

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

Protótipo de Sistema Especialista para padronização de modelo de dados baseado no Oracle Designer 2000

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

CLAUDIOMAR KAMKE

BLUMENAU, JUNHO/2001

2001/1-16

Protótipo de Sistema Especialista para padronização de modelo de dados baseado no Oracle Designer 2000

CLAUDIOMAR KAMKE

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

BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

Prof. Wilson Pedro Carli — Orientador na FURB

Prof. José Roque Voltolini da Silva — Coordenador do TCC

BANCA EXAMINADORA

Prof. Wilson Pedro Carli

Prof. Roberto Heinzle

Prof. Marcel Hugo

DEDICATÓRIA

Aos meus pais, que dedicaram suas vidas a mim, para proporcionar-me um futuro melhor e mais digno, que sempre exigiram de mim o máximo de desempenho nos estudos e que nunca deixaram de apoiar-me e incentivar-me.

AGRADECIMENTOS

À Deus, por iluminar meu caminho, fortalecer minhas esperanças, alimentar meus sonhos e ainda por tornar muitos de meus sonhos uma questão de vida.

Aos meus pais, por minha vida, por minhas conquistas e por toda experiência de vida que sempre transmitiram a mim, mesmo nos momentos mais difíceis e dolorosos.

Aos meus irmãos. Por me acompanharem ao longo de meus estudos, auxiliando-me desde soluções dos problemas educacionais como também nos exemplos de vida.

Ao anjo de minha vida, representado neste mundo por minha namorada, que desde o primeiro dia em que nos conhecemos, não exitou em demonstrar a mim o quanto é importante viver, o quanto posso ser útil e principalmente, a razão de acreditarmos em um Deus, um ser que construiu a nós e a este mundo maravilhoso.

A todos os professores e amigos que de alguma forma contribuíram para minha formação.

SUMÁRIO

DEDICATÓRIA.....	I
AGRADECIMENTOS.....	II
SUMÁRIO	III
LISTA DE FIGURAS	V
LISTA DE QUADROS	VI
LISTA DE ANEXOS	VII
RESUMO	VIII
ABSTRACT	IX
1 INTRODUÇÃO	1
1.1 OBJETIVOS DO TRABALHO	3
1.2 ORGANIZAÇÃO DO TRABALHO	3
2 SISTEMAS ESPECIALISTAS.....	5
2.1 INTRODUÇÃO.....	5
2.2 HISTÓRICO DOS SISTEMAS ESPECIALISTAS	5
2.3 CARACTERÍSTICAS DO SISTEMA ESPECIALISTA.....	6
2.4 REPRESENTAÇÃO DO CONHECIMENTO.....	7
2.5 REGRAS DE PRODUÇÃO.....	8
2.6 COMPONENTES	9
2.6.1 BASE DE CONHECIMENTOS.....	9
2.6.2 MECANISMO DE APRENDIZAGEM E AQUISIÇÃO DO CONHECIMENTO	10
2.6.3 MÁQUINA OU MOTOR DE INFERÊNCIA.....	10
2.6.4 SISTEMA DE CONSULTA.....	11
2.6.5 SISTEMA DE JUSTIFICAÇÃO.....	11
2.6.6 QUADRO NEGRO	12
2.7 SHELL.....	12
3 PADRONIZAÇÃO	14
3.1 INTRODUÇÃO.....	14
3.2 CONCEITOS DA PADRONIZAÇÃO	14
3.3 BENEFÍCIOS DA UTILIZAÇÃO DE PADRÕES	15
3.4 DESENVOLVIMENTO DA PADRONIZAÇÃO	16
3.5 MÉTODO DE PADRONIZAÇÃO	17
3.6 CARACTERÍSTICAS BÁSICAS DOS PADRÕES	18
4 FERRAMENTAS	21
4.1 EXPERT SINTA SHELL.....	21
4.1.1 ARQUITETURA DE UM SISTEMA ESPECIALISTA NO EXPERT SINTA SHELL	22
4.1.2 EXPERT SINTA VISUAL COMPONENT LIBRARY (VCL).....	23
4.2 DELPHI.....	24

4.3	FERRAMENTAS CASE	25
4.4	ORACLE DESIGNER 2000	27
	4.4.1 TÉCNICAS DE MODELAGEM.....	27
	4.4.2 ARQUITETURA DO REPOSITÓRIO	29
5	DESENVOLVIMENTO DO TRABALHO	31
	5.1 AQUISIÇÃO DO CONHECIMENTO	31
	5.2 FORMALIZAÇÃO DO CONHECIMENTO.....	33
	5.3 IMPLEMENTAÇÃO.....	34
	5.3.1 TÉCNICAS E FERRAMENTAS UTILIZADAS	34
	5.3.2 OPERACIONALIDADE DA IMPLEMENTAÇÃO.....	36
6	CONCLUSÕES.....	40
	6.1 EXTENSÕES.....	40
7	REFERÊNCIAS BIBLIOGRÁFICAS	41

LISTA DE FIGURAS

FIGURA 1 – FERRAMENTA EXPERT SINTA	21
FIGURA 2 – ARQUITETURA DO EXPERT SINTA.....	22
FIGURA 3 – BARRA DE FERRAMENTAS DO DELPHI COM A BIBLIOTECA (EXPERT SINTA VCL)	24
FIGURA 4 – FLUXOGRAMA DO ORACLE DESIGNER 2000	28
FIGURA 5 – TELA INICIAL DO ORACLE DESIGNER 2000.....	29
FIGURA 6 – DIAGRAMA DE CONTEXTO.....	33
FIGURA 7 – DIAGRAMA DE FLUXO DE DADOS	34
FIGURA 8 – TELA DE LOGIN	36
FIGURA 9 – TELA PRINCIPAL DO APLICATIVO	37
FIGURA 10 – CADASTRO DE TERMOS NO GLOSSÁRIO.....	38
FIGURA 11 – TELA PRINCIPAL DO APLICATIVO COM OS RESULTADOS.....	39

LISTA DE QUADROS

QUADRO 1 – IMPLEMENTAÇÃO DO EVENTO ONPROMPT	35
QUADRO 2 – IMPLEMENTAÇÃO DO EVENTO ONSHOWRESULT.....	35
QUADRO 3 – IMPLEMENTAÇÃO DA CHAMADO DO EXPERT SINTA	36

LISTA DE ANEXOS

ANEXO 1 – LISTA DE VARIÁVEIS	43
ANEXO 2 – LISTA DE VARIÁVEIS E PERGUNTAS.....	45
ANEXO 3 – LISTA DE TABELAS DO REPOSITÓRIO UTILIZADAS NO PROTÓTIPO.....	55

RESUMO

Este trabalho objetiva realizar a validação de modelos de dados construídos na ferramenta Oracle Designer 2000, utilizando-se para tal sistemas especialistas, que através de regras de produção, determinarão as divergências encontradas neste modelo, minimizando assim os problemas e as dificuldades encontradas na padronização destes modelos de dados. O aplicativo foi implementado em ambiente Delphi. Os resultados encontrados demonstram a facilidade de realizar a validação de um modelo de dados, auxiliado por um sistema especialista.

ABSTRACT

The aim of this paper is to show the validation of models of data which were created on Oracle Designer 2000. In order to do that we used specialized systems that, through production rules, will determine the divergences found in this model, minimizing the problems and difficulties as well. The application was implemented at Delphi. The results signal that the validity of a model of data is easier to find when they're supported by a specialized system.

1 INTRODUÇÃO

A necessidade da utilização de novas tecnologias para o desenvolvimento de sistemas computacionais, devido a crescente necessidade tanto das empresas quanto dos profissionais por novos e complexos sistemas, tem demonstrado ao longo dos anos, as limitações das técnicas convencionais.

Esta necessidade incentivou a Oracle, empresa especializada em banco de dados (Abbey, 1997), a desenvolver suas ferramentas capazes de gerenciar e administrar o banco de dados e ainda ferramentas de projeto como o Oracle Designer 2000, ferramenta CASE (*Computer Aided Software Engeneering* – Engenharia de Software Assistida por Computador), que segundo Abbey (1997) “é um conjunto de produtos projetados para ajudar a definir requisitos comerciais, projetar sistemas, gerar entradas de dados e módulos de relatórios, e ainda, uma ferramenta gráfica que demonstra através de diagramas toda a especificação do sistema, ou seja, toda a modelagem de dados do mesmo”. Estes diagramas ou modelos de dados são definidos por Maffeo (1992) como sendo: “uma representação abstrata que permite descrever e/ou prever comportamentos específicos de um sistema, através do estudo de um número reduzido de características relevantes do sistema”.

Um passo importante para o desenvolvimento da tecnologia gerenciada por uma ferramenta CASE é a padronização destes modelos de dados, que pode ocorrer através de um sistema especialista, ou seja, um programa de computador que procura atingir soluções de determinados problemas do mesmo modo que se supõe que os especialistas humanos os resolvam (Lia, 2000). Os mesmos podem utilizar um conhecimento especializado e procedimentos de inferência para resolver problemas de padronização, que normalmente são solucionados por especialistas humanos altamente experientes, que baseados em alguma metodologia, aplicam estes conhecimentos para reduzir os referidos problemas de padronização.

Um sistema especialista é aquele projetado e desenvolvido para atender a uma aplicação determinada e limitada do conhecimento humano que possui capacidade para emitir uma decisão, com apoio em conhecimento justificado, a partir de uma base de informações. Os sistemas especialistas são capazes de aprender, analisar, controlar, interpretar, aconselhar,

consultar, monitorar, instruir, classificar, diagnosticar, predizer, projetar, testar e muitas outras. Além da capacidade de aprender, pode também melhorar o seu desempenho, aprimorar seu raciocínio e aperfeiçoar as suas decisões. Os sistemas especialistas são estruturas para o atendimento de uma aplicação restrita de domínio limitado do conhecimento. Quanto mais restrito for o domínio e mais conhecimento tiver o programa, maior a sua eficiência (Lima, 2000).

O conhecimento em um sistema especialista pode ser representado utilizando-se regras de produção que podem ser consideradas como um par de condição - ação, sendo que o conjunto de regras representa algum conhecimento sobre o assunto abordado no sistema. Para auxiliar na construção desta base de conhecimento, existem várias ferramentas, sendo uma delas, o Expert Sinta que é uma ferramenta computacional que usa técnicas de inteligência artificial para geração automática de sistemas especialistas. Esta ferramenta utiliza um modelo de representação do conhecimento baseado em regras de produção e probabilidade, tendo como principal objetivo simplificar o trabalho de implementação de sistemas especialistas (Lia, 2000).

Desta forma, trabalhando-se atualmente com a ferramenta CASE Oracle Designer 2000, percebe-se que existe uma grande dificuldade entre os analistas de sistemas em padronizar os modelos de dados dos sistemas. Portanto, existe a necessidade de desenvolver uma ferramenta que auxilie na aplicação das especificações de determinadas metodologias de padronização de sistemas, utilizando-se para tal, um sistema especialista, definido por regras de produção. Estas regras são determinadas a partir de um levantamento de informações, baseados na experiência de profissionais ligados a análise de sistemas, bem como nos históricos de problemas encontrados nos padrões de bases e modelos de dados, ou seja, o principal objetivo de se criar um sistema especialista para esta área é facilitar a aplicação de uma metodologia de padronização de sistemas, sem intervenção humana para os casos em que o sistema especialista possuir conhecimento suficiente para corrigir os problemas, minimizando assim estes problemas de padronização.

1.1 OBJETIVOS DO TRABALHO

O objetivo principal do trabalho é desenvolver um sistema especialista capaz de analisar um modelo de dados descrito na ferramenta CASE Oracle Designer 2000 e demonstrar as divergências de padronização encontradas neste modelo de dados.

Os objetivos específicos do trabalho são:

- a) construção da base de conhecimento na ferramenta Expert Sinta;
- b) definição do sistema especialista baseado em regras de produção;
- c) identificação das divergências em um modelo de dados descrito na ferramenta Oracle Designer 2000.

1.2 ORGANIZAÇÃO DO TRABALHO

No capítulo 1, é apresentada a introdução do trabalho, com os objetivos, justificativas e organização do texto.

No capítulo 2, será enfocada uma introdução sobre sistemas especialistas, alguns conceitos e características, bem como seus componentes principais, o histórico da sua evolução, conceito das regras de produção, utilizadas como base do sistema especialista.

No capítulo 3, será apresentada a importância da padronização de sistemas.

No capítulo 4, serão apresentadas as técnicas e ferramentas que servirão de apoio a implementação, fazendo um breve estudo sobre elas para facilitar o desenvolvimento do protótipo de software e torná-lo mais legível.

No capítulo 5, será apresentado o desenvolvimento e implementação do protótipo de software. Para tal, será feita uma especificação do trabalho através de um problema descrito, utilizando uma ferramenta PowerDesigner para facilitar a posterior implementação, na qual será apresentado o desenvolvimento do sistema especialista utilizando a ferramenta Expert Sinta Shell e a interface com o usuário desenvolvida em Delphi.

O capítulo 6 relata as conclusões, considerações finais e extensões para o aprimoramento do trabalho.

2 SISTEMAS ESPECIALISTAS

2.1 INTRODUÇÃO

Segundo Lia (2000), sistemas especialistas são programas de computador que procuram atingir soluções de determinados problemas do mesmo modo que especialistas humanos, se estiverem sob as mesmas condições. Apesar das limitações das máquinas, é possível, hoje, a construção de sistemas especialistas com alto grau de desempenho, dependendo da complexidade de sua estrutura e do grau de abrangência desejado.

Para Genaro (1986), sistemas especialistas são programas sofisticados que manipulam conhecimento para resolução de problemas em áreas restritas de um dado domínio. Da mesma forma que peritos humanos, tais sistemas se utilizam de lógica simbólica e regras oriundas da prática para encontrar as soluções. E, ainda como especialistas reais, podem errar, mas aprendem com os erros.

2.2 HISTÓRICO DOS SISTEMAS ESPECIALISTAS

No início da década de 1960 começaram os primeiros trabalhos nos sistemas que hoje são chamados de especialistas. Inicialmente pretendia-se construir máquinas inteligentes com grande poder de raciocínio e solução de problemas. Imaginava-se que a partir de um pequeno conjunto de normas ou regras de raciocínio introduzidas num poderoso computador criariam-se sistemas de capacidade superior a humana. (Heinzle, 1995).

Em 1964 foi construído o DENDRAL, por Joshua Lederberg da Universidade de Stanford. O DENDRAL a partir de um determinado conjunto de dados como massa espectrográfica e ressonância magnética, deduz a possível estrutura de um determinado composto químico. Este programa era do tipo algorítmico. Em 1965, Josima juntou-se à Edward Feigenbaum e Bruce Buchanan para tentar construir um programa que não fosse algorítmico e que usasse regras heurísticas para resolver os mesmos problemas do DENDRAL. Este novo DENDRAL mostrou a viabilidade dos sistemas especialistas e levou pesquisadores de outras universidades a trabalharem no assunto (Heinzle, 1995).

Em 1968, surge no *Massachusetts Institute of Technology* - MIT, o MACSYMA,

destinado a auxiliar matemáticos na resolução de problemas complexos. O programa foi originalmente elaborado por Carol Englemari, Wiliam Martin e Joel Mores. MACSYMA é um sistema ainda hoje amplamente utilizado em universidades e laboratórios de pesquisa (Heinzle, 1995).

Nos meados dos anos 70, havia grande interesse em criar funções heurísticas para a contagem de pontos que de certa forma, seriam análogas às probabilidades, embora sem estarem sujeitas às limitações matemáticas humanas. Diversos esquemas foram cuidadosamente analisados, tais como as funções para contagem de pontos do MYCIN e do PROSPECTOR. Estas funções são aproximações heurísticas que não são probabilisticamente exatas, mas que estão relacionadas com determinadas outras imposições ou hipóteses sobre o raciocínio. Se as hipóteses não forem confirmadas, então grandes imprecisões podem ocorrer. Contudo, o conhecimento para muitas aplicações pode ser representado sem o uso de medidas do grau de incerteza (Weiss, 1988).

A década de 1980 foi marcada pelo grande crescimento de aplicações, inclusive, com larga disponibilização de produtos comerciais no mercado de software. Este acelerado processo de desenvolvimento de aplicações deve-se ao avanço dos recursos de equipamentos, ou hardware, ocorrida paralelamente neste período. Entre os produtos surgidos nesta década poderiam ser destacados os das área médica e matemática.

No Brasil, a Universidade Católica do Rio de Janeiro desenvolveu, nos anos 80, importantes trabalhos com sistemas especialistas. O principal resultado da universidade nesta área é um sistema chamado SAFO cuja finalidade é a demonstração de teoremas matemáticos. Outra referência histórica no Brasil, é o Instituto Militar de Engenharia que há alguns anos vem desenvolvendo sistemas de recuperação em grandes bases de conhecimentos. Quanto aos produtos comerciais brasileiros vale citar o surgimento, também nos anos 80, do sistema PATER. Este produto é um software para a construção de sistemas especialistas de caráter geral e foi projetado para utilização em microcomputadores (Heinzle, 1995).

2.3 CARACTERÍSTICAS DO SISTEMA ESPECIALISTA

Um sistema dito tradicional, normalmente é estabelecido para resolver um problema bem especificado que pode ser colocado de forma algoritmizada, isto é, pode-se determinar

uma série de etapas finais para se resolver esse problema, em um tempo considerado aceitável para a obtenção da resposta. Desenvolvido para a solução de um problema bem determinado, um sistema tradicional é projetado para sempre terminar emitindo um resultado final correto, isto é, sistemas tradicionais já são concebidos para apresentar uma solução correta. Esse sistema também processa dados, isto é, os algoritmos são projetados para processar volumes de dados de maneira repetitiva (Weiss, 1988).

Para entender quais são as características comuns aos sistemas especialistas, Genaro (1986) explica o que estes fazem:

- a) resolvem problemas muito complexos tão bem quanto ou melhor que especialistas humanos;
- b) raciocinam heurísticamente, usando o que os pontos consideram efetivamente regras práticas;
- c) interagem com usuários humanos utilizando inclusive linguagem natural;
- d) manipulam e raciocinam sobre descrições simbólicas;
- e) funcionam com dados errados e regras incertas de julgamento;
- f) contemplam hipóteses múltiplas simultaneamente;
- g) explicam por que estão fazendo determinada pergunta;
- h) justificam suas conclusões.

2.4 REPRESENTAÇÃO DO CONHECIMENTO

Um sistema especialista necessita de grande quantidade de conhecimento (fatos implícitos e explícitos) e algum mecanismo para manipulá-lo (regras de inferência), permitindo assim a solução de novos problemas. Uma forma bem comum de representar os fatos é através de sentenças em linguagem natural. Para permitir comunicação de dados com o programa é necessário um relacionamento entre as formas internas de representar os fatos e sua expressão em linguagem natural. Entre os formalismos existentes para representação de

fatos, a lógica aparece como opção atraente, dado que imediatamente sugere um poderoso mecanismo de derivar novos fatos dos antigos, a dedução matemática. Neste formalismo, conclui-se que uma nova sentença é verdadeira caso seja conseqüência de outras já estabelecidas. Então a idéia de prova, da forma que é utilizada em matemática, como um rigoroso meio de demonstrar a verdade de uma suposição, pode ser estendida para derivar respostas a perguntas e soluções de problemas (Genaro, 1986).

Um sistema especialista é aquele que possui regras e permite pesquisa exaustiva, com boa eficiência, raciocina por manipulação de símbolos, entende os princípios fundamentais do domínio e tem competência para retornar quando as regras falham e para usá-las na produção de explanações. Procede com problemas difíceis em um domínio complexo, pode tornar uma descrição de problema em termos de leis e convertê-las em representação interna apropriada para processamento com suas regras do conhecimento e pode raciocinar sobre seu próprio conhecimento, especialmente para reconstruir caminhos de inferência racionalmente para explicação ou auto - justificação (Genaro, 1986).

Num sistema baseado em regras de produção o conhecimento a ser processado é representado através do uso de regras com uma arquitetura previamente definida. Estas regras utilizam um par condição - ação onde as condições são premissas e a ação é a conclusão (Heinzle, 1995).

2.5 REGRAS DE PRODUÇÃO

No processo de inferência o sistema busca uma primeira regra arbitrariamente, ou em alguns casos aquela predefinida como regra inicial, e tenta atender as premissas da regra. As premissas descritas na regra são apresentadas ao usuário em forma de questionamentos. As respostas fornecidas pelo usuário são então armazenadas na lista de verdades fazendo com que estas informações fiquem disponíveis ao longo do processo de raciocínio e possam ser utilizadas para a validação de suas regras. Se as respostas fornecidas pelo usuário atenderem as premissas da regra e a regra contiver na sua parte conclusiva uma solução para o problema, o processo de inferência estará concluído com sucesso. Se, por outro lado, a regra não permitiu alcançar uma solução para o problema, o sistema seguirá avaliando outras regras, sempre acumulando o conhecimento ao longo do processo na sua lista de verdades. O processo continua até que seja alcançada uma regra que leve à solução do problema, ou

quando não for mais continuá-lo (Heinzle, 1995).

2.6 COMPONENTES

A composição de um sistema especialista depende de fatores como a generalidade pretendida, os objetivos do mesmo, a representação interna do conhecimento e as ferramentas usadas na implementação. Também na terminologia empregada há diferenças entre os autores, mas de uma forma geral o sistema é constituído de seis elementos básicos que são: base de conhecimentos, mecanismo de aprendizagem e aquisição de conhecimento, máquina ou motor de inferência, sistema de justificação, sistema de consulta e quadro negro (Heinzle, 1995).

2.6.1 BASE DE CONHECIMENTOS

Os motivos gerais para a criação de um sistema especialista aplicam-se a qualquer sistema deste tipo, porém, quando é construído um modelo específico de raciocínio, os motivos têm que ser muito mais precisos. Cria-se um modelo especialista para solucionar uma classe específica de problemas e fornecer recomendações sobre as respectivas soluções. Um especialista humano tem muitos tipos diferentes de informações para fornecer ao criador de um modelo de raciocínio especialista (Weiss,1988), incluindo:

- a) experiência pessoal, anterior, na solução de problemas;
- b) perícia pessoal ou métodos para a solução de problemas;
- c) conhecimento pessoal sobre as razões da seleção dos métodos utilizados.

A base de conhecimentos é o local onde se armazenam fatos, heurísticas, etc, ou seja, é um depósito de conhecimentos acerca de um determinado assunto. Este conhecimento é passado ao sistema pelo especialista e armazenado de forma própria que permite ao sistema fazer posteriormente o processamento. A qualidade deste conhecimento armazenado é o fator determinante no potencial do sistema especialista. O sistema deve possuir uma base de conhecimentos o mais flexível possível de ser atualizada, e também poder fazer aprendizagem direta durante as consultas (Heinzle, 1995).

A fase de construção da base de conhecimentos é uma das mais complexas na

implementação de um sistema especialista pois o conhecimento de um especialista não se encontra formalizado, precisando portanto de um trabalho prévio para tal. A base de conhecimentos está interligada com quase todos os demais elementos do sistema, especialmente com a máquina de inferência, o mecanismo de aprendizagem e aquisição do conhecimento e o quadro negro (Heinzle, 1995).

2.6.2 MECANISMO DE APRENDIZAGEM E AQUISIÇÃO DO CONHECIMENTO

Segundo Rabuske (1995), “obter o conhecimento é a parte mais complexa da construção de um sistema especialista. A aquisição do conhecimento tende a caracterizar áreas de pesquisa específicas nas universidades e centros de pesquisas, geralmente ligadas à engenharia do conhecimento”.

Para Levine (1988), “a fase de aquisição do conhecimento é a que representa maior dificuldade na construção de um sistema especialista. Esta dificuldade advém do fato de não existir uma linguagem comum de entendimento entre as partes envolvidas no projeto”. O especialista não tem suas idéias organizadas utilizando processos indutivos e dedutivos na obtenção das soluções.

2.6.3 MÁQUINA OU MOTOR DE INFERÊNCIA

As informações armazenadas numa base de conhecimentos são, evidentemente, estáticas até que uma força externa analise e processe este conhecimento para dele tirar proveito. Este mecanismo que permite “tirar o proveito” de uma base de conhecimentos é chamado de motor, máquina, ou ainda, engenho de inferência. O motor de inferência é o elemento do sistema especialista que é capaz de buscar na base o conhecimento necessário a ser avaliado em cada situação, direcionar o processo de raciocínio, gerenciar situações de incerteza e levar ao resultado final (Heinzle, 1995).

O processo de inferência está diretamente associado com a estrutura utilizada para o armazenamento do conhecimento na base de conhecimentos. Entretanto, de forma geral, pode-se afirmar que o processo envolve um encadeamento lógico que permita tirar conclusões a partir do conhecimento existente. O motor de inferência é, portanto, o responsável pela ação

repetitiva de buscar, analisar e gerar novos conhecimentos. Este elemento não é, geralmente, um módulo único no sistema. Pois na verdade são vários processos que devem trabalhar de forma conjunta, tais como busca do momento relevante num determinado contexto, resolução de conflitos, de incertezas e finalmente a execução propriamente dita (Heinzle, 1995).

2.6.4 SISTEMA DE CONSULTA

Os usuários de sistemas especialistas interagem de forma intensa com o sistema pois além de receberem dele as conclusões alcançadas, também, participam ativamente do processo de inferência e da construção da base de conhecimentos. Estes sistemas devem portanto oferecer bons recursos de comunicação que permitam, até ao usuário sem conhecimentos computacionais, tirar proveito dos mesmos. Aspectos internos dos sistemas, terminologia computacional devem ser evitados e detalhes técnicos relativos a implementação devem ser transparentes ao usuário. A linguagem a ser utilizada deve ser orientada para o problema ou para a área do especialista e o mais perto possível da linguagem natural.

A maioria dos sistemas existentes usam técnicas simples de interação com o usuário, quase sempre utilizando perguntas já pré formatadas e respostas tipo múltipla escolha. Outra técnica é a definição de uma gramática sintética simples com um vocabulário restrito e limitado, própria para utilização no sistema. Esta tecnologia é todavia um outro campo de estudo da inteligência artificial cujo desenvolvimento será de extrema valia para toda a área da computação (Heinzle, 1995).

2.6.5 SISTEMA DE JUSTIFICAÇÃO

O módulo de justificação, também chamado de explanação, tem a intenção de esclarecer o usuário a respeito de uma conclusão apresentada pelo sistema ou ainda explicar por que uma pergunta está sendo feita. Ele é na verdade um recurso de questionamentos fornecido ao usuário (Heinzle, 1995).

A justificação é um requisito importante dos sistemas especialistas. Em muitos dos domínios nos quais os sistemas operam, as pessoas não aceitam resultados se estes não estiverem devidamente justificados. Na medicina, por exemplo, onde um médico tem a responsabilidade final por um diagnóstico, certamente um sistema teria que mostrar os

motivos que o levaram a alcançar uma determinada conclusão. De forma geral, os sistemas são implementados para responder as seguintes perguntas:

- a) como chegou a esta conclusão?;
- b) por que chegou a esta conclusão?;
- c) por que não chegou a tal conclusão?.

2.6.6 QUADRO NEGRO

O quadro-negro, também denominado rascunho, é uma área da memória onde o sistema vai gravando e apagando os dados que vai usando, no processo de inferência, até chegar a uma solução. Para se chegar a uma solução, há necessidade de se avaliarem regras que são recuperadas da base de conhecimento para uma área de trabalho na memória, nesse local essas regras são ordenadas periodicamente em uma nova ordem para serem avaliadas. Durante essa avaliação deve-se verificar fatos e hipóteses e também há necessidade de uma área onde são guardados os valores das variáveis para se trabalhar tais fatos e hipóteses. As conclusões dessas regras irão gerar novos fatos e novas hipóteses que precisam ser guardados (gravados) temporariamente durante o processo de inferência (dedução), em algum local. Essa área na memória usada para a execução das operações descritas acima segundo Ribeiro (1987) chama-se quadro-negro ou rascunho.

Para Heinzle (1995), o quadro negro, ou rascunho, é uma área de trabalho que o sistema utiliza durante o processo de inferência. Nesta área são armazenadas informações de apoio e suporte ao funcionamento do sistema quando este está raciocinando. Embora todos os sistemas especialistas usem o quadro negro, nem todos o explicitam como do sistema.

2.7 SHELL

Para Lia (2000), “a principal função de uma *shell* é simplificar ao máximo o trabalho de implementação de um sistema especialista e permitir seu uso por qualquer outra pessoa sem conhecimento de informática”.

Segundo Rabuske (1995), “*shells* é o nome dado a uma família de ferramentas, não linguagens de programação, que objetivam apoiar e simplificar o processo de construção de sistemas especialistas”. São softwares que contêm alguns dos principais elementos de um sistema especialista, tais como motor de inferência, o justificador e outros. Estas ferramentas também pré definem a estruturação do conhecimento a ser utilizada pelo sistema. Ao projetista do sistema especialista usuário de uma *shell* cabe apenas a tarefa de construir a base de conhecimentos.

3 PADRONIZAÇÃO

3.1 INTRODUÇÃO

Nas empresas modernas a padronização é considerada a mais fundamental das ferramentas gerenciais. Na qualidade total a padronização é a base para a rotina (gerenciamento da rotina do trabalho diário). No entanto, a experiência tem demonstrado que a situação brasileira no tocante à padronização não é boa, pois falta literatura, educação e treinamento das pessoas que ocupam cargos de chefia. Nas raras empresas que se consideram padronizadas, este assunto tem sido relegado aos técnicos, quando de fato é um encargo essencialmente gerencial. As pessoas que ocupam cargos gerenciais precisam entender que a padronização é o caminho seguro para a produtividade e competitividade ao nível internacional, pois é uma das bases onde se assenta o moderno gerenciamento (Campos, 1992).

3.2 CONCEITOS DA PADRONIZAÇÃO

O ser humano convive com a padronização há milhares de anos e dela depende para sua sobrevivência mesmo que disto não tenha consciência. Imagine uma pequena tribo ou aldeia no passado onde a alimentação básica era o peixe. Pescava-se de alguma forma até que alguém testou uma rede feita de cipós e pegou uma quantidade maior de peixes com menor trabalho. Evidentemente que os outros habitantes da aldeia, tendo em vista os resultados obtidos, passaram a utilizar a rede como método de pesca. Estava assim padronizado o método de pescar com rede. Mais tarde alguém julgou que seria melhor utilizar fios de juta do que cipó para fazer a rede. Tentou e isto resultou numa maior quantidade de peixes com menor trabalho. Os outros imediatamente adotaram a idéia (padronizaram). De acordo com Campos (1992), algumas observações são importantes para serem comentadas:

- a) ninguém era obrigado a padronizar o método da pesca, fizeram isto somente porque dava melhor resultado;
- b) a padronização é meio, o objetivo é conseguir melhores resultados;
- c) o método padronizado não é fixo, ele pode e deve ser melhorado para a obtenção de melhores resultados;

- d) originalmente não havia a necessidade de se registrar o método padronizado. A aldeia era pequena e todos aprendiam o novo método naturalmente. A memória da aldeia era a memória das pessoas. Hoje a sociedade é complexa e para garantir a padronização é necessário registrar de forma organizada (em papel ou memória de computador) e conduzir formalmente o treinamento no trabalho.

Não há muito sentido em se estabelecerem padrões, a menos que se respeite a sua necessidade e que sejam considerados fundamentais e perceptíveis. Os padrões, para serem utilizados consistentemente, devem ser totalmente aceitos em sua forma geral (Longworth, 1985).

A palavra "Padrões" tende a ser julgada emotivamente, como uma restrição de liberdade. Porém, os padrões são apenas a essência de experiência sobre o que constitui uma boa prática de programação. Estabelecem uma fonte de referência baseada no aprendizado, poupando assim tempo e aborrecimentos para outros no futuro (Longworth, 1985).

3.3 BENEFÍCIOS DA UTILIZAÇÃO DE PADRÕES

A programação precisa ser disciplinada, caso contrário a qualidade, o planejamento, medidas e controle não existirão consistentemente. Os padrões proporcionam essa disciplina, guiando o trabalho da equipe, estabelecendo "*checkpoints*" e níveis de desempenho, ajudando no controle efetivo e corrente dos ativos da empresa, auxiliando pessoas inexperientes a se tornarem eficientes com mais rapidez e fazendo a comunicação mais fácil para os muitos indivíduos que trabalham em um projeto ao mesmo tempo. Facilitam, também, o entendimento do programa para aqueles que fazem parte da manutenção de sistemas e, conseqüentemente, permitem uma aplicação mais rápida, mais precisa e mais completa de alterações (Longworth, 1985).

Segundo Campos (1992), a utilização de padronização representa:

- a) um ciclo de manutenção cujo objetivo é a previsibilidade dos resultados, para isto, no ciclo de manutenção, deve-se cumprir os padrões e atirar no resultado e nas causas dos desvios, quando indicado no procedimento operacional. Este papel é cumprido principalmente pelos operadores (secretárias, torneiros, compradores, operadores de ponte rolante, professores);

- b) um ciclo de melhorias cujo objetivo é a competitividade da empresa, conseguida mediante a melhoria contínua dos resultados. As melhorias são conseguidas pela análise do processo e adoção de novo padrão (método de solução de problemas). Este papel é cumprido principalmente pelas chefias, pelos operadores que participam do ciclo das melhorias por meio dos círculos de controle da qualidade e supervisores.

3.4 DESENVOLVIMENTO DA PADRONIZAÇÃO

Antes que um grupo de garantia de qualidade possa funcionar efetivamente, o processo de desenvolvimento e manutenção do código deve ser padronizado. De outra forma, o controle de qualidade se deteriora em batalhas de opinião sobre o que é e o que não é qualidade. Os padrões eliminam estas discussões e simplificam a tarefa. Padrões apropriados não restringem o programador ou o analista; aprimoram sua interação com os outros e selecionam a melhor forma de se codificar ou desenhar, e encorajar a sua utilização. Pode ser debatida a melhor maneira, mas uma deverá ser selecionada e escolhida. Sem esta maneira comum de trabalhar, torna-se difícil manter verificações significativas, inspeções de código ou auditoria de qualidade (Arthur, 1985).

A organização para a padronização, irá variar dependendo do tamanho e tipo da empresa. No entanto, conforme Campos (1992), três aspectos fundamentais devem ser observados:

- a) a padronização é responsabilidade da mais alta autoridade da empresa;
- b) deve ser organizado um sistema de padronização da empresa;
- c) as funções do sistema de padronização devem ser gerenciadas por alguma organização interna.

O primeiro passo no estabelecimento da padronização na empresa é estabelecer um "Comitê de Direção da Padronização". Este comitê pode ser subcomitê de implantação da qualidade total e ficar diretamente ligado ao presidente, as funções deste comitê segundo Campos (1992) são as seguintes:

- a) avaliar e aprovar um plano de implantação e programa para planejamento, estabelecimento, disseminação e atualização dos padrões de sistemas; avaliação da necessidade dos padrões já estabelecidos; estabelecimento de programa de redução do número de reclamações; estabelecimento de programa para solução de problemas críticos; estabelecimento de padrões de qualidade nas organizações de produção; preparação dos padrões de qualidade do projeto; estabelecimento dos padrões de avaliação da qualidade e padrões de auditoria;
- b) estabelecimento de padrões de avaliação do sistema de garantia da qualidade para toda empresa;
- c) avaliar e aprovar o estabelecimento, revisão ou rescisão dos procedimentos de padronização;
- d) determinar critérios de prioridade para o estabelecimento, revisão ou rescisão de todos os padrões.

3.5 MÉTODO DE PADRONIZAÇÃO

Para Campos (1992), jamais se estabelece um padrão sem que haja um objetivo definido (qualidade, custo, atendimento, moral e segurança) e a consciência de sua necessidade. A padronização deve seguir as seguintes etapas básicas:

- a) especialização: escolher o sistema a ser padronizado determinando a sua repetibilidade. Por exemplo: o sistema de compras (o que é repetitivo neste sistema, embora cada compra possa ser diferente?); o sistema de desenvolvimento de novos produtos pode parecer inicialmente que não é repetitivo, pois a cada vez se desenvolve um produto diferente. No entanto, se formos observar com cuidado, iremos verificar que existe uma série de procedimentos básicos que são repetitivos e, portanto, padronizáveis. O mesmo poderia-se dizer do planejamento e do gerenciamento pelas diretrizes;
- b) simplificação: uma vez delimitada a repetibilidade e definido o sistema (processo), o próximo passo é a simplificação, que consta da redução do número de produtos, componentes, materiais e procedimentos e da simplificação do

projeto dos produtos (visando a reduzir custos);

- c) redação: redigir numa linguagem que as pessoas entendam, inclusive gíria e linguagem coloquial local;
- d) comunicação: comunicar e consensar com todas as outras pessoas ou departamentos afetados pelo padrão;
- e) educação e treinamento: o objetivo da padronização é conseguir que as pessoas façam exatamente aquilo que tem de ser feito e sempre da mesma maneira. O alvo principal é a mente das pessoas. O objetivo é fazer com que cada um seja "o mais competente do mundo em sua função";
- f) verificação da conformidade aos padrões: este é o principal papel de todas as chefias. O gerente supervisiona o sistema e o aperfeiçoa. O supervisor audita o trabalho do operador e o ensina. As metas da qualidade, custo, atendimento, moral e segurança devem ser alcançadas.

3.6 CARACTERÍSTICAS BÁSICAS DOS PADRÕES

Os padrões não devem restringir o trabalho desnecessariamente, devem, entretanto padronizar o que é feito. Se houver duas maneiras igualmente boas de executar uma tarefa de programação ou de desenho, os padrões deverão indicar qual a melhor maneira a ser usada (Arthur , 1985).

Os modelos de padronização poderão variar de acordo com Campos (1992), de empresa para empresa em função do tipo, tamanho e das condições locais. No entanto, alguns aspectos básicos devem ser observados:

- a) sempre que for redigido um padrão pergunte: quem é o usuário? A padronização é conduzida para que os padrões sejam utilizados. Padrões expostos em arquivos não apresentam valor prático. Sob este aspecto, existe muita incompreensão no Brasil. Pensa-se que utilizar o padrão é carregá-lo no bolso (costuma-se observar até padrões reduzidos e plastificados);
- b) sempre que for redigido um padrão, pergunte: este documento está na forma mais

simples possível? O padrão deve ter o menor número de palavras possível e ser colocado em forma simples, sem prolixidade. Aqueles padrões de várias páginas que descrevem o trabalho de várias pessoas são de difícil acesso e utilização;

- c) o padrão pode ser cumprido? Padrões que não equivalem à situação atual são inúteis. Por exemplo: valores padrão podem ter sido especificados sem considerar as tolerâncias, ou as tolerâncias podem ter sido estabelecidas sem considerar o nível tecnológico atingido pela empresa;
- d) o padrão está suficientemente concreto? Padrões abstratos e de difícil entendimento também são inúteis. Por exemplo: simplesmente estipular que a "aparência e forma devem ser precisas e atraentes" não especifica o significado de "precisas e atraentes". Neste caso é necessário especificar concretamente o número máximo de arranhões permissível, ou o índice de reflexão da superfície, etc. Uma outra maneira é ter fotografias padrão representando o que é "preciso" e o que é "atraente";
- e) todo conhecimento técnico da empresa deve fluir para os padrões como forma de serem utilizados pelos operadores para o benefício da comunidade;
- f) passíveis de serem revistos pelo menos uma vez por ano devido à incorporação de inovações;
- g) não se basear somente na teoria ou ser fruto de idealismos, porém ser solidamente baseado na prática;
- h) deve ter a sua elaboração não restrita à delimitação da sequência do trabalho, mas voltada ao atendimento das necessidades do mesmo. A própria redação deverá ser conforme o espírito de que o usuário é cliente do redator;
- i) indicar claramente as datas de emissão e de revisão, o período da validade e as responsabilidades específicas;
- j) os esboços deverão ser resultantes de um consenso, principalmente das áreas responsáveis;

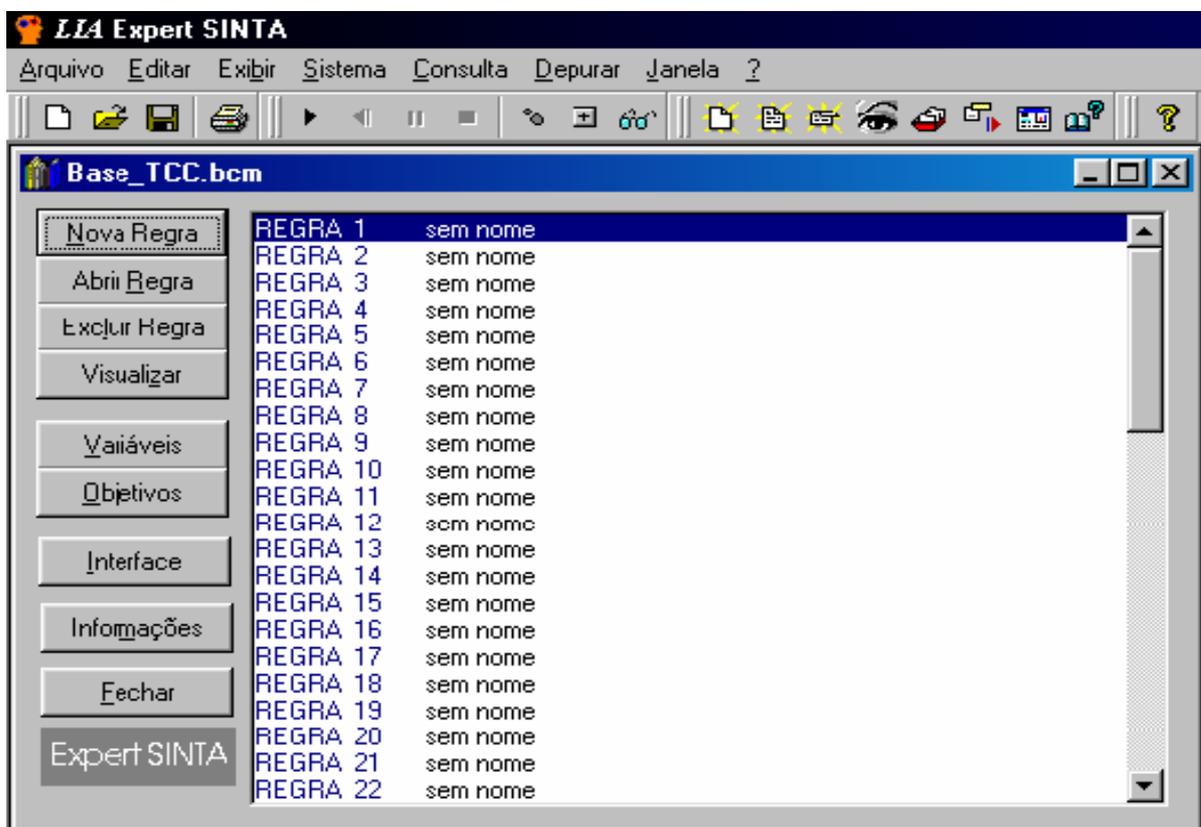
- k) os padrões devem ser autorizados por hierarquia imediatamente superior e cumpridos;
- l) um padrão, sendo parte de um sistema, nunca poderá contradizer outro;
- m) deverá ser mantido um controle da manutenção dos padrões e do número de revisões;
- n) os padrões devem ter seus nomes e formas padronizados para toda empresa;
- o) os padrões devem direcionar-se para o futuro a partir de uma situação atual dominada.

4 FERRAMENTAS

4.1 EXPERT SINTA SHELL

O Expert Sinta Shell é uma ferramenta computacional que utiliza técnicas de inteligência artificial para geração automática de sistemas especialistas. Esta ferramenta utiliza um modelo de representação do conhecimento baseado em regras de produção tendo como objetivo principal simplificar o trabalho de implementação de sistemas especialistas através do uso de uma máquina de inferência compartilhada, da construção automática de telas e menus, do tratamento das regras de produção e da utilização de explicações sensíveis ao contexto da base de conhecimento modelada. Um sistema especialista baseado em tal tipo de modelo é bastante útil em problemas de classificação. O usuário responde a uma seqüência de menus, e o sistema encarregar-se-á de fornecer respostas que se encaixem nas características apontadas pelo usuário (Lia, 2000).

FIGURA 1 – FERRAMENTA EXPERT SINTA



Para utilizar o Expert Sinta como ferramenta de apoio a construção do sistema especialista (figura 1), é necessário que todas as variáveis utilizadas, bem como seus respectivos valores, sejam criados. Através desse mecanismo, a base fica organizada, fácil de manter e as regras podem ser criadas visualmente.

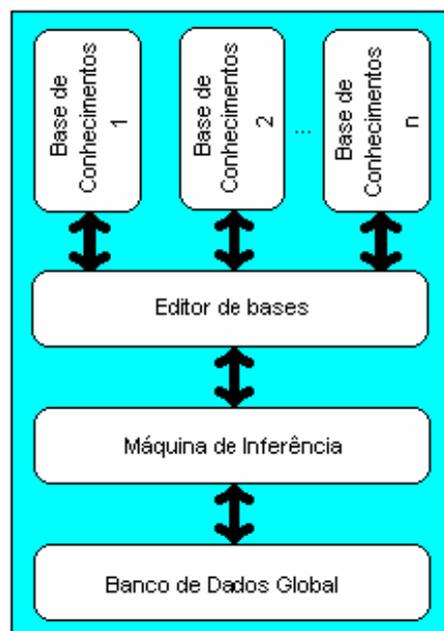
O objetivo de uma consulta a um especialista é encontrar a resposta para um determinado problema, assim também é um sistema especialista, a diferença é que, aqui, os problemas são representados por variáveis. Antes de poder-se executar esse sistema pela primeira vez, é preciso que defina-se quais são as variáveis (chamadas variáveis objetivo) que irão controlar o modo como a máquina de inferência se comporta.

Um sistema especialista implementado com o Expert Sinta comunica-se com o usuário final através de menus de múltipla escolha (ou escolha simples se a variável em questão for univalorada).

4.1.1 ARQUITETURA DE UM SISTEMA ESPECIALISTA NO EXPERT SINTA SHELL

Conforme Lia(2000), os sistemas especialistas gerados no Expert Sinta seguem a arquitetura (figura 2):

FIGURA 2 – ARQUITETURA DO EXPERT SINTA



onde:

- a) base de conhecimentos: representa a informação (fatos e regras) que um especialista utiliza, representada computacionalmente;
- b) editor de bases: é o meio pelo qual a *shell* permite a implementação das bases desejadas;
- c) máquina de inferência: é a parte do Sistema Especialista responsável pelas deduções sobre a base de conhecimentos;
- d) banco de dados global: são as evidências apontadas pelo usuário do sistema especialista durante uma consulta.

O objetivo do Expert Sinta é simplificar ao máximo as etapas de criação de um sistema especialista completo. Para tanto, já oferece uma máquina de inferência básica, fundamentada no encadeamento para trás (*backward chaining*).

4.1.2 EXPERT SINTA VISUAL COMPONENT LIBRARY (VCL)

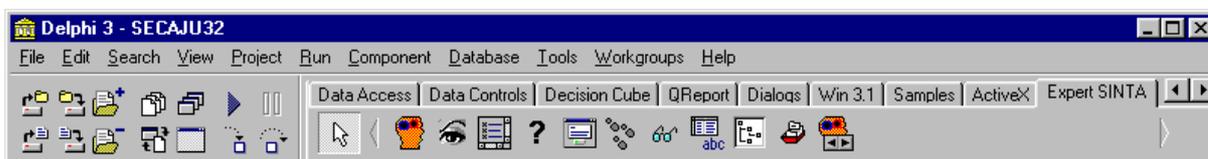
De acordo com Lia (2000), a Expert Sinta VCL é uma biblioteca de componentes para programação de Sistemas Especialistas. Na figura 3 temos a barra de ferramentas da Expert Sinta VCL para Delphi. De uma forma geral, esta biblioteca de componentes torna possível a criação de aplicações para bases de conhecimento geradas com o Expert Sinta. Esta biblioteca de componentes vem sendo utilizada na construção de sistemas especialistas.

Os principais motivos da utilização do Expert Sinta VCL são:

- a) o Expert Sinta (*Shell*) não provém toda a funcionalidade necessária a certos Sistemas Especialistas;
- b) não há meios de aproveitar os dados obtidos com o *shell* em outros programas;
- c) é inviável o acréscimo de vários recursos de interface e intercâmbio de dados na ferramenta em si;
- d) os sistemas especialistas devem ser compilados em uma dada linguagem de programação e utilizados de forma totalmente independente do Expert Sinta;

- e) é possível reaproveitar milhares de linhas de código já escritas na construção do *shell*.

FIGURA 3 – BARRA DE FERRAMENTAS DO DELPHI COM A BIBLIOTECA (EXPERT SINTA VCL)



As seguintes tarefas são desempenhadas pela Expert Sinta VCL (Lia, 2000):

- encapsulamento da máquina de inferência e a estrutura de dados que representa o conhecimento (regras de produção);
- fornecimento de mecanismos para entrada de dados do usuário;
- fornecimento de mecanismos para depuração;
- permitir a personalização da aplicação final.

4.2 DELPHI

O Delphi é um ambiente de desenvolvimento que usa vários recursos e conceitos avançados de interface gráfica. Ele permite que se estabeleça um abrangente controle do aplicativo, utilizando para tal conceitos de programação baseada em eventos e baseada em objetos (Fraga, 1997).

A programação baseada em eventos sempre foi utilizada antes do surgimento das *GUIs* (*Graphical User Interfaces*) e está disponível a qualquer programador. Antes dos ambientes de programação baseados em eventos, os estilos de programação procedural de cima para baixo eram considerados de última geração. O projeto de programas de cima para baixo é muito útil para a construção de segmentos de código que tratam um grande volume de processamento. Na verdade, aplicativos construídos com técnicas de programação de cima para baixo geram normalmente um código, eficiente e que permite manutenção. No entanto, antes da programação baseada em evento, os aplicativos construídos através do método de cima para baixo tinham seqüências de botões e menus extremamente complexas que estavam estreitamente vinculadas ao processamento de aplicativos. A programação baseada em

eventos não substitui a programação procedural. Em vez disso, ela a complementa através de uma estrutura que oferece uma melhor separação entre a interface com o usuário e o processamento específico da atividade empresarial. O Delphi e outros ambientes baseados em eventos fornecem essas estruturas que permitem que o programador se concentre mais na lógica específica do aplicativo, em vez de se preocupar com as possíveis maneiras de tratar e controlar solicitações de usuários (Fraga, 1997).

A programação orientada a objetos apresenta grande capacidade de criar códigos reutilizáveis e adapta-se muito bem a situações reais, utilizando mecanismos básicos como objetos, propriedades, atributos, mensagens, métodos, classes, instâncias e herança (Fraga, 1997).

Entre as principais características do Delphi apresentadas por Matcho (1996), pode-se citar:

- a) é baseado na linguagem Pascal;
- b) é orientado a objetos;
- c) acessa as funções do Windows, permitindo que o aplicativo manipule os recursos do mesmo;
- d) permite o uso de bibliotecas, criadas em C, C++ e Pascal;
- e) gera arquivos executáveis;
- f) cria de forma simples e rápida aplicativos que utilizam bancos de dados;
- g) manipula arquivos de bancos de dados de diversos formatos;
- h) cria aplicativos que manipulam bancos de dados Cliente - Servidor;
- i) cria também de forma simples e rápida relatórios e etiquetas.

4.3 FERRAMENTAS CASE

A sigla CASE significa Engenharia de Software Auxiliada por Computador

(*Computer Aided Software Engeneering*), apesar de alguns autores terem tentado estender seu significado para Engenharia de Sistemas Auxiliada por Computador, argumentando que o campo se estende além da produção de software. O termo foi criado no começo dos anos oitenta, quando ficou claro que ferramentas gráficas, tais como diagramas de fluxo de dados (DFDs), diagramas entidade relacionamento (ERDs) e gráficos de estruturas poderiam ser úteis em análise e projeto de sistemas. Como os engenheiros das áreas automotiva, aeroespacial e outras tiraram grandes vantagens dos sistemas de projeto auxiliados por computador (*Computer Aided Design - CAD*) para desenvolver desenhos e fazer cálculos, acreditou-se que gráficos com auxílio de computador poderiam igualmente ser úteis para os profissionais de sistemas de gerência de informação (Gane, 1990).

Segundo Fisher (1990), as ferramentas para a 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 pelo seu arquiteto. O objetivo principal da tecnologia CASE é separar o projeto do programa aplicativo da implementação do código. Em geral, quanto mais afastado estiver o processo de projeto, melhor. Várias organizações e programadores já reconhecem este planejamento básico de software e princípio de estruturação há muito tempo. Nos últimos quinze anos têm-se desenvolvido e apresentado diversas metodologias estruturadas a vários programadores. Estas metodologias ofereceram uma estrutura de projeto e um conjunto de formalismos e normas em que se basear para a condução do trabalho de desenvolvimento de software.

O futuro da computação pertence à tecnologia CASE, na qual o analista projeta aplicações em uma tela em uma estação de trabalho e a máquina gera código executável. A máquina pode fazer todos os tipos de checagens cruzadas e validações, podendo automatizar muitas tarefas que eram consumidoras de tempo. Um desafio ao projeto apoiado por computador é tornar o processo de fácil utilização de modo que, quando apropriado, ele possa ser executado por usuários adequadamente treinados. Os usuários finais precisam ter a capacidade de resolver seus próprios problemas utilizando computadores. Esses usuários incluem desde engenheiros que executam cálculos intrincados até empresários que tomam decisões complexas com o apoio de informações computadorizadas (Martin, 1991).

4.4 ORACLE DESIGNER 2000

O *Oracle Designer 2000* é um ambiente de desenvolvimento baseado em uma ferramenta CASE que envolve todo o ciclo de vida do desenvolvimento de sistemas (planejamento estratégico, análise, projeto, construção, documentação, transição e produção). O *Oracle Designer 2000* suporta as metodologias e técnicas de engenharia da informação, Engenharia Reversa, BPR (Reengenharia de Processos), Programando em Oracle e RAD (Desenvolvimento Rápido de Aplicações). Todas as informações utilizadas no ciclo do desenvolvimento são armazenadas num repositório único e poderoso, permitindo uma completa integração das fases e disponibilizando toda a documentação gerada pelo produto (Saraiva, 1999).

O *Oracle Designer 2000* oferece uma solução completa quando os clientes Oracle precisam projetar, programar, implementar e manter sistemas. Eles proporcionam rápido desenvolvimento de aplicativo em um ambiente Windows cliente/servidor. Sua funcionalidade avançada suporta *Business Process Reengineering - BPR* (Reestruturação dos Processos do Negócio) e mecanismos para tirar proveito do processamento de servidor que pode ser feito usando-se a máquina de banco de dados do Oracle (Abbey, 1997).

4.4.1 TÉCNICAS DE MODELAGEM

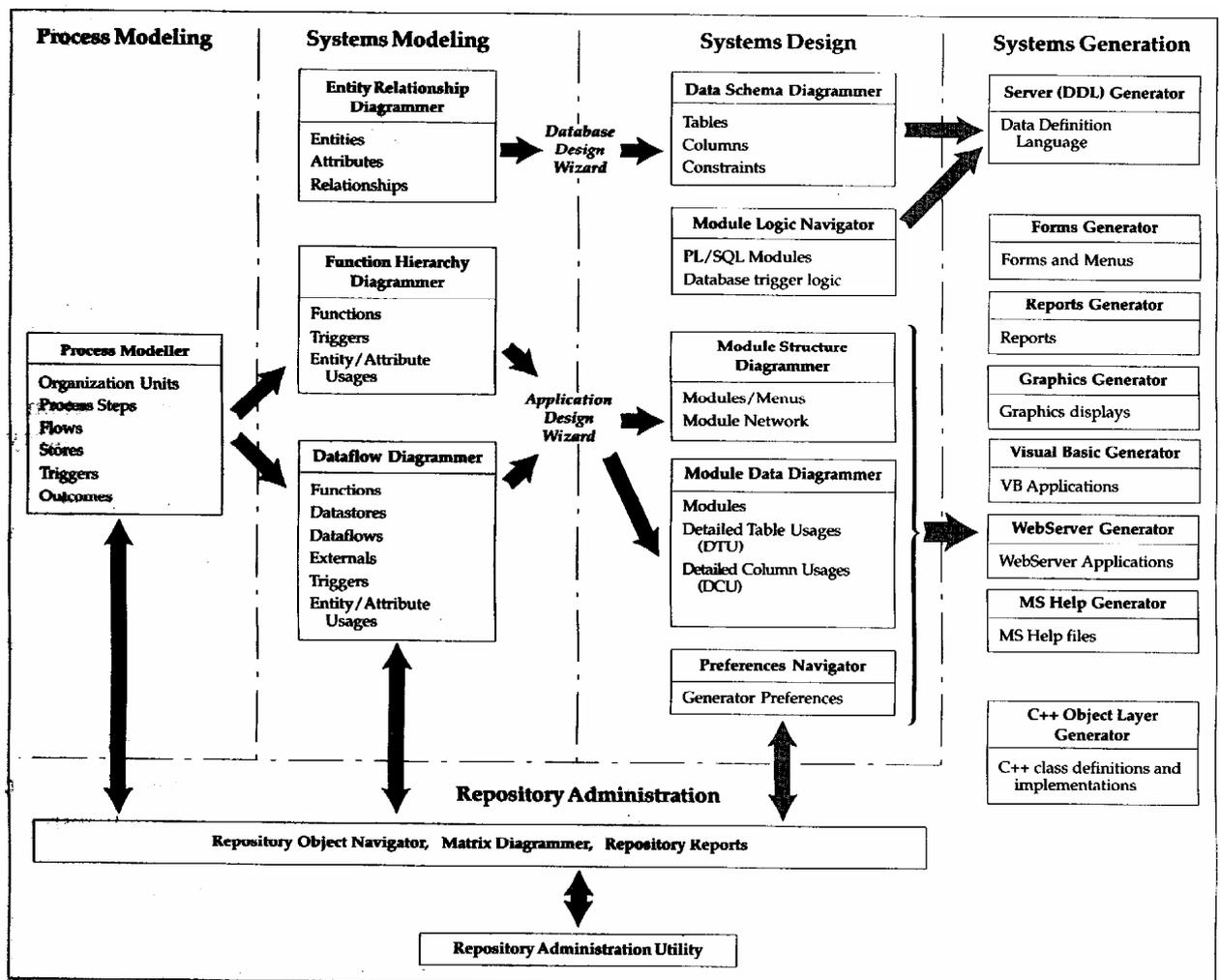
Para usar o *Oracle Designer 2000* efetivamente, é necessário entender as técnicas (figura 4) que as ferramentas suportam, que são as seguintes:

- a) modelagem de processos (*Process Modeller*): produz diagramas de processo que ilustram os processos executados numa organização a fim de satisfazer as necessidades de seus clientes;
- b) modelagem entidade relacionamento (*Entity Relationship Diagrammer*): produz diagramas MER que define os objetos acerca dos quais a organização necessita manter informações, bem como as associações entre eles;
- c) decomposição de funções (*Function Hierarchy Diagrammer*): produz diagramas hierárquicos de função (DHF ou FHD) que mostra as atividades que uma

organização faz ou precisa fazer no futuro;

- d) referência cruzada usando diagrama de matrizes (*Matrix Diagrammer*): identifica e descreve as associações entre elementos em seus sistemas;
- e) referência cruzada usando diagramas de fluxo de dados (*Dataflow Diagrammer*): produz conjuntos de diagramas de fluxos de dados (DFD) que mostram como as informações fluem numa área de atividade da organização, fornecendo as “*business functions*” com os dados que elas requerem.

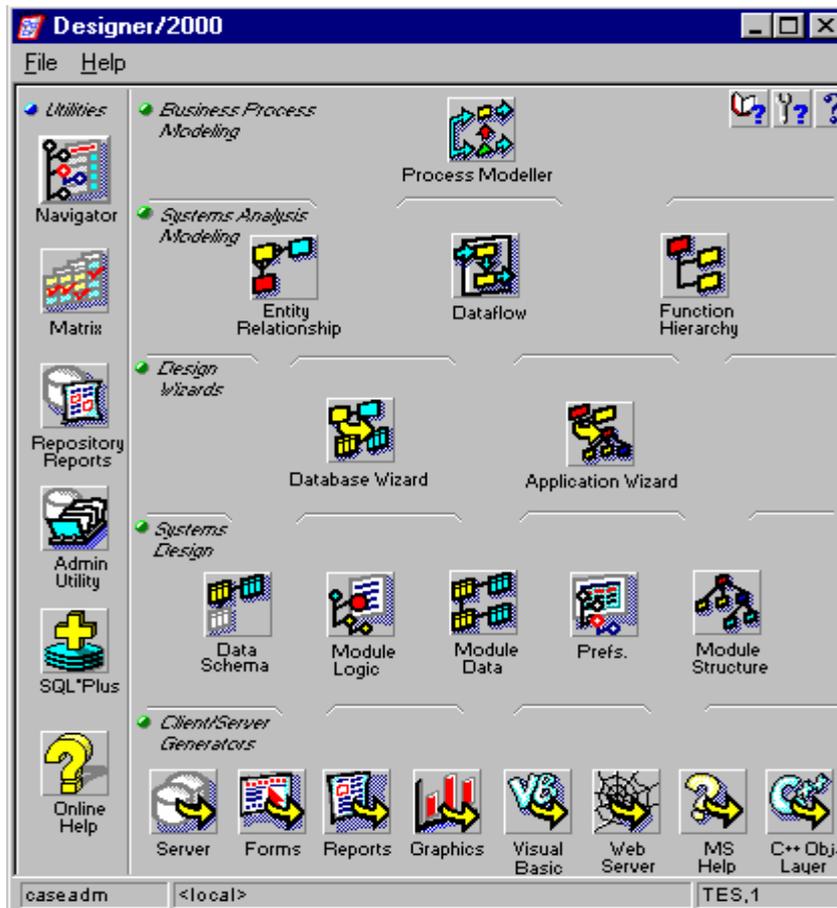
FIGURA 4 – FLUXOGRAMA DO ORACLE DESIGNER 2000



Fonte: ORACLE CORPORATION (1996)

A figura 5 demonstra a tela inicial do Oracle Designer 2000 e as aplicações disponíveis.

FIGURA 5 –TELA INICIAL DO ORACLE DESIGNER 2000



4.4.2 ARQUITETURA DO REPOSITÓRIO

O *Oracle Designer 2000* grava os modelos de negócio num banco de dados chamado repositório. Para usar o Designer 2000 efetivamente deve-se entender como são guardadas as informações no repositório:

- a) aplicações: todos os objetos que são definidos nos modelos devem pertencer a uma aplicação ou sistema de aplicação. Um sistema de aplicação pode ser o projeto em que se está trabalhando;
- b) tipos de objeto no repositório: os objetos gravados no repositório podem ser:
 - tipo de elemento, por exemplo entidades;

- elemento, por exemplo a entidade Pedido;
 - tipo de associação, por exemplo *Function Entity Usages*;
 - associação, por exemplo a função que cadastra pedido lê a entidade pedido;
- c) elementos e tipos de elemento: o relacionamento entre elementos e tipos de elementos no repositório é simples: "Cada elemento deve ser um e somente um tipo de elemento":
- tipos de elementos primários são subordinados diretamente a aplicação. Por exemplo, instâncias de entidades e funções são elementos primários;
 - tipos de elementos secundários são subordinados aos tipos de elementos primários. Por exemplo, um atributo é subordinado à entidade a qual ele faz parte da definição.
- d) associações e tipos de associação: o relacionamento entre associações e tipos de associação é: "Cada associação deve ser de um e somente um tipo de associação";
- e) elementos e associações: uma associação é uma interseção que resolve o relacionamento muitos-para-muitos entre elementos;
- f) tipos de elementos e tipos de associação: somente certos tipo de elementos podem ser associados com outros tipos de elementos determinados;
- g) propriedades: os elementos e associações têm propriedades que determinam sua definição completa.

5 DESENVOLVIMENTO DO TRABALHO

O principal objetivo do trabalho é desenvolver um sistema especialista capaz de auxiliar o especialista humano nas atividades de validação de modelos de dados.

A dificuldade dos especialistas humanos em validar modelos de dados deve-se a diversidade e a quantidade de objetos e informações existentes nestes modelos. Portanto, ao utilizar um sistema especialista para validação destas bases de dados, o especialista humano encontrará com simplicidade os problemas de padronização dos modelos de dados.

5.1 AQUISIÇÃO DO CONHECIMENTO

A aquisição de conhecimento é a transferência dos conhecimentos acumulados de um especialista de uma determinada área para um sistema especialista. Para a aquisição de conhecimento utilizado no desenvolvimento do sistema, entrou-se em contato com alguns profissionais da área de informática como analistas de sistemas, administradores de bases de dados e administradores de dados, que forneceram algumas informações sobre as características mais relevantes de um modelo de dados.

Estas informações e características determinaram os padrões utilizados na implementação do protótipo são:

a) Tabela (*table*):

- sufixo: o nome da tabela deveser terminar com “_[Sigla da Aplicação]”;
- termos do nome: todos os termos devem estar cadastrados no glossário de termos;
- atributos: deve possuir pelo menos um atributo;
- chave primária (*primary key*): deve possuir uma chave primária;
- sinônimo: deve possuir um sinônimo;
- apelido (*alias*): deve possuir um *alias* com menos de 6 caracteres;

- descrição (*description*): deve possuir uma descrição explicando a existência da tabela;
- b) domínio (*domain*):
- prefixo: o nome do domínio deverá começar com “ID_”;
 - termos do nome: todos os termos devem estar cadastrados no glossário de termos;
 - Valores: deve possuir pelo menos um valor associado;
 - descrição (*description*): deve possuir uma descrição explicando a existência do domínio;
- c) atributo:
- prefixo: o nome do atributo deverá começar com “[SIGLA DO TIPO DE DADO]_”;
 - termos do nome: todos os termos devem estar cadastrados no glossário de termos;
 - tipo de dado: deverá ser compatível com a sigla do tipo de dado;
 - sinônimo: deve possuir um sinônimo;
 - comentário (*prompt*): deverá possuir um *prompt*;
 - alerta (*hint*): deverá possuir um *hint*;
 - descrição (*description*): deve possuir uma descrição explicando a existência do atributo;
 - tamanho máximo (*maximum length*): deverá possuir *maximum length*;
- d) chave primária (*primary key*):

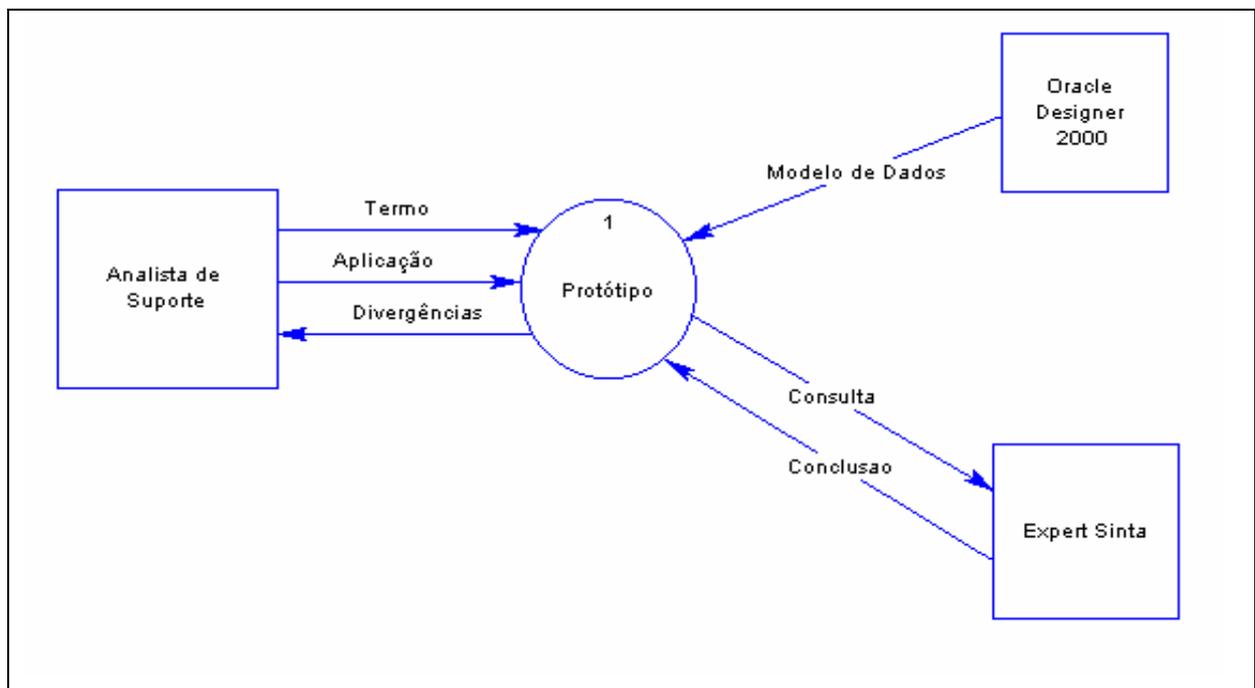
- prefixo: o nome da chave primária deverá começar com “[ALIAS DO OBJETO]_”;
- sufixo: o nome da chave primária deverá terminar com “_PK”;
- atributos: deve possuir pelo menos um atributo.

Através destas informações começou-se a elaborar a base de conhecimentos do sistema especialista utilizando regras de produção para representar o conhecimento. A seguir foram feitos refinamentos na base de conhecimento com o parecer do especialista humano. Para finalizar, foram feitos alguns testes com situações fictícias.

5.2 FORMALIZAÇÃO DO CONHECIMENTO

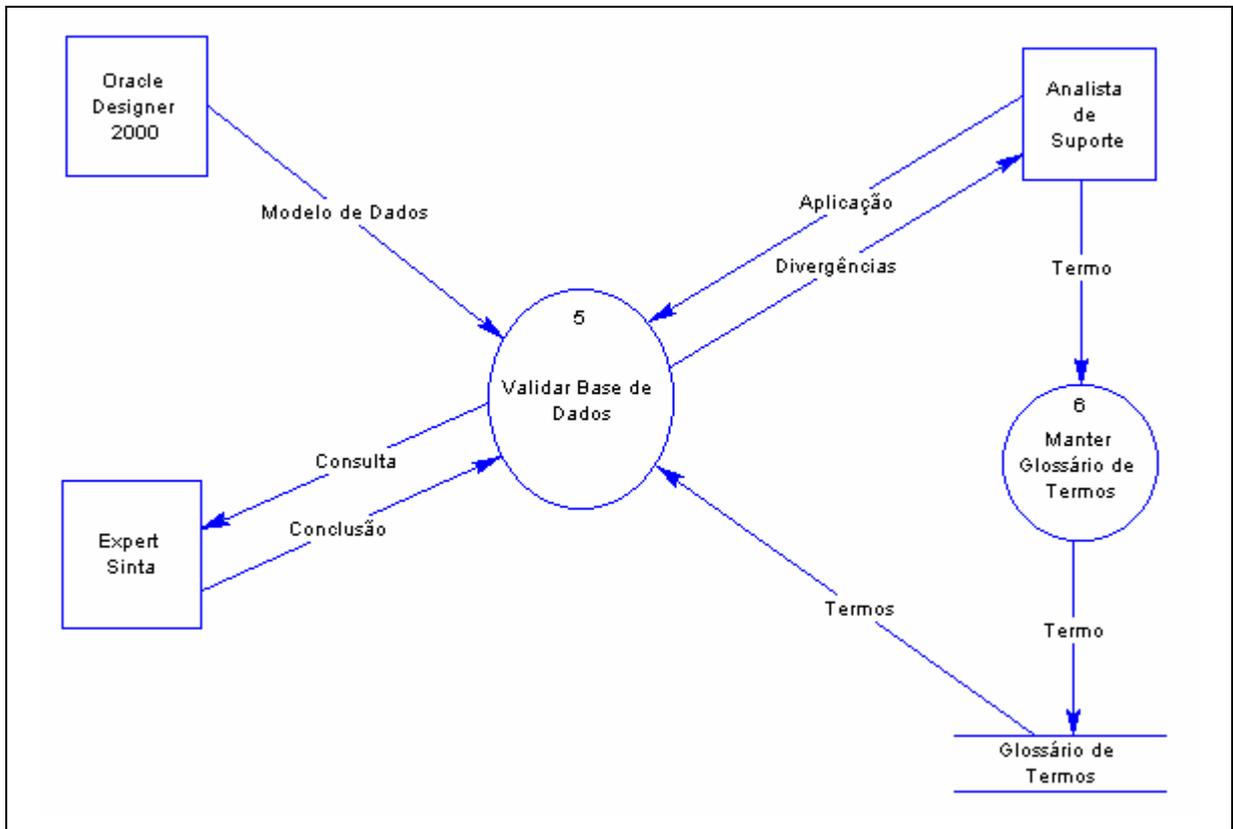
O problema foi especificado utilizando-se a ferramenta PowerDesigner, baseado nos conceitos da análise estruturada. No diagrama de contexto (figura 6), pode-se observar claramente as entidades externas do problema em questão, bem como, as informações que serão utilizadas na solução e/ou implementação do protótipo.

FIGURA 6 – DIAGRAMA DE CONTEXTO



O problema pode ser visualizado também utilizando-se o diagrama de fluxo de dados (figura 7), no qual é possível entender o fluxo dos dados relacionados ao problema e função das entidades externas (Analista de Suporte, Expert Sinta, Oracle Designer 2000).

FIGURA 7 – DIAGRAMA DE FLUXO DE DADOS



5.3 IMPLEMENTAÇÃO

5.3.1 TÉCNICAS E FERRAMENTAS UTILIZADAS

Inicialmente, com a utilização do Expert Sinta, foram descritas as variáveis (anexo 1), as perguntas ou interface com usuário (anexo 2) e as regras de produção (anexo 3) que constituem o sistema especialista.

A integração da linguagem de programação Delphi com o Sistema Especialista, desenvolvido na ferramenta Expert Sinta, foi realizada através dos componentes Expert Sinta VCL, que incorporam o Sistema Especialista ao Delphi.

Os modelos de dados descritos na ferramenta Oracle Designer 2000 são acessados pelo Delphi através de visões existentes no banco de dados Oracle.

Em seguida foram programados os eventos “OnPrompt” e “OnShowResults”, que respectivamente, são utilizados para solicitar as entradas das informações para o sistema especialista (quadro 1) e a obtenção da conclusão do sistema especialista (quadro 2).

Quadro 1 – Implementação do evento OnPrompt

```

procedure TForm1.ExpertSystem1Prompt(Sender: TObject; V: Integer);
begin
    ExpertSystem1.AttribVarFromBinary(V, busca_valor(V) );
    ExpertSystem1.WaitingAnswer := false;
end;

```

Quadro 2 – Implementação do evento OnShowResult

```

procedure TForm1.ExpertSystem1ShowResults(Sender: TObject; V: Integer);
var
    i, total: integer;
    values_list, cnfs_list: TStringList;
begin
    values_list := TStringList.Create;
    cnfs_list := TStringList.Create;
    try
        ExpertSystem1.VarInstances(V, values_list, cnfs_list);
        total := values_list.Count - 1;
        for i := 0 to total do
            begin
                if values_list[i] = 'Sim' then
                    memoResults.Lines.Add(busca_descricao(ExpertSystem1.VarName(V) ));
                end;
            finally
                values_list.Free;
                cnfs_list.Free;
            end;
        end;

```

Depois, foram identificadas as tabelas do Oracle (anexo 4), onde são armazenadas as informações da ferramenta Oracle Designer 2000, para que a partir destas tabelas o sistema especialista seja executado (quadro 3), gerando suas conclusões, ou seja, determinando os problemas de padronização da base de dados.

Quadro 3 – Implementação da chamado do Expert Sinta

```
//-----
//   Domínios
//-----
CI_DOMAINS.first;
while not CI_DOMAINS.EOF do begin

    CDI_DOMAINS_TEXT.first;
    CI_DOMAINS_SYNONYMS.first;
    CI_DOMAINS_ATTRIBUTE_VALUES.first;

    Prepara_dados_dominios(CI_APPLICATION_SYSTEMS,
                           CI_DOMAINS,
                           CI_DOMAINS_SYNONYMS,
                           CDI_DOMAINS_TEXT,
                           CI_DOMAINS_ATTRIBUTE_VALUES);

    ED_Nivel.text := '4';
    memoResults.Lines.Add('      '+ CI_DOMAINS.FieldName('NAME').AsString );
    ExpertSystem1.StartConsultation;
    CI_DOMAINS.next;
end;
```

5.3.2 OPERACIONALIDADE DA IMPLEMENTAÇÃO

Ao executar o aplicativo, o mesmo apresenta a tela de login (figura 8), onde deve ser informado o usuário e a senha.

FIGURA 8 – TELA DE LOGIN

The image shows a 'Database Login' dialog box. The title bar is blue with the text 'Database Login' and a close button (X). The dialog has a light gray background. It contains a 'Database:' label followed by the text 'tcc'. Below this are two input fields: 'User Name:' and 'Password:'. At the bottom of the dialog are two buttons: 'OK' and 'Cancel'.

Em seguida é apresentado a tela principal do sistema (figura 9), que disponibiliza em seu menu, uma opção de “Arquivo”, utilizado para finalizar o sistema e uma opção “Cadastros”, pelo qual pode-se acessar o glossário de termos (figura 10) para cadastrar os termos que serão utilizados pelo sistema na validação dos objetos do modelo de dados.

FIGURA 9 – TELA PRINCIPAL DO APLICATIVO

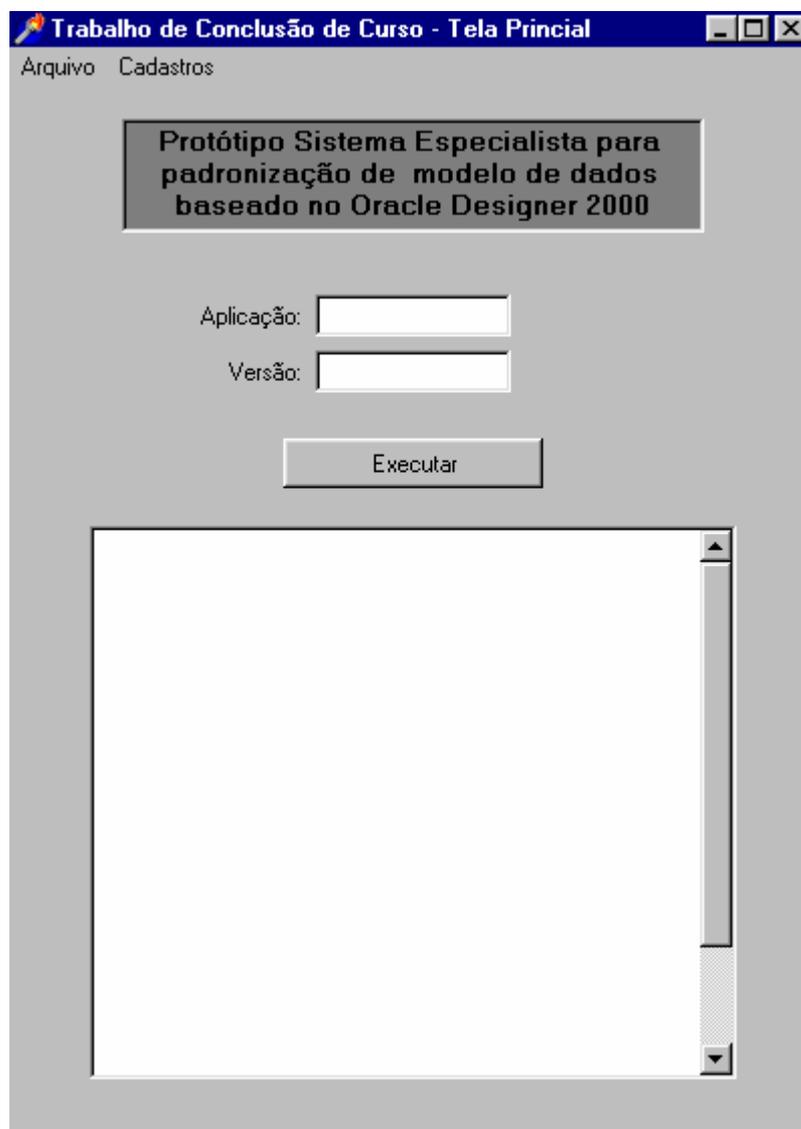


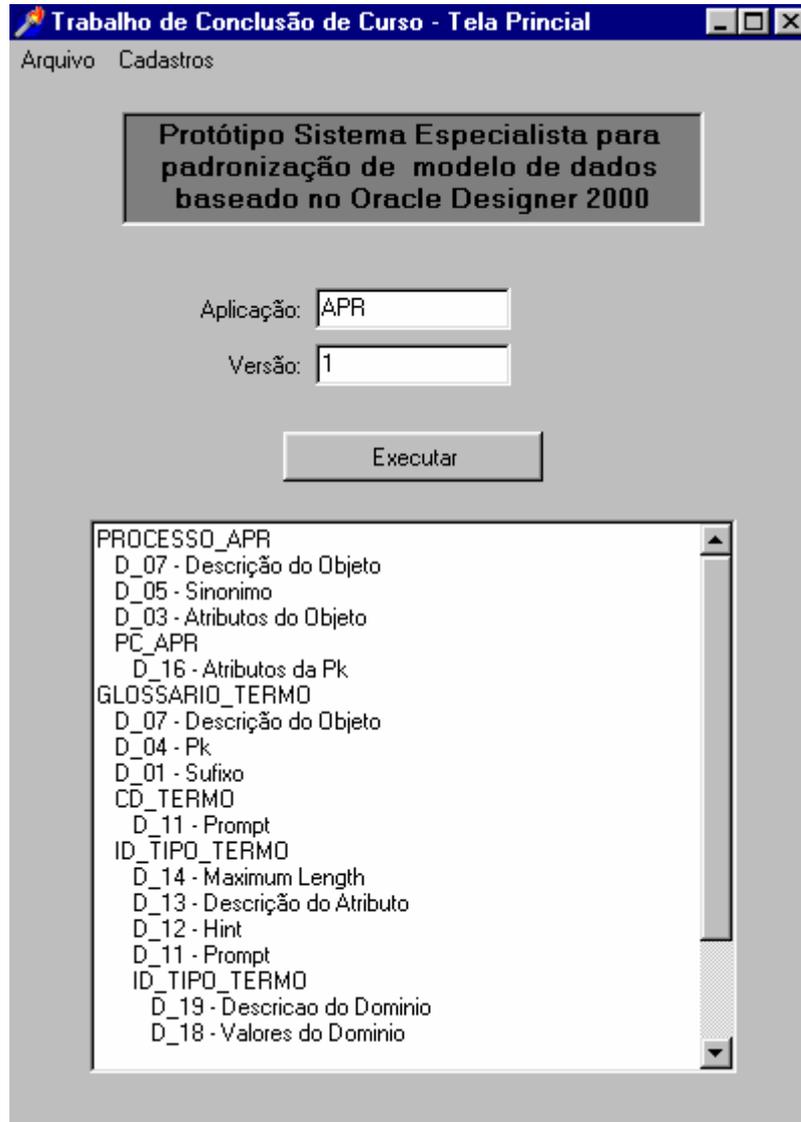
FIGURA 10 – CADASTRO DE TERMOS NO GLOSSÁRIO

Código	Descrição
TESTE	Teste
TES	Teste
DS	Prefixo Descricao
NDOME	Nome
TES	Aplicacao Teste
FUNC	Funcionario
CD	Codigo
P	Procedure
SEQ	Sequencia
TIPO	Tipo
ID	Identificador
TABELA	Tabela

Para executar a análise do sistema especialista sobre um modelo de dados do Oracle Designer 2000, basta informar a sigla e versão de uma aplicação criada na referida ferramenta e pressionar em seguida o botão “Executar” .

Quando a execução estiver finalizada, são apresentadas na tela principal (figura 11), todos os objetos do modelo de dados, juntamente com as divergências encontradas.

FIGURA 11 – TELA PRINCIPAL DO APLICATIVO COM OS RESULTADOS



6 CONCLUSÕES

O presente trabalho permitiu um estudo dos aspectos relacionados à utilização da tecnologia dos sistemas especialistas, as formas de representação do conhecimento, regras de produção, o processo de inferência e a aplicação prática de um sistema especialista.

A utilização das regras de produção para a representação do conhecimento, apontado pela literatura especializada como a mais comum e viável, mostrou-se adequada, pois permitiu representar o conhecimento necessário na elaboração da base de conhecimentos do protótipo.

O sistema especialista no domínio de conhecimento no qual foi construído demonstrou resultados satisfatórios para os objetivos pretendidos, onde constatou-se a possibilidade de simplificar os trabalhos de validação de uma base de dados construída na ferramenta Oracle Designer.

6.1 EXTENSÕES

Como possibilidades de extensões para este trabalho, pode-se citar:

- a) desenvolvimento de um sistema especialista para validar modelos de dados independente da ferramenta geradora;
- b) criação de um sistema especialista que consiga corrigir automaticamente os problemas encontrados na análise.

7 REFERÊNCIAS BIBLIOGRÁFICAS

ABBEY, Michel. **Oracle**: guia do usuário. São Paulo: Makron Books, 1997.

ARTHUR, Lowel Jay. **Produtividade do programador**: um guia para gerentes, analistas e programadores. Rio de Janeiro: Editora Campus, 1985.

CAMPOS, Vicente Falvoni. **Qualidade total**: padronização de empresas. Belo Horizonte: Fundação Christiano Ottoni, 1992.

FISHER, Alan S. **Case**: utilização de ferramentas para desenvolvimento de software. Rio de Janeiro: Campus, 1990.

FRAGA, Simone. **Explorando**: Visão geral do delphi 2. Florianópolis: Visual Books, 1997.

GANE, Chris. **CASE**: o relatório Gane. Rio de Janeiro: LTC – Livros Técnicos e Científicos, 1990.

GENARO, Sérgio. **Sistemas Especialistas**: O conhecimento Artificial. São Paulo: LTC - Livros Técnicos e Científicos Editora AS, 1986.

HEINZLE, Roberto. **Protótipo de uma ferramenta para criação de sistemas especialista baseados em regras de produção**. 1995. 145 f. Dissertação (mestrado em Engenharia da Produção e Sistemas) - Setor de Engenharia da Produção e Sistemas, Universidade Federal de Santa Catarina, Florianópolis.

LEVINE, Robert I. **Inteligência artificial e sistemas especialistas**. São Paulo: McGraw-Hill, 1988.

LIA, Laboratório de Inteligência Artificial. **Expert Sinta**, uma ferramenta para criação de sistemas especialista, Ceara, nov. 2000. Disponível em: <<http://www.lia.ufc.br>>. Acesso em: 20 jan. 2001.

LIMA, Cynthia Moreira. **Introdução à inteligência artificial**, São Luiz, set. 2000. Disponível em: <<http://www.elo.com.br/~cynthia/2000>>. Acesso em: 15 fev. 2001.

LONGWORTH, G. **Padrões em programação**: métodos e procedimentos. Rio de Janeiro:

Editora Campus, 1985.

MAFFEO, Bruno. **Engenharia de software e especificação de sistemas**. Rio de Janeiro: Campus Ltda, 1992.

MARTIN, James. **Técnicas estruturadas e case**. São Paulo: Makron, 1991.

MATCHO, Jonathan. **Usando delphi 2**. Rio de Janeiro: Campus, 1996.

ORACLE CORPORATION. **Oracle Designer/2000 for windows 95/NT**. Andy Page: Oracle, 1996.

RIBEIRO, Horácio da Cunha e Souza. **Introdução aos sistemas especialistas**. São Paulo: LTC-Livros Tecnicos e Cientificos, 1987.

RABUSKE, Renato Antônio. **Inteligência Artificial**. Florianópolis, UFSC, 1995.

SARAIVA, Armando. **Programando em Oracle**. Rio de Janeiro: Infobook, 1999.

WEISS, Sholom. **Guia prático para projetar sistemas especialistas**. Rio de Janeiro: LTC-Livros Tecnicos e Cientificos Editora, 1988.

ANEXO 1 – Lista de Variáveis

VARIÁVEIS

Descricao do Objeto: univalorada

Dominio: univalorada

Alias: univalorada

Atributos da Pk: univalorada

Atributos do Objeto: univalorada

CONSISTE_ATRIBUTO: univalorada

CONSISTE_DOMINIO: univalorada

CONSISTE_PK: univalorada

Descricao do Atributo: univalorada

Descricao do Dominio: univalorada

Hint: univalorada

Maximum Length: univalorada

Nome da Pk: univalorada

Nome do Dominio: univalorada

Pk: univalorada

Prefixo da Pk: univalorada

Prompt: univalorada

Sinonimo: univalorada

Sufixo: univalorada

Termos do Atributo: univalorada

Termos do Dominio: univalorada

Termos do Objeto: univalorada

Tipo de Dado: Date / Number / Varchar / Outro / Long / Varchar2 / Char

Tipo de Objeto: Visão / Snapshot / Tabela

Tipo de Prefixo: PR / NR / ID / SG / TX / DS / NM / CD / VL / Outro / QT / DT

Valores do Dominio: Univalorada

VARIÁVEIS OBJETIVOS

D_01 – Sufixo do Objeto;

D_02 – Termos do Objeto;

D_03 – Atributos do Objeto;

D_04 – Pk;

D_05 – Sinônimo;

D_06 – Alias do Objeto;

D_07 – Descrição do Objeto;

D_08 – Tipo de Dado;

D_09 – Tipo de Prefixo;

D_10 – Termos do Atributo;

D_11 – Prompt;

D_12 – Hint;

D_13 – Descrição do Atributo;

D_14 – Maximum Length;

D_15 – Prefixo da Pk;

D_16 – Atributos da Pk;

D_17 – Nome do Domínio;

D_18 – Valores do Dominio;

D_19 – Descrição do Dominio.

ANEXO 2 – Lista de Variáveis e Perguntas

Variável: Descrição do Objeto

Pergunta: "O Objeto possui descrição?"

Variável: Domínio

Pergunta: "O Atributo possui Domínio relacionado?"

Variável: Alias

Pergunta: "O Objeto possui Alias?"

Variável: Atributos da Pk

Pergunta: "A PK possui atributos relacionados?"

Variável: Atributos do Objeto

Pergunta: "O Objeto possui atributos relacionados?"

Variável: Descrição do Atributo

Pergunta: "O Atributo possui descrição?"

Variável: Descrição do Domínio

Pergunta: "O Domínio possui descrição?"

Variável: Hint

Pergunta: "O Atributo possui Hint?"

Variável: Maximum Length

Pergunta: "O Atributo possui maximum length?"

Variável: Nome da Pk

Pergunta: "Os termos do nome da Pk estão no glossário de dados?"

Variável: Nome do Dominio

Pergunta: "O nome do Dominio esta correto?"

Variável: Pk

Pergunta: "O Objeto possui pk?"

Variável: Prefixo da Pk

Pergunta: "O prefixo da Pk esta correto?"

Variável: Prompt

Pergunta: "O Atributo possui prompt?"

Variável: Sinônimo

Pergunta: "O Objeto possui sinônimo?"

Variável: Sufixo

Pergunta: "O sufixo do Objeto esta correto?"

Variável: Termos do Atributo

Pergunta: "Os termos do nome do Atributo estão no glossariol de dados?"

Variável: Termos do Dominio

Pergunta: "Os termos do nome do Dominio estão no glossariol de dados?"

Variável: Termos do Objeto

Pergunta: "Os termos do nome do Objeto estão no glossario de dados?"

Variável: Tipo de Dado

Pergunta: "Qual o tipo de dado?"

Variável: Tipo de Objeto

Pergunta: "Qual o tipo de objeto?"

Variável: Tipo de Prefixo

Pergunta: "Qual o prefixo do atributo?"

Variável: Valores do Domínio

Pergunta: "O Domínio possui valores associados"

ANEXO 3 - Lista de Regras

Regra 1

SE Tipo de Objeto = Tabela
E Sufixo = Não
ENTÃO D_01 = Sim CNF 100%

Regra 2

SE Tipo de Objeto = Tabela
E Termos do Objeto = Não
ENTÃO D_02 = Sim CNF 100%

Regra 3

SE Tipo de Objeto = Tabela
E Atributos do Objeto = Não
ENTÃO D_03 = Sim CNF 100%

Regra 4

SE Tipo de Objeto = Tabela
E Atributos do Objeto = Sim
ENTÃO CONSISTE_ATRIBUTO = Sim CNF 100%

Regra 5

SE Tipo de Objeto = Tabela
E Pk = Não
ENTÃO D_04 = Sim CNF 100%

Regra 6

SE Tipo de Objeto = Tabela
E Pk = Sim
ENTÃO CONSISTE_PK = Sim CNF 100%

Regra 7

SE Tipo de Objeto = Tabela

E Sinonimo = Não

ENTÃO D_05 = Sim CNF 100%

Regra 8

SE Tipo de Objeto = Tabela

E Alias = Não

ENTÃO D_06 = Sim CNF 100%

Regra 9

SE Tipo de Objeto = Tabela

E Descricao do Objeto = Não

ENTÃO D_07 = Sim CNF 100%

Regra 10

SE CONSISTE_ATRIBUTO = Sim

E Tipo de Prefixo = CD

E Tipo de Dado <> Number

ENTÃO D_08 = Sim CNF 100%

Regra 11

SE CONSISTE_ATRIBUTO = Sim

E Tipo de Prefixo = DS

E Tipo de Dado <> Varchar2

E Tipo de Dado <> Varchar

E Tipo de Dado <> Char

ENTÃO D_08 = Sim CNF 100%

Regra 12

SE CONSISTE_ATRIBUTO = Sim

E Tipo de Prefixo = ID

E Tipo de Dado <> Char

E Tipo de Dado <> Number

ENTÃO D_08 = Sim CNF 100%

Regra 13

SE CONSISTE_ATRIBUTO = Sim

E Tipo de Prefixo = NM

E Tipo de Dado <> Varchar2

ENTÃO D_08 = Sim CNF 100%

Regra 14

SE CONSISTE_ATRIBUTO = Sim

E Tipo de Prefixo = NR

E Tipo de Dado <> Number

ENTÃO D_08 = Sim CNF 100%

Regra 15

SE CONSISTE_ATRIBUTO = Sim

E Tipo de Prefixo = DT

E Tipo de Dado <> Date

ENTÃO D_08 = Sim CNF 100%

Regra 16

SE CONSISTE_ATRIBUTO = Sim

E Tipo de Prefixo = TX

E Tipo de Dado <> Long

ENTÃO D_08 = Sim CNF 100%

Regra 17

SE CONSISTE_ATRIBUTO = Sim
E Tipo de Prefixo = QT
E Tipo de Dado <> Number
ENTÃO D_08 = Sim CNF 100%

Regra 18

SE CONSISTE_ATRIBUTO = Sim
E Tipo de Prefixo = VL
E Tipo de Dado <> Number
ENTÃO D_08 = Sim CNF 100%

Regra 19

SE CONSISTE_ATRIBUTO = Sim
E Tipo de Prefixo = SG
E Tipo de Dado <> Char
E Tipo de Dado <> Varchar
E Tipo de Dado <> Varchar2
ENTÃO D_08 = Sim CNF 100%

Regra 20

SE CONSISTE_ATRIBUTO = Sim
E Tipo de Prefixo = PR
E Tipo de Dado <> Number
ENTÃO D_08 = Sim CNF 100%

Regra 21

SE CONSISTE_ATRIBUTO = Sim
E Tipo de Prefixo = Outro
E Tipo de Dado = Date
ENTÃO D_09 = Sim CNF 100%

Regra 22

SE CONSISTE_ATRIBUTO = Sim
E Tipo de Prefixo = Outro
E Tipo de Dado = Long
ENTÃO D_09 = Sim CNF 100%

Regra 23

SE CONSISTE_ATRIBUTO = Sim
E Tipo de Prefixo = Outro
E Tipo de Dado <> Date
E Tipo de Dado <> Long
ENTÃO D_09 = Sim CNF 100%

Regra 24

SE CONSISTE_ATRIBUTO = Sim
E Termos do Atributo = Não
ENTÃO D_10 = Sim CNF 100%

Regra 25

SE CONSISTE_ATRIBUTO = Sim
E Prompt = Não
ENTÃO D_11 = Sim CNF 100%

Regra 26

SE CONSISTE_ATRIBUTO = Sim
E Hint = Não
ENTÃO D_12 = Sim CNF 100%

Regra 27

SE CONSISTE_ATRIBUTO = Sim
E Descricao do Atributo = Não
ENTÃO D_13 = Sim CNF 100%

Regra 28

SE CONSISTE_ATRIBUTO = Sim

E Maximum Length = Não

ENTÃO D_14 = Sim CNF 100%

Regra 29

SE CONSISTE_ATRIBUTO = Sim

E Dominio = Sim

ENTÃO CONSISTE_DOMINIO = Sim CNF 100%

Regra 30

SE CONSISTE_PK = Sim

E Prefixo da Pk = Não

ENTÃO D_15 = Sim CNF 100%

Regra 31

SE CONSISTE_PK = Sim

E Atributos da Pk = Não

ENTÃO D_16 = Sim CNF 100%

Regra 32

SE CONSISTE_DOMINIO = Sim

E Nome do Dominio = Não

ENTÃO D_17 = Sim CNF 100%

Regra 33

SE CONSISTE_DOMINIO = Sim

E Valores do Dominio = Não

ENTÃO D_18 = Sim CNF 100%

Regra 34

SE CONSISTE_DOMINIO = Sim

E Descricao do Dominio = Não

ENTÃO D_19 = Sim CNF 100%

ANEXO 3 – Lista de Tabelas do Repositório utilizadas no Protótipo.

CI_APPLICATION_SYSTEMS		
Coluna	Opcional	Tipo de Dado
APPLICATION_SYSTEM_OWNER		VARCHAR2(30)
APPLICATION_TYPE		VARCHAR2(10)
AUTHORITY		VARCHAR2(30)
CHANGED_BY		VARCHAR2(30)
CHECKOUT_VERSION		NUMBER(38)
CONSTRAINTS		VARCHAR2(240)
CREATED_BY		VARCHAR2(30)
DATE_CHANGED		DATE
DATE_CREATED		DATE
DISPLAY_TITLE		VARCHAR2(240)
ELEMENT_TYPE_NAME		VARCHAR2(17)
ID	Não	NUMBER(38)
LATEST_VERSION_FLAG		VARCHAR2(1)
NAME		VARCHAR2(100)
NEW_VERSION_GROUP		NUMBER(38)
NUMBER_OF_TIMES_MODIFIED		NUMBER(38)
PARENT_REFERENCE		NUMBER(38)
PRIORITIES		VARCHAR2(240)
REMARK		VARCHAR2(240)
STATUS		VARCHAR2(1)
TYPES		NUMBER(38)
USER_DEFINED_PROPERTY_0		VARCHAR2(240)
USER_DEFINED_PROPERTY_1		VARCHAR2(240)
USER_DEFINED_PROPERTY_2		VARCHAR2(240)
USER_DEFINED_PROPERTY_3		VARCHAR2(240)
USER_DEFINED_PROPERTY_4		VARCHAR2(240)
USER_DEFINED_PROPERTY_5		VARCHAR2(240)
USER_DEFINED_PROPERTY_6		VARCHAR2(240)
USER_DEFINED_PROPERTY_7		VARCHAR2(240)
USER_DEFINED_PROPERTY_8		VARCHAR2(240)
USER_DEFINED_PROPERTY_9		VARCHAR2(240)
VERSION		NUMBER(3)
VERSION_DATE		DATE

CI_TABLE_DEFINITIONS		
Coluna	Opcional	Tipo de Dado
ALIAS		VARCHAR2(20)

ALL_DATABASES_IND		VARCHAR2(1)
APPLICATION_SYSTEM_OWNED_BY		NUMBER(38)
CACHED		VARCHAR2(1)
CHANGED_BY		VARCHAR2(30)
CHECKOUT_VERSION		NUMBER(38)
CLUSTER_REFERENCE		NUMBER(38)
COLUMN_PREFIX		VARCHAR2(4)
CREATE_STATUS		VARCHAR2(1)
CREATED_BY		VARCHAR2(30)
DATABASE_REFERENCE		NUMBER(38)
DATABASE_TYPE		VARCHAR2(1)
DATE_CHANGED		DATE
DATE_CREATED		DATE
DISPLAY_TITLE		VARCHAR2(240)
ELEMENT_TYPE_NAME		VARCHAR2(17)
ID	Não	NUMBER(38)
INITIAL_NUMBER_OF_ROWS		NUMBER(23,3)
INITIAL_TRANSACTION		NUMBER(3)
JOURNAL_FLAG		VARCHAR2(1)
LOCKING_SEQUENCE		NUMBER(3)
MAX_SCANS		NUMBER(23,3)
MAXIMUM_NUMBER_OF_ROWS		NUMBER(23,3)
MAXIMUM_TRANSACTION		NUMBER(3)
MIN_PARTITIONS		NUMBER(23,3)
NAME		VARCHAR2(100)
NUMBER_OF_TIMES_MODIFIED		NUMBER(38)
PARALLEL		VARCHAR2(1)
PERCENT_FREE		NUMBER(3)
PERCENT_USED		NUMBER(3)
REMARK		VARCHAR2(240)
STORAGE_DEFINITION_REFERENCE		NUMBER(38)
TABLE_TYPE		VARCHAR2(10)
TABLESPACE_REFERENCE		NUMBER(38)
TYPES		NUMBER(38)
USER_DEFINED_PROPERTY_0		VARCHAR2(240)
USER_DEFINED_PROPERTY_1		VARCHAR2(240)
USER_DEFINED_PROPERTY_2		VARCHAR2(240)
USER_DEFINED_PROPERTY_3		VARCHAR2(240)
USER_DEFINED_PROPERTY_4		VARCHAR2(240)
USER_DEFINED_PROPERTY_5		VARCHAR2(240)
USER_DEFINED_PROPERTY_6		VARCHAR2(240)
USER_DEFINED_PROPERTY_7		VARCHAR2(240)
USER_DEFINED_PROPERTY_8		VARCHAR2(240)
USER_DEFINED_PROPERTY_9		VARCHAR2(240)

CI_DOMAINS		
Coluna	Opcional	Tipo de Dado
APPLICATION_SYSTEM_OWNED_BY		NUMBER(38)
ATTRIBUTE_PRECISION		VARCHAR2(4)

AUTHORITY		VARCHAR2(30)
AVERAGE_ATTRIBUTE_LENGTH		NUMBER(23,3)
AVERAGE_COLUMN_LENGTH		NUMBER(23,3)
CHANGED_BY		VARCHAR2(30)
CHECKOUT_VERSION		NUMBER(38)
COLUMN_PRECISION		NUMBER(2)
CREATED_BY		VARCHAR2(30)
DATATYPE		VARCHAR2(20)
DATATYPE_NAME		VARCHAR2(40)
DATE_CHANGED		DATE
DATE_CREATED		DATE
DEFAULT_VALUE		VARCHAR2(40)
DERIVATION		VARCHAR2(240)
DESCRIPTION		VARCHAR2(240)
ELEMENT_TYPE_NAME		VARCHAR2(17)
FORMAT		VARCHAR2(70)
ID	Não	NUMBER(38)
MAXIMUM_ATTRIBUTE_LENGTH		NUMBER(23,3)
MAXIMUM_COLUMN_LENGTH		NUMBER(23,3)
NAME		VARCHAR2(100)
NULL_VALUE		VARCHAR2(40)
NUMBER_OF_TIMES_MODIFIED		NUMBER(38)
SOFT_LOV		VARCHAR2(1)
SUPERTYPE_NAME		VARCHAR2(100)
SUPERTYPE_REFERENCE		NUMBER(38)
TYPES		NUMBER(38)
UNIT_OF_MEASURE		VARCHAR2(10)
USER_DEFINED_PROPERTY_0		VARCHAR2(240)
USER_DEFINED_PROPERTY_1		VARCHAR2(240)
USER_DEFINED_PROPERTY_2		VARCHAR2(240)
USER_DEFINED_PROPERTY_3		VARCHAR2(240)
USER_DEFINED_PROPERTY_4		VARCHAR2(240)
USER_DEFINED_PROPERTY_5		VARCHAR2(240)
USER_DEFINED_PROPERTY_6		VARCHAR2(240)
USER_DEFINED_PROPERTY_7		VARCHAR2(240)
USER_DEFINED_PROPERTY_8		VARCHAR2(240)
USER_DEFINED_PROPERTY_9		VARCHAR2(240)

CDI_TEXT		
Coluna	Opcional	Tipo de Dado
TXT_REF	Não	NUMBER(38)
TXT_SEQ	Não	NUMBER(6)
TXT_TYPE	Não	VARCHAR2(6)
TXT_TEXT		VARCHAR2(240)

CI_KEY_COMPONENTS		
Coluna	Opcional	Tipo de Dado
CHANGED_BY		VARCHAR2(30)
CHECKOUT_VERSION		NUMBER(38)
COLUMN_REFERENCE		NUMBER(38)
CONSTRAINT_NAME		VARCHAR2(100)
CONSTRAINT_REFERENCE		NUMBER(38)
CONSTRAINT_TYPE		VARCHAR2(10)
CREATED_BY		VARCHAR2(30)
DATE_CHANGED		DATE
DATE_CREATED		DATE
ELEMENT_TYPE_NAME		VARCHAR2(17)
FOREIGN_COLUMN_REFERENCE		NUMBER(38)
ID	Não	NUMBER(38)
NUMBER_OF_TIMES_MODIFIED		NUMBER(38)
REMARK		VARCHAR2(240)
SEQUENCE_NUMBER		NUMBER(3)
TYPES		NUMBER(38)
USER_DEFINED_PROPERTY_0		VARCHAR2(240)
USER_DEFINED_PROPERTY_1		VARCHAR2(240)
USER_DEFINED_PROPERTY_2		VARCHAR2(240)
USER_DEFINED_PROPERTY_3		VARCHAR2(240)
USER_DEFINED_PROPERTY_4		VARCHAR2(240)
USER_DEFINED_PROPERTY_5		VARCHAR2(240)
USER_DEFINED_PROPERTY_6		VARCHAR2(240)
USER_DEFINED_PROPERTY_7		VARCHAR2(240)
USER_DEFINED_PROPERTY_8		VARCHAR2(240)
USER_DEFINED_PROPERTY_9		VARCHAR2(240)

CI_COLUMNS		
Coluna	Opcional	Tipo de Dado
AUTO_GENERATED		VARCHAR2(9)
AVERAGE_LENGTH		NUMBER(23,3)
BASE_COLUMN_REFERENCE		NUMBER(38)
CHANGED_BY		VARCHAR2(30)
CHECKOUT_VERSION		NUMBER(38)
COLUMN_DENORMALISED_FROM		NUMBER(38)
CONSTRAINT_DENORMALISED_VIA		NUMBER(38)
CREATE_STATUS		VARCHAR2(1)
CREATED_BY		VARCHAR2(30)
DATATYPE		VARCHAR2(70)
DATE_CHANGED		DATE
DATE_CREATED		DATE
DECIMAL_PLACES		VARCHAR2(4)
DEFAULT_VALUE		VARCHAR2(40)
DENORMALIZED_SUMMARY_FUNCTION		VARCHAR2(40)
DESCRIPTOR_COLUMN		NUMBER(38)

DISPLAY_DATATYPE		VARCHAR2(20)
DISPLAY_FLAG		VARCHAR2(1)
DISPLAY_HEIGHT		NUMBER(23,3)
DISPLAY_LENGTH		NUMBER(23,3)
DISPLAY_SEQUENCE		NUMBER(3)
DOMAIN_REFERENCE		NUMBER(38)
ELEMENT_TYPE_NAME		VARCHAR2(17)
EXPRESSION_TEXT		VARCHAR2(240)
FIELD_INITIAL_VALUE		VARCHAR2(120)
FIELD_LEVEL		VARCHAR2(2)
FIELD_NULL_VALUE		VARCHAR2(40)
FINAL_VOLUME		NUMBER(3)
FKEY_CASCADE_DELETE		VARCHAR2(1)
FKEY_CASCADE_UPDATE		VARCHAR2(1)
FKEY_TRANSFERABLE		VARCHAR2(1)
FORMAT_MODIFIER		VARCHAR2(60)
HELP_TEXT		VARCHAR2(240)
HIGHLIGHTING		VARCHAR2(70)
ID	Não	NUMBER(38)
INITIAL_VOLUME		NUMBER(3)
JUSTIFICATION		VARCHAR2(10)
MAXIMUM_LENGTH		NUMBER(23,3)
NAME		VARCHAR2(100)
NULL_INDICATOR		VARCHAR2(10)
NUMBER_OF_TIMES_MODIFIED		NUMBER(38)
OCCURS		VARCHAR2(30)
ORDER_SEQUENCE		NUMBER(3)
PROMPT		VARCHAR2(240)
REDEFINES		VARCHAR2(30)
REMARK		VARCHAR2(240)
SEQUENCE_NUMBER		NUMBER(3)
SEQUENCE_REFERENCE		NUMBER(38)
SERVER_DEFINED		VARCHAR2(1)
SIGNED		VARCHAR2(1)
SOFT_LOV		VARCHAR2(1)
SORTING_ORDER		VARCHAR2(1)
SOURCE_ATTRIBUTE_REFERENCE		NUMBER(38)
SOURCE_ENTITY_DERIVED_FROM		NUMBER(38)
SOURCE_REL_ENTITY		NUMBER(38)
SOURCE_REL_ENTITY_ATTRIBUTE		NUMBER(38)
SOURCE_RELATIONSHIP_END		NUMBER(38)
STORAGE_FORMAT		VARCHAR2(15)
SUGGESTION_LIST_FLAG		VARCHAR2(1)
TABLE_REFERENCE		NUMBER(38)
TYPES		NUMBER(38)
UPPERCASE		VARCHAR2(1)
USER_DEFINED_PROPERTY_0		VARCHAR2(240)
USER_DEFINED_PROPERTY_1		VARCHAR2(240)
USER_DEFINED_PROPERTY_2		VARCHAR2(240)
USER_DEFINED_PROPERTY_3		VARCHAR2(240)
USER_DEFINED_PROPERTY_4		VARCHAR2(240)
USER_DEFINED_PROPERTY_5		VARCHAR2(240)

USER_DEFINED_PROPERTY_6		VARCHAR2(240)
USER_DEFINED_PROPERTY_7		VARCHAR2(240)
USER_DEFINED_PROPERTY_8		VARCHAR2(240)
USER_DEFINED_PROPERTY_9		VARCHAR2(240)
VALIDATION_FAILURE_MESSAGE		VARCHAR2(240)

CI_ATTRIBUTE_VALUES		
Coluna	Opcional	Tipo de Dado
ABBREVIATION		VARCHAR2(20)
ATTRIBUTE_REFERENCE		NUMBER(38)
ATTRIBUTE_VALUE_FOR		VARCHAR2(15)
CHANGED_BY		VARCHAR2(30)
CHECKOUT_VERSION		NUMBER(38)
COLUMN_REFERENCE		NUMBER(38)
CREATED_BY		VARCHAR2(30)
DATE_CHANGED		DATE

CI_PRIMARY_KEY_CONSTRAINTS		
Coluna	Opcional	Tipo de Dado
CHANGED_BY		VARCHAR2(30)
CHECKOUT_VERSION		NUMBER(38)
CONSTRAINT_TYPE		VARCHAR2(10)
CREATE_STATUS		VARCHAR2(1)
CREATED_BY		VARCHAR2(30)
DATE_CHANGED		DATE
DATE_CREATED		DATE
ELEMENT_TYPE_NAME		VARCHAR2(17)
ENTITY_REFERENCE		NUMBER(38)
ERROR_MESSAGE		VARCHAR2(70)
EXCEPTION_TABLE		VARCHAR2(40)
ID	Não	NUMBER(38)
IMPLEMENTATION_LEVEL		VARCHAR2(20)
INIT_TRANS		NUMBER(3)
KEY_UPDATEABLE		VARCHAR2(1)
MAX_TRANS		NUMBER(3)
NAME		VARCHAR2(100)
NUMBER_OF_TIMES_MODIFIED		NUMBER(38)
PCT_FREE		NUMBER(3)
SEQUENCE_NUMBER_FOR_UNIQUE_IDS		NUMBER(3)
STATUS		VARCHAR2(10)
STORAGE_DEFINITION_REFERENCE		NUMBER(38)
TABLE_REFERENCE		NUMBER(38)
TABLESPACE_REFERENCE		NUMBER(38)
TYPES		NUMBER(38)
USER_DEFINED_PROPERTY_0		VARCHAR2(240)

USER_DEFINED_PROPERTY_1		VARCHAR2(240)
USER_DEFINED_PROPERTY_2		VARCHAR2(240)
USER_DEFINED_PROPERTY_3		VARCHAR2(240)
USER_DEFINED_PROPERTY_4		VARCHAR2(240)
USER_DEFINED_PROPERTY_5		VARCHAR2(240)
USER_DEFINED_PROPERTY_6		VARCHAR2(240)
USER_DEFINED_PROPERTY_7		VARCHAR2(240)
USER_DEFINED_PROPERTY_8		VARCHAR2(240)
USER_DEFINED_PROPERTY_9		VARCHAR2(240)

CI_SYNONYMS		
Coluna	Opcional	Tipo de Dado
APPLICATION_SYSTEM_NAME		VARCHAR2(100)
APPLICATION_SYSTEM_REFERENCE		NUMBER(38)
CHANGED_BY		VARCHAR2(30)
CHECKOUT_VERSION		NUMBER(38)
CREATE_STATUS		VARCHAR2(1)
CREATED_BY		VARCHAR2(30)
DATABASE_OBJECT_FLAG		VARCHAR2(1)
DATE_CHANGED		DATE
DATE_CREATED		DATE
ELEMENT_TYPE_NAME		VARCHAR2(17)
ENTITY_REFERENCE		NUMBER(38)
ID	Não	NUMBER(38)
MODULE_REFERENCE		NUMBER(38)
NAME		VARCHAR2(100)
NUMBER_OF_TIMES_MODIFIED		NUMBER(38)
SEQUENCE_REFERENCE		NUMBER(38)
SNAPSHOT_DEFINITION_REFERENCE		NUMBER(38)
SYN_TYPE		VARCHAR2(10)
SYNONYM_FOR		VARCHAR2(15)
TABLE_DEFINITION_REFERENCE		NUMBER(38)
TERM_REFERENCE		NUMBER(38)
TYPES		NUMBER(38)
USER_DEFINED_PROPERTY_0		VARCHAR2(240)
USER_DEFINED_PROPERTY_1		VARCHAR2(240)
USER_DEFINED_PROPERTY_2		VARCHAR2(240)
USER_DEFINED_PROPERTY_3		VARCHAR2(240)
USER_DEFINED_PROPERTY_4		VARCHAR2(240)
USER_DEFINED_PROPERTY_5		VARCHAR2(240)
USER_DEFINED_PROPERTY_6		VARCHAR2(240)
USER_DEFINED_PROPERTY_7		VARCHAR2(240)
USER_DEFINED_PROPERTY_8		VARCHAR2(240)
USER_DEFINED_PROPERTY_9		VARCHAR2(240)
VIEW_DEFINITION_REFERENCE		NUMBER(38)