

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO
(Bacharelado)

**APLICATIVO PARA GERENCIAMENTO DE FORÇA DE
TRABALHO EM EMPRESAS DE TELEFONIA MÓVEL**

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À UNIVERSIDADE
REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS CRÉDITOS NA
DISCIPLINA COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA
COMPUTAÇÃO — BACHARELADO

TATIANA BOGO FUJII

BLUMENAU, NOVEMBRO/2001

2001/2-49.

APLICATIVO PARA GERENCIAMENTO DE FORÇA DE TRABALHO EM
EMPRESAS DE TELEFONIA MÓVEL

TATIANA BOGO FUJII

ESTE TRABALHO DE CONCLUSÃO DE CURSO FOI JULGADO ADEQUADO
PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE TRABALHO DE
CONCLUSÃO DE CURSO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

Prof. Wilson Pedro Carli — Orientador na FURB

Prof. José Roque Voltolini da Silva — Coordenador do TCC

BANCA EXAMINADORA

Prof. Wilson Pedro Carli

Prof. Roberto Heinzle

Prof. Marilda Maria de Souza

SUMÁRIO

DEDICATÓRIA.....	VI
AGRADECIMENTOS	VII
LISTA DE FIGURAS	VIII
LISTA DE QUADROS	IX
LISTA DE ABREVIATURAS.....	X
RESUMO	XII
ABSTRACT	XIII
1 INTRODUÇÃO	1
1.1 OBJETIVOS.....	3
1.2 ESTRUTURA DO TRABALHO.....	3
2 TELEFONIA CELULAR.....	5
2.1 HISTÓRICO.....	5
2.2 DEFINIÇÃO	6
2.3 TECNOLOGIAS	6
2.3.1 TDMA.....	6
2.3.2 CDMA.....	7
2.4 ELEMENTOS DO SISTEMA CELULAR.....	7
2.4.1 ESTAÇÃO MÓVEL (EM)	8
2.4.2 ESTAÇÃO RÁDIO-BASE (ERB).....	9
2.4.3 CENTRAL DE COMUTAÇÃO E CONTROLE (CCC).....	9
2.4.4 CONEXÕES	9
3 INTERNET.....	10
3.1 WORLD WIDE WEB (WWW).....	11

3.2 ASP – ACTIVE SERVER PAGES	11
3.2.1 RECURSOS DO ASP	12
3.2.2 VANTAGENS E DESVANTAGENS DO ASP.....	12
4 INTELIGÊNCIA ARTIFICIAL (IA)	14
4.1 RACIOCÍNIO BASEADO EM CASOS (RBC).....	15
4.1.1 USO DE CASOS	16
4.1.2 CASOS OU REGRAS?	17
4.1.3 HISTÓRICO	17
4.1.4 DEFINIÇÃO	18
4.1.5 TIPOS DE PROBLEMAS RESOLVIDOS ATRAVÉS DE RBC.....	19
4.1.5.1 Validação de um sistema rbc	20
4.2 ETAPAS DE DESENVOLVIMENTO DE UM SISTEMA DE RBC.....	20
4.2.1 REPRESENTAÇÃO DOS CASOS	22
4.2.2 MEMÓRIA OU BASE DE DADOS	22
4.2.3 INDEXAÇÃO	22
4.2.4 RECUPERAÇÃO	23
4.2.4.1 Similaridade.....	23
4.2.4.1.1 Métrica da Similaridade	24
4.2.4.2 Matching e Ranking.....	24
4.2.4.3 O vizinho mais próximo (nearest neighbour retrieval).....	24
4.2.4.4 Recuperação indutiva (<i>inductive retrieval</i>).....	25
4.2.4.5 Flat Memory, Serial Search	26
4.2.4.6 Shared Feature Networks (SFN).....	26
4.2.4.7 Discrimination Networks (DN)	27
4.2.4.8 Redundant Discrimination Networks (RDN)	27

4.2.4.9 Flat Library, Parallel Search	28
4.2.4.10 Hierarchical Memory, Parallel Search.....	29
4.2.4.11 Best Match.	29
4.2.5 REVISÃO E ADAPTAÇÃO	30
4.2.6 REUTILIZAÇÃO, AVALIAÇÃO E APRENDIZAGEM.....	30
4.3 APLICAÇÕES	31
4.4 METODOLOGIAS	32
4.5 RESUMO DAS VANTAGENS DO RBC	33
5 PROJETO DO APLICATIVO.....	35
5.1 ESPECIFICAÇÃO	36
5.2 APRESENTAÇÃO DO SISTEMA.....	41
5.2.1 TELA PRINCIPAL.....	41
6 CONCLUSÃO	56
6.1 DIFICULDADES ENCONTRADAS	56
6.2 EXTENSÕES	56
7 REFERÊNCIAS BIBLIOGRÁFICAS	58

DEDICATÓRIA

Dedico este trabalho aos meus pais Benvindo e Doraci pelo apoio dado neste trabalho, em especial ao meu esposo Renato que diretamente colaborou para que fosse possível a realização deste curso e ao meu orientador Wilson Pedro Carli que me auxiliou com muita dedicação, sem o qual este trabalho não teria se concretizado.

AGRADECIMENTOS

À Deus, por todas as oportunidades que eu tive durante a minha vida.

Ao meu orientador, Wilson Pedro Carli, pela sua paciência, dedicação e sabedoria.

À professora Marilda Maria de Souza, pelo seu esforço e dedicação.

Aos meus amigos, que me ajudaram no desenvolvimento deste trabalho e que, sempre que possível, tentaram fazer com que eu não pensasse apenas no TCC.

Ao Coordenador do trabalho de conclusão de curso, Prof José Roque Voltolini da silva por toda atenção e apoio dispensados.

A todos aqueles que, diretamente ou indiretamente, contribuíram para a realização deste trabalho.

LISTA DE FIGURAS

Figura 1 - Elementos do Sistema Celular.	8
Figura 2- Ciclo de RBC.	21
Figura 3 – diagrama de contexto	36
Figura 4- diagrama de fluxo de dados – nível zero.	38
Figura 5- Modelo de Entidade e Relacionamento (MER).....	40
Figura 6– tela do menu principal.....	41
Figura 7– tela cadastrar cargo.....	42
Figura 8– tela de confirmação do cadastro de cargos.....	42
Figura 9– tela cadastrar localidade	43
Figura 10– tela de confirmação do cadastro de localidade.....	43
Figura 11– tela cadastrar CCC.....	44
Figura 12– tela de confirmação do cadastro de CCC	44
Figura 13– tela cadastrar ERB.....	45
Figura 14– tela de confirmação do cadastro de ERB	45
Figura 15– tela cadastrar Problema	46
Figura 16– tela de confirmação do cadastro de Problemas	46
Figura 17– tela cadastra Ordem de Serviço.....	47
Figura 18– comparação de similaridade.....	49
Figura 19– tela de confirmação da Ordem de serviço.....	50

LISTA DE QUADROS

Quadro 1 - algoritmo do RBC	50
-----------------------------------	----

LISTA DE ABREVIATURAS

AMPS	<i>Advanced Mobile Phone System</i>
ASP	<i>Active Server Pages</i>
CCC	Central de Comutação e Controle
CDMA	Acesso Múltiplo por Divisão de Código
CGI	<i>Common Gateway Interface</i>
DN	<i>Discrimination Networks</i>
D-AMPS	<i>Digital-American Mobile Phone Service</i>
EM	Estação Móvel
ERB	Estação Rádio-Base
FAPESP	Fundação de Amparo a Pesquisa do Estado de São Paulo
FCC	Comissão Federal de Comunicações
GSM	<i>Global System for Mobile Communication</i>
IA	Inteligência Artificial
IIS	<i>Internet Information Server</i>
ODBC	<i>Open Database Connectivity</i>
PCM	<i>Pulso Code Modulation</i>
PDC	<i>Personal Digital Cellular</i>
RBC	Raciocínio Baseado em Casos
RDN	<i>Redundant Discrimination Networks</i>
RX	Recepção
SFN	<i>Shared Feature Networks</i>
TDMA	Tempo com Acesso Múltiplo
TX	Transmissão

WWW

World Wide Web

RESUMO

Este trabalho visa demonstrar através de um aplicativo para Gerenciamento de Força de Trabalho em Empresas de Telefonia Móvel, o intercâmbio das informações de uma base de dados do aplicativo e a internet através de *Active Server Pages* (ASP), um ambiente de *script* de servidores e ainda a utilização da técnica de inteligência artificial chamada de Raciocínio Baseado em Casos (RBC), a qual vai auxiliar na escolha do técnico mais apropriado para resolver determinado problema.

ABSTRACT

This work seeks to demonstrate by a specific software of working power management in mobile telephone enterprises. The information exchange from a applicative data base and the internet using Active Server Pages (ASP), a language of script of servers and still using the artificial intelligence technique called Case Based Reasoning (CBR) that will help to choose the most appropriate professional to resolve some specific problem.

1 INTRODUÇÃO

Com a evolução da comunicação via Internet as empresas de telefonia tiveram a necessidade de investir em tecnologia e sistemas que facilitassem as tomadas de decisão para suprir as necessidades de maximização de atendimento com qualidade, através de sistemas rápidos e seguros com acesso pela Internet.

Sabe-se que a Internet evolui dia a dia em um ritmo cada vez mais acelerado, e o volume de acessos e de negócios cresce exponencialmente. A Internet foi crescendo e tornou-se um campo de possibilidades incontáveis, ampliaram-se as propriedades da linguagem HTML, surgindo o *Javascript* e em seguida o *Jscript*, que junto com o HTML permitem dar mais efeitos a página. A partir de 1997 os servidores com sistema operacional Unix começaram uma revolução na Internet: surgiram os *scripts* que rodam no servidor com uma interface de comunicação comum, a *Common Gateway Interface* (CGI), onde o usuário entrava com dados em um formulário e enviava para um *script* em CGI, que lia, interpretava e processava estes *scripts* (Wille, 1999).

Outra tecnologia desenvolvida para a Internet é *Active Server Pages* (ASP) que é um ambiente para programação de páginas dinâmica e interativa que utiliza recursos do HTML, suporte a linguagem de *scripts* (*Java Script*, *Vbscript*), acesso à base de dados. O acesso à base de dados realizado com uma aplicação ASP pode ser usada com qualquer banco de dados compatível com *Open Database Connectivity* (ODBC). Isto inclui dados do Access, Microsoft SQL Server, Oracle, DB2, entre outros, tornado o acesso às páginas Web mais rápidas e eficientes.

Desta forma as empresas produtoras de *software* nos últimos anos passaram a disponibilizar novas linguagens e tecnologias que propiciaram a renovação de aplicativos nas empresas em geral. Entre estas pode-se citar: CGI, ASP, JAVA, DELPHI, PHP, que em muito beneficiaram o desenvolvimento de aplicativos, principalmente atendendo as necessidades da rede mundial de computadores (Internet).

Para auxiliar o desenvolvimento destes aplicativos fazendo com que os mesmos sejam mais eficientes na relação cliente/empresa, técnicas de Inteligência Artificial são utilizadas muitas vezes para suprir determinadas necessidades na resolução de problemas. De acordo

com Rabuske (1995) Inteligência Artificial é o resultado da aplicação de técnicas e recursos, especialmente de natureza não numérica, viabilizando a solução de problemas que exigiriam do humano certo grau de raciocínio e de perícia. A solução destes problemas com recursos tipicamente numéricos são muito difíceis. Por isso é que a Inteligência Artificial caracteriza uma nova era da computação, a era do processamento não numérico, ou ainda Inteligência Artificial consiste na especificação do conhecimento necessário a uma capacidade cognitiva, em que capacidade cognitiva inclui atividades tais como solução de problemas, uso de linguagem, tomada de decisões, perdiz respeito ao planejamento de trajetórias (Bittencourt, 1998).

Uma das técnicas de Inteligência Artificial é o RBC (Raciocínio Baseado em Casos), que se baseia na premissa de que seres humanos utilizam um raciocínio analógico ou experimental para aprender e resolver problemas complexos. Os Sistemas RBC utilizam um processo interativo constituído genericamente por: identificação da situação atual, busca da experiência mais semelhante na memória e aplicação do conhecimento desta experiência na situação atual (Costa, 1999).

Em um estágio desenvolvido no curso de Ciências da Computação na Universidade do Sul de Santa Catarina (UNISUL), junto a uma empresa de telefonia móvel, sentiu-se a necessidade e informatizar o setor de controle de ordem de serviço, que era feito manualmente por todo setor de operação e manutenção da empresa. O setor onde o controle de ordem de serviço é realizado é chamado de Operação e Manutenção, onde existem vários funcionários controlando os softwares das estações celulares. A comunicação entre o setor e os técnicos externos é feita manualmente, sendo que o funcionário entra em contato via telefone com o provável técnico fazendo com que esta comunicação seja lenta e muitas vezes não aconteça, deixando o cliente aborrecido, pois seu problema geralmente não é solucionado.

A automatização dos controles de ordem de serviço terá como propósito tornar mais rápida e eficiente a tomada de decisão de qual será o melhor técnico para a resolução do problema e controlando o acesso de atendentes, técnicos e gerentes no acesso ao aplicativo, que a qualquer momento poderão acompanhar o trâmite de um chamado, desde sua entrada, reencaminhamentos e conclusão. Poder-se-á, ainda, emitir estatística de anormalidades mais ocasionais, tipos de equipamentos que mais sofrem avarias e técnicos mais atuantes.

Diante desse contexto, o aplicativo desenvolvido, a partir de um chamado técnico registrado na base operacional através da Internet, utilizando o ambiente ASP a técnica de Inteligência Artificial (RBC), selecionará o técnico de maior competência para resolver o problema.

1.1 OBJETIVOS

O objetivo principal do trabalho é o desenvolvimento do aplicativo visando à informatização dos chamados técnicos com recursos voltados à Internet através de Páginas de Servidor Ativas (ASP) e utilizando técnicas de Inteligência Artificial como o RBC o aplicativo encaminhe a ordem de serviço à solução, indicando o técnico de maior competência para resolver o problema, além da geração de relatórios estatísticos.

Os objetivos específicos do aplicativo são:

- a) registrar todas as ordens de serviço e seus trâmites na base de dados operacional através da Internet;
- b) aplicar técnicas de Inteligência Artificial (RBC) para auxiliar na decisão de qual o técnico mais indicado para solucionar o problema;
- c) alimentação do sistema por boletins de anormalidades existentes nas Estações Rádio Base (ERBs), Plano de Rádio Frequência (Sinal), Entroncamentos (Rotas), emitidos pelos operadores do sistema.

1.2 ESTRUTURA DO TRABALHO

O trabalho encontra-se estruturado da seguinte maneira:

O capítulo 1 apresenta uma introdução sobre o trabalho com seus objetivos.

O capítulo 2 tem como finalidade dar uma visão geral da telefonia móvel com definições de algumas nomenclaturas que serão utilizadas no decorrer do trabalho.

O capítulo 3 apresenta um resumo sobre a internet englobando seu surgimento, histórica e formas de acesso às páginas descrevendo com maiores detalhes o ASP, seu funcionamento, arquitetura e requisitos de sistema necessários para sua execução.

O capítulo 4 apresenta informações sobre inteligência artificial, sua origem, seus campos de aplicação e uma área em específico a qual será utilizada neste trabalho que é o RBC, falando de seu histórico, definição, etapas de desenvolvimento de um sistema.

O capítulo 5 está voltado ao desenvolvimento do aplicativo, onde será apresentado a modelagem de dados, as telas de entrada de dados bem como a implementação do RBC.

O capítulo 6 finaliza o trabalho com uma conclusão e sugestões para trabalhos futuros.

2 TELEFONIA CELULAR

A Telefonia Celular é uma modalidade de comunicação telefônica que dispensa fios, destinada a permitir que seus usuários possam se comunicar em qualquer lugar onde haja cobertura, através de ligações locais, nacionais ou internacionais.

Uma das maiores razões para o desenvolvimento do sistema móvel celular e sua aplicação em larga escala foi à limitação operacional verificada nos sistemas de telefonia fixa convencionada basicamente no que se refere às suas capacidades limitadas de atendimento, desempenho técnico fraco e utilização ineficiente do espectro de frequências.

2.1 HISTÓRICO

A primeira aplicação de comunicações móveis surgiu há mais de um século simultaneamente à utilização do rádio quando o italiano Guglielmo Marconi em 1880 experimentou várias transmissões de rádio para seu iate que se encontrava a 18 milhas da costa.

Em 1921 o Departamento de Polícia de Detroit fez o primeiro uso regular de um sistema de comunicação móvel enviando ordens via rádio para as viaturas e vice-versa, operando na faixa de 2MHz, apesar da precariedade dos equipamentos da época, fez um sucesso muito grande no combate à criminalidade, e logo em 1932 o Departamento de Polícia Civil de Nova York também adotava algo semelhante.

Em 1938 o FCC (Comissão Federal de Comunicações) regulamentou a utilização de quatro canais entre 30 e 40 MHz, fazendo com que se houvesse um grande impulso para o desenvolvimento de um sistema de grande porte nos Estados Unidos. Foram feitas experiências em várias cidades americanas utilizando outras faixas de frequências e número de canais, mas somente em 1955 com a implementação da seleção automática de canais vagos pelos equipamentos de rádios móveis, interligados a uma central de comutação automática através de um computador é que se desenvolveu a base dos sistemas móveis atuais.

No Brasil a telefonia celular iniciou-se no Rio de Janeiro em 1991 e expandiu-se rapidamente para todo país.

2.2 DEFINIÇÃO

Conforme Anatel (2001) serviço móvel celular é o serviço de telecomunicações móvel terrestre, aberto à correspondência pública, que utiliza sistema de radiocomunicações com técnica celular, interconectado à rede pública de telecomunicações, e acessado por meio de terminais portáteis, transportáveis ou veiculares, de uso individual.

2.3 TECNOLOGIAS

2.3.1 TDMA

Conforme Celular (2001) divisão de Tempo com Acesso Múltiplo, é uma das tecnologias digitais mais usadas pelo mercado norte-americano, que transforma sinais analógicos de voz em dados digitais e aumenta em três vezes a capacidade de atendimento a usuários em relação ao da tecnologia analógica.

A tecnologia TDMA é usada em comunicação de telefones celulares digitais para dividir cada canal celular em três slots para aumentar a quantidade de dados transmitidos. TDMA é usado pelo sistema D-AMPS (*Digital-American Mobile Phone Service*), pelo GSM (*Global System for Mobile Communication*), e pelo PDC (*Personal Digital Cellular*).

TDMA foi primeiramente especificado como um padrão no EIA/TIA *Interim Standard 54* (IS-54 ou UDSC) que não é completamente digital, em 1988. IS-136 é uma nova versão dos IS-54 completamente digital, e é o padrão dos EUA para TDMA. A conexão pode ser associada a uma estação móvel ou uma base fixa.

2.3.2 CDMA

Conforme Celular (2001) acesso Múltiplo por Divisão de Código (CDMA) é o nome da tecnologia usada para comunicação sem fio em que o transporte das informações ocorre por meio de ondas de rádio. O CDMA foi desenvolvido inicialmente para uso militar. O sistema gasta pouca energia; usa as frequências disponíveis de forma eficiente; simplifica o planejamento, pois todos os equipamentos transmitem e recebem na mesma frequência; usa exclusivo sistema de códigos que permite receber o sinal desejado mesmo em condições adversas.

O CDMA difere muito dos demais sistemas. No modo analógico AMPS (*Advanced Mobile Phone System*), cada telefone usa uma par de frequências de rádio (canal) quando se que falar, sendo um para transmissão (Tx) e outro para recepção (Rx). Cada assinante utiliza um só canal. No sistema CDMA, todos os assinantes transmitem e recebem informações usando o mesmo canal, ao mesmo tempo. A cada assinante, e atribuído um código exclusivo. Para receber informações de um assinante específico, só conhecendo o seu código.

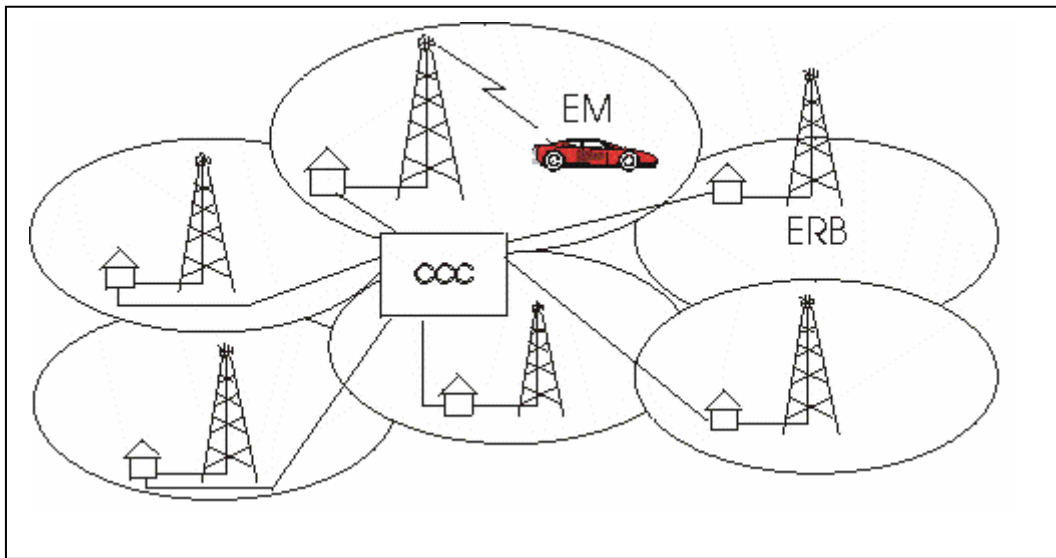
O sistema CDMA é digital, isto é, inicialmente a voz do assinante é convertida em bits. O processo é feito através do método de conversão PCM (*Pulso Code Modulation*), a digitalização de 1 segundo de conversa resulta em 64.000 bits.

2.4 ELEMENTOS DO SISTEMA CELULAR

Conforme Tim (1999) um sistema celular é constituído de três elementos, além das conexões entre estes elementos, como mostra a figura 1.

- a) estação móvel (EM);
- b) estação rádio-base (ERB);
- c) central de comutação e controle (CCC).

Figura 1 - Elementos do Sistema Celular.



Fonte: (Tim, 1999).

2.4.1 ESTAÇÃO MÓVEL (EM)

De acordo com Tim (1999) uma estação móvel possui uma unidade de controle, um transceptor e uma antena. Transmite e recebe sinais de voz, possibilitando a conversação. Transmite e recebe sinais de controle, permitindo o estabelecimento de chamada. São produzidas por um grande número de fabricantes, o que faz com que o projeto e as facilidades de assinantes implementadas variem bastante. Podem ser utilizadas em diferentes aplicações como:

- a) instaladas em carro (telefone celular veicular);
- b) transportáveis (telefone celular transportável);
- c) portáteis (telefone celular portátil).

A potência máxima de saída da unidade montada em carro é de três watts, enquanto que da unidade portátil é de 0,6W. Quando a EM acessa a Central de Comutação e Controle (CCC), envia a sua classe de estação, onde está indicada a sua potência máxima de saída.

2.4.2 ESTAÇÃO RÁDIO-BASE (ERB)

De acordo com Tim (1999) uma estação rádio-base fornece a interface entre a central de comutação e controle e as estações móveis. Contém uma unidade de controle, rádios transceptores, antenas, planta de alimentação e terminais de dados. Transmite e recebe sinais de controle para o estabelecimento e supervisão das chamadas. Transmite e recebe sinais de voz para, e de várias estações móveis dentro de sua área de cobertura.

2.4.3 CENTRAL DE COMUTAÇÃO E CONTROLE (CCC)

Conforme Tim (1999) é a central de comutação do serviço celular, considerada o coração do sistema móvel celular. É o elemento de coordenação central de toda a rede celular, pois comuta e controla um aglomerado celular. Estabelece a interface com a rede de comutação pública. Comuta as chamadas originadas/terminadas de e para a estação celular. Permite que a estação móvel tenha à sua disposição os mesmos serviços e facilidades fornecidas pela rede pública aos assinantes fixos.

2.4.4 CONEXÕES

De acordo com Tim (1999) quando uma chamada entre um assinante móvel e um assinante fixo for estabelecido, a conversação será transmitida via rádio entre a estação móvel e a estação rádio-base situada mais próxima desta estação móvel. Em seguida, esta conversação é encaminhada a uma CCC do serviço móvel, e daí comutada para a rede de telefonia pública onde o assinante fixo está conectado. Da mesma forma, um assinante da rede fixa pode acessar automaticamente qualquer telefone móvel, através de uma busca que é processada pela CCC do serviço móvel.

As conexões de ERB e CCC são feitas normalmente por troncos de linhas físicas, sendo também possíveis conexões por rádio e fibras ópticas. Desta mesma forma, as CCCs são conectadas às centrais da rede fixa.

3 INTERNET

Conforme Uninformare (2001) tão importante quanto uma TV a Cabo, a Internet vem tomando espaço considerável na sociedade. Até mesmo nas revistas que não são da área de informática, é comum a publicação de reportagens que esclarecem e fazem comentários sobre as novidades da Internet e similares. A importância desses recursos está ficando cada vez mais aparente e por esse motivo está sendo passada para todas as áreas: economia, comércio, indústria e outros.

A Internet consiste na ligação de vários computadores no mundo inteiro ligados através de teleprocessamento, formando uma rede de milhões de computadores. Na verdade essa rede é formada por várias outras pequenas redes, por exemplo, *Bitnet*, *Usenet* e outras. O conjunto dessas pequenas redes é que forma a Internet.

A Internet surgiu nos anos 70 através de um projeto do governo americano de defesa que queria criar uma rede que fosse à prova de sabotagens, uma rede que mesmo tendo um de seus “caminhos” quebrados, continuasse a funcionar por um caminho alternativo. Para isso, usaram uma técnica de comunicação denominada “Rede Comutada de Pacotes”. As mensagens são quebradas em pacotes e distribuídas através da teia da rede, misturando-se com outros pacotes, todos com endereço de chegada fixo que navegam até chegarem ao destino, para daí serem reorganizados.

Conforme Pereira (2000) as primeiras conexões do Brasil com a Internet foram feitas em 1.988, através da Fundação de Amparo a Pesquisa do Estado de São Paulo (FAPESP) e do Laboratório Nacional de Computação Científica (Rio de Janeiro). Em 1.989, o Ministério da Ciência e Tecnologia criou a Rede Nacional de Pesquisa, que construiu o primeiro backbone nacional. Em dezembro de 1.994, foram iniciados os testes comerciais com linhas discadas. Em agosto de 1995, deu-se o início comercial através de acessos dedicados por provedores de acesso comercial.

A grande vantagem da Internet é a ligação entre computadores de todo o mundo sem sair de sua casa. Você pode se comunicar com um computador no Japão com a mesma facilidade com que se comunica com o computador do seu vizinho.

3.1 WORLD WIDE WEB (WWW)

Conforme Uninformare (2001) World Wide Web (WWW), ou simplesmente Web, é um poderoso serviço oferecido pela Internet que tem como função à comunicação entre os computadores da rede através de uma interface gráfica.

Utilizando um software específico para este recurso, os chamados Browser, o usuário conecta-se as *Home-Pages* e pode percorrer o mundo inteiro à procura de uma informação simplesmente apertando botões da tela com o seu mouse.

As *Home-Pages* são páginas com textos, imagens, sons e vídeo, como uma revista “viva”. Nessas páginas utiliza-se um processo conhecido como “Hiper Texto”, ou seja, algumas palavras-chave ou figuras são grifadas (Links) e, ao selecioná-la, entra-se em uma nova página com outros links e assim por diante.

Um dos grandes trunfos desse recurso é o comércio, pois com o Web a pessoa pode verificar uma lista de produtos com suas respectivas fotos e, se quiser, até comprá-los via cartão de crédito pela Internet.

3.2 ASP – ACTIVE SERVER PAGES

O ASP (*Active Server Pages* – páginas de Servidor Ativas) é um ambiente para programação por *script* no servidor, que se pode usar para criar páginas dinâmicas, interativas e de alta performance. Com as páginas ASP, os *scripts* são executados no servidor e não no cliente. É próprio servidor que transforma os *scripts* em HTML padrão, fazendo com que qualquer *browser* do mercado seja capaz de acessar um site que usa ASP. Um arquivo que contém código ASP é identificado pela extensão “.asp”.

ASP é uma página em HTML estendida por comandos ASP que também controlam a aparência da página Web e *tags* da HTML, fora dos blocos de *script*, que é executado no fluxo do programa do *script* de servidor. Ela contém, basicamente, código HTML e *scripts*

que irão rodar no servidor, sendo o *Vbscript* a linguagem script mais utilizada. Pode-se usar também *Javascript*, *CGI*, *Pearl*, *C*, dentre outras.

O *Active Server Pages* é um software que acompanha o *Internet Information Server (IIS) for NT Server*, com o *Personal Web Server for Windows NT Workstation* ou o *Windows 98*. Na verdade o ASP surgiu juntamente com o lançamento do *Internet Information Server 3.0 do Windows NT*, uma solução da empresa Microsoft, que exige que o servidor rode um sistema *Windows 98*, *NT* ou superior para processar páginas dinâmicas na internet. Este requisito é solicitado apenas no servidor, o ASP aceita qualquer linguagem o que significa que o programa não é limitado ao *Vbscript*, pode ser utilizada qualquer outra linguagem desde que essa linguagem suporte o modelo *Activex Scripting* e que seja alterado nas configurações do IIS.

3.2.1 RECURSOS DO ASP

Entre os recursos que podem ser implementados via ASP, Weissinger (1999) cita:

- a) programação em VBScript ou Java Script;
- b) acesso à banco de dados;
- c) sessões (persistência de informações no servidor).

3.2.2 VANTAGENS E DESVANTAGENS DO ASP

Na utilização de páginas dinâmicas com ASP foram verificadas algumas vantagens e desvantagens que podem influenciar na escolha do ambiente a ser utilizado para a criação das páginas para a internet. De acordo com Weissinger (1999) pode-se citar como vantagens do ASP o seguinte:

- a) independência do browser: ASP pode rodar páginas complexas no servidor e enviar somente os resultados para o cliente;
- b) páginas x bancos de dados: Permite visualizar, atualizar e adicionar informações nos servidores SQL;
- c) segurança do código fonte: Como o Servidor retorna somente o resultado HTML, o código fonte (lógica) fica preservada;
- d) linguagens: O ASP pode utilizar de comandos em VBScript, JavaScript e HTML.

As desvantagens mais relevantes do ASP de acordo com Raduenz (2000) apud Marinho (2000) são:

- a) a mistura de códigos HTML, Vbscript;
- b) ASP dificulta a compreensão do código fonte;
- c) ASP é vinculado aos sistemas operacionais da Microsoft. Embora existam versões ASP para outros sistemas operacionais, estas versões não são muito conhecidas.

4 INTELIGÊNCIA ARTIFICIAL (IA)

Desde seus primórdios, a IA (Inteligência Artificial) gerou polêmica, a começar pelo seu próprio nome, considerada presunçosa por alguns, até a definição de seus objetivos e metodologias. O desconhecimento dos princípios que fundamentam a inteligência, por um lado, e dos limites práticos da capacidade de processamento dos computadores, por outro, levou periodicamente a promessas exageradas e às correspondentes decepções.

Dada a impossibilidade de uma definição formal precisa para IA, visto que para tanto seria necessário definir, primeiramente, a própria inteligência, foram propostas algumas definições operacionais: “É simplesmente uma maneira de fazer o computador pensar inteligentemente” (Levine, 1988); “É o estudo de como fazer os computadores realizarem tarefas em que, no momento, as pessoas são melhores” (Rich, 1988); “IA é o resultado da aplicação de técnicas e recursos, especialmente de natureza não numérica, viabilizando a solução de problemas que exigiriam do humano certo grau de raciocínio e de perícia” (Rabuske, 1995). Outros se recusam a propor uma definição para o termo e preferem estabelecer os objetivos da IA: “Tornar os computadores mais úteis e compreender os princípios que tornam a inteligência possível” (Winston, 1987).

Os programas desenvolvidos através da IA caracterizam-se pelo fato de objetivarem uma interação com o usuário ou seu ambiente (sistema) por meio da simulação de atitudes e reações humanas que envolvem a compreensão, análise, planejamento, tomada de decisão, aprendizado, etc. Esses programas já são utilizados com significativa eficiência substituindo ou auxiliando o homem em tarefas de aprendizagem, diagnóstico médico, reconhecimento de padrões, previsões econômicas, tradução, reconhecimento de padrões de imagens e voz, automação de processos industriais, controle de qualidade.

Existem vários campos de estudo dentro da IA com o propósito de dotar a máquina de capacidade de raciocínio, aprendizado e auto-aperfeiçoamento, alguns desses campos são descritos abaixo:

- a) Processamento de Linguagem Natural - Processamento de linguagem compreende a compreensão de uma seqüência de símbolos (geralmente sonoros, chamado palavras) e

a geração de outra seqüência, não necessariamente usando os mesmos símbolos (Barreto 1997);

- b) Reconhecimento de Padrões - É uma das áreas de pesquisa bem avançadas da IA. A capacidade de reconhecimento de padrões permite reconhecer o dono de uma impressão digital, validar a assinatura num cheque bancário, ler e digitalizar um texto escrito (Rabuske, 1995);
- c) Jogos - É o estudo voltado para a construção de programas de jogos envolvendo raciocínio. Ex. O jogo de xadrez que foi utilizado para as primeiras experiências em programação do raciocínio artificial (Rabuske, 1995);
- d) Robótica - É o campo de estudo voltado para desenvolver meios de construir máquinas que possam interagir com o meio (ver, ouvir e reagir aos estímulos sensoriais). Robótica e inteligência artificial são assuntos distintos. Em robótica entra toda uma parte de construção mecânica, elétrica, etc, que nada tem a ver com Inteligência Artificial (Barreto, 1997).

Existem muitos outros campos de estudo que são englobados no desenvolvimento da IA. Por ser uma ciência relativamente nova possui um potencial muito grande ainda inexplorado, certamente muitas outras áreas de pesquisa e de aplicações deverão surgir nas próximas décadas.

4.1 RACIOCÍNIO BASEADO EM CASOS (RBC)

Segundo Fernandes (2001) Raciocínio Baseado em Casos (RBC) é uma ferramenta de raciocínio da Inteligência Artificial (IA). A filosofia básica desta técnica é a de buscar a solução para uma situação atual através da comparação com uma experiência passada semelhante. O processo característico de RBC consiste em: identificar a situação atual, buscar a experiência mais semelhante na memória e aplicar o conhecimento desta experiência passada na situação atual.

Este procedimento pode ser detalhado em alguns passos:

- a) identificação de um problema a ser resolvido (problema de entrada);
- b) definição das principais características que identificam este problema;
- c) busca e recuperação na memória de casos com características similares;
- d) seleção de um ou mais dentre os casos recuperados;
- e) revisão deste(s) caso(s) para determinar a necessidade de adaptação;
- f) reutilização do caso adaptado para resolver o problema de entrada;
- g) avaliação da solução do problema de entrada;
- h) inclusão do caso adaptado na memória de casos (aprendizagem).

4.1.1 USO DE CASOS

Segundo Fernandes (2001) Experiências mostram como pessoas usam casos específicos para resolver problemas:

- a) aprendendo novas tarefas no serviço, em casa e em qualquer lugar;
- b) em diagnósticos de problemas mecânicos, diagnósticos médicos;
- c) problemas em linhas telefônicas;
- d) concepção de projetos arquitetônicos entre outros;

Estas experiências vividas pela primeira vez são denominadas como um caso, nas demais vezes em que for executada a mesma tarefa, é intuitivo que se use aquilo que foi aprendido na primeira vez, tornando-se uma rotina. Um exemplo clássico é a do motorista, a primeira vez que ele dirige é um caso, e após ter aprendido a dirigir você não precisa mais

lembrar da última vez que dirigiu um carro para saber como dirigir.

4.1.2 CASOS OU REGRAS?

Uma regra é colocada na memória independente de um exemplo de como ela foi usada, quando a regra não pode ser usada (exceção), um caso é guardado. Estes casos são guardados separadamente até que existam casos suficientemente para se criar uma nova regra.

Citamos a seguir um exemplo de casos ou regras.

Para um motorista dirigir é normal se tornando uma regra, mas se o mesmo bater o veículo, então, isto será um caso. Mas se o mesmo motorista torna a batida do veículo um habito, então, se tornará abstração (regra) por ser generalizado, mas sempre haverá um link entre os dois (Fernandes, 2001).

4.1.3 HISTÓRICO

Raciocínio Baseado em Casos teve origem em Inteligência Artificial nos trabalhos de Roger Schank em memória dinâmica, e o papel central que lembra situações recentes (episódios, casos) e situações modelos (*scripts*, MOP's ou episódios generalizados) têm uma solução e aprendizado do problema. O primeiro sistema que pode ser considerado de RBC foi o CYRUS, desenvolvido por Janet Kolodner. Este sistema foi baseado no modelo de memória dinâmica de Schank e na teoria dos MOP's para aprendizagem e solução de problemas.

Outras bases para RBC e outros conjuntos de modelos foram desenvolvidos por Bruce Porter e seu grupo. Inicialmente eram endereçados os problemas de aprendizado de máquina de conceitos aprendidos para classificando de tarefas. Esta linha de desenvolvimento deu origem ao sistema PROTOS que enfatiza a integrando do domínio geral de conhecimento e especifica o conhecimento de casos dentro de uma estrutura de representação unificada.

Atualmente, trabalhos envolvendo RBC tanto nos EUA quanto na Europa têm se expandindo e o número de publicações tem crescido consideravelmente em atividades

relacionadas a IA (Fernandes, 2001).

4.1.4 DEFINIÇÃO

Segundo Fernandes (2001) Raciocínio Baseado em Casos (RBC) é uma técnica de Inteligência Artificial (IA) que resolve novos problemas através da recuperação e adaptação de soluções anteriores. A inspiração inicial de RBC advém da maneira como o raciocínio humano recorda e aplica experiências passadas para solucionar novos problemas.

Em vez de confiar em conhecimento geral de um domínio do problema somente, ou fazendo associações contendo relações generalizadas entre descrições de problema e conclusões, RBC pode utilizar o conhecimento específico de situações de problemas concretos (casos). Um novo problema é resolvido achando um caso passado que seja semelhante, e usando de novo isto na situação de um problema novo. Uma segunda diferença importante do RBC é a técnica de aprender com incremento, contínuo, desde que uma experiência nova é retida cada vez que um problema foi resolvido e foi feito isto estará imediatamente disponível para resolução de problemas futuros.

Um RBC faz pesquisas em sua memória e verifica se há um caso semelhante com as características do problema, podendo achar um caso ou diversos que serão utilizados na solução do problema. O sistema é capaz de encontrar e modificar partes de casos que não se adequam ao problema, criando assim um novo caso para uso posterior.

Os seres humanos são considerados robustos resolvedores de problemas, muitas vezes resolvem problemas difíceis apesar do conhecimento limitado e incerto. Além disso, com a experiência adquirida a performance deles melhora cada vez mais. Todas estas qualidades são desejáveis em um sistema de IA para o mundo real. Portanto existem boas razões para se utilizar RBC:

- a) aquisição do conhecimento: o conhecimento presente em um sistema de RBC fica armazenado na própria base de casos;
- b) manutenção do conhecimento: um usuário do sistema pode ser habilitado para adicionar novos casos na base de casos sem a intervenção do especialista;

- c) eficiência crescente na resolução de problemas: a reutilização de soluções anteriores ajuda a incrementar a eficiência na resolução de problemas. RBC armazena as soluções que não obtiveram sucesso, assim como aquelas bem sucedidas. Isto faz com que eventuais insucessos sejam evitados;
- d) qualidade crescente das soluções: quando os princípios de um domínio não são bem conhecidos, regras não são a melhor solução. Nesta situação, as soluções sugeridas pelos casos refletem o que realmente aconteceu em uma determinada circunstância;
- e) aceitação do usuário: um dos pontos chaves para o sucesso de um sistema de IA é a aceitação do usuário. Os sistemas de RBC podem comprovar o seu raciocínio apresentando ao usuário os casos armazenados na base. Este é um dos grandes problemas enfrentados por outras técnicas de IA como as Redes Neurais.

4.1.5 TIPOS DE PROBLEMAS RESOLVIDOS ATRAVÉS DE RBC

Sistemas de RBC já estão presentes em diversos tipos de aplicações como suporte ao consumidor, controle de qualidade, manutenção de aeronaves, planejamento de processos, apoio à decisão, roteamento de mensagens, *help desks* computadorizados, previsões, pesquisa jurisprudências, etc. São alguns tipos de problemas resolvidos com o uso de RBC.

Seja qual for a aplicação, os sistemas desenvolvidos em RBC resolvem problemas utilizando o mesmo princípio: compara o novo caso com os casos já armazenados na memória ou base de casos. Faz-se uma pesquisa pelos casos mais semelhantes utilizando algoritmos que calculam o grau de similaridade entre o novo caso e os casos da base. Se a consulta retornar um caso idêntico ao caso novo, dificilmente acontece, o caso encontrado pode ser usado naturalmente para resolver o problema. Caso contrário, será necessário que o usuário ou o próprio sistema faça uma adaptação da solução encontrada para resolver o problema. Quando é apropriado, novos casos serão acrescentados à base de casos para aumentar o conhecimento do sistema (Fernandes, 2001).

4.1.5.1 VALIDAÇÃO DE UM SISTEMA RBC

Segundo Fernandes (2001) a validação de um sistema de RBC pode ser feita em função de vários aspectos. Métodos de validação de outros sistemas inteligentes são adequados, mas ainda há um conjunto de técnicas de validação exclusivas para os sistemas de RBC. Uma questão exclusiva dos sistemas de RBC trata-se de avaliar se a tarefa para qual o sistema é proposto, quando executada por especialista humano, é uma tarefa baseada em casos.

Em termos gerais, estas são algumas das características a serem consideradas para a validação de um sistema de RBC:

- a) características técnicas; tais como estabilidade e operacionalidade; escolha do problema; se o problema é próprio para o tipo de raciocínio;
- b) características organizacionais; se o sistema é adequado à operação dentro de uma organização;
- c) características econômicas, (retorno do investimento, aumento na qualidade de serviços);
- d) características estratégicas, se a memória de casos pode ser vista como um ativo;
- e) qualidade e eficiência com relação às principais etapas de um sistema baseado em casos.

4.2 ETAPAS DE DESENVOLVIMENTO DE UM SISTEMA DE RBC

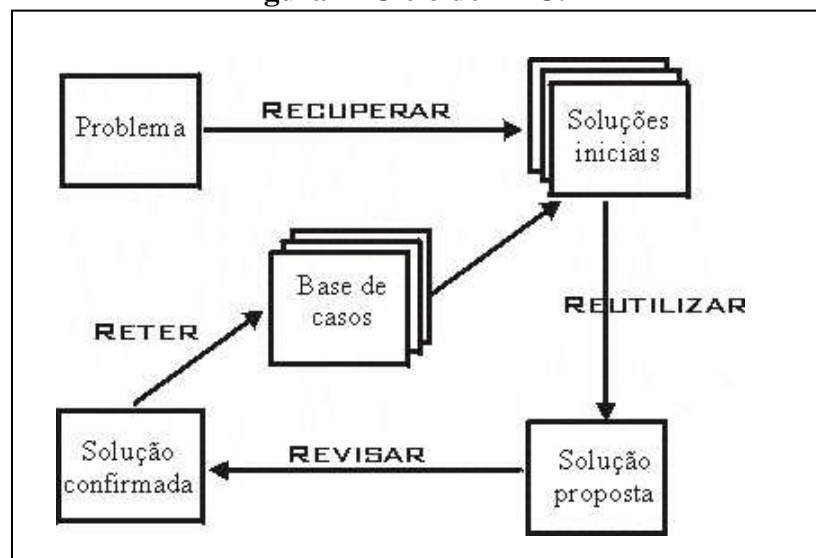
Segundo Fernandes (2001) o desenvolvimento de um sistema de RBC é uma tarefa complexa que não se sujeita a ser formulada através de uma metodologia. Entretanto, há quatro etapas distintas, cada uma com seus próprios desafios, que quando resolvidas, modeladas e implementadas compreendem um sistema de RBC. As quatro etapas são:

- a) representação dos casos;
- b) recuperação dos casos;
- c) revisão e adaptação;
- d) reutilização, avaliação e aprendizagem.

Alguns autores apresentam estas etapas com diferentes enfoques. Algumas etapas são difíceis de ser apresentadas isoladamente, daí a razão do agrupamento de algumas delas. Além disso, etapas como a recuperação é ainda subdividida em outras etapas, conforme será descrito posteriormente.

Neste estágio ainda não se pode garantir a competência desta ferramenta para a solução do problema proposto, pois a validação do sistema é uma etapa à parte. A validação não foi incluída porque não é uma etapa indispensável no desenvolvimento de um sistema de RBC (Fernandes, 2001). A figura 2 mostra o ciclo do RBC.

Figura 2- Ciclo de RBC.



Fonte: adaptação de Fernandes (2001).

4.2.1 REPRESENTAÇÃO DOS CASOS

O conhecimento nos sistemas baseados em casos é representado através de uma memória de casos. O problema da representação do conhecimento trata-se, então, de como descrever os casos e de como organizar os casos na memória. Na representação dos casos há dois componentes básicos: a descrição do problema e a descrição da solução, como um algoritmo, por exemplo.

4.2.2 MEMÓRIA OU BASE DE DADOS

Um conjunto de casos compreende uma base de casos ou memória. O tipo de memória que organiza estes casos deve ser cuidadosamente escolhido. As questões de eficiência e tempo computacional são aqui consideradas. Os modelos de memória são estruturas de organização dos casos.

Um dos primeiros modelos de representação de conhecimento na IA foi o modelo de memória com o uso de redes semânticas. Entretanto, este modelo não é capaz de representar o conhecimento em sua totalidade. O modelo posterior que veio tentar complementar o modelo de redes semânticas é o modelo de memória episódica, de um psicólogo. Os pesquisadores de IA continuavam em busca de um paradigma capaz de representar o conhecimento e os melhores resultados foram conquistados pela equipe de Roger Schank em Yale. Suas pesquisas evoluíram para o desenvolvimento do modelo de Memória Conceptual, dos *scripts* e, posteriormente, dos "MOPS" (*memory organization packets*), pacotes de organização de memória, e do modelo de Memória Dinâmica (Fernandes, 2001).

4.2.3 INDEXAÇÃO

A indexação é uma questão que pode ser bastante importante conforme a estrutura e o conteúdo da memória. A memória deve ser indexada para proporcionar uma recuperação e reutilização eficiente.

Características dos casos tomam-se índices que caracterizam um evento. Por exemplo, uma operação de investimento em CDB por 30dd fica caracterizada como um investimento de curto prazo. A indexação é elaborada quando da definição dos casos enquanto sua utilização prática se dá quando da recuperação. Ao tentarmos comparar um investimento em renda fixa, a busca será por um investimento de curto prazo.

4.2.4 RECUPERAÇÃO

A partir de um problema a ser resolvido (problema de entrada), a etapa de recuperação consistem em fazer uma busca na memória de casos e selecionar quais poderão ser aproveitados. A busca por casos é feita por algoritmos que selecionam casos com determinada similaridade com relação ao problema de entrada.

A etapa de recuperação envolve algumas tarefas, a saber: similaridade, métrica da similaridade, recuperação de casos e seleção do "*best match*". Segundo Aamodt e Plaza (1994), há três maneiras de recuperar casos, a saber:

- a) buscando diretamente os índices das características;
- b) fazendo uma busca numa estrutura de índices;
- c) fazendo a busca em um modelo de conhecimento mais amplo.

4.2.4.1 SIMILARIDADE

A primeira questão a ser estudada na etapa de recuperação é a similaridade entre o caso alvo (problema de entrada) e o caso na memória a ser comparado. O que faz um caso ser similar ao outro? Esta resposta só é possível ser respondida dentro do domínio de conhecimento da aplicação. Quando a recuperação é do tipo que busca a similaridade diretamente comparando os índices, uma métrica de similaridade é usada para este cálculo.

4.2.4.1.1 MÉTRICA DA SIMILARIDADE

A métrica da similaridade é a medida da similaridade entre dois casos. Em métodos de recuperação baseados em restrições ou classificação, uma métrica talvez seja apenas necessária para estabelecer o grau de similaridade dos casos e ordená-los. Normalmente, são atribuídas diferentes importâncias às características e métodos de agregação como a média ponderada é utilizada.

4.2.4.2 MATCHING E RANKING

Os algoritmos de recuperação procuram diretamente determinados locais na memória, acessando casos com algum potencial para serem úteis, mas isto é feito por heurísticas de *match* e *ranking*, as quais escolhem os casos mais úteis daquele conjunto.

Match: é o processo de comparar dois casos entre si e determinar o seu grau de similaridade. Pode ser produzido um *score* que representa o grau de similaridade ou simplesmente determinar se um caso é similar o suficiente ou não.

Ranking: é o processo de ordenar os casos *partially-matching* de acordo com a utilidade, isto é, determinar qual é melhor que os outros. Frequentemente, o procedimento de *Ranking* usa a saída do procedimento de *Match* para ordenar os casos *partially-matching* de acordo com a sua utilidade (Fernandes, 2001).

4.2.4.3 O VIZINHO MAIS PRÓXIMO (NEAREST NEIGHBOUR RETRIEVAL)

Segundo Fernandes (2001) é uma técnica simples e para entendê-la não é essencial muitos cálculos. O primeiro passo é identificar as características essenciais para a solução do problema. Estes atributos (características) devem ser representados em algum sistema de coordenadas, de tal forma que possa ser medida a distância entre o novo problema e os casos já existentes na base de casos. O especialista pode considerar outros fatores além dos

considerados, transformando cada atributo tenha pesos diferenciados, alterando as distâncias.

A similaridade é expressa pela fórmula a seguir:

$$\text{Similaridade (N, F)} = \sum_{i=1}^n f(N_i, F_i) * w_i$$

Onde: N = caso novo;

F = casos existentes na base de casos;

n = número de atributos;

i = atributo individual;

f = função de similaridade para o atributo i nos casos N e F

w = peso do atributo i.

4.2.4.4 RECUPERAÇÃO INDUTIVA (*INDUCTIVE RETRIEVAL*)

Indução é uma técnica desenvolvida por máquinas de aprendizagem buscando extrair regras ou árvores de decisão de um caso. Em sistemas RBC a base de casos é analisada por um algoritmo de indução para produzir uma árvore de decisão que classifica os casos. O algoritmo mais utilizado é chamado de ID3.

O ID3 constrói uma árvore de decisão da base de casos. Ele usa a heurística chamada de informação mais proveitosa (*information gain*) para encontrar o atributo mais promissor de todos (pelo conhecimento do domínio), dividindo a base de casos. ID3 requer um atributo alvo e uma condição para analisar os casos a partir deste critério.

A construção da árvore de decisão pode ser um processo muito lento, principalmente se a base de casos for grande (Fernandes, 2001).

4.2.4.5 FLAT MEMORY, SERIAL SEARCH

Na *flat memory* casos são armazenados seqüencialmente em uma lista simples de *array* ou arquivo. Casos são recuperados pela aplicação de uma função de casamento seqüencialmente a cada caso do arquivo, mantendo uma trilha de valor do casamento de cada caso e retornando aqueles casos que casaram melhor.

O algoritmo de recuperação é muito simples. O casamento heurístico, de fato, faz todo o trabalho. Existem vantagens e desvantagens neste esquema. A maior vantagem é que a entrada da base de casos é pesquisada. Como um resultado, a precisão da recuperação é apenas uma função de quão boa é a função de casamento. Se o casamento heurístico é bom, então o melhor caso casado sempre será recuperado. Outra vantagem é que adicionar novos casos na base de casos é barato.

A maior desvantagem, claro, é que o esquema é caro. Com uma base de caso grande, então há necessidade de tempo para recuperação. Um esquema como este trabalha quando uma base é pequena, mas rapidamente pega confuso quando a base cresce (Fernandes, 2001).

4.2.4.6 SHARED FEATURE NETWORKS (SFN)

Quando uma base de casos é grande, há a necessidade de organizar casos hierarquicamente para que apenas alguns pequenos conjuntos precisem ser considerados durante a recuperação. Este subconjunto, entretanto, deve ser parecido para ter o melhor casamento ou o caso mais útil nele.

Há vários métodos indutivos que podem ser usados. Por exemplo, pode-se agrupar casos que são similares a outros e se representar o grupo de melhores casamentos com a nova situação, entre apenas itens do grupo são considerados encontrados para melhor casamento do caso. Hierarquias são formadas quando grupos se unem em subgrupos e assim por diante.

Métodos de agrupamento indutivo geralmente procuram por similaridades e por uma série de instâncias de categorias baseadas nesta similaridade.

A maior vantagem é que a recuperação é mais eficiente quando os casos estão organizados em uma lista. Especialmente quando tem que ser feito um casamento de todos os casos, ele considera apenas um subconjunto de casos. Este método também possui desvantagens, pois adicionar casos na base requer trabalho, os casos devem ser colocados no lugar certo. Ele também necessita de mais espaço para alocar a base de casos. É difícil manter a estrutura ótima com a adição de casos. Também não há garantia de que um caso bom não seja esquecido (Fernandes, 2001).

4.2.4.7 DISCRIMINATION NETWORKS (DN)

Árvore de características particionadas agrupa casos similares. Este é um lado deste agrupamento, estas árvores trabalham casos distintos. Agrupar é primordial, discriminação é secundário.

O algoritmo para achar itens na DN inicia no topo da árvore e a primeira questão é feita ao novo caso (entrada). O nodo correspondente a sua resposta é selecionado e a questão neste nodo é respondida. Ele continua até um item único ser encontrado no último nodo da árvore.

As vantagens da DN é que a recuperação é mais eficiente que SFN, pois arcos transversos podem ser implementados de forma mais eficiente do que casamento de subnodos. Outra vantagem é que a compreensão da conexão entre índices e organização da árvore é intuitiva. Os atributos e valores estão separados, fazendo uma trilha das características úteis mais facilmente.

Além das desvantagens das SFNs, a DN não deixa claro como são tratadas as informações perdidas (Fernandes, 2001).

4.2.4.8 REDUNDANT DISCRIMINATION NETWORKS (RDN)

Redundant Discrimination Networks tentam responder ao problema de informações perdidas. Uma RDN organiza os itens usando diferentes DN, cada uma com uma ordem

diferente das questões. As árvores são pesquisadas em paralelo. Se a resposta a uma questão em uma árvore não é conhecida, a pesquisa naquela árvore é interrompida, mas continua em outra árvore. Por causa das redundâncias nas árvores, a última rede estará encontrando um casamento de caso que realmente existe.

Por causa do significativo *overhead* envolvendo RDN, estas árvores são geralmente combinadas com SFN para manter a redundância sob controle.

A RDN possui vantagens como muitos diferentes caminhos para cada item na árvore, seguindo a recuperação sempre quando características são perdidas em um problema. Outra vantagem é que generalizações são formadas quando casos são adicionados (Fernandes, 2001).

4.2.4.9 FLAT LIBRARY, PARALLEL SEARCH

Até agora, os métodos eram como um único processador serial, uma melhor organização do conjunto de casos é a vista quando usamos processadores paralelos.

Numa primeira análise, as pessoas freqüentemente comentam que com um esquema paralelo, toda a indexação e todo o esquema de organização discutido anteriormente torna-se obsoleto. E, de fato, com uma máquina paralela pesada, pode aplicar funções de casamento a todos os casos na ordem deles. Entretanto, se pensa que o problema de indexação é um problema de predizer sob quais circunstâncias um caso particular poderá ser apropriadamente recuperado, então o problema de indexação solucionado se transforma em problema de casamento. Num esquema paralelo, índices podem não precisar de pesquisas diretas; contudo, ainda torna-se necessária designação de que prioridades no casamento e ordenação de casos, porque casamentos de algumas características ou combinação de outras características casam melhor do que de outras, e casos que casam características mais importantes são melhores do que caso que casam características menos importantes.

Uma forma é usar um SIMD-paralelo (instrução simples, dados múltiplos) como memória de casos é armazenar casos num vetor de características, colocando cada vetor de característica em um processador. A recuperação é feita casando a entrada de casos a todos os

vetores de características em paralelo. Isto pode ser implementado através de algoritmos genéticos.

As vantagens da recuperação paralela é que a base de casos completa é pesquisada, a precisão depende apenas da função de casamento, desta forma adicionar um novo caso na base é fácil (Fernandes, 2001).

4.2.4.10 HIERARCHICAL MEMORY, PARALLEL SEARCH

Segundo Fernandes (2001), SFN e DN provêm um modo de selecionar designação de divisão entre casos. Embora máquinas paralelas não possam requerer uma organização para ser imposta nos itens da memória de casos, elas são freqüentemente vantajosas para fazê-lo. Hierarquias são uma maneira econômica para especificação de informações e das inferências.

Apesar das implementações serem feitas em máquinas seriais, há alguns lugares onde MIND-parallelism (instruções múltiplas, dados múltiplos) pode ser adicionado para fazer o processo de recuperação mais eficiente. Um processador pode fazer uma pergunta e então determina qual arco é apropriado para chegar ao novo nodo, chamando um processador adicional para questões em cada nodo.

A memória pode ser organizada hierarquicamente de uma maneira que faça sentido, mas sem índices. Cada caso e cada nodo na hierarquia é associado a um processador. Em um esquema semelhante, recuperar é arquivar espalhando cada valor o par do atributo em uma nova situação na memória, encontrando sempre um nodo na estrutura da hierarquia que casa. O nodo mais específico em cada parte da hierarquia tem alguns pontos a mais que os nodos antecedentes, e o que possuir mais ponto sobre limiar (previamente especificado) ganha.

4.2.4.11 BEST MATCH.

A seleção do best match é a etapa final de recuperação. Segundo algum método de seleção, (pergunta ao usuário, heurísticas, regras, etc.) realiza-se a escolha do caso cuja solução será utilizada como saída do sistema.

4.2.5 REVISÃO E ADAPTAÇÃO

Assim que o "best match" é escolhido, o próximo passo é revisar este caso e sua solução para verificar a necessidade de adaptação ao problema de entrada para, finalmente, utilizar sua solução no problema de entrada.

A etapa de ajuste de situação é necessária quando o usuário encontra dificuldades em apresentar o problema de entrada nos moldes dos casos da base. Se este problema ocorre, deve-se fazer os ajustes necessários para que se possa comparar o caso de entrada com os casos da base.

Esta atividade pode ser tão simples quanto substituir um componente do caso recuperado ou tão complexo quanto modificar a estrutura geral da solução (Fernandes, 2001).

4.2.6 REUTILIZAÇÃO, AVALIAÇÃO E APRENDIZAGEM

Assim que a adaptação é feita, a solução do caso escolhido pode ser então reutilizada para resolver o problema de entrada.

A etapa de avaliação pode ser programada para execução automática ou com a participação do usuário. A avaliação tem por objetivo avaliar a qualidade da solução adaptada ao problema de entrada para definir se esta tem condições de ser adicionada à memória. Alguns autores sugerem que a solução que não foi bem avaliada deve ser incluída de uma observação que permita o sistema agregar este conhecimento para, após repetidas situações, ser retirado da memória.

A aprendizagem pode acontecer tanto com os casos como com a base de dados. A aprendizagem da base de casos acontece com a adição de uma solução adaptada ao problema de entrada, cuja qualidade tenha sido considerada satisfatória.

A aprendizagem além de avaliar a eficiência e qualidade, deve ser considerado se o aumento de robustez resultante da aprendizagem irá realmente beneficiar a qualidade do sistema ou diminuir sua velocidade, utilidade e eficiência;

Avalia-se o sistema comparativamente a especialistas humanos em dois momentos: num primeiro, apenas compara-se o sistema ao especialista; num segundo momento, o especialista utiliza o sistema como um colega ou um assistente técnico. Nesta etapa, calcula-se os percentuais em que o especialista utilizou as sugestões oferecidas pelo sistema (Fernandes, 2001).

4.3 APLICAÇÕES

As aplicações já desenvolvidas de sistemas de RBC são um forte indicativo do potencial do paradigma. Três são os tópicos pertinentes às aplicações que consideramos relevantes: quais os tipos de aplicações são adequados; quais as ferramentas disponíveis para implementação; e quais são as importantes aplicações que podem orientar e ilustrar o potencial dos sistemas de RBC.

Muitos das aplicações que podem ser implementadas em um sistema baseado em casos são comuns a outras técnicas de IA. Genericamente, estes são alguns dos tipos de problemas endereçados: interpretação, projeção, diagnóstico, análise e elaboração de projetos, formação de preços, desenvolvimento de propostas, planejamento, configuração, "scheduling", monitoramento, "debugging", consertos, análise situacional, classificação, instrução, aprendizagem e controle.

As ferramentas para desenvolvimento e implementação de sistemas de RBC não têm a mesma aplicabilidade do que as "*shells*" de outras técnicas de IA pela diversidade de estilos de estruturação das bases de dados destes sistemas. Um dos fatores que torna um problema adequado para ser implementado num sistema de RBC é o fato de existir um banco de dados com os casos para utilização. Sendo assim, muitas vezes é mais eficiente desenvolver o sistema com uma ferramenta que seja compatível com os dados disponíveis do que o oposto. Mesmo assim, as seguintes ferramentas servem como plataforma para implementação de sistemas de RBC: ART*Enterprise; CASE-1; CasePower; CBR2; CBR Express; The Easy Reasoner; Eclipse; ESTEEM; Induce-It; KATE; MEM- 1; ReCall; ReMind; S3-CASE (Fernandes, 2001).

4.4 METODOLOGIAS

A idéia de que a lembrança de experiências passadas está presente na resolução de problemas fundamentou a evolução da técnica de raciocínio baseado em casos, que compreende a seqüência de passos: identificar o problema, buscar o problema mais similar na memória, adaptar, reutilizar e adicionar à base de casos. No desenvolvimento de aplicações desta metodologia, foram implementados sistemas que abordam estes passos distintamente. Algumas aplicações, por diferenciarem-se significativamente em um ou mais destes passos, adotaram nomes que mais se aproximam do principal enfoque de suas implementações. Raciocínio baseado em memória, raciocínio baseado em instância, raciocínio baseado em analogia ou simplesmente raciocínio análogo, raciocínio baseado em exemplares são alguns exemplos.

Todos são metodologias de RBC, mas formam classes distintas. A seguir apresentamos resumidamente algumas destas metodologias. Aamodt & Plaza (1994) discutem algumas destas metodologias:

- a) raciocínio análogo; uma aplicação de raciocínio análogo prevê que a solução do problema de entrada possa ser buscada em universos de conhecimento diferentes do universo do problema de entrada. Como exemplo de aplicação de raciocínio análogo;
- b) raciocínio baseado em memória; caracteriza-se por aplicações com uma memória de dados muito grande, onde sua organização e acesso tomam-se os pontos principais. O raciocínio baseado em memória de Starifill e Waltz, por exemplo, decompõe os problemas de entrada em blocos e cada um destes blocos ativa os elementos da memória paralelamente;
- c) raciocínio baseado em exemplares; este tipo de metodologia classifica os exemplares da memória. Não há adaptação;
- d) raciocínio baseado em metas; este modelo agrega metas ao processo de recuperação que podem ser mutuamente exclusivas;
- e) raciocínio baseado em modelos; o objeto básico é um modelo ao invés de um caso. Os

modelos são estruturas de informação a respeito de um domínio de conhecimento expresso em linguagem formal.

4.5 RESUMO DAS VANTAGENS DO RBC

Nos parágrafos a seguir resume-se as vantagens de um sistema RBC.

A difícil fase de extração do conhecimento especialista é facilitada nos sistemas de RBC, pois a representação do conhecimento pode ser feita com o preenchimento direto dos fatos que descrevem uma experiência. O paradigma proporciona um excelente mecanismo de aprendizagem, que pode ser utilizado para aquisição automática de conhecimento. Para se começar um sistema RBC bastará apenas ter um banco de dados contendo de forma organizada o conhecimento obtido do especialista.

A tarefa de representação do conhecimento resume-se em escolher o tipo de estrutura da base de casos. Em aplicações onde a complexidade exigir, é necessária a criação de índices, entretanto muitas aplicações omitem este passo.

A reutilização do conhecimento contido nos casos pode ser utilizado ou combinado e adaptado para gerar novas soluções além das originalmente presentes na memória.

A aprendizagem que podemos chamar de atualização do conhecimento é feita automaticamente, na medida que as experiências são utilizadas, assim o sistema pode crescer e incrementar sua robustez e eficiência.

As justificativas são sempre consistentes com as soluções por serem as próprias experiências, representando mais um aspecto de proximidade ao comportamento humano do sistema. Além disso, as justificativas podem avisar sobre possíveis riscos que o uso de determinada abordagem pode implicar.

Uma outra vantagem dos sistemas de RBC é uma forma de consciência que estes sistemas possuem com relação a suas próprias limitações. Se o sistema não encontra casos com a devida similaridade com o caso em análise, o sistema não gera solução alguma. Isto evita possíveis problemas gerados por sistemas sob outros paradigmas que podem gerar uma

solução aproximada.

Existem sistemas de RBC que reduzem o espaço de solução. Primeiro estes sistemas podem recuperar uma solução rapidamente, ao contrário de outros sistemas que precisam fazer todo um trabalho para buscar uma solução numa vasta base de conhecimento. Outra vantagem refere-se ao fato de que o problema deve ser identificado pelo sistema o suficiente para recuperar uma solução, não é necessário que o sistema entenda perfeitamente as condições e circunstâncias do problema para propor uma solução. Os sistemas de RBC também proporcionam um meio de resolução de um problema quando não houver um algoritmo disponível para avaliação e solução do mesmo.

A incerteza implícita nas informações contidas nos casos é utilizada sem a necessidade de um tratamento específico desta incerteza.

5 PROJETO DO APLICATIVO

Neste capítulo apresentam-se características de *software* e *hardware*. Para esta aplicação desenvolveram-se várias telas, consultas e alguns relatórios, mas o enfoque principal do trabalho é a utilização do RBC na busca da solução para uma ordem de serviço através da comparação com uma experiência passada semelhante.

No levantamento de informações, fez-se o levantamento de todas as variáveis utilizadas no protótipo, identificando quais os procedimentos que são executados desde a entrada até o encerramento de uma ordem de serviço, os campos e as funções necessárias para o controle destas ordens de serviço e de outras informações relevantes para o sistema de trabalho do setor de operação e manutenção. Estas informações foram adquiridas em um estágio feito em uma empresa de telefonia celular, onde em conversa e debates com o gerente, funcionários e técnicos do setor de operação e manutenção chegou-se à conclusão de que seria viável fazer a automatização do processo de seleção do técnicos mais indicado para resolver os defeitos encontrados nas estações rádio base (ERB).

As ferramentas utilizadas para o desenvolvimento do aplicativo foram escolhidas principalmente pelo conhecimento básico sobre algumas delas, o que facilitou a utilização para alcançar o objetivo proposto pelo trabalho.

Com a utilização do gerenciador de banco de dados Access 97 foi feita a implementação da base de dados (tabela) e o modelo entidade-relacionamento e o desenvolvimento da aplicação, ou seja, toda parte de cadastro, alteração, exclusão e consulta de dados, utilizou-se o ambiente ASP; e no desenvolvimento das páginas para Internet utilizou-se FrontPage 2000.

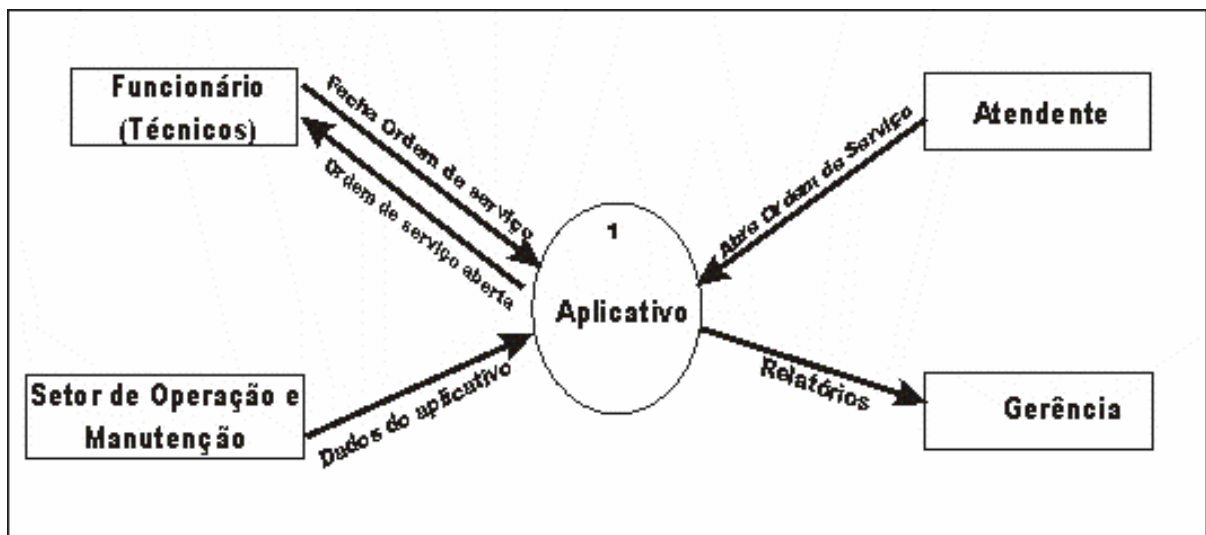
É importante ressaltar que no aplicativo, não foi dada ênfase à ergonomia e à segurança do aplicativo e sequer todas as opções exigidas para um controle eficaz de ordem de serviço puderam ser incluídas devido à complexidade e a abrangência da proposta, mas foram desenvolvidas as funções mais importantes e necessárias para exemplificar com clareza o objetivo do trabalho.

5.1 ESPECIFICAÇÃO

A metodologia utilizada no desenvolvimento da especificação do aplicativo foi à análise estruturada. A análise estruturada se propõe a fornecer um meio de comunicação comum entre o usuário e o analista, um meio de lapidar o sistema do usuário no início do projeto e uma especificação manutenível, que seja não-redundante, não-ambígua e completa. A técnica utilizada para a análise foi a de Yourdon/Demarco (Keller, 1990) e (Petry, 2001).

A figura 3 apresenta o diagrama de contexto onde são mostradas as atividades externas envolvidas no aplicativo. A entidade funcionário está relacionada com o aplicativo por a figura chave é ele que vai receber a ordem de serviço, executá-la e depois informar ao aplicativo que a ordem já foi encerrada; a entidade atendente é responsável pela abertura da ordem de serviço; a entidade gerência lê relatório e executa consultas no aplicativo; a entidade setor de operação e manutenção coloca como entidade responsável pela alimentação do aplicativo, são eles que atualizam as tabelas .

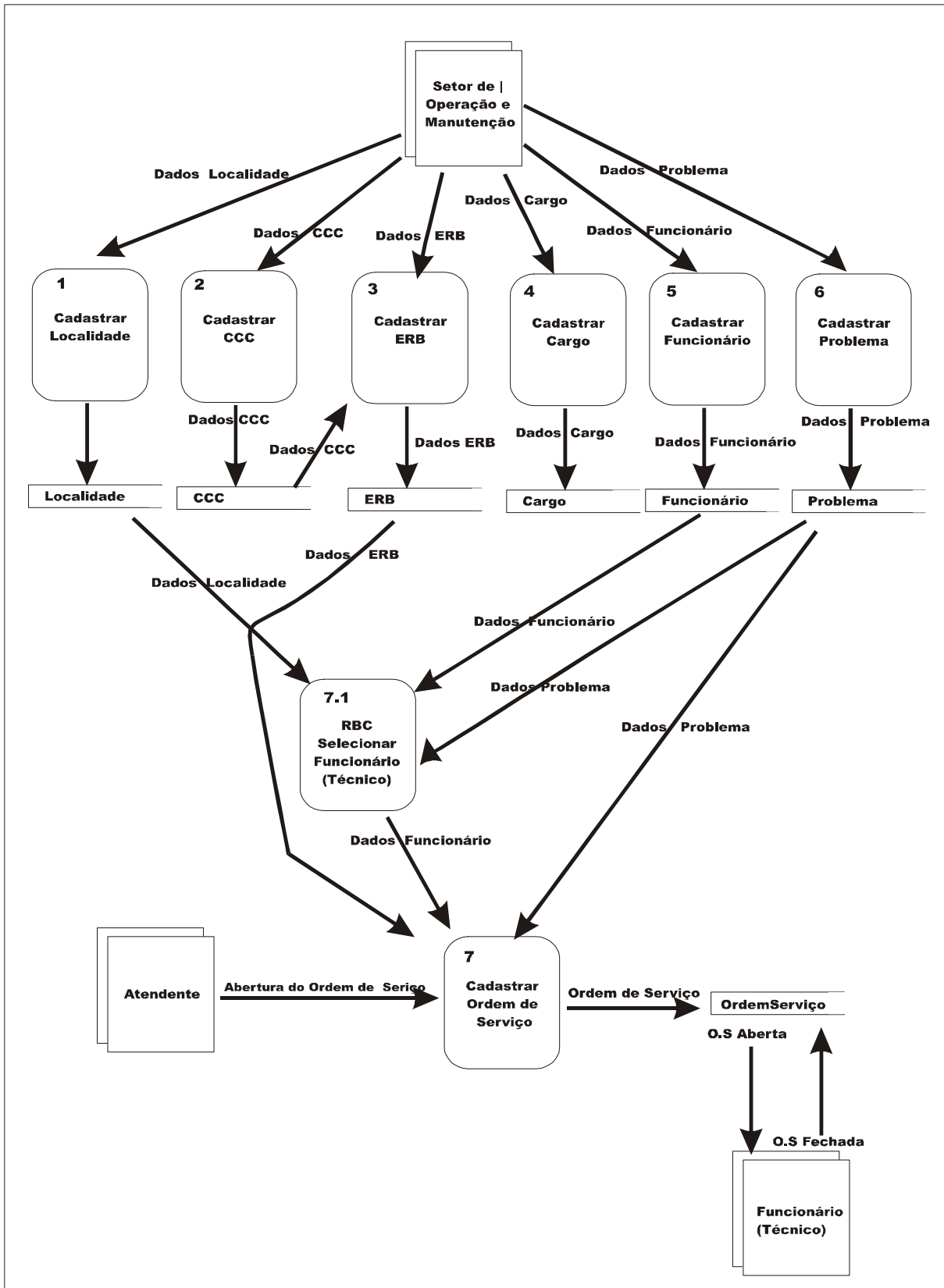
Figura 3 – diagrama de contexto



Na figura 4 será apresentado o diagrama de fluxo de dados nível zero, onde cada processo está representado por uma seqüência de número que podemos descrever da seguinte forma:

- a) processo 1 cadastro localidade;
- b) processo 2 cadastro de CCC. Este cadastro vem antes do cadastro de ERB, pois uma CCC possui várias ERBs;
- c) processo 3 cadastro de ERB;
- d) processo 4 cadastro de cargo;
- e) processo 5 cadastro de funcionário (Técnico);
- f) processo 6 cadastro de problemas;
- g) processo 7 cadastro de ordem de serviço;
- h) processo 7.1 seleção de funcionário (Técnico) acontece entre o cadastro de ordem de serviço e a confirmação da mesma, pois ocorre a aplicação da técnica de RBC.

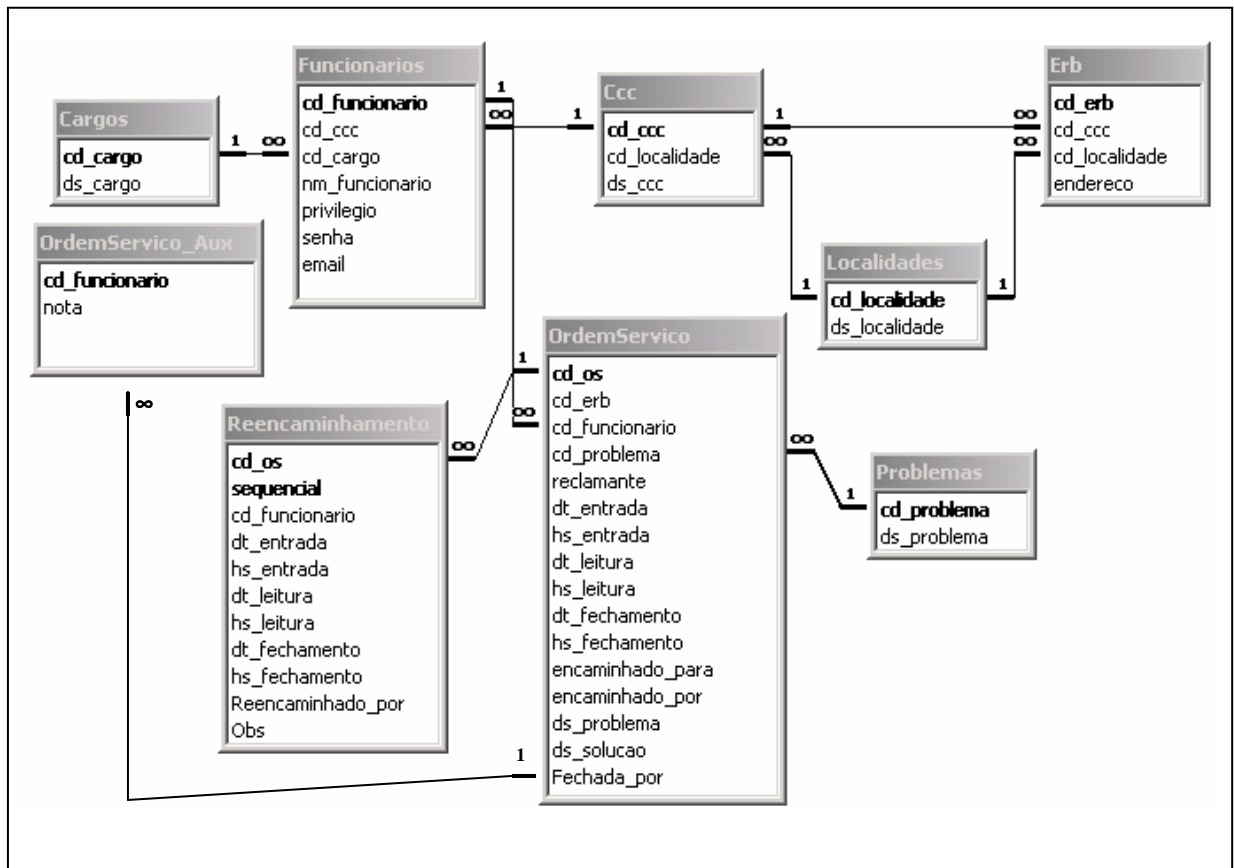
Figura 4- diagrama de fluxo de dados – nível zero.



A figura 5 apresenta o modelo de entidade e relacionamento (MER), onde são apresentadas as tabelas do sistema e seus relacionamentos. Para explicar melhor o que a figura 6 mostra, tem-se o seguinte:

- a) ordemserviço: armazena cada ordem de serviço criada;
- b) problemas: armazena informações sobre os possíveis problemas;
- c) reencaminhamentos: armazena cada ordem de serviço que é reencaminhada;
- d) funcionários: armazena informações sobre os funcionários aptos a solucionar os problemas nas estações rádio-base (ERB);
- e) cargos: armazena informações sobre o cargo de cada funcionário;
- f) ordemserviço_aux: utilizada apenas para o processamento do RBC;
- g) ccc: armazena informações sobre cada centro de comutação e controle (CCC);
- h) localidades: armazena informações sobre localidade das ERBs e CCCs;
- i) erb: armazena informações sobre cada estação rádio base(ERB).

Figura 5- Modelo de Entidade e Relacionamento (MER).



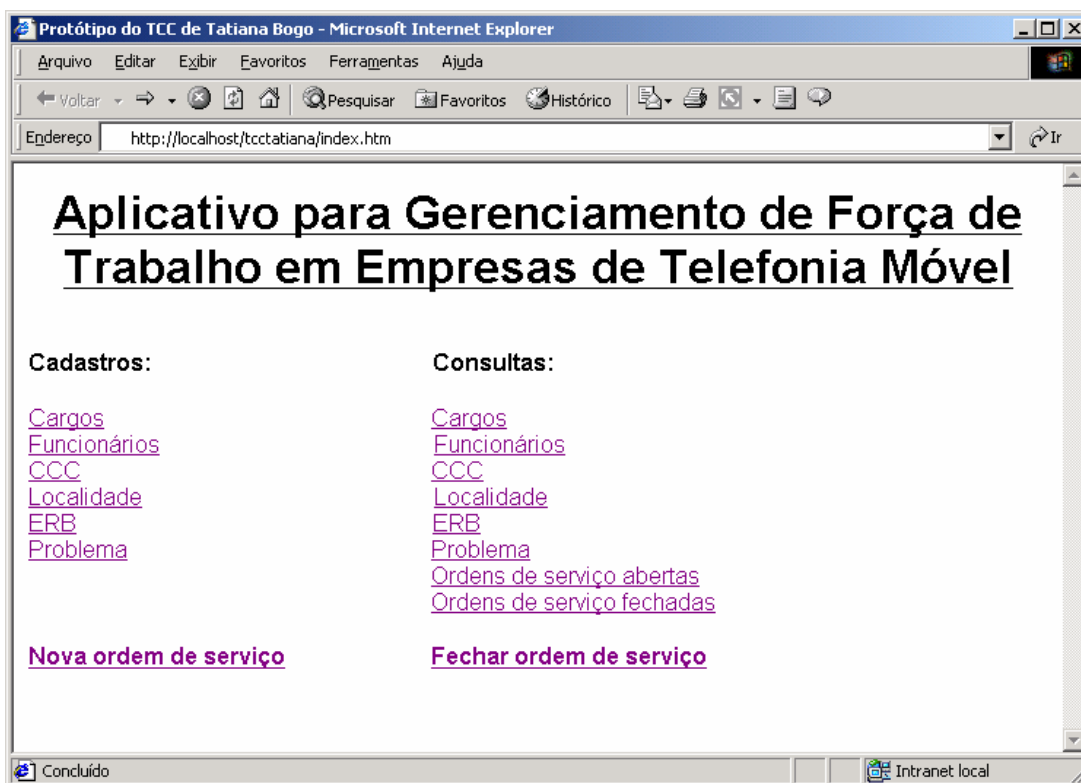
5.2 APRESENTAÇÃO DO SISTEMA

A implementação do sistema foi organizada em três módulos principais (Cadastros, Consultas e Ordem de Serviço) e cada módulo com suas respectivas funções. Serão apresentadas as funções mais importantes de cada módulo, deixando de lado detalhes de implementação.

5.2.1 TELA PRINCIPAL

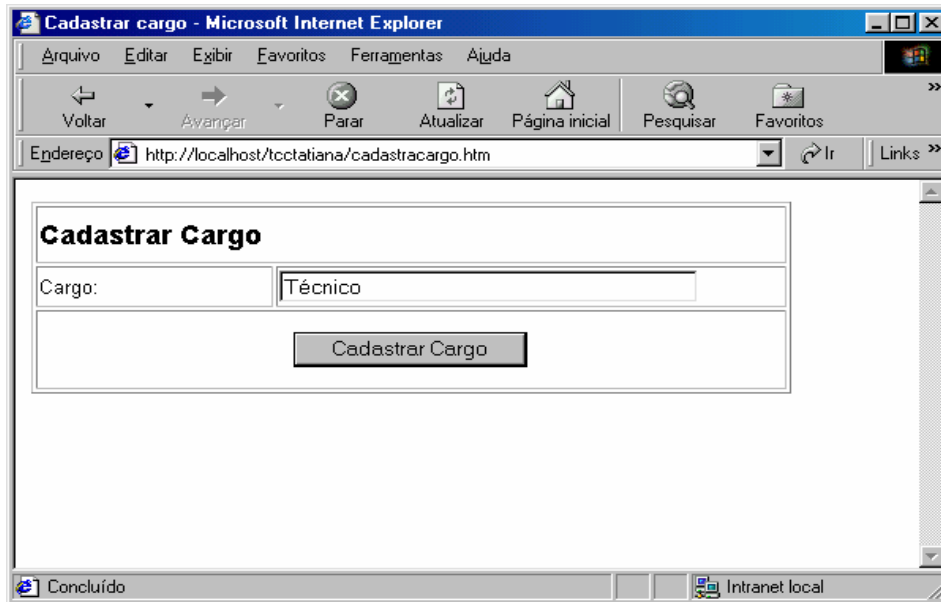
A figura 6 mostra a tela do menu principal onde o usuário tem acesso através da internet a todas as demais opções do sistema.

Figura 6– tela do menu principal



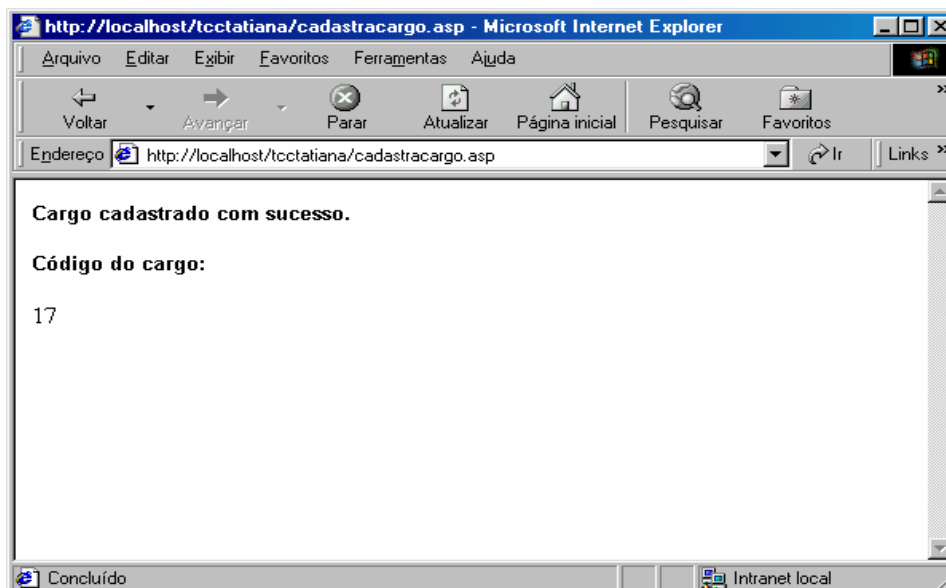
A figura 7 mostra a tela de cadastramento de cargos que cada funcionário poderá ocupar na empresa. A tabela cargos está relacionada com a tabela funcionários como mostrada na figura 5.

Figura 7– tela cadastrar cargo



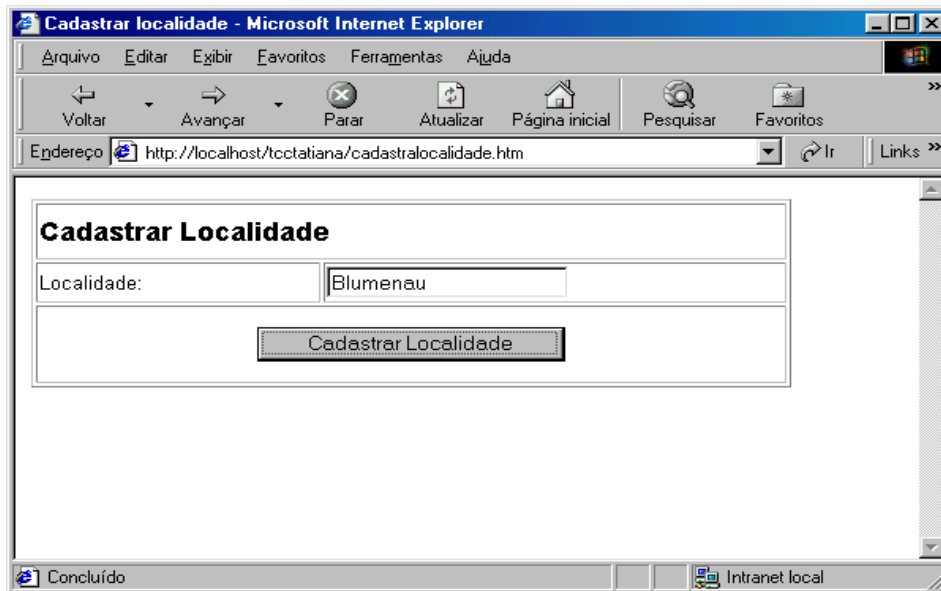
A figura 8 mostra a tela de confirmação do cadastramento de cargos

Figura 8– tela de confirmação do cadastro de cargos



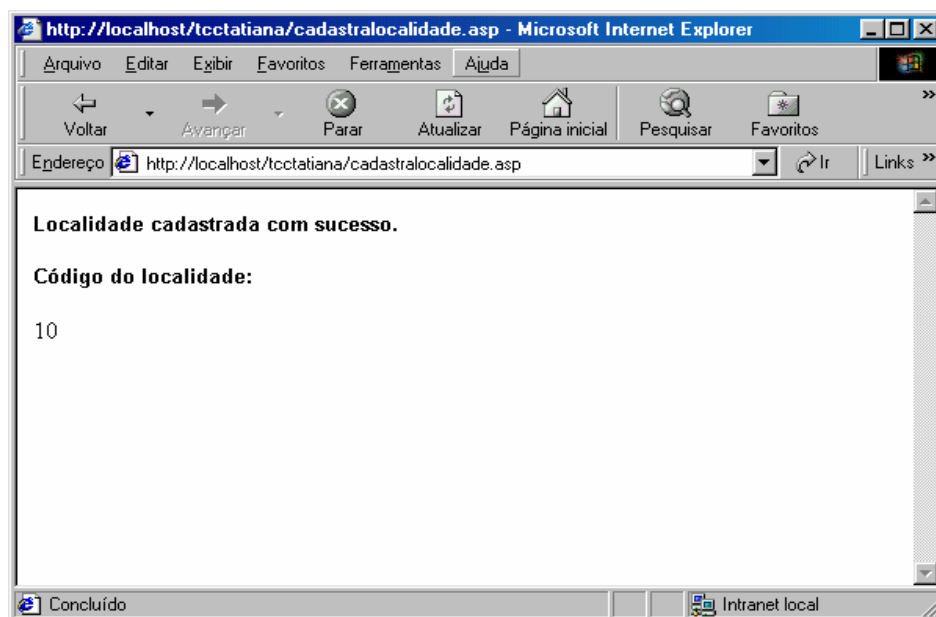
A figura 9 mostra a tela de cadastramento de localidades da CCC e da ERB. A tabela localidade está relacionada com a tabela CCC e a tabela ERB como mostrada na figura 5.

Figura 9– tela cadastrar localidade



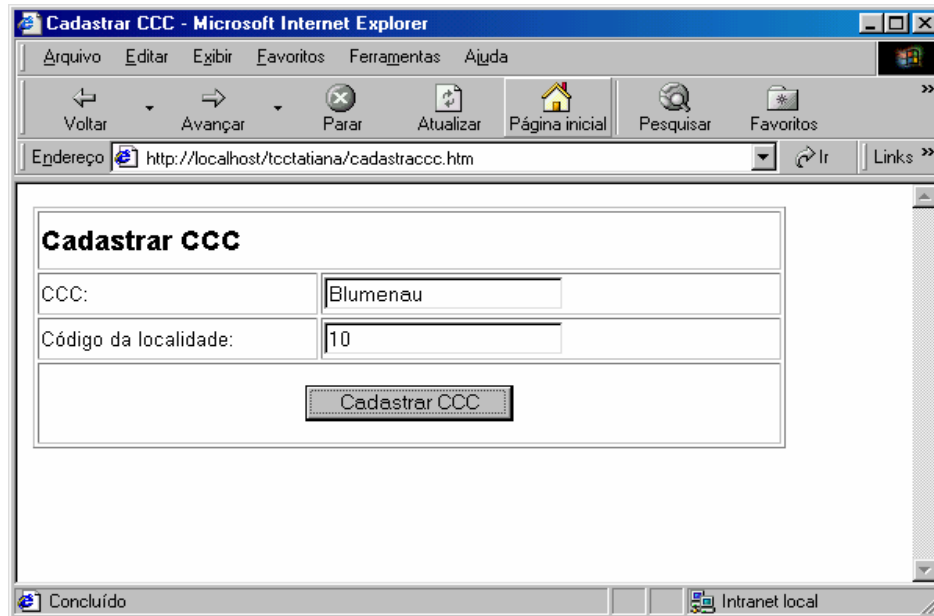
A figura 10 mostra a tela de confirmação do cadastramento das localidades.

Figura 10– tela de confirmação do cadastro de localidade



A figura 11 mostra a tela de cadastramento de Centro de Comutação e Controle (CCC). A tabela CCC está relacionada com as tabelas ERB, Localidade e Funcionários como mostrada na figura 5.

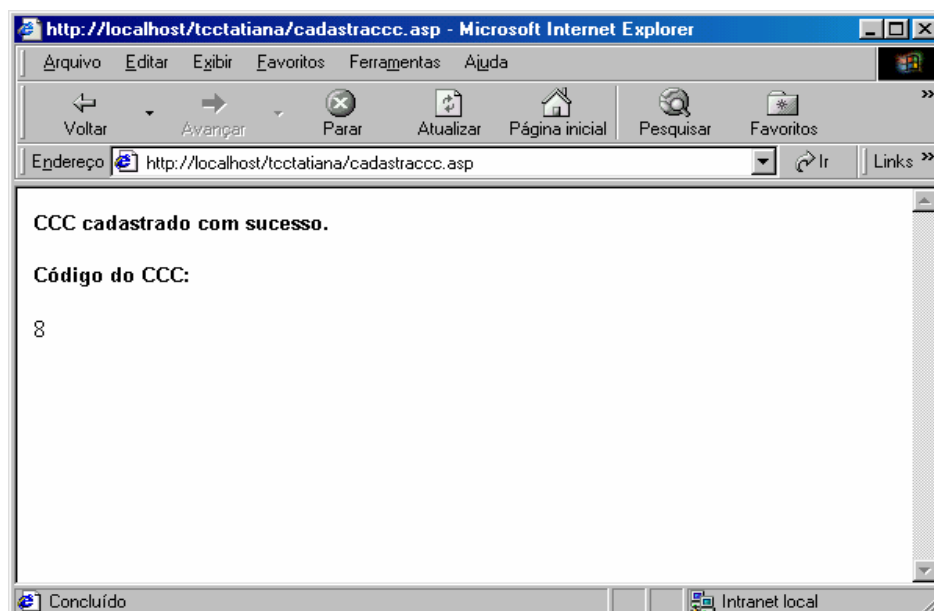
Figura 11– tela cadastrar CCC



The screenshot shows a web browser window titled "Cadastrar CCC - Microsoft Internet Explorer". The address bar displays "http://localhost/tcctatiana/cadastracc.htm". The browser's menu bar includes "Arquivo", "Editar", "Exibir", "Favoritos", "Ferramentas", and "Ajuda". The navigation bar contains "Voltar", "Avançar", "Parar", "Atualizar", "Página inicial", "Pesquisar", and "Favoritos". The main content area displays the form "Cadastrar CCC" with two input fields: "CCC:" containing "Blumenau" and "Código da localidade:" containing "10". A "Cadastrar CCC" button is positioned below the fields. The status bar at the bottom indicates "Concluído" and "Intranet local".

A figura 12 mostra a tela de confirmação do cadastramento de CCC

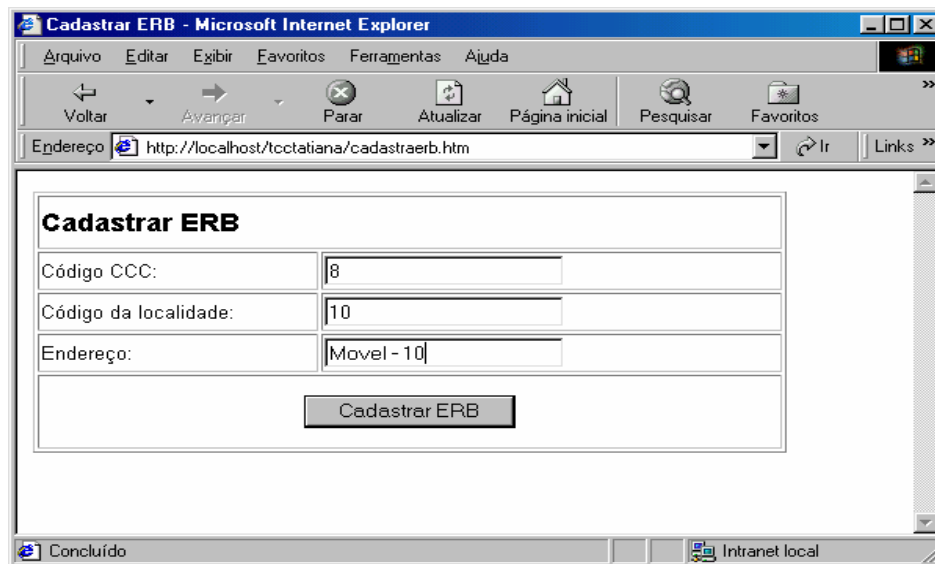
Figura 12– tela de confirmação do cadastro de CCC



The screenshot shows a web browser window titled "http://localhost/tcctatiana/cadastracc.asp - Microsoft Internet Explorer". The address bar displays "http://localhost/tcctatiana/cadastracc.asp". The browser's menu bar includes "Arquivo", "Editar", "Exibir", "Favoritos", "Ferramentas", and "Ajuda". The navigation bar contains "Voltar", "Avançar", "Parar", "Atualizar", "Página inicial", "Pesquisar", and "Favoritos". The main content area displays the confirmation message "CCC cadastrado com sucesso." followed by "Código do CCC:" and the value "8". The status bar at the bottom indicates "Concluído" and "Intranet local".

A figura 13 mostra a tela de cadastramento da Estação Rádio-Base (ERB). A tabela ERB está relacionada com a tabela CCC e a tabela Localidade como mostrada na figura 5.

Figura 13– tela cadastrar ERB



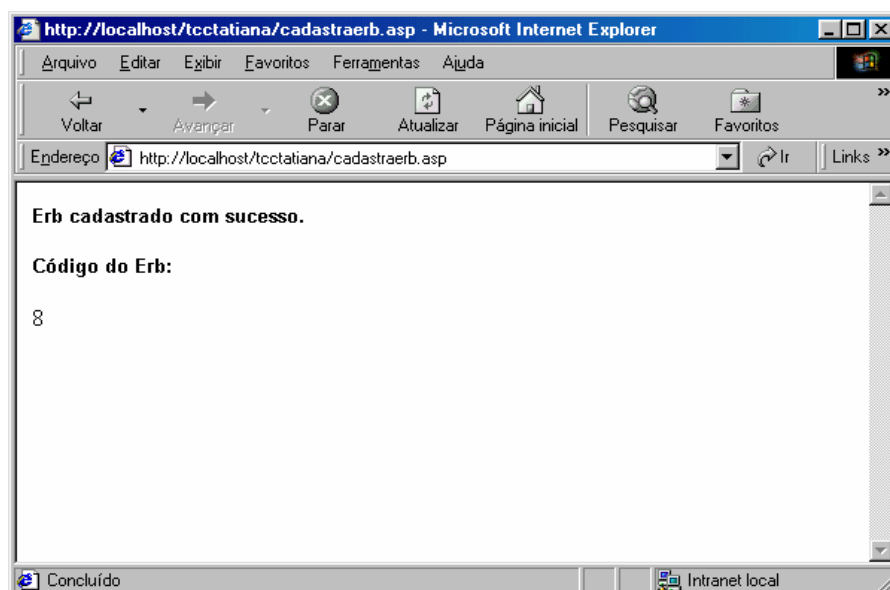
The screenshot shows a web browser window titled "Cadastrar ERB - Microsoft Internet Explorer". The address bar displays "http://localhost/tcctatiana/cadastreaerb.htm". The page content includes a form with the following fields and values:

Cadastrar ERB	
Código CCC:	8
Código da localidade:	10
Endereço:	Movel - 10
<input type="button" value="Cadastrar ERB"/>	

The status bar at the bottom indicates "Concluído" and "Intranet local".

A figura 14 mostra a tela de confirmação do cadastramento das Estações Rádio-Base (ERB).

Figura 14– tela de confirmação do cadastro de ERB



The screenshot shows a web browser window titled "http://localhost/tcctatiana/cadastreaerb.asp - Microsoft Internet Explorer". The address bar displays "http://localhost/tcctatiana/cadastreaerb.asp". The page content displays a confirmation message:

Erb cadastrado com sucesso.

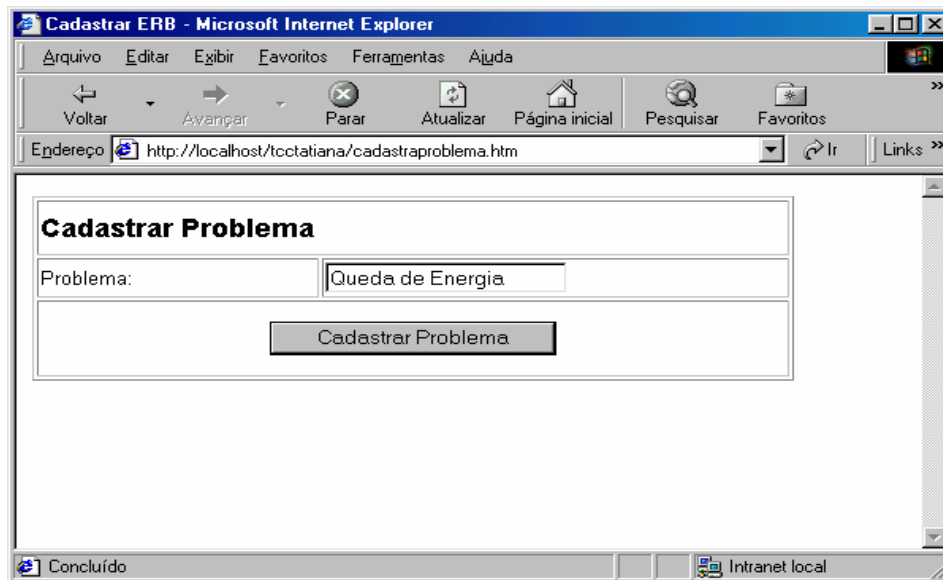
Código do Erb:

8

The status bar at the bottom indicates "Concluído" and "Intranet local".

A figura 15 mostra a tela de cadastramento dos possíveis problemas com a Estação Rádio-Base (ERB). A tabela Problema está relacionada com a tabela OrdemServiço como mostrada na figura 5.

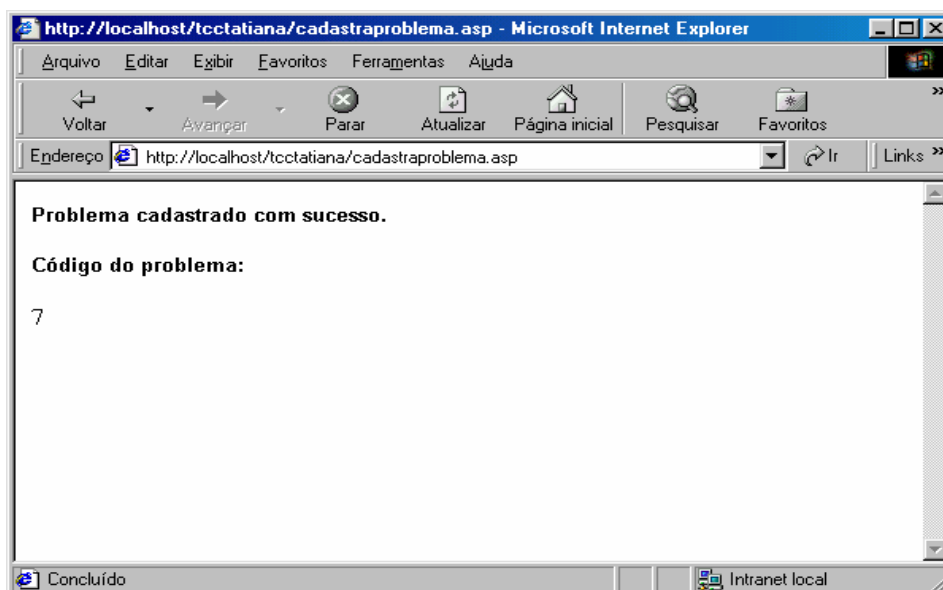
Figura 15– tela cadastrar Problema



The screenshot shows a web browser window titled "Cadastrar ERB - Microsoft Internet Explorer". The address bar displays "http://localhost/tcctatiana/cadastraproblema.htm". The main content area contains a form titled "Cadastrar Problema". The form has a label "Problema:" followed by a text input field containing the text "Queda de Energia". Below the input field is a button labeled "Cadastrar Problema". The browser's status bar at the bottom shows "Concluído" and "Intranet local".

A figura 16 mostra a tela de confirmação do cadastramento dos Problemas.

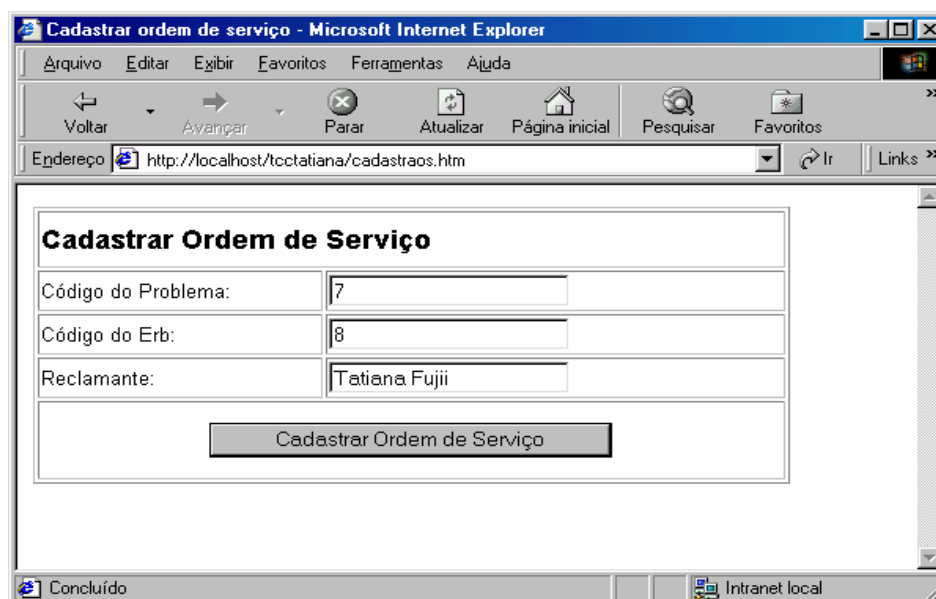
Figura 16– tela de confirmação do cadastro de Problemas



The screenshot shows a web browser window titled "http://localhost/tcctatiana/cadastraproblema.asp - Microsoft Internet Explorer". The address bar displays "http://localhost/tcctatiana/cadastraproblema.asp". The main content area displays the message "Problema cadastrado com sucesso." followed by "Código do problema:" and the number "7". The browser's status bar at the bottom shows "Concluído" and "Intranet local".

A figura 17 mostra a tela de cadastro de uma nova ordem de serviço. A tabela OrdemServiço está relacionada com a tabela Funcionários, Problemas e Reencaminhamento como mostrada na figura 5.

Figura 17– tela cadastra Ordem de Serviço



The screenshot shows a Microsoft Internet Explorer browser window titled "Cadastrar ordem de serviço - Microsoft Internet Explorer". The address bar displays "http://localhost/tcotatiana/cadastros.htm". The main content area contains a form titled "Cadastrar Ordem de Serviço" with the following fields:

Código do Problema:	<input type="text" value="7"/>
Código do Erb:	<input type="text" value="8"/>
Reclamante:	<input type="text" value="Tatiana Fujii"/>

Below the form is a button labeled "Cadastrar Ordem de Serviço". The browser's status bar at the bottom shows "Concluído" and "Intranet local".

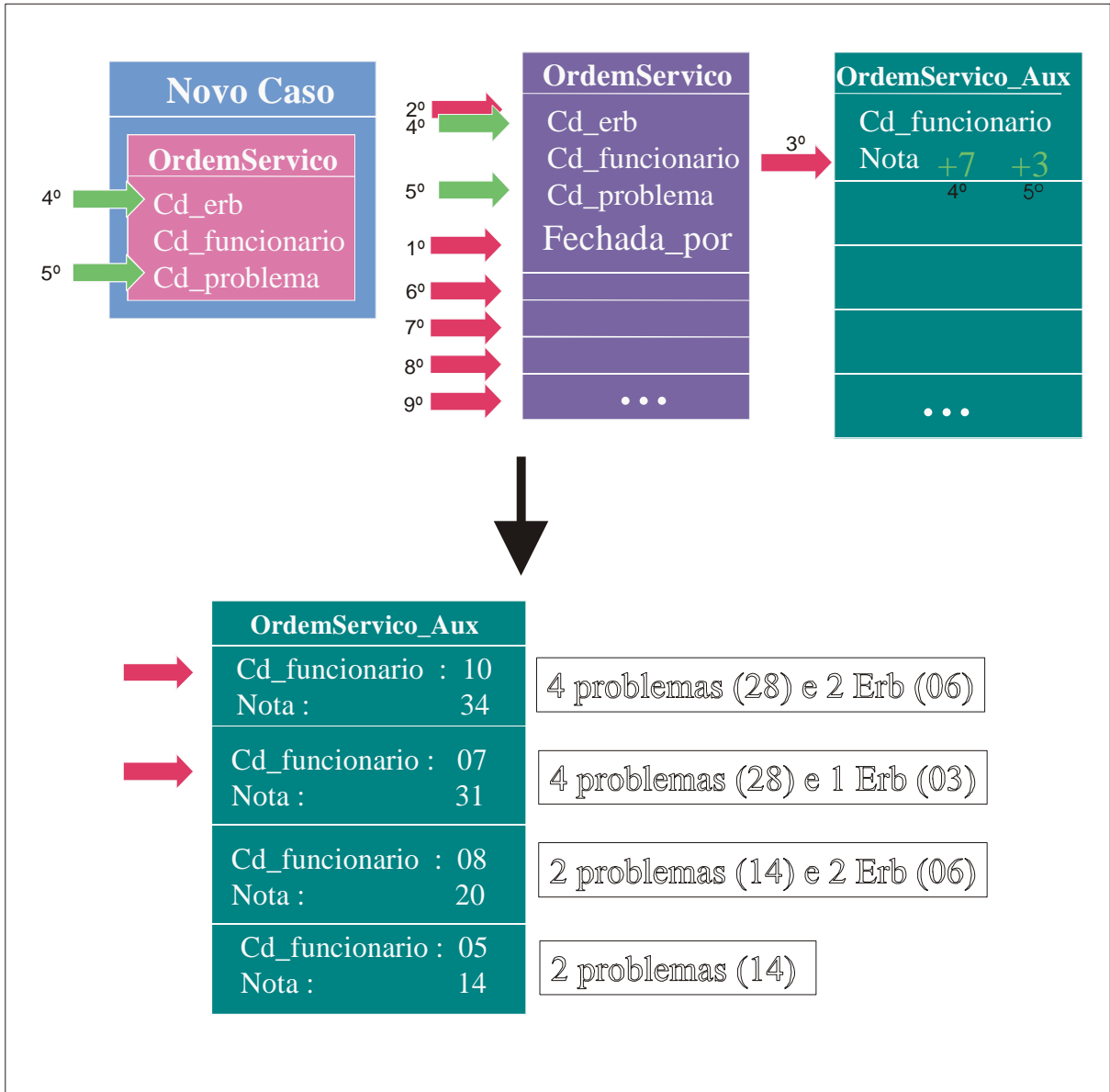
Após clicar o botão Cadastrar Ordem de Serviço o sistema irá ativar o algoritmo do RBC (Quadro 1) para a pesquisa e escolha do técnico mais apropriado para a realização da tarefa. O algoritmo pode ser visto no quadro 01.

A técnica de RBC utilizada neste algoritmo para a recuperação das informações é a de Matching e Ranking que compara dois casos entre si determinando o grau de similaridade (Nota) e após comparar todos os casos da base ordena e escolhe pelo grau de similaridade o melhor, neste caso o funcionário que tem a maior nota.

No aplicativo a comparação de similaridade é feita através das tabelas OrdemServiço e OrdemServiço_Aux, conforme figura 18, o algoritmo varre a tabela OrdemServiço_Aux e limpa todas as informações constantes, em seguida começa a comparação entre a ordem de serviço a ser cadastrada e as ordens de serviço constante na base de dados. O algoritmo segue os passos relacionados abaixo:

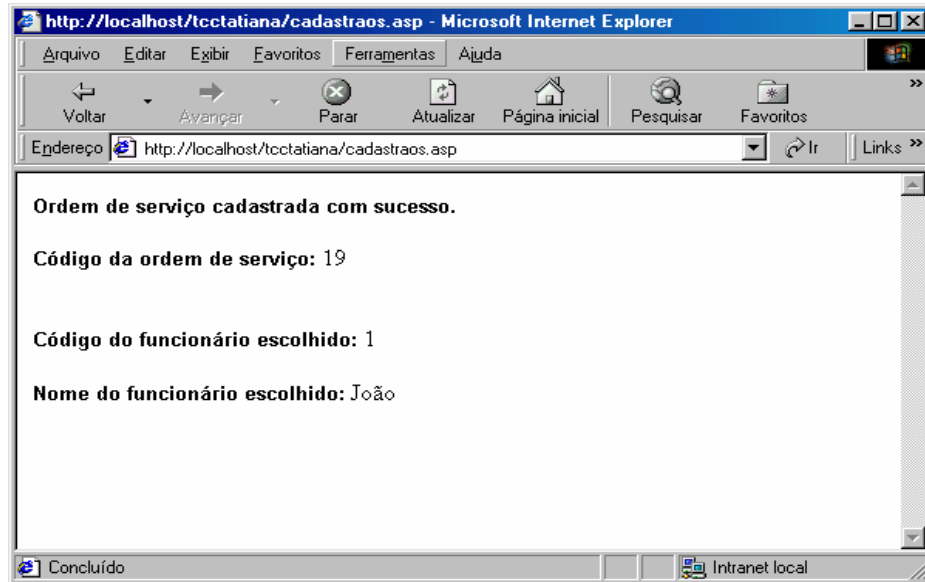
- a) 1°. Passo: Verificar se a ordem de serviço está fechada;
- b) 2°. Passo: Verificar se o funcionário trabalha na CCC em que a ERB está cadastrada;
- c) 3°. Passo: Começa o cálculo da nota de cada funcionário;
- d) 4°. Passo: Comparar a variável Cd_erb das tabelas Ordem Serviço e OrdemServiço_Aux, se forem iguais será atribuído à nota o valor +7 e caso a comparação seja diferente não será somado nenhuma nota;
- e) 5°. Passo: Comparar a variável Cd_problema das tabelas Ordem Serviço e OrdemServiço_Aux, se forem iguais será atribuído à nota o valor +3 e caso a comparação seja diferente não será somado nenhuma nota;
- f) após percorrer toda a tabela OrdemSevico é feito uma ordenação em ordem decrescente da tabela OrdeServiço_aux;
- g) o sistema verificará:
 - se o técnico de código 10 tem mais de cinco ordens de serviço aberta o aplicativo selecionará o técnico de código 7;
 - se o técnico de código 7 também tiver mais de cinco ordens de serviço aberta o aplicativo então selecionará o técnico de código 10;
- h) o aplicativo dará ao usuário duas opções:
 - usuário escolher seu próprio funcionário;
 - aceitar o funcionário indicado;
- i) após o atendente selecionar uma das opções acima o sistema abrirá a ordem de serviço para o técnico selecionado.

Figura 18– comparação de similaridade



A figura 19 mostra a tela de confirmação da Ordem de Serviço e informa qual é o técnico que vai receber esta ordem de serviço.

Figura 19– tela de confirmação da Ordem de serviço



Quadro 1 - Algoritmo do RBC

```
Código do Cadastro de Ordem de Serviço:

<% @LANGUAGE = VBScript %>

<%
Option Explicit
Response.Expires = 0
%>

<!--#include file="adovbs.inc"-->

<%
Dim objConn, objRs, objRsAux, strConn, strOutput, StrQ, xCod_Problema,
xCod_Erb, xNota

Dim xCodigoFuncionarioEscolhido, xNomeFuncionarioEscolhido

xCod_Problema = Request.Form("edCodProblema")
xCod_Erb = Request.Form("edCodErb")

Set objConn = Server.CreateObject("ADODB.Connection")
strConn = "Data Source=TCCTatiana;User ID=sa;Password=;"

objConn.Open strConn

Set objRsAux = Server.CreateObject("ADODB.RecordSet")
Set objRsAux.ActiveConnection = objConn

'-- procurar funcionário mais adequado para resolver o problema
'-- deleta todos os registros existentes na tabela OrdemServico_Aux
strQ = "select * from OrdemServico_Aux"

objRsAux.CursorLocation = 2 '--adUseServer
```



```
objRsAux.CursorType = 1 '--adOpenKeyset
objRsAux.LockType = 3 '--adLockOptimistic

objRSAux.Open strQ
while not objRSAux.EOF
    objRSAux.Delete
    objRSAux.MoveNext
Wend

objRSAux.Close
Set objRSAux = Nothing

strQ = "select * from OrdemServico"

'--objRS.Open strQ
Set objRS = objConn.Execute(strQ)
while not objRS.EOF
    xNota = 0

    If (objRS("Fechada_por") <> 0) then
        if (objRS("cd_erb") = xCod_Erb) then
            xNota = xNota + 7

        End if

        if (objRS("cd_problema") = xCod_Problema) then
            xNota = xNota + 3

        End if
    End if
End if
```

```
'-- atualiza nota do funcionário na tabela auxiliar OrdemServico_Aux
'-- se o funcionário ainda não existe na tabela ele é adicionado

StrQ = ""

StrQ = "select * from OrdemServico_Aux where cd_funcionario = " &
objRS("Fechada_por")

Set objRsAux = Server.CreateObject("ADODB.RecordSet")

Set objRsAux.ActiveConnection = objConn

objRsAux.CursorLocation = 2 '--adUseServer
objRsAux.CursorType = 1 '--adOpenKeyset
objRsAux.LockType = 3 '--adLockOptimistic

    objRSAux.Open strQ

If (objRsAux.Eof) then

    objRsAux.AddNew

    objRsAux("cd_funcionario") = objRS("Fechada_por")
    objRsAux("Nota") = xNota

    objRsAux.Update

Else

    objRsAux("Nota") = objRsAux("Nota") + xNota

    objRsAux.Update

End If

objRSAux.Close

Set objRSAux = Nothing

objRS.MoveNext

Wend
```

```
'-- lê funcionário com a maior nota

StrQ = "SELECT cd_funcionario, nota FROM OrdemServico_Aux where nota =
(select Max(OrdemServico_Aux.Nota) from OrdemServico_Aux)"

Set objRsAux = ObjConn.Execute(StrQ)

'-- Response.Write "cd_funcionario: " & objRsAux("cd_funcionario")
'-- Response.Write "Nota: " & objRsAux("nota")

xCodigoFuncionarioEscolhido = objRsAux("cd_funcionario")

ObjRSAux.Close

Set objRSAux = Nothing

objRS.Close

Set objRS = Nothing

'-- procura nome do funcionário escolhido

StrQ = "SELECT nm_funcionario FROM Funcionarios where cd_funcionario = " &
xCodigoFuncionarioEscolhido

Set objRsAux = ObjConn.Execute(StrQ)

xNomeFuncionarioEscolhido = objRsAux("nm_funcionario")

Set objRs = Server.CreateObject("ADODB.RecordSet")

Set objRs.ActiveConnection = objConn

'--

objRS.CursorLocation = 2 '--adUseServer

objRS.CursorType = 1 '--adOpenKeyset

objRS.LockType = 3 '--adLockOptimistic

objRS.Open "OrdemServico", objConn, , , &H0002 '--adCmdTable
```

```
objRS.AddNew

objRS("cd_problema") = Request.Form("edCodProblema")

objRS("cd_erb") = Request.Form("edCodErb")

objRS("reclamante") = Request.Form("edReclamante")

objRS.Update

strOutput = objRS("cd_os")

Response.Write "<p><font face=Arial size=2><b>Ordem de serviço cadastrada
com sucesso. </b></font></p>"

Response.Write "<p><font face=Arial size=2><b>Código da ordem de serviço:
</b></font>" & strOutput & "</p>"

Response.Write "<BR>"

Response.Write "<p><font face=Arial size=2><b>Código do funcionário
escolhido: </b></font>" & xCodigoFuncionarioEscolhido & "</p>"

Response.Write "<p><font face=Arial size=2><b>Nome do funcionário
escolhido: </b></font>" & xNomeFuncionarioEscolhido & "</p>"

objRS.Close : objConn.Close

Set objRS = Nothing

Set objConn = Nothing

%>
```

6 CONCLUSÃO

O objetivo principal deste trabalho é automatizar o processo de ordem de serviço do setor de operação e manutenção de uma empresa de telefonia celular.

Foram estudados vários algoritmos de RBC para poder recuperar da base de dados o técnico mais apropriado para resolver uma ordem de serviço cadastrada dentre aqueles que foram encontrados durante o desenvolvimento do trabalho: *match* e *ranking*. O algoritmo de *match* e *ranking* mostrou-se bastante eficiente, tornando a implementação do aplicativo mais simplificado.

O aplicativo desenvolvido pode auxiliar as empresas de telefonia celular a reduzir o tempo de contato com seus técnicos e tornado mais rápido a manutenção de sua ERBs e com isso satisfazendo melhor as necessidades de seus clientes.

Durante o desenvolvimento do protótipo foram utilizadas 3 ferramentas: *Personal Web Server* (PWS) para compilar o ASP, *FrontPage2000*, para construir as páginas e editar o código ASP e o *Access 98* para criar a base de dados e o modelo entidade-relacionamento. Cabe ressaltar que com a utilização do ASP temos aos scripts rodando no servidor e somente os resultados apresentados para o usuário, fazendo com que qualquer *browser* do mercado seja capaz de acessar a página do aplicativo e ainda podemos ressaltar a utilização do RBC como um meio inteligente de utilizar a computação para solucionar problemas que até então “só o homem seria capaz de resolver”.

6.1 DIFICULDADES ENCONTRADAS

As maiores dificuldades encontradas foram aquisição de material para desenvolver a parte textual e lógica do RBC.

6.2 EXTENSÕES

Pode-se aplicar outros algoritmos de RBC para poder recuperar da base de dados o técnico mais apropriado para resolver uma ordem de serviço cadastrada.

Pode-se estudar também formas de utilizar *Data mining* junto com o RBC para não deixar o aplicativo lento com o tempo.

Seria interessante fazer um controle de tempo entre a leitura da ordem de serviço e a hora do fechamento da mesma.

Outra possibilidade é a criação de controle de disponibilidade de cada técnico em horários que os mesmo estão se deslocando de uma Erb à outra.

7 REFERÊNCIAS BIBLIOGRÁFICAS

AAMODT, A & PLAZA, E, *Case-Based Reasoning: Foundational Issues, Methodological Variation, and System Approaches*, In: *Artificial Intelligence Communications*, 7(1), 39-59, 1994.

ANATEL. **Serviço de telecomunicações**, out. [2001?]. Disponível em: <<http://www.anatel.gov.br/serviços/smc/smc.asp/>> Acesso em: 10 out. 2001.

BARRETO, Jorge Muniz. **Inteligência Artificial no limiar do século XXI**. Florianópolis: J.M Barreto, 1997.

BITTENCOURT, Guilherme. **Inteligência artificial: ferramentas e teorias**. Florianópolis: Ed.da UFSC, 1998.

BURD, Daniel José. **Access avançado 1.1**. São Paulo: Makron Books, 1994.

CELULAR, Planeta. **Guia completo sobre telefonia celular**, out. [2001]. Disponível em: <<http://www.planetacelular.com.br/default2.htm/>>. Acesso em: 10 out. 2001.

COSTA, Marcello Thiry Comicholi. **Uma arquitetura baseada em agentes para suporte ao ensino à distância**. Florianópolis SC [1999]. Disponível em: <<http://www.eps.ufsc.br/teses99/thiry>>. Acesso em 08 outubro 2001.

FERNANDES, R. Anita Maria. **Raciocínio Baseado em Casos**. Itajaí, novembro. [2001]. Disponível em: <http://inf.univali.br/~anita/paginas_ia/index.htm>. Acesso em: 21 nov. 2001.

KELLER, Robert. **Análise estruturada na prática**. São Paulo: McGraw-Hill, 1990.

KENT, Peter. **JavaScript para netscape: guia oficial**. São Paulo: Makron Books, 1997.

LEVINE, Robert I, **Inteligência Artificial e Sistemas Especialistas**. São Paulo: McGraw-Hill, 1988.

MCFEDRIES, Paul. **Guia incrível do Access 2.0**. São Paulo: Makron Books, 1995.

OLIVEIRA, Adelize Generini de. **Java Script**. Florianópolis: Bookstore, 1996.

PEREIRA, Marco Antônio. **Conceitos Gerais**, março. [2000]. Disponível em: <<http://www.marco.eng.br/>>. Acesso em: 10 out. 2001.

PETRY, G. Patrícia. **Análise e projetos de sistema I**, Florianópolis, novembro. [2001]. Disponível em: <<http://inf.unisul.br/~patricia/>>. Acesso em: 21 nov. 2001.

PROWAY, Informática Ltda. **Curso básico de ASP**. Blumenau: Proway, 2001.

RABUTSKE, Renato Antônio. **Inteligência artificial**. Florianópolis: Ed da UFSC, 1995.

RADUENZ, Gilmar Vereano. **Protótipo de aplicativo para escritório de advocacia com acesso a Internet baseado em ASP**. 2000. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

RICH, Elaine. **Inteligência artificial**. São Paulo: McGraw-Hill, 1988.

TIM, Celular. **Sistema Móveis Celulares**. Florianópolis: Tim Celular, 1999.

UNIFORMARE, Treinamento e Informática Ltda. **Apostila do Curso Dinâmico de Informática**. Blumenau: Uninformare, 2001.

WEISSINGER, A. Keyton. **ASP – Guia Completo**. Rio de Janeiro: Ciência Moderna, 1999.

WILLE, Christoph. KOLLER, Christian. **Aprenda em 24 horas active server pages**. Rio de Janeiro: Campus, 1999.

WINSTON, Patrick Henry. **Inteligência artificial**. Rio de Janeiro: Livros Técnicos e Científicos, 1987.