

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

**SISTEMA DE AJUDA NA RESOLUÇÃO DE PROBLEMAS NA
ABERTURA DE ARQUIVOS DE ANÚNCIOS DE UM JORNAL
DE CIRCULAÇÃO DIÁRIA UTILIZANDO A TÉCNICA DO
VIZINHO MAIS PRÓXIMO**

EMERSON DOUGLAS ROCHA

BLUMENAU
2004

2004/2-14

EMERSON DOUGLAS ROCHA

**SISTEMA DE AJUDA NA RESOLUÇÃO DE PROBLEMAS NA
ABERTURA DE ARQUIVOS DE ANÚNCIOS DE UM JORNAL
DE CIRCULAÇÃO DIÁRIA UTILIZANDO A TÉCNICA DO
VIZINHO MAIS PRÓXIMO**

Trabalho de Conclusão de Curso submetido à
Universidade Regional de Blumenau para a
obtenção dos créditos na disciplina Trabalho
de Conclusão de Curso II do curso de Ciência
da Computação — Bacharelado.

Ricardo Alencar de Azambuja – Orientador

**BLUMENAU
2004**

2004/2-14

**SISTEMA DE AJUDA NA RESOLUÇÃO DE PROBLEMAS NA
ABERTURA DE ARQUIVOS DE ANÚNCIOS DE UM JORNAL
DE CIRCULAÇÃO DIÁRIA UTILIZANDO A TÉCNICA DO
VIZINHO MAIS PRÓXIMO**

Por

EMERSON DOUGLAS ROCHA

Trabalho aprovado para obtenção dos créditos
na disciplina de Trabalho de Conclusão de
Curso II, pela banca examinadora formada
por:

Presidente: _____
Prof. Ricardo Alencar de Azambuja, Orientador, FURB

Membro: _____
Prof. Mauro Marcelo Mattos, FURB

Membro: _____
Prof. Roberto Heinzle, FURB

Blumenau, 10 de Novembro de 2004

Dedico este trabalho à minha família, minha namorada, ao meu filho e a todos aqueles que me ajudaram direta ou indiretamente na realização deste.

A imaginação é mais importante que o conhecimento.

Albert Einstein

AGRADECIMENTOS

A Deus, pelo seu imenso amor e graça.

A meus pais, que souberam compreender minha ausência.

À minha namorada, pela força e crença na minha capacidade.

Ao meu irmão, Júnior, pelo indispensável e eficaz auxílio técnico em editoração eletrônica.

Ao meu orientador, Ricardo Alencar de Azambuja, por ter acreditado na conclusão deste trabalho.

RESUMO

Este trabalho descreve um sistema de auxílio na resolução de problemas ocorridos na abertura de arquivos, enviados por meio eletrônico, para um jornal de circulação diária local utilizando a técnica de Raciocínio Baseado em Casos juntamente com a implementação do Algoritmo do Vizinho Mais Próximo para a busca do caso mais similar. Para a elaboração do sistema foi utilizada a metodologia de Análise Estruturada e, para a implementação, o ambiente de desenvolvimento Delphi 6 e o banco de dados Paradox.

Palavras-chave: Raciocínio; Similaridade; Arquivo.

ABSTRACT

This work describes a system of aid in the resolution problems occurred in the opening of archives, envoys for electronic way, for a periodical of daily circulation local using the Case-Based Reasoning together with the implementation of the Nearest Neighbor Algorithm for the search of the case most similar. For the elaboration of system it was used the Structured Analyzes methodology and, for the implementation, the environment of development Delphi 6 and the database Paradox.

Palavras-chave: Reasoning; Similarity; File.

LISTA DE ILUSTRAÇÕES

Figura 1 – Ciclo de um sistema de RBC	19
Figura 2 – O algoritmo do vizinho mais próximo	23
Figura 3 – Símbolos nos diagramas de fluxo de dados	27
Figura 4 – Exemplo de relacionamento 1:N	28
Figura 5 – Exemplo de relacionamento 1:1	29
Figura 6 – Exemplo de relacionamento N:N	29
Figura 7 – DFD particionado.....	32
Figura 8 - Tela principal do Sistema Planarte	40
Figura 9 - Cadastro de casos.....	41
Figura 10 - Consulta de casos.....	42
Figura 11 – Caso mais similar	43
Figura 12 – Adaptação de um caso.....	45
Figura 13 - Comparativo de desempenho.....	46
Quadro 1 – Exemplo de casos e busca da similaridade.....	24
Quadro 2 – Principais componentes do ambiente Delphi.....	37
Quadro 3 - Trecho de código da busca do vizinho mais próximo.....	39
Quadro 4 - Índice de similaridade	44

LISTA DE TABELAS

Tabela 1 – Modelos da inteligência artificial	14
Tabela 2 – Perfil de caso.....	15
Tabela 3 – Comparação de casos.....	43

LISTA DE SIGLAS

BMP – Bitmap

DER – Diagrama Entidade-Relacionamento

DFD – Diagrama de Fluxo de Dados

EPS – Encapsulated PostScript

FAQs – *Frequently Asked Questions*

GIF – Graphic Interchange File

IA – Inteligência Artificial

JPG ou JPEG – Joint Photographics Experts Group

PS – PostScript

RBC – Raciocínio Baseado em Casos

RNA – Redes Neurais Artificiais

TIF – Tagged Image File

WMF – Windows Meta Files

LISTA DE SÍMBOLOS

% - por cento

SUMÁRIO

1 INTRODUÇÃO	12
1.1 OBJETIVOS.....	16
1.2 ORGANIZAÇÃO DO TEXTO	16
2 RBC.....	18
2.1 ETAPAS DE UM SISTEMA DE RBC.....	19
2.2 CICLO DE UM SISTEMA DE RBC.....	19
2.3 REPRESENTAÇÃO DO CONHECIMENTO (CASOS).....	20
2.4 INDEXAÇÃO	21
2.5 RECUPERAÇÃO DOS CASOS.....	22
2.6 ADAPTAÇÃO	24
2.7 APRENDIZAGEM.....	25
2.8 TRABALHOS CORRELATOS	25
3 ANÁLISE ESTRUTURADA	26
3.1 LISTA DE EVENTOS	26
3.2 DFD	27
3.3 DER.....	27
3.3.1 DER LÓGICO	28
3.3.2 DER FÍSICO	29
4 DESENVOLVIMENTO DO SISTEMA.....	30
4.1 REQUISITOS PRINCIPAIS	30
4.2 ESPECIFICAÇÃO DO SISTEMA	30
4.2.1 LISTA DE EVENTOS	30
4.2.2 DFD PARTICIONADO	31
4.2.3 DFD DE NÍVEL.....	32
4.2.4 DER LÓGICO	33
4.2.5 DER FÍSICO	34
4.3 IMPLEMENTAÇÃO	34
4.3.1 BANCO DE DADOS	34
4.3.1.1 BANCO DE DADOS PARADOX.....	35
4.3.2 AMBIENTE DE PROGRAMAÇÃO DELPHI.....	36
4.3.3 FERRAMENTAS CASE.....	37
4.3.3.1 POWER DESIGNER.....	38

4.3.4 TÉCNICAS E FERRAMENTAS UTILIZADAS	38
4.3.5 OPERACIONALIDADE DA IMPLEMENTAÇÃO	39
4.4 RESULTADOS E DISCUSSÃO	45
5 CONCLUSÕES.....	47
5.1 EXTENSÕES	47
REFERÊNCIAS BIBLIOGRÁFICAS	49
ANEXO A - FORMATOS MAIS UTILIZADOS NA EDITORAÇÃO ELETRÔNICA E SUAS CARACTERÍSTICAS MAIS IMPORTANTES.....	51

1 INTRODUÇÃO

A área de classificados de um determinado jornal opera com a confecção de anúncios, através da mão-de-obra de profissional arte-finalista e da recepção de material (anúncios prontos, logomarcas, fotos, etc) por meio eletrônico efetuado por um coordenador de arte e, após o expediente comercial, por um plantonista. Os materiais podem ser enviados em arquivos magnéticos de vários formatos/extensões - JPEG, TXT, TIF, PS, CDR, EPS, entre outros – e cada tipo de arquivo pode apresentar problemas diversos na hora da abertura.

A recepção dos materiais é efetuada por um plantonista após o encerramento das atividades da coordenação de arte, que geralmente carece da experiência e, em determinadas situações, de suporte da coordenação. Esta situação geralmente acarreta atraso nas atividades, pois o suporte é feito por meio telefônico. Este cenário poderia ser resolvido com um treinamento adequado ao funcionário de plantão, mas a rotatividade de pessoas nesta função é muito grande. Além disso, o estudo sobre qualquer atividade ou ofício compõe apenas o conhecimento genérico do ser humano. Através da aplicação desse conhecimento ao longo do tempo, adquire-se o conhecimento específico e, por conseguinte, especializando-se em determinada atividade na resolução de problemas encontrados. Pode-se aprender tanto por experiência própria quanto em base nas experiências de outrem.

É com a perspectiva do aprendizado em experiências e do conhecimento passado que um dos modelos da IA é o de RBC, o qual será abordado neste trabalho, juntamente com a técnica do Vizinho Mais Próximo para a recuperação dos casos.

De acordo com Luger (1989), IA pode ser definida como uma parte da ciência da computação que se preocupa com a automação do comportamento inteligente. Seus princípios incluem estruturas de dados utilizadas na representação do conhecimento, os algoritmos necessários para aplicar esse conhecimento, e as linguagens e técnicas de programação usadas na sua implementação. De qualquer modo, esta definição sofre com o fato de que a própria inteligência não é bem definida ou entendida. Apesar do ser humano estar seguro de conhecer um comportamento inteligente quando se depara com isso, é duvidoso para qualquer um poder definir a inteligência de maneira suficiente para ajudar na avaliação de um programa de computador supostamente inteligente, enquanto ainda está se aprendendo a força vital e a complexidade da mente humana.

Conforme Tvetter (1998), o objetivo da IA é produzir programas de computador que trabalhem semelhante às pessoas, de tal modo que os computadores poderão ser tão ou mais proveitosos, que poderão fazer muito mais coisas que as pessoas e, talvez, melhor e mais rápido. Para isso, também é objetivo da IA descobrir como a inteligência e o pensamento humano trabalham.

Segundo Nilsson (1980), métodos e técnicas de IA têm sido aplicados em problemas de áreas diversas, como:

- a) processamento de linguagem natural: estudo voltado para a construção de programas capazes de compreender a linguagem natural (interpretação) e gerar textos. A geração de linguagem natural é a produção de textos por um programa a partir de um conteúdo semântico representado internamente no próprio programa. Objetiva aperfeiçoar a comunicação entre as pessoas e os computadores;
- b) recuperação inteligente de banco de dados: objetiva permitir a recuperação de dados relevantes em uma base de dados;
- c) sistemas de consulta inteligente: sistemas automáticos de consulta para poder diagnosticar doenças, avaliar depósitos de minério em potencial, sugerir estruturas para química orgânica complexa, e até fornecer dispositivos sobre como usar outros sistemas de computador;
- d) demonstração de teoremas: ajudar a provar ou refutar teoremas na área da matemática, através de processos dedutivos utilizando lógica de predicados;
- e) robótica: resolver os problemas de ações físicas de robôs móveis, através de técnicas de modelagem dos “estados do mundo” e descrever processos de mudança desses estados, para a interação da máquina com o meio;
- f) programação automática: estuda a geração automática de programas através da especificação;
- g) problemas de sincronização e combinação: estuda resolver problemas de sincronização entre processos e problemas de combinações possíveis (menor caminho, quebra-cabeças, jogos);
- h) problemas de percepção/visão: estudo da percepção do computador em relação ao meio. Desenvolvimento de formas de o computador trabalhar com a “visão” e “audição”;
- i) aprendizado: desenvolvimento de programas que conseguem aprender certos fatos por meio da experiência “própria” ou de outrem.

Para aplicar os métodos e técnicas da IA ao longo dos anos foram sendo criados diferentes modelos, que são demonstrados na tabela 1:

Tabela 1 – Modelos da inteligência artificial

Modelo	Característica
Algoritmos Genéticos	Objetivam emular operadores genéticos (específicos, como <i>crossing-over</i> , <i>mutação</i> e <i>reprodução</i>) da mesma forma como é observado na natureza. Isso é feito criando-se dentro da máquina uma <i>população</i> de <i>indivíduos</i> representados por <i>cromossomos</i> . Os <i>indivíduos</i> passam por um processo simulado de evolução, seleção e reprodução, gerando uma nova população.
Lógica Fuzzy	Também denominada de Conjuntos Difusos ou Lógica Nebulosa. É uma metodologia que serve para representar, manipular e modelar informações incertas.
Programação Evolutiva	Assemelha-se aos algoritmos genéticos, sendo que é dado maior ênfase na relação comportamental entre os parentes e seus descendentes. As soluções para os problemas são obtidas por meio tentativas e transmitidas para a nova população (simulada em programas).
Programação Genética	É um campo de estudo da IA voltado para a construção de programas que visam imitar o processo natural da genética. Trabalha com métodos de busca aleatória
RBC	É o campo de estudo da IA que utiliza uma grande biblioteca de casos para consulta e resolução de problemas. Os problemas atuais são resolvidos, através da recuperação e consulta de casos já solucionados e da conseqüente adaptação das soluções encontradas.
RNA	Possui várias denominações, dentre elas redes neuronais, modelo conectista, neurocomputação, modelo de processamento paralelo distribuído, sistemas neuromórficos e computadores biológicos.
Sistemas Baseados em Conhecimento	São sistemas que implementam comportamentos inteligentes de especialistas humanos.

Fonte: adaptado de Torres (1997)

O RBC “[...] é utilizado para encontrar uma solução para um problema atual olhando um problema similar em uma base de conhecimento, tomando a solução do passado e usando como ponto de partida para encontrar a solução do problema atual”.(LENZ, 1998, p. 05).

RBC é um método que se baseia em experiências anteriores. É um “raciocinador” utilizando conhecimento antigo (casos), podendo usá-los para sugerir soluções para problemas, apontar potenciais problemas com uma solução existente, interpretar uma nova situação ou fazer previsões sobre o que poderá acontecer, ou para criar argumentos comprovando alguma conclusão. (KOLODNER, 1993, p. 01).

Para a recuperação dos casos o RBC utiliza-se, entre outras, da técnica conhecida como Vizinho Mais Próximo ou Algoritmo da Vizinhança. Esse método baseia-se na comparação entre um novo caso e aqueles armazenados no banco de casos, utilizando uma soma ponderada de suas características. Para isso, de acordo com Abel (2002), é necessário atribuir um peso a cada uma das feições que descrevem o caso e que serão utilizadas na recuperação.

A coordenação de arte da área de Operações Comerciais – Opec do Jornal de Santa Catarina opera com recebimento de anúncios em mídia eletrônica – *e-mail*, *CDs*, *zip*, disquete – que muitas vezes apresentam problemas como: ausência de imagens, estouro de fontes,

baixa qualidade de resolução, etc. O coordenador de arte possui o conhecimento necessário para a resolução da maioria dos problemas, mas na falta deste, o trabalho fica a cargo de um plantonista que carece de experiência, haja vista a rotatividade nessa função. Por conseguinte, ocorrem atrasos, pois o plantonista precisa contatar o coordenador, via telefone e quando da não-disponibilidade do mesmo, pedir desnecessariamente o reenvio da mídia.

A partir deste cenário surge a proposta de desenvolvimento de um sistema de auxílio utilizando RBC, onde a experiência da coordenação de arte será armazenada em um banco de casos na forma de um perfil de caso, semelhante a um *checklist*, em que o funcionário indicará o perfil do arquivo que está sendo aberto e o sistema mostrará o caso mais similar, utilizando a técnica do Vizinho Mais Próximo. O perfil de caso é formado pelos atributos Formato, Abre, Fontes, Gráficos, Imagens, Sistema de Cor, Versão, Sangria, Cor Preta, Visualização, Válido, Imprime Fitolito e Solução. Em cada atributo pode assumir determinados valores. Por exemplo, o atributo Formato assume um nome de formato (EPS, PDF, TIFF, JPEG, PSD, etc), o atributo gráfico pode assumir que está OK, Baixa Resolução, Sem gráficos, Desaparecimento, Sem efeito, etc, e assim por diante. A tabela 2 demonstra o perfil de caso e alguns valores que os atributos podem assumir.

Tabela 2 – Perfil de caso

Atributos	Valores				
Formato	EPS	PDF	TIFF	JPEG	PSD
Abre?	Sim	Não			
Fontes	OK	Trocou	Sem fontes	Desaparecimento	Sem efeito
Gráficos	OK	Baixa resolução	Sem gráficos	Desaparecimento	Sem efeito
Imagens	OK	Baixa resolução	Sem imagens	Desaparecimento	Sem efeito
Sistema de Cor	RGB	CMYK	Pantone	Lab	Sem efeito
Versão	Compatível	Incompatível	Sem efeito		
Sangria	Sim	Não	Sem efeito		
Cor preta	OK	4 camadas	Sem efeito		
Visualização	Correta	Incorreta	Sem efeito		
Válido?	Sim	Não	Sem efeito		
Imprime Fitolito?	Sim	Não	Sem efeito		
Solução:	Pedir reenvio da mídia...	Abrir no Corel...	Fazer corte da sangria....		

O grau de similaridade será indicado em forma de porcentagem, pois nem sempre haverá um caso 100% (cem por cento) similar na base de casos. Verificando a solução proposta pelo sistema, o plantonista efetuará os passos indicados pelo mesmo. Após aplicar a solução indicada, se houver êxito na resolução do problema, o usuário irá reter o caso pesquisado como um novo caso na memória de casos, excetuando-se os casos em que a

similaridade encontrada for total, o que já indica a existência do caso na base. Não havendo sucesso o plantonista irá ignorar a indicação do sistema e relatar o caso ao coordenador, que irá tomar as providências que achar necessárias como, por exemplo, encontrar a solução para o problema não resolvido e registrar na memória de casos. Também poderá ser feita a adaptação de um ou mais atributos com base no perfil indicado, onde o sistema irá procurar o caso que seja exatamente igual, com exceção dos atributos escolhidos para adaptar, através de novos valores indicados pelo usuário e, se obtiver sucesso, armazenará automaticamente como um novo caso.

1.1 OBJETIVOS

O objetivo do trabalho é o desenvolvimento de um sistema de auxílio à resolução de problemas na abertura de arquivos magnéticos – de diversos formatos, como JPG, CDR, EPS, PDF, PS, entre outros – recebidos por meio eletrônico, na área de classificados de um jornal de circulação diária.

Os objetivos específicos do trabalho são:

- a) carga da memória de casos através de consulta com um especialista;
- b) controlar os problemas encontrados na abertura de arquivos mediante solução proposta pelo sistema;
- c) controlar as providências (soluções) a serem aplicadas com base nos problemas encontrados na abertura de arquivos, através da inclusão, ou não, de um novo caso na memória de casos ou de sua adaptação.

1.2 ORGANIZAÇÃO DO TEXTO

O presente trabalho está disposto em 5 (oito) capítulos descritos a seguir:

O primeiro capítulo apresenta a introdução e os objetivos pretendidos com a elaboração do trabalho.

O segundo capítulo explana a técnica de RBC, o algoritmo do Vizinho Mais Próximo e trabalhos correlatos.

O terceiro capítulo trata dos conceitos e da metodologia de análise estruturada para a construção de sistemas.

O quarto capítulo demonstra o desenvolvimento do sistema no que tange aos requisitos, especificação e implementação do sistema proposto, bem como as tecnologias e ferramentas utilizadas, sua operacionalidade, resultados e discussões.

O quinto capítulo mostra as conclusões e sugestões para trabalhos futuros.

2 RBC

Segundo Abel (2004), o conceito de RBC é que, para um domínio particular, os problemas a serem resolvidos tendem a ser recorrentes e repetir-se com pequenas alterações em relação a sua versão original. Dessa forma, soluções anteriores podem ser reaplicadas também com pequenas modificações.

A síntese do RBC é a utilização de experiências passadas – tecnicamente chamadas de casos – utilizadas como parâmetros para resolver problemas atuais.

De acordo com Tvetter (1998), na IA tradicional a suposição sempre foi de que as pessoas agiam de acordo com o conhecimento sobre o mundo, isto é, uma coleção de fatos que descrevem o mundo. Há algum tempo alternativas têm sido propostas: pessoas armazenam experiências e agem comparando suas experiências passadas com os problemas atuais. Esta idéia de sobre como as pessoas agem resultou em métodos baseados em casos para resolver problemas.

Uma base de conhecimento para Raciocínio Baseado em Casos, mais do que regras gerais são, um conjunto de exemplos relevantes. Estes casos são adaptados para novos problemas de um processo de raciocínio analógico. Eles são uma outra resposta para dificuldades encontradas na difícil manipulação de uma base de conhecimento de regras gerais, que desejam cobrir todas as situações. Proponentes de RBC argumentam que isto é muito próximo do raciocínio humano. (LUGER, 1989, p. 569)

Conforme Heinzen (2002), um sistema de RBC tem três estruturas básicas. A primeira delas é a base de casos, onde um caso representa a experiência ou a interpretação de uma experiência, de um sistema de RBC. A segunda estrutura é representada por um mecanismo de pesquisa, um algoritmo que tem por finalidade localizar na base de casos o caso mais parecido com o caso atual. Em terceiro deve ter, ainda, a descrição de cada caso, juntamente com os índices que permitam que este caso seja acessado.

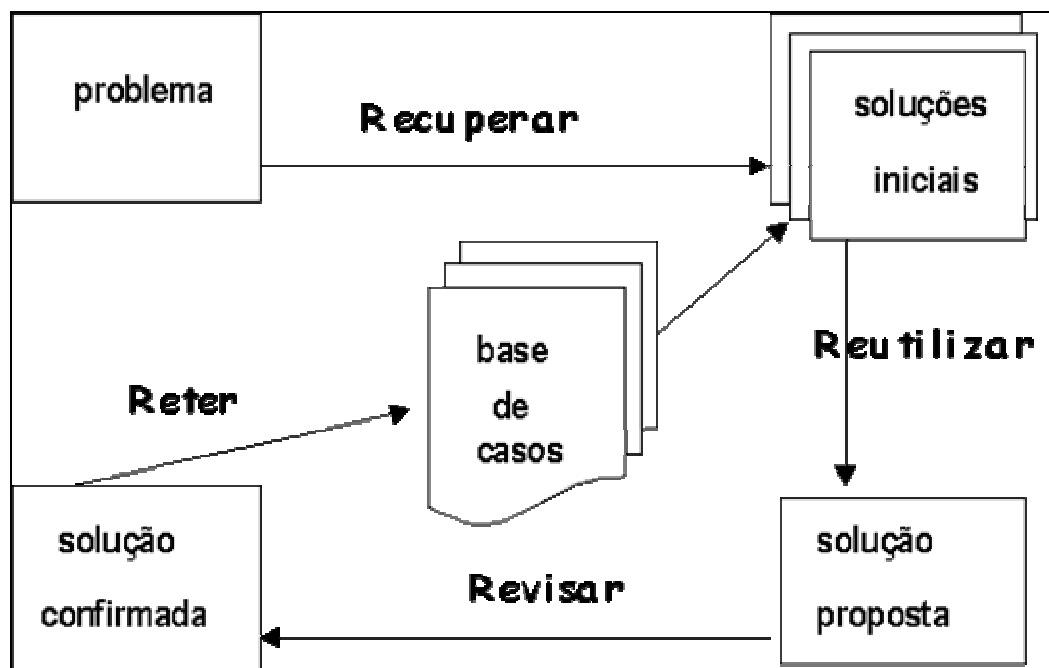
2.1 ETAPAS DE UM SISTEMA DE RBC

As etapas mais importantes que envolvem um sistema de RBC que devem ser observadas são as seguintes:

- representação do conhecimento em RBC (casos);
- indexação dos casos;
- recuperação dos casos;
- adaptação;
- aprendizagem.

2.2 CICLO DE UM SISTEMA DE RBC

Um sistema de RBC funciona por comparar a descrição de um problema a ser resolvido com os casos descritos em uma base de casos. De acordo com Abel (2004), o sistema recupera então o caso mais similar, avalia a necessidade de adaptar a solução associada e aplica essa solução ao novo problema, como pode ser visto na figura 1:



Fonte: Camargo (1999)

Figura 1 – Ciclo de um sistema de RBC

Em resumo, o ciclo de um sistema de RBC, visto na figura 1, irá funcionar da seguinte forma:

- coletar de informações: o usuário irá preencher uma tabela de informações sobre o problema;

- recuperar o caso mais similar: retorna o caso mais similar ao caso de entrada, comparando cada caso com o novo caso, utilizando-se o Algoritmo do Vizinho Mais Próximo, que mede a diferença de cada um dos atributos de ambos os casos;
- transformar a solução do caso recuperado para satisfazer restrições do problema atual;
- escolher o caso que obtiver a menor soma das diferenças;
- avaliar a solução associada ao caso para ver se é apropriada;
- adaptar a solução para compensar as diferenças entre os casos, se necessário;
- reter o novo caso na base de casos com a solução adotada.

2.3 REPRESENTAÇÃO DO CONHECIMENTO (CASOS)

Um caso nada mais é do que a representação de um conhecimento anterior, na forma explícita, de como uma situação foi solucionada. Os casos devem ser representados, então, como exemplares de acontecimentos e indexados de uma forma que facilite sua recuperação. Os índices de um caso são tão importantes como os índices de livros em uma biblioteca. A indexação é que determina quais atributos que devem ser comparados para se avaliar a similaridade entre o caso de entrada e os casos da base.

De acordo com Castoldi e Santos (2002), quando montado um sistema, cada diferente solução ou interpretação do problema, é um novo caso. Os casos são obtidos, em geral, consultando um especialista humano. Um profissional experiente, com certeza tem muito a ensinar, se inserirmos toda sua experiência em um sistema e recuperarmos os casos de maneira eficiente (ou seja, os casos recuperados têm que ser os que melhor respondem às necessidades), teremos um sistema inteligente. A representação de casos envolve a descrição dos elementos e das propriedades relevantes da experiência modelada, a organização e o método de acesso destes em memória. No presente trabalho a indexação é feita de forma manual, na qual o usuário indica os atributos que serão comparados.

Segundo Ramos (2000) *apud* Bergmann, existem três abordagens principais para a representação de casos que diferem segundo a origem, a forma e o conhecimento empregado:

- a) abordagem baseada em textos;
- b) abordagem baseada em conversação;
- c) abordagem estruturada.

Na abordagem baseada em textos, os casos são representados em forma de texto livre. É muito usada em ambientes onde já existe uma grande quantidade de documentos, normalmente chamados de *how to do* - como fazer e/ou listas de discussão FAQs. Estes documentos permitem o usuário rapidamente utilizar o conhecimento descrito. Esta abordagem é indicada para domínios que não dispõem de muitos casos (menos do que uma centena). Em contrapartida, recuperam um grande número de casos que são irrelevantes.

A abordagem baseada em conversação é voltada para domínios onde uma grande quantidade de problemas simples deve ser resolvida constantemente (ex. suporte a usuários do *windows*, a usuários da internet, etc.). A base de casos é organizada normalmente pelo autor do caso, o que torna esta atividade complexa e de alto custo. É voltada para aplicações onde poucas questões são necessárias para a tomada de decisão. O custo de manutenção é alto.

A abordagem estrutural, que foi utilizada no presente trabalho, descreve casos através de atributos e valores pré-definidos. Os atributos podem ser estruturados em tabelas relacionais ou segundo o paradigma da orientação a objetos. O modelo do domínio define um padrão para representar casos, especificando o conjunto de atributos usados. Em geral, um caso pode ser representado a partir de alguma estrutura de dados (par: atributo, valor) para descrever um problema e uma solução.

2.4 INDEXAÇÃO

Conforme Kolodner (1993), a indexação é vista como um problema de escolher características que servirão de índices para os casos colocados na memória, de forma que eles possam ser recuperados quando apropriados, ou seja, selecionar características da descrição dos casos que permitam recuperação eficiente dos casos relevantes ao problema. Estas características identificam lições que o caso tem a ensinar. A resposta para a questão: "*o que faz um caso similar a outro?*" é o conjunto de índices que representam características relevantes dos casos. Mas, indexação pode também ser vista como um problema de organizar a memória de casos, de modo que a recuperação seja feita de forma eficiente e precisa.

Indexar casos depende da compreensão do conteúdo e finalidade da informação que eles armazenam. Um bom índice permite reconhecer similaridades úteis entre os casos recuperados e essa utilidade só pode ser percebida se os índices forem escolhidos com base em uma boa compreensão do problema.

Para Abel (2002), As qualidades necessárias a um bom índice podem ser descritas como:

- prever a futura utilização da informação para solução de diferentes problemas;
- endereçar as similaridades úteis entre os casos;
- ser abstrato o suficiente para tornar um caso útil em uma variedade de diferentes situações;
- ser concreto o suficiente para ser facilmente reconhecido em futuras situações.

Segundo Heinzen (2002), dependendo de qual característica (ou contexto) o índice reflete, diferentes formas de atribuição de peso podem ser consideradas, entre elas: números inteiros, números reais, conjuntos, cadeia de caracteres, caracteres, etc.

2.5 RECUPERAÇÃO DOS CASOS

No RBC a recuperação é executada por meio de um algoritmo que compara o caso de entrada com os casos existentes na base de casos, e retorna o caso (ou casos) mais semelhante ao atual.

Para Kolodner (1993), recuperar é o processo de retornar um ou mais casos da base de casos em resultado à comparação de um novo caso com cada um dos casos da base. Essa comparação é feita por meio de uma avaliação de similaridade. O resultado dessa comparação é a seleção de um caso ou uma combinação de casos que sugere uma solução para o novo caso.

Nesta etapa, deve ser escolhido um método de recuperação, que irá recuperar o caso mais similar ou um conjunto de casos da memória de casos. Para o presente trabalho foi escolhido o Algoritmo de Vizinhança, mais conhecido como Vizinho Mais Próximo (*Nearest, Neighbour*). Segundo Abel (2004), esse método baseia-se na comparação entre um novo caso e aqueles armazenados no banco de utilizando uma soma ponderada das suas características. Para isso, é necessário atribuir um peso a cada uma das feições que descrevem o caso e que serão utilizadas na recuperação. A figura 2 mostra a fórmula empregada no algoritmo do Vizinho Mais Próximo:

$$\frac{\sum_{i=1}^n w_i \times \text{sim}(f_i^I, f_i^R)}{\sum_{i=1}^n w_i}$$

Fonte: Abel (2004, p. 15)

Figura 2 – O algoritmo do vizinho mais próximo

Entendendo o algoritmo mostrado na figura 1:

- a) *sim* é a função de similaridade;
- b) f_i^I e f_i^R são os valores da feição *i* para o novo caso;
- c) *n* é o número de atributos;
- d) *i* é um atributo individual;
- d) *w* corresponde ao peso de uma feição *i* qualquer que descreve o caso.

A maioria das ferramentas RBC utiliza algoritmos, onde o resultado deve estar entre zero (0) e um (1), onde zero é totalmente dissimilar e um é exatamente similar. Observa-se no quadro 1 um exemplo de casos e a busca da similaridade:

Casos \ Atributos	A	B	AB	O
Alimentação 1	Café	Carneiro	Couve	Presunto
Alimentação 2	Laranja	Beterraba	Bolo Inglês	Bacon
Alimentação 3	Beterraba	Laranja	Azeite de Oliva	Bacalhau
Alimentação 4	Batata-doce	Amendoim	Laranja	Camarão

Atributos \ Casos	Alimentação 1	Alimentação 2	Alimentação 3	Alimentação 4
Novo Caso	Café	Amendoim	Couve	Batata-Doce

Atributos \ Casos	Alimentação 1	Alimentação 2	Alimentação 3	Alimentação 4
Caso Novo : A	1	0	0	1
Caso Novo : B	0	1	0	0
Caso Novo : AB	0	0	1	0
Caso Novo : O	0	0	0	0

Considerando todos os atributos com o mesmo peso, a comparação entre os casos será:

Caso Novo, A = $\frac{1+1}{4} = 0,5$

Caso Novo, B = $\frac{1}{4} = 0,25$

Caso Novo, AB = $\frac{1}{4} = 0,25$

Caso Novo, O = $\frac{0}{4} = 0,25$

O caso A é o mais semelhante, pois é o que mais se aproxima de 1.

Fonte: adaptado de Minella (2002, p. 11)

Quadro 1 – Exemplo de casos e busca da similaridade

2.6 ADAPTAÇÃO

A etapa de adaptação serve como ajuste dos valores e atributos – no todo ou em parte – dos casos, quando for possível, haja vista que às vezes o caso solucionado não é exatamente o ideal para resolver o problema do usuário, pois conforme Heinzen (2002) os casos podem não ser suficientemente semelhantes ou diferem em um ponto crucial do caso de entrada. A adaptação pode ser uma simples substituição de um atributo da solução por outro ou uma complexa modificação na estrutura da solução, através da inclusão, eliminação ou substituição de um comportamento da solução recuperada.

A adaptação deve suportar as diferenças entre todos os futuros casos e os casos conhecidos, disponíveis na base, e determinar as adaptações requeridas por estas diferenças. Isto requer um profundo conhecimento do domínio da aplicação, o que muitas vezes inviabiliza o seu uso. De acordo com Ramos (2000) *apud* Watson, a menos que seja usada em um domínio bem conhecido, a adaptação deve ser evitada, pois pode se tornar o “calcanhar de Aquiles” de um sistema de RBC.

2.7 APRENDIZAGEM

A aprendizagem de um sistema de RBC é a capacidade de armazenamento de novas soluções ou interpretações geradas. É obtida pelo incremento na quantidade e diversidade de casos à memória de casos.

A solução de um problema de entrada que tenha respondido às expectativas do usuário em todo ou em parte, desde que satisfatória, é adicionada na base de casos e consiste na aprendizagem do sistema.

A inclusão prévia de casos, ou seja, a aquisição de conhecimento, através da consulta com um especialista na área em que será utilizado o sistema de RBC, pode fortalecer ainda mais a aprendizagem.

2.8 TRABALHOS CORRELATOS

Alguns trabalhos envolvendo o tema de RBC podem ser encontrados. Entre os que utilizaram o algoritmo do Vizinho Mais Próximo, podem ser citados:

- a) Em Heinrich (2001), foi desenvolvido um sistema de apoio ao técnico de eletrônica com o intuito de facilitar o serviço prestado pelo mesmo, oferecendo serviços com qualidades apreciadas pelo cliente. O sistema permite que o técnico consiga solucionar os problemas dos aparelhos eletrônicos tomando como base as soluções oferecidas através do uso do RBC, utilizando a técnica da similaridade do Vizinho mais Próximo;
- b) Em Heinzen (2002), o trabalho descreve uma ferramenta de auxílio à lógica de programação. O sistema recebe o enunciado de um problema de lógica de programação e recupera o perfil do caso que mais se assemelhe ao caso que o aluno tenta resolver no momento;
- c) Em Kienen (2003), a tecnologia de RBC foi utilizada no desenvolvimento de um sistema de suporte à tomada de decisões em processos jurídicos. A finalidade do sistema é encontrar os processos semelhantes e classificá-los de acordo com o índice de similaridade existente entre os mesmos, utilizando o algoritmo do Vizinho Mais Próximo.

3 ANÁLISE ESTRUTURADA

Segundo Gane e Sarson (1983), a análise estruturada de sistemas compõe-se de um conjunto de técnicas e ferramentas que nasceram da programação e do projeto estruturados. Seu conceito fundamental é a construção de um modelo lógico de um sistema, utilizando técnicas gráficas capazes de levar usuários, analistas e projetistas a formarem um quadro claro e geral do sistema e de como suas partes se encaixam para atender suas necessidades. As técnicas estruturadas permitem ao analista de sistemas desenvolver uma perspectiva dos requisitos computacionais que seja compatível com os procedimentos existentes para os usuários. Devido à especificação ser um modelo gráfico conciso e de fácil compreensão, o usuário pode familiarizar-se bem com o sistema antes de sua implementação.

Outras vantagens oferecidas pela análise estruturada, conforme Bellin e Suchman (1993), dizem respeito à facilidade das organizações mudarem à medida que os recursos computacionais evoluem, e isto é conseguido de várias maneiras:

- a) utilizando modelos gráficos em vez de especificações por escrito;
- b) usando conjuntos nivelados de diagramas ao invés de conjuntos e subconjuntos hierárquicos durante a fase de análise;
- c) traduzindo diagramas de funções e de fluxo de dados em diagramas de estrutura de módulos de programas;
- d) através de diagramas entidade-relacionamento e outros modelos de dados.

Utiliza-se na análise estruturada, ainda, os seguintes documentos: lista de eventos, diagrama de contexto, dicionário de dados, DFD, DER, entre outros, não sendo obrigatoriamente necessária à utilização de todos. No presente trabalho optou-se pelo uso da lista de eventos, DFD (total e particionado), DER (lógico e físico).

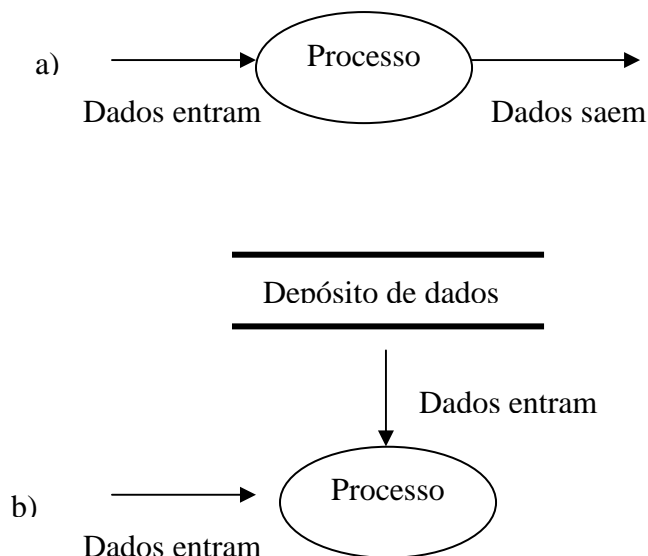
3.1 LISTA DE EVENTOS

Os eventos, segundo Bellin e Suchman (1993), são situações no ambiente do usuário que resultam na geração de um fluxo de dados de entrada pelo usuário ou que requerem que o usuário receba dados do sistema. É uma lista narrativa dos estímulos que ocorrem no mundo exterior, os quais o sistema deve responder. A seguir encontra-se uma lista de eventos para matrícula em cursos de uma escola técnica:

1. Funcionário cadastra cursos;
2. Funcionário cadastra disciplinas;
3. Aluno efetua matrícula;
4. Departamento financeiro emite faturas de mensalidades.

3.2 DFD

Os DFDs são utilizados para exibir os requisitos do fluxo de dados. Segundo Pfleeger (2004), o diagrama mostra os dados que fluem para dentro do sistema, como eles são transformados e como eles saem do sistema. Os fluxos são representados por uma simbologia apropriada. A entrada é representada por uma seta entrando em uma bolha e a saída por uma seta saindo da mesma, como mostra a figura 3(a). Algumas vezes os dados estão em um repositório formal ou um banco de dados – também chamado de depósito de dados – que são representados por duas barras paralelas, como mostra a figura 3(b).



Fonte: adaptado de Pfleeger (2004).

Figura 3 – Símbolos nos diagramas de fluxo de dados

3.3 DER

Num sistema, os diagramas de entidade-relacionamento são valiosos por mostrar uma estrutura geral dos dados representados. O DER é um modelo que descreve a diagramação dos dados armazenados de um sistema em alto nível de abstração e tem representação lógica e física.

Em Gane (1988), entidades são descritas como sendo objetos, que duram através do tempo, ou eventos que ocorrem num momento específico, e deve-se ter em mente que cada entidade termina representada pelo menos por uma tabela (armazém de dados num DFD). Os relacionamentos são as associações entre as entidades.

3.3.1 DER LÓGICO

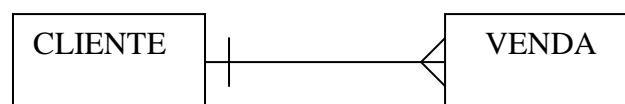
Conforme Gane (1988), a modelagem lógica de entidade-relacionamento serve para informar a respeito da estrutura dos dados a serem armazenados no sistema, ou seja, quais entidades são de interesse para armazenamento de dados e o relacionamento entre elas. Por relacionamento entende-se qual a associação que existe entre as entidades.

Para expressar os relacionamentos precisamos informar a cardinalidade, que é o comportamento (número de ocorrências) que existem entre as entidades, que podem ser:

- a) N:N – muitos para muitos – onde n pode ser 0 (zero);
- b) N:1 – muitos para um;
- c) 1:1 – um para um;
- d) 1:N – um para muitos.

Alguns exemplos de notação de relacionamento e cardinalidade entre entidades, utilizada em Gane (1988), são vistos a seguir.

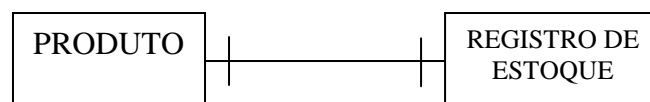
A figura 4 mostra um exemplo de relacionamento “um” (representado através de uma barra simples na vertical) para “muitos” (demonstrado por uma linha com símbolo de tridente), onde uma entidade cliente pode associar-se a muitas vendas, mas cada venda associada a apenas um cliente.



Fonte: Gane (1988, p. 9)

Figura 4 – Exemplo de relacionamento 1:N

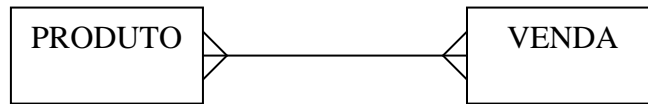
Na figura 5 é demonstrado um relacionamento 1:1:



Fonte: Gane (1988, p. 9)

Figura 5 – Exemplo de relacionamento 1:1

Um exemplo de relacionamento N:N é mostrado na figura 6:



Fonte: Gane (1988, p. 9)

Figura 6 – Exemplo de relacionamento N:N

3.3.2 DER FÍSICO

Um modelo físico é a tradução do modelo lógico para a linguagem do software escolhido para implementar o sistema e define as estruturas que serão utilizadas para armazenar os dados. Representa detalhes de armazenamento interno de informações (representação computacional), onde são realmente definidos os atributos de cada entidade e o banco de dados se concretiza.

4 DESENVOLVIMENTO DO SISTEMA

Neste capítulo é apresentados o projeto desenvolvido neste trabalho, os requisitos, as tecnologias e ferramentas utilizadas na especificação e implementação, bem como a operacionalidade do sistema, resultados e discussões.

4.1 REQUISITOS PRINCIPAIS

A essência do sistema é a base de casos, para que haja um parâmetro na resolução dos problemas e na busca da similaridade. Através de consulta com um especialista na área de editoração eletrônica é feita a carga dos casos lembrados pelo mesmo. A partir daí o usuário poderá consultar um caso e receber a solução mais similar que o sistema buscará através do algoritmo do Vizinho Mais Próximo. O caso mais similar é mostrado, em forma de porcentagem e então colocado em prática. Se a solução proposta for satisfatória poderá ser incluída como um novo caso na base de casos ou feita sua adaptação, mas se porventura não houver sucesso, o usuário informará o problema para o especialista, que indicará as providências a serem tomadas e, mais tarde, poderá carregar o novo caso na base.

4.2 ESPECIFICAÇÃO DO SISTEMA

No desenvolvimento do sistema foi utilizada a metodologia da engenharia de software de análise estruturada através de lista de eventos, diagrama de entidade-relacionamento (lógico e físico), diagrama de fluxo de dados (completo e particionado), utilizando para os diagramas a ferramenta *case Power Designer*.

4.2.1 LISTA DE EVENTOS

Na lista de eventos enumerada a seguir, são definidos os acontecimentos a partir dos quais o sistema é criado:

- 1) Coordenador cadastra caso;
- 2) Coordenador mantém memória de casos;
- 3) Plantonista consulta caso;
- 4) Resposta é emitida;
- 5) Plantonista retém solução;
- 6) Plantonista adapta caso.

No primeiro evento é efetuado o cadastramento de casos, ou seja, a carga da memória de casos que será utilizada nos eventos posteriores.

O segundo evento é a manutenção dos casos pelo coordenador (especialista) que irá consultar, alterar, editar e deletar casos.

O terceiro evento é a consulta de um caso pelo plantonista, informando qual o perfil caso com base no formato de arquivo que está sendo aberto, onde o sistema compara o perfil indicado com cada um armazenado na memória de casos.

No quarto evento é emitida a resposta do sistema através da busca do caso mais similar ao perfil indicado, utilizando o algoritmo do Vizinho Mais Próximo.

No quinto evento, se o caso encontrado não for 100% (cem por cento) similar, o plantonista poderá retê-lo na base de casos.

No sexto evento o plantonista poderá adaptar um caso.

4.2.2 DFD PARTICIONADO

O DFD particionado, mostrando os fluxos e as transformações dos dados, é apresentado na figura 7.

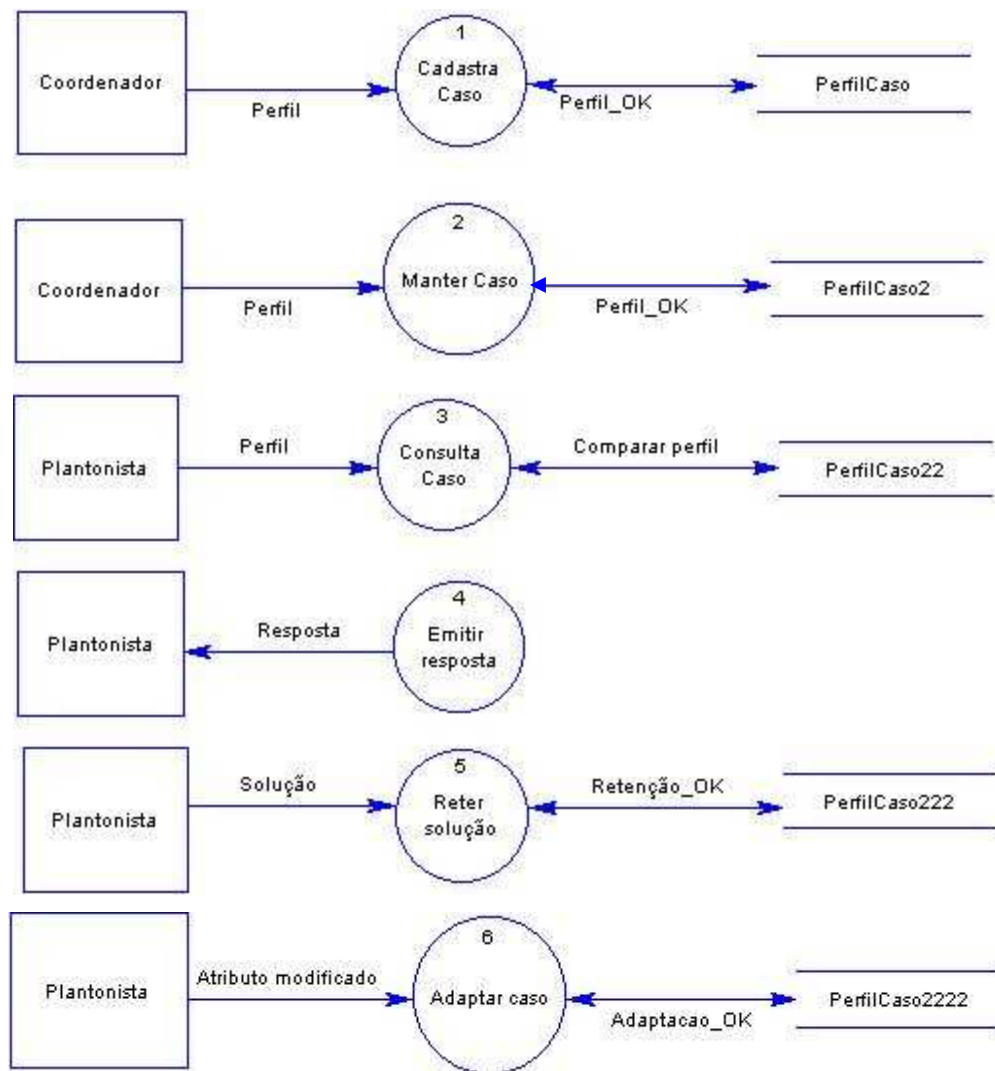


Figura 7 – DFD particionado

4.2.3 DFD DE NÍVEL

O DFD completo do sistema pode ser visualizado na figura 8:

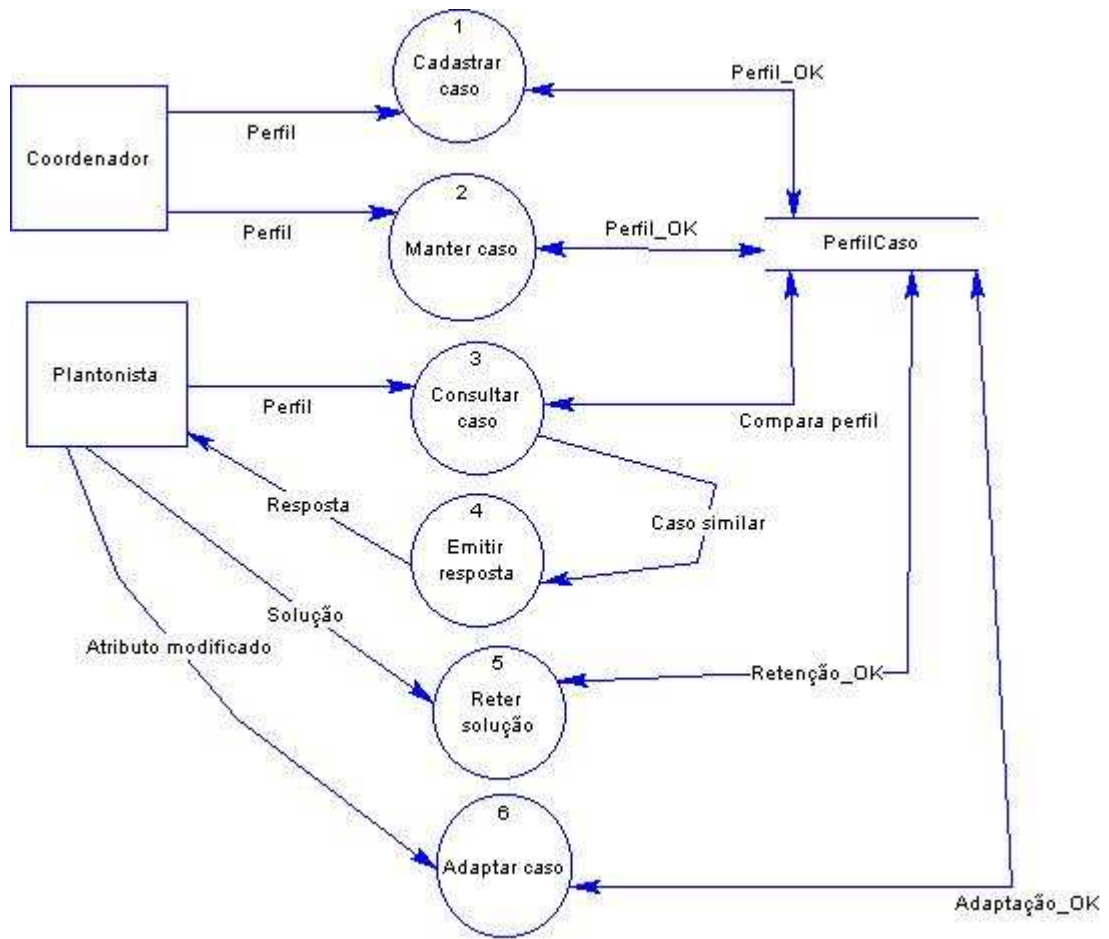


Figura 8 – DFD completo

4.2.4 DER LÓGICO

A estrutura da entidade que representa o perfil de caso é mostrada através do DER lógico na figura 9:

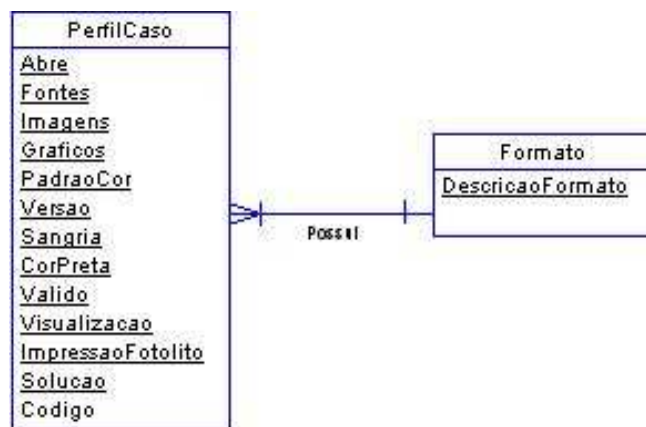


Figura 9 – DER lógico

4.2.5 DER FÍSICO

A tradução do modelo lógico para o modelo físico, utilizando modelo de banco de dados Paradox é visto na figura 10:

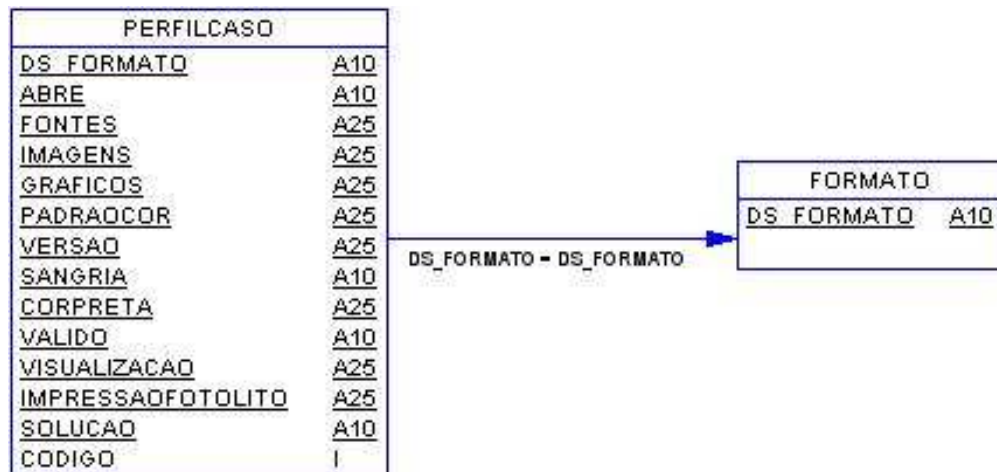


Figura 10 – DER físico

4.3 IMPLEMENTAÇÃO

Para implementar o sistema foi utilizado o ambiente de desenvolvimento Delphi com banco de dados Paradox, e na busca da similaridade foi utilizado o algoritmo do Vizinho Mais Próximo visto no capítulo 4.

4.3.1 BANCO DE DADOS

Conceitualmente, um banco de dados é uma coleção de informações inter-relacionadas representando um determinado tópico ou domínio específico.

Segundo Date (1991), um sistema de banco de dados nada mais é do que um sistema de manutenção de registros por computador. O próprio banco de dados pode ser considerado uma espécie de sala de arquivo eletrônica – ou seja, um depósito de um conjunto de arquivos de dados computadorizados que oferece diversos recursos ao usuário, possibilitando-lhe a realização de várias operações, incluindo, entre outras, as seguintes:

- a) adição de novos arquivos;
- b) inserção de novos dados nos arquivos existentes;
- c) recuperação de dados dos arquivos existentes;
- d) atualização de dados nos arquivos existentes;
- e) eliminação de dados nos arquivos existentes;

f) renovação permanente de arquivos existentes.

4.3.1.1 BANCO DE DADOS PARADOX

O *Paradox* é um sistema de banco de dados baseado em arquivos (*desktop*), ou seja, a interface com o usuário e o mecanismo de acesso ficam no mesmo computador (a estação) e apenas os arquivos de dados ficam num servidor de rede. Operações de consulta ou pesquisa devem passar através da rede. Os arquivos de dados contêm registros de dados que possuem uma ordem fixa. De acordo com Oliveira (2000), no *Paradox*, são distribuídas definições de banco de dados em vários arquivos, como índices, validação e Blobs (Binary Large Objects – objetos binários grandes), que são todos unidos em uma tabela particular. No *Paradox* existe o conceito de um ou mais diretórios de funcionamento onde as tabelas *Paradox* residem.

Conceitualmente, também existem regras empresariais (ou de negócios) como integridade referencial, chaves primárias, campos exigidos e valores de campo de falta e um mecanismo de travamento para garantir a estabilidade dos dados, pois quando dois usuários tentam acessar a mesma faixa de dados podem surgir problemas ameaçando a integridade do banco de dados. Segundo Lopes (1996), como o *Paradox* não possui nenhum conhecimento do que outro processo *Paradox* está fazendo, ele utiliza um esquema pessimista de travamento (*pessimistic locking scheme*). Assim que um usuário tenta mudar um registro o registro é travado. Nenhum outro usuário pode fazer alterações neste registro até que o primeiro usuário termine as alterações ou cancele a operação. Por um lado, este esquema de travamento é positivo se considerarmos que o usuário que obteve o travamento poderá concluir com sucesso a sua a sua operação de alteração. Por outro lado, isto é negativo no sentido em que um usuário poderá monopolizar um registro indefinidamente.

Para Oliveira (2000), as operações de navegação em tabelas *Paradox* são muito velozes, haja vista que o índice aponta para um endereço físico onde se encontra o registro. O cursor move quase instantaneamente para a posição determinada, pois conhece a localização física do registro na tabela.

Conforme Lopes (1996), para a implementação de um sistema mono-usuário, certamente o *Paradox* é a melhor escolha.

4.3.2 AMBIENTE DE PROGRAMAÇÃO DELPHI

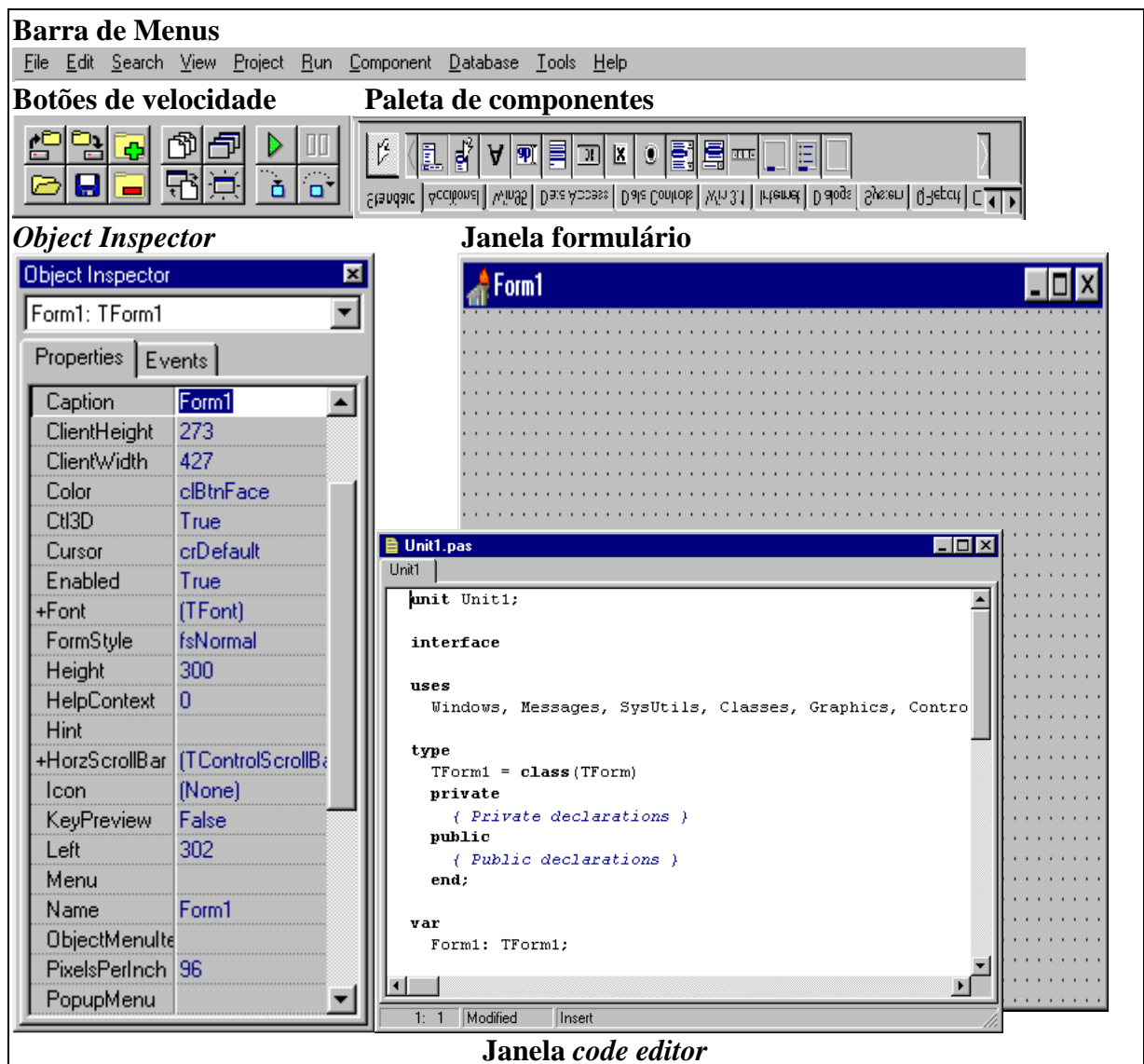
O Delphi, segundo Cantú (1993), é um ambiente rico e com facilidade de uso, com tratamento baseado em formulários e orientado a objetos que utiliza como base a linguagem *Object Pascal*.

Segundo Kienen (2003), o Delphi pode usar ou criar novas bibliotecas de funções, chamadas de DLLs (*dynamic link libraries*), que são bibliotecas de ligação dinâmicas, e seus programas podem responder a elas e iniciar qualquer evento do Windows. O Delphi realiza uma tarefa completa de compilação dos programas, indo direto para o nível de código de máquina, ficando muito mais veloz. Conforme Damasceno (1995), o Delphi é voltado para o trabalho com o conceito de projeto, que seria um conjunto de programas. O próprio Delphi escreve parte dos programas, ou seja, as aplicações são desenvolvidas com o auxílio do ambiente de programação.

De acordo com Swan (1996), há diversos elementos principais no ambiente de desenvolvimento do Delphi, que podem ser vistos a seguir:

- a) botões de velocidade: botões de apontar e clicar de alguns comandos de menu selecionados;
- b) barra de menus: estilo de menu padrão do Windows;
- c) paleta de componentes: contém ícones que representam componentes da VCL (*visual component library*), que é a biblioteca de componentes visuais do Delphi;
- d) formulário: representação visual da janela principal do programa – pode representar outras janelas como uma caixa de diálogo ou uma janela-filha;
- e) *object inspector* (inspetor de objetos): exibe todas as propriedades e eventos de um ou mais componentes ou formulários selecionados;
- f) janela *unit* (janela de unidade): exibe o trecho de programação em Pascal associado a cada formulário do aplicativo;
- g) guias de página *properties* (propriedades) e *events* (eventos): define os atributos dos componentes e as ações;

No quadro 2 podem ser visualizados os principais componentes do ambiente Delphi:



Fonte: adaptado de Ortega (1999)

Quadro 2 – Principais componentes do ambiente Delphi

4.3.3 FERRAMENTAS CASE

Segundo Garcia (2004), ferramentas *Case* (*Computer-Aided Software Engineering* – Engenharia de Software Auxiliada por Computador) são um conjunto de ferramentas desenvolvidas para auxiliar no processo de desenvolvimento de software. Seu uso existe nas empresas através de padrões e procedimentos, tais como folha de pagamento, estudos de custos de desenvolvimento, análise, programação, implementação e testes. As soluções existem, mas requerem novos softwares, novos procedimentos e novas estruturas gerenciais, muitas vezes entrando em choque com a própria cultura da organização.

Muitas ferramentas *Case* incluem poderosos editores gráficos que implementam uma interface com o usuário transformando o desenvolvimento de software num processo visual. Conforme Garcia (2004), com *Case*, o projetista do sistema pode focalizar mais o trabalho da análise e projeto e menos a codificação e teste. Especificações do projeto tomam a forma de diagramas estruturados, como Diagrama de Fluxo de Dados, Diagrama Entidade-Relacionamento e Diagrama Hierárquico de Funções. Toda a informação do sistema é armazenada num único repositório de dados onde ele pode ser automaticamente gerenciado e distribuído.

4.3.3.1 POWER DESIGNER

Segundo Sybase (2004), a ferramenta *Case Power Designer* oferece capacidades de modelagem de dados físicos, inclusive projeto de banco de dados, geração, manutenção, engenharia reversa e documentação para arquitetos de banco de dados e DBAs (*Database Administrator* – Administrador de Banco de Dados). Permite a projetistas de banco de dados criar estruturas de dados flexíveis, eficientes e eficazes para utilização por uma ferramenta de banco de dados de aplicativos. Oferece projeto de modelo de dados, conceituais geração automática de modelo de dados físicos, projeto físico desnormalizado, sistema de gerenciamento de múltiplos bancos de dados (DBMS – *Database Management System*), suporte para ferramentas de desenvolvimento e características de geração de relatórios de classe de apresentação. Um tutorial do Power Designer pode ser visto em Souza (2004).

4.3.4 TÉCNICAS E FERRAMENTAS UTILIZADAS

A utilização da metodologia de análise estruturada, com o auxílio da ferramenta *case Power Designer*, foi satisfatória para a especificação do sistema.

O ambiente de desenvolvimento Delphi e a utilização do banco de dados Paradox não apresentaram problemas na implementação do sistema como um todo nem na implementação da técnica do Vizinho Mais Próximo.

No quadro 3 pode ser visto o trecho de código responsável pela busca do vizinho mais próximo.

```

Dm.TbPerfilCaso.First;
{percorre a tabela de Perfil de Caso e soma os atributos coincidentes}
while not Dm.TbPerfilCaso.Eof do
begin
  if cbFormato.Text = Dm.TbPerfilCasoFormato.Text then
    inc(indice);
  if cbAbre.Text = Dm.TbPerfilCasoAbre.Text then
    inc(indice);
  if cbFontes.Text = Dm.TbPerfilCasoFontes.Text then
    inc(indice);
  if cbGraficos.Text = Dm.TbPerfilCasoGraficos.Text then
    inc(indice);
  if cbImagens.Text = Dm.TbPerfilCasoImagens.Text then
    inc(indice);
  if cbPadraoCor.Text = Dm.TbPerfilCasoPadraoCor.Text then
    inc(indice);
  if cbVersao.Text = Dm.TbPerfilCasoVersao.Text then
    inc(indice);
  if cbSangria.Text = Dm.TbPerfilCasoSangria.Text then
    inc(indice);
  if cbCorPreta.Text = Dm.TbPerfilCasoCorPreta.Text then
    inc(indice);
  if cbVisualizacao.Text = Dm.TbPerfilCasoVisualizacao.Text then
    inc(indice);
  if cbValido.Text = Dm.TbPerfilCasoValido.Text then
    inc(indice);
  if cbImpFotolito.Text = Dm.TbPerfilCasoImpressaoFotolito.Text then
    inc(indice);

  indice := round(indice/12*100); {calcula o grau de similaridade}
  {se o caso é 100% similar quer dizer que já existe na memória
  de casos e não é necessária sua retenção}

```

Quadro 3 - Trecho de código da busca do vizinho mais próximo

4.3.5 OPERACIONALIDADE DA IMPLEMENTAÇÃO

O nome adotado para a aplicação foi Sistema Planarte. A tela principal do sistema – que compõe o acesso à memória de casos e consulta de casos – pode ser conferida na figura 8.



Figura 8 - Tela principal do Sistema Planarte

O botão “memória” chama o formulário de cadastro de casos, onde o coordenador indica um perfil de caso e a solução para resolver o mesmo. Possui opções de navegação (primeiro, próximo, anterior, último), de manutenção (incluir novo caso, confirmar inclusão, gravar, excluir, editar, cancelar ação) e de retorno à tela principal. O cadastro de casos pode ser visto na figura 9.

The screenshot displays the 'SISTEMA PLANARTE - MEMÓRIA DE CASOS' interface. On the left, there is a vertical list of dropdown menus for configuration: Formato (EPS), Abre? (SIM), Fontes (OK), Gráficos (BAIXA RESOLUÇÃO), Imagens (BAIXA RESOLUÇÃO), Padrão de Cor (CMYK), Versão (COMPATIVEL), Sangria? (NÃO), Cor Preta (OK), Visualização (CORRETA), Válido? (SIM), and Imprime Fotelito? (NIHIL). Below these is a text area for 'Solução' containing the text 'PEDIR NOVO MATERIAL EM RESOLUÇÃO 300 DPI'. In the center, a box labeled 'Opções' contains six buttons: Novo, Confirmar, Gravar, Excluir, Editar, and Cancelar. On the right, a box labeled 'Navegação' contains five buttons: Primeiro, Próximo, Anterior, and Último. A 'Sair' button is located at the bottom right of the interface.

Figura 9 - Cadastro de casos

Na tela de consulta de casos, o plantonista indica o perfil do arquivo que está sendo aberto e clica em consultar, sendo então disparada a busca pelo caso mais similar e retornando uma solução e a porcentagem de similaridade. O formulário é composto de opções para consultar, reter, ignorar ou adaptar um caso. A opção de adaptação é composta por um *checkbox* de cada atributo do perfil de casos, onde poderá ser informado um ou mais atributos para ser feita a adaptação. A adaptação pode ser feita com base no caso consultado e na resposta obtida ou apenas preenchendo os valores do perfil e clicando em adaptar. O formulário de consulta de casos é demonstrado na figura 10.

SISTEMA PLANARTE - CONSULTA DE CASOS

Formato: EPS

Abre?: SIM

Fontes: OK

Gráficos: OK

Imagens: OK

Padrão de Cor: CMYK

Versão: COMPATÍVEL

Sangria?: NÃO

Cor Preta: 4 CAMADAS

Visualização?: CORRETA

Válido?: SIM

Imprime Fotelito?: SEM NECESSIDADE

Solução proposta: ATRAVÉS DO AJUSTE DE COR SELETIVA SELECIONAR A COR E REDUZIR A 0 (ZERO) A PORCENTAGEM DE OUTRAS CORES E ELEVAR A 100 (CEM) A PORCENTAGEM DE PRETO

Similaridade: **92 %**

Formato Versao
 Abre Sangria
 Fontes CorPreta
 Graficos Visualizacao
 Imagens Valido
 PadraoCor ImpFotelito
 Solucao

Adaptar

Planarte

Fim da busca

OK

Consultar Petic Ignorar Sair

Figura 10 - Consulta de casos

O caso acima foi encontrado através da comparação de cada atributo do perfil indicado com os existentes na memória de casos. O caso mais similar ao consultado pode ser conferido na figura 11.

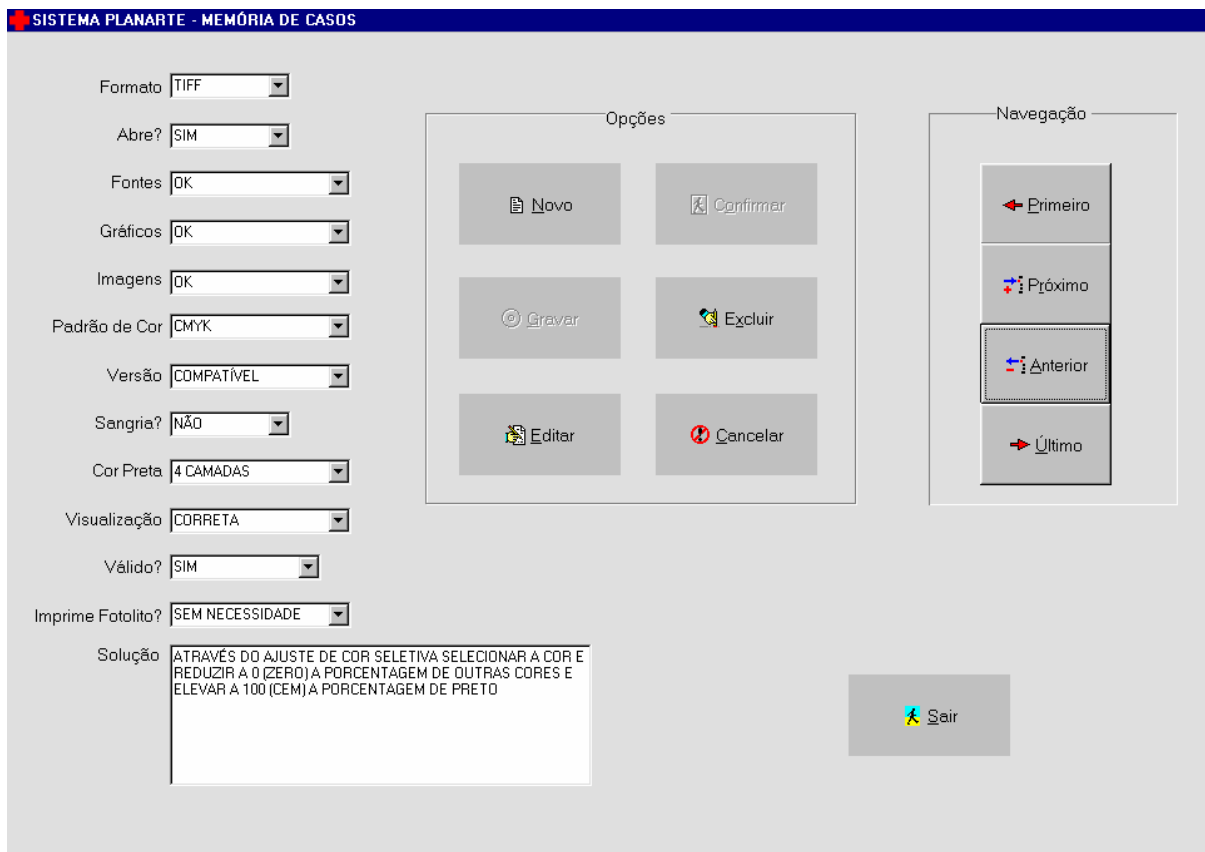


Figura 11 – Caso mais similar

A comparação entre os casos pode ser vista na tabela 3.

Tabela 3 – Comparação de casos

Atributos	Caso novo	Caso da base
Formato	EPS	TIFF
Abre?	Sim	Sim
Fontes	OK	OK
Gráficos	OK	OK
Imagens	OK	OK
Sistema de Cor	CMYK	CMYK
Versão	Compatível	Compatível
Sangria	Não	Não
Cor preta	4 camadas	4 camadas
Visualização	Correta	Correta
Válido?	Sim	Sim
Imprime Fotelito?	Sem necessidade	Sem necessidade
Solução:		<p>ATRAVÉS DO AJUSTE DE COR SELETIVA SELECIONAR A COR E REDUZIR A 0 (ZERO) A PORCENTAGEM DE OUTRAS CORES E ELEVAR A 100 (CEM) A PORCENTAGEM DE PRETO</p>

Como pode ser visto na figura acima, somente o formato não é coincidente. Considerando todos os atributos com peso 1 (um), o índice de similaridade entre os casos pode ser visto no quadro 4.

<p>Caso novo, Caso da base : $\frac{0 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1}{12} = \frac{11}{12} = 0,92$</p> <p>11 – número de atributos coincidentes;</p> <p>12 – número de atributos de um caso;</p> <p>0,92 – índice de similaridade (multiplicando por 100 = 92%).</p> <p>Solução proposta: ATRAVÉS DO AJUSTE DE COR SELETIVA SELECIONAR A COR E REDUZIR A 0 (ZERO) A PORCENTAGEM DE OUTRAS CORES E ELEVAR A 100 (CEM) A PORCENTAGEM DE PRETO</p>
--

Quadro 4 - Índice de similaridade

Como exemplo de adaptação utilizou-se o caso pesquisado na figura 10. Os atributos escolhidos para adaptação foram CorPreta, com valor “5 CAMADAS” e Imprime Fitolito, com valor “DESCONHECIDO”. Clicando em “adaptar” o sistema varre a base de casos procurando algum caso em que o perfil seja igual ao pesquisado anteriormente, com exceção destes dos dois atributos escolhidos. No exemplo, o sistema adaptou o caso, ou seja, encontrou e incluiu um caso com os novos valores na memória de casos. O resultado da adaptação pode ser conferido na figura 12.

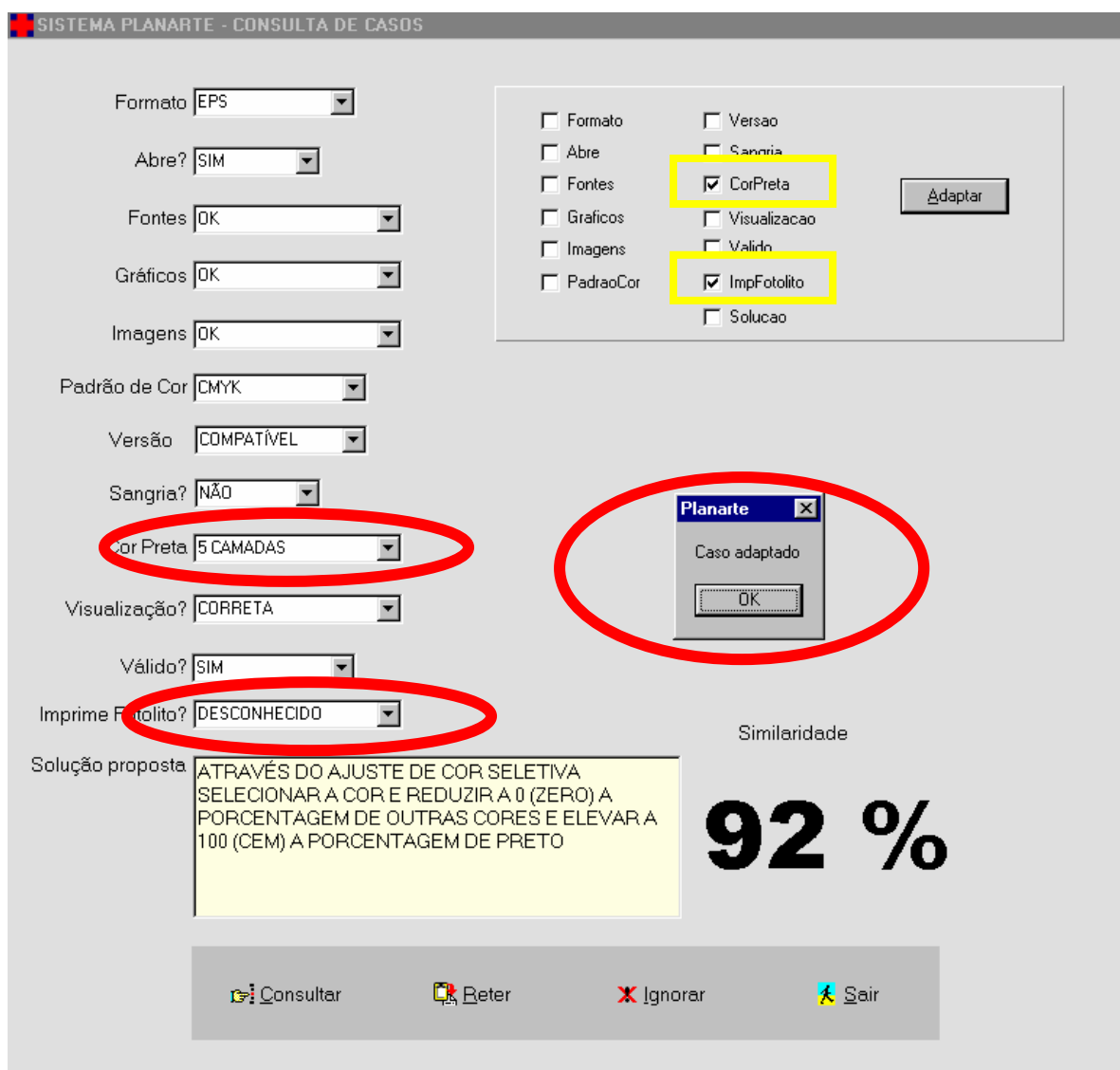


Figura 12 – Adaptação de um caso

4.4 RESULTADOS E DISCUSSÃO

Através dos testes feitos com a consultoria da empresa, os resultados mostraram-se promissores apresentando um ótimo índice de problemas resolvidos e com pouco tempo gasto. De acordo com o gráfico da figura 15, que mostra os resultados obtidos sem a ajuda do software de RBC, obviamente os resultados são melhores em relação ao número de casos solucionados, mas com custo de tempo maior. Em contrapartida, pode ser visto também o resultado obtido com o auxílio de RBC o qual possui uma taxa de acertos menor, mas com muito menos tempo requerido.

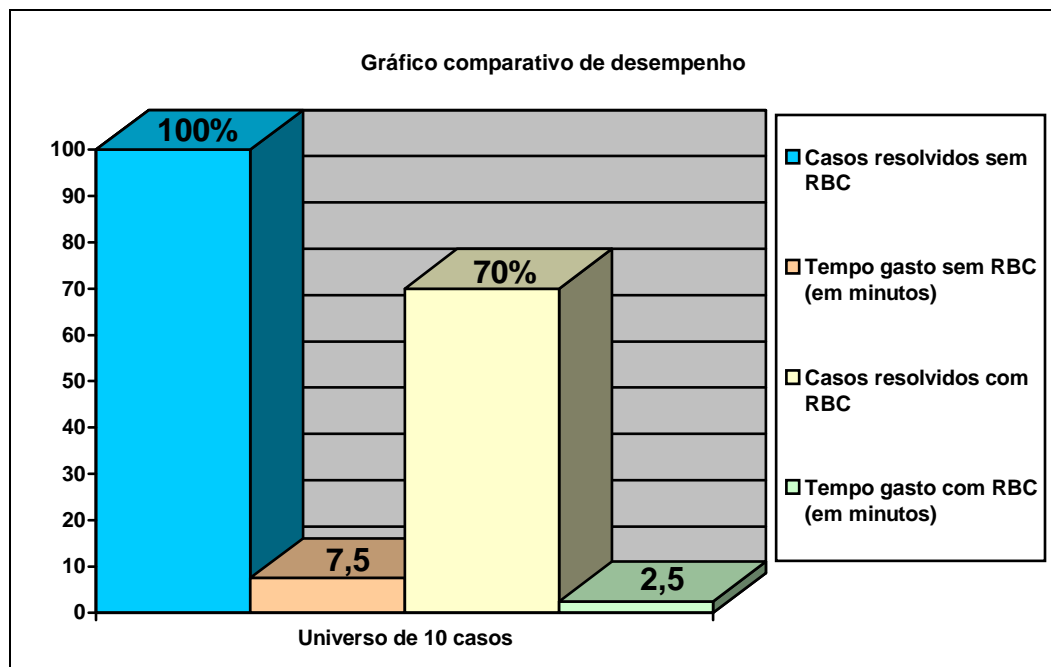


Figura 13 - Comparativo de desempenho

O desempenho, ao longo do tempo, deverá melhorar consideravelmente, pois a memória de casos tende a ser aumentada.

Assim como no trabalho correlato desenvolvido por Heinrich (2001), o presente sistema também pode ser utilizado como ferramenta de treinamento e aprimoramento dos plantonistas, através da periódica utilização.

5 CONCLUSÕES

Tendo em vista os objetivos propostos neste trabalho, conclui-se que foram alcançados. Foi implantada uma memória de casos aplicando os conceitos de RBC. Foram elaboradas pesquisas dos casos através da busca da similaridade pela técnica do Vizinho Mais Próximo; o sistema emitiu a resposta mais similar para a solução dos problemas; foi realizada a adaptação de casos.

As técnicas e ferramentas – análise estruturada, ambiente de desenvolvimento Delphi, banco de dados Paradox, técnica do Vizinho Mais Próximo – utilizadas foram consideradas suficientes e satisfatórias para especificação e implementação do sistema.

O sistema apresenta limitações no sentido de não informar que eventos causaram o problema no arquivo, como também os riscos que determinados procedimentos aplicados para a solução dos problemas podem acarretar no mesmo.

Apesar destas limitações, o sistema contribuiu para que os problemas em que as utilizações do software sejam eficazes, evitem submeter funcionários que não estejam em seu horário normal de trabalho a jornadas extras, evitando que sejam convocados para solucionar problemas, e sem utilização de recursos que poderiam onerar ainda mais o trabalho da empresa, como atraso no desenvolvimento das atividades e ligações telefônicas demoradas e às vezes desnecessárias.

Ao longo do tempo, a constante carga na memória de casos irá gerar as séries históricas de casos, que tornará o sistema menos limitado e mais confiável. Além disso, poderá se transformar também em um tutor para novos usuários e no suporte do dia a dia dos usuários.

5.1 EXTENSÕES

Como sugestão para trabalhos futuros, o sistema pode ser incrementado com as seguintes melhorias:

- a) salvar e monitorar a data de criação/adaptação do caso para verificar a “idade” da informação em relação à idade da base ou a frequência de uso do caso em relação à frequência de uso da base;

- b) apresentar outras respostas e não somente a mais similar, pois, em alguns casos, outras soluções podem ser mais adequadas que a primeira;
- c) manter a segurança da base de dados monitorando qual usuário manipulou o sistema e o que foi modificado, como também incorporar um sistema de *backup*;
- d) além da resposta obtida na consulta de um caso, apresentar também a maneira correta de emissão do material, para que o funcionário possa orientar o contato que o enviou;
- e) apresentar qual a consequência acarretada pela aplicação da solução proposta;

REFERÊNCIAS BIBLIOGRÁFICAS

ABEL, Mara. **Raciocínio baseado em casos – CBR**, [Porto Alegre], [2002]. Disponível em: <<http://marabel.inf.ufrgs.br/Publico/Disciplinas/SistEspecialistasN/CBR-Mara.pdf>>. Acesso em: 25 ago. 2004.

BELLIN, David & SUCHMAN, Susan. **Manual de desenvolvimento de sistemas estruturados**. São Paulo: Makron Books, 1993. 223 p.

BITTENCOURT, Guilherme. **Inteligência artificial: ferramentas e teorias**. 2ª. ed. Florianópolis: Ed. da UFSC, 2001. 362 p.

CAMARGO, Kátia Gavranich. **Inteligência artificial aplicada à nutrição na prescrição de planos alimentares**. Florianópolis, 1999. Disponível em: <<http://www.eps.ufsc.br/disserta99/camargo/figura/Image138.gif>>. Acesso em: 15 set. 2004.

CANTÚ, Marco. **Dominando o Delphi 3: a bíblia**. São Paulo: Makron Books, 1997. 1097 p.

CASTOLDI, Augusto César & SANTOS, Marcos de Oliveira. **Raciocínio baseado em casos**. Florianópolis, 2002. Disponível em: <<http://www.inf.ufsc.br/~barreto/trabaluno/IA20022AugMarc.pdf>>. Acesso em: 03 fev. 2005.

DAMASCENO, Américo Jr. **Aprendendo Delphi em Windows 95**. São Paulo: Érica, 1995. 191 p.

DATE, C.J. **Introdução a sistemas de bancos de dados**. Rio de Janeiro: Campos, 1991. 674 p.

GANE, Chris & SARSON, Trish. **Análise estruturada de sistemas**. Rio de Janeiro: Livros Técnicos e Científicos Editora Ltda (LTC), 1983. 257 p.

GANE, Chris. **Desenvolvimento rápido de sistemas**. Rio de Janeiro: Livros Técnicos e Científicos Editora Ltda (LTC), 1988. 170 p.

GARCIA, Luís Fernando Fortes. **Ferramentas case**, [Porto Alegre], [2004]. Disponível em: <<http://www.garcia.pro.br/ucs-engsoft/UCS-ES-P8-Case-Material1.pdf>>. Acesso em 29 ago. 2004.

HEINRICH, Daniel Jonas. **Sistema de apoio para o diagnóstico de defeitos em equipamentos eletrônicos aplica a oficinas eletrônicas utilizando Raciocínio baseado em casos**. 2001. 47 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

HEINZEN, Luiz Ângelo. **Módulo de raciocínio baseado em casos em uma ferramenta de apoio ao ensino de lógica de programação.** 2002. 62 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

KIENEN, Paulo César. **Sistema de informação aplicado na advocacia utilizando raciocínio baseado em casos.** 2003. 66 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

KOLODNER, Janet L. (ed.). **Case-based learning.** Norwell, Massachusetts: Kluwer Academic Publishers, 1993. 169 p.

LENZ, Mario et al. (ed). **Case-based reasoning technology: from foundations to applications.** Tokyo: Springer, 1998. 405 p.

LIMA, Cynthia Moreira & LABIDI, Sofiane. **Modelos da inteligência artificial.** São Luiz, Maranhão, [2004]. Disponível em: <http://cynthia_m_lima.sites.uol.com.br/ia.html#modelo>. Acesso em 01 set. 2004.

LOPES, Daniele Fortes. **Paradox x Interbase.** [Rio de Janeiro], [1996]. Disponível em: <<http://www.delphibr.com.br/artigos/pdxvsib.php>>. Acesso em 09 set. 2004.

LUGER, George F. & STUBBLEFIELD, William A. **Artificial intelligence and the design of expert systems.** The Benjamin/Cummings, Redwood City, California: 1989. 660 p.

PFLEEGER, Shari Lawrence. **Engenharia de software: teoria e prática.** 2. ed. São Paulo: Prentice Hall, 2004. 537 p.

MINELLA, Chaiene M. da Silva et. al. **Raciocínio baseado em casos utilizando a dieta do tipo sanguíneo.** Blumenau, [2004]. Disponível em: <http://campeche.inf.furb.br/siic/siego/docs/rbc_dieta.pdf>. Acesso em: 27 ago. 2004.

NILSSON, Nils J. **Principles of artificial intelligence.** Palo Alto, California: Morgan Kaufmann Publishers Inc, 1980. 476 p.

OLIVEIRA, Wilson José de. **Banco de dados interbase com Delphi.** Florianópolis: Bookstore Livraria Ltda, 2000. 196 p.

ORTEGA, Enrique. **Como fazer seu primeiro programa em Delphi 2.** Campinas, 1999. Disponível em: <<http://www.unicamp.br/fea/ortega/Delphi/Aulas/aula01.htm>>. Acesso em: 10 set. 2004.

RAMOS, Alexandre Moraes. **Modelo para incorporar conhecimento baseado em experiências à arquitetura TMN.** Florianópolis, 2000. Disponível em: <<http://teses.eps.ufsc.br/defesa/pdf/2992.pdf>>. Acesso em 22 set. 2004.

SWAN, Tom. **Delphi**: bíblia do programador. São Paulo: Berkeley Brasil, 1996. 654 p.

SYBASE. Power designer. São Paulo, [2002]. Disponível em:
<http://www.sybase.com.br/sybasenova/produtos/pdf/2002_SybasePowerDesigner9.5.pdf>.
Acesso em: 29 ago. 2004.

SOUZA, André Luiz. **Geração de SQL com PowerDesigner AppModeler**, [Uberaba], [2004]. Disponível em: <<http://xfk.vila.bol.com.br/any/appmod/appmod.htm>>. Acesso em: 26 ago. 2004.

TORRES, Gabriel. **Clube do hardware**, [Rio de Janeiro], [1997]. Disponível em:
<<http://www.clubedohardware.com.br/d281197.html>>. Acesso em: 02 set. 2004.

TVETER, Donald R. *The pattern recognition basis of artificial intelligence*. Los Alamitos, California: IEEE Computer Society, 1998. 369 p.

ANEXO A - Formatos mais utilizados na editoração eletrônica e suas características mais importantes.

FORMATO	CARACTERÍSTICA
BMP	O formato BMP é o formato mais "puro" que uma imagem pode aparecer. Os arquivos BMP são grandes, pois não utilizam nenhum esquema de compressão de dados. Por outro lado, arquivos BMP são compatíveis com todos os programas, inclusive com o próprio sistema operacional (o papel de fundo do Windows, por exemplo, é um arquivo BMP) e também não possuem limite de cores (um arquivo BMP pode ter qualquer número de cores).
CDR	Extensão de arquivo do Corel Draw que é um programa gráfico que gera desenhos vetoriais.
EPS e PS	Formato de um arquivo PostScript que contém uma imagem descritiva desse mesmo arquivo, possibilitando sua utilização em programa de paginação. Diferente do formato PS, ou seja, PostScript puro, que não contém essa imagem.
GIF	O formato GIF ocupa bem menos espaço, pois os dados são compactados. O padrão GIF utiliza somente 256 cores. Existem dois tipos de arquivos GIF: GIF padrão e GIF 89a, que permite imagens com fundo transparente.
JPG ou JPEG	Este é o padrão que permite arquivos de menor tamanho. Entretanto, seu esquema de compactação de dados é baseado em perda de informações. Por este motivo, o padrão JPG só deve ser utilizado em imagens prontas.
PCX (PC Paintbrush)	Este formato era muito utilizado na época do MS-DOS e está cada vez mais em desuso.
PSD (Photoshop)	Formato de arquivos utilizado pelo programa Photoshop. Alguns programas (como o Photostyler) também conseguem ler este tipo de arquivo. Sua principal característica é a utilização de layers (camadas), ou seja, várias imagens em camadas em um único arquivo.
TIF	Um dos padrões mais utilizados. Apesar de compactar a imagem, não há perda de dados.
WMF	Padrão gráfico da Microsoft, utilizado pelo Windows e aplicativos Microsoft, como os cliparts do Microsoft Office.

Fonte: adaptado de Lima & Labidi (2004).