo como os especialistas expressam o processo de decisão;
- Modelagem de sistemas envolvendo múltiplos especialistas, que são capazes de conciliar informação de especialistas consistentes (colaboradores) ou conflitantes (contraditórios);
- Complexidade reduzida, pois possuem poucas regras similares às expressas por especialistas;
- Manipulação de incertezas, que são lidam de forma consistente e matemática com incertezas.

### 2.3.2 Sistema Especialista

Os Sistemas Especialistas (SEs) são sistemas de IA, baseados em conhecimento, que emulam um especialista humano na resolução de um problema significativo em um domínio específico. O SE é projetado e desenvolvido para atender a uma aplicação determinada e limitada do conhecimento humano. É capaz de emitir uma decisão, apoiada em conhecimento justificado, a partir de uma base de informações, tal qual um especialista de determinada área do conhecimento humano [62] [63] [64] [65].

As pesquisas iniciais para o desenvolvimento dos SEs, segundo [66], podem ser atribuídas às seguintes entidades e grupos: universidades, com destaque para a *Stanford*

*University* (SU)<sup>7</sup> e a *Carnegie-Mellon University*<sup>8</sup> (CMU), os institutos de pesquisas, como *Rand Corporation*, instituto de pesquisa que atua na área da segurança pública, e as empresas de engenharia de *softwares*, como a *APEX-Applied Systems, Inc* e o *CGI-Carnegie Group, Inc*.

O SE, além de inferir conclusões, deve melhorar as decisões e o desempenho de raciocínio. Para tal, utiliza regras de “condição-ação”, que têm como objetivo ajudar a resolver determinado problema por meio de informações reunidas em um banco de dados, inseridas por um especialista humano no assunto [67]. Estas regras, denominadas regras de produção, baseiam-se em pressupostos lógicos, utilizando para isso um vocabulário próprio, de forma que o computador entenda as afirmações verdadeiras, falsas ou condicionais. Desta forma, a utilização do SE / E / ENTÃO definem a lógica de um sistema computacional, fazendo com que ele apresente os resultados pretendidos, baseado na lógica programada. O conhecimento do especialista no assunto a ser programado é então fundamental, para que o sistema chegue a resultados plausíveis [67] [68].

Geralmente o formato das regras é o seguinte:

- Identificador: SE <LHS> ENTÃO <RHSI>;
- SE <condição> ENTÃO <ação>;
- SE <premissa 1 > E <premissa 2 > ENTÃO <conclusão A >.

Onde:

- <LSH (*Left-Handed Side*)> e <condição>: representam as condições a serem atendidas, envolvendo conjunção, negação ou disjunção e os termos que podem ser desenvolvidos pelas conclusões de outras regras e do estado atual da base de conhecimento;
- <RHS (*Right-Handed Side*)> e <ação>: produz as conclusões, sendo as condições verdadeiras, altera o estado atual da base de conhecimento, adicionando, modificando ou removendo unidades de conhecimento presentes na base.

A arquitetura de um sistema especialista é dividida em componentes básicos: um banco de informações que contenha todo o conhecimento relevante sobre o problema de

---

<sup>7</sup> <http://www.stanford.edu>

<sup>8</sup> <http://www.cmu.edu>

uma forma organizada (base de conhecimento); um conjunto de métodos inteligentes de manipulação destes conhecimentos e os mecanismos de inferência e interface com o usuário [64] [65] [68]. Essa arquitetura é representada na Figura 2.6.

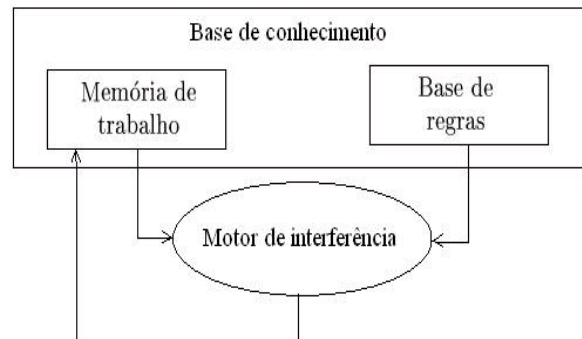


Figura 2.6: Arquitetura de um SE [61]

Cada componente básico é descrito a seguir.

- Base de Conhecimento: consiste na parte central de um SE. É a representação do conhecimento no domínio do problema em questão, geralmente extraída de um ou vários especialistas, de forma declarativa e livre de detalhes de controle e implementação. Idealmente é composta de declarações, em algum formalismo de representação de conhecimento disponível, descrevendo o domínio da aplicação. Na interação entre base, regras, fatos e heurística, motor de inferência e usuário, o sistema encontra a solução para os problemas propostos, podendo inclusive ampliar essa capacidade e sistematicamente aprender com o usuário através de dados novos inseridos;
- Motor de Inferência: também chamado máquina de inferência, é responsável pela manipulação da base de conhecimento durante a resolução de problemas. Recebe esta denominação, porque usa o conhecimento da base e os fatos relativos a uma determinada consulta para obter conclusões. A natureza do motor de inferência depende do formalismo utilizado para representar a base de conhecimento e da estratégia de solução de problemas considerada apropriada pelo projetista do sistema. Ela pode utilizar diversas formas de raciocínio: dedução lógica, regras de decisão, Tabelas de decisão, redes semânticas, relacionamentos lógicos, redes Bayesianas, probabilística ou Lógica *Fuzzy*;



- Interface com o Usuário: todo sistema especialista é interativo e precisa de um componente para gerenciar a interação entre o usuário e o sistema. A interação básica numa sessão de uso de um SE consiste em perguntar questões relevantes, apresentar conselhos, respostas e prover explicações requeridas pelo usuário.

As vantagens e desvantagens dos SEs são as seguintes:

- Vantagens: a decisão é assentada em uma base de conhecimento e existe estabilidade, flexibilidade, integração de ferramentas e agilidade na determinação dos problemas;
- Desvantagens: os sistemas especialistas tendem a trabalhar de uma forma rotineira, sem inspiração ou criatividade, onde não é sintetizado novo conhecimento, há dificuldade de lidar com situações inesperadas (entrada de informação e representação simbólica e não sensória) e não existe conhecimento do senso comum [66].

No próximo capítulo será apresentada a modelagem de um Sistema Inteligente de Supervisão (SIS), bem como a sua concepção e simulação, utilizando conceitos abordados neste capítulo.

### **3 - SISTEMA SUPERVISÃO PREDIAL: APLICAÇÃO DE INTELIGÊNCIA ARTIFICIAL E ENGENHARIA DE *SOFTWARE***

A realização de estudos iniciais sobre automação predial, técnicas de IA e engenharia de *software* se fez necessária para o desenvolvimento do sistema de supervisão predial. A adoção de automação em ambientes prediais permite uma gestão de alto nível das principais componentes envolvidas e das suas inter-relações.

Em uma instalação predial típica encontram-se diversos subsistemas tais como: climatização, comunicação, diagnóstico e manutenção, elevador, energia, iluminação, sistema de irrigação, incêndio, portaria, patrimonial, segurança e gestão de presença. A descrição do sistema em partes está mostrada na Tabela 3.1.

No contexto de edifícios inteligentes é fundamental a noção de integração, que está associada à capacidade de vários subsistemas poderem se comunicar, trocarem informações e cooperarem para atingir objetivos comuns.

Atualmente, a utilização de sistemas de supervisão chamados SCADA (*Supervisory Control & Data Acquisition Systems*) permite a monitoração em tempo real para todos os processos existentes no sistema predial. Isso é possível com a utilização de equipamentos que coletam e enviam dados ao sistema de supervisão (como o SCADA) e, a partir dessas variáveis, o SCADA analisa, armazena e informa ao usuário todos os procedimentos realizados, através de uma interface.

Tabela 3.1: Descrição dos subsistemas

<i><b>Subsistema</b></i>	<i><b>Descrição</b></i>
<b>Climatização</b>	<p>Ar condicionado/ Ventilação/ Aquecimento – Liberdade para abertura de janelas.</p> <p>Controle dos sistemas em situações de emergência - iluminação nas áreas vizinhas do sinistro, por exemplo, em caso de incêndio.</p> <p>Diminuição de custos.</p> <p>Monitoração de tempos de funcionamento.</p> <p>Monitoração do número de pessoas presentes e locais aonde se encontram.</p> <p>Programação de horários, estações do ano.</p>
<b>Comunicação</b>	<p>Acesso Internet.</p> <p>CFTV – Controle de circuito fechado de televisão.</p> <p>Comunicação de dado.</p> <p>Controle da distribuição de música ambiente.</p> <p>Interfone e vídeo da portaria.</p> <p>Redes de computadores.</p> <p>Telefonia móvel/ fixo.</p> <p>TV a cabo e satélite.</p>
<b>Diagnóstico e Manutenção</b>	<p>Diagnóstico de falhas.</p> <p>Manutenção de todos os subsistemas existentes.</p> <p>Monitoração e teste do estado de funcionamento dos equipamentos.</p>
<b>Elevador</b>	<p>Elevadores e escadas e escadas rolantes.</p> <p>Controle dos sistemas em situações de emergência - elevadores se deslocam para pisos seguros, mudam o sentido das escadas/ escadas rolantes de modo a facilitar evacuação.</p>
<b>Energia</b>	<p>Distribuição.</p> <p>Geração.</p> <p>Otimização de operação.</p> <p>Sistema de monitoramento/ análise.</p> <p>Transmissão.</p>
<b>Gestão de Presença</b>	<p>Controle de acesso.</p> <p>Disponibilidade da informação sobre tempos de trabalho e saldos horários.</p> <p>Frequência.</p> <p>Marcação de férias.</p>

<b><i>Subsistema</i></b>	<b><i>Descrição</i></b>
<b>Iluminação</b>	Interna/ Externa.  Exaustores.  Monitoração do número de pessoas presentes e locais aonde se encontram .
<b>Incêndio</b>	Alarme de emergência  Controle dos sistemas em situações de emergência - iluminação nas áreas vizinhas do sinistro  Detecção(fumaça, combate ao incêndio), prevenção
<b>Patrimonial</b>	Controle de cada equipamento em cada departamento  Gestão do próprio edifício, como contrato de aluguel e prestação de serviço  Registro do espaço ocupado por cada departamento  Registro e gestão dos bens patrimoniais
<b>Portaria</b>	Controle e registro de entrada/ saída de pessoas e equipamentos  Fornecimento de informações sobre a organização do edifício  Identificação das pessoas  Manutenção preventiva e corretiva  Registro de mensagens de visitas
<b>Segurança</b>	Detecção, prevenção, arrombamento e intrusão  Alarmes de emergência  Controle de acesso  Controle de estacionamento de veículos  Detecção e combate a incêndio  Detecção de fuga de gás/ água  Sistema de vídeo portaria  Monitoração
<b>Sistema de Irrigação</b>	Controle de irrigação de acordo com as condições atmosféricas  Corte de custos  Monitoração de consumos

Esse estudo inicial foi realizado para oferecer uma contribuição ao gerenciamento e ao comando de supervisão predial, uma concepção de um Sistema Inteligente de Supervisão (SIS), incluindo a implementação física de um demonstrador conceitual, previsto para operar inicialmente no segundo andar do Palácio do TJDF e depois adotado nas demais instalações do Tribunal no DF.

Um SIS tem como propósitos a centralização das informações referentes ao funcionamento dos diversos subsistemas vitais às operações do edifício; a redução dos custos operacionais e economia de energia, através da utilização racional dos recursos disponibilizados; a comunicação do estado dos subsistemas através de interface homem-máquina (IHM), visando antecipar os problemas e facilitar a tomada de decisões; o fornecimento de subsídios para a programação da manutenção preventiva dos diversos equipamentos e aumento da segurança da instalação, através da imediata detecção de situações anormais e da tomada das devidas de decisão quanto às providências.

Um SIS irá monitorar, supervisionar e atuar em diversas variáveis e dispositivos do sistema de controle. Além disso, poderá interagir com o sistema de supervisão (como o SCADA) ou atuar diretamente nos subsistemas realizando intervenção e aplicando técnica de IA nas tomadas de decisões. Também apresenta uma estrutura que independe do tipo de aplicação e que, por isso, permite aplicações diversas, além da automação predial. Para exemplificar, um SIS pode ser utilizado no gerenciamento do Laboratório de Dispositivos de Circuito Integrado, LDCI, do Departamento de Engenharia Elétrica da UnB na execução de um projeto de circuitos integrados.

O grande diferencial do SIS está na sua capacidade de intervir de forma inteligente nos subsistemas, isto é, sendo capaz de fornecer possíveis soluções para situações adversas que venham a ocorrer. Assim, por exemplo, enquanto o SCADA só detecta possíveis falhas, um SIS, além de identificá-las, também fornece as possíveis soluções. Além disso, um SIS pode enviar alertas via e-mail, mensagens para celular ou mesmo para *paggers*, quando alguma situação de falha, capaz de comprometer o funcionamento do prédio, for detectado. Por fim, a integração do SIS e do SCADA é realizada através de uma interface de comunicação baseada no padrão COM/DCOM, que permite a leitura e escrita de variáveis através de OPC.

A concepção do software do SIS seguiu os fundamentos de Engenharia de *software* apresentados nas seções 2.1.1 a 2.1.3.

Algumas características desejáveis foram consideradas para a definição do processo de desenvolvimento do *software*. São elas: leveza e praticidade; interatividade; completa informação; inteligência; compatibilidade e funcionalidades [1] [9] [11] [20] [25].

Foi realizada também uma pesquisa de linguagens de programação para atender melhor às necessidades específicas, avaliadas na seção 2.4. Foi selecionada a linguagem Java, sendo algumas das características desejáveis apresentadas na Tabela 3.2.

Tabela 3.2: Características da linguagem Java.

Java*	
Características gerais	Linguagem orientada a objetos, multi-plataforma, altamente extensível, <i>open-source</i> , escalonável e compilada em <i>byte-code</i>
Linguagem de script	JSP, tags customizáveis, sistemas de templates com suporte a internacionalização
Complexidade	Complexa devido ao seu alto poder de extensibilidade
Portabilidade	Várias Plataformas (Microsoft Windows, Solaris, Linux e outras implementações de plataformas UNIX)
Servidor	Qualquer servidores Web, incluindo Apache, Netscape e IIS utilizando container de Servlets
Integração com base de dados	Bases de dados em conformidade do ODBC, JDBC e bases de dados ODBMS

\*Utilização da plataforma web (JSP *Java Server Pages*)

O digrama de classe (Figura 3.1) representa abstrações do *software* do SIS e de suas relações. Nessa representação temos os atributos, caracterizando objetos da classe, e os métodos, comportamento da classe. Para uma visão geral do SIS apresentado na Figura 3.1, têm as classes: Conexão que é representada por uma conexão entre as outras; Usuário possibilita o cadastro, a exclusão e a edição de um determinado usuário e suas permissões concedidas; Log verifica se o usuário é cadastrado no servidor de dados e sua permissão para a navegação no SIS; Tipo, classe que há níveis de que o SIS é elaborado, por exemplo, a verificação do nível de permissão do usuário ou status de um determinado equipamento; Sistema classe que representa o sistema geral a ser desenvolvido; Aplicação\_IA que nessa classe encontra a técnica de IA a ser empregada dependendo do tipo de aplicação do SIS.

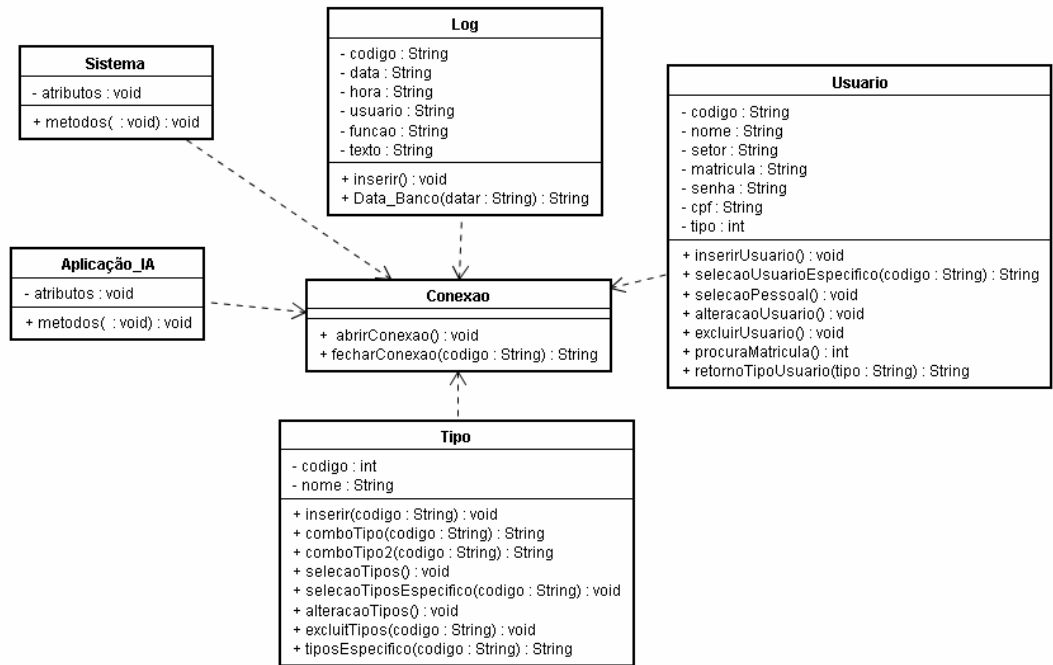


Figura 3.1: Diagrama de classe da concepção de um SIS

É importante salientar que o diagrama de atividade apresentado na Figura 3.2 representa o funcionamento do modo de acesso a um SIS e pode ser aplicado a qualquer tipo de sistema a ser implantado. Inicialmente há uma tela de acesso composta pelos campos *login* e senha. Uma consulta ao banco de dados verifica, o Log, se o usuário está cadastrado ou não. Se o usuário não for cadastrado é enviado uma mensagem, ainda na tela de acesso, que não foi possível realizar a conexão. No entanto, se houver cadastro, é verificado o nível de permissão associada ao acesso, Tipo. E finalmente o Sistema, que em função dessa verificação, é definido se o acesso permite, por exemplo, a visualização, cadastro e exclusão de dados contidos em um SIS.

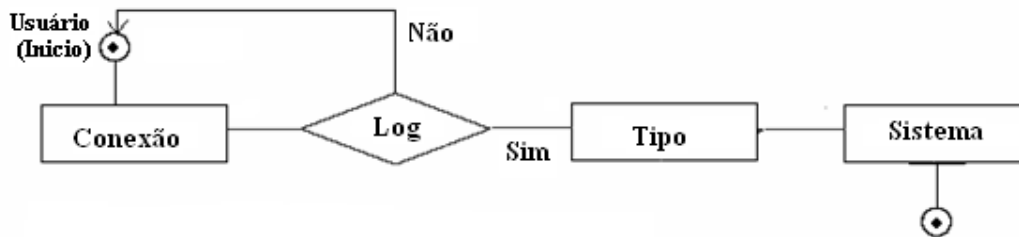


Figura 3.2: Diagrama de atividade do acesso ao *software* do SIS

Essa automação pode ser estendida a outros subsistemas existentes, mostrados no Apêndice A, e conseqüentemente poderá ser inserida no sistema de gerenciamento predial.

Atualmente o sistema de supervisão existente no TJDF é do tipo SCADA da empresa Johnson Controls<sup>9</sup>, mostrado na Figura 3.3, que está sendo implantado e que coleta dados de tabelas armazenadas em controladores tipo PLC. Ainda na Figura 3.3 pode ser observada cada comarca que compõe o TJDF. O projeto prevê a alteração de todos os sistemas de supervisão (Elipse<sup>10</sup>, Alerton<sup>11</sup> e Indusoft<sup>12</sup>) para o sistema de supervisão da Johnson Controls.

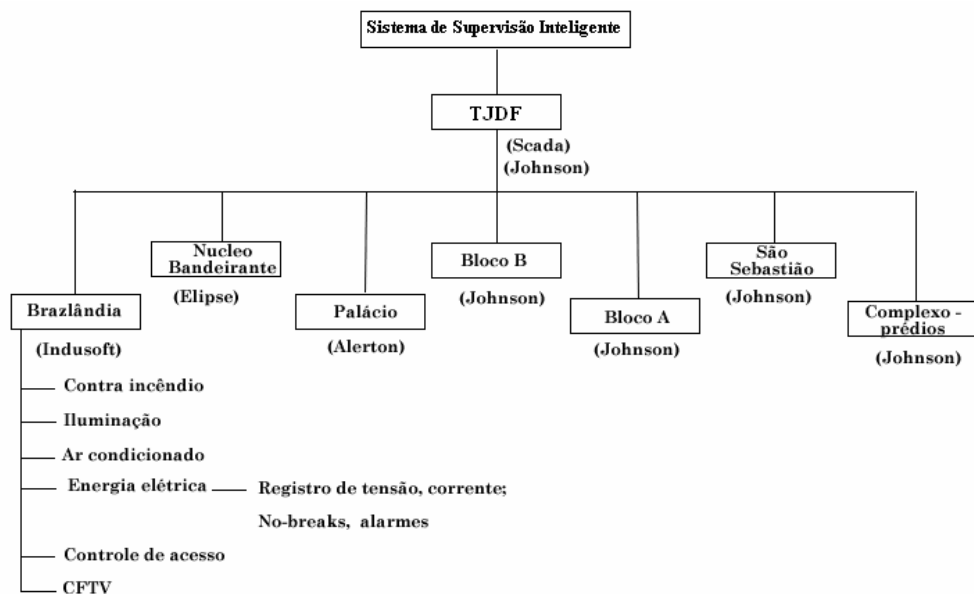


Figura 3.3: Futura implantação do SIS nas dependências do TJDF. Apenas a comarca de Brazlândia apresenta subsistemas prediais com automação

O desenvolvimento deste trabalho foi realizado através de modelagem e programação do *Software* do SIS e do banco de dados correspondente. Conforme mencionado anteriormente, devido às freqüentes alterações nos projetos do TJDF, a implementação física de um demonstrador conceitual tornou-se inviável dentro do prazo previsto para a realização desta dissertação. Assim para a validação do Sistema Inteligente

<sup>9</sup> [www.johnsoncontrols.com](http://www.johnsoncontrols.com)  
<sup>10</sup> [www.elipse.com.br](http://www.elipse.com.br)  
<sup>11</sup> [www.alerton.com](http://www.alerton.com)  
<sup>12</sup> [www.alerton.com/](http://www.alerton.com/)  
[www.indusoft.com.br](http://www.indusoft.com.br)



de Supervisão (SIS) foram utilizados recursos de simulação para essa parte que não foi possível a implementação física.

### 3.1 DESENVOLVIMENTO DO *SOFTWARE* DO SIS E DO SIMULADOR DO SIS

Como em todas as áreas de projeto, é primordial haver documentação, para que se obtenha os resultados desejados, facilitando também a reutilização e manutenção do projeto. A modelagem do *software* do SIS foi construída utilizando a IDE (*Integrated Development Environment* ou Ambiente Integrado de Desenvolvimento) JUDE<sup>13</sup>, que é uma IDE para UML criada com Java e de fácil uso.

A programação do *software* do SIS foi desenvolvida na linguagem Java utilizando a plataforma web (JSP *Java Server Pages*). Também há utilização de Sistema Especialistas, técnica de IA para a tomada de decisão e resolução de problemas.

Para armazenamento dos dados foi necessária a utilização de um banco de dados. A opção foi pelo MySQL<sup>14</sup> sendo gerenciado por um SGBD (Sistema de Gerenciamento de Banco de Dados), que é um mais populares, devido à sua otimização para aplicações *Web*, disponibilidade para praticamente qualquer sistema operacional e, principalmente, por ser um *software* livre (sob licença GPL - *General Public License*), o que significa que qualquer um pode estudá-lo ou alterá-lo conforme a necessidade.

A comunicação entre o *software* do SIS com o sistema de supervisão (como o SCADA) ou diretamente nos subsistemas é via HTTP (*Hyper Text Transfer Protocol*), protocolo de comunicação, mostrado na Figura 3.4.

Pode ser observado ainda na Figura 3.4 um sistema de supervisão (SCADA) que relaciona recebendo e transmitindo dados aos subsistemas via protocolo de comunicação. Também há um SGBD que gerencia os dados e informações que são armazenadas em um banco de dados por outro protocolo de comunicação.

---

<sup>13</sup> <http://jude.change-vision.com/jude-web/download/index.html>

<sup>14</sup> <http://www.mysql.com>

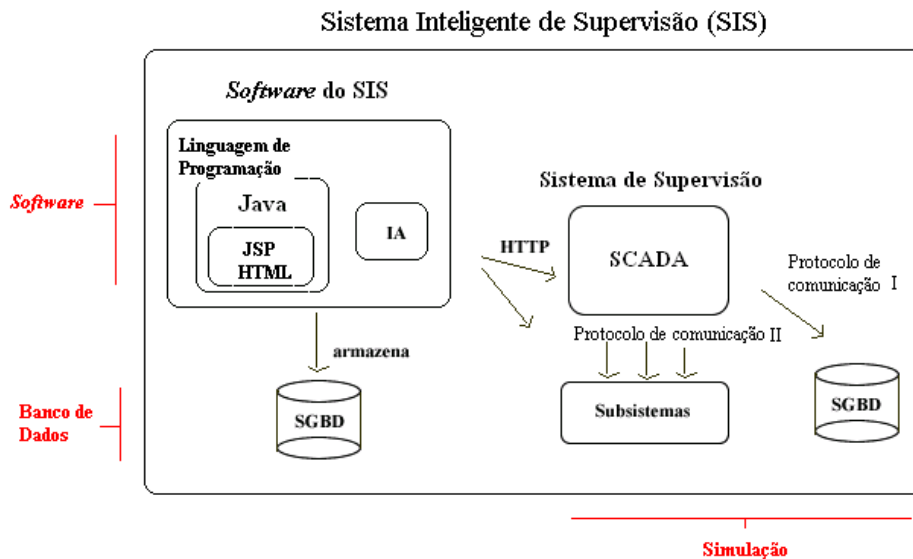


Figura 3.4: Desenvolvimento do *Software* e Simulador do SIS

O simulador do SIS foi realizado em torno de uma aplicação para um sistema de um prédio do TJDF que já possuía automação aplicada aos subsistemas de refrigeração, ar condicionado e iluminação, com a utilização de técnicas de IA. No Apêndice B são descritos a funcionalidade, a automação, os equipamentos e os problemas de cada subsistema. Utilizou as mesmas ferramentas utilizadas no desenvolvimento do *software* do SIS.

Assim deu origem a um interesse por sistemas de gerenciamento, uma vez que eles integram vários subsistemas para controlar de forma eficiente e eficaz os recursos, visando facilitar o trabalho do operador e racionalizar a mão de obra de operação. Outras vantagens são: a economia de custos, por considerar a estrutura geral dos serviços, dos subsistemas e do próprio gerenciamento em si, conforto e segurança ao usuário.

### 3.1.1 *Software* do SIS

É fundamental para qualquer *software* o conceito de segurança. O *software* do SIS restringe o acesso da operação mediante senhas. Apenas o administrador do sistema

fornece as permissões a cada usuário, informando o tipo correspondente. Pode haver alteração no tipo do usuário, mudando assim as permissões que lhe forem concedidas. Neste *software* há quatro tipos de usuários: administrador, usuário comum, operador de manutenção e operador do sistema. As permissões de cada tipo citado podem ser visualizadas na Tabela 3.3. Há também possibilidade de edição e exclusão de qualquer usuário, tipo de usuário e suas permissões (Figura 3.5).

Tabela 3.3: Tipos de permissões do *software* do SIS

	Administrador	Usuário	Operador de Manutenção	Operador do sistema
Verificar equipamentos	X	X	X	X
Excluir equipamentos	X			
Cadastrar equipamentos	X			X
Editar equipamentos	X		X	X
Verificar Monitor	X	X	X	X
Excluir Monitor	X			X
Cadastrar Monitor	X			
Editar Monitor	X			X
Verificar Falhas	X	X	X	X
Excluir Falhas	X			
Cadastrar Falhas	X			
Editar Falhas	X			

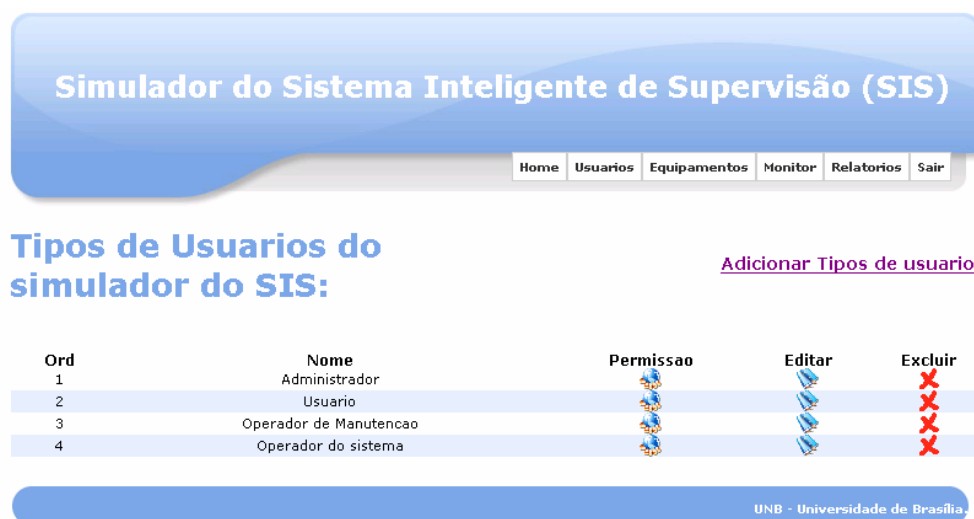


Figura 3.5: Tela do *software* do SIS com tipos de usuários e permissões

No *software* do SIS, há menus de navegação, tais como equipamentos, relatórios, monitor, que podem ser observados na Figura 3.6 e são descritos a seguir:

- Equipamentos: possui o cadastro de equipamentos, informando sua identificação, localização, tipo e estado (indicando se está em funcionamento ou não). Também, como no usuário, é possível editar e excluir qualquer equipamento;
- Relatórios: gera relatórios, baseados no banco de dados atual;
- Monitor: lista todos os equipamentos que não estão em funcionamento, ou seja, nos quais há falhas.



Figura 3.6: Tela de saída do *software* do SIS - Menu

No menu relatório há informações sobre *logs*, ou seja, é possível identificar qual usuário, através da sua identificação, a data e o horário de entrada no sistema e a alteração realizada (Figura 3.7).

Visando à construção de relatórios, foi utilizado o *iReport*<sup>15</sup>, que é uma ferramenta *open-source* através de uma aplicação Java, utilizando a biblioteca *JasperReports*. Para a geração de relatório foi realizado o mapeamento de dados para campos de acordo com o *layout* definido e com a determinação do design, tendo sido escolhido, dentre os diversos formatos possíveis, o PDF.

<sup>15</sup> <http://www.jasperforge.org/sf/projects/ireport>

Codig	Nome
7	Administrador
13	Usuario
14	Operador de Manutencao
15	Operador do sistema

Figura 3.7: Tela de relatório do *software* do SIS - Menu

Foi buscado um *shell, software* para desenvolver o sistema de classificação, que montasse uma base de conhecimento e que fosse transportado para o *software* do SIS. O Expert SINTA, uma ferramenta computacional gratuita desenvolvida pelo LIA – Laboratório de Inteligência Artificial da Universidade Federal do Ceará [62], foi escolhido. Esse *shell* utiliza um modelo de representação do conhecimento baseado em fatos e regras de produção através do uso de uma máquina de inferência e tem como vantagens uma interface amigável e a utilização de fatores de confiança.

As regras têm a forma geral de premissas e conclusões (se <condições> então <conclusões>) e a base de conhecimento foi implementada em forma de regras, em um total de 27. A Figura 3.8 exemplifica uma das regras aplicadas no *shell*. O Apêndice C mostra todas as regras utilizadas para desenvolvimento do SE.

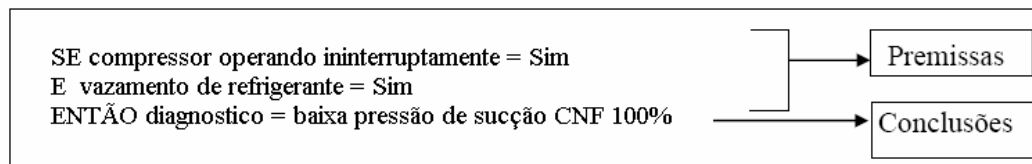


Figura 3.8: Exemplo de regra de produção do subsistema ar condicionado

A Figura 3.9 mostra o cadastro de regras de produção carregadas no sistema.

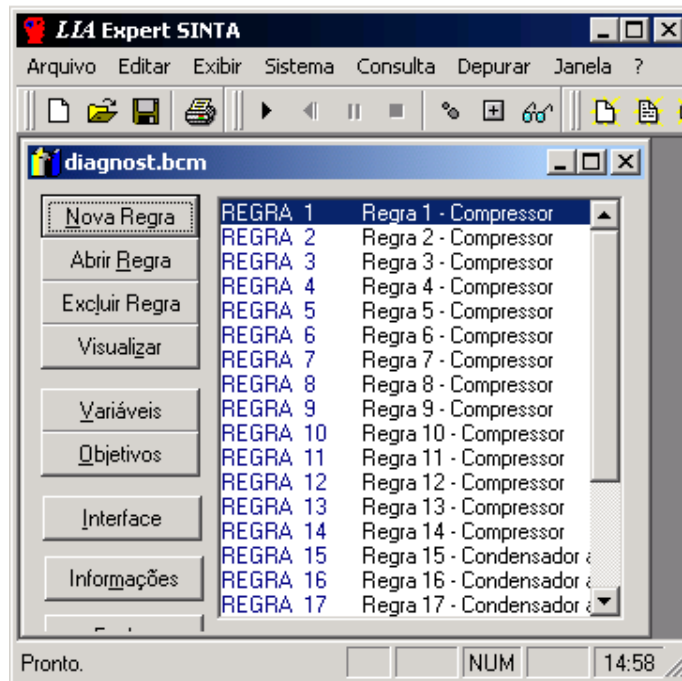


Figura 3.9: Regras de Produção – Expert SINTA

Adicionalmente, no Expert SINTA há um banco de dados relacional protegido, no qual ficam guardados os dados do *shell*, como regras, variáveis (problemas, fatores) e objetivos (o resultado da consulta) (Figura 3.9).

Por meio de perguntas iniciais, geradas pelo próprio *shell* e indicadas pelo especialista, o *shell* direciona as próximas perguntas, formando o caminho necessário para a base de conhecimento, resultando assim no diagnóstico e, posteriormente, na informação ao usuário.

No Expert SINTA, há possibilidade de exportar códigos no formato de texto (códigos, nomes das variáveis e objetivos, dentre outros) e também a base de conhecimento (variáveis objetivos e regras). A partir das regras de produção geradas na base de conhecimento, foi possível interligar os *softwares*, fazendo com que o *software* do SIS conseguisse ler as informações. Deste modo, o Expert SINTA e o *software* do SIS, funcionam de forma integrada.

Por fim, foi desenvolvida uma interface gráfica no *software* do SIS que permite ao operador a escolha do equipamento no qual há falhas e possíveis sintomas, retornando

assim o resultado. Desta maneira, o operador não necessita acessar o *shell* Expert SINTA, cuja operação fica restrita apenas a um especialista habilitado para alterar as regras de produção. O Apêndice D é apresentado o código referente ao desenvolvimento do SE.

Para modelar o *software* do SIS, foram utilizados os diagramas classe e caso de uso, conforme mostrado nas Figuras 3.10 e 3.11. Esses diagramas facilitam a compreensão do funcionamento do *software* do SIS desenvolvido.

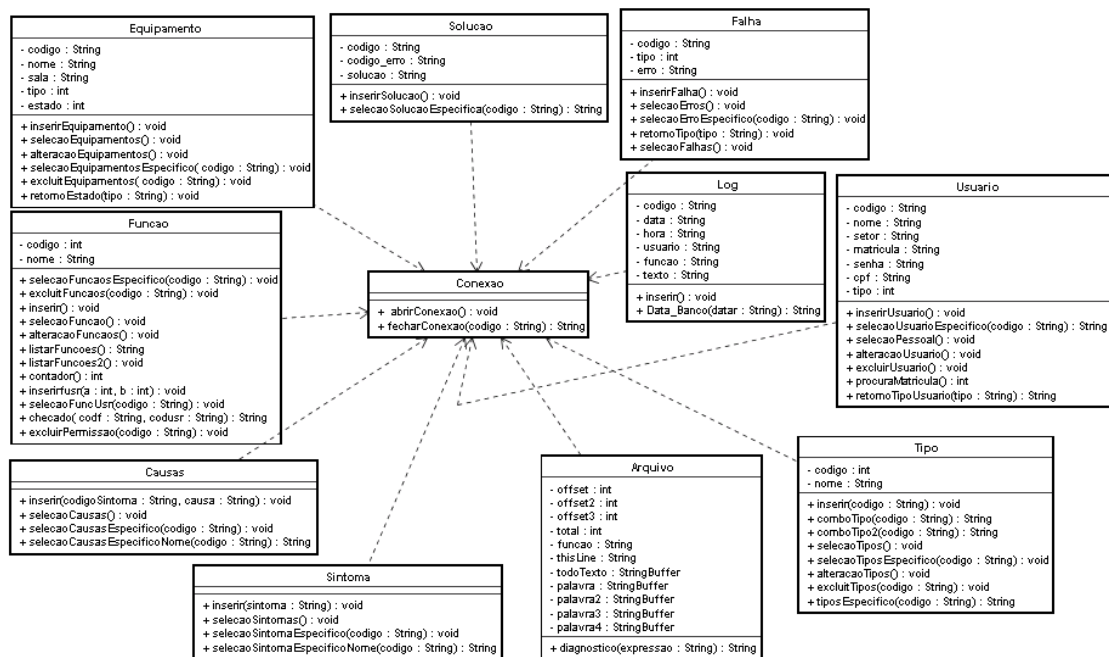


Figura 3.10: Diagrama de classe

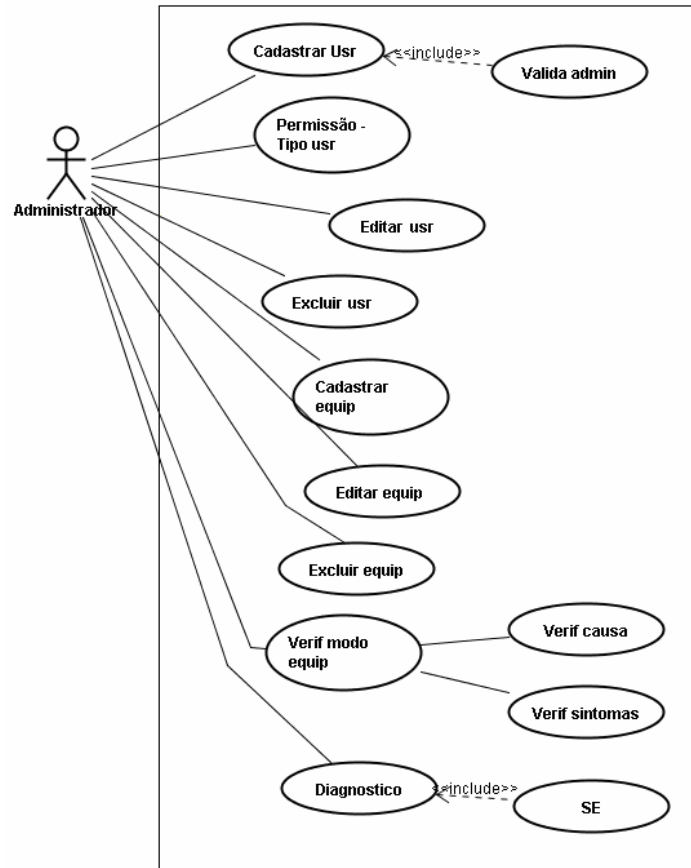


Figura 3.11: Diagrama de caso de uso

O diagrama de classe representado pela Figura 3.10 comparada a Figura 3.1 possui todas as características sendo as classes Equipamento e Função extensões da classe Sistema, representado por algum subsistema e as classes Solução, Falha, Causas, Sintoma e Arquivo extensões da classe Aplicação\_IA, representando a aplicação de uma técnica de IA. Já o diagrama de caso de uso (Figura 3.11), que indica algumas funcionalidades e requisitos do *software* do SIS, especifica as permissões que o administrador possui.

### 3.1.2 Descrição do Simulador do SIS

Com base nas falhas típicas em sistemas de refrigeração [69] e na finalidade do trabalho, foi decidido trabalhar com Sistemas Especialistas, especificamente Regras de Produção, conforme discutido na seção 2.3.2.



No *software* do SIS há nos menus de navegação: equipamento e monitor que serão utilizados no Simulador do SIS. No menu Equipamento, durante o cadastro de um equipamento é informado o seu estado (indicando se está em funcionamento ou não) para a simulação de um equipamento com defeito. No menu Monitor é possível listar todos os equipamentos que possuem falhas e, através de um interface amigável, permite ao operador alterar o estado do equipamento.

## **4 – APLICAÇÃO DE INTELIGÊNCIA ARTIFICIAL EM UM SUBSISTEMA PREDIAL**

Dentre os propósitos de um Sistema Inteligente de Supervisão (SIS) tem a centralização das informações referentes ao funcionamento dos diversos subsistemas vitais às operações do edifício. Porém é possível otimizar um subsistema que ainda não possui automação.

Neste capítulo é apresentado uma técnica de Inteligência Artificial no processo de otimização, baseada em Lógica *Fuzzy*, voltada para um subsistema predial, o elevador, e sua validação foi realizada através de simulações.

### **4.1 SIMULADOR DO ELEVADOR**

Para a simulação do conjunto de elevadores, aplicou-se a Lógica *Fuzzy* a um sistema de simulação de elevadores disponibilizado em [70]. A realização deste trabalho baseou-se no modelo apresentado em [71], assim como algumas de suas regras *Fuzzy*.

O motor de inferência utilizado foi o *FuzzyF* [72], um conjunto de APIs (*Framework*) com código fonte aberto que permite o uso de regras *Fuzzy* em aplicações Java.

Nos cenários utilizados tem-se inicialmente os elevadores no andar térreo do prédio. O passageiro, através de um painel, seleciona o botão indicando o andar que deseja ir e, com o uso da Lógica *Fuzzy*, são definidas prioridades para cada elevador, sendo que o de maior prioridade será o escolhido. O simulador permite estimar o tempo utilizado na viagem do passageiro.

Nas simulações, trabalhou-se com um cenário com parâmetros fixos e variáveis. Foi fixado o número de elevadores em três e a capacidade de cada elevador em oito pessoas. Os seguintes parâmetros foram variados: número de andares e usuários. Foram simulados prédios de dez, quinze e trinta andares, e foi adotada uma distribuição aleatória das pessoas pelos andares.

O simulador de elevadores permite a visualização gráfica do fluxo de passageiros e de elevadores pelos andares, como pode ser visto na Figura 4.1.



Figura 4.1: Tela de saída do simulador

O simulador possui um módulo de controle simples implementado, onde o elevador mais próximo é requisitado a cada chamada. Esse controle simples é utilizado como parâmetro de comparação com o controle *Fuzzy* desenvolvido, conforme apresentado na Figura 4.1.

## **5 - RESULTADOS E DISCUSSÃO**

Neste capítulo serão abordados o desenvolvimento e a validação, através de simulações, dos temas relatados no trabalho.

### **5.1 DESENVOLVIMENTO DE UMA ARQUITETURA DE SISTEMA SUPERVISÃO**

O desenvolvimento de uma arquitetura de um SIS espera conseguir reduções nos custos de manutenção, aumentando a produtividade e assim maximizar o uso eficaz do edifício.

Através dos recursos de simulações, que permite avaliar reações e comportamentos antes que uma solução seja implementada em sua plenitude, foram realizados para o desenvolvimento do trabalho, apresentado a seguir.

#### **5.1.1 Validação**

Durante o cadastro de um equipamento, no Simulador do SIS, é informado o seu estado (indicando se está em funcionamento ou não) para a simulação de um equipamento com defeito. No menu Monitor há um submenu denominado Listagem de Equipamento, que apresenta todos os equipamentos nos quais há falhas, permitindo ao operador um melhor controle operacional e possibilitando alteração do estado de um equipamento (Figura 5.1).



Figura 5.1: Monitoramento de equipamentos do Simulador do SIS

Cada equipamento (ar condicionado ou lâmpada) possui um item denominado Soluções (Figura 5.2) no qual o operador, através de uma interface (IHM) amigável e simples, denomina um sintoma para o equipamento selecionado (Figura 5.3). Só é permitida a seleção de apenas um sintoma para cada equipamento.



Figura 5.2: Menu Monitor do Simulador do SIS



Figura 5.3: Listagem dos possíveis sintomas de um determinado equipamento

O diagnóstico de falhas inicialmente foi definido pelo *shell* Expert SINTA, que contém as variáveis utilizadas e seus respectivos valores, que definem as regras de produção do SE e posteriormente integrado ao *software* do SIS (Figura 5.4).

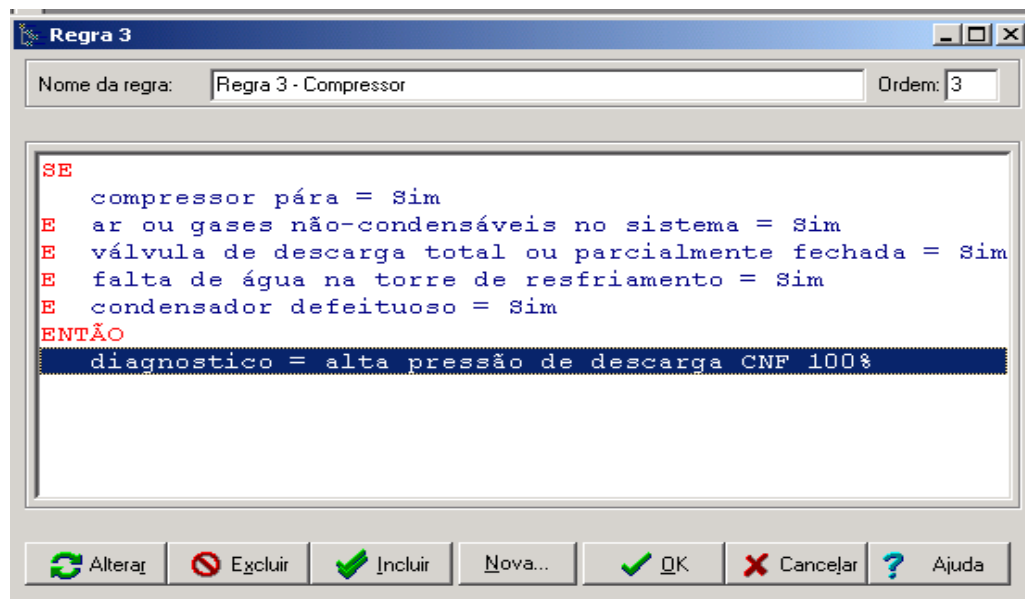


Figura 5.4: Exemplo de regra estabelecida no SE (Expert SINTA)

O operador informa as causas do sintoma através de uma interface do *software* do SIS (Figura 5.5). O SE compara então as causas informadas com as regras de produção

carregadas, o que resulta na definição do melhor caminho para uma possível solução, conforme apresentado na Figura 5.6.

**Simulador do Sistema Inteligente de Supervisão (SIS)**

Home Usuarios Equipamentos Monitor Relatorios Sair

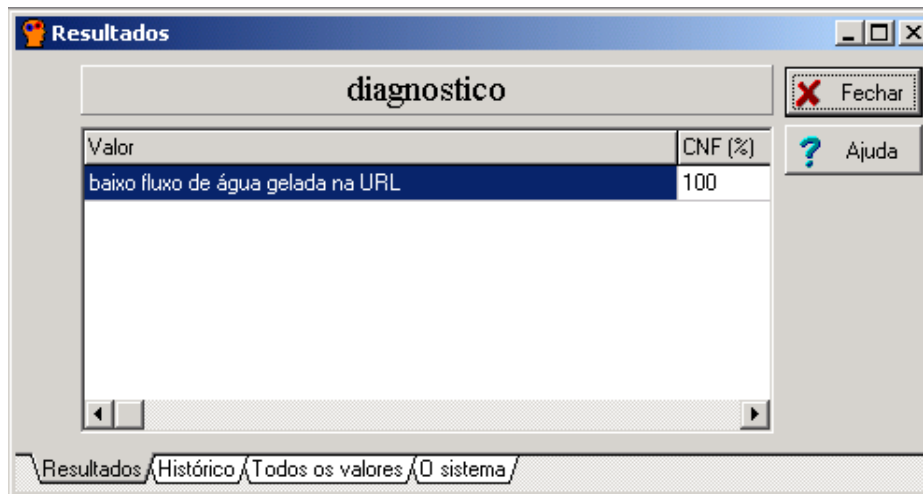
Escolha as causas do sintoma:

- válvula de óleo desajustada
- filtro entupido
- baixo nível de óleo
- defeito na bomba de óleo
- ar ou gases não-condensáveis no sistema
- válvula de descarga total ou parcialmente fechada
- falta de água na torre de resfriamento
- condensador defeituoso
- parcial ou completo congelamento do evaporador  
válvula
- solenóide defeituosa
- defeito mecânico na válvula de expansão
- ajuste errado no controle do evaporador
- má troca de calor no condensador
- sujeira no evaporador
- evaporador pequeno
- vazamento de refrigerante
- sujeira no radiador de óleo
- óleo no refrigerante
- superaquecimento excessivo
- alta temperatura de descarga

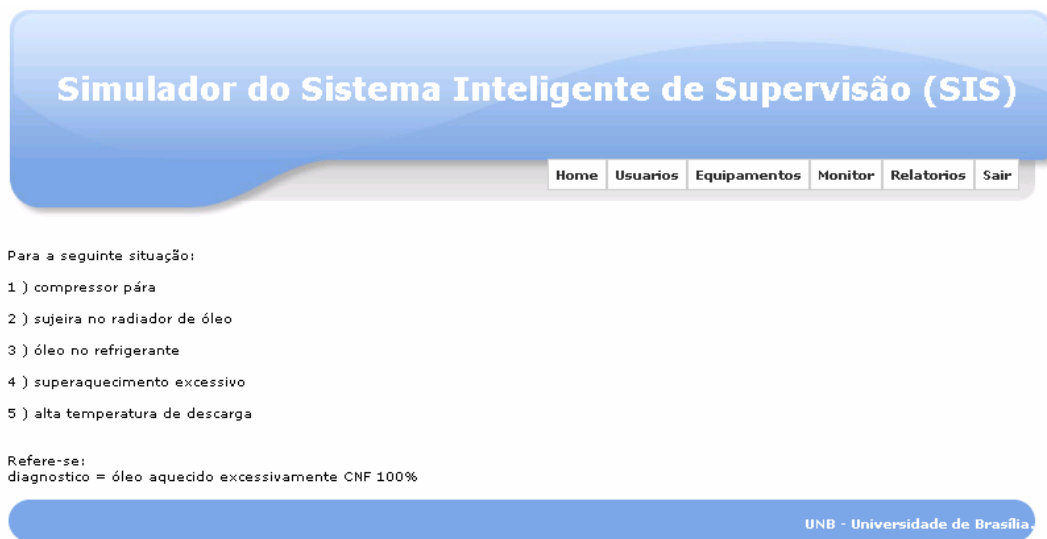
Encontrar Falha

UNB - Universidade de Brasília

Figura 5.5: Características estabelecida pelas Regras de Produção



(a)



(b)

Figura 5.6: Resultado de uma consulta ao (a) Expert SINTA e (b) *Software* do SIS

Caso o operador informa de maneira errada as causas do sintoma ao Simulador do SIS (Figura 5.7) o SE, após a comparação das causas informadas com as regras de produção carregadas, informa que o diagnóstico não é reconhecido pelo mesmo, conforme apresentado na Figura 5.8.



## Simulador do Sistema Inteligente de Supervisão (SIS)

[Home](#) [Usuarios](#) [Equipamentos](#) [Monitor](#) [Relatorios](#) [Sair](#)

Escolha as causas do sintoma:

- válvula de óleo desajustada
- filtro entupido
- baixo nível de óleo
- defeito na bomba de óleo
- ar ou gases não-condensáveis no sistema
- válvula de descarga total ou parcialmente fechada
- falta de água na torre de resfriamento
- condensador defeituoso
- parcial ou completo congelamento do evaporador
- válvula
- solenóide defeituosa
- defeito mecânico na válvula de expansão
- ajuste errado no controle do evaporador
- má troca de calor no condensador
- sujeira no evaporador
- evaporador pequeno
- vazamento de refrigerante
- sujeira no radiador de óleo
- óleo no refrigerante
- superaquecimento excessivo
- alta temperatura de descarga

UNB - Universidade de Brasília

Figura 5.7: Características estabelecidas pelas Regras de Produção - Erro

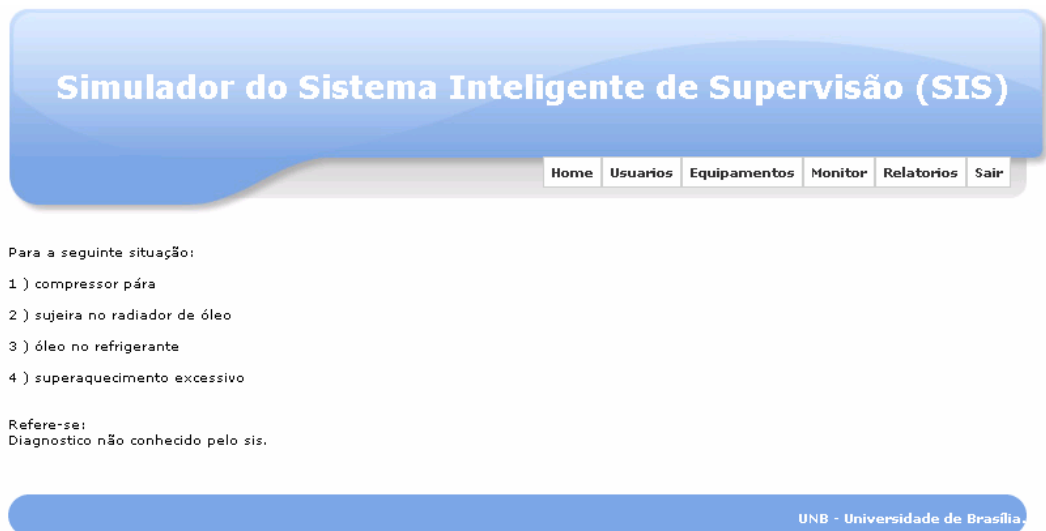


Figura 5.8: Resultado de uma consulta ao Simulador do SIS - Erro

As simulações realizadas no protótipo mostram que o conhecimento apresentado através do *software* do SIS correspondeu aos pressupostos para o qual foi avaliado. E que poderia ser aplicados a outros subsistemas de automação predial, em função da definição de regras específicas para cada um.

Também foi avaliado o simulador do elevador e apresentado sua eficiência no item seguinte.

## 5.2 OTIMIZAÇÃO DO SISTEMA DO ELEVADOR

Visando a otimização de um subsistema, o elevador, foi aplicado Lógica *Fuzzy* e sua validação foi realizada através de simulações, que será apresentado a seguir.

### 5.2.1 Validação

No simulador do elevador, o controle *Fuzzy* implementado utiliza como variáveis de controle o tempo de espera do passageiro, tempo estimado da viagem do passageiro, distância entre o elevador e o passageiro e número de passageiros dentro do elevador. Os gráficos de domínio dessas variáveis são apresentados na Figura 5.9. A saída do motor de inferência *Fuzzy* é um número indicativo da prioridade de determinado elevador para atender determinado chamado. Assim o elevador com a maior prioridade será o escolhido.

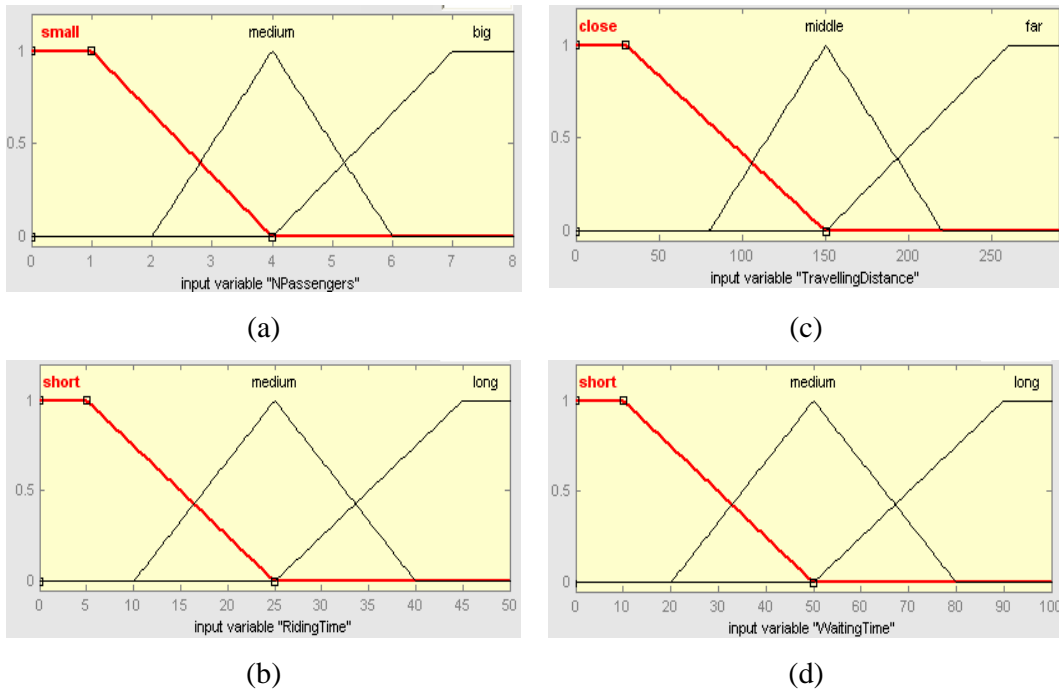


Figura 5.9: Funções de Pertinência: (a) número de passageiros; (b) tempo de viagem; (c) distância da viagem; e (d) tempo de espera.

As regras usadas pelo módulo de inferências são apresentadas na Figura 5.10.

```
#Rules defintions
#Rules to priority;
BLOCK_RULES
RL = IF Waiting IS Short THEN Priority IS Big CV 0.7
RL = IF Waiting IS Medium THEN Priority IS Medium CV 0.7
RL = IF Waiting IS Long THEN Priority IS Small CV 0

RL = IF Riding IS Short THEN Priority IS Big CV 0.4
RL = IF Riding IS Medium THEN Priority IS Medium CV 0.4
RL = IF Riding IS Long THEN Priority IS Small CV 0.4

RL = IF Distance IS Close THEN Priority IS Big CV 0.05
RL = IF Distance IS Middle THEN Priority IS Medium CV 0.05
RL = IF Distance IS Far THEN Priority IS Small CV 0.05

RL = IF Passengers IS Small THEN Priority IS Big CV 0.5
RL = IF Passengers IS Medium THEN Priority IS Medium CV 0
RL = IF Passengers IS Big THEN Priority IS Small CV 0.6

END_RULES

#Finish domain
```

Figura 5.10: Regras *Fuzzy*

Uma das maiores dificuldades encontradas para a implementação desse trabalho foi definir as regras do módulo de inferência adequadamente. Foram realizadas diversas

simulações até que os resultados obtidos, através do controle *Fuzzy*, fossem superiores ao controle simples. Os resultados conseguidos são apresentados nas Tabelas 5.1, 5.2 e 5.3.

Tabela 5.1: Três elevadores e dez andares

Número de Passageiros	Tempo médio de espera (s)	Distância média percorrida pelo elevador (m)
20	68,702	70,10
	50,421	50,60
30	80,119	79,25
	69,68	65,84
40	87,624	77,11
	83,731	711,93
50	85,682	78,03
	83,765	74,07
60	94,173	71,02
	85,308	71,02
100	155,735	98,45
	140,671	90,22
<b>Médias</b>		
50	95,339	78,99
	85,596	70,61

Legenda

	Simple
	Nebulosa

Tabela 5.2: Três elevadores e quinze andares

Número de Passageiros	Tempo médio de espera (s)	Distância média percorrida pelo elevador (m)
20	85,533	104,55
	63,589	83,21
30	103,772	112,78
	80,489	81,08
40	109,271	106,68
	106,844	105,46
50	113,641	105,46
	111,708	107,59
60	125,009	106,68
	111,684	99,36
100	181,443	123,75
	172,775	115,82
Médias		
50	119,778	109,98
	107,848	98,76

Legenda

	Simplex
	Nebulosa

Tabela 5.3: Três elevadores e trinta andares.

Número de Passageiros	Tempo médio de espera (s)	Distância média percorrida pelo elevador (m)
20	136,661	181,86
	135,898	142,24
30	140,214	185,93
	145,134	185,93
40	176,959	216,41
	173,178	202,18
50	184,947	206,25
	186,362	209,30
60	189,285	187,96
	206,951	210,31
100	327,62	285,50
	299,371	257,05
Médias		
50	192,614	210,65
	191,149	201,17

Legenda

	Simplex
	Nebulosa

As Tabelas mostram que, na maioria dos casos, o controle *Fuzzy* apresenta um desempenho consideravelmente superior ao controle simples. Em apenas algumas exceções, como por exemplo, no caso de três elevadores e trinta andares com 50 e 60 passageiros, os resultados do controle simples são superiores (Tabela 5.3). Isso ocorre devido as regras determinadas, alterando o valor dessas regras pode obter o controle *Fuzzy* superior

A superioridade do controle *Fuzzy* é especialmente evidente nos casos de três elevadores com dez e quinze andares (Tabela 5.1 e Tabela 5.2). Na Tabela 5.1, por exemplo, o tempo médio de espera é reduzido em 10,21% e a distância média, em 10,60%, o que conseqüentemente resulta na redução do consumo de energia elétrica.

## 6 - CONCLUSÕES E RECOMENDAÇÕES

Neste capítulo são apresentadas as conclusões do trabalho, e as sugestões de novos tópicos a serem pesquisados em futuros trabalhos.

O desenvolvimento do presente trabalho foi realizado em torno de uma aplicação para um sistema do prédio do TJDF (Tribunal de Justiça do Distrito Federal), que já possuía automação aplicada aos sistemas de refrigeração, ar condicionado e iluminação, utilizando técnicas de IA.

A aplicação de Sistemas Especialistas utilizando Regras de Produção nos subsistemas ar condicionado e iluminação resultou no acréscimo de informações que auxiliarão o operador do sistema nas tomadas de decisão, facilidades decorrentes do controle operacional eficaz. Também diminuiu os custos de manutenção por meio da gestão integrada através do diagnóstico de falhas dos equipamentos e prolongamento da vida útil dos equipamentos instalados nas edificações. Além do que outros subsistemas de automação predial também podem ser otimizados, desde que sejam utilizadas regras específicas para cada um.

Foi aplicada Lógica *Fuzzy* para otimizar a alocação de elevadores. De acordo com os resultados obtidos, notou-se que o controle *Fuzzy* possui um desempenho consideravelmente superior ao do controle simples. Foi possível notar uma redução considerável tanto do tempo de espera dos usuários do edifício quanto da distância média percorrida pelos elevadores, inferindo-se, neste último caso, uma diminuição no consumo de energia.

As maiores dificuldades encontradas na implementação deste trabalho foram a definição adequada das regras e a busca para o embasamento do diagnósticos e a prescrição recomendada.

Também a realização de um exaustivo levantamento bibliográfico sobre automação, especialmente predial, e o desafio da proposta de uma arquitetura genérica podendo ser aplicada a qualquer sistema de gerenciamento e supervisão.

Em relação a atividades futuras, o sistema desenvolvido pode ser aprimorado aplicando-se o controle *fuzzy* no desenvolvimento de um sistema especialista para controle e gerenciamento automático. Além disso, um demonstrador conceitual para os subsistemas apresentados está previsto para operar no segundo andar do Palácio do TJDF. Após a implantação e testes, uma solução completa para os prédios do TJDF pode ser elaborada incluindo outros subsistemas. Também deverá ser realizada a integração entre eles.

Ainda, em relação a atividades futuras, já que o sistema tem uma arquitetura geral e permite sua aplicação em diversos sistemas de supervisão, sua implantação no Laboratório de Dispositivos de Circuito Integrado, LDCI na execução de um projeto de circuitos integrados.



## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] - Pressman, R. S. (2005). *Software engineering: a practitioner's approach*, McGraw-Hill, Columbus, E.U.A.
- [2] – Rumbaugh, J. (1994). *Modelagem e Projetos Baseados em Objetos*, Campus, São Paulo, São Paulo.
- [3] – Fernandes, J.H.C. (2004). *Introdução À Engenharia de Software Com Foco no RUP - Rational Unified Process*. (Notas de aula) Disponível em: <<http://www.cic.unb.br/~jhcf/MyBooks/iess/RUP/IntroducaoaEngenhariadeSoftwarecomFoconoRUP-ParteI-138slides.pdf>>. Acesso em: 25 agosto de 2007.
- [4] – Cheesman, J. e Daniels, J. (2001). *UML Components – A Simple Process for Specifying Component-Based Software*, Addison-Wesley, Boston, E.U.A.
- [5] – Boaventura, I. A. G. (2007). *Introdução à Orientação a Objetos e UML*. - UNESP Disponível em: <[http://www.dcce.ibilce.unesp.br/~ines/cursos/proj-si/2007/aula02\\_IntrOOeUML.pdf](http://www.dcce.ibilce.unesp.br/~ines/cursos/proj-si/2007/aula02_IntrOOeUML.pdf)>. Acesso em: 22 maio de 2007.
- [6] – Souza, V. (2006). "*Introdução à Orientação a Objetos*". (Notas de Aula da Universidade Federal do Espírito Santo). Disponível em: <<http://www.inf.ufes.br/~vsouza/files/CursoOOSlides03.pdf>>. Acesso em: 15 de agosto de 2007.
- [7] – Ricarte, I.L.M. (2002). "*Programação Orientada a Objetos - Uma abordagem com Java*". Disponível em: <<http://www.dca.fee.unicamp.br/cursos/PooJava/objetos/conceito.html>>. Acesso em: 10 de julho de 2007.
- [8] – Souza, C. (sem data). "*Conceitos de Orientação a Objetos*". (Notas de Aula da Universidade Federal do Pará). Disponível em:

<<http://www2.ufpa.br/cdesouza/teaching/es/3-OO-concepts.pdf>>. Acesso: em 10 de julho de 2007.

[9] – Carvalho, L. F. S. (2002). *BLOOM – BLAST Object Oriented Management: uma solução integrada para gerenciamento dos resultados do BLAST por meio de um paradigma orientado a objetos*, Dissertação de Mestrado, Universidade Católica de Brasília, 203 p. Disponível em: <<http://www.biofoco.org/wedo/bloom.pdf> >. Acesso em: 10 de novembro de 2007.

[10] – Booch, G., Rumbaugh, J. e Jacobson, I. (1999). *The Unified Modeling Language User Guide*, Addison-Wesley Longman, Inc, Boston, E.U.A.

[11] – Jacobson, I. (1994). *Object-Oriented Software Engineering: A Use Case Driven Approach*, Addison-Wesley, Boston, E.U.A.

[12] – Schneider, G., Winters, J.P. e Jacobson. I. (1998) *Applying Use Cases: A Practical Guide*, Addison-Wesley, Boston, E.U.A.

[13] – Alcântara, A.A., Maciel, T. M. M., Meira, S.L. e Silva, F.Q.B. (2005). *Uso do Processo RUP na Implantação Da ISO9000-3*. Disponível em: <[http://inf.unisul.br/~vera/egs/Rup\\_iso9000.pdf](http://inf.unisul.br/~vera/egs/Rup_iso9000.pdf)>. Acesso em: 15 de maio de 2007.

[14] –RUP - Rational Unified Process

Disponível em: <<http://www.wthreex.com/rup/>>. Acesso em: 16 de agosto de 2007.

[15] – IBM Rational Unified Process - RUP

Disponível em: <<http://www-306.ibm.com/software/awdtools/rup/>>. Acesso em: 16 de fevereiro de 2007.

[16] – Piske, O.R. (2003). *RUP - Rational Unified Process*. (Notas de Aula) Disponível em: <[http://www.angusyong.org/arquivos/artigos/trabalho\\_rup.pdf](http://www.angusyong.org/arquivos/artigos/trabalho_rup.pdf)>. Acesso em: 16 de fevereiro de 2007.

- [17] – Broering, E. (2007). *RUP - Rational Unified Process*. (Artigo) Disponível em: <<http://www.devmedia.com.br/articles/viewcomp.asp?comp=4574>>. Acesso em: 20 de outubro de 2007.
- [18] – Silva, M. R. (2004). *Processos de desenvolvimento de software: RUP e ICONIX*, Monografia, Universidade Estadual de Londrina, 51 p. Disponível em: <<http://www.dc.uel.br/nou-rau/document/?code=198>>. Acesso em: 25 de julho de 2007.
- [19] - Leite, J. C. (sem data). *RUP - Rational Unified Process*. (Apresentação) Disponível em: <<http://www.dimap.ufrn.br/~jair/ES/slides/rup.pdf>>. Acesso em: 20 de maio de 2007 .
- [20] – Deitel, H. M. e Deitel P. J. (2005). *Java: Como Programar*, Pearson Prentice Hall, New Jersey, E.U.A.
- [21] – Horstmann, C. S. e Cornell, G. (2004). *Core Java (TM) 2, Volume I-Fundamentals*, Prentice Hall PTR, New Jersey, E.U.A.
- [22] – Eckel, B. (2006). *Thinking in Java*, Prentice Hall, New Jersey, E.U.A.
- [23] – Jandl Jr, Pr. (1999). *Introdução ao Java*. (Núcleo de Educação a Distância da Universidade São Francisco). Disponível em: <<http://www.abusar.org/ftp/pub/pitanga/ij.pdf>>. Acesso em: 25 de maio de 2007 .
- [24] - Cassandras, C G. e Lafortune, S. (1999). *Introduction to discrete Event Systems*, Kluwer Academic Publishers, Boston, USA
- [25] – Tibola, L. R. (1999). “*Geração de Sistemas Supervisórios a Partir de Modelos Orientados a Objetos de Aplicações Industriais*”. In: IV Semana Acadêmica do PPGC/UFRGS, UFRGS, Porto Alegre. Disponível em: <<http://www.inf.ufrgs.br/pos/SemanaAcademica/Semana99/tibola/tibola.html>>. Acesso em: 18 de outubro de 2007.

[26] – Amory, A. e Petrini, J.J. (2000) “*Sistema Integrado e multiplataforma para controle de residências*”, Monografia de Graduação, Pontifícia Universidade Católica do Rio Grande do Sul, 105 p.

[27] – Silva, A. P. G. e Salvador, M. (2005). “*O que são sistemas supervisórios?*” Disponível em: <[www.elipse.com.br/download/download/artigos/rt025.04.pdf](http://www.elipse.com.br/download/download/artigos/rt025.04.pdf)>. Acesso em: 18 de outubro de 2007.

[28] – Filho, C. S. (2007). “*Capítulo 3 – SCADA*”. (Curso de Informática Industrial). Disponível em: <<http://www.cpdee.ufmg.br/~seixas/PaginaII/Download/DownloadFiles/Scada.PDF>>. Acesso em: 11 de julho de 2007.

[29] – Stouffer, K. , Falco, J., Kent, K. (2006). “*Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security*”. (Recommendations of the National Institute of Standards and Technology - U.S. Department of Commerce). Disponível em < <http://csrc.nist.gov/publications/drafts/800-82/Draft-SP800-82.pdf>>. Acesso em: 15 de junho de 2007.

[30] – Penteado, F. e Kimi, L. (2007). “Sistema de Supervisão e Controle”. In: *Revista Climatização e Refrigeração*, 84, 52-55.

[31] – Castrucci, P.L. e Moraes, C.C (2001). “*Engenharia de Automação Industrial*”. LTC Livros Técnicos e Científicos Editores S.A., Rio de Janeiro.

[32] – Pinheiro, J. M. S. (2006) “*Introdução às Redes de Supervisão e Controle*”. Disponível em: <[http://www.projetoderedes.com.br/artigos/artigo\\_redes\\_de\\_supervisao\\_e\\_controle.php](http://www.projetoderedes.com.br/artigos/artigo_redes_de_supervisao_e_controle.php)>. Acesso em: 10 de maio de 2007.

[33] – Maia, W. U. (1998). “*Sistema Integrado de Operação e Diagnóstico de Falhas para Sistemas de Energia Elétrica : SODF*”. Disponível em: < <http://uvirtual.eps.ufsc.br/disserta98/maia>>. Acesso em: 10 de maio de 2007.

- [34] – Villani, E., Miyagi, P.E., Santos Filho, D.J. dos, e Maruyama, N. (2000) “Modelagem de Sistemas de ar condicionado em Edifícios Inteligentes”. In: *Revista Controle e Instrumentação*, 5(46), 76-86.
- [35] – Shaw, A. (ed) (1989). “*Energy design for architects*”, The American Architectural Foundation, Prentice Hall, New York, EUA.
- [36] – Sousa, M. A. B. (2004). “*Estudo comparativo entre a aplicação de sistemas dedicados e a utilização de controladores lógico-programáveis na automação de sistemas prediais*”, Dissertação de Mestrado, Escola Politécnica da Universidade de São Paulo, 127 p. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3146/tde-26052004-154102/>>. Acesso em: 10 de agosto de 2007.
- [37] – Villani, E. (2000). “*Abordagem Híbrida para Modelagem de Sistemas de Ar Condicionado Em Edifícios Inteligentes*”, Dissertação de Mestrado, Escola Politécnica da Universidade de São Paulo, 154p. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3132/tde-26032001-154604/>>. Acesso em: 10 de agosto de 2007.
- [38] – Graça, M. E. A. (2006). “*SISTEMAS PREDIAIS II – Condicionamento de ar I*”. (Notas de Aula da Escola Politécnica da Universidade de São Paulo). São Paulo. Disponível em: <<http://pcc2466.pcc.usp.br/Apostilas/PCC2466-condicionamento%20de%20ar%20II.pdf>>. Acesso em: 12 de agosto de 2007.
- [39] – Lima, F. P., Radskevicz, R. e Varela, T. D. (2006). “*Bancada para Simulação em Automação Predial*”, Projeto de Graduação, Universidade Tecnológica Federal do Paraná, 129 p. Disponível em: <[www.pessoal.cefetpr.br/winder/AR](http://www.pessoal.cefetpr.br/winder/AR)>. Acesso em: 10 de agosto de 2007.
- [40] – Osram (2007). “*Catálogo Geral da Linha de Produtos da OSRAM*” (2007). Disponível em: <[http://br.osram.info/download\\_center/download\\_catalogo.htm](http://br.osram.info/download_center/download_catalogo.htm)>. Acesso em: 10 de agosto de 2007.

- [41] – Otis (2006). “*Otis Corporation*”. Disponível em: <<http://www.otis.com>>. Acesso em: 15 de abril de 2005.
- [42] – Siikonen, M.L. (1997). “*Elevator Group Control with Artificial Intelligence*”, Systems Analysis Laboratory Research Reports, A67, Helsinki University of Technology, Helsinki.
- [43] – Bronowski, J. (1992). “*A Escalada do Homem*”, Martins Fontes Editora, São Paulo.
- [44] – Gustin, G. D. B. (1999). “*Aplicação de Redes de Petri Interpretadas na Modelagem de Sistemas de Elevadores em Edifícios Inteligentes*”, Dissertação de Mestrado, Escola Politécnica da Universidade de São Paulo, 139 p. Disponível em: <http://www.teses.usp.br/teses/disponiveis/3/3132/tde-30032001-150611/>>. Acesso: em 18 de julho de 2006.
- [45] – Luger G.F. e Stubblefield W.A. (1997). “*Artificial Intelligence: Structure and Strategies for Complex Problem Solving*”, Addison-Wesley Longman, Boston, EUA.
- [46] – Bauchspiess, A. (2004). “*Introdução aos Sistemas Inteligentes - Aplicações em Engenharia de Redes Neurais Artificiais, Lógica Fuzzy e Sistemas Neuro-Fuzzy*”. (Apostila de Aula da Universidade de Brasília). Brasília. Disponível em: <<http://www.ene.unb.br/~adolfo/ISI/isi.pdf>>. Acesso em: 18 de junho de 2006.
- [47] – Sousa F. L. (2003) “*Otimização Extrema Generalizada: Um novo Algoritmo Estocástico para o Projeto Ótimo*”, Tese de Doutorado, Instituto Nacional de Pesquisas Espaciais, 142p.
- [48] – Arora, J.S. (1989) “*Introduction to optimum design*”. MacGraw-Hill Series in Mechanical Engineering, New York, E.U.A.
- [49] – Johnson, R.C. (1978). “*Mechanical design synthesis – Creative design and optimization*”. Robert E. Krieger Publishing Company, New York, EUA.

- [50] – Michalewicz, Z. e Fogel, D.B. (2000). “*How to solve it: Modern heuristics*”. Springer-Verlag publishing, Berlin, Alemanha.
- [51] – Pardalos, P.M. e Romeijn, H.E. (ed). “*Handbook of global optimization*”. Kluwer Academic Publishers, Dordrecht, Holanda.
- [52] – Reklaitis, G.V., Ravindran, A. e Ragsdell, K.M. (1983) “*Engineering optimization – Methods and applications*”. John Wiley and Sons, New York, E.U.A.
- [53] – Vanderplaats, G.N. (1998). “*Numerical optimization techniques for engineering design*”. Vanderplaats Research & Development, Colorado Springs, E.U.A.
- [54] – Wield, D.J. (1978). “*Globally optimal design*”. John Wiley & Sons, New York, E.U.A.
- [55] – Wismer, D.A.; Chattergy, R. (1979). “*Introduction to Nonlinear Optimization*”. North-Holland Publishing Company, Holanda.
- [56] – Wolpert, D.H. e Macready, W.G. (1995). “*No Free Lunch Theorems for Search*”, SFI-TR-95-02-010, Santa Fe Institute Technical Report, Santa Fe, E.U.A.
- [57] – Eldred, M.S. (1998). “*Optimization strategies for complex engineering applications*”, SANDIA Report - SAND98-0340 - UC-705, Sandia National Laboratories, Albuquerque, E.U.A.
- [58] – Romariz, A. R. S. (2006). “*Sistema Inteligente*”. Disponível em: <<http://www.ene.unb.br/~romariz/>>. Acesso em: 25 de maio de 2006.
- [59] – Thomé, A. C. G. (2003). “*Inteligência Computacional - Visão Geral de Inteligência Computacional e Redes Neurais*”. (Notas de aula da Universidade Federal do Rio de Janeiro) Disponível em: <[http://equipe.nce.ufrj.br/thome/grad/nn/mat\\_didatico/aula4.pdf](http://equipe.nce.ufrj.br/thome/grad/nn/mat_didatico/aula4.pdf)>. Acesso em: 4 de março de 2007.

[60] – Camargo, H. A. (2005). “*Sistemas Nebulosos*”. (Notas de Aula da Universidade Federal de São Carlos) Disponível em: < <http://www.dc.ufscar.br/~heloisa/COF.pdf> >. Acesso em: 26 de julho de 2006.

[61] – Bittencourt, G. (2001). “*Inteligencia Artificial- Ferramentas e Teorias*” 2 ed. Florianópolis, Editora da UFSC 362 p.

[62] – Camargo, K. G. (1999). “*Inteligência Artificial aplicada à Nutrição na Prescrição de Planos Alimentares*”, Dissertação de Mestrado, Universidade Federal de Santa Catarina. Disponível em: <<http://www.eps.ufsc.br/disserta99/camargo/index.html>>. Acesso em: 26 jul. 2006.

[63] – LIA – Laboratório de Inteligência Artificial da Universidade Federal do Ceará (1998). “*Software Expert SINTA – Manual do Usuário*”. Disponível em:<<http://www.lia.ufc.br/~bezerra/exsinta/exsintashell.htm>> Acesso em: 05 de outubro de 2006.

[64] – Cardoso et all. (2004). “Um Sistema Especialista para Apoio à Decisão em Exames Ortopédicos de Ombro, Cotovelo e Punho”. In: *IX Congresso Brasileiro de Informática*, Ribeirão Preto, Brasil. Disponível em: <<http://www.sbis.org.br/cbis9/arquivos/648.pdf> >. Acesso em: 19 de abril de 2007.

[65] – Guarda, A. (2006). “*Inteligência Artificial*”. (Notas de Aula da Universidade Federal de Ouro Preto). Disponível em: <[www.decom.ufop.br/prof/guarda/CIC250/Apostila\\_IA.pdf](http://www.decom.ufop.br/prof/guarda/CIC250/Apostila_IA.pdf) >. Acesso em: 19 de abril de 2007.

[66] – Waterman, D.A. (1986). “*A Guide to Expert System*”. Addison-Wesley Publishing Company, Boston, E.U.A.

[67] – Caetano, K.C, Peres, H.H.C. e Fugulin, F.M.T. (2007). “An expert system prototype for the classification of the complexity of the assistance in nursing”, *Online Brazilian Journal of Nursing*, 6(1). Disponível em:



<<http://www.uff.br/objnursing/index.php/nursing/rt/printerFriendly/735/167>> Acesso em: 20 de outubro de 2007.

[68] – Mendes R.D. (1997). “Inteligência Artificial: Sistemas especialistas no gerenciamento da informação”, *Ciência da Informação*, 26(1). Disponível em: <<http://www.ibict.br/cienciadainformacao/viewarticle.php?id=458>>. Acesso em: 01 de dezembro de 2006.

[69] – Carvalho, C. M. (2005). "*Detecção e diagnóstico de falha em sistemas de refrigeração e ar condicionado*", Monografia, Universidade de Brasília, 70 p. Disponível em: <[http://www.laar.unb.br/Relatorios\\_PG/Relatorio\\_final\\_Celso\\_Carvalho.pdf](http://www.laar.unb.br/Relatorios_PG/Relatorio_final_Celso_Carvalho.pdf)>. Acesso em: 10 de abril de 2007.

[70] –McKellar, N. e Dailey, C. (2005). “*Elevator Simulator*”. Disponível em: <<http://sourceforge.net/projects/elevatorsim> >. Acesso em: 25 de julho de 2006.

[71] - [3] Nunes, R. (2005). “*Integração de Serviços para Edifícios Inteligentes*”. (Notas de Aula da Universidade Técnica de Lisboa). Disponível em: <<http://mega.ist.utl.pt/~ic-eid/cap4.pdf>>. Acesso em: 25 de julho de 2006.

[72] - Bittencourt, J. R. (2001). “*Fuzzy Logic Framework*”. Disponível em: <<http://www.inf.unisinos.br/~jrbitt/fuzzyf/>>. Acesso em: 25 jul. 2006.

[73] – Santos, W. E., Lima, F.P., Radaskievicz, R. e Varela, T. D. (2006). “*Bancada para Simulação em Automação Predial*”, Monografia de Graduação, Universidade Tecnológica Federal do Paraná, 129 p.

## **APÊNDICES**

## A – Possíveis intervenções nos sistemas do TJDF

Esse apêndice apresenta um quadro no qual mostra que a automação já existente no prédio do TJDF pode ser estendida a outros sistemas e inserida no sistema de gerenciamento inteligente.

SUBSISTEMA	OBJETIVO	INTERVENÇÕES POSSÍVEIS	EQUIPAMENTOS NECESSÁRIOS
<b>Climatização</b>	<ul style="list-style-type: none"> <li>- Garantir conforto térmico (ar condicionado), renovação do ar no interior do prédio (exaustão e ventilação) consumindo o mínimo de energia.</li> <li>- Controlar a frequência de funcionamento dos equipamentos.</li> </ul>	<ul style="list-style-type: none"> <li>- Ar condicionado/ ventilação/ aquecimento: Controle dos sistemas em situações de emergência - iluminação nas áreas vizinhas do sinistro, por exemplo, em caso de incêndio.</li> <li>- Controle do funcionamento dos equipamentos de acordo com a necessidade.</li> <li>- Programação de horários, estações do ano.</li> <li>- Funcionamento: regulagem manual, automática local, automática geral.</li> </ul>	<ul style="list-style-type: none"> <li>- Sensores (de vazão, de temperatura, de pressão), atuadores, controladoras, computadores com programa de gerenciamento do sistema.</li> </ul>
<b>Comunicação</b>	<ul style="list-style-type: none"> <li>- Fornecer meio de comunicação eficiente (trânsito de qualquer tipo de informação: dados, voz, sinais ou imagens).</li> <li>- Disponibilizar dados, programas e recursos a todos, desconsiderando a localização física do recurso e do usuário.</li> <li>- Baratear processamento.</li> </ul>	<ul style="list-style-type: none"> <li>- Acesso Internet.</li> <li>- Circuito Fechado de Televisão - CFTV.</li> <li>- Comunicação de dado.</li> <li>- Controle da distribuição de música ambiente.</li> <li>- Interfone e vídeo portaria.</li> <li>- Intranet.</li> <li>- Redes de computadores.</li> <li>- Telefonia móvel/ fixa.</li> <li>- TV a cabo e satélite.</li> <li>- Videoconferência.</li> <li>- Funcionamento: regulagem manual, automática local, automática geral.</li> </ul>	<ul style="list-style-type: none"> <li>- Cabos (tipos de cabos) ou <i>wi-fi</i> (equipamentos <i>w-fi</i>), equipamentos de redes (<i>hubs</i>, servidores...), protocolos de comunicação, TV, comunicação via satélite, câmaras, telefones, aparelhos transmissores, computadores, entre outros.</li> </ul>

<b>Diagnóstico e Manutenção</b>	- Controlar a frequência de funcionamento dos equipamentos, evitando ou minimizando os possíveis danos.	- Diagnóstico de falhas. - Manutenção de sistema - Monitoração/ inspeção e teste do estado de funcionamento dos equipamentos. - Funcionamento: regulagem manual, automática local, automática geral.	- Todos os subsistemas envolvidos.
<b>Emergência</b>	- Evitar ou minimizar os danos provocados por alguma situação de emergência.	- Alarme de emergência. - Controle dos sistemas em situações de emergência - iluminação nas áreas vizinhas do sinistro. - Detecção, prevenção. - Detecção e combate ao incêndio. - Detecção de fumaça. - Monitoramentos (água, fogo, fumaça). - Funcionamento: regulagem manual, automática local, automática geral.	- Alarmes, controladores, monitores de zona, centrais de incêndio, computadores com programa de gerenciamento do sistema.
<b>Garagem/ Estacionamento</b>	- Restringir e monitorar o acesso a garagem. - Verificar disponibilidade de vagas.	- Abertura/ fechamento portão. - Entrada/ saída. - Monitorar. - Verificação de vagas. - Funcionamento: regulagem manual, automática local, automática geral.	- Portões automáticos com leitura de cartões ou biometria, leitoras de cartão, sensores, computadores com programa de gerenciamento do sistema.
<b>Gestão Energética</b>	- Disponibilizar energia para os equipamentos e consumindo o mínimo de energia.	- Distribuição. - Geração. - Otimização de operação.	- Lâmpadas, sensores, gerador, transformadores, disjuntores, <i>no-breaks</i> , <i>dimmer</i> , computadores com programa de

	- Iluminar os ambientes.	- Sistema de monitoramento/ análise. - Transmissão. - Funcionamento: regulagem manual, automática local, automática geral.	gerenciamento do sistema.
<b>Gestão de Presença</b>	- Restringir e monitorar o acesso a áreas restritas.	- Controle de acesso. - Disponibilidade da informação sobre tempos de trabalho, saldos horários. - Frequência. - Marcação de férias.	- Fechaduras magnéticas ou por biometria, leitoras de cartão, sensores, controladoras, computadores com programa de gerenciamento do sistema.
<b>Iluminação</b>	- Iluminar os ambientes, evitar o desperdício, localizando consumo.	- Interno/ Externo. - Exaustores. - Funcionamento: regulagem manual, automática local, automática geral.	- Lâmpadas, sensores, gerador, transformadores, disjuntores, no-breaks, dimmer, computadores com programa de gerenciamento do sistema
<b>Segurança</b>	- Proteger pessoas e pertences em caso de eventualidades como invasão, vazamentos, incêndios (situação de emergência). - Sistemas internos e externos interligados.	- Detecção, prevenção, arrombamento e intrusão. - Alarmes de emergência. - Cicruitos fechados de Televisão – CFTV - Controle de acesso. - Controle de estacionamento de veículos. - Detecção combate a incêndio. - Detecção de fuga de gás/ água/ combustível líquido. - Detecção de inundação. - Sistema de vídeo portaria. - Monitoramento. - Funcionamento: regulagem manual, automática local, automática geral.	- Sensores (por exemplo: fumaça, de presença, de portas), controladoras, detectores (fumaça, temperatura), alarmes, hot-lines, sirenes, sinalizações, circuito fechado de TV, câmaras, equipamentos de gestão de presença, no-breaks, computadores com programa de gerenciamento do sistema.

<p><b>Transporte</b></p>	<ul style="list-style-type: none"> <li>- Reduzir consumo mínimo de energia.</li> <li>- Controlar a frequência de funcionamento dos equipamentos.</li> </ul>	<ul style="list-style-type: none"> <li>- Elevadores/ Escadas/ Escadas Rolantes.</li> <li>- Controle dos sistemas em situações de emergência - elevadores se deslocam para pisos seguros, mudar o sentido das escadas/ escadas rolantes de modo a facilitar evacuação.</li> <li>- Monitorar o estado de funcionamento dos vários sistemas.</li> <li>- Funcionamento: Regulagem manual, automática local, automática geral.</li> </ul>	<ul style="list-style-type: none"> <li>- Sensores, botoeiras (na cabina e andar), motores, atuadores das portas, fotocélulas, detectores do peso nas cabinas, detectores de posição, controladoras, sinalizadores, alarmes sonoros, computadores com programa de gerenciamento do sistema.</li> </ul>
<p><b>Patrimonial</b></p>	<ul style="list-style-type: none"> <li>- Proteger pertences em caso de eventualidades como furto, roubo.</li> </ul>	<ul style="list-style-type: none"> <li>- Controle de cada equipamento em cada departamento.</li> <li>- Gestão de informações a seguros (por exemplo, incêndio).</li> <li>- Gestão do próprio edifício, como contrato de aluguel, prestação de serviço.</li> <li>- Registro do espaço ocupado por cada departamento.</li> <li>- Registro e gestão dos bens patrimoniais.</li> <li>- Funcionamento: Regulagem manual, automática local, automática geral.</li> </ul>	<ul style="list-style-type: none"> <li>- Identificadores do patrimônio, documentação de cada equipamento, computadores com programa de gerenciamento do sistema e contendo um banco de dados.</li> </ul>
<p><b>Portaria</b></p>	<ul style="list-style-type: none"> <li>- Proteger pessoas e pertences em caso de eventualidades.</li> </ul>	<ul style="list-style-type: none"> <li>- Controle e registro de entrada/ saída de pessoas e equipamentos</li> <li>- Fornecer informações sobre a organização do edifício</li> <li>- Identificação das pessoas</li> <li>- Manutenção preventiva e corretiva</li> <li>- Registro de mensagens de visitas</li> <li>- Funcionamento: regulagem manual, automática local, automática geral.</li> </ul>	<ul style="list-style-type: none"> <li>- Catraca (com leitura de cartões ou biometria), fechaduras magnéticas ou por biometria, leitoras de cartão, sensores, controladoras, computadores com programa de gerenciamento do sistema.</li> </ul>

<b>Sistema Hidráulico</b>	- Garantir o abastecimento de água potável e a eliminação de águas pluviais e esgoto.	<ul style="list-style-type: none"> <li>- Estação de tratamento de água.</li> <li>- Estação de tratamento de esgoto.</li> <li>- Poços artesianos.</li> <li>- Monitorar consumos</li> <li>- Controladores de bombas.</li> <li>- Controladores da qualidade da água.</li> <li>- Identifica vazamentos.</li> <li>- Aciona as bombas dos reservatórios para o caso de baixa nos níveis.</li> <li>- Identifica entupimentos.</li> <li>- Manutenção preventiva.</li> <li>- Funcionamento: regulagem manual, automática local, automática geral.</li> </ul>	-Sensores, controladores, bombas, computadores com programa de gerenciamento do sistema.
---------------------------	---	---	--

## **B – Descrição dos sistemas existentes no TJDF**

São apresentadas a descrição da funcionalidade, da automação, dos equipamentos e dos problemas nos subsistemas de ar condicionado e iluminação do TJDF.

### **Ar Condicionado**

- **Funcionalidade**

Controla o fluxo de ar condicionado nos ambiente, de acordo com informações do usuário, e o funcionamento dos equipamentos segundo a necessidade.

- **Automação**

O atual sistema de ar condicionado apresenta possibilidade de aquecimento e resfriamento dos ambientes. Água gelada e quente são produzidas centralmente e distribuídas através do edifício para onde resfriam ou aquecem o ar em unidades de condicionamento. O controle da temperatura nos ambientes é realizado através da modificação da posição de válvulas que variam a vazão de água quente ou fria que passa pelas serpentinas. O ar a ser condicionado é renovado em caixas de mistura de acordo com a necessidade da zona [33].

- **Equipamentos**

Sensores (temperatura, vazão, pressão), atuadores de válvulas, controladoras e computadores com programa de gerenciamento dos equipamentos.

- **Problemas**

- Falta de comunicação entre o sistema de supervisão implantado e o *chiller*, devido um problema um problema físico em uma peça do *chiller* chamada York talk, do fabricante York. Esse problema acabada afetando também a configuração dos sensores de pressão (linhas de água gelada e condensada);
- Nas válvulas, como de água gelada, do *fancoil* e do ar, que vão para o ambiente. A intervenção sob cada uma é realizada manualmente;
- Não existe um controle sob a temperatura da água na torre de resfriamento pois o sensor lá existente não funciona corretamente. E necessário intervenção manual.



## **Iluminação**

- **Funcionalidade**

A luz é um dos principais itens e tem importância fundamental em um projeto, já que ela é capaz de influenciar o comportamento das pessoas e tem por finalidade garantir condições ideais de visibilidade para seus usuários e complementar a decoração de ambientes. Trata-se não somente de necessidade, mas também de conforto [73].

- **Automação**

O sistema de iluminação existente no prédio do TJDF é realizado por programação horária.

- **Equipamentos**

Lâmpadas, sensores, *dimmer*, medidores, controladoras, computador com programa de gerenciamento dos equipamentos.

- **Problemas**

- É realizada a programação horária.

## **C – Regras de Produção do SE**

### Regra 1

SE compressor não liga = Sim

E interruptor de fluxo de água gelada acionado = Sim

ENTÃO diagnostico = baixo fluxo de água gelada na URL CNF 100%

### Regra 2

SE compressor pára = Sim

E válvula de óleo desajustada = Sim

E filtro entupido = Sim

E baixo nível de óleo = Sim

E defeito na bomba de óleo = Sim

ENTÃO diagnostico = baixa pressão de óleo CNF 100%

### Regra 3

SE compressor pára = Sim

E ar ou gases não-condensáveis no sistema = Sim

E válvula de descarga total ou parcialmente fechada = Sim

E falta de água na torre de resfriamento = Sim

E condensador defeituoso = Sim

ENTÃO diagnostico = alta pressão de descarga CNF 100%

### Regra 4

SE compressor pára = Sim

E parcial ou completo congelamento do evaporador válvula = Sim

E solenóide defeituosa = Sim

E defeito mecânico na válvula de expansão = Sim

E filtro entupido = Sim

E ajuste errado no controle do evaporador = Sim

E má troca de calor no condensador = Sim

E sujeira no evaporador = Sim

E evaporador pequeno = Sim

E vazamento de refrigerante = Sim

ENTÃO diagnostico = baixa pressão de sucção CNF 100%

#### Regra 5

SE compressor pára = Sim

E sujeira no radiador de óleo = Sim

E óleo no refrigerante = Sim

E superaquecimento excessivo = Sim

E alta temperatura de descarga = Sim

ENTÃO diagnostico = óleo aquecido excessivamente CNF 100%

#### Regra 6

SE compressor operando em ciclos curtos = Sim

E sistema liga-desliga está cronometrado errado = Sim

E uso de refrigerante inadequado = Sim

E vazamento de refrigerante = Sim

E filtro secador congelado = Sim

ENTÃO diagnostico = baixa pressão de sucção CNF 100%

#### Regra 7

SE válvula de expansão bloqueada = Sim

E tamanho incorreto da válvula de expansão = Sim

E baixa carga de refrigerante = Sim

ENTÃO diagnostico = baixa capacidade no sistema CNF 100%

#### Regra 8

SE compressor operando ininterruptamente = Sim

E vazamento de refrigerante = Sim

ENTÃO diagnostico = baixa pressão de sucção CNF 100%

#### Regra 9

SE compressor operando ininterruptamente = Sim

E tamanho incorreto da válvula de expansão = Sim

E baixa carga de refrigerante = Sim

ENTÃO diagnostico = sistema com pequena capacidade de refrigeração CNF 100%

#### Regra 10

SE compressor operando ininterruptamente = Sim

E ar no sistema = Sim

ENTÃO diagnostico = alta pressão de descarga CNF 100%

#### Regra 11

SE compressor - falta de óleo = Sim

E carga de óleo insuficiente de óleo = Sim

ENTÃO diagnostico = baixo nível de óleo no compressor CNF 100%

#### Regra 12

SE compressor - falta de óleo = Sim

E vazamento no cárter = Sim

ENTÃO diagnostico = vazamento de óleo CNF 100%

#### Regra 13

SE compressor - falta de óleo = Sim

E vazamento de óleo = Sim

ENTÃO diagnostico = queda na pressão de óleo CNF 100%

#### Regra 14

SE compressor - barulho na bomba de oleo = Sim

E refrigerante líquido no óleo = Sim

ENTÃO diagnostico = baixa temperatura do óleo e/ou alta pressão do óleo CNF 100%

#### Regra 15

SE condensador a agua - quente = Sim

E vazamento na bomba d'água = Sim

E vazamento de refrigerante = Sim

E sujeira nos dutos = Sim

ENTÃO diagnostico = queda da temperatura de condensação CNF 100%

#### Regra 16

SE condensador a água - quente = Sim  
E válvula de expansão parcialmente fechada = Sim  
E lâmpada já substituída = Sim  
ENTÃO diagnostico = linha de líquido quente CNF 100%

#### Regra 17

SE condensador a água - que deixa o condensador muito fria = Sim  
E vazamento de refrigerante = Sim  
E alto fluxo de água = Sim  
ENTÃO diagnostico = baixa pressão de descarga CNF 100%

#### Regra 18

SE condensador a ar - quente = Sim  
E baixo escoamento de ar = Sim  
E ventiladores girando em sentido contrário = Sim  
E sujeiras no condensador = Sim  
ENTÃO diagnostico = queda da temperatura de condensação CNF 100%

#### Regra 19

SE condensador a ar - quente = Sim  
E vazamento de refrigerante = Sim  
E válvula de expansão parcialmente fechada = Sim  
ENTÃO diagnostico = linha de líquido quente CNF 100%

#### Regra 20

SE condensador a ar - quente = Sim  
E excesso de refrigerante no sistema = Sim  
E ar ou gases não-condensáveis no sistema = Sim  
ENTÃO diagnostico = alta pressão de descarga CNF 100%

#### Regra 21

SE evaporador - alta temperatura da água = Sim  
E aumento na temperatura de evaporação resultando em um aumento no grau de superaquecimento = Sim

ENTÃO diagnostico = alta pressão de descarga CNF 100%

#### Regra 22

SE evaporador - alta temperatura da água = Sim

E bloqueamento ou defeito na válvula de expansão = Sim

ENTÃO diagnostico = baixa pressão de sucção CNF 100%

#### Regra 23

SE evaporador - alta pressão de condensação no evaporador = Sim

E alto grau de superaquecimento = Sim

ENTÃO diagnostico = baixa pressão de sucção CNF 100%

#### Regra 24

SE evaporador sujo = Sim

E sujeira de óleo ou depósito de impurezas na superfície do evaporador = Sim

ENTÃO diagnostico = baixa pressão de sucção CNF 100%

#### Regra 25

SE lâmpada intermitente = Sim

ENTÃO diagnostico = trocar lâmpada CNF 100%

#### Regra 26

SE lâmpada intermitente = Sim

E lâmpada já substituída = Sim

ENTÃO diagnostico = trocar reator CNF 100%

#### Regra 27

SE Lâmpada queimou = Sim

ENTÃO diagnostico = trocar lâmpada CNF 100%

## D – Parte do Código do Programa desenvolvido para o Simulador do SIS

### Código da Classe Arquivo.java (Armazena as Regras de Produção)

```
package psis;
import java.io.*;
import java.util.StringTokenizer;

public class Arquivo {

    public Arquivo() {
    }
    public String diagnostico(String expressao) throws Exception
    {

        try{
            int offset=1,offset2=1,offset3=1,total=0;
            String thisLine;
            StringBuffer todoTexto=new StringBuffer("");
            StringBuffer palavra=new StringBuffer("");
            StringBuffer palavra2=new StringBuffer("");
            StringBuffer palavra3=new StringBuffer("");
            StringBuffer palavra4=new StringBuffer("");
            FileInputStream fis = new FileInputStream("C:\\\\leticia\\diagnost.txt");
                DataInputStream myInput = new DataInputStream(fis);
            while ((thisLine = myInput.readLine()) != null) {
                todoTexto.append(thisLine);
            }
            offset = todoTexto.indexOf(expressao);
            System.out.println(todoTexto);
            if(offset!=-1)
            {
                total=total+offset;
            }

            int contador=0,cont=total-1,cont2=0,cont3=0,cont4=0,tam4=0,tamexp=0;//as
ultimas duas variaveis são para ver o tamanho das expresos palavra 4 expressao
            contador=todoTexto.length();

            while(cont<contador)
            {
                palavra.append(todoTexto.charAt(cont));
                cont++;
            }

            offset2=palavra.indexOf("Regra");

            while(cont2<offset2)
            {
                palavra2.append(palavra.charAt(cont2));
```

```

        cont2++;
    }
    offset3=palavra2.indexOf("ENTÃO");
    while(cont4<offset3)
    {
        palavra4.append(palavra2.charAt(cont4));
        cont4++;
    }
    offset3=offset3+5;
    while(offset3<offset2)
    {
        palavra3.append(palavra2.charAt(offset3));
        offset3++;
    }
    tam4=palavra4.toString().length();
    tamexp=expressao.length();
    if(tam4==tamexp+1 || tam4==tamexp)
    {
        return palavra3.toString();
    }
    else{
        return("Diagnostico não conhecido pelo sis.");
    }
}
catch(Exception e)
{return " Problema encontrados na leitura do arquivo"; }
}
}

```

### **Código da Classe Causas.java**

```

package psis;
import java.sql.*;

public class Causas extends conexao{

    public Causas() {
    }
    public void inserir(String codigoSintoma,String causa)throws SQLException
    {
        abrirConexao();
        Statement stm=conn.createStatement();
        String sql="insert into causa(cau_codsintoma,cau_nome)
values("+codigoSintoma+", "+causa+"));";
        stm.executeUpdate(sql);
        fecharConexao();
    }
    public ResultSet selecaoCausas()throws SQLException

```



```

    {
        abrirConexao();
        ResultSet rs;
        Statement stm=conn.createStatement();
        String sql="select * from causa";
        rs=stm.executeQuery(sql);
        return rs;
    }
    public ResultSet selecaoCausasEspecifico(String codigo)throws SQLException
    {
        abrirConexao();
        ResultSet rs;
        Statement stm=conn.createStatement();
        String sql="select * from causa where cau_codsintoma="+codigo+"";
        rs=stm.executeQuery(sql);
        return rs;
    }
    public String selecaoCausasEspecificoNome(String codigo)throws SQLException
    {
        abrirConexao();
        ResultSet rs;
        Statement stm=conn.createStatement();
        String sql="select * from causa where cau_codigo="+codigo+"";
        rs=stm.executeQuery(sql);
        String nome="";
        while(rs.next())
        {
            nome=rs.getString("cau_nome");
        }
        rs.close();
        return(nome);
    }
}

```

### **Código da Classe Sintomas.java**

```

package psis;
import java.sql.*;

public class Sintomas extends conexao{
    public Sintomas() {
    }

    public void inserir(String sintoma)throws SQLException
    {
        abrirConexao();
        Statement stm=conn.createStatement();
        String sql="insert into sintoma(sint_nome) values('"+sintoma+"')";
        stm.executeUpdate(sql);
    }
}

```

```

        fecharConexao();
    }
    public ResultSet selecaoSintomas()throws SQLException
    {
        abrirConexao();
        ResultSet rs;
        Statement stm=conn.createStatement();
        String sql="select * from sintoma";
        rs=stm.executeQuery(sql);
        return rs;
    }
    public ResultSet selecaoSintomaEspecifico(String codigo)throws SQLException
    {
        abrirConexao();
        ResultSet rs;
        Statement stm=conn.createStatement();
        String sql="select * from sintoma where sint_codigo="+codigo+"";
        rs=stm.executeQuery(sql);
        return rs;
    }
    public String selecaoSintomaEspecificoNome(String codigo)throws SQLException
    {
        abrirConexao();
        ResultSet rs;
        Statement stm=conn.createStatement();
        String sql="select * from sintoma where sint_codigo="+codigo+"";
        rs=stm.executeQuery(sql);
        String nome="";
        while(rs.next())
        {
            nome=rs.getString(2);
        }
        rs.close();
        return nome;
    }
    public void alteracaoSintoma(String codigo,String nome)throws SQLException
    {
        abrirConexao();

        Statement stm=conn.createStatement();
        String sql="update sintoma set sint_nome="+nome+" where
sint_codigo="+codigo+"";
        stm.executeUpdate(sql);
        fecharConexao();
    }
    public void excluirSintoma(String codigo)throws SQLException
    {
        abrirConexao();

        Statement stm=conn.createStatement();

```

```
String sql="delete from sintoma where sint_codigo="+codigo+"";
stm.executeUpdate(sql);
fecharConexao();
}

}
```

This document was created with Win2PDF available at <http://www.daneprairie.com>.  
The unregistered version of Win2PDF is for evaluation or non-commercial use only.