

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

**SISTEMA DE AUXÍLIO A RESTAURANTES PARA
ELABORAÇÃO DE CARDÁPIOS**

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

GILVANI BRUNO KISNER

BLUMENAU, JUNHO/2003

2003/1-30

SISTEMA DE AUXÍLIO A RESTAURANTES PARA ELABORAÇÃO DE CARDÁPIOS

GILVANI BRUNO KISNER

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. Paulo Roberto Dias — Orientador na FURB

Prof. José Roque Voltolini da Silva — Coordenador do TCC

BANCA EXAMINADORA

Prof. Paulo Roberto Dias

Prof. Roberto Heinzle

Prof. Dr. Oscar Dalfovo

AGRADECIMENTOS

Agradeço aos meus pais que sempre me ajudaram em toda minha vida, muitas vezes, se privando de seu próprio conforto, para que eu pudesse estudar.

Ao professor orientador Paulo Roberto Dias, que me direcionou e me motivou a executar este trabalho.

Em especial, ao amigo, Roberto Fabiano Fernandes, que nestes anos de estudos, compartilhou com perseverança e empenho.

RESUMO

Neste trabalho de conclusão de curso aborda-se a aplicação de uma das técnicas da Inteligência Artificial o Raciocínio Baseado em Casos (RBC), mais especificamente a técnica da similaridade, no desenvolvimento de Sistemas de Informação.

Descrevem-se os principais conceitos relacionados com o assunto como Sistemas, Sistemas de Informação, Inteligência Artificial e RBC entre outros. O principal assunto se refere à aplicação do RBC no desenvolvimento de um Sistema de Informação para auxiliar profissionais de restaurantes na elaboração de cardápios.

ABSTRACT

This is a final term thesis on the application of one of the techniques of the Artificial Intelligence or Reasoning Based on Cases (RBC), more specifically the similarity technique, in the development of the Information Systems.

The main concepts related to the subject, such as Systems, Information Systems, Artificial Intelligence and RBC among others are explained here. The main subject is related to the application of the RBC in the development of an Information System to help professionals of restaurants in the preparation of menus.

LISTA DE FIGURAS

Figura 01: Diagrama de contexto	39
Figura 02: DFD – parte 1	40
Figura 03: DFD – parte 2	41
Figura 04: DFD – parte 3	42
Figura 05: MER – Modelo lógico	43
Figura 06: MER – Modelo físico	44
Figura 07: Menu principal	47
Figura 08: Cadastro de clientes	48
Figura 09: Cadastro de tipos de alimentos	48
Figura 10: Cadastro de alimentos	49
Figura 11: Cadastro de pesos	50
Figura 12: Cadastro de reservas	50
Figura 13: Elaboração do pré-cardápio	51
Figura 14: Coleta parâmetros para busca do RBC	52
Figura 15: Cardápios similares	53
Figura 16: Cadastro de refeições	53
Figura 17: Exemplo de busca com RBC por cardápios similares	54

LISTA DE QUADROS

Quadro 01: Tabela de clientes	44
Quadro 02: Tabela de empresa	44
Quadro 03: Tabela de estados.....	45
Quadro 04: Tabela de alimentos.....	45
Quadro 05: Tabela de tipos de alimentos	45
Quadro 06: Tabela de cardápios	45
Quadro 07: Tabela de itens do cardápio	45
Quadro 08: Tabela de reservas	45
Quadro 09: Tabela de itens da reserva.....	45
Quadro 10: Tabela de refeições	46
Quadro 11: Tabela de itens da refeição	46
Quadro 12: Tabela de pesos	46
Quadro 13: Linhas de código referente à busca com RBC e cálculo da similaridade.....	56

SUMÁRIO

1. INTRODUÇÃO.....	10
1.1 OBJETIVOS	11
1.2 ORGANIZAÇÃO	12
2. FUNDAMENTAÇÃO TEÓRICA	13
2.1 INTELIGÊNCIA ARTIFICIAL	13
2.1.1 CONCEITO	14
2.1.2 TÉCNICAS DE IA	15
2.1.3 PRINCIPAIS APLICAÇÕES DA IA.....	16
2.1.4 RACIOCÍNIO BASEADO EM CASOS	18
2.1.4.1 HISTÓRIA DO RBC.....	19
2.1.4.2 CASOS.....	20
2.1.4.3 CICLO DE UM SISTEMA RBC	20
2.1.4.4 MEMÓRIA DE CASOS	21
2.1.4.5 REPRESENTAÇÃO DE CASOS.....	21
2.1.4.6 INDEXAÇÃO DE CASOS.....	22
2.1.4.7 RECUPERAÇÃO DE CASOS	23
2.1.4.8 SIMILARIDADE	24
2.1.4.9 ADAPTAÇÃO	26
2.1.4.10 TIPOS DE SISTEMAS RBC	27
2.1.4.11 APLIAÇÕES DO RBC	27
2.2 RESTAURANTES.....	30
2.3 TECNOLOGIAS E FERRAMENTAS UTILIZADAS	32
2.3.1 ANÁLISE ESTRUTURADA.....	32
2.3.2 FERRAMENTAS <i>CASE</i>	34
2.3.3 POWER DESIGNER.....	35
2.3.4 A FERRAMENTA <i>CASE</i> GENEXUS	35
3 ESPECIFICAÇÃO E MODELAGEM DO SISTEMA.....	38
3.1 LISTA DE EVENTOS	38
3.2 DIAGRAMA DE CONTEXTO.....	38
3.3 DIAGRAMA DE FLUXO DE DADOS (DFD)	40

3.4	MODELO ENTIDADE RELACIONAMENTO (MER).....	43
3.5	DICIONÁRIO DE DADOS	44
4	IMPLEMENTAÇÃO DO SISTEMA	47
5	CONCLUSÃO.....	58
5.1	SUGESTÕES PARA TRABALHOS FUTUROS	58
	REREFÊNCIAS	59

1 INTRODUÇÃO

Atualmente o campo profissional de restaurantes vem crescendo muito e atuando em diferentes ramos da culinária humana. No que diz respeito ao atendimento diferenciado a clientes os restaurantes vem há algum tempo buscando novas alternativas na elaboração de cardápios alternados baseando-se na frequência e no grau de satisfação de seus clientes.

No entanto, uma das maiores dificuldades que estes profissionais enfrentam é selecionar manualmente as preferências alimentares de seus clientes e seu grau de satisfação. Baseando-se no grau de satisfação dos cardápios anteriores e a frequência de seus clientes, estes profissionais elaboram novos cardápios que serão servidos no decorrer da semana.

De acordo com Fayyad (1996) há tempos atrás a maioria das empresas, independentemente de seu ramo de atuação, eram de certa forma despreocupadas com o armazenamento das informações para uso futuro. O armazenamento era manual e a recuperação destas informações era difícil e lenta.

No que diz respeito ao armazenamento de dados de clientes, existem grandes quantidades de *softwares*, mas poucos são os que atuam juntamente com estes profissionais no auxílio da elaboração de cardápios.

De acordo com Rich (1988) Inteligência Artificial (IA) é o estudo de como fazer computadores realizarem tarefas em que, no momento, as pessoas são melhores. O comportamento inteligente por parte das máquinas sempre foi um objetivo para os homens. O computador, com seu poder de cálculo e armazenamento, pode ser a máquina destinada para isso. Por comportamento inteligente pode-se considerar, dentre outras coisas, a capacidade humana de compreender e aprender a partir de recordações, considerando-se um ponto chave de nossa memória (SCHILDT, 1989). Os novos fatos apresentados às pessoas são sempre comparados àqueles armazenados em busca de um padrão reconhecido e tratável.

O objetivo deste trabalho é abordar a relação entre o uso do computador no auxílio da elaboração de cardápios, mais especificamente como “instrumento de apoio”.

Para o desenvolvimento desse sistema utilizou-se de uma das técnicas da IA, a técnica de Raciocínio Baseado em Casos (RBC), que vem como uma ferramenta para se emular a inteligência em um sistema computacional, baseado na técnica de similaridade mais especificamente a técnica de recuperação do vizinho mais próximo. Através da utilização da

técnica de RBC foi possível a recuperação de cardápios similares ao novo cardápio elaborado, prevenindo para que não haja repetição de cardápios nos períodos estabelecidos. Conforme Abel (1996) nesta técnica utiliza-se uma soma ponderada das características entre um novo caso e um armazenado no banco de dados. A aplicação da técnica de RBC em sistemas de informações agiliza a busca de soluções para problemas. Banco de dados são eficientes em achar um problema igual, mas não é capaz de realizar um casamento por aproximação. Estas soluções podem ser respostas para as dúvidas dos clientes com relação a procedimentos ou a correção de problemas.

Conforme Reis (1997) faz parte do processo cognitivo do ser humano resolver problemas utilizando situações passadas que sejam similares. Especialistas, no geral, confiam excessivamente em sua memória de casos já ocorridos quando resolvem novos problemas. O raciocínio baseado em experiências passadas é importante, pois evita novas inferências sobre um problema, aproveitando as experiências já aprendidas. A maioria dos programas de IA soluciona problemas raciocinando com base em princípios básicos, ou seja, a partir dos dados de entrada, eles fazem uma série de deduções para chegar a uma conclusão. Mais especificamente em RBC esta técnica soluciona os problemas sob uma outra forma de raciocínio: fazendo uma analogia aos problemas já ocorridos. A experiência na solução de problemas é guardada em uma Base de Casos.

Conforme Kolodner (1993) caso é descrito como “um pedaço de conhecimento que representada uma experiência real e é utilizada para sugerir um meio de resolver um novo problema, avisar o usuário de possíveis falhas observadas no passado e para interpretar a situação atual”. Um sistema RBC tem como ponto forte uma enorme biblioteca de casos a qual usa para comparar com os dados de entrada em lugar de um conjunto de princípios básicos.

1.1 OBJETIVOS

O objetivo deste trabalho é implementar um sistema de apoio a restaurantes capaz de auxiliar na elaboração de novos cardápios.

Os objetivos específicos do trabalho são:

- a) permitir inclusões, alterações e exclusões de clientes, alimentos, cardápios e grau de satisfação das refeições na base de dados;

- b) buscar sugestões de cardápios similares;
- c) possibilitar a impressão de cardápios e dados cadastrais;
- d) evitar repetição de cardápios dentro de períodos pré-estabelecidos.

1.2 ORGANIZAÇÃO

Este trabalho está dividido em cinco capítulos, sendo que:

O primeiro capítulo é de introdução, apresenta uma visão geral do presente trabalho, o contexto em que está inserido, sua importância e objetivos.

O segundo capítulo apresenta uma fundamentação teórica sobre os temas: Inteligência Artificial, Raciocínio Baseado em Casos e uma visão geral sobre a história dos restaurantes e as técnicas utilizadas para o desenvolvimento deste trabalho.

O terceiro capítulo apresenta a especificação e a modelagem do sistema.

O quarto capítulo apresenta o sistema desenvolvido. Suas características, e as principais telas.

O quinto capítulo apresenta as conclusões e sugestões para que o mesmo possa ter continuidade e seja melhorado.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta uma fundamentação teórica sobre os temas: Inteligência Artificial, Raciocínio Baseado em Casos, história dos restaurantes e as técnicas utilizadas para o desenvolvimento do sistema.

2.1 INTELIGÊNCIA ARTIFICIAL

Conforme Bittencourt (1998), IA foi constituída das idéias filosóficas, científicas e tecnológicas herdadas de outras ciências, algumas tão antigas quanto à lógica, com seus 23 séculos.

A primeira análise filosófica séria sobre a possibilidade de implementos num programa inteligente em computadores digitais foi proposto por *Turing*, onde propõe o teste de *Turing*, que na versão original é chamado Jogo de Imitação.

Os primeiros estudos sobre IA surgiram na década de 40, que foi marcada pela II Guerra Mundial. Este fato resultou na necessidade de desenvolver uma tecnologia voltada para a análise de balística, quebra de códigos e cálculos para projetar a bomba atômica. Surgia, assim, os primeiros grandes projetos de construção de computadores, assim chamados por serem máquinas utilizadas para fazer cálculos (*cômputos*).

Conforme Barreto (1997) o segundo grande passo foi dado nos Estados Unidos, em 1956, quando John McCarthy reuniu em uma conferência proferida ao Darmouth College, na Universidade de New Hampshire, vários pesquisadores de renome para estudar o que foi denominado por Minsky, McCarthy, Newell e Simon de IA, expressão utilizada para designar um tipo de inteligência construída pelo homem para dotar a máquina de comportamentos inteligentes.

A partir da estruturação desse novo campo do conhecimento o fenômeno da inteligência começou a ser pesquisado de forma intensa.

A IA é amplamente utilizada como um auxiliar que expande a capacidade de inteligência do homem e até mesmo o substitui em diversas funções.

2.1.1 CONCEITO

Conforme Levine (1988) IA é simplesmente uma maneira de fazer o computador pensar inteligentemente. Isto é conseguido estudando como as pessoas pensam quando estão tentando tomar decisões e resolver problemas, dividindo esses processos de pensamento em etapas básicas e desenhando um programa de computador que solucione problemas usando essas mesmas etapas. IA então fornece um método simples e estruturado de se projetar programas complexos de tomada de decisão.

Para Rich (1988) IA é o estudo de como fazer computadores realizarem tarefas em que, no momento, as pessoas são melhores.

Winston (1988) diz que: “IA é o estudo de conceitos que permitem aos computadores serem inteligentes”.

Para Rabuske (1998) IA é o resultado da aplicação de técnicas e recursos, especialmente de natureza não numérica que viabiliza a solução de problemas que exigiram do humano certo grau de raciocínio e de perícia. A solução destes problemas com recursos tipicamente numéricos é muito difícil. Por isso é que IA caracteriza uma nova era da computação, a era do processamento não numérico.

Conforme Passos (1989) os objetivos da IA são, principalmente fazer máquinas e processos computacionais:

- a) mais úteis: ajudar o ser humano a tomar decisões mais rápidas;
- b) com entendimentos inteligentes: as comunicações com as máquinas podem ser feitas na linguagem corrente, linguagem natural, sem que o usuário tenha que aprender a linguagem do computador.

IA não trata fundamentalmente de processos numéricos, como faz PD (processamento de dados). E sim trata com conhecimento que são coisas abstratas, simbólicas. Programas computacionais com as quais IA está relacionada são primeiramente processos simbólicos, que envolvem:

- a) complexidade: tem de pensar para responder, não é uma resposta imediata;
- b) incerteza: o computador com um *software* de IA só responde o que sabe. Isto é, de acordo com o conhecimento nele armazenado;

- c) ambigüidade: como um ser humano ao escutar uma frase, às vezes, pode interpretar com dois sentidos diferentes, a mesma coisa pode acontecer com o computador inteligente.

2.1.2 TÉCNICAS DE IA

Conforme Rich (1988) IA é o fato de que a inteligência requer conhecimento. Para compensar sua característica predominante, a indispensabilidade, o conhecimento também possui algumas propriedades menos desejáveis, incluindo:

- a) é volumoso;
- b) é difícil de caracterizar com precisão;
- c) está em constante mutação.

Uma comparação levanta uma pergunta interessante a respeito do modo como as pessoas resolvem problemas e o modo como os computadores o fazem. Por que estas pessoas acham a abordagem de leitura de linhas mais fácil enquanto a abordagem de contagem de números é mais eficiente para o computador? Não se pode saber o suficiente sobre como as pessoas operam, para responder de um modo completo a essa pergunta. Em parte a resposta está no fato de que as pessoas são processadores paralelos, podendo observar por exemplo diversas partes de um tabuleiro de xadrez, enquanto o computador convencional só pode observar um quadro de cada vez. Às vezes uma investigação de como as pessoas resolvem um problema lança muita luz sobre o modo como os computadores deveriam fazê-lo. Em outras ocasiões, as diferenças entre tais modos afiguram-se tão grandes que exigem estratégias diferentes.

Três importantes técnicas de IA são citadas por Rich (1988) sendo:

- a) busca: fornece um meio de resolver problemas para os quais haja uma abordagem mais direta disponível, bem como uma estrutura dentro da qual qualquer técnica direta que estiver disponível pode ser encaixada;
- b) utilização do conhecimento: fornece um meio de resolver problemas complexos explorando as estruturas dos objetos que estão envolvidos;
- c) abstração: fornece um meio de separar características e variações importantes de outras irrelevantes que poderiam, de modo, inutilizar qualquer processo.

Para a solução de problemas difíceis, programas que explorem essas técnicas levam vantagem sobre aquelas que não exploram. Sendo muito menos frágeis, não serão completamente desviados se ocorrer uma pequena perturbação em sua entrada. As pessoas podem facilmente compreender o que seja o conhecimento do programa. Essas técnicas podem funcionar para grandes problemas onde os métodos mais diretos podem desmoronar.

2.1.3 PRINCIPAIS APLICAÇÕES DA IA

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 a seguir.

Processamento de Linguagem Natural: é o 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 (LEVINE, 1988). Objetiva aperfeiçoar a comunicação entre as pessoas e os computadores.

Reconhecimento de Padrões: é uma das áreas de pesquisa bem avançadas da IA. De acordo com Barreto (1997) Reconhecimento de Padrões é o processo de identificar objetos, através da extração de suas características a partir de dados sobre o objeto. Esta capacidade é altamente desenvolvida nos seres humanos e em muitos animais. A capacidade de Reconhecimento de Padrões permite ao programa reconhecer a fala em linguagem natural, os caracteres digitados e a escrita (ex.: assinatura). Os *scanners*, por exemplo, utilizam programas de reconhecimento óptico desenvolvidos pelas pesquisas em IA.

Programação de Jogos: é o estudo voltado para a construção de programas de jogos envolvendo raciocínio. Conforme Rich (1988) os jogos computadorizados são um grande sucesso, ainda mais quando exibem um tipo de inteligência capaz de desafiar as habilidades do jogador. O jogo de xadrez, por exemplo, foi utilizado para as primeiras experiências em programação do raciocínio artificial, onde o computador se tornou capaz de analisar milhões de jogadas por segundo para tentar derrotar o adversário. Além de analisar as jogadas, os programas utilizam um método heurístico que consiste na utilização de uma árvore de busca, a mesma possui ramificações a partir de certos nós, que representam pontos de decisão no caminho a tomar, com um certo número de etapas, para chegar a um objetivo. Deste modo, o

computador pode analisar vários nós, de acordo com a situação atual do jogo, e escolher o melhor caminho (o mais curto ou menos arriscado).

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). A expressão robô vem do tchêco *robot*, significa trabalhador, foi criada por Karel Capek, em 1917. O primeiro robô industrial do mundo, batizado de *UNIMATE*, surgiu em 1962. Barreto (1997) define robô como: “Um robô é um manipulador programável multifuncional capaz de mover material, partes, ferramentas ou dispositivos especializados através de movimentos variáveis programados para realizar uma variedade de tarefas”.

Redes Neurais Artificiais: conforme Tatibana (2003) redes neurais artificiais consistem em um método de solucionar problemas de IA, construindo um sistema que tenha circuitos que simulem o cérebro humano, inclusive seu comportamento, ou seja, aprendendo, errando e fazendo descobertas. São mais que isso, são técnicas computacionais que apresentam um modelo inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência. Uma grande rede neural artificial pode ter centenas ou milhares de unidades de processamento, enquanto que o cérebro de um mamífero pode ter muitos bilhões de neurônios.

Sistemas Especialistas: conforme Bittencourt (1998) um Sistema Especialista (SE) é aquele que é projetado e desenvolvido para atender a uma aplicação determinada e limitada do conhecimento humano. É capaz de emitir uma decisão, apoiado em conhecimento justificado, a partir de uma base de informações, tal qual um especialista de determinada área de conhecimento humano. Todo o conhecimento em um SE é fornecido por pessoas que são especialistas naquele domínio. Os SE são baseados em informações atuais e precisas e utilizam o formato de regras de produção como método de representação de conhecimento. Os SE são concebidos para reproduzir o comportamento de especialistas humanos na resolução de problemas do mundo real, mas o domínio destes problemas é altamente restrito.

Raciocínio Baseados em Casos: conforme Kolodner (1993) RBC é uma ferramenta da IA, que utiliza conhecimento de experiências passadas para resolver problemas atuais. O paradigma que sustenta esta técnica, que em muitos aspectos é fundamentalmente diferente das outras técnicas de IA, é a capacidade de utilizar o conhecimento específico de uma experiência anterior para resolver uma situação nova.

2.1.4 RACIOCÍNIO BASEADO EM CASOS

Se observarmos a maneira como as pessoas resolvem seus problemas, estaremos observando o RBC na prática. Em vários momentos o RBC é aplicado no dia-a-dia: quando pedimos um tipo de carne no restaurante, estamos fazendo este pedido baseado em momentos anteriores que estivemos naquele restaurante; quando planejamos nossas atividades diárias, lembramos quais delas funcionaram ou não, e em cima disto, criam-se nossos planos. Quando as pessoas resolvem problemas, a segunda vez é sempre mais fácil que a primeira porque lembra-se e repete-se a mesma solução, com suas devidas características atuais, tentando evitar os erros cometidos na primeira vez.

Conforme Lens (1998) a resolução de um problema requer as seguintes ações:

- a) o que são problemas e como são representados;
- b) o que significa dizer que algo é a solução de um problema;
- c) como uma experiência passada é representada;
- d) o que significa dizer que algo é semelhante;
- e) o que significa procurar uma solução;
- f) como usar uma solução antiga.

Os conceitos gerais de problemas e soluções não têm nenhuma definição geral e concreta e variam de aplicação para aplicação. Problemas e soluções podem ser na realidade noções gerais. Uma solução também pode ser uma ação, um plano, ou uma parte de informação fornecida pelo usuário.

Conforme Kolodner (1993) 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.

Sistemas de RBC armazenam e indexam informações sobre experiências passadas em uma biblioteca de casos. Casos representam conhecimento operacional específico amarrado a situações específicas. Todo caso possui um conjunto de índices que são combinações de discriminadores que unicamente identificam um caso do outro. Os sistemas de RBC simulam o ato humano de relembrar um episódio prévio para resolver um determinado problema em função da identificação de afinidades entre os mesmos.

O ato de lembrar um episódio anterior é simulado em um sistema de RBC por meio da comparação de um novo problema com um conjunto de casos do mesmo tipo. Este conjunto chama-se Base de Casos. A comparação é efetuada através da avaliação de similaridade entre o novo episódio com os contidos na base de casos. A avaliação de similaridade é executada ao nível dos índices. Os casos mais similares são recuperados. Uma fase de seleção determina qual o caso mais útil completando o ato de lembrar.

Conforme Lens (1998) o RBC tem duas motivações principais:

- a) do ponto de vista cognitivo: é o de modelar o raciocínio e aprendizado nos homens;
- b) do ponto de vista computacional: é de construir sistemas de IA eficientes.

2.1.4.1 HISTÓRIA DO RBC

Conforme Reis (1997) a origem do RBC em IA é encontrada nos trabalhos de Roger Schank, sendo que o primeiro sistema que pode ser considerado de RBC foi o *CYRUS*, desenvolvido por Janet Kolodner em 1983. Este sistema foi baseado no modelo de memória dinâmica de Roger Schank e na teoria de MOPs (*memory organization packets* – pacotes de organização de memória) para aprendizagem e solução de problemas.

Outras bases para RBC e outros conjuntos de modelos foram desenvolvidos por Bruce Porter e seu grupo.

Conforme Reategui (1999) o RBC tem uma motivação cognitiva muito forte. Um grupo liderado por Roger Schank (que originalmente era psicólogo) em meados de 80 começou a definir esta teoria de que pessoas raciocinam usando casos; isto pode ser usado de maneira muito eficiente em sistemas de computadores.

Roger Schank em 1982 descreve a memória dinâmica e traz como regra central a importância das recordações de casos passados na resolução de problemas e aprendizagem Delpizzo (1997). Atualmente, trabalhos envolvendo RBC tanto nos Estados Unidos quanto na Europa têm se expandido e o número de publicações vem crescendo consideravelmente em atividades relacionadas a IA.

2.1.4.2 CASOS

Caso é uma forma de conhecimento contextualizado representando uma experiência que ensina uma lição útil. Lições úteis são aquelas que tem o potencial para ajudar o raciocinador alcançar uma meta ou um conjunto de metas ou advertem sobre a possibilidade de uma falha ou apontam para um futuro problema Reis (1997).

Conforme Reategui (1999) os casos podem ser classificados em 3 tipos:

- a) ossificados: são casos que parecem muito com regras, porque foram abstraídos de casos; provérbios, conhecimentos "populares". Se uma pessoa está magra é porque não está comendo bem. São como regras generalizadas, mas diferem um pouco de casos paradigmáticos porque não precisam necessariamente estar ligados a um contexto;
- b) paradigmáticos: Casos típicos, vistos seguidamente, que ajudam a resolver problemas;
- c) histórias: casos únicos, e cheios de detalhes. Uma pessoa vai ao *MacDonalds* e acontece sempre a mesma coisa. Um dia, esta pessoa vai ao *MacDonalds* e ao invés de hambúrguer eles servem feijão, bife, batata doce, porque era o dia da descoberta do Brasil. O caso é tão único, que esta pessoa não vai esquecer. E talvez usar isto pra compreender outras situações. Histórias devem ter índices mais complexos para que sejam recuperadas somente em ocasiões mais raras e apropriadas.

2.1.4.3 CICLO DE UM SISTEMA RBC

Conforme Reis (1997) um sistema de RBC funciona por comparar a descrição de um problema a ser resolvido com os casos descritos em uma base de casos. O sistema recupera então o caso mais parecido, avalia a necessidade de adaptar a solução associada e aplica essa solução ao novo problema. Resumidamente, os passos de solução podem ser descritos por:

- a) coleta de informações sobre o problema;
- b) recupera um conjunto de casos candidatos: um conjunto de casos que assemelham-se ao novo problema é selecionado na base de casos;

- c) cada um dos casos candidatos é comparado ao novo problema utilizando-se o algoritmo de vizinhança, que mede a diferença de valores de cada um dos atributos de ambos os casos;
- d) o caso que obtiver a menor soma das diferenças é escolhido o melhor caso;
- e) a solução associada ao caso é avaliada para ver se é apropriada;
- f) se necessário, a solução é adaptada para compensar as diferenças entre os casos;
- g) o novo caso assim gerado é, por sua vez, armazenado na base de casos com a solução adotada.

2.1.4.4 MEMÓRIA DE CASOS

A principal fonte de conhecimento do modelo RBC são as experiências vividas pelos especialistas. Essas experiências são problemas que os especialistas já resolveram. Cada uma delas é representada como um caso. Conforme Kolodner (1993) o objetivo do modelo RBC é fazer uso dessas experiências para resolver novos problemas, torna-se necessário que os casos representem o que elas têm de importante em relação às resoluções dos problemas. Eles devem representar estas experiências numa forma tal que elas possam ser recuperadas quando forem úteis, ou seja, os casos devem ser identificados pelo que eles têm de útil.

Para Abel (1996), existem dois modelos de organização de casos: memória dinâmica e categoria de exemplares. O modelo de memória dinâmica usa uma estrutura hierárquica de pacotes de organização de memória (MOPs), também chamados de episódios generalizados. Um MOP engloba um conjunto de casos que contém características comuns. O modelo de categoria de exemplares considera que os casos do mundo real podem ser vistos como exemplares de acontecimento. Cada caso é associado a uma categoria e suas feições têm importância para enquadrá-lo ou não na categoria. Para que um novo caso seja armazenado um caso semelhante é buscado na memória de casos. Se houver pequenas diferenças entre os dois, apenas um será armazenado, ou será feita uma combinação dos dois.

2.1.4.5 REPRESENTAÇÃO DE CASOS

Conforme Abel (1996) casos podem ser armazenados no computador nos mais diferentes formatos, no entanto, muitos deles são excessivamente complexos para serem manipulados pelo desenvolvedor do sistema ou para serem compreendidas pelo usuário do sistema. A representação deve considerar formatos que tenham uma correspondência natural

com a forma como a informação costuma estar disponível para evitar o desperdício de processamento em traduções de uma estrutura para outra.

Os casos registram experiências concretas que podem auxiliar a alcançar um determinado objetivo, porém nem todos os casos devem ser selecionados para serem incluídos no sistema. Apenas aqueles que contêm uma lição útil em relação aos demais devem ser armazenados. Ou seja, os casos que ampliam a capacidade de solução de problemas do sistema. Isso significa que casos que repetem uma situação anterior, apenas com pequenas modificações não deveriam ser incluídos, já que essas diferenças podem ser compensadas pelos algoritmos de adaptação do sistema.

Conforme Heinrich (2001) ao se desenvolver um sistema utilizando RBC, é preciso estipular como a memória de casos será organizada e indexada para que a recuperação de um novo caso seja de forma eficiente.

2.1.4.6 INDEXAÇÃO DE CASOS

Conforme Kolodner (1993) indexação de casos é a associação de rótulos em casos de forma a caracterizá-los para posteriormente recuperá-los em base de casos.

Conforme Heinrich (2001) a indexação de casos é uma das tarefas mais importantes no RBC. Uma boa indexação permite que o caso seja recuperado da base de casos com mais eficiência e certeza. A recuperação certa para o caso certo é um fator chave para a credibilidade de uma aplicação em RBC. Para Rodrigues (1999) RBC utiliza índices para acelerar a recuperação. Informações em um caso podem ser de dois tipos: informações indexadas que serão utilizadas para recuperação e informações não indexadas que provêm informação contextual de valores para o usuário, mas não são usadas diretamente na recuperação. Índices devem ser preditivos, endereçar os propósitos para os quais o caso será usado, ser abstrato para permitir ampliação futura da base de casos, e ser concreto para ser reconhecido no futuro.

A indexação é uma forma de distinguir o que é importante sobre um caso e, os índices são a base para algoritmos de procura e as estruturas de organização utilizadas na base de casos Reis (1997).

Conforme Kolodner (1993) dois aspectos devem ser enfocados ao tratar os índices: o primeiro é definir o vocabulário e o segundo é como estes índices vão ser valorados.

De acordo com Delpizzo (1997) a indexação dos casos é feita a partir de um conjunto de características que representam um caso. A função da indexação é orientar a avaliação de similaridade dos casos na base de casos.

2.1.4.7 RECUPERAÇÃO DE CASOS

Conforme Reis (1997) a recuperação começa com uma descrição do problema e termina quando o melhor caso é encontrado, sendo esta tarefa subdividida em:

- a) identificação das características: informa ao sistema as características do caso atual. Para Lee (1998) “a sub-tarefa de identificação é somente necessária em domínios de aplicação onde as características, índices ou quaisquer que sejam os elementos alvo da avaliação da similaridade não sejam diretos ou não estejam claros”;
- b) casamento inicial: recupera um conjunto de candidatos plausíveis;
- c) busca: é um processo mais elaborado de selecionar o melhor candidato entre os casos recuperados durante o casamento inicial;
- d) seleção: o processo de seleção gera conseqüências e expectativas de cada caso recuperado e tenta avaliar as conseqüências e justificar as expectativas. Os casos são eventualmente ordenados de acordo com a métrica ou algum critério de classificação, desta forma o caso que possui a mais forte sustentação de similaridade ao novo problema é escolhido.

Conforme Rodrigues (1999) a recuperação de casos é dependente do método de representação utilizado. Em geral duas técnicas são usadas por aplicações de RBC comerciais:

Nearest-Neighbor Retrieval: Calcula-se a distância relativa entre um caso alvo e os outros casos. Aquele que obtiver o menor valor é o caso vizinho mais próximo. Esta solução utiliza-se de uma medida de similaridade, que pode utilizar-se de pesos diferentes para cada atributo, onde a soma da similaridade de todos os atributos é calculada.

Conforme Abel (1996) nesta técnica utiliza-se uma soma ponderada das características entre um novo caso e o armazenado no banco de dados, sendo que cada um dos atributos que compõem o caso possui um peso, de acordo com a sua relevância.

Inductive Retrieval: Esta técnica é utilizada por muitas ferramentas de RBC, envolve um processo chamado indução. Indução é uma técnica desenvolvida por pesquisadores de aprendizado de máquina para extrair regras ou construir árvores de decisão a partir de dados passados. Em sistemas RBC o Baseamento de Casos é analisado por um algoritmo de indução para produzir a árvore de decisão que classifica ou indexa os casos. O algoritmo de indução mais largamente utilizado é chamado de ID3. Este algoritmo constrói uma árvore de decisão a partir de uma base de casos. Utiliza-se uma heurística chamada ganho de informação para procurar o atributo mais promissor, a partir do qual a árvore será dividida ao meio.

2.1.4.8 SIMILARIDADE

De acordo com Lee (1998), similaridade é a essência do RBC, uma vez que o fundamento do paradigma de RBC é solucionar um problema atual, reutilizando a solução de uma experiência passada similar.

O processo de avaliação da similaridade em sistemas de RBC refere-se à comparação de dois casos para avaliar como um está relacionado ao outro no que concerne a como estes casos podem compartilhar soluções e conseqüências. Esta avaliação é executada no nível dos atributos, associando-se valores cuja natureza determina a função de combinação a ser empregada. Funções de combinação comparam valores numéricos e alfanuméricos exatamente ou dentro de intervalos. A comparação entre textos pode ser feita por similaridade sintática ou semântica. Uma forma simples e direta de modelar similaridade é atribuindo-se um valor de uma (1) unidade para a função resultante similar e o valor zero (0) para o resultado não similar; e ainda atribuir valores intermediários entre estes.

Do acordo com Delpizzo (1997) a métrica de similaridade é uma função que mede a similaridade entre dois casos, ela é utilizada para guiar a busca pelos casos mais similares que serão ordenados segundo a mesma. Conforme Lee (1998) uma métrica é normalmente necessária em sistemas nos quais os casos são comparados um a um e a medida de sua similaridade é o meio de distinguir entre os casos candidatos similares e não similares.

A métrica de similaridade tem por objetivo dar um valor numérico à similaridade entre dois casos. Todos os casos da memória são avaliados comparativamente ao problema

(caso) de entrada. Normalmente, estabelece-se um grau de similaridade entre os atributos de determinadas dimensões, um número dentro do intervalo [0,1].

Existem dois aspectos fundamentais da recuperação: qual a similaridade entre os casos para cada dimensão e qual a importância de cada dimensão. Para modelar computacionalmente os diferentes graus de importância das características utiliza-se pesos ou algum mecanismo similar.

Há ainda o conceito de *threshold* (limiar), que limita a quantidade de casos que serão recuperados.

Delpizzo (1997) cita a seguinte fórmula de similaridade:

$$\text{Similaridade } (T, S) = \sum_{i=1}^n f(T_i, S_i) \times W_i$$

Onde:

- a) T é o caso de entrada;
- b) S é o caso da base;
- c) n é o número de atributos de cada caso;
- d) i é um atributo individual;
- e) f é a função de similaridade para o atributo i nos casos T e S ;
- f) W é o peso dado ao atributo i .

Conforme Heinrich (2001) a maioria das ferramentas RBC utiliza algoritmos como este. Normalmente o resultado deve ser zero (0) e um (1), onde zero não tem nenhuma similaridade e um é exatamente similar.

Exemplo de cálculo de similaridade para recuperação de casos, conforme Varela (1998):

Atributos \ Casos	A	B	C
X ₁	Raciocínio	Sistemas	Inteligente
X ₂	Inteligente	Inteligente	Métricas
X ₃	Análise	Robótica	Similaridade
X ₄	Casos	Computador	Análise
X ₅	Baseado	Análise	Prototipagem

Casos Atributos	X ₁	X ₂	X ₃	X ₄	X ₅
Caso novo	Raciocínio	Inteligente	Análise	Casos	Sistemas

Atribuindo 1 para atributos coincidentes e 0 para não coincidentes:

Casos Atributos	X ₁	X ₂	X ₃	X ₄	X ₅
Caso novo A	1	1	1	1	0
Caso novo B	0	1	1	0	1
Caso novo C	0	1	0	1	0

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

$$Sim(casosnovo, A) = \frac{1 + 1 + 1 + 1}{5} = \frac{4}{5} = 0,8$$

$$Sim(casosnovo, B) = \frac{1 + 1 + 1}{5} = \frac{3}{5} = 0,6$$

$$Sim(casosnovo, C) = \frac{1 + 1}{5} = \frac{2}{5} = 0,4$$

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

2.1.4.9 ADAPTAÇÃO

A tarefa final do raciocínio baseado em casos é adaptar a solução associada a um caso recuperado para as necessidades do problema corrente. Quando uma situação é fornecida, o algoritmo de recuperação traz o melhor caso que ele encontrar para a memória. Normalmente, o caso selecionado não casa perfeitamente com a descrição do problema do usuário. Existem diferenças entre o problema do usuário e o caso contido no banco de dados que devem ser levadas em conta. O processo de adaptação procura por diferenças salientes entre as duas descrições e aplica regras de forma a compensá-las Abel (1996).

Conforme Kolodner (1993) adaptação tem a função de alterar um caso, se necessário, para solucionar o problema de entrada. Avaliam-se as diferenças entre os casos recuperados e o problema de entrada. A adaptação pode tomar várias formas: inclusão, exclusão, substituição ou transformação de algum item no caso antigo.

2.1.4.10 TIPOS DE SISTEMAS RBC

Conforme Reategui (1999) os sistemas RBC são geralmente classificados de acordo com alguns parâmetros: quanto ao tipo de problema tratado e quanto à forma de utilização.

Quanto ao tipo de tratado:

- a) *planificação*: método para encontrar uma seqüência de passos capazes de nos levar a um estado desejado (objetivo). Planejamento de um prato, planejamento dos passos necessários para se fazer um diagnóstico;
- b) *design*: problemas são definidos através de *constraints* e o sistema tem que encontrar uma solução que resolva o *constraint*. Concepção de pequenos sistemas mecânicos e elétricos. Concepção de projetos arquitetônicos;
- c) *interpretação*: encontrar uma justificativa para um problema (*legal reasoning*), encontrar casos que possam justificar uma atitude do réu;
- d) *diagnóstico e classificação*: encontrar um diagnóstico a partir de uma descrição do problema (encontrar uma classe de uma descrição: encontrar a classificação de uma pedra preciosa a partir de uma descrição).

Quanto à forma de utilização:

- a) sistemas completamente automatizados: resolução total de problemas;
- b) sistemas de recuperação de casos: sistemas que trabalham de forma interativa com uma pessoa para resolver um problema. Estes sistemas têm o objetivo de aumentar a memória da pessoa resolvendo o problema, fornecendo casos para a pessoa considerar que ela talvez conhecesse ou não. Mas a pessoa é que é responsável por decidir qual é a solução a adotar.

2.1.4.11 APLICAÇÕES DO RBC

Conforme Rodrigues (1999) a princípio qualquer problema pode ser implementado em RBC. Não é necessário que exista um algoritmo para a resolução de um problema para resolvê-lo em RBC, inclusive aponta-se para uma de suas vantagens, a capacidade de prover a resolução de um problema sem a necessidade de algoritmos ou aquisição de conhecimento. Basta que o mesmo problema (ou outro similar) já tenha sido resolvido.

As aplicações do RBC podem ser classificadas em dois tipos de problemas principais: tarefas de classificação e tarefas de sintetização. Tarefas de classificação cobrem

uma grande faixa de aplicações que compartilham de características comuns, onde um novo caso é identificado com os que estão na base de casos de forma a determinar o tipo ou classe do caso em si. Podem ser tarefas como: diagnóstico, previsão, avaliação, controle de processos ou planejamento. Tarefas de sintetização tentam criar uma nova solução a partir da combinação de partes de soluções previamente aplicadas, podendo ser tarefas de desenho, planejamento e configuração.

Conforme Kolodner (1993) RBC pode ser usado para várias tarefas de raciocínio como, por exemplo: propor soluções para novos problemas, antecipar, evitar e explicar fracassos em soluções propostas, e adequar e/ou reparar soluções propostas. Em todas estas tarefas RBC tem suas vantagens quando comparado a outros métodos de raciocínio mais tradicionais. Um sistema que usa o modelo RBC pode desenvolver uma grande capacidade para aprender, porque seu processo de aprendizagem é muito simples. Um sistema que usa esse modelo de raciocínio aprende quando ele permite que problemas recém-resolvidos possam ser acrescentados à sua memória de casos, de forma que eles possam ser utilizados para resolver novos problemas. Esse é um aprendizado que ocorre a partir dos sucessos do sistema ao determinar a solução de um problema. Mas um sistema RBC também aprende de seus fracassos: quando uma porção proposta falha, o sistema pode gerar uma explicação dos motivos da falha e armazenar um exemplar no caso fracassado em sua memória.

Um sistema RBC pode solucionar problemas em domínio apenas parcialmente compreendidos porque ele trabalha com experiências. Mas, ao armazenar um caso de sucesso ou de fracasso em sua memória o sistema, na realidade, está aumentando a sua compreensão e o seu conhecimento do domínio. E, isso, permite que a sua performance possa melhorar como, por exemplo:

- a) aumentando a sua capacidade de recuperar casos mais similares aos novos problemas, o que implica em soluções mais adequadas e que exigem menos adaptações;
- b) casos similares a casos passados, que foram resolvidos em vários passos devido a suas complexidades, podem agora ser solucionados mais facilmente, porque não necessitam repetir os caminhos traçados pelos outros, basta adaptar as suas soluções;
- c) situações similares a casos passados que fracassaram evitam repetir o mesmo erro, sendo indexadas por características que impedem esses casos de serem

recuperados. Casos fracassados também tem outras utilidades: antecipar fracassos em novos casos;

- d) consciência: é uma forma de consciência que sistemas de RBC possuem em relação a suas próprias limitações (RODRIGUES, 1999). Se o sistema não encontra casos com a devida similaridade como 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;
- e) permitem a construção de protótipo nas fases iniciais do projeto, antes da compreensão e estruturação completa do domínio. Aquisição de casos em domínios pouco estruturados ou com base teórica mal definida;
- f) boa amostragem dos tipos de problemas que o sistema deve resolver;
- g) casos podem ser bons indicadores de quais partes do problema são importantes e quais as características que o especialista considera relevantes para a solução. Substituem entrevistas demoradas;
- h) ajuda a "clusterização" de conhecimento (conhecimento que vai ser usado junto, é adquirido e armazenado modularmente);
- i) extração de conhecimento: Basta ter um banco de dados. Conforme Rodrigues (1999) a difícil extração de 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 de conhecimento.

Abel (1996) cita as seguintes desvantagens na utilização do RBC:

- a) dificilmente os casos estão disponíveis de forma confiável, em quantidade suficiente e com boa representatividade sobre o domínio;
- b) armazenar casos demanda maior espaço de armazenamento do que armazenar um modelo que represente todas as instâncias de problemas;
- c) uma representação por modelo cobre todo o domínio, enquanto que a representação por casos irá representar somente a porção do domínio a que se referem às instâncias dos problemas;

- d) os algoritmos de recuperação por similaridade não mostram bom desempenho em aplicações reais, onde as comparações feitas por um especialista tendem a ser muito mais sofisticadas e de difícil compreensão;
- e) não existem bons algoritmos de adaptação que permitam ao sistema fornecer soluções adequadas ao problema apresentado pelo usuário.

O RBC está sendo muito utilizado nas indústrias, muitas companhias de grande porte internacionais selecionaram o RBC como método de resolução para certos tipos de problemas.

Por exemplo:

- a) *CFM International* (França e USA): diagnóstico de falhas nos motores de Boeings 737;
- b) *Sextant Avionique* (França): diversos tipos de diagnóstico em aeronaves *Airbus*;
- c) *Analog Devices* (USA): suporte de vendas para circuitos integrados;
- d) *Legrand* (França): estimativa do custo de produção de peças em plástico;
- e) *Ericsson* (França): análise de cartas de telecomunicação e sistemas eletrônicos de testes de dados;
- f) *Sepro Robotique* (França): "help desk" para suporte após venda de robôs;
- g) *Alstom* (França): otimização no *shceduling* de trens;
- h) *Ansaldo Transporti* (Itália): manutenção no metrô de Napoles;
- i) *Wartsila* (Suíça): extensão do ciclo de vida de motores marinhos.

2.2 RESTAURANTES

Conforme Filho (2000) a palavra restaurante vem do latim "*restaurare*" (restaurar), no sentido de restabelecer o vigor físico, de reparar, de remeter ao bom estado físico ou recuperar as forças. Lugar, onde comida e bebida, podem ser procurados pelo público, mediante pagamento.

A sala pública que ficou conhecida como "Restaurante", tem sua origem na França, e os franceses continuam ainda hoje em dia, a oferecer as maiores contribuições para o desenvolvimento dos restaurantes. O primeiro proprietário de um restaurante acredita-se que tenha sido Monsieur Boulanger, um vendedor ambulante de sopas, que abriu um negócio em Paris, em 1765. Os vendedores ambulantes na Europa Ocidental, após a Idade Média, ocuparam um papel significativo para a expansão do mercado, vendendo as mais diferentes e exóticas mercadorias. Compreende-se que a inovação de Monsieur Boulanger, que vendia

sopa de porta em porta, deve ter advindo de uma refeição de excelente paladar e qualidade, pois foi capaz de inverter o sentido comercial do seu negócio.

"Boulangier", na língua francesa, significa "padeiro". Os nomes de família ou sobrenomes, na França, tem suas origens na profissão ou local de origem da família. Provavelmente Monsieur Boulangier fazia honrar seus antepassados padeiros, pois é impensável uma refeição na França, sem o acompanhamento de um bom pão. Muito menos, em se tratando de uma sopa. Provavelmente seu grande chamariz deve ter sido a possibilidade de oferecer a refeição quente, com um pão quentinho feito na hora. Na realidade, seu prato principal era uma "Sopa de Coxa de Carneiro ao Molho Branco".

O desenho de um prato e uma colher, em cima da porta do seu estabelecimento, referia-se às sopas e caldos revigorantes ou restaurativas, disponíveis no interior do prédio. A palavra "Restaurante", com pequenas modificações, denota um lugar público para comer no Brasil, França, Espanha, Inglaterra, Alemanha e muitos outros países. "*Ristorante*" na Itália, "*Restaurang*" na Suécia, "*Restoran*" na Rússia ou "*Restaurancia*" na Polônia. É verdade que muitas estalagens e hospedarias já ofereciam refeições pagas em "*table d'hôtel*" (mesa do hotel) e bebidas oferecidas em "Cafés". O Restaurante de Boulangier foi provavelmente o primeiro lugar público onde se podia solicitar uma refeição a partir de um "menu", uma relação de pratos. A primeira cafeteria que se tem notícia, a "*Kiva Han*", foi inaugurada em 1475 em Constantinopla, logo após a tomada da cidade pelos Turcos Otomanos em 1453, que determinou o fim do Império Bizantino e do Período Medieval.

A idéia do modesto restaurante de Monsieur Boulangier produziu em 1782 a inauguração do primeiro restaurante de luxo em Paris, o "*La Grande Taverne de Londres*". O proprietário, Antoine Beauvilliers, uma autoridade gastronômica e mais tarde autor culinário, escreveu em 1814 "*L'Art du Cuisinier*" ("A Arte do Cozinheiro"), um livro que se tornou uma referência para na arte culinária francesa da época. Podemos dizer, que a Revolução Francesa de 1789, originou, entre outras coisas, as refeições públicas de qualidade, em consequência de um nobre que teve que trabalhar para garantir seu sustento.

Beauvilliers alcançou rapidamente a reputação de "*restaurateur*" eficiente e bom hospedeiro. O cronista e referência gastronômica francesa da época, Jean-Athelme Brillat-Savarin, um convidado freqüente de Beauvilliers, creditava a este como sendo o primeiro a combinar quatro itens essenciais a um restaurante de sucesso; um salão elegante, garçons

simpáticos, farto cardápio e cozinha de superior qualidade. Antes da Revolução Francesa, muitos aristocratas franceses proprietários de mansões, tinham estabelecimentos com culinária bem elaborada. Mas a Revolução de 14 de julho de 1789 reduziu o número de proprietários e permitiu que os empregados, "*Chefs*" e "*Cuisiniers*", abrissem seus próprios estabelecimentos. Por volta de 1804, Paris tinha cerca de 500 restaurantes, produzindo os maiores "*Chefs*" e receitas da história da culinária mundial.

As equipes eram compostas de um "*Chef*", com um chapéu de pano branco e alto, que era o encarregado da cozinha, um "*Chef*" de molhos, um "*Entremetier*" encarregado da preparação das sopas, vegetais e doces, um "*Rôtisseur*" para preparar as carnes fritas e grelhadas, um "*Gardemanger*" responsável pelos suprimentos e pratos frios. Cada um, na equipe tinha seus deveres e obrigações claramente definidas, nos tempos modernos atuais, para diminuir o custo com pessoal e aumentar a rapidez do serviço, essas regras rígidas deixaram de existir.

2.3 TECNOLOGIAS E FERRAMENTAS UTILIZADAS

Para a realização deste trabalho foram utilizadas algumas ferramentas e tecnologias, a fim de especificar a análise e desenvolver o sistema. Neste capítulo serão abordadas a análise estruturada, as ferramentas *CASE Power Designer* e *Genexus*.

2.3.1 ANÁLISE ESTRUTURADA

Conforme Demarco (1989) análise é o estudo de um problema, que acontece à tomada de uma ação. No domínio específico do desenvolvimento de sistemas computacionais, análise refere-se ao estudo de alguma área de trabalho ou de uma aplicação, levando quase sempre à especificação de um novo sistema.

A conclusão bem-sucedida da fase de análise envolve:

- a) selecionar um alvo ótimo;
- b) produzir documentação detalhada desse alvo de tal forma que a implementação subsequente possa ser avaliada para ver se o alvo foi ou não atingido;
- c) produzir prognósticos precisos dos importantes parâmetros associados ao alvo, incluindo custos, benefícios, escalonamentos e características de desempenho;

- d) obter acordo (conformidade e opiniões) em cada um dos itens acima, de cada uma das áreas afetadas.

Análise Estruturada é a utilização das ferramentas descritas abaixo:

Diagrama de Fluxo de Dados (DFD): DFD é uma representação em rede de um sistema. O sistema pode ser automatizado, manual ou misto. O DFD retrata o sistema em termos de suas partes componentes, com todas as interfaces entre componentes indicados.

Diagrama Entidade-Relacionamento (DER): é uma forma de representação gráfica do conhecimento que se tem sobre o ambiente (realidade) qualquer. Mostra uma visão estática das informações (entidades) de interesse e dos vínculos (relacionamentos) existentes entre elas. Conforme Martin (1994) o modelo de dados é uma nova forma de comunicação entre o técnico de processamento de dados e o usuário.

Essa nova forma de comunicação assegurará que:

- a) o modelo de dados conterá todos os dados necessários para suportar os processos de responsabilidade do usuário;
- b) o modelo de dados conterá os dados para suportar processos que serão modificados ou introduzidos em um futuro próximo.

Dicionário de Dados (DD): conforme Yourdon (1990) DD é uma listagem organizada de todos os elementos de dados pertinentes ao sistema, com definições precisas e rigorosas para que o usuário e o analista de sistemas possam conhecer todas as entradas, componentes de depósitos e cálculos intermediários.

Português Estruturado: conforme Demarco (1989) Português Estruturado é uma linguagem de especificação, que faz uso de um vocabulário limitado e uma sintaxe limitada. O vocabulário do Português Estruturado consiste apenas em:

- a) verbos no imperativo na linguagem portuguesa;
- b) termos definidos no DD;
- c) algumas palavras reservadas para formulação lógica.

Tabelas de Decisão: conforme Yourdon (1990) Tabela de Decisão é uma maneira de expressar, em forma de tabela, qual o conjunto de condições que é necessário ocorrer para que um determinado conjunto de ações deva ser executado. O ponto principal de uma tabela de

decisão é a regra de decisão, que define o conjunto de ações a ser tomado, a partir de um conjunto de condições.

Árvores de Decisão: conforme Demarco (1989) Árvore de Decisão é a representação gráfica de uma Tabela de Decisão, nada mais, nada menos. É útil para exatamente o mesmo tipo de programa de ação como o das Tabelas de Decisão. Recomenda-se a utilização de uma Árvore de Decisão apenas para usuários que não podem simplesmente ser persuadidos a olhar uma Tabela de Decisão. Devido a sua aparência familiar e apresentação gráfica, uma Árvore de Decisão funciona como uma ferramenta autodidática.

2.3.2 FERRAMENTAS CASE

Sua origem teve início na década de 80 quando ficou evidente que ferramentas gráficas, tais como DFD, DER e gráficos de estruturas poderiam ser úteis em análise de projetos. Conforme Fisher (1990) estas metodologias oferecem uma estrutura de projeto e um conjunto de formalismos e normas que servem como base para conduzir o desenvolvimento de *software*.

Ferramentas *CASE* (*Computer-Aided Software Engineering*) são um conjunto de ferramentas desenvolvidos para auxiliar no processo de desenvolvimento de *softwares*. Dentre as grandes questões no desenvolvimento de *softwares* destacam-se: melhor qualidade, maior produtividade, melhor gerenciamento e custos menores.

As ferramentas para engenharia de *software* computadorizada reduzem substancialmente, ou eliminam inúmeros problemas de projeto e desenvolvimento inerentes aos projetos de médio e grande porte, por meio da geração automática de grande parte do *software* e com base nos projetos especificados.

O objetivo principal da tecnologia *CASE* é separar o projeto do programa aplicativo da implementação de código, em geral, quanto mais afastado estiver o processo de projeto melhor.

As ferramentas *CASE* são uma expansão natural das ferramentas já existentes. A maioria das ferramentas atuais, compiladores, depuradores e geradores de perfis concentra-se nas fases de implementação, teste e lançamento (gerência de configuração) do desenvolvimento do *software*. As ferramentas *CASE* estendem esta cobertura ajudando os analistas e engenheiros nas fases de análise dos requisitos e de especificação do projeto.

Se uma ferramenta *CASE* de projeto gera código (ou pelo menos parte dele) automaticamente tem-se duas vantagens: em primeiro lugar como a ferramenta está gerando parte do código automaticamente, o tempo de implementação será substancialmente menor. Em segundo lugar a qualidade do código gerado torna-se mais confiante, pois foi gerado por uma ferramenta totalmente depurada e testada. Assim poupa-se tempo na fase do teste de unidade como na implementação.

O desenvolvimento de *software* é um processo complexo e complicado. É preciso que se adquira um conhecimento razoável do que acontece em cada fase é onde as ferramentas *CASE* podem oferecer assistência. A essência da engenharia de *software* computadorizada é oferecer impulso durante as fases do desenvolvimento, afim de economizar esforços e despesas nas fases posteriores.

2.3.3 POWER DESIGNER

Com o crescimento da informática, as empresas obtiveram grande crescimento na busca de informações relacionadas ao mercado, produto, financeiro que ajudam a empresa na tomada de decisão. Para que este processo ocorra de forma correta é necessário partir de uma boa estruturação na base de dados.

Segundo Souza (2002), *PowerDesigner* é uma ferramenta que não pode faltar à um administrador de banco de dados, pois é uma ferramenta que facilita a administração e criação lógica e física em vários banco de dados como Access, Oracle, Paradox, e outros. Através do *PowerDesigner* é possível também fazer uma engenharia reversa de uma base de dados, facilitando a correção de erros na mesma.

Neste trabalho optou-se pela ferramenta *CASE Power Designer* por trazer facilidades na utilização da Análise Estruturada.

2.3.4 A FERAMENTA CASE GENEXUS

Para implementação deste trabalho utilizou-se da ferramenta *CASE Genexus*. Conforme ARTech (2002) Genexus é uma ferramenta para desenvolvimento de aplicações. Seu objetivo é auxiliar os analistas de sistemas na implementação de aplicações no menor tempo e com a melhor qualidade possível. Em linhas gerais, o desenvolvimento de uma aplicação implica tarefas de análise, projeto e implementação. O caminho seguido por

Genexus para atingir estes objetivos é o de liberar as pessoas das tarefas automatizáveis, por exemplo à implementação, permitindo a elas concentrar-se nas tarefas realmente difíceis e não automatizáveis, como compreender o problema do usuário.

Do ponto de vista teórico, Genexus é uma ferramenta associada a uma metodologia de desenvolvimento de aplicações. O processo de desenvolvimento e manutenção da aplicação é bastante simplificado quando realizado pelo Genexus. O ciclo de vida de uma aplicação desenvolvida em Genexus é composto de três fases:

Desenho: nesta fase é realizada a análise do sistema. É também nesta fase que são descritas as visões dos usuários, as estruturas de dados, os atributos, as fórmulas e as regras de trabalho. O Genexus utiliza-se de cinco objetos para desenhar uma aplicação:

- a) transações: visões do usuário que têm um diálogo associado e que podem modificar o conteúdo da base de dados. São a entrada básica do conhecimento da ferramenta Genexus. O conhecimento da vida real é capturado através de transações definidas pelo usuário construindo-se uma base de conhecimento, a partir da qual cria-se uma base de dados e os programas que permitem modificações e consultas. Neste caso, por uma transação entende-se a modificação interativa da base de dados, permitindo ao usuário criar, modificar ou eliminar informação;
- b) painéis de trabalho: são consultas interativas que permitem a recuperação de informações de forma dinâmica, orientando a pesquisa em tempo de execução. Os painéis de trabalho conhecidos como *work panel* permitem ao usuário o encadeamento por evento para capturar informações, navegar livremente através de telas, elegendo as ações que serão disparadas e executadas sobre elementos selecionados. Indiretamente chamando procedimentos adequados, podem determinar modificações na base de dados;
- c) relatórios: são os objetos que permitem realizar consultas, geradas pela aplicação. Os relatórios são consultas *batch* (não interativas), que não modificam o conteúdo da base de dados;
- d) procedimentos: estes objetos têm todas as características dos relatórios, só que permitem ainda criar ou modificar as informações da base de dados, mas sem necessariamente, envolver diálogo, ou seja, tipicamente processos *batch*;

- e) *web objects*: possuem as mesmas características das *Work Panels*, porém com diálogo assíncrono, via *Internet* ou *Intranet*.

Prototipação: O protótipo é importante por que permite que toda aplicação seja testada antes de se passar à produção. Durante os testes o usuário pode trabalhar com dados reais, prever alterações necessárias a adequação correta do sistema. Descreve-se abaixo algumas vantagens da prototipação:

- a) permite ver resultados de forma imediata;
- b) permite a entrega parcelada dos requerimentos do usuário;
- c) detecção de erros de forma rápida;
- d) cumpre o compromisso dos usuários com o desenvolvimento;
- e) sistemas de melhor qualidade.

Produção: a diferença existente na prototipação e produção é que a prototipação é desenvolvida em um ambiente de microcomputador, enquanto a produção é realizada no ambiente desejado pelo usuário. A ferramenta Genexus trabalha com os seguintes ambientes:

- a) IBM OS/2, AT, XT, PC;
- b) IBM AS/400;
- c) rede de microcomputadores;
- d) processamento corporativo entre microcomputadores e AS/400.

Utilizando o Genexus, a tarefa básica do analista é a descrição da realidade. A partir do momento que isto modifica a atividade do analista e, também, o de seu perfil de otimização, isto o transforma em um analista de negócios. Agora trabalha em alto nível, discutindo problemas com o usuário e testando com ele as especificações em nível de protótipo, em vez de desenvolver sua atividade e através de tarefas de baixo nível como, desenhar arquivos, normalizar, desenhar programas, programar, buscar e eliminar os erros dos programas.

A partir da base de conhecimento, o Genexus gera automaticamente tanto os programas de criação / reorganização da base de dados como os da aplicação.

3 ESPECIFICAÇÃO E MODELAGEM DO SISTEMA

Este capítulo apresenta os métodos utilizados para a especificação e modelagem do sistema conforme descritos a seguir:

3.1 LISTA DE EVENTOS

Conforme Shiller (1992) a Lista de Eventos é a ferramenta que ajuda a examinar o que acontece no ambiente (sobre o qual o sistema não tem controle). A Lista de Eventos é uma lista textual dos "eventos", ou estímulos no ambiente externo, aos quais o sistema deve responder a uma indicação da pessoa ou sistema que inicia o evento. Os eventos podem ser obtidos: do nada, colhendo-os de um documento de especificação do usuário ou colhendo-os do sistema atual.

Os eventos são enumerados a seguir:

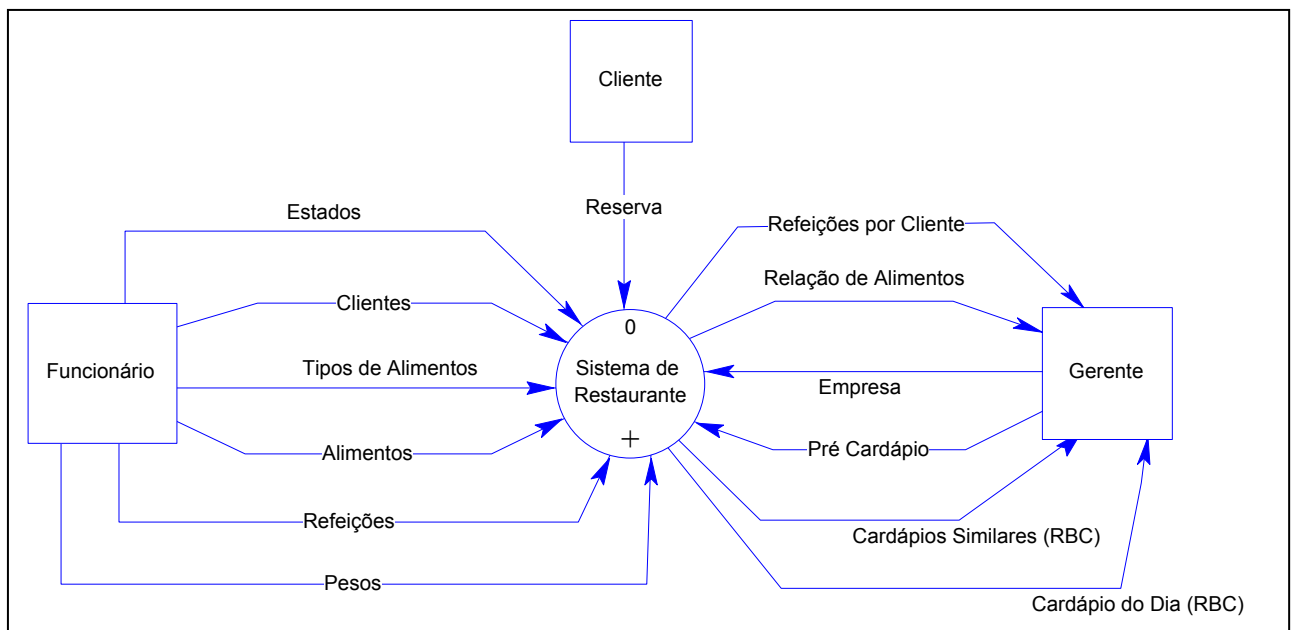
- 1) estado é cadastrado;
- 2) empresa é cadastrado;
- 3) cliente é cadastrado;
- 4) tipo de alimento é cadastrado;
- 5) alimento é cadastrado;
- 6) peso é cadastrado;
- 7) cliente efetua reserva;
- 8) elaborar pré-cardápio;
- 9) gerar cardápios similares;
- 10) refeição é cadastrada;
- 11) emitir cardápio do dia;
- 12) emitir refeições por clientes;
- 13) emitir relação de alimentos.

3.2 DIAGRAMA DE CONTEXTO

Conforme Shiller (1992) o Diagrama de Contexto é um Diagrama de Fluxo de Dados (DFD), que possui apenas um processo representando o sistema inteiro. Os fluxos de dados mostrados no DFD (Contexto) representam os fluxos de dados entre o sistema e com quem ele se comunica (outros sistemas, pessoas, departamentos, e outros). O evento determina um propósito para o comportamento do sistema e dá uma perspectiva diferente do Diagrama de Contexto.

No diagrama de contexto mostrado na figura 01 são apresentados os relacionamentos do RBC com as entidades externas.

Figura 01: Diagrama de contexto



3.3 DIAGRAMA DE FLUXO DE DADOS (DFD)

O DFD do sistema é apresentado nas figuras 02,03 e 04

Figura 02: DFD – parte 1

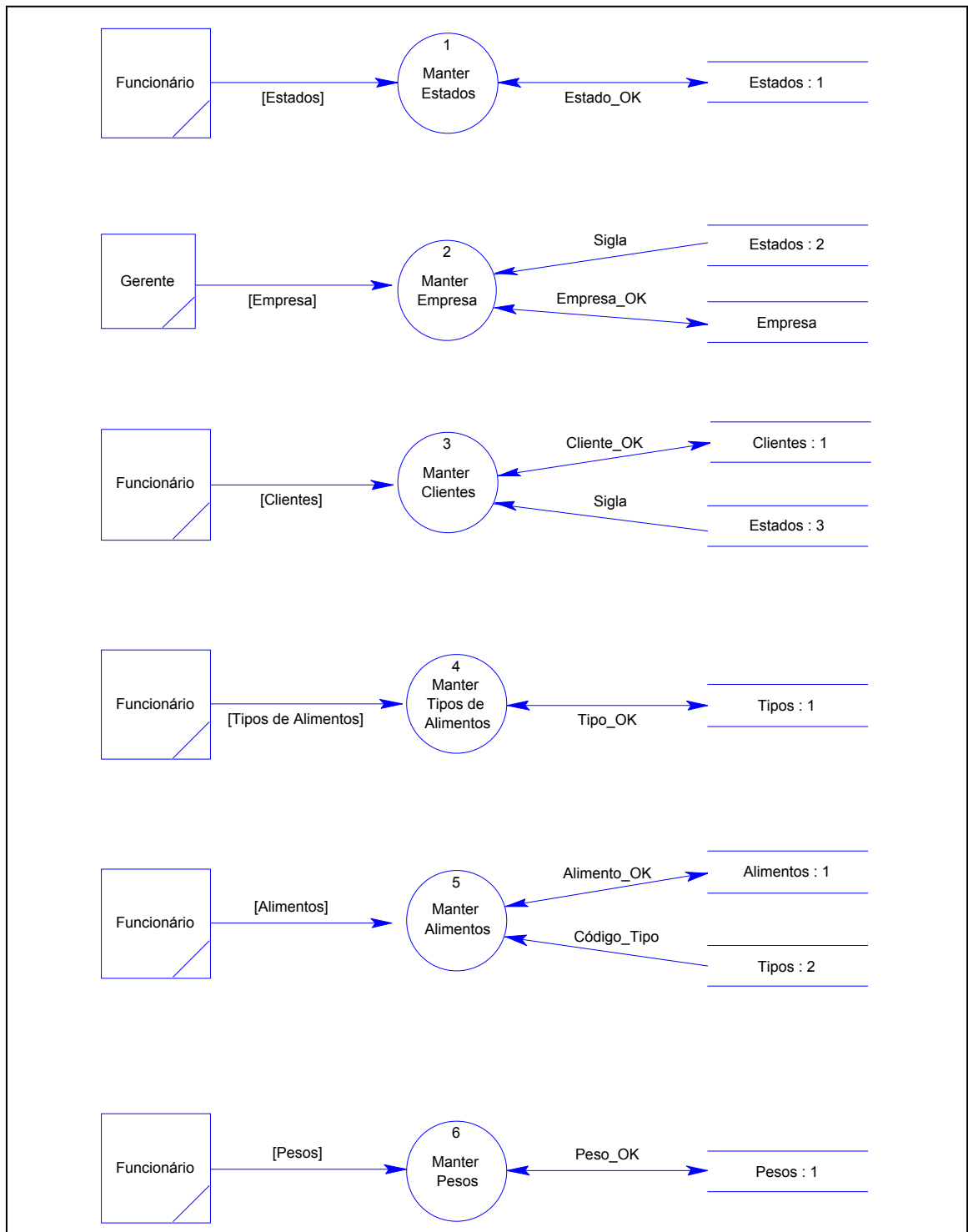


Figura 03: DFD – parte 2

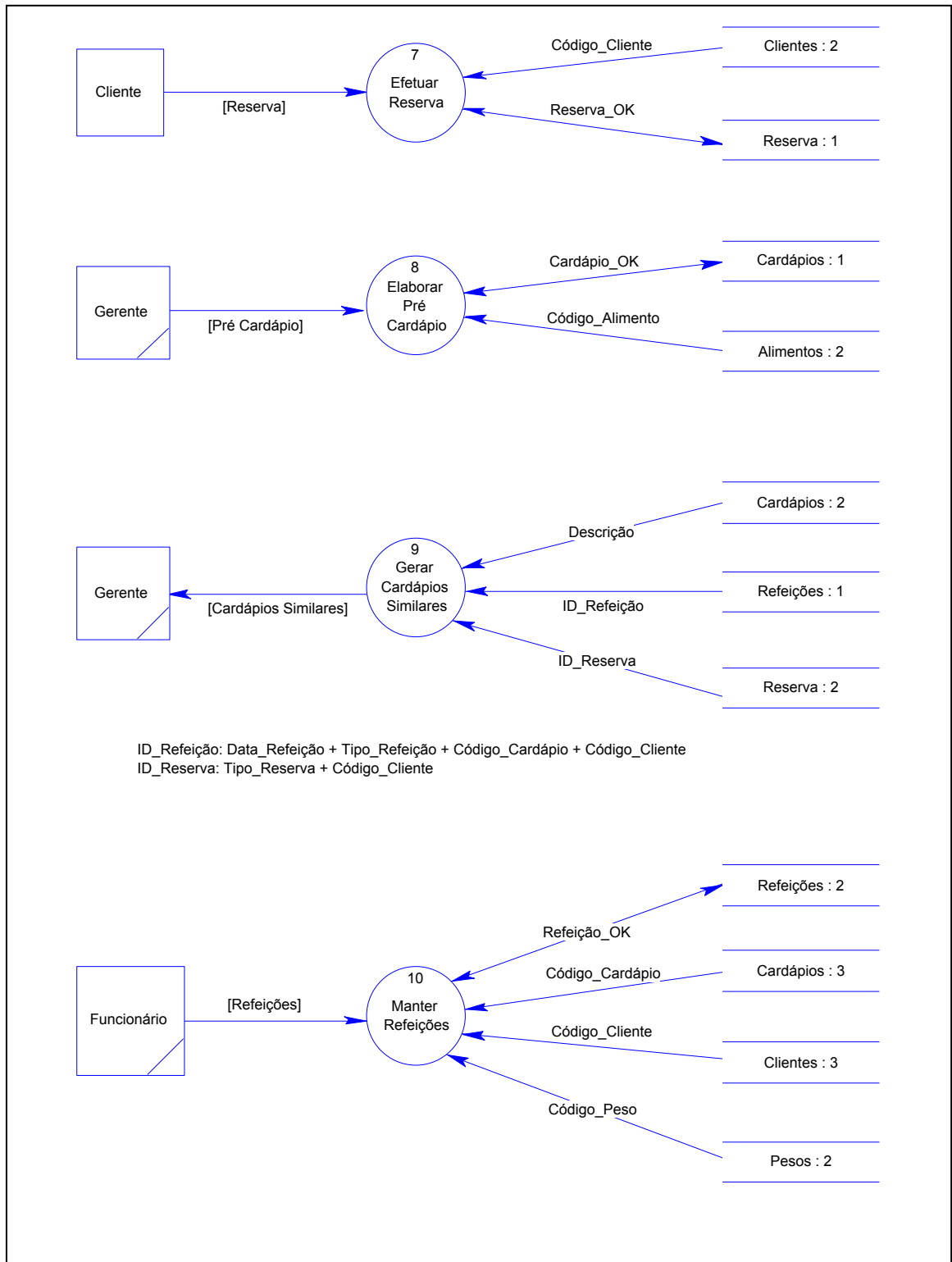
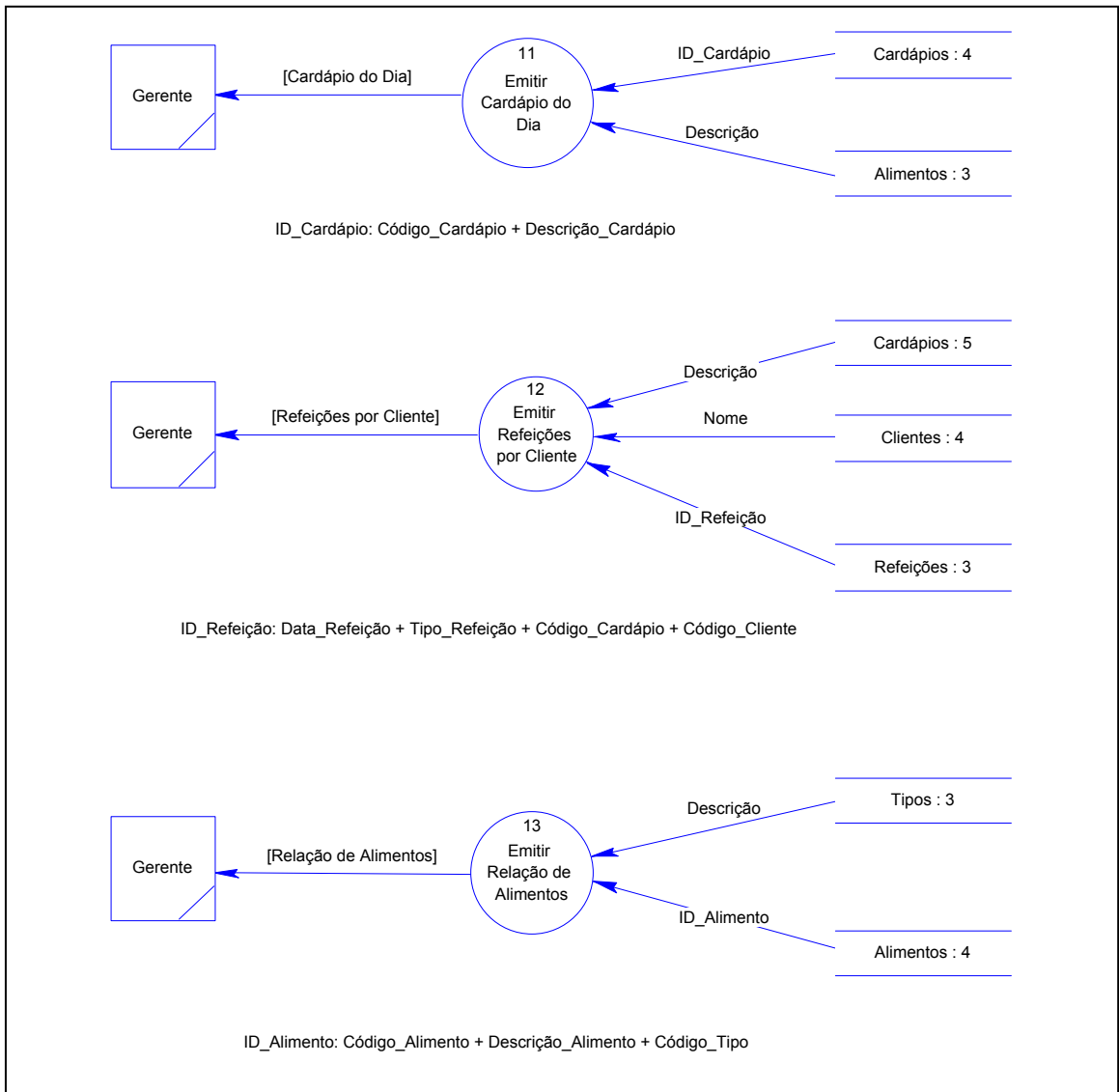


Figura 04: DFD – parte 3



3.4 MODELO ENTIDADE RELACIONAMENTO (MER)

O modelo Entidade Relacionamento Lógico do sistema é apresentado na figura 05. No modelo lógico, são apresentados os nomes dos campos. O no modelo físico apresenta os códigos dos campos conforme ilustrado na figura 06.

Figura 05: MER Lógico

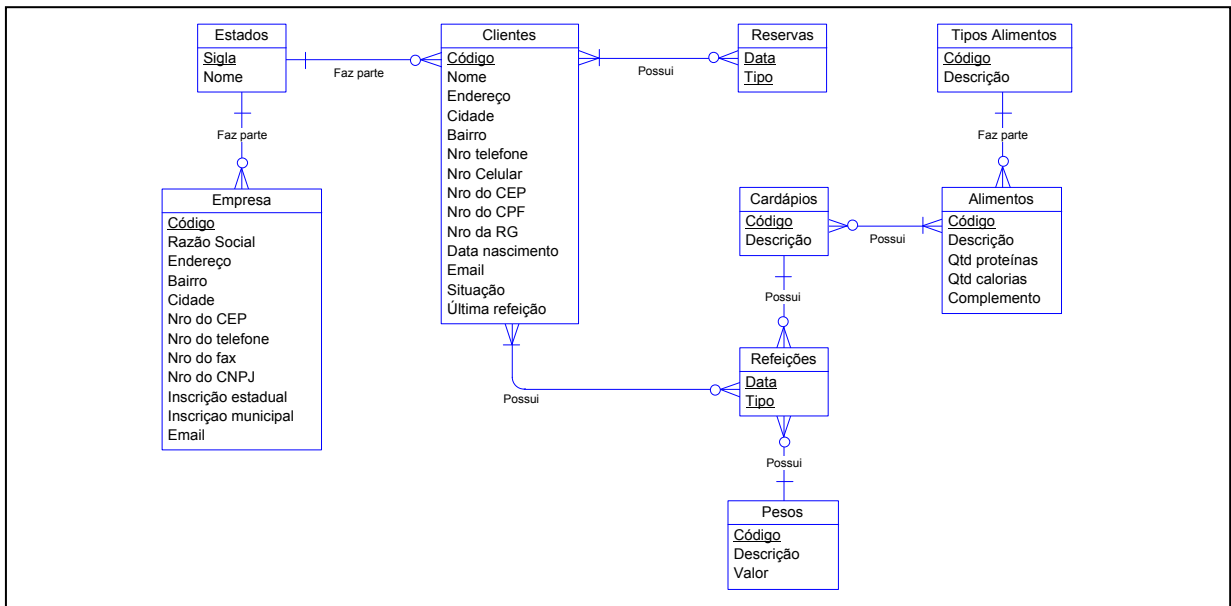
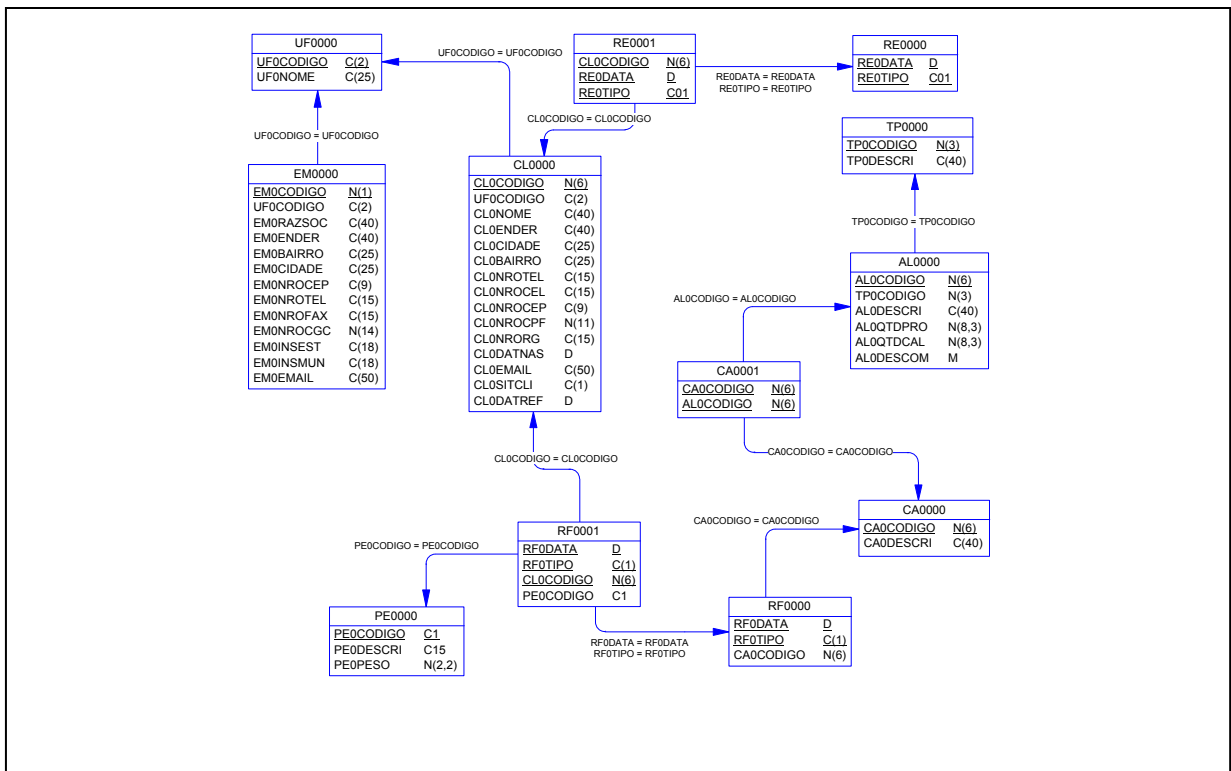


Figura 06: MER Físico



3.5 DICIONÁRIO DE DADOS

O Dicionário de Dados do sistema descreve:

- a) *name*: nome do campo;
- b) *code*: código;
- c) *type*: tipo;
- d) p: campo chave: (PK) chave primária (FK) chave estrangeira (AN) atributo normal;
- e) m: campo obrigatório.

Quadro 1 – Tabela de Clientes

Name	Code	Type	P	M
Código do cliente	CL0Codigo	N6	PK	Yes
Nome do cliente	CL0Nome	C40	AN	Yes
Endereço do cliente	CL0Ender	C40	AN	Yes
Cidade do cliente	CL0Cidade	C25	AN	Yes
Bairro do cliente	CL0Bairro	C25	AN	No
Estado do cliente	UF0Codigo	C02	FK	Yes
Nro do telefone do cliente	CL0NroTel	C15	AN	No
Nro do celular do cliente	CL0NroCel	C15	AN	No
Nro do CEP do cliente	CL0NroCep	C09	AN	No
Nro do CPF do cliente	CL0NroCpf	N11	AN	No
Nro da RG do cliente	CL0NroRg	C15	AN	No
Data de nascimento do cliente	CL0DatNas	D	AN	No
E-mail do cliente	CL0Email	C50	AN	No
Situação do cliente	CL0SitCli	C01	AN	Yes
Data da última refeição	CL0DatRef	D	AN	No

Quadro 2 – Tabela da Empresa

Name	Code	Type	P	M
Código da empresa	EM0Codigo	N01	PK	Yes
Razão social da empresa	EM0RazSoc	C40	AN	Yes
Endereço da empresa	EM0Ender	C40	AN	Yes
Bairro da empresa	EM0Bairro	C25	AN	Yes
Cidade da empresa	EM0Cidade	C25	AN	Yes
Sigla do estado da empresa	UF0Codigo	C02	FK	Yes
Nro do CEP da empresa	EM0NroCep	C09	AN	Yes
Nro do telefone da empresa	EM0NroTel	C15	AN	No
Nro do fax da empresa	EM0NroFax	C15	AN	No
Nro do CNPJ da empresa	EM0NroCgc	N14	AN	Yes
Nro da inscrição estadual da empresa	EM0InsEst	C18	AN	No
Nro da inscrição municipal da empresa	EM0InsMun	C18	AN	No
E-mail da empresa	EM0Email	C50	AN	No

Quadro 3 – Tabela de Estados

Name	Code	Type	P	M
Sigla do estado	UF0Codigo	C02	PK	Yes
Nome do estado	UF0Nome	C25	AN	Yes

Quadro 4 – Tabela de Alimentos

Name	Code	Type	P	M
Código do alimento	AL0Codigo	N06	PK	Yes
Descrição do alimento	AL0Descri	C(40)	AN	Yes
Quantidade de proteínas do alimento	AL0QtdPro	N8.3	AN	No
Quantidade de calorias do alimento	AL0QtdCal	N8.3	AN	No
Tipo do alimento	TP0Codigo	N03	FK	Yes
Descrição complementar do alimento	AL0DesCom	Texto	AN	No

Quadro 5 – Tabela de Tipos de Alimentos

Name	Code	Type	P	M
Código do tipo	TP0Codigo	N03	PK	Yes
Descrição do tipo	TP0Descri	C40	AN	Yes

Quadro 6 – Tabela de Cardápios

Name	Code	Type	P	M
Código do cardápio	CA0Codigo	N06	PK	Yes
Descrição do cardápio	CA0Descri	C40	AN	Yes

Quadro 7 – Tabela de Itens do Cardápio

Name	Code	Type	P	M
Código do cardápio	CA0Codigo	N06	PK	Yes
Código do alimento	AL0Codigo	N06	PK	Yes

Quadro 8– Tabela de Reservas

Name	Code	Type	P	M
Data da reserva	RE0Data	D	PK	Yes
Tipo da reserva	RE0Tipo	C01	PK	Yes

Quadro 09 – Tabela de Itens da Reserva

Name	Code	Type	P	M
Data da reserva	RE0Data	D	PK	Yes
Tipo da reserva	RE0Tipo	C01	PK	Yes
Código do cliente	CL0Codigo	N06	FK	Yes

Quadro 10 – Tabela de Refeições

Name	Code	Type	P	M
Data da refeição	RF0Data	D	PK	Yes
Tipo da refeição	RF0Tipo	C01	PK	Yes
Código do cardápio	CA0Codigo	N06	FK	Yes

Quadro 11 – Tabela de Itens da Refeição

Name	Code	Type	P	M
Data da refeição	RF0Data	D	PK	Yes
Tipo da refeição	RF0Tipo	C01	PK	Yes
Código do cliente	CL0Codigo	N06	FK	Yes
Código do peso	PE0Codigo	C01	FK	Yes

Quadro 12 – Tabela de Pesos

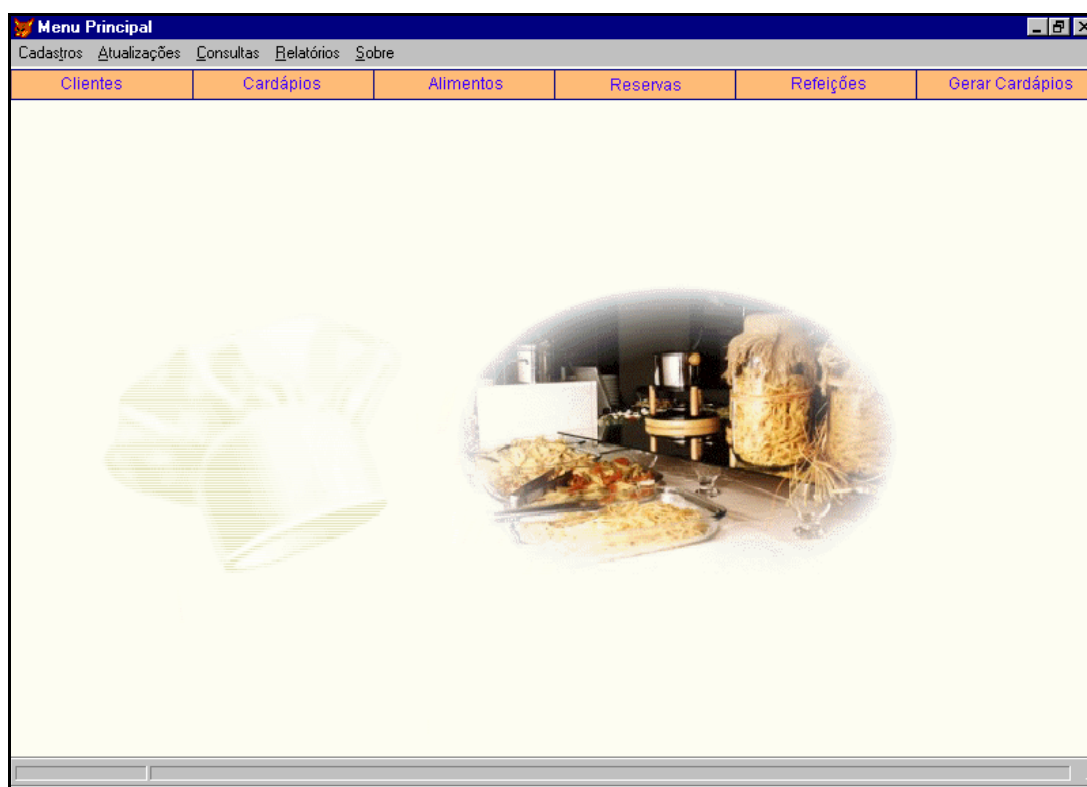
Name	Code	Type	P	M
Código do peso	PE0Codigo	C01	PK	Yes
Descrição do peso	PE0Descri	C15	AN	Yes
Valor do Peso	PE0Peso	N02	AN	Yes

4 IMPLEMENTAÇÃO DO SISTEMA

Neste capítulo, mostra-se a funcionalidade do sistema exemplificando o seu funcionamento e o processo de busca com RBC. Serão mostradas apenas as telas principais do sistema.

Quando o sistema é inicializado, a principal tela que aparece é o menu principal, conforme ilustrado na figura 07. Através do menu principal é possível ter acesso às demais opções do sistema.

Figura 07: Menu Principal



Na figura 08 é possível visualizar a tela de cadastro de clientes. Para que um cliente possa solicitar uma reserva de almoço ou jantar é necessário que ele esteja previamente cadastrado.

Figura 08: Cadastro de Clientes

Cadastro de Clientes

Código: 1

Nome: GILVANI BRUNO KISMER

Endereço: RUA BARAO DO RIO BRANCO, 102

Cidade: RODEIO

Bairro: CENTRO

Estado: SC SANTA CATARINA

CEP: 89136-000

Telefone: (47) 384-0609 Celular:

Identidade: 3/R 3.062.689 CPF: 845,655,439-15

Data Nascimento: 06/01/1974

E-mail: gilvani@mueller.ind.br

Situação: Ativo Última Refeição: 05/05/2003

Botões: Confirmar, Consultar, Fechar, Excluir, ?

Na figura 09 é possível visualizar a tela de cadastro de tipos de alimentos, que serão utilizados futuramente no cadastro de alimentos. O tipo do alimento é a principal fonte de informação para a busca do RBC, pois é através do tipo de alimento que o sistema procura por cardápios similares. O tipo do alimento agiliza o processo de busca, pois a quantidade de alimentos será muito maior que a quantidade de tipos. O tipo do alimento também tem a função de agrupar os alimentos auxiliando no processo de elaboração do pré-cardápio.

Figura 09: Cadastro de tipos de alimentos

Tipos de Alimentos

Código: 3

Descrição: MACARRAO

Botões: Confirmar, Consultar, Fechar, Excluir, ?

Na figura 10 é possível visualizar a tela de cadastro de alimentos, que futuramente serão utilizados para elaborar os cardápios.

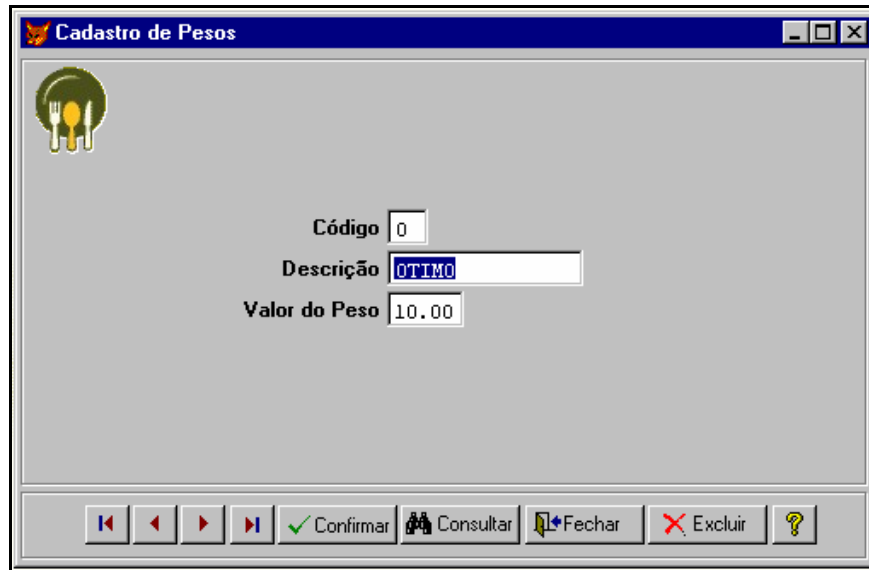
Figura 10: Cadastro de Alimentos

Para que um novo alimento seja cadastrado é necessário que o mesmo possua apenas uma característica diferente dos demais alimentos do seu tipo. Exemplo: macarrão bolonhesa é diferente de macarrão parisiense, ambos podem pertencer ao tipo macarrão, mas a forma de preparar o alimento é diferente.

Na figura 11 é possível visualizar a tela de cadastro de pesos que serão utilizados futuramente ao registrar informações do cliente sobre sua satisfação perante o cardápio servido. Após o cliente ter realizado a sua refeição ou jantar o usuário coleta informações do cliente quanto ao grau de satisfação do cardápio oferecido. O cliente dará sua opinião através de seu voto: Ótimo, Bom, Regular ou Ruim estas opiniões possuem peso (valor) e este peso será utilizado para calcular o grau de satisfação do cardápio. A soma de todos os pesos atribuídos ao cardápio pelos clientes resultará no percentual de satisfação do cardápio.

Percentual de aceitação = ((soma dos pontos do cardápio / (quantidade de clientes da refeição * 10) * 100)).

Figura 11: Cadastro de pesos



Cadastro de Pesos

Código 0

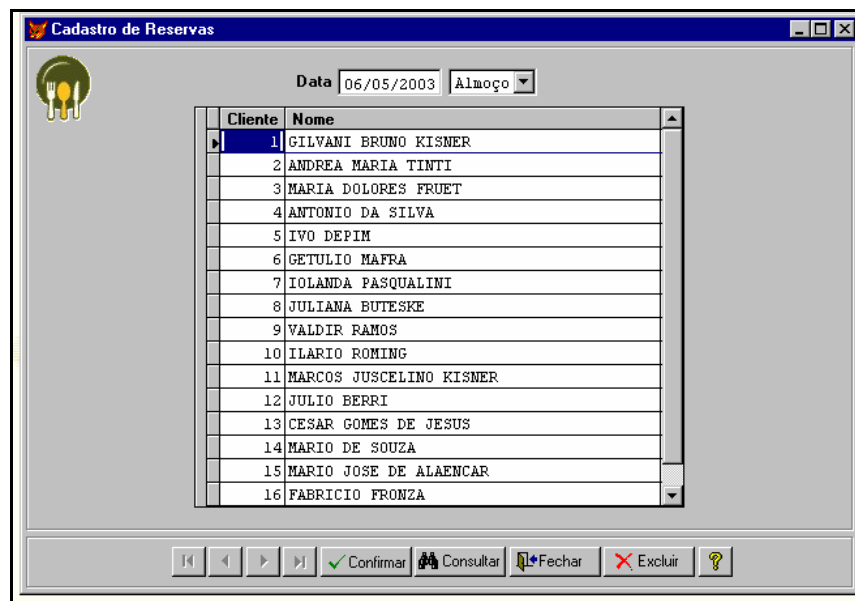
Descrição OTTIMO

Valor do Peso 10.00

Confirmar Consultar Fechar Excluir ?

Na figura 12 é possível visualizar a tela de cadastro das reservas dos clientes que serão utilizados para elaborar o cardápio do dia. Neste cadastro o usuário registrará todos os clientes que efetuarem reserva de almoço ou jantar. Após o término do horário da reserva o usuário pode dar início ao processo de elaboração do cardápio para atender os clientes da reserva. Depois de elaborado o cardápio esta reserva poderá ser impressa e a mesma será excluída no momento em que o usuário efetuar uma nova reserva.

Figura 12: Cadastro de Reservas



Cadastro de Reservas

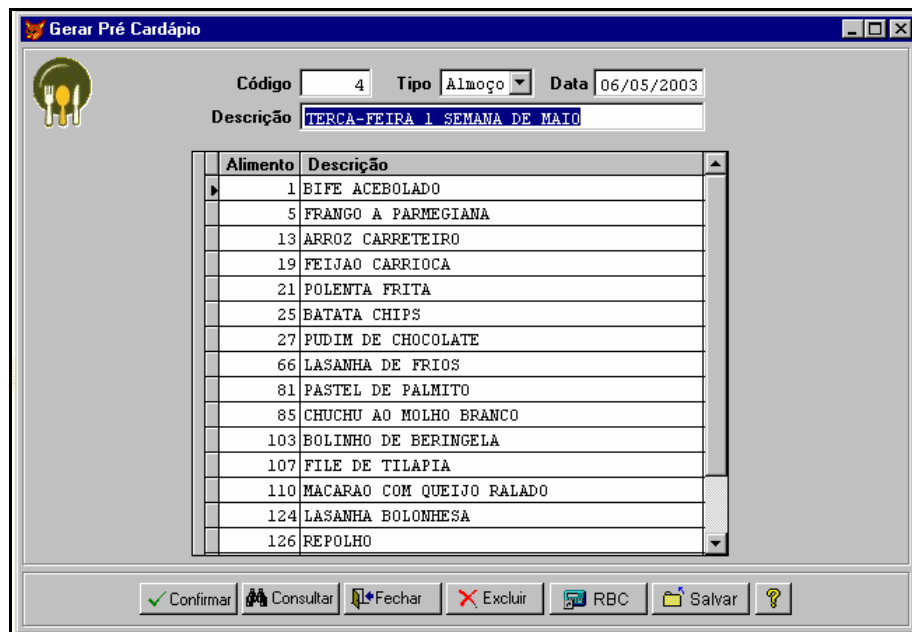
Data 06/05/2003 Almoço

Cliente	Nome
1	GILVANI BRUNO KISMER
2	ANDREA MARIA TINTI
3	MARIA DOLORES FRUET
4	ANTONIO DA SILVA
5	IVO DEPIM
6	GETULIO MAFRA
7	IOLANDA PASQUALINI
8	JULLIANA BUTESKE
9	VALDIR RAMOS
10	ILARIO ROMING
11	MARCOS JUSCELINO KISMER
12	JULIO BERRI
13	CESAR GOMES DE JESUS
14	MARIO DE SOUZA
15	MARIO JOSE DE ALAENCAR
16	FABRICIO FRONZA

Confirmar Consultar Fechar Excluir ?

Na figura 13 é possível visualizar a tela de elaboração do pré-cardápio. Depois de encerrado o tempo das reservas de almoço ou jantar o usuário irá elaborar o pré-cardápio que será servido como almoço ou jantar. O pré-cardápio é um cardápio novo, mas que ainda não faz parte da base de dados dos cardápios. O pré-cardápio só será um novo cardápio através da confirmação do usuário. O sistema utiliza-se do pré-cardápio para fazer as comparações com os cardápios das refeições servidas anteriormente.

Figura 13: Elaboração do pré-cardápio



Na figura 14 é possível visualizar a tela aonde serão coletados os dados para a busca do RBC e cálculo da similaridade. Nesta tela o usuário informará:

- grau de satisfação entre: esta opção faz com que o sistema, analise somente os cardápios das refeições passadas que obtiveram um percentual de aceitação entre o intervalo informado;
- período entre: este parâmetro servirá de índice para a busca de cardápios similares. O sistema analisará somente os cardápios servidos dentro do período informado, não havendo com isso, a necessidade do sistema percorrer toda a base de dados à procura de cardápios similares;
- similaridade maior que: este parâmetro filtrará os cardápios que possuírem uma similaridade maior que a informada, isto evitará a visualização dos cardápios menos similares;

- d) clientes atendidos: a busca por cardápios similares sempre levará em consideração os clientes da reserva, mas nem sempre todos os clientes que efetuaram reserva para almoço ou jantar, serão necessariamente os mesmos clientes que estão na base de casos. Se a quantidade de clientes das refeições da base de casos atingir o percentual de clientes atendidos informado, este cardápio será analisado, caso o contrário será ignorado.

Figura 14: Coleta parâmetros para a busca do RBC

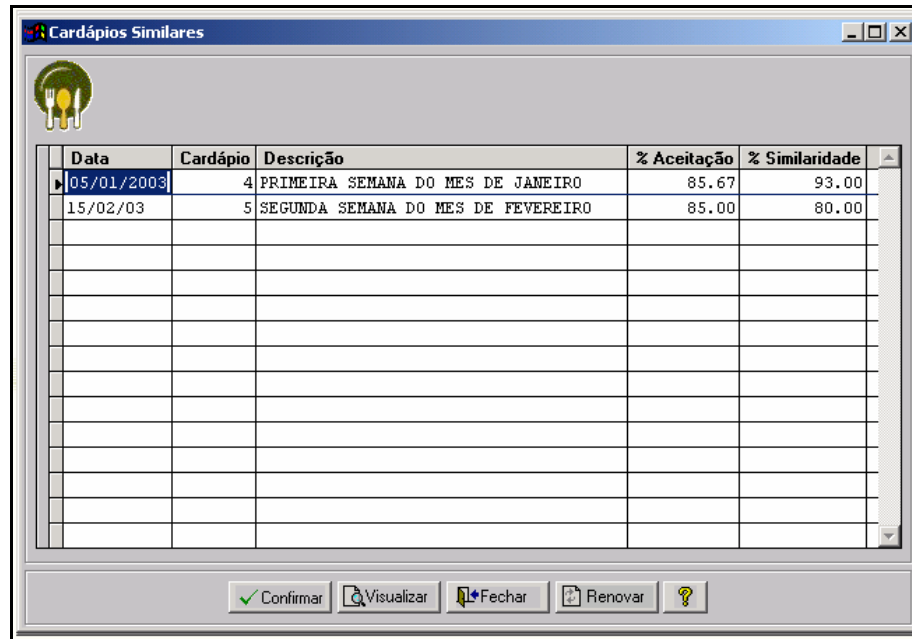
A janela de software, intitulada "Coleta Pesos para o Cálculo da Similaridade", apresenta os seguintes campos de configuração:

- Grau de satisfação entre:** Campo com o valor "70.00 %" e "Até 100.00 %".
- Período entre:** Campo com "De 01/01/2003" e "Até 07/05/2003".
- Similaridade maior que:** Campo com o valor "60.00 %".
- % de Clientes atendidos:** Campo com o valor "70.00 %".

Na base da janela, há três botões: "Confirmar" (com um ícone de checkmark verde), "Fechar" (com um ícone de janela fechando) e um ícone de ajuda (uma interrogação amarela).

Ao ser confirmada a busca por cardápios similares, o sistema retornará a quantidade de cardápios similares. Se o número de cardápios similares for maior que zero será apresentada à visualização dos cardápios em ordem de similaridade seguido do grau de satisfação. Se o número de cardápios similares for igual a zero, isto indica que não foram encontrados nenhum cardápio que atenda aos clientes da reserva dentro dos parâmetros informados, ou este cardápio já foi servido. Através desta consulta, o usuário poderá visualizar os alimentos que foram servidos para cada cardápio. Se o usuário optar pelo pré-cardápio, este cardápio será salvo na base de dados como um novo cardápio. Se o cardápio já existir este cardápio não será incluído. A figura 15 mostra a tela dos cardápios similares.

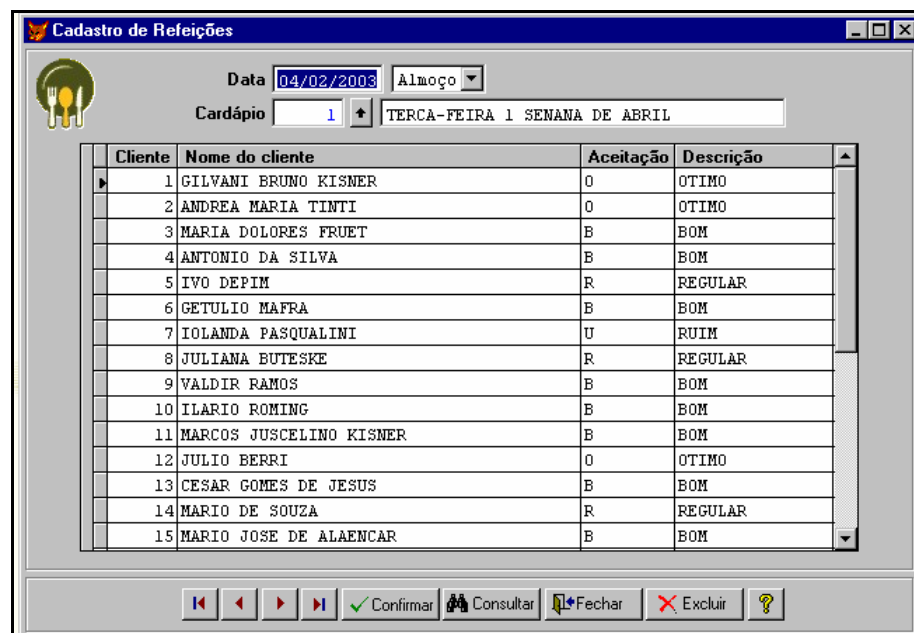
Figura 15: cardápios similares



Data	Cardápio	Descrição	% Aceitação	% Similaridade
05/01/2003	4	PRIMEIRA SEMANA DO MES DE JANEIRO	85.67	93.00
15/02/03	5	SEGUNDA SEMANA DO MES DE FEVEREIRO	85.00	80.00

Depois de estabelecido o cardápio do almoço ou jantar, o usuário começará a registrar os clientes que almoçaram ou jantaram, pois não necessariamente todos os clientes que fizeram sua reserva irão almoçar ou jantar. Ao registrar os clientes, o usuário registrará a aceitação do cliente de acordo com sua opinião perante o cardápio através de seu voto: Ótimo, Bom, Regular ou Ruim.

Figura 16: Cadastro de refeições



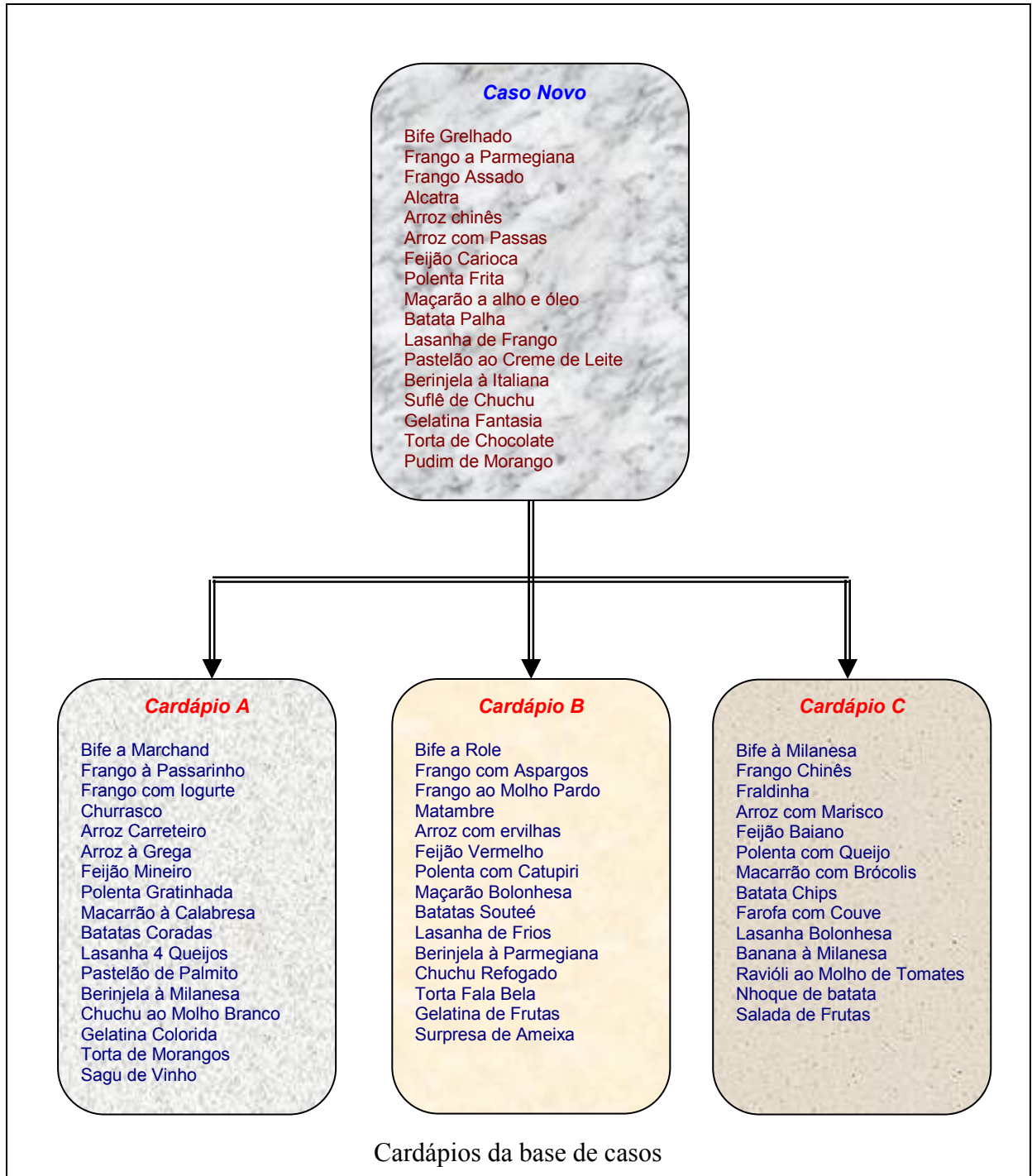
Data: 04/02/2003 Almoço

Cardápio: 1 TERÇA-FEIRA 1 SEMANA DE ABRIL

Ciente	Nome do cliente	Aceitação	Descrição
1	GILVANI BRUNO KISNER	O	OTIMO
2	ANDREA MARIA TINTI	O	OTIMO
3	MARIA DOLORES FRUET	B	BOM
4	ANTONIO DA SILVA	B	BOM
5	IVO DEPIM	R	REGULAR
6	GETULIO MAFRA	B	BOM
7	IOLANDA PASQUALINI	U	RUIM
8	JULIANA BUTESKE	R	REGULAR
9	VALDIR RAMOS	B	BOM
10	ILARIO ROMING	B	BOM
11	MARCOS JUSCELINO KISNER	B	BOM
12	JULIO BERRI	O	OTIMO
13	CESAR GOMES DE JESUS	B	BOM
14	MARIO DE SOUZA	R	REGULAR
15	MARIO JOSE DE ALAENCAR	B	BOM

A figura 17 ilustra o processo de busca com RBC por cardápios similares aonde foram encontrados três casos similares ao caso novo.

Figura 17: Exemplo de busca com RBC por cardápios similares



Atribuindo 1 para os tipos de alimentos coincidentes e 0 para os tipos de alimentos não coincidentes:

Casos \ Tipos	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	X ₁₂	X ₁₃	X ₁₄	X ₁₅
Caso novo => A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
Caso novo => B	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
Caso novo => C	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0

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

$$Sim(casosnovo, A) = \frac{1+1+1+1+1+1+1+1+1+1+1+1+1+1+1}{15} = \frac{14}{15} = 0,93$$

$$Sim(casosnovo, B) = \frac{1+1+1+1+1+1+1+1+1+1+1+1+1+1+1}{15} = \frac{13}{15} = 0,86$$

$$Sim(casosnovo, C) = \frac{1+1+1+1+1+1+1+1+1+1}{15} = \frac{9}{15} = 0,6$$

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

O Sistema procurará por cardápios que possuam alimentos semelhantes, porém do mesmo tipo. Os alimentos iguais serão desconsiderados, pois o principal objetivo é evitar a repetição de cardápios em períodos pré-estabelecidos. Cada cardápio similar terá seu grau de aceitação, nem sempre o cardápio mais similar terá o maior grau de aceitação; fica a critério do usuário se o novo cardápio será servido ou se será alterado, iniciando-se assim um novo processo de busca.

Quadro 13: Linhas de código referente à busca com RBC e cálculo da similaridade

```

&QtdCliRes = 0

//-> Busca a quantidade de clientes da reserva
For each RE0Data RE0Tipo
  Where RE0Data = &Data
  Where RE0Tipo = &Tipo
    For each RE0Data RE0Tipo CL0Codigo
      &QtdCliRes = &QtdCliRes + 1
    Endfor
  Endfor

//-> Busca as refeições no intervalo informado verificando se a qtd de
//-> clientes atende ao % informado
For each RF0Data RF0Tipo
  Where RF0Data >= &DatDe
  Where RF0Data <= &DatAte
  Where RF0Tipo = &Tipo
    &QtdCliRef = 0
    For each RF0Data RF0Tipo CL0Codigo
      For each RE0Data RE0Tipo CL0Codigo
        Where RE0Data = &Data
        Where RE0Tipo = &Tipo
        Where CL0Codigo = CL0Codigo
          &QtdCliRef = &QtdCliRef + 1
        Endfor
      Endfor
    Endfor
  If ((&QtdCliRef / &QtdCliRes ) * 100) >= &PerCli
    //-> Grava arquivo auxiliar dos cardápios que atingiu os percentual de
    //-> clientes
    New
      AX1Data = RF0Data //-> Data
      AX1Tipo = RF0Tipo //-> Tipo
      AX1CodCar = CA0Codigo //-> Código do cardápio
      AX1Descr = CA0Descr //-> Descrição do cardápio
    EndNew
  Endif
Endfor

//-> Seleciona os cardápios que atingiram o percentual de aceitação informado
For each AX1Data AX1Tipo
  For each RF0Data RF0Tipo
    Where RF0Data = AX1Data
    Where RF0Tipo = AX1Tipo
      &QtdPontos = 0
      &QtdCliAte = 0
      For each RF0Data RF0Tipo CL0Codigo
        &QtdPontos = &QtdPontos + PEOPeso
        &QtdCliAte = &QtdCliAte + 1
      Endfor
    Endfor
  &PercAce = ((&QtdPontos / (&QtdCliAte * 10)) * 100)
  If &PercAce <= &GrauDe .OR. &PercAce >= &GrauAte
    Delete
  Else
    AX1PerAce = &PercAce //-> Percentual de aceitação
  Endif
Endfor

```


Quadro 13: Continuação

```

//-> Busca os totais de tipos de alimentos do pré-cardápio
&TipAnt = 0
For each AX0TipAli
  If AX0TipAli <> &TipAnt
    &TotTipAux = &TotTipAux + 1
  Endif
  &TipAnt = AX0TipAli
Endfor

//-> Compara os tipos de alimentos dos cardápios selecionados com os tipos do
cardápio
For each AX1Data AX1Tipo
  &CodCar = AX1CodCar
  &TotTip = 0
  &TipAnt = 0
  For each AX0TipAli
    &TipAli = TP0Codigo
    &CodAli = AL0Codigo
    Do 'Le_Alimento_Cardapio'
      If &Encontrou = 'N'
        If AX0TipAli <> &TipAnt
          Do 'Le_Tipo_Cardapio'
            If &Encontrou = 'S'
              &TotTip = &TotTip + 1
            Endif
          Endif
        Endif
        &TipAnt = AX0TipAli
      Endfor
    Endfor
  //-> Peso da similaridade para os cardápios analisados. Se o total de tipos do
  //-> cardápio da base de tipos for igual ou maior que o total dos tipos do pré-
  //-> cardápio a similaridade é 100% (igual = 1), caso o contrário calcula-se
  //-> a similaridade entre o número de tipos do pré-cardápio com o cardápio da
  //-> base de casos analisado.
  If &TotTip >= &TotTipAux
    AX1PesSim = 1
  Else
    AX1PesSim = round(&TotTip / &TotTipAux, 2)
  Endif
Endfor

//-> Procura por alimentos iguais
Sub 'Le_Alimento_Cardapio'
  &Encontrou = 'N'
  For each CA0Codigo AL0Codigo
    Where CA0Codigo = &CodCar
    Where AL0Codigo = &CodAli
    &Encontrou = 'S'
  Endfor
EndSub

//-> Procura por alimentos diferentes mas com tipos iguais
Sub 'Le_Tipo_Cardapio'
  &Encontrou = 'N'
  For each CA0Codigo AL0Codigo
    Where CA0Codigo = &CodCar
    Where TP0Codigo = &TipAli
    &Encontrou = 'S'
  Endfor
EndSub

```

5 CONCLUSÃO

Este trabalho relatou a aplicação do RBC como ferramenta de apoio na gestão culinária aplicada a restaurantes na elaboração de cardápios. A técnica de RBC mostrou-se eficiente quanto aos objetivos a serem alcançados, assim como os métodos adotados.

É importante ressaltar, que a técnica utilizada para confecção deste trabalho não têm por objetivo a resolução de problemas, mas sim auxiliar os profissionais donos de restaurantes na tomada de decisões no momento da elaboração de novos cardápios.

Um dos pontos fortes alcançados pelo sistema através do RBC foi à busca por cardápios similares através da técnica de similaridade e a prevenção para que não haja a repetição de cardápios nos períodos pré-estabelecidos. Para que se tenham melhores resultados será necessária a inserção de uma quantidade maior de casos. Quanto mais consolidada apresentar-se à base de casos do sistema, mais fiel tornar-se-ão as decisões a serem tomadas.

Neste trabalho não houve a necessidade da adaptação de casos, pois os casos passados devem permanecer inalterados fornecendo subsídios para a criação de novos casos.

5.1 SUGESTÕES PARA TRABALHOS FUTUROS

Disponibilizar o sistema na *WEB* para que os clientes efetuem suas reservas e possam elaborar seus próprios cardápios através dos alimentos oferecidos pelo restaurante. Ao final do término das reservas analisar cada cardápio e, através da utilização de determinada(s) técnica(s) formar um único cardápio com o objetivo de atender todos os clientes de acordo com suas preferências alimentares.

Outra sugestão é adaptar o sistema para atender pessoas portadoras de doenças que exijam cuidados alimentares. Para cada doença específica, estabelecer determinados cardápios. O médico ou nutricionista registrará as características da doença de cada paciente, e o sistema fará sugestões de alimentos que poderão ser consumidos pelo paciente e elaborará um ou mais cardápios.

REFERÊNCIAS

- ABEL, Mara. **Raciocínio baseado em casos**. Porto Alegre: Editora da Universidade Federal do Rio Grande do Sul, 1996.
- ARTech Consultores SRL. **Designing applications with Genexus**. Montevideo: 2002.
- BARRETO, Jorge Muniz. **Inteligência artificial no limiar do século XXI**. Florianópolis: J. M. Barreto, 1997.
- BITTENCOURT, Guilherme. **Inteligência artificial: ferramentas e teorias**. Florianópolis: Editora da Universidade Federal de Santa Catarina, 1998.
- DELPIZZO, Vanessa Lins Francalacci. **Prescrição de atividades físicas através do uso da inteligência artificial**. 1997. Dissertação de Pós Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina, Florianópolis.
- DEMARCO, Tom. **Análise estruturada e especificação de sistema**. Tradução Maria Beatriz Gomes Soares Veiga de Carvalho. Rio de Janeiro: Campus, 1989.
- FAYYAD, U. M.; SHAPIRO, G. P.; UTHURUSAMY, R. **Advances in knowlwdge dicoverly and data mining**. Massachusetts: The MIT Press, 1996.
- FILHO, Moacyr Mallemont Rebello, **A história dos restaurantes**. [s.l], 2000 <<http://www.armazemdesonhos.com.br/colunaMoacyr/historiadosrestaurantes/historiadosrestaurantes.htm>>. Acesso em: 20 de Mar. 2003.
- FISHER, Alan S. **CASE: utilização de ferramentas para desenvolvimento de software**. Tradução Info-Rio. Rio de Janeiro: Campus, 1990.
- HEINRICH, Daniel Jonas. **Sistema de apoio para diagnóstico de defeitos em equipamentos eletrônicos aplicado 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.
- KOLODNER, Janet L. **Case-based reasoning**. Los Altos: Morgan Kaufmann, 1993.

- LEE, Rosina Weber. **Pesquisa Jurisprudencial Inteligente**. Florianópolis, 1998. Tese (Doutorado em engenharia) – Programa de Pós Graduação em Engenharia de Produção PPGEP/UFSC, Universidade Federal de Santa Catarina.
- LEVINE, Robert I. **Inteligência artificial e sistemas especialistas**. São Paulo: McGraw-Hill, 1988.
- LENS, Mario; **Case-based reasoning technology: from foundations to applications**. New York: Springer, 1998.
- MARTIN, James. **Princípios de análise e projeto baseados em objetos**; tradução Cristina Bazan. Rio de Janeiro: Campus, 1994.
- PASSOS, Emmanuel Lopes, **Inteligência artificial e sistemas especialistas ao alcance de todos**. Rio de Janeiro: LTC – Livros Técnicos e Científicos, 1989.
- RABUSKE, Renato Antônio. **Inteligência artificial**. Florianópolis: Editora da Universidade Federal de Santa Catarina, 1998.
- REATEGUI, Eliseu; **Case-based reasoning**: Rio Grande do Sul 1999, disponível em <<http://www.ulbra.tche.br/~iforumia/anais.html>>. Acesso em 31 Mar 2003.
- REIS, Lisiane Albuquerque; CARGNIN, Moema Luz. **SDDEP – uma aplicação na área médica utilizando raciocínio baseado em casos**. Florianópolis: Editora da Universidade Federal de Santa Catarina, 1997.
- RICH, Elaine. **Inteligência artificial**. São Paulo: Makron Books, 1988.
- RODRIGUES, Marcos Roberto; **Reutilização de planos de mensuração utilizando raciocínio baseado em casos**. 1999. 57 f. Monografia apresentada ao Curso de Pós-Graduação em Nível de especialização em Tecnologias de Desenvolvimento de Sistemas, promovido pela Universidade do Contestado – UnC, Campus Universitário de Caçador e Universidade Regional de Blumenau, Caçador.
- SCHILDT, Herbert. **Inteligência artificial utilizando linguagem C**. São Paulo: McGraw-Hill, 1989.
- SCHILLER, Larry. **Excelência em software**. São Paulo: Makron Books, 1992.

SOUZA, André Luiz. **Geração de SQL com PowerDesigner AppModeler**, Uberaba, set. 2002. Disponível em: <<http://xfk.vila.bol.com.br/any/appmod/appmod.htm>>. Acesso em: 24 jun. 2003.

TATIBANA, Cássia Yuri. **Seja bem vindo à homepage de redes neurais**. [s.l], 2003 Disponível em <<http://www.din.uem.br/ia/neurais/>>. Acesso em: 08 de Mar. 2003.

VARELA, Geraldo Menegazzo. **Utilização de raciocínio baseado em casos no sistema para controle e gerenciamento de projetos do Instituto de Pesquisas Ambiental**. 1998. 110f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau.

WINSTON, Patrick Henry. **Inteligência artificial**. Rio de Janeiro: LTC – Livros Técnicos e Científicos, 1988.

YOURDON, Edward. **Análise estruturada moderna**. Tradução Dalton Conde de Alencar. Rio de Janeiro. Campus, 1990.