



AUTARQUIA ASSOCIADA À UNIVERSIDADE DE SÃO PAULO

**MODELO HÍBRIDO DE BANCO DE DADOS RELACIONAL,
DE ALTO DESEMPENHO E CAPACIDADE DE
ARMAZENAMENTO, PARA APLICAÇÕES VOLTADAS À
ENGENHARIA NUCLEAR**

JOSÉ GOMES NETO

Dissertação apresentada como parte
dos requisitos para obtenção do Grau
de Mestre em Ciências na Área de
Tecnologia Nuclear - Reatores.

Orientador:
Dr. Delvonei Alves de Andrade

**São Paulo
2008**



INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES
Autarquia associada à Universidade de São Paulo

**MODELO HÍBRIDO DE BANCO DE DADOS RELACIONAL, DE ALTO
DESEMPENHO E CAPACIDADE DE ARMAZENAMENTO, PARA
APLICAÇÕES VOLTADAS À ENGENHARIA NUCLEAR**

JOSÉ GOMES NETO



**Dissertação apresentada
como parte dos requisitos
para obtenção do grau de
Mestre em Ciências na Área de
Tecnologia Nuclear – Reatores**

**Orientador:
Dr. Delvonei Alves de Andrade**

SÃO PAULO

2008

À minha amada mãe, Leonor Eid,
pela determinação, caráter,
empenho e dedicação em minha
formação e com quem sempre
tanto aprendo.

AGRADECIMENTOS

Ao IPEN-CNEN/SP por permitir a conclusão deste trabalho e ao CEN, por apoiar o desenvolvimento acadêmico de seus colaboradores.

Ao sempre colega Prof. Dr. Mário Guerra Boaratti, pela confiança.

Ao Prof. Dr. Delvonei Alves de Andrade, pela paciência, competência e valorosa orientação prestada ao desenvolvimento deste trabalho.

Aos Profs. Mauro da Silva Dias, Roberto Navarro de Mesquita, Gaiianê Sabundjian, Helio Yoriyaz e Pedro Umbehaun, pelo incentivo e apoio, com quem tive o privilégio de aprender muito.

Aos colegas do CEN, em especial Alfredo José Alvim de Castro, Álvaro Luis Guimarães Carneiro, Roberto Carlos dos Santos e Paulo Henrique Ferraz Masotti pela amizade, companheirismo e compreensão nos meus dias de mau humor.

Aos funcionários do reator IEA-R1, em especial ao Sr. Walter Ricci Filho, pelas informações fornecidas.

Aos colegas da Marinha, em especial ao Sr. Vadin Surkov, pela prontidão em me atender.

Aos colegas da Aeronáutica, em especial aos Doutores Alexandre David Caldeira e Brett Vern Carlson, pela contagiante paixão às partículas sub-atômicas carregadas.

À Dra. Sônia Geraij Mokarzel, fantástica professora de Física, incansável mesmo nos momentos mais difíceis e que sempre me incitou a estudar um pouco mais do que eu queria.

A meu ex-professor e amigo João Usberco, extraordinário professor de Química e minha eterna referência na arte de lecionar.

À minha saudosa tia Aida Eid, pela convicção inconsciente de simplicidade, honradez e honestidade.

A meu saudoso, notável, incomparável e inesquecível tio, Farid Eid, por, mesmo após mais de duas décadas e meia de sua partida, se fazer sempre presente em minhas atitudes.

A todos meus alunos pela inocência, determinação e incentivo, velados.

Àqueles que, direta ou indiretamente, contribuíram para que este trabalho se realizasse.

A Deus, por tudo!

MODELO HÍBRIDO DE BANCO DE DADOS RELACIONAL, DE ALTO DESEMPENHO E CAPACIDADE DE ARMAZENAMENTO, PARA APLICAÇÕES VOLTADAS À ENGENHARIA NUCLEAR

José Gomes Neto

RESUMO

O objetivo deste trabalho é apresentar o banco de dados relacional, denominado FALCAO, que foi criado e implementado com a função de armazenar as variáveis monitoradas no reator de pesquisa IEA-R1, localizado no Instituto de Pesquisas Energéticas e Nucleares, IPEN – CNEN/SP.

O modelo lógico de dados e sua influência direta na integridade da informação fornecida são cuidadosamente considerados.

São apresentados os conceitos e etapas de normalização e desnormalização, incluindo as entidades e relacionamentos do modelo lógico de dados. São também apresentadas as influências dos relacionamentos e regras do modelo de dados nos processos de aquisição, carga e disponibilização da informação final, sob a óptica do desempenho, visto que estes processos ocorrem em lotes e em pequenos intervalos de tempo.

A aplicação SACD, através de suas funcionalidades, apresenta as informações armazenadas no banco FALCAO de maneira prática e otimizada.

A implementação do banco de dados FALCAO ocorreu com o êxito esperado, mostrando-se indispensável ao cotidiano dos pesquisadores envolvidos por conta da substancial melhoria dos processos e da confiabilidade associada a estes.

RELATIONAL DATABASE HYBRID MODEL, OF HIGH PERFORMANCE AND STORAGING CAPACITY, FOR NUCLEAR ENGINEERING APPLICATIONS

José Gomes Neto

ABSTRACT

The objective of this work is to present the relational database, named FALCAO. It was created and implemented to support the storing of the monitored variables in the IEA-R1 research reactor, located in the Instituto de Pesquisas Energéticas e Nucleares, IPEN – CNEN/SP.

The data logical model and its direct influence in the integrity of the provided information are carefully considered.

The concepts and steps of normalization and denormalization including the entities and relations involved in the logical model are presented. It is also presented the effects of the model rules in the acquisition, loading and availability of the final information, under the performance concept since the acquisition process loads and provides lots of information in small intervals of time.

The SACD application, through its functionalities, presents the information stored in the FALCAO database in a practical and optimized form.

The implementation of the FALCAO database occurred successfully and its existence leads to a considerably favorable situation. It is now essential to the routine of the researchers involved, not only due to the substantial improvement of the process but also to the confiability associated to it.

1 INTRODUÇÃO.....	4
1.1 HISTÓRICO DA UTILIZAÇÃO DE BANCOS DE DADOS.....	5
1.2 BANCOS DE DADOS DA ÁREA NUCLEAR NO BRASIL.....	7
1.3 ITENS DO TRABALHO	8
2 OBJETIVOS.....	10
2.1 MOTIVAÇÃO DO TRABALHO	10
3 REVISÃO BIBLIOGRÁFICA.....	12
3.1 ESTUDOS PERTINENTES.....	13
3.2 PROGRAMAS E SISTEMAS RELACIONADOS	15
3.3 SUBSÍDIOS E CONCLUSÕES FORNECIDAS PELA REVISÃO BIBLIOGRÁFICA	16
4 MATERIAIS E MÉTODOS.....	17
4.1 PARTE CONCEITUAL.....	17
4.1.1 <i>Fator integridade</i>	18
4.1.1.1 <i>Modelagem dos dados</i>	18
4.1.2 <i>Fator desempenho</i>	26
4.1.3 <i>Fator disponibilidade</i>	27
4.1.4 <i>Fator invulnerabilidade</i>	27
4.1.5 <i>Coexistência dos quatro fatores fundamentais</i>	27
4.1.6 <i>O Modelo de dados do Banco FALCAO</i>	28
4.1.7 <i>Banco de dados FALCAO versus os quatro fatores fundamentais</i>	30
4.2 PROGRAMAS UTILIZADOS	31
4.2.1 <i>Gerenciador de banco de dados ORACLE</i>	31
4.2.2 <i>Sistema operacional Linux Fedora Core 4</i>	31
4.2.3 <i>Linguagem de programação JAVA</i>	32
4.3 IMPLEMENTAÇÃO DA INFRA-ESTRUTURA	32
4.4 PARTE EXPERIMENTAL.....	32
4.4.1 <i>Implementação do modelo de dados</i>	33
4.4.2 <i>Conectividade ao banco de dados</i>	34
4.4.3 <i>O processo de transmissão, consistência e carga dos dados</i>	36
4.4.4 <i>As variáveis carregadas no banco de dados FALCAO</i>	37
4.4.5 <i>O processo de disponibilização dos dados</i>	44
4.4.5.1 <i>Acesso aos dados carregados através da aplicação SACD</i>	44
4.4.5.2 <i>Outras formas de acesso aos dados carregados</i>	46
4.4.6 <i>Dificuldades encontradas</i>	47
5 RESULTADOS	49
5.1 O SERVIDOR FALCAO	49

	2
5.1.1 Afinidade das peculiaridades do SGBD Oracle 10G com este trabalho.....	51
5.1.2 Otimização de acesso aos dados versus Desempenho da aplicação SACD.....	52
5.2 A TRANSFERÊNCIA DOS DADOS	52
5.3 A CARGA DOS DADOS.....	53
5.3.1 Procedimento periódico de garantia de recuperação dos dados.....	55
5.4 A APLICAÇÃO SACD	55
6 ANÁLISE E DISCUSSÃO DOS RESULTADOS	58
6.1 COTIDIANO DAS ATIVIDADES SEM A UTILIZAÇÃO DOS RECURSOS GERADOS POR ESTE TRABALHO.....	58
6.2 COTIDIANO DAS ATIVIDADES UTILIZANDO OS RECURSOS GERADOS POR ESTE TRABALHO.....	60
7 CONCLUSÕES.....	71
7.1 A PROPOSTA ATUAL	71
7.2 CONTINUIDADE DO TRABALHO	72
APÊNDICE A – ARQUIVOS RESPONSÁVEIS PELA GERAÇÃO DAS TABELAS, RESTRIÇÕES, PARÂMETROS DE LOGON E RELACIONAMENTOS DO BANCO DE DADOS FALCAO	73
APÊNDICE B - FONTES DOS PROGRAMAS QUE CONSISTEM, TRANSMITEM, CARREGAM E DISPONIBILIZAM OS DADOS ORIUNDOS DO REATOR IEA-R1 PARA O BANCO DE DADOS FALCAO, NO CEN.....	78
ANEXO A – ASPECTOS BÁSICOS DA LINGUAGEM SQL APLICADOS A ESTE TRABALHO.....	85
ANEXO B – PROCEDIMENTOS PARA INSTALAÇÃO E CUSTOMIZAÇÃO DOS PRODUTOS ORACLE 10G, JAVA 5.0 E TOMCAT 5.5.....	87
REFERÊNCIAS BIBLIOGRÁFICAS.....	94

LISTA DE TABELAS

TABELA 1– CONVERSÕES DE NOMENCLATURAS PARA O REATOR IEA-R1	40
TABELA 2– COMPARATIVO DE <i>HARDWARE</i> DO SERVIDOR FALCAO	49

LISTA DE FIGURAS

FIGURA 1 - FORMAS DE REPRESENTAÇÃO DO RELACIONAMENTO CONCEITUAL 1:N	19
FIGURA 2 – MODELO CONCEITUAL DO BANCO DE DADOS FALCAO	29
FIGURA 3 – DER DO BANCO DE DADOS FALCAO	30
FIGURA 4- ARQUIVO DE CONFIGURAÇÃO DA CONECTIVIDADE DOS CLIENTES <i>ORACLE</i> AO SGBD <i>ORACLE</i>	36
FIGURA 5 – DIAGRAMA – COLETA DAS VARIÁVEIS TEMPERATURA E VAZÃO NO PRIMÁRIO IEA-R1	38
FIGURA 6 – DIAGRAMA – ARREFECIMENTO DO SECUNDÁRIO DO IEA-R1	39
FIGURA 7 – TEMPERATURA - EDITOR NATIVO SQL-PLUS.....	46
FIGURA 8 – TEMPERATURA – UTILIZAÇÃO DA FERRAMENTA TOAD	47
FIGURA 9 - MONITORAÇÃO DO BANCO DE DADOS FALCAO ATRAVÉS DO SPOTLIGHT	50
FIGURA 10 – LISTAGEM DO ARQUIVO QUE REGISTRA O PROCESSO DE TRANSFERÊNCIA DOS DADOS.....	53
FIGURA 11 - REGISTRO DA CARGA DO ARQUIVO DE TEMPERATURA NO BANCO DE DADOS FALCAO.....	54
FIGURA 12 - TELA DE AUTENTICAÇÃO - SACD.....	56
FIGURA 13 - TELA PRINCIPAL DA APLICAÇÃO SACD	56
FIGURA 14 - TELA INICIAL TERMO-HIDRÁULICA.....	57
FIGURA 15 - TELA INICIAL TERMO-HIDRÁULICA - DETALHAMENTO	57
FIGURA 16 - ARQUIVO TIPO TXT GERADO PELO SISTEMA SAD	58
FIGURA 17 - PLANILHA GERADA MANUALMENTE.....	59
FIGURA 18 – GRÁFICO DA TEMPERATURA DO PRIMÁRIO DO REATOR IEA-R1, GERADO MANUALMENTE.....	59
FIGURA 19- TEMPERATURA - CIRCUITO PRIMÁRIO DO REATOR IEA-R1 VIA SACD	60
FIGURA 20- INFORMAÇÕES EM FORMATO TEXTO, UTILIZADAS NA GERAÇÃO DO GRÁFICO DA FIG. 19.....	61
FIGURA 21- DADOS UTILIZADOS NA GERAÇÃO DO GRÁFICO DA FIG. 19	61
FIGURA 22 - POTÊNCIA DO CIRCUITO PRIMÁRIO DO REATOR IEA-R1 VIA SACD	62
FIGURA 23 - POTÊNCIA DO CIRCUITO PRIMÁRIO DO IEA-R1 - PERÍODOS DISTINTOS VIA SACD	62
FIGURA 24- POTÊNCIA DOS CIRCUITOS PRIMÁRIO E SECUNDÁRIO, PARA O MESMO PERÍODO, VIA SACD	63
FIGURA 25- TEMPERATURA DA ÁGUA DA PISCINA DO REATOR IEA-R1	64
FIGURA 26 – FUNÇÃO REATIVIDADE	66
FIGURA 27 – POSIÇÃO CRÍTICA NA CURVA <i>S</i>	67
FIGURA 28 – <i>Er</i> INICIAL DO ARRANJO 234, VIA O SISTEMA SACD	68
FIGURA 29 – <i>Er</i> DO ARRANJO 234, EM 26/12/2007.....	69
FIGURA 30 - VALORES DE CORRENTE DOS CANAIS DE POTÊNCIA DO REATOR IEA-R1, EM 26/12/2007	69
FIGURA 31 - TELA DO INSTALADOR DO SGBD <i>ORACLE</i> 10G.....	89
FIGURA 32 - ESTADO DE ATIVAÇÃO DO <i>LISTENER</i>	90
FIGURA 33 - TELA INICIAL DA INSTALAÇÃO DO JAVA 5.0	91
FIGURA 34 - TELA FINAL DA INSTALAÇÃO DO JAVA 5.0.....	92
FIGURA 35- TELA INICIAL DA INSTALAÇÃO DO <i>TOMCAT</i> 5.5.23	93
FIGURA 36 - TELA FINAL DA INSTALAÇÃO DO <i>TOMCAT</i> 5.5.23	93

1 INTRODUÇÃO

Com a sinalização do governo federal na direção da retomada da construção de novas usinas nucleares no Brasil, como importante fonte de energia limpa, evidenciam-se cada vez mais processos críticos e caros, tais como processos nucleares, petroquímicos, siderúrgicos e outros sistemas industriais complexos que geram uma grande quantidade de informações, muitas das vezes de forma redundante por segurança, mas que invariavelmente necessitam ter seus históricos armazenados com eficiência para uma consulta, comparação, estudo ou ação futura. Informações distorcidas, mal armazenadas, obsoletas ou inválidas podem comprometer gravemente as conclusões sobre estes processos ou até mesmo, em casos interativos, o funcionamento adequado destes.

Prováveis perdas econômicas e principalmente questões de segurança apontam para a necessidade de se implementar sistemas eficientes de armazenamento e disponibilização de dados.

Desta forma, é de grande relevância para processos complexos, um confiável histórico de medidas, adequadamente formatado, prontamente utilizável, de fácil acesso, capaz de armazenar e disponibilizar as informações, e que seja também disponível para grandes volumes de dados em pequenos intervalos de tempo, com informações íntegras, regras claras e versáteis o suficiente para permitirem conclusões e/ou ações rápidas. Seguindo por esta linha de raciocínio, uma forma criteriosa, segura e eficiente de armazenar e disponibilizar a informação mediante demanda, é o *Relational Data Base Management System*, *RDBMS*, conhecido no idioma português por Sistema Gerenciador de Banco de Dados, SGBD, ou ainda simplesmente, banco de dados. Esta eficaz e consagrada tecnologia, largamente utilizada principalmente nos âmbitos comercial e industrial, armazena e provê a informação de forma segura, programável, íntegra e padronizada, por intermédio de mecanismos eficientes e adequáveis a cada necessidade, com quesitos de segurança implícitos confiáveis e de extrema utilidade, versatilidade e eficiência. O banco de dados consiste basicamente de uma estrutura principal subdividida em estruturas menores ligadas entre si, fisicamente dispostas em arquivos gravados em meios magnéticos e logicamente integrados por memórias do tipo *Randomic Access Memory*, *RAM*, cuja função é garantir fisicamente a segurança da informação e logicamente a integridade e

disponibilidade desta, de forma a conferir acesso a quem consulta e respaldo a quem administra o banco de dados.

1.1 HISTÓRICO DA UTILIZAÇÃO DE BANCOS DE DADOS

Na década de sessenta, os sistemas computacionais, embora ainda embrionários sob o ponto de vista da difusão, já davam mostras de que grandes volumes de informação precisavam ser tratados de forma mais criteriosa, para que sua integridade fosse preservada e seu acesso garantido, lhes conferindo assim credibilidade. Sendo assim e contando com a fundamental evolução do *hardware*, cada vez mais a tendência de armazenar grandes quantidades de informação foi se consolidando, até que no final dos anos sessenta, grandes fabricantes de máquinas como IBM, Borroughs, G.E., Unisys etc. Aos poucos disponibilizavam ao cenário internacional, já então caracterizado pelo uso de computadores de grande capacidade de processamento, produtos mais especificamente voltados ao armazenamento e disponibilização da informação, visto que em áreas estratégicas distintas, como indústria automobilística, siderúrgicas e instituições financeiras, a necessidade era iminente.

Desta forma, como contemporâneo do arquivo indexado *Virtual Storage Access Method*, *VSAM*, **CÂNDIDO [2007]**¹, tecnologia mais utilizada até então para armazenamento e disponibilização de grandes volumes, surgia o *software*, banco de dados, produto que por intermédio de instruções peculiares normalizadas, permitia que basicamente quatro eventos ocorressem, não simultaneamente: Gravação, leitura, atualização e destruição do dado, de forma global ou pontual. Permitia também, caso houvesse alguma falha de caráter físico com a máquina, ou lógico, durante a manipulação dos dados, que a informação fosse posteriormente recuperada por intermédio de mecanismos de cópias prévias ou ainda de expurgo controlado de dados. Consolidava-se assim o banco de dados hierárquico, **IBM [1998]**² cujo engajamento no cenário computacional mundial foi único, mesmo porque nada similar existia disponível neste sentido.

Entretanto, os sistemas comerciais de computador sinalizavam para a necessidade de uma forma de armazenamento mais simples no que tange aos códigos de acesso, de algo mais versátil no que se refere aos recursos físicos mínimos necessários e, sobretudo, de um produto mais acessível economicamente, mas que também fosse eficiente e confiável.

Surge assim, no início dos anos setenta, o conceito de banco de dados relacional, **ALVES [2004]**³, ferramenta de fácil manuseio, eficiente e

sensivelmente mais acessível financeiramente quando comparada aos bancos hierárquicos utilizados até então, embora também capaz de armazenar e disponibilizar grandes volumes de informação e também contemplar os mecanismos de recuperação e expurgo.

O banco relacional se consolidou como substituto natural do banco hierárquico em praticamente cem por cento das aplicações, mas o advento dos computadores pessoais no início dos anos oitenta mudou radicalmente este cenário sob o ponto de vista do *hardware*, onde sistemas de armazenamento de dados se faziam necessários em equipamentos ditos pessoais. Surge então o conceito de concentradores, cujo *hardware* era dotado de componentes especiais, denominados microprocessadores. Estas máquinas eram equipadas, sob o ponto de vista do *software*, com produtos compatíveis aos existentes na então denominada alta plataforma, porém muito mais acessíveis sob todos os aspectos, incluindo os SGBDs.

Dez anos mais tarde, meados dos anos noventa, com o advento quase que simultâneo de sistemas operacionais providos de interfaces gráficas aliados ao surgimento do *World Wide Web*, *WWW*, da internet, tornou-se imperativo nestes servidores a existência de sistemas gerenciadores de banco de dados que pudessem disponibilizar a informação de forma íntegra e rápida. Estes servidores, também proviam recursos de restabelecimento, monitoração e controle dos dados aos moldes de ambientes de grande porte.

Desta forma, hoje existem sistemas gerenciadores de banco de dados abrigados em equipamentos de custo acessível quando comparados aos equipamentos de grande porte, interligados por redes internas e/ou externas e que movimentam grandes volumes de informação. Esta tendência se confirma no âmbito dos SGBDs com a recente implementação da tecnologia *grid*. Tal tecnologia objetiva majoritariamente interligar equipamentos simples sob a óptica da capacidade de processamento, transformando-os em um supercomputador. Esta tecnologia é encontrada no SGBD que apóia este trabalho.

1.2 BANCOS DE DADOS DA ÁREA NUCLEAR NO BRASIL

Por conta das características peculiares dos fenômenos nucleares, ou seja, grande variedade e volume de informações monitoradas, aliadas a confiabilidade e sigilo necessários na ocasião de seu armazenamento e disponibilização, um sistema gerenciador de banco de dados na área nuclear precisa, sobretudo, contemplar satisfatória e concomitantemente estas características. Assim, além da capacidade de armazenar e disponibilizar grandes volumes de dados, de forma eficiente e sigilosa, é absolutamente necessário mecanismos que permitam a recuperação em tempo hábil destes dados, no caso de eventual falha ou indisponibilidade.

Sistemas pesquisados na área nuclear, e que utilizam o produto banco de dados como repositórios de informações, hoje não raramente encontrados, mostram claramente tanto na sua concepção como em sua utilização, as nítidas diferenças entre sistemas da área nuclear que tratam dados não nucleares e sistemas da área nuclear que tratam dados nucleares.

Estes primeiros sistemas, ou seja, da área nuclear tratando dados não nucleares, são caracteristicamente maleáveis, versáteis, interativos e intuitivos, logo comparáveis a sistemas comerciais convencionais.

O sistema direcionado a *marketing*, **BARROS [1996]⁴**, utilizado pelo Centro de Informações Nucleares, CIN, desde meados da década passada, é uma evidência clara da aplicação destas características. Embora este sistema utilize a ferramenta banco de dados como apoio ao *marketing* de relacionamento, mostra pelas suas peculiaridades que as características do sistema gerenciador de banco de dados são diretamente associadas ao tipo de informação armazenada no banco e não necessariamente à instituição que provém.

Também da área nuclear, mas se opondo a este primeiro exemplo no que tange as características principais, outrora citadas, destacamos o sistema denominado *Computer Aided Search System*, **COMPASS [1997]⁵**.

Este sistema, da mesma forma que o primeiro, é aplicado na tecnologia nuclear, porém diferentemente deste, manipula dados exclusivamente nucleares.

Desta forma, percebe-se, tanto na sua concepção quanto na disponibilização dos seus dados, características anteriormente citadas, peculiares aos sistemas nucleares.

Como o sistema *COMPASS*, podemos citar estudos que também se caracterizam pela utilização de dados nucleares, **RIBEIRO [2005]**⁶, **CARVALHO [2007]**⁷, ligados a tecnologia nuclear, e que também utilizam a ferramenta banco de dados como repositório de informações.

Porém, para que um sistema seja seguro e sigiloso, não precisa necessariamente ser limitado em suas funcionalidades, como os sistemas anteriormente citados.

Um modelo de dados eficiente que garante a integridade da informação, como o proposto neste trabalho, e regras de relacionamento conferem versatilidade e desempenho satisfatórios. Assim, se ganha duplamente ao unir as duas vertentes, ou seja, fazer com que funcionalidade e segurança coexistam.

Deseja-se então, sobretudo na área nuclear, um sistema gerenciador de banco de dados seguro, compatível com sistemas já existentes e versátil o suficiente para atender prontamente novas demandas, sem que para isso limitações como a necessidade da compra de novas versões ou um tempo de reprogramação muito dilatado sejam necessários.

1.3 ITENS DO TRABALHO

Neste Capítulo consta o histórico da utilização de bancos de dados e também exemplos da utilização destes na área nuclear.

No Capítulo dois da dissertação, são apresentados o objetivo principal e a motivação do trabalho.

No Capítulo três está descrita a revisão da literatura, incluindo trabalhos relacionados ao tema e sistemas já existentes ligados a este trabalho, assim como os comentários e subsídios gerados pela revisão bibliográfica.

O Capítulo quatro contém o embasamento teórico assim como a metodologia empregada neste trabalho, permitindo, a futuros trabalhos desta natureza, a seqüência de procedimentos necessários a implementação destes.

No Capítulo cinco, apresenta-se os resultados obtidos com a implementação detalhada do Capítulo quatro, associando-os diretamente às peculiaridades dos programas envolvidos e a customizações destes.

No Capítulo seis, está a análise e discussão dos resultados obtidos. Também está neste Capítulo a comparação dos cenários que precederam e sucederam este trabalho.

Finaliza-se o trabalho no Capítulo sete com as conclusões e propostas para trabalhos futuros.

O Apêndice A apresenta os arquivos responsáveis pela implementação física do modelo de dados, assim como os parâmetros que incluem novos usuários no sistema desenvolvido neste trabalho.

O Apêndice B apresenta o código-fonte dos programas de consistência, transmissão, carga e disponibilização dos dados oriundos do reator IEA-R1.

O Anexo A traz uma visão geral da linguagem *Structure Query Language, SQL*, utilizada para manipular dados em bancos de dados relacionais, como o banco de dados FALCAO, utilizado neste trabalho.

O Anexo B mostra procedimentos básicos para a instalação dos produtos *ORACLE*, em sistema operacional Linux Fedora e *JAVA*, em sistema operacional Windows XP Service Pack 2, assim como a customização destes à realidade deste trabalho.

2 OBJETIVOS

O objetivo principal deste trabalho é prover, por intermédio do SGBD *ORACLE* 10G, acoplando a este uma aplicação *Web*, subsídios necessários e suficientes para que, via um histórico íntegro de dados e representativo das variáveis monitoradas cotidianamente no reator IEA-R1, seja possível efetuar comparações, obter dados, estimar resultados e sobretudo ganhar tempo e qualidade acerca de conclusões sobre fenômenos particulares e/ou correlatos que estejam ligados direta ou indiretamente a estas variáveis.

2.1 MOTIVAÇÃO DO TRABALHO

Observando-se o cotidiano dos pesquisadores que necessitam direta ou indiretamente de informações relacionadas às variáveis de estado ou grandezas derivadas destas variáveis, oriundas do reator IEA-R1, percebe-se a necessidade de uma base de dados consolidada, com informações recentes e/ou históricas para permitirem conclusões e viabilizarem, através do cruzamento destas informações, ganho substancial de tempo.

Assim, a principal motivação deste trabalho baseia-se na necessidade de um sistema gerenciador de banco de dados híbrido, maleável sob o ponto de vista de novas necessidades, com baixo custo, de fácil implementação e utilização, confiável no que se refere à recuperação dos dados, com alta capacidade de armazenamento e disponibilização, com informações compartilháveis e capaz de interagir com sistemas já existentes.

Também motiva o desenvolvimento deste trabalho a atual demanda de trabalhos ligados a áreas afins como máquinas rotativas, **BENEVENUTTI [2004]**⁸, diagnóstico de falhas por monitoração, **GONÇALVES [2006]**⁹, modelagens analíticas, **BOARATTI [2006]**¹⁰, neutrônica, termo-hidráulica etc. Estes trabalhos em geral não possuem um histórico único e próprio de medidas, não podendo ao menos comparar dados com áreas equivalentes ou até mesmo evitar retrabalho na aquisição destes.

Hoje em dia, os poucos sistemas que armazenam este tipo de dado, ou não estão abertos a consultas, ou não possuem a informação desejada, quer por serem dedicados a apenas um tipo de assunto, como o sistema *COMPASS*, quer

por não permitirem compartilhamento dos dados de forma eficiente, como o sistema denominado Sistema de Aquisição de Dados, SAD, **ELIPSE SCADA**¹¹.

Desta forma é de extrema utilidade para os pesquisadores, principais usuários destas informações, a existência de um sistema gerenciador de banco de dados integrado, que forneça informações sobre as variáveis monitoradas no reator de pesquisas IEA-R1 de forma confiável e que também permita a disponibilização destas informações de forma eficiente assim como a comparação destas. Este sistema gerenciador de banco de dados constitui o objetivo maior deste trabalho.

Tal repositório, acoplado a um sistema *Web*, disponibiliza de forma segura e automática as informações geradas no reator de pesquisas IEA-R1, permitindo aos pesquisadores acesso a estas, otimizando assim suas pesquisas e conseqüentemente o tempo gasto nelas na ocasião da aquisição de dados, comparações e/ou estudos de tendências.

3 REVISÃO BIBLIOGRÁFICA

Ocasões como a necessidade de comparação do comportamento de um determinado fenômeno ou componente, cuja monitoração tenha uma grande frequência de amostragem diária, característica peculiar de processos nucleares, ou ainda a covariância entre variáveis monitoradas de forma independente, mas fortemente relacionadas, como por exemplo, a vida útil de um determinado componente em função da variação do comportamento de uma variável de estado indiretamente ligada a ele, motiva a pesquisa de soluções.

Particularmente a este estudo, estas soluções devem ser versáteis o suficiente para atenderem as necessidades operacionais mais básicas do cotidiano do reator de pesquisas IEA-R1, como uma simples pesquisa aos dados neste monitorados, mas também precisam prover recursos confiáveis o suficiente para concluir-se, sem perda de tempo e com clareza, acerca do comportamento específico ou correlacionado de um determinado fenômeno, complexo ou não.

A pesquisa bibliográfica realizada teve como principal objetivo a verificação da existência de programas, processos ou sistemas que contemplassem estas necessidades da área nuclear, mais especificamente ligados as variáveis monitoradas no reator de pesquisas IEA-R1. Tal pesquisa revelou resultados diversos, alguns destes voltados a propósitos mais genéricos, outros com peculiaridades mais próximas desta proposta, outros completamente desconectados dos propósitos deste trabalho.

Dentre estes, foram encontrados alguns programas utilitários, ou seja, de uso genérico, oriundos da década de 80.

Na área de mecânica dos fluidos computacional, foram encontrados os programas denominados *Computational Fluid Dynamic*, *CFD*, **DINÂMICA DOS FLUIDOS COMPUTACIONAL**¹².

Dentre eles, pode-se citar: *FLOTRAN*, **ANSYS**[1997]¹³, *FLUENT*, **FLUENT**¹⁴, *PHOENICS*, **PHOENICS**¹⁵ e *CFX*, **CFX**¹⁶. Basicamente estes programas consistem em utilizar métodos computacionais para predição quantitativa de características de escoamento, contemplando mudança de fase, transferência de calor, transferência de massa, reações químicas e tensões de deslocamento em sólidos imersos ou circundantes.

Todavia, embora manipulem dados nucleares, foram desenvolvidos para resolver problemas localizados, ou seja, resolvem a dinâmica de fluidos muito bem em um trecho do circuito ou em um determinado componente dele e são aplicados principalmente na solução de problemas com escoamento de fluidos monofásicos ou multifásicos, em um trecho, ou em um componente específico de uma instalação industrial.

Na área de neutrônica computacional, foram encontrados códigos específicos como *LEOPARD*, **LEE e ZHANG**¹⁷, *CITATION*, **FOWLER, VONDY e CUNNINGHAM**¹⁸ e *MCNP*, **BRIESMEISTER**¹⁹. Embora imprescindíveis à neutrônica, pouco se relacionam com os propósitos deste trabalho.

Estes programas, tanto da área de termo-hidráulica como da área de neutrônica, são largamente utilizados em plantas nucleares, mas não utilizam bases de dados corporativas e também não se dedicam exclusivamente ao armazenamento e compartilhamento de informações geradas no reator de pesquisas IEA-R1.

Com resultados ainda não satisfatórios, uma revisão bibliográfica mais refinada e direcionada aos propósitos deste trabalho foi então realizada, objetivando identificar trabalhos e/ou sistemas que utilizam e recomendam mecanismos seguros para prover informações voltadas à área nuclear ou que sejam adequados às necessidades do cotidiano dos pesquisadores do IPEN que utilizam as informações geradas no reator de pesquisas IEA-R1. Esta pesquisa engloba estudos e sistemas, ligados direta ou indiretamente aos propósitos deste trabalho, dos últimos onze anos.

Os resultados desta pesquisa apontam para três estudos e dois sistemas aplicativos, entre dissertações, periódicos e sistemas comerciais, que mais se aproximam desta proposta, cronologicamente detalhados a seguir.

3.1 ESTUDOS PERTINENTES

O primeiro estudo analisado trata aspectos interessantes da evolução do *marketing* tradicional como ciência, e atrela sua sobrevivência à inevitável associação desta aos bancos de dados de relacionamentos, como citado em [4].

Vale dizer aqui, que, o termo relacionamento não se refere ao fato do banco de dados utilizado ser relacional e sim, ao histórico dos relacionamentos interpessoais da ciência *marketing*.

O estudo descreve as experiências do CIN, da Comissão Nacional de Energia Nuclear, CNEN, na construção e utilização de um banco de dados para

fins de *marketing*. Mostra as vantagens de apelos comerciais mais convincentes baseados em informações reais e não em suposições, diferencia o *database marketing*, estilo de fazer negócios, do *marketing database*, uma inovação tecnológica decorrente de avanços da tecnologia da informação. Estabelece estratégias tangíveis por conta de experiências bem sucedidas baseadas em informações fidedignas, caracteriza o valor dos bancos de dados, associando-o diretamente a seu conteúdo e conclui enfatizando que o sucesso de atividades baseadas em dados, científicas ou não, está diretamente ligado ao desenvolvimento criativo de aplicações competitivas, estrategicamente significantes e a um custo permissível. Estreitamente ligado a área nuclear e utilizando, como neste trabalho, o produto banco de dados para armazenar o registro de suas informações, o estudo do CIN/CNEN preocupa-se exclusivamente com o uso de bancos de dados para analisar o comportamento da ciência *marketing* e, desta forma, em nada se relaciona aos processos nucleares e as variáveis que estes geram no reator de pesquisas IEA-R1, mas chama a atenção por ter suas premissas básicas ainda perfeitamente válidas, mesmo após onze anos de publicação.

Um segundo estudo, como citado em [6], objetiva apresentar a aplicabilidade da inferência *bayesiana*, consagrada técnica matemática, em estudos de confiabilidade, usando-a para especializar dados de falha em análises de segurança, demonstrando o impacto do uso da mesma em estudos de análises de riscos ambientais em plantas industriais de processo, assim como o seu uso em análises probabilísticas de segurança em instalações nucleares.

Neste trabalho, embora os dados ditos "de falha" sejam informações fortemente ligadas a influência do uso da inferência *bayesiana*, a forma como eles são gerados ou providos não caracterizam um sistema computacional compartilhado, cujo objetivo principal é disponibilizar a informação. Este estudo se propõe a utilizar estes dados como parte de um processo que objetiva exclusivamente segurança e não a comparação de informações.

Também ligado a área nuclear e direcionado à área de qualidade e confiabilidade, preocupa-se com uma determinada massa de dados mas sobretudo, associa a eficiência do uso desta massa de dados à forma como foi gerada. Embora estas informações também sejam armazenadas, não possuem ligação alguma com as variáveis monitoradas no reator de pesquisas IEA-R1.

Um terceiro estudo, como citado em [7], aborda a revisão crítica do emprego de banco de dados de falhas em análises probabilísticas da segurança

fins de *marketing*. Mostra as vantagens de apelos comerciais mais convincentes baseados em informações reais e não em suposições, diferencia o *database marketing*, estilo de fazer negócios, do *marketing database*, uma inovação tecnológica decorrente de avanços da tecnologia da informação. Estabelece estratégias tangíveis por conta de experiências bem sucedidas baseadas em informações fidedignas, caracteriza o valor dos bancos de dados, associando-o diretamente a seu conteúdo e conclui enfatizando que o sucesso de atividades baseadas em dados, científicas ou não, está diretamente ligado ao desenvolvimento criativo de aplicações competitivas, estrategicamente significantes e a um custo permissível. Estreitamente ligado a área nuclear e utilizando, como neste trabalho, o produto banco de dados para armazenar o registro de suas informações, o estudo do CIN/CNEN preocupa-se exclusivamente com o uso de bancos de dados para analisar o comportamento da ciência *marketing* e, desta forma, em nada se relaciona aos processos nucleares e as variáveis que estes geram no reator de pesquisas IEA-R1, mas chama a atenção por ter suas premissas básicas ainda perfeitamente válidas, mesmo após onze anos de publicação.

Um segundo estudo, como citado em [6], objetiva apresentar a aplicabilidade da inferência *bayesiana*, consagrada técnica matemática, em estudos de confiabilidade, usando-a para especializar dados de falha em análises de segurança, demonstrando o impacto do uso da mesma em estudos de análises de riscos ambientais em plantas industriais de processo, assim como o seu uso em análises probabilísticas de segurança em instalações nucleares.

Neste trabalho, embora os dados ditos "de falha" sejam informações fortemente ligadas a influência do uso da inferência *bayesiana*, a forma como eles são gerados ou providos não caracterizam um sistema computacional compartilhado, cujo objetivo principal é disponibilizar a informação. Este estudo se propõe a utilizar estes dados como parte de um processo que objetiva exclusivamente segurança e não a comparação de informações.

Também ligado a área nuclear e direcionado à área de qualidade e confiabilidade, preocupa-se com uma determinada massa de dados mas sobretudo, associa a eficiência do uso desta massa de dados à forma como foi gerada. Embora estas informações também sejam armazenadas, não possuem ligação alguma com as variáveis monitoradas no reator de pesquisas IEA-R1.

Um terceiro estudo, como citado em [7], aborda a revisão crítica do emprego de banco de dados de falhas em análises probabilísticas da segurança

de plantas nucleares e químicas. A principal diferença, quando comparado com o estudo imediatamente anterior, se deve ao fato deste terceiro estudo partir de uma base de dados de falhas preexistente e estabelecer critérios rigorosos na sua utilização, e não na sua geração.

Da mesma forma que o segundo, é ligado à área nuclear e utiliza banco de dados. Porém, diferentemente deste, onde a abordagem é feita sobre a qualidade da geração do dado, aborda segurança e faz uma revisão crítica, de forma probabilística, sobre o emprego de informações outrora geradas, ligadas a falhas em plantas nucleares e químicas.

3.2 PROGRAMAS E SISTEMAS RELACIONADOS

Os programas do tipo *CFD*, embora largamente utilizados no âmbito nuclear, mais especificamente na área de termo-hidráulica, destinam-se a propósitos mais específicos, como detalhados anteriormente. Desta forma, não são próprios para a realização de comparação entre variáveis monitoradas em uma planta nuclear e muito menos para correlacionar historicamente o comportamento das variáveis envolvidas nos processos nucleares do reator de pesquisas IEA-R1.

Com peculiaridades não integralmente dedicadas ao fim que este trabalho se propõe, dos sistemas pesquisados, os que mais se aproximam desta proposta são os sistemas *COMPASS*, como citado em [5], e *SAD*, como citado em [11], sendo que destes, apenas o sistema *COMPASS* possui base de dados relacional, teoricamente passível de compartilhamento.

O sistema *COMPASS*, de origem dinamarquesa, monitora exclusivamente a vibração das bombas moto-operadas do circuito primário do reator IEA-R1, gravando estas informações em um banco de dados relacional *ORACLE* versão 8i, **FANDERUFF [2000]**²⁰ e disponibilizando-as em forma de gráficos.

Trata-se de uma aplicação comercialmente fechada, ou seja, sem a possibilidade de implementar-se novas funcionalidades sem custo associado, com base de dados exclusivamente dedicada a ela e com informações dispostas de forma pré-definida e padronizada.

Qualquer nova demanda de funcionalidade, por menor que seja, implicará necessariamente em outra versão do sistema e, conseqüentemente, em custo adicional.

Já o sistema *SAD* provê o registro das principais variáveis monitoradas no reator de pesquisas IEA-R1 em arquivos do tipo texto. Diferentemente do

sistema *COMPASS*, embora monitore uma gama maior de variáveis, não possui base de dados relacional e não tem versatilidade suficiente para permitir comparações entre estas variáveis, dispostas originalmente em arquivos muito extensos e passíveis de retrabalho.

O sistema SAD, diferentemente do sistema *COMPASS*, não possui base de dados própria, mesmo que fechada, mas em compensação, registra as variáveis monitoradas no reator IEA-R1 e, embora passível de retrabalho e com limitações, as disponibiliza.

3.3 SUBSÍDIOS E CONCLUSÕES FORNECIDAS PELA REVISÃO BIBLIOGRÁFICA

Os estudos e sistemas outrora citados em muito colaboraram para o real direcionamento deste trabalho, servindo inclusive, no caso do sistema SAD, de subsídio a este. O uso da ferramenta SGBD para apoiar a tomada de decisões estratégicas, a preocupação com a integridade do dado de falha, gerado pela inferência *bayesiana*, os rigorosos critérios estatísticos de validação destes dados, o banco de dados dedicado exclusivamente a um determinado sistema proprietário e, sobretudo, as variáveis monitoradas por um sistema de aquisição de dados do reator IEA-R1 enfatizaram a real necessidade de um repositório de dados único. Assim, quer nos estudos outrora realizados, quer nos sistemas atualmente utilizados, percebe-se a iminente necessidade de um sistema computacional que se comunique com os sistemas existentes ligados às variáveis do reator de pesquisas IEA-R1, mas que também permita, via funcionalidades ausentes atualmente nos sistemas encontrados, o armazenamento e a disponibilização das informações de forma customizada, versátil, rápida e confiável, características estas deficientes e muitas das vezes ausentes nos poucos sistemas que atualmente se aproximam deste propósito.

A demanda atual, conseqüência direta das deficiências outrora citadas, justifica e condiciona o desenvolvimento deste trabalho.

4 MATERIAIS E MÉTODOS

4.1 PARTE CONCEITUAL

Ao longo dos últimos quarenta e cinco anos, a evolução contínua do *hardware*, alavancada pelo advento dos microprocessadores no início dos anos oitenta, vem provendo proporcional avanço aos sistemas que destes dependem, particularmente sistemas que interagem com dados. Para estes sistemas em particular, sobretudo para os que acessam grandes volumes de informações sigilosas, como no caso deste trabalho, é nítida a existência de quatro fatores implícitos, principais. Estes fatores independem do produto e/ou fabricante do SGBD, surgiram por demanda, separadamente, em épocas distintas, de forma complementar entre si e hoje, configuram conjunta e irreversivelmente os sistemas gerenciadores de bancos de dados. Na seqüência de surgimento, estes fatores são:

- integridade;
- desempenho;
- disponibilidade e
- invulnerabilidade.

Os fatores integridade e desempenho estão integralmente contidos no modelo de dados.

O modelo de dados é o processo através do qual, de forma ainda teórica, mensura-se as entidades do banco de dados e seus respectivos relacionamentos.

Estas entidades e seus relacionamentos traduzem as peculiaridades das informações que armazenarão e, conseqüentemente, sua confiabilidade. Num segundo momento, mais especificamente na ocasião da implementação física do modelo de dados, as entidades tornar-se-ão tabelas e os relacionamentos, restrições ou simplesmente, *constraints*. Neste momento, tabelas e *constraints* passam a ser tratadas fisicamente, e recebem o *status* de objeto de banco de dados. Já os fatores disponibilidade e invulnerabilidade independem do modelo lógico de dados.

O fator disponibilidade está ligado a infra-estrutura, onde cópias do banco de dados coexistem em máquinas diferentes para suprir eventual falha, física ou lógica, de alguma delas.

O fator invulnerabilidade, também ligado a infra-estrutura, está mais intimamente relacionado a algoritmos complexos de transmissão de dados, que visam, de forma resumida, identificar e impedir acessos indevidos aos dados.

A seguir, a abrangência destes fatores.

4.1.1 Fator integridade

Este fator é representado pelos relacionamentos e *constraints* existentes no modelo lógico de dados. É no modelo de dados, processo através do qual projetamos as entidades, que encontramos a integridade do banco de dados como um todo, ainda que teoricamente. A implementação do modelo de dados resulta invariavelmente no banco de dados físico, propriamente dito.

O banco de dados relacional, como o utilizado neste trabalho, é composto fisicamente de tabelas que, ao se relacionarem por intermédio de regras e *constraints*, contemplam a integridade da informação, sua unicidade. O processo através do qual a unicidade é garantida denomina-se normalização que, por sua vez, é composta de sub-etapas denominadas formas normais e pertence, em um âmbito superior, ao processo denominado modelagem de dados.

Assim, aplicando as formas normais, dizemos que o modelo lógico de dados está normalizado, garantindo com isso a unicidade de suas informações e por conseqüência, sua integridade.

4.1.1.1 Modelagem dos dados

A modelagem de dados é composta seqüencialmente por dois processos denominados modelagem conceitual ou teórica e modelagem lógica.

O modelo conceitual ou teórico serve de base para a construção do modelo lógico de dados, **TEOREY, LIGHTSTONE e NADEAU [2007]**²¹. Existem diferentes terminologias para representar-se a modelagem conceitual, mas basicamente, podemos observar três destas mais freqüentemente utilizadas.

Nos modelos conceituais, os relacionamentos podem assumir a condição de um para um, 1:1, um para muitos, 1:N ou muitos para muitos, N:N, porém jamais duas ou mais destas condições simultaneamente.

As notações mais freqüentemente utilizadas para um relacionamento do tipo 1:N são representadas na FIG. 1.

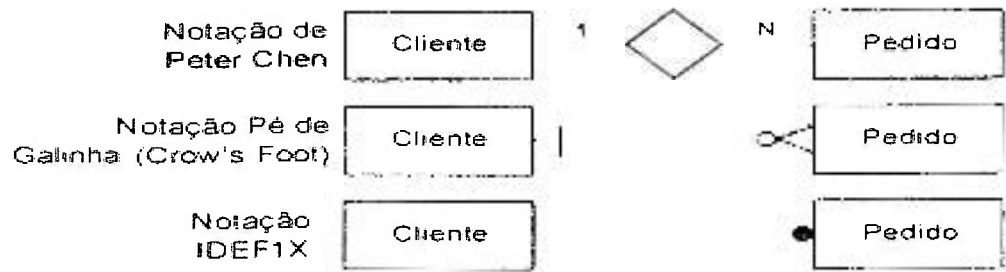


FIGURA 1 - Formas de representação do relacionamento conceitual 1:N

A primeira notação, de Peter Chen, **CHEN [1990]**²², é a mais utilizada para a construção do modelo conceitual enquanto a segunda e a terceira, vulgarmente conhecidas como Pé de Galinha e Bola Cheia respectivamente, **MECENAS [2005]**²³, além de utilizadas para o modelo conceitual, são também utilizadas para a construção do modelo lógico, o qual, posteriormente, dará subsídio para a confecção do modelo físico do banco de dados.

A seguir, são apresentadas as premissas ditadas pela técnica de modelagem, técnica esta que majoritariamente recomenda a eliminação de toda e qualquer redundância de dados. Assim, redundâncias permitidas são aquelas relativas a chaves estrangeiras, *Foreign Keys*, *FKs*, que fazem referência à chave primária, *primary key*, *PK*, de outra tabela, somente.

Então, visto a importância do fator integridade, e de acordo com os princípios básicos da modelagem de dados, **MARTINS [1994]**²⁴, recomenda-se que:

1-se utilize um padrão para nomear as entidades e que esta nomenclatura seja coerente e significativa em relação a informação que a futura tabela armazenará. Denota também uma boa prática, identificar os relacionamentos, pois, embora obrigatoriamente não precisem de denominação, suas identificações facilitam a compreensão e eventual manutenção do modelo lógico de dados;

2-se associe, sempre que possível, o nome do atributo ao tipo de dado que ele armazenará;

3-se mantenha fora do diagrama, entidades que representem parâmetros genéricos;

4-se evite dependência recursiva, eliminando-se os relacionamentos com participação total, ou seja, obrigatoriedade de ambos os lados;

5-jamais se elimine a etapa da modelagem conceitual, partindo do modelo lógico;

6-toda entidade do modelo conceitual torne-se uma entidade no modelo lógico e uma tabela, no modelo físico;

7-se relacione diretamente cada coluna ao tipo da tabela a qual pertence. Se uma coluna descreve o assunto de uma tabela diferente, esta coluna deve pertencer à outra tabela;

8-se elimine colunas repetidas no modelo. Se uma informação se repete em diversas tabelas, é indício de que existem colunas desnecessárias e deve-se buscar a tabela que faça mais sentido para conter a coluna em questão;

9-se utilize colunas de texto com tamanho variável, pois tendem a consumir menos espaço de armazenamento;

10-se elimine atributos derivados ou calculados. Não é recomendado armazenar o resultado de cálculos nas tabelas. O correto é que o cálculo seja gerado sob demanda, normalmente em uma consulta;

11-toda tabela deve ter uma *PK*, que pode ser simples ou composta;

A *PK* é o identificador do registro e deve ser única dentro de uma tabela. Para uma coluna ser candidata à *PK*, deverá atender aos principais requisitos:

- a. deverá ser a menor possível;
- b. o seu valor deverá ser diferente de vazio ou zero (not null);
- c. deverá ser de preferência numérica; e
- d. o seu valor deverá ser único para toda a tabela;

12- *FKs* devem corresponder a *PKs* da relação associada ou nulas, quando não for uma coluna obrigatória;

13-se crie chaves cegas, *BKs*, toda vez que não for possível identificar a *PK*, ou quando esta for muito complexa, composta por muitos atributos. Esses tipos de atributos geralmente são conhecidos por atributos falsos, ou seja, não fazem parte de forma explícita da regra da aplicação, porém foram criados para garantir flexibilidade e integridade ao modelo de dados desenvolvido;

14-se relacione por *FKs*, atributos correspondentes à *PK* de outra tabela, estabelecendo a base para a integridade referencial;

15- os relacionamentos 1:1 possam ser mapeados numa única tabela, quando possuem a mesma *PK*, em duas tabelas quando as *PKs* são diferentes e um dos lados do relacionamento é obrigatório ou em três tabelas quando o relacionamento é opcional em ambos os lados;

16- relacionamentos N:N devem virar novas relações, associadas através de dois relacionamentos 1:N. Isso se deve ao fato de, no modelo relacional, não serem permitidos atributos de múltiplos valores, ou seja, admitindo vários valores. Neste caso, a entidade criada possui como chave primária a concatenação das chaves das duas entidades relacionadas, formando uma chave composta;

17-se identifique a cardinalidade de um relacionamento entre duas entidades, tendo-se em mente que a análise de todo relacionamento entre entidades é bidirecional;

18- os auto-relacionamentos do tipo 1:N gerem um atributo de ligação na própria tabela; e

19- os auto-relacionamentos do tipo N:N gerem uma nova tabela;

As dezenove recomendações anteriormente expostas por regras básicas da modelagem de dados, separadas e identificadas no contexto da normalização, são sintetizadas a seguir.

Normalização

A normalização é o processo que assegura a integridade do dado, através da diminuição de redundâncias e incoerências do modelo de dados. O processo de normalização aplica até seis regras, complementares entre si, sobre as entidades de um modelo lógico de dados, para verificar se estas estão coerentemente projetadas e sobretudo, devidamente interligadas. Embora existam seis regras de normalização ou seis formas normais, para sistemas com até 35 entidades aproximadamente e sem auto-relacionamentos na entidade, como o modelo de dados deste trabalho, não se aplicam as três últimas formas, ficando a integridade do modelo de dados assegurada pela aplicação das três primeiras formas.

Primeira forma normal

Eliminar atributos compostos é o objetivo da primeira forma normal. Exemplificando, atributos como simplesmente temperatura, seriam genéricos demais e insuficientemente claros sob a óptica da primeira forma normal. Assim, necessariamente precisam ser decompostos e conseqüentemente detalhados em temperatura da água, temperatura ambiente, temperatura do turbo-compressor etc. Em outro exemplo, se o atributo considerado for endereço e este não for decomposto, como saber se alguém mora na cidade do Rio de Janeiro ou na Rua Rio de Janeiro na cidade de Angra dos Reis? A primeira forma normal também se destina a eliminar atributos de múltiplos valores. Se o atributo em questão for por exemplo temperatura da água, outrora decomposto, vários valores são possíveis. Desta forma, neste exemplo, deve-se detalhar os multivalores possíveis do atributo decomposto como temperatura da água na superfície da piscina do reator, temperatura da água a meia altura da piscina do reator, temperatura da água sobre o núcleo do reator, temperatura da água de entrada do tanque de decaimento etc.

Outro exemplo, se o atributo considerado for telefone, a primeira forma normal deve classificá-lo em diversos telefones, como comercial, celular e residencial, ou então gerar uma nova entidade apenas com estas possibilidades e relacionar o atributo telefone da primeira entidade(1) para a segunda entidade(N). Se a primeira forma normal não for obedecida, corre-se o risco de perda de informações.

Na primeira forma normal também se deve observar se todas as relações possuem *PK* definida. Isto é importante, pois as formas normais seguintes baseiam-se na definição da *PK*.

Segunda forma normal

A segunda forma normal pressupõe a primeira forma normal contemplada e determina que atributos devam ser funcionalmente dependentes apenas da *PK*.

Assim, não é necessário atentar para esta forma normal em entidades que possuam chaves primárias simples, com apenas um atributo. Considere, por exemplo, a relação entre pedido e item-pedido. Imagine que o atributo data-pedido tivesse sido colocado na entidade item-pedido. Através da análise da segunda forma normal nesta entidade, que possui *PK* composta por pedido e item-pedido, pode-se verificar que a data-pedido depende apenas de parte da *PK*, ou seja, num-pedido, e não da *PK* completa, composta por num-pedido e item-pedido. Sem esta análise, a data do pedido seria repetida para todos os itens de um mesmo pedido. A verificação da segunda forma normal, neste exemplo, determina que o atributo data-pedido deve ser transferido para outra entidade. Outras implicações viriam na tentativa de se incluir a data do pedido enquanto não se registra um item do pedido ou se fossem excluídos todos os itens do pedido, perder-se-ia a data do mesmo. No caso de alteração, se a data do pedido precisar mudar, dever-se-ia mudar a mesma em todos os seus itens.

Antes de proceder com a análise da terceira forma normal, deve-se garantir que todas as relações estejam na segunda.

Terceira forma normal

A terceira forma normal trata atributos mutuamente independentes, ou seja, que não dependem uns dos outros. Nesta forma, é verificado se existe dependência funcional entre os atributos. Consideremos, por exemplo, uma entidade dita funcionário. Admita que esta entidade esteja nas primeira e segunda formas normais. Entretanto, existe um atributo nesta entidade denominado nome-departamento, que depende de outro atributo denominado cod-departamento. Este é o tipo de problema de que trata a terceira forma normal.

Para resolvê-lo, cria-se uma nova entidade, no caso departamento, contendo os atributos relacionados e mantém-se na tabela original aquele que determina o outro como *FK*. Ao não se obedecer a terceira forma, corre-se os

mesmos riscos apresentados no não cumprimento da segunda forma normal.

Então, não se poderia inserir um departamento antes que existisse um funcionário alocado nele ou a eliminação de todos os funcionários de um departamento acarretaria na eliminação das informações do próprio departamento, ou ainda, no caso de mudança de nome do departamento, todos os funcionários seriam afetados.

Desnormalização

O processo inverso à normalização é chamado de desnormalização.

Nesse processo, duas ou mais entidades do modelo lógico são aglutinadas em uma única entidade. A princípio, a única razão para a aplicação da desnormalização é a de eliminar o custo das junções em operações de seleção sobre as tabelas envolvidas. Entretanto, com uma análise superficial dessa frase pode-se concluir, erroneamente, que a desnormalização sempre aumenta o desempenho no processamento de consultas de seleção. O raciocínio, errôneo, para essa conclusão, seria o seguinte: se a quantidade de entidades é maior em um banco relacional normalizado, então haverá um maior número de operações de junção para a obtenção do resultado das consultas que em um modelo desnormalizado correspondente. Como operações de junção são sabidamente bastante custosas do ponto de vista do desempenho, conseqüentemente o custo da execução de seleções em um modelo normalizado é sempre maior que o custo sobre o modelo desnormalizado correspondente. Considerando as junções, ou seja, comparação entre valores afins de colunas de tabelas distintas, pode-se constatar que a sobrecarga de processamento necessária para manter a integridade dos dados pode não compensar o ganho de desempenho obtido com a desnormalização. De fato, a manutenção da integridade pode necessitar das mesmas operações de junção que a desnormalização se propõe eliminar.

Usando o mesmo exemplo anteriormente comentado, considere uma versão desnormalizada da tabela pedido, contendo as colunas das tabelas item-pedido e produto. Para obter informações sobre pedidos, itens de pedidos e produtos, a utilização dessa tabela desnormalizada realmente produz uma execução mais eficiente, pois todos os dados necessários estão armazenados em uma única tabela, o que normalmente leva o SGBD a posicionar essas informações em blocos de disco contíguos. No entanto, quando um nome de produto é atualizado se faz necessário que esta atualização seja efetivada em todos os registros em que o produto em questão consta, onerando mais do que o faria na condição

normalizada. O eventual benefício obtido pela desnormalização, aumento do desempenho em consultas de seleção, tem seu preço.

Uma tabela desnormalizada fica vulnerável ao surgimento de anomalias quando manipulações de atualizações ou inserções são realizadas sobre ela, e com isso, a integridade dos dados fica ameaçada.

A sobrecarga necessária para manutenção da integridade dos dados não é o único fator a considerar quando se cogitar desnormalizar uma entidade. Considere novamente os dados da entidade pedido. Essa entidade armazena dados sobre três conceitos distintos, ou seja, pedidos, itens de pedidos e produtos.

O que acontece quando aplicações de bancos de dados, como por exemplo, a aplicação deste trabalho, precisam acessar esses dados separadamente? Como esses dados estão em uma tabela que possui diversas outras colunas não relacionadas a produtos, a quantidade de informações sobre produtos por bloco de disco será maior do que se houvesse uma tabela armazenando somente dados sobre produtos. Conseqüentemente, o SGBD levará mais tempo para resgatar os dados necessários para montar o resultado da consulta ao utilizar a tabela desnormalizada. Perceba que esse ponto acaba indo de encontro ao citado anteriormente, ou seja, um dos benefícios da desnormalização, o não uso de junções, acaba sendo prejudicado em outras situações. De uma forma geral, se for tomada a decisão de aglutinar dados de duas ou mais entidades em uma única entidade, ou seja, desnormalizar o modelo, as aplicações que necessitam ter acesso aos dados, que ao contrário estariam em uma entidade separada, terão agora que ler desnecessariamente outras informações.

É preciso então ponderar os extremos, tanto da normalização, visando a integridade, quanto da desnormalização, visando o desempenho.

Ferramentas de modelagem

Existem ferramentas próprias, não necessariamente gráficas, para apoiar o processo de modelagem de dados. Algumas destas ferramentas, além deste apoio, também podem ser utilizadas como repositório da estrutura do banco de dados e utilizadas como referência quando da necessidade de uma cópia de segurança desta estrutura. A este grupo particular de ferramentas de modelagem, denominamos ferramentas case. Neste trabalho, embora o modelo de dados seja pequeno e não sofra alterações freqüentes, utiliza-se a versão gratuita da ferramenta *case Erwin*, **SOARES [2003]**²⁵. As versões gratuitas de outras

ferramentas como **EMBARCADERO**²⁶, são também bons exemplos disso.

A maioria destas ferramentas é de fácil utilização, atende aos SGBDs comerciais, possui engenharia reversa para geração dos arquivos de implementação física e os salva em formato XML, inclusive.

Tais ferramentas importam modelos gerados por outras ferramentas, sincronizam bancos via alteração em seus modelos e possuem fóruns oficiais de discussão na Internet, além de necessitarem de poucos recursos de *hardware* para serem instaladas.

Caso não se utilize ferramentas deste tipo, o modelo tenderá a ficar desatualizado rapidamente, pois dificilmente as modificações que são necessárias ao banco de dados são refletidas para a documentação sem o auxílio de uma ferramenta apropriada. Até pouco tempo, aproximadamente uma década e meia, a realização da atividade de modelagem era dificultada para aqueles que não possuíam recursos financeiros para aquisição de ferramentas de modelagens comerciais. Ficava assim difícil realizar atividades de modelagem. Com o advento da Internet e do *software* de código aberto, essa realidade foi se transformando e hoje já é possível encontrar algumas destas ferramentas gratuitamente.

4.1.2 Fator desempenho

Com o modelo de dados normalizado e o banco de dados fisicamente implementado, a próxima etapa é providenciar para que o seu desempenho seja satisfatório, sem que para isso haja qualquer tipo de comprometimento à integridade dos dados, outrora garantida pela normalização. O fator que mede o desempenho de um banco de dados, relacional ou não, é denominado tempo de resposta. Tempo de resposta é o tempo medido entre o início da manipulação da informação e a respectiva efetivação desta. Este fator é relativo, variando assim de sistema para sistema. Então, o tempo de resposta satisfatório para um sistema pode ser insatisfatório para outro. Otimizar o desempenho requer cuidados múltiplos, ora por parte da readequação do modelo de dados, desnormalizando-o se necessário, ora por parte da aplicação que acessa os dados, através da otimização das consultas, ora baseando-as em indicadores fornecidos pelo próprio SGBD tais como estatísticas de operação, adequação de índices e contenção de *Input/Output*, *I/O*, ora por armazenamento de consultas recentes.

Se necessário for, a princípio, este fator pode ser otimizado através da desnormalização. Outra forma nem sempre eficiente, mais dispendiosa, porém de mais simples implementação é a melhoria dos recursos de *hardware* através da

otimização de processadores, aumento da memória física dedicada ao banco de dados, redes mais eficientes no que tange a transferência dos dados e discos magnéticos mais rápidos.

4.1.3 Fator disponibilidade

Independente da modelagem dos dados, o fator disponibilidade resume-se em fazer com que a mesma informação esteja disponível em mais de um local fisicamente, mas que seja tratada logicamente como única. Com isso, não se interfere na integridade e desempenho, e ao mesmo tempo se garante a disponibilidade da informação, no caso de eventual falha de um dos locais onde fisicamente ela está. Hoje em dia, esta disponibilidade é provida por sistemas gerenciadores independentes do banco de dados, responsáveis por equalizar as bases de dados em tempo real e mantê-las em funcionamento simultâneo e sincronizado.

Esta prática é conhecida como contingência e/ou redundância, caracteriza-se por seu alto custo operacional e diferencia-se de processos nativos de bancos de dados pelo discernimento entre duplicação do dado e replicação deste.

4.1.4 Fator invulnerabilidade

Também independente da modelagem dos dados, este importante fator prima para que o dado saia de sua origem e chegue a seu destino, independentemente do sentido, sem que no trajeto haja qualquer tipo de interferência indesejada no sentido de manipular estes dados.

Para isso, algoritmos de criptografia aliados a equipamentos de segurança, como Firewalls, Proxies, redes protegidas etc. atuam em conjunto.

Alguns equipamentos providos de algoritmos mais sofisticados, além de impedirem a interferência ao dado, mapeiam os potenciais invasores e por intermédio de técnicas heurísticas, os identificam.

4.1.5 Coexistência dos quatro fatores fundamentais

Para que se possa efetivamente entender a importância destes fatores e, sobretudo a necessidade da sua utilização, faz-se necessário uma breve retrospectiva utilizando-se um exemplo real: No início dos anos oitenta, grandes instituições financeiras, mais especificamente bancos privados, iniciavam campanhas de popularização de seus serviços. Surge então o conceito do Banco 24 horas, onde era possível efetuar fora do expediente bancário convencional,

algumas tarefas que até então só eram possíveis serem efetuadas dentro do estabelecimento bancário, no horário convencional de atendimento. Desta forma, a possibilidade de um saldo desatualizado, por conta do uso desta facilidade, era na ocasião tão comum quanto tolerável. No final dos anos oitenta, já com a implementação de processos que refletiam a atualização de informações em tempo real, rotulados *on-line* e *real-time*, um saldo desatualizado era inadmissível, assim como também não mais se tolerava períodos muito dilatados para a conclusão de operações desta natureza.

Então algo mudou, e a tecnologia fomentou o aumento do nível de exigência! É evidente que, além de obter-se o saldo atualizado, utilizando-se das propriedades do fator integridade, é imperativo que este mesmo saldo fosse apresentado em um tempo aceitável, fazendo valer também o fator desempenho.

Mais alguns anos se passaram e hoje em dia, a certeza de que o mesmo saldo virá de forma íntegra e no menor tempo possível é absolutamente insuficiente. A demanda aponta para que isso ocorra, aliado a garantia de se obter este saldo quando solicitado (propriedade do fator disponibilidade) e que este esteja protegido no que se refere a manipulações indesejadas ao longo de seu trajeto (propriedade do fator invulnerabilidade).

Percebe-se claramente, através deste exemplo, a importância da coexistência destes fatores, em algumas ocasiões antagônicos entre si, mas jamais mutuamente exclusivos no contexto geral. Logo, o fato deste trabalho se engajar nos preceitos de sistemas transacionais, cuja principal característica é a manipulação de grandes volumes de dados sigilosos por um grande número de usuários, ao mesmo tempo, os fatores expostos anteriormente foram considerados, estudados e implementados, ajustando-se assim ao trabalho outrora proposto. Hoje o banco de dados no qual se baseia este trabalho, denominado FALCAO, localizado nas dependências físicas do Centro de Engenharia Nuclear, CEN, opera de forma a contemplar estes quesitos, enfatizando, por conta das características peculiares da área nuclear, os fatores integridade e desempenho.

4.1.6 O Modelo de dados do Banco FALCAO

O modelo de dados do banco FALCAO é representado pelo respectivo Diagrama Entidade-Relacionamento, DER, que foi concebido e inserido nos critérios da normalização e desnormalização, exposto anteriormente, para que o banco de dados pudesse ser fisicamente gerado.

A FIG. 2 mostra o modelo conceitual e a FIG. 3, o diagrama entidade-relacionamento, implementados no banco de dados FALCAO.

O DER mostra de forma gráfica as entidades que se tornarão tabelas na implementação física do banco de dados, assim como suas *constraints*.

Nota-se que a simbologia utilizada para representação conceitual dos relacionamentos entre as entidades, assim como a utilizada no DER, foi a coloquialmente conhecida como pé de galinha.

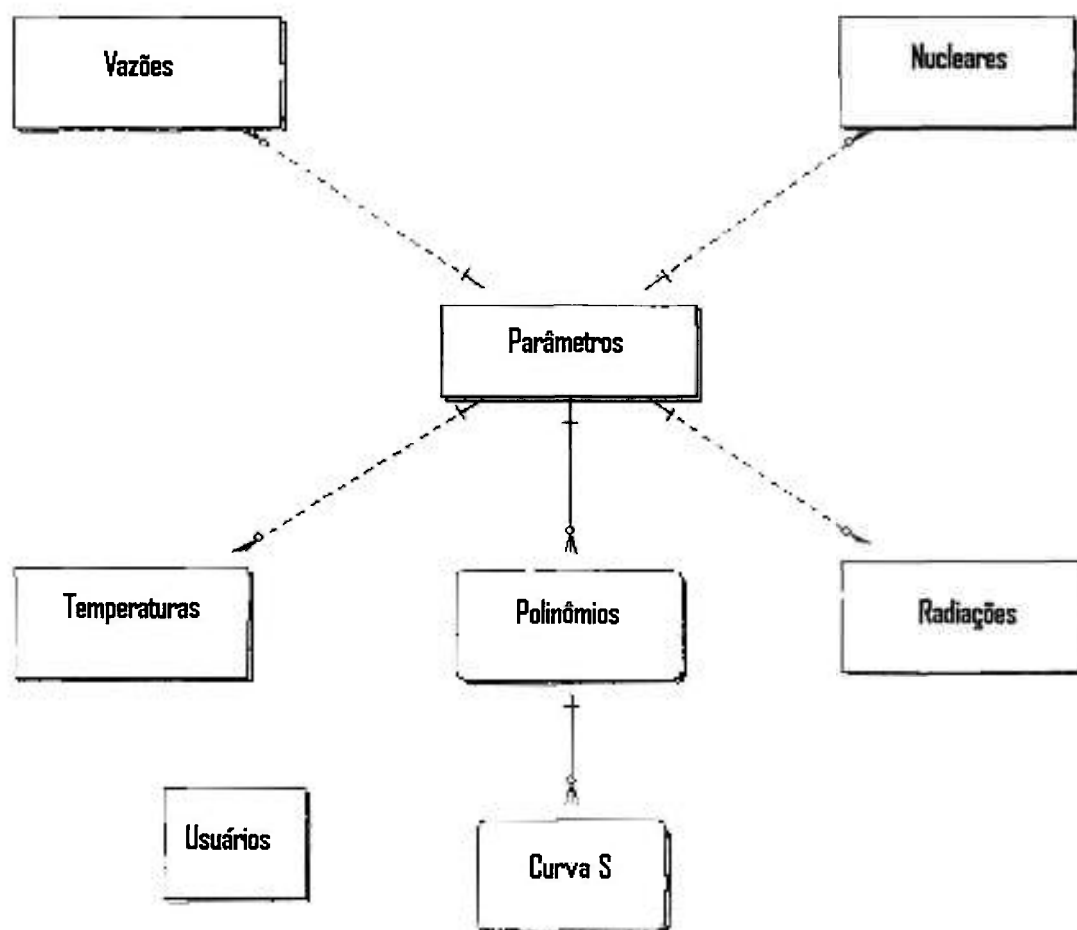


FIGURA 2 – Modelo conceitual do banco de dados FALCAO

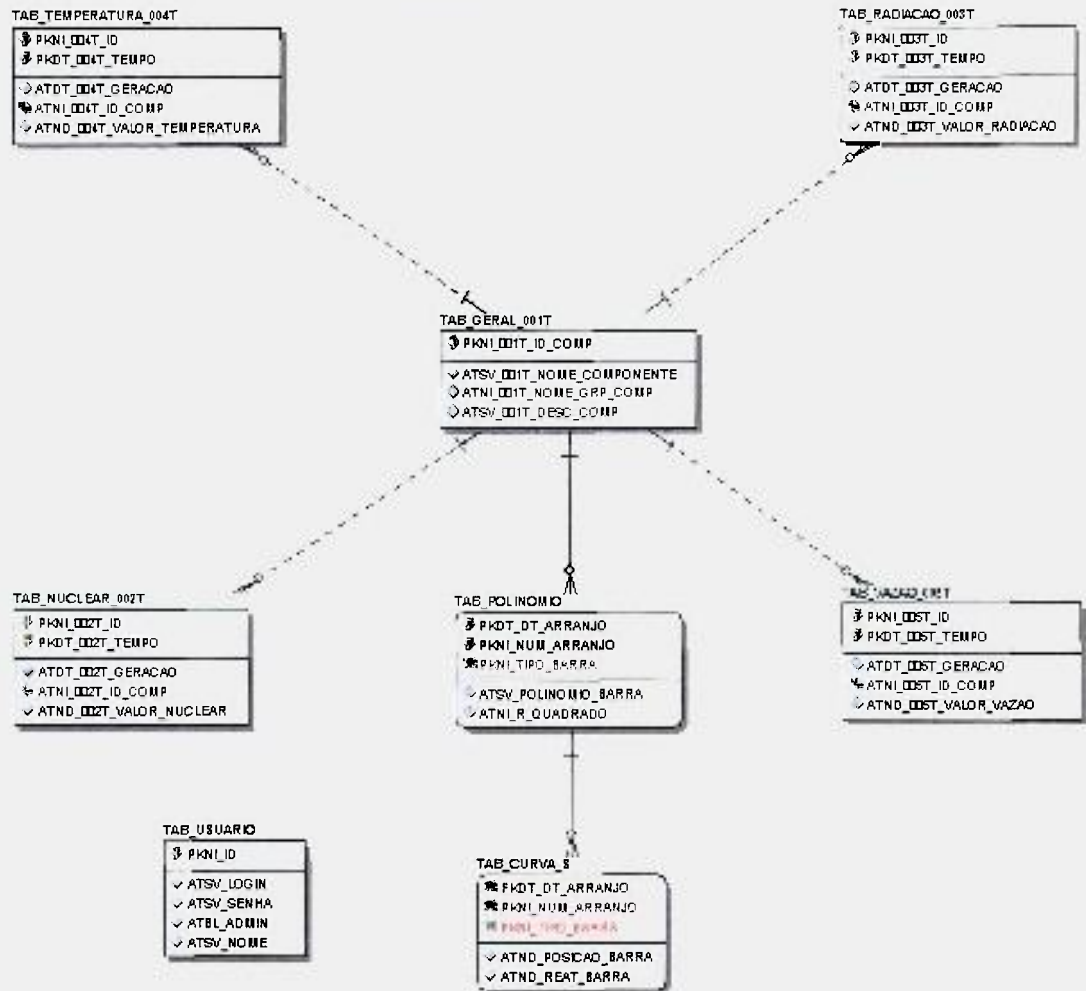


FIGURA 3 – DER do banco de dados FALCAO

4.1.7 Banco de dados FALCAO versus os quatro fatores fundamentais

Observa-se pelo próprio DER, FIG. 3, que todas as entidades periféricas, excetuando-se a entidade usuário, independente das demais, armazenam os dados referentes a sua identificação em suas colunas do tipo atributo e utilizam *PKs* compostas, onde nestas constam o identificador incremental crescente e a data de inserção do dado, representadas respectivamente pelas colunas *pkni_002T_id* e *pkdt_002T_tempo*, pertencentes, por exemplo, a tabela *TAB_NUCLEAR_002T* e fisicamente implementadas pela execução das rotinas do Apêndice A. Essa característica, encontrada em modelos normalizados como o deste trabalho, visa impossibilitar a existência de informações duplicadas sob a óptica da integridade e, sobretudo, permitir um melhor desempenho.

Caso haja necessidade de frequências de amostragens maiores do que as frequências hoje aplicadas haverá um aumento no volume de dados gravados.

Isto justificaria, por exemplo, a viabilidade do estudo de transientes, os quais poderiam ser praticados com a adequação da atual frequência de coleta de dados. O banco de dados FALCAO está preparado para este ajuste.

4.2 PROGRAMAS UTILIZADOS

Os programas utilizados na elaboração deste trabalho são: o utilitário SGBD *ORACLE 10G*, **SERSON [2004]²⁷**, o sistema operacional Linux, **DANESH [2000]²⁸**, distribuição Fedora Core 4 e a linguagem de programação *JAVA.5.0*, **JANDL [2002]²⁹**.

Com o intuito de melhor explicar a metodologia aplicada, são descritas a seguir algumas características relevantes de cada um dos programas citados, as peculiaridades que justificam suas escolhas e detalhes de suas respectivas implementações.

4.2.1 Gerenciador de banco de dados *ORACLE*

Trata-se de um gerenciador de banco de dados relacional, com alta capacidade de gerenciamento/processamento e que incorpora a tecnologia *grid*, tecnologia esta baseada no melhor aproveitamento dos recursos de *hardware* e por conseqüência, otimização destes via processamento paralelo, gerando melhor resultado no que se refere ao desempenho e custo associado de processamento. Neste trabalho utiliza-se o *ORACLE 10G* que, entre outras melhorias, possui novos mecanismos de recuperação de dados, de modo a disponibilizar informações perdidas em tempo hábil muito menor do que nas versões anteriores.

4.2.2 Sistema operacional Linux Fedora Core 4

Trata-se de um sistema operacional pré-emptivo, ou seja, com regras de prioridade de processamento programáveis e executadas dinamicamente, com distribuição livre, com *kernell* totalmente compatível com aplicativos e utilitários reconhecidos do mercado e sobretudo, com notável escalabilidade e versatilidade no que tange as necessidades operacionais do SGBD *ORACLE 10G*.

4.2.3 Linguagem de programação JAVA

JAVA é uma linguagem de programação compilável, orientada a objeto, bastante versátil e, por suas características, altamente recomendada para aplicativos que acessam bases relacionais transacionais de grande volume de dados, como no caso deste trabalho, que utiliza *JAVA 5.0*. Para este trabalho, seu principal atrativo é a facilidade de reprogramação do código e sobretudo a conveniência no que se refere ao desempenho, visto que por ser uma linguagem intrinsecamente não performática, além de não depender da plataforma onde está sendo executada, o desempenho da aplicação depende diretamente do desempenho do banco de dados.

Hoje em dia, grande parte das aplicações complexas acessa banco de dados relacionais e *JAVA* é a linguagem mais amplamente utilizada para tal, pois além de consagrada, é de fácil suporte por intermédio dos fabricantes, e também com vasta e difundida documentação.

Como servidor *web*, módulo de apoio a execução de programas *JAVA*, optou-se pelo produto *Tomcat*, sobretudo por ser homologado pelo próprio fabricante da linguagem de programação *JAVA* e reconhecidamente funcional.

4.3 IMPLEMENTAÇÃO DA INFRA-ESTRUTURA

O primeiro passo consiste em instalar o sistema gerenciador de banco de dados *ORACLE 10G*, sob o sistema operacional Linux Fedora Core 4. Após isso, e em um servidor distinto, instalam-se os *softwares JAVA/Tomcat* e programa-se a conectividade da aplicação, residente neste servidor e estações, ao banco de dados, com os critérios de segurança pré-estipulados. A seqüência e os detalhes das instalações e customizações são encontrados no Anexo B.

4.4 PARTE EXPERIMENTAL

Para que se entenda o procedimento através do qual os métodos foram aplicados, permitindo assim que a informação saia da sua origem até encontrar-se disponível, é necessário saber que as etapas ocorreram de forma mutuamente exclusivas e, sobretudo, na seqüência que estão mostradas a seguir.

4.4.1 Implementação do modelo de dados

Antes de gerar os arquivos da criação física do banco de dados, correspondentes ao modelo lógico de dados, é fundamental identificar neste modelo a correspondência entre as regras e os resultados esperados. Então, não se deve utilizar muitos atributos nas chaves das tabelas, apesar de muitas vezes o modelo lógico assim o recomendar. Quando se implementa no banco de dados um modelo com muitos atributos na chave, complica-se desnecessariamente a programação da aplicação que acessará esta tabela e sobretudo, o desempenho das consultas ao banco de dados. Neste estudo as chaves *PKs* são compostas por dois atributos, apenas.

O projeto físico é a implementação do modelo de dados e prima por questões relacionadas ao desempenho das consultas, majoritariamente, além de traduzir a integridade existente no modelo lógico. Para isso, utiliza-se da criação de índices, particionamento de mídias, paralelismo de processamento, ajustes etc. Estes detalhes não pertencem ao modelo lógico de dados. Assim, diversas características do projeto físico estão associadas a peculiaridades do produto SGBD específico, pois nesta etapa, além de efetivarem-se fisicamente as rígidas regras de integridade ditadas pela normalização do modelo lógico de dados, é necessário também que o desempenho seja satisfatório. A criação de índices para chaves estrangeiras, por exemplo, tende a melhorar o desempenho de consultas complexas ou que envolvam muitas tabelas ou ainda tabelas com muitos registros. No caso deste trabalho, embora fisicamente o banco de dados possua apenas cinco tabelas, estas possuem um grande e crescente volume, justificando assim a existência de índices em colunas do tipo *PK*, claramente observados no Apêndice A, assim como a implementação física das *constraints* que traduzem os relacionamentos do modelo lógico, fazendo valer a integridade referencial ditada pela normalização.

Assim, com a implementação física das tabelas, parâmetros, relacionamentos e índices, temos o banco de dados FALCAO fisicamente gerado, exaustivamente testado e disponível.

4.4.2 Conectividade ao banco de dados

Parte-se dos seguintes pré-supostos para iniciar esta etapa: O sistema operacional Linux Fedora 4 está instalado e o banco de dados FALCAO instalado e customizado neste, ou seja, está preparado para receber conexões oriundas da aplicação SACD e de estações clientes *ORACLE*.

A seguir, são descritos os procedimentos para que estas conexões coexistam, inclusive simultaneamente.

Conexões oriundas da aplicação SACD

Como detalhado a seguir no Capítulo 4, sub-item 4.4.5.1, a aplicação SACD foi desenvolvida na linguagem *JAVA*. Para que uma aplicação *JAVA* possa explorar de forma eficiente os recursos do banco de dados e operar com este de maneira estável, se faz necessário a existência de um ambiente operacional que permita isso.

Este ambiente, homologado pelos fabricantes tanto da linguagem *JAVA* como do banco de dados *ORACLE*, é responsável por fornecer subsídios para que a linguagem possa operar e também comunicar-se com o banco de dados.

Também conhecido como *Web server*, o produto que neste trabalho representa este ambiente operacional, denomina-se *Tomcat*.

Desta forma, configurar o produto *Tomcat* significa fornecer subsídios para que o programa *JAVA* seja executado e caso necessário, conectá-lo a base de dados, como neste trabalho.

Assim, com os produtos *JAVA* e *Tomcat* instalados e a aplicação SACD compilada, duas configurações são necessárias para que a aplicação SACD efetivamente se conecte ao banco de dados:

1-Instalação do *driver* de conexão *JAVA/ORACLE*

Trata-se de um programa específico que conecta uma aplicação *JAVA*, executada sob o produto *Tomcat*, a um banco de dados *ORACLE* 10G.

A versão mais recente e recomendada pela própria *ORACLE* é a *ojdbc14.jar*, utilizada neste trabalho e encontrada no sistema operacional Windows XP, Service Pack 2.

O procedimento consiste em copiar este programa para o exato local escolhido na ocasião da instalação do produto *Tomcat*.

2-Configuração do arquivo JNDI

Trata-se do arquivo que armazena os parâmetros do banco de dados, necessários para que a aplicação se conecte a ele, através do *driver* outrora mencionado.

Os parâmetros identificação do servidor de banco de dados, usuário/senha através dos quais a aplicação se conectará ao banco de dados e a respectiva porta na qual o banco de dados foi instalado, devem, necessariamente, constar do arquivo tipo JNDI, denominado *server.xml*, gerado na instalação do *software Tomcat*, entre as tags <host> e </host>, como mostrados a seguir.

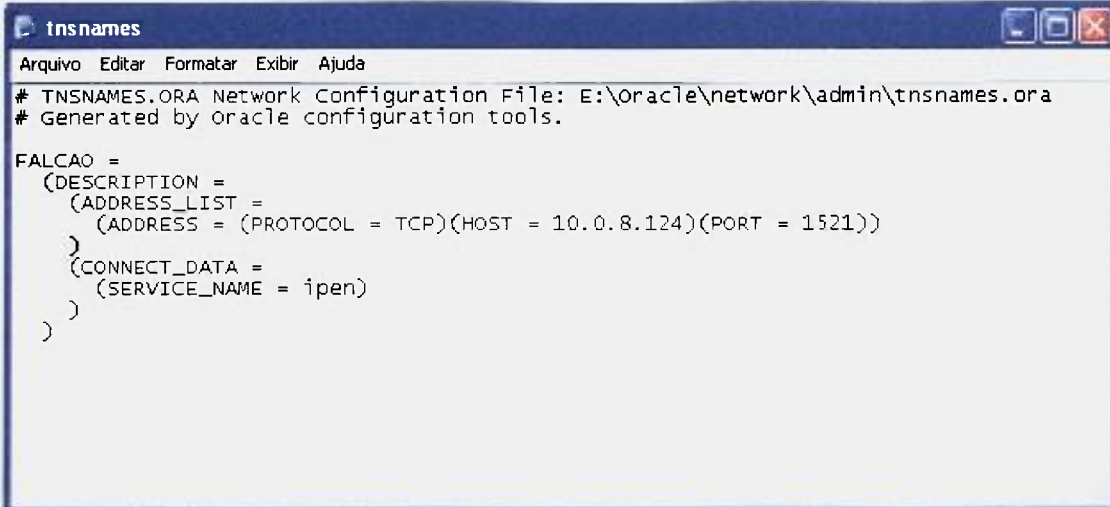
```
<Host>
  <Context docBase="sacd" path="/sacd" reloadable="true"
    source="org.eclipse.jst.j2ee.server:sacd">
    <Resource
      driverClassName="oracle.jdbc.driver.OracleDriver"
      maxActive="4"
      maxIdle="2" maxWait="5000" name="jdbc/FALCAO"
      password="IPEN2007"
      type="Javax.sql.DataSource"
      url="jdbc:oracle:thin:@10.0.8.124:1521:IPEN"
      username="CEN"
      validationQuery="SELECT 1 FROM DUAL" />
    </Context>
  </Host>
```

Em destaque, os parâmetros que variam para cada ambiente e que correspondem a configuração do ambiente operacional deste trabalho.

Conexões oriundas de estações cliente

Para que estas conexões ocorram, o procedimento é mais simples se comparado ao anterior, e resume-se na configuração dos mesmos parâmetros utilizados para a aplicação, porém registrados em um único arquivo, arquivo este gerado na instalação do cliente *ORACLE*, denominado *tnsnames.ora*. Trata-se de um arquivo texto, localizado no caminho C:/oracle/10.2.0/network/admin da estação onde o cliente *ORACLE* foi instalado.

A FIG. 4 traz um exemplo de conexão de uma estação cliente *ORACLE* ao banco de dados FALCAO.



```

tnsnames
Arquivo Editar Formatar Exibir Ajuda
# TNSNAMES.ORA Network Configuration File: E:\oracle\network\admin\tnsnames.ora
# Generated by oracle configuration tools.

FALCAO =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 10.0.8.124)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = ipen)
    )
  )

```

FIGURA 4- Arquivo de configuração da conectividade dos clientes *Oracle* ao SGBD *Oracle*

Vale observar que neste arquivo não se faz necessário o registro dos parâmetros usuário e senha, além do que, seu uso é genérico, ou seja, serve para qualquer versão de SGBDs *ORACLE*.

4.4.3 O processo de transmissão, consistência e carga dos dados

Setenta e sete variáveis são monitoradas no reator IEA-R1, gravadas no servidor SAD, nas dependências físicas do reator IEA-R1, em arquivos com extensão *txt* e transmitidas através da rede interna do IPEN para o banco de dados FALCAO, baseado no produto *ORACLE* versão 10G, instalado no servidor também denominado FALCAO sob o sistema operacional Linux Fedora Core 4, localizado no CEN, fora das dependências do reator.

Estes arquivos são subdivididos em quatro grupos, a saber:

-Dezenove variáveis relacionadas à grandeza vazão foram agrupadas, formando o arquivo denominado vazao.txt; trinta variáveis relacionadas à grandeza temperatura foram agrupadas, formando o arquivo denominado temperatura.txt; quinze variáveis relacionadas à grandeza radiação foram agrupadas, formando o arquivo denominado radiacao.txt e treze variáveis relacionadas às grandezas especificamente nucleares foram agrupadas, formando o arquivo nucleares.txt, contendo em média 1.700 KB cada um. Atualmente estes arquivos são definidos em períodos de sete dias corridos, quando são gerados e transmitidos manualmente.

Este trabalho automatiza a transferência dos dados e está preparado para suportar tal automação em intervalos de tempo menores do que os intervalos de

geração hoje aplicados, cujos detalhes são encontrados no Capítulo 5 deste documento.

Os processos de transmissão, consistência e carga dos dados estão implementados e operando com êxito.

Nesta implementação, pelo fato do servidor SAD estar remotamente localizado em relação ao servidor FALCAO, a etapa relacionada à transmissão dos arquivos exigiu uma maior automação, pois além da influência do fator remoto, o horário e seqüência de geração destes não podem, atualmente, ser previamente programados. Esta automação pode ser encontrada na rotina *Transfere_Dados_IEA-R1* que, após agendamento autorizado em uma estação de trabalho que opere com sistema operacional Windows ou Linux, verifica no servidor SAD a existência dos arquivos a serem carregados e em caso afirmativo, os transmite para o servidor FALCAO. Uma vez que estas informações são transmitidas para o servidor FALCAO, o programa *Carrega_Dados_IEA-R1* é iniciado automaticamente, de modo a validar os quesitos necessários estabelecidos previamente para que os dados sejam considerados válidos, promovendo-os assim ao processo de carga, carregando-os ou devolvendo-os a seu local de origem para correção e posterior retransmissão e carga. Vale destacar que toda e qualquer etapa do processo relacionado à transmissão, consistência e carga de dados é sinalizado em detalhes através de arquivos com extensão *log*, tanto na origem dos dados como no destino destes e que contém na sua denominação, data, hora, tipo do processo em questão, arquivo envolvido e o resultado final do processo. O Apêndice B traz os códigos-fonte dos programas *Transfere_Dados_IEA-R1* e *Carrega_Dados_IEA-R1*.

4.4.4 As variáveis carregadas no banco de dados FALCAO

Instalado nas dependências físicas do IPEN, através do sistema SAD, o reator IEA-R1 provê as variáveis de estado a serem carregadas no banco de dados FALCAO.

O IEA-R1 é um reator de pesquisas do tipo piscina, cujos principais propósitos são a produção de radioisótopos para uso na medicina, testes de materiais e combustíveis nucleares, irradiação de amostras com nêutrons, realização de pesquisas fundamentais em diversas áreas tais como em física, radioquímica, radiobiologia, análise por ativação, auxílio à formação de recursos

humanos em nível de pós-graduação e treinamento de pessoal especializado para operação de reatores.

Originalmente projetado, pela empresa norte-americana Babcox & Wilcox, para 5 MW, sempre operou a 2 MW e a sua primeira criticalidade ocorreu em 16 de setembro de 1957.

Em 1995 o IPEN-CNEN/SP deu início ao projeto de aumento de potência para 5 MW visando a produção de radioisótopos, em especial o molibdênio-99. No processo de modernização do IEA-R1 foi introduzido o Sistema de Aquisição de Dados, SAD.

As FIG. 5 e 6 mostram os diagramas esquemáticos, oriundos do manual do fabricante do sistema SAD, destacando os pontos do reator IEA-R1 onde são efetuadas, por exemplo, as coletas dos valores das variáveis de temperatura e parte das variáveis de vazão, assim como o sistema de arrefecimento do circuito secundário deste.

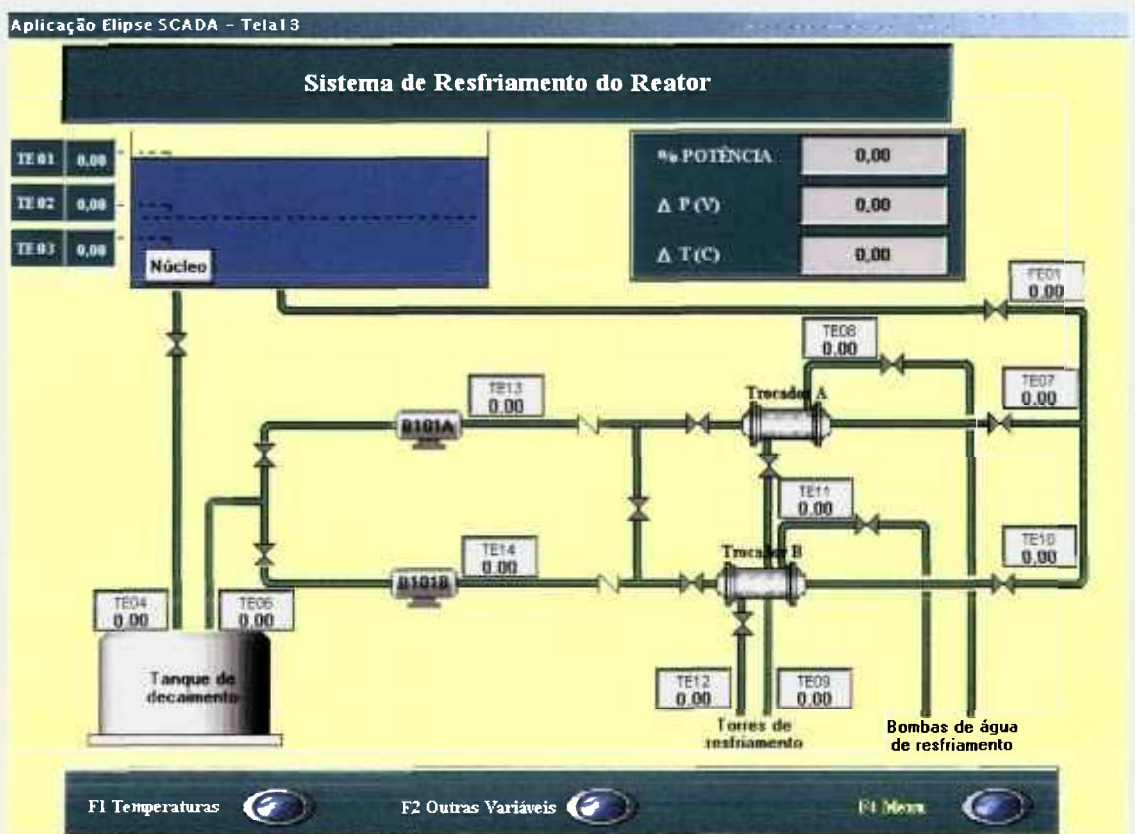


FIGURA 5 – Diagrama – Coleta das variáveis temperatura e vazão no primário IEA-R1

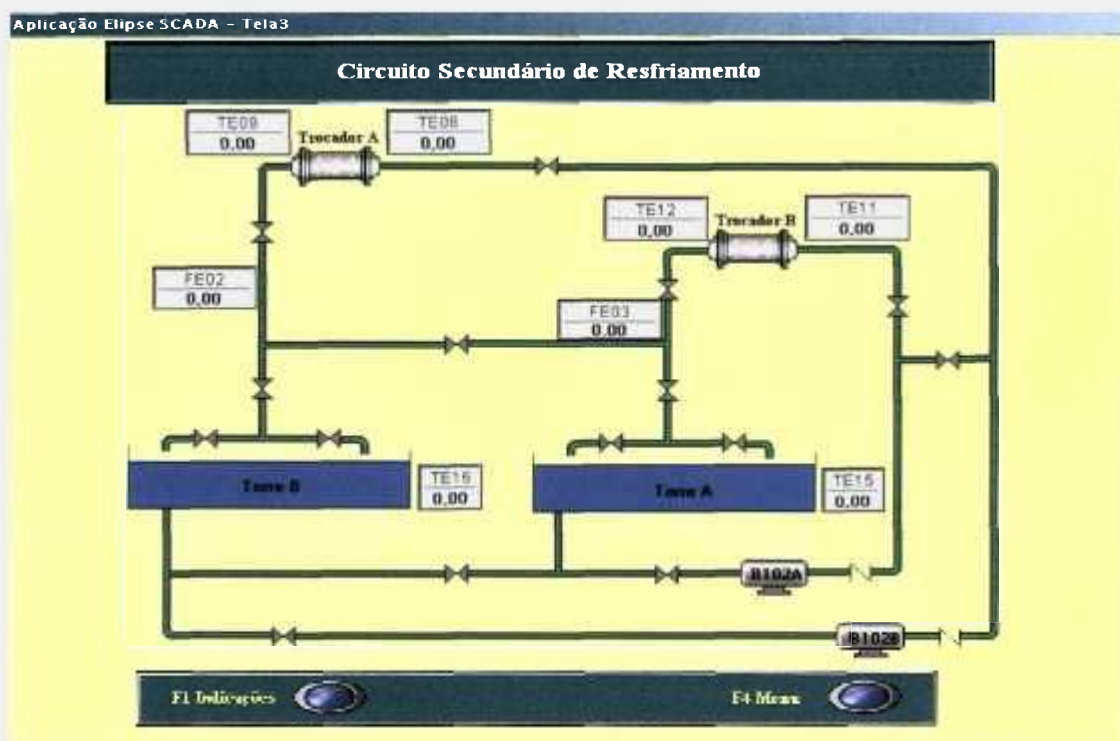


FIGURA 6 – Diagrama – Arrefecimento do secundário do IEA-R1

A TAB. 1 apresenta a conversão da nomenclatura das principais variáveis do sistema SAD, mostradas nas FIG. 5 e 6, com a denominação da variável carregada no banco de dados FALCAO.

TABELA 1- Conversões de Nomenclaturas para o reator IEA-R1

Nomenclatura do sistema SAD	Variável de estado correspondente
TE01	T1
TE02	T2
TE03	T3
TE04	T4
TE06	T6
TE07	T7
TE08	T8
TE09	T9
TE10	T10
TE11	T11
TE12	T12
TE13	T13
TE14	T14
TE15	T15
TE16	T16
FE01	F1M3
FE02	F3M3
FE03	F2M3

Como citado anteriormente, atualmente estas variáveis somam setenta e sete e são coletadas cotidianamente.

A seguir, é descrito o conteúdo dos quatro grupos de arquivos mencionados anteriormente.

Variáveis de vazão

F1M3	Vazão do circuito primário com a bomba B101A ou B101B [Gal/min]
F23	Vazão do circuito secundário com a bomba B102A ou B102B [Gal/min]
F2M3	Vazão de saída do trocador A [Gal/min]
F3M3	Vazão de saída do trocador B [Gal/min]
C1	Condutividade da água da piscina, após o tratamento [μ Siemens/cm]
C2	Condutividade da água da piscina, antes do tratamento [μ Siemens/cm]
L1	Nível da água da piscina do reator [m]
F1	Constante de entrada para cálculo da vazão
F2	Constante de entrada para cálculo da vazão
F3	Constante de entrada para cálculo da vazão
F4	Desligado
DP	Sinal elétrico relativo à diferença de pressão do núcleo do reator [v]

PR	Cálculo da potência do reator [MWatts]
PRPA	Cálculo da potência do reator, circuito primário, trocador de calor A [MWatts]
PRPB	Cálculo da potência do reator, circuito primário, trocador de calor B [MWatts]
PRSA	Cálculo da potência do reator, circuito secundário, trocador de calor A [MWatts]
PRSB	Cálculo da potência do reator, circuito secundário, trocador de calor B [MWatts]
NBPA	Cálculo da potência do reator, novo balanço, trocador de calor A [MWatts]
NBPB	Cálculo da potência do reator, novo balanço, trocador de calor B [MWatts]

Variáveis de temperatura

T1	Temperatura na superfície da piscina do reator [$^{\circ}\text{C}$]
T2	Temperatura a meia altura da piscina do reator [$^{\circ}\text{C}$]
T3	Temperatura sobre o núcleo do reator [$^{\circ}\text{C}$]
T4	Temperatura de entrada do tanque de decaimento [$^{\circ}\text{C}$]
DT1	Variação de temperatura, $T4 - T3$ [$^{\circ}\text{C}$]
T6	Temperatura de saída do tanque de decaimento [$^{\circ}\text{C}$]
T7	Temperatura de saída do circuito primário, trocador de calor A [$^{\circ}\text{C}$]
T8	Temperatura de entrada do circuito secundário, trocador de calor A [$^{\circ}\text{C}$]
T9	Temperatura de saída do circuito secundário, trocador de calor A [$^{\circ}\text{C}$]
DT2	Variação de temperatura, $T9 - T8$ [$^{\circ}\text{C}$]
T10	Temperatura de saída do circuito primário, trocador de calor B [$^{\circ}\text{C}$]
T11	Temperatura de entrada do circuito secundário, trocador de calor B [$^{\circ}\text{C}$]
T12	Temperatura de saída do circuito secundário, trocador de calor B [$^{\circ}\text{C}$]
DT3	Variação de temperatura, $T12 - T11$ [$^{\circ}\text{C}$]
T13	Temperatura na carcaça da bomba do circuito primário A [$^{\circ}\text{C}$]
T14	Temperatura na carcaça da bomba do circuito primário B [$^{\circ}\text{C}$]
T15	Temperatura externa da torre de refrigeração A [$^{\circ}\text{C}$]
T16	Temperatura externa da torre de refrigeração B [$^{\circ}\text{C}$]
DT4	Variação de temperatura, $T16 - T15$ [$^{\circ}\text{C}$]

T17	Temperatura da carcaça do motor do turbo compressor [$^{\circ}\text{C}$]
T18	Temperatura do <i>NO-BREAK</i> -220V [$^{\circ}\text{C}$]
DT5	Variação de temperatura, T17 – T18 [$^{\circ}\text{C}$]
T19	Temperatura do <i>NO-BREAK</i> -440V [$^{\circ}\text{C}$]
T20	Temperatura do primário do trocador de calor A, entrada [$^{\circ}\text{C}$]
DELTA A	Variação de temperatura, T20 – T7 [$^{\circ}\text{C}$]
T21	Temperatura do primário do trocador de calor B, entrada [$^{\circ}\text{C}$]
DELTA B	Variação de temperatura T21 – T10 [$^{\circ}\text{C}$]
T22	Temperatura externa [$^{\circ}\text{C}$]
T23	Não utilizado atualmente [$^{\circ}\text{C}$]
T24	Não utilizado atualmente [$^{\circ}\text{C}$]

Variáveis de radiação

MA1	Monitor de área 1, ponte móvel de sustentação do núcleo do reator, lado esquerdo [MicroSv/h]
MA2	Monitor de área 2, ponte móvel de sustentação do núcleo do reator, lado direito [MicroSv/h]
MA3	Monitor de área 3, saguão da piscina do reator [MicroSv/h]
MA4	Monitor de área 4, tubo colimador 8 – 1 ^o andar [MicroSv/h]
MA5	Monitor de área 5, tubos colimadores 3 e 4 - 1 ^o andar [MicroSv/h]
MA6	Monitor de área 6, tubos de armazenamento - 1 ^o andar [MicroSv/h]
MA7	Monitor de área 7, porão do reator [MicroSv/h]
MA8	Monitor de área 8, coluna de resinas [MicroSv/h]
MA9	Monitor de área 9, trocador de calor-1 [MicroSv/h]
MD1	Monitor de duto 1, sala de experimentos - 1 ^o andar [cpm]
MD2	Monitor de duto 2, saguão da piscina do reator 3 ^o andar [cpm]
MD3	Monitor de duto 3, exaustão geral - 4 ^o andar [cpm]
MD4	Monitor de duto 4, exaustão da chaminé - 4 ^o andar [cpm]
MD5	Monitor de iodo, 3 ^o andar [cpm]
MD6	Monitor de gases nobres, 3 ^o andar [MBq/m ³]

Variáveis nucleares

N1	Período do reator, câmara de fissão [s]
N2	Percentual de potência (<i>Safety</i>) do canal de segurança 1 da câmara de fissão [%N]
N3	Percentual de potência (<i>Safety</i>) do canal de segurança 2 da câmara de fissão [%N]
N4	Percentual de potência (<i>Safety</i>) do canal de segurança 3 da câmara de fissão [%N]
N5	Potência logarítmica, câmara de fissão [%]
N6	Percentual de potência do canal linear, da câmara de ionização compensada [%]
N7	Percentual de demanda de potência [%]
N8	Potência do detector de nitrogênio 16, da câmara de ionização [%N]
Z1	Posição da barra de controle [cm]
Z2	Posição da barra de segurança 1 [cm]
Z3	Posição da barra de segurança 2 [cm]
Z4	Posição da barra de segurança 3 [cm]
PERIOD	Período de operação do reator [s]

Estas variáveis são gravadas na tabela de parâmetros. Assim, o número de linhas da TAB. tab_geral_001T será coincidente com o número de variáveis monitoradas. As informações contidas na tabela de parâmetros, tab_geral_001T, servem de referência para, através dos relacionamentos, identificar-se a seqüência, o nome, o valor e a descrição das variáveis gravadas nas tabelas efetivas. A seguir, são mostrados os trechos da rotina de inserção responsáveis pela gravação da primeira e da última variável, na tabela de parâmetros.

Inserção da primeira variável

```
insert into cen.tab_geral_001T
(pkni_001T_id_comp,atsv_001T_nome_componente,atni_001T_nome_grp_com
p,atsv_001T_desc_comp)
values (1,'N1',11,'Período do reator, câmara de fissão');
commit;
```

Inserção da última variável

```
insert into cen.tab_geral_001T
(pkni_001T_id_comp,atsv_001T_nome_componente,atni_001T_nome_grp_com
p,atsv_001T_desc_comp)
values
(77,'NBPB',14,'Cálculo da potência do reator - novo balanço, trocador de calor
B');
commit;
```

4.4.5 O processo de disponibilização dos dados

Padronizado pela norma *American National Standards Institute, ANSI*, a linguagem *Structure Query Language, SQL*, permite a interação do meio externo ao banco de dados relacional, quer para controle, através da sub-linguagem *Data Control Language, DCL*, quer para definições, através da sub-linguagem *Data Definition Language, DDL*, ou ainda para manipulações, através da sub-linguagem *Data Manipulation Language, DML*, esta última possibilitando selecionar, gravar, apagar ou atualizar no banco de dados relacional o que se deseja (*select, insert, delete, update*) no lugar desejado (*from*) de acordo com uma ou mais condições previamente estabelecida(s) (*where, order by, group by, having*). Estas cláusulas da sub-linguagem *DML*, combinadas ou não, são responsáveis por levar a informação até a base de dados ou trazer a informação da base de dados que por sua vez, estará customizada de modo a provê-la sob demanda, com regras e critérios de integridade. Desta forma, a aplicação SACD, evidentemente utiliza a sub-linguagem *DML* para acessar os dados do banco de dados FALCAO.

O Anexo A traz mais detalhes da linguagem *SQL* como um todo.

4.4.5.1 Acesso aos dados carregados através da aplicação SACD

Neste trabalho, o acesso aos dados ocorre preferencialmente através da aplicação *Web SACD*, via navegadores (*browsers*) instalados em estações remotas espalhados pelas dependências do IPEN. Todavia, o acesso foi estudado e implementado de modo a permitir que consultas simultâneas ao banco de dados FALCAO, de diferentes pontos do IPEN, obtenham suas respostas em tempo computacional aceitável, mas que por outro lado não interfiram em informações oriundas de outros processos que hoje trafegam com êxito pela mesma rede interna de computadores do IPEN.

Vale destacar que atualmente o fluxo de informações do servidor FALCAO, embora bidirecional, ocorre exclusivamente nas dependências da rede interna do IPEN, mas que mesmo assim, o acesso aos dados é controlado.

Aplicação *Web* SACD

Sistemas aplicativos, ou seja, de uso específico, têm por característica principal a possibilidade de armazenarem em seus códigos-fonte, na ocasião da programação, as regras e restrições impostas aos dados, ou então de permitirem que este tratamento seja efetuado pelo próprio banco de dados, através de suas regras e *constraints*. Adota-se por prática o primeiro caso quando os processos são menos críticos sob o ponto de vista do desempenho, quando a natureza da informação é de domínio geral e, sobretudo, quando as regras impostas aos dados são dinâmicas o suficiente ao ponto de justificarem a reprogramação freqüente do código fonte em vez de eventual modificação da estrutura do banco de dados.

Já o segundo caso é mais utilizado quando o desempenho é fator fundamental, quando as alterações na aplicação são esporádicas e, sobretudo, quando o acesso aos dados não ocorre através de um único aplicativo.

Assim, por características intrínsecas a área nuclear, por necessitar do resultado da consulta aos dados em tempo computacional aceitável, por destacar o banco e não alterações freqüentes na aplicação e também por se cogitar a possibilidade de acessar os dados via outros aplicativos simultaneamente e por usuários distintos, as regras e *constraints* responsáveis pela integridade dos dados e desempenho das consultas estão implícitas ao modelo de dados, ou seja, residem no banco de dados e não em regras da aplicação.

Neste trabalho o segundo caso foi adotado. Baseado nestes conceitos, a aplicação *Web* foi concebida e desenvolvida em linguagem *JAVA*, pelo fato desta linguagem possuir sólida padronização e permitir uma boa portabilidade e desempenho, ou seja, o mesmo código-fonte pode ser executado sob sistemas operacionais distintos tais como: Windows 95, Windows 98, Windows XP, Linux e Unix, com desempenhos similares. A linguagem *JAVA* apresenta um bom desempenho mesmo em máquinas com pouca capacidade de processamento.

Assim, através do aplicativo SACD o usuário pode gerar relatórios, comparar informações, analisar gráficos e gerar arquivos em tempo computacional adequado.

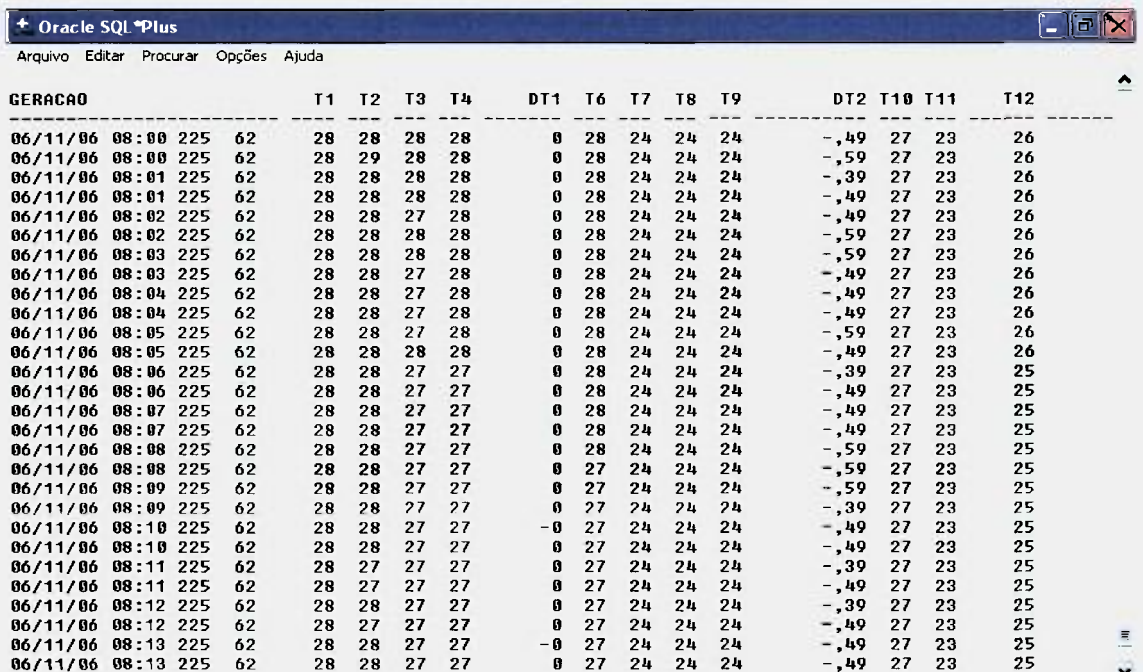
4.4.5.2 Outras formas de acesso aos dados carregados

O acesso também se faz possível através de interfaces nativas ou ferramentas comerciais gratuitas ou não, dedicadas a este fim. Neste caso, usuários administradores do banco de dados, através destas ferramentas, permitem ou revogam acessos, de acordo com a política de segurança pré-estabelecida. Estes acessos são controlados por senha e têm permissões correspondentes às necessidades de acesso dos respectivos usuários. Como dito anteriormente, embora a aplicação SACD seja a forma mais eficiente de acesso aos dados gravados no banco de dados FALCAO, pelas funcionalidades que convenientemente oferece, não é a única forma de acesso.

Assim sendo, basta a permissão de acesso através de um usuário no banco de dados FALCAO, para que outras formas se tornem possíveis, embora menos adequadas quando comparadas ao SACD. Como exemplo, pode-se citar a própria interface nativa do SGBD ORACLE 10G, denominada Sql-plus.

Orientada a *caracter* e por este motivo bem menos interativa e versátil do que a aplicação SACD, também permite com limitações, acesso aos dados.

A FIG. 7 mostra um exemplo de uso da interface nativa Sql-plus conectada ao banco de dados FALCAO, a partir de uma estação cliente remota a ele.



GERACAO	T1	T2	T3	T4	DT1	T6	T7	T8	T9	DT2	T10	T11	T12		
06/11/06 08:00 225 62	28	28	28	28	0	28	24	24	24	-	49	27	23	26	
06/11/06 08:00 225 62	28	29	28	28	0	28	24	24	24	-	59	27	23	26	
06/11/06 08:01 225 62	28	28	28	28	0	28	24	24	24	-	39	27	23	26	
06/11/06 08:01 225 62	28	28	28	28	0	28	24	24	24	-	49	27	23	26	
06/11/06 08:02 225 62	28	28	27	28	0	28	24	24	24	-	49	27	23	26	
06/11/06 08:02 225 62	28	28	28	28	0	28	24	24	24	-	59	27	23	26	
06/11/06 08:03 225 62	28	28	28	28	0	28	24	24	24	-	59	27	23	26	
06/11/06 08:03 225 62	28	28	27	28	0	28	24	24	24	-	49	27	23	26	
06/11/06 08:04 225 62	28	28	27	28	0	28	24	24	24	-	49	27	23	26	
06/11/06 08:04 225 62	28	28	27	28	0	28	24	24	24	-	49	27	23	26	
06/11/06 08:05 225 62	28	28	27	28	0	28	24	24	24	-	59	27	23	26	
06/11/06 08:05 225 62	28	28	28	28	0	28	24	24	24	-	49	27	23	26	
06/11/06 08:06 225 62	28	28	27	27	0	28	24	24	24	-	39	27	23	25	
06/11/06 08:06 225 62	28	28	27	27	0	28	24	24	24	-	49	27	23	25	
06/11/06 08:07 225 62	28	28	27	27	0	28	24	24	24	-	49	27	23	25	
06/11/06 08:07 225 62	28	28	27	27	0	28	24	24	24	-	49	27	23	25	
06/11/06 08:08 225 62	28	28	27	27	0	28	24	24	24	-	59	27	23	25	
06/11/06 08:08 225 62	28	28	27	27	0	27	24	24	24	-	59	27	23	25	
06/11/06 08:09 225 62	28	28	27	27	0	27	24	24	24	-	39	27	23	25	
06/11/06 08:09 225 62	28	28	27	27	0	27	24	24	24	-	49	27	23	25	
06/11/06 08:10 225 62	28	28	27	27	-	0	27	24	24	24	-	49	27	23	25
06/11/06 08:10 225 62	28	28	27	27	0	27	24	24	24	-	49	27	23	25	
06/11/06 08:11 225 62	28	27	27	27	0	27	24	24	24	-	39	27	23	25	
06/11/06 08:11 225 62	28	27	27	27	0	27	24	24	24	-	49	27	23	25	
06/11/06 08:12 225 62	28	28	27	27	0	27	24	24	24	-	39	27	23	25	
06/11/06 08:12 225 62	28	27	27	27	0	27	24	24	24	-	49	27	23	25	
06/11/06 08:13 225 62	28	28	27	27	-	0	27	24	24	24	-	49	27	23	25
06/11/06 08:13 225 62	28	28	27	27	0	27	24	24	24	-	49	27	23	25	

FIGURA 7 – Temperatura - Editor nativo Sql-plus

A FIG. 8 ilustra uma terceira forma de acesso, através da ferramenta comercial gratuita denominada *Toad*, MCDANIEL [2002]³⁰ que, embora com características próprias e distintas da interface nativa, também objetiva acessar os dados.

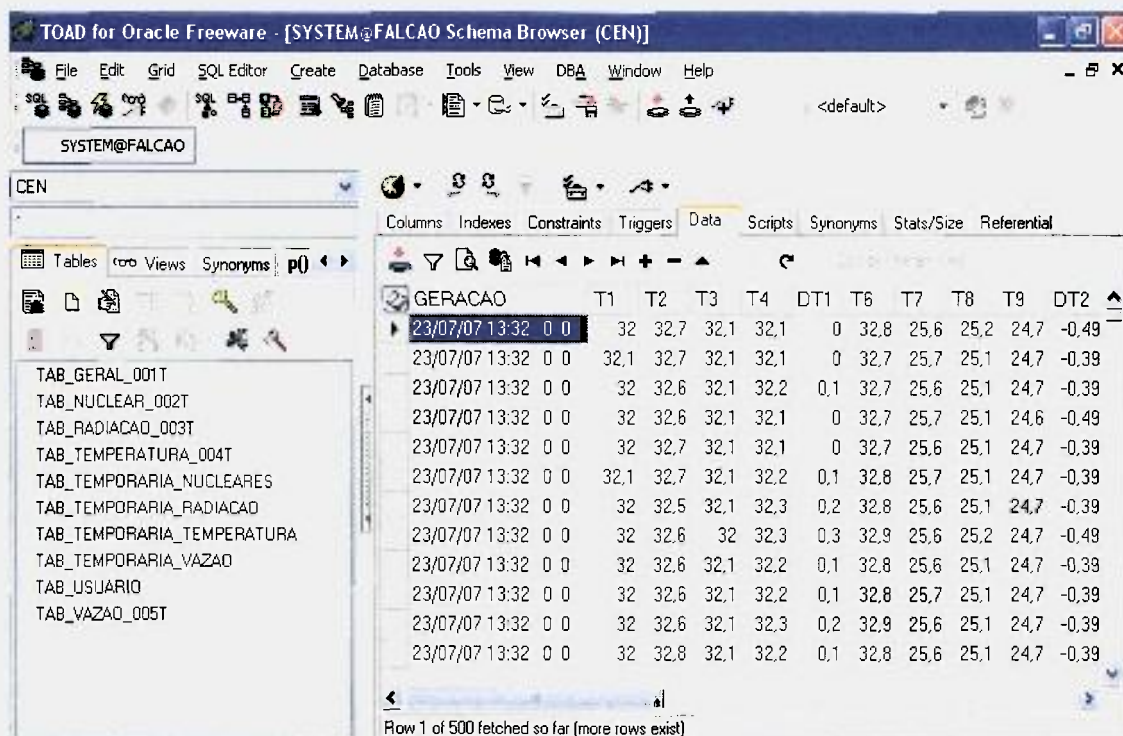


FIGURA 8 – Temperatura – Utilização da ferramenta Toad

4.4.6 Dificuldades encontradas

A principal dificuldade encontrada na implementação da infra-estrutura operacional como um todo foi em relação a conectividade ao banco de dados. Mesmo com os firewalls do sistema operacional desabilitados, com a instalação do produto *ORACLE* estes agiram impedindo, por segurança, a conexão das estações clientes ao banco de dados *ORACLE*. Visto a semelhança e a natureza da mensagem de erro, somente após exaustivos testes, constatou-se que a negação à tentativa de conexão era oriunda do sistema operacional GNU/Linux Fedora Core 4 e não do SGBD.

Este problema foi resolvido após um estudo detalhado das regras que definem o firewall ativo e efetuada a reconfiguração adequada. Os seguintes comandos foram necessários:

```
# iptables -nL
# -Xflush
```

Hoje, com o banco de dados FALCAO disponível para conexões, usa-se esta propriedade dos *firewalls* do Linux Fedora 4 para evitar-se conexões indevidas, habilitando-os manualmente, reforçando ainda mais a segurança de acesso aos dados do banco.

Vale também destacar que a habilitação do processo de transferência dos dados do reator ao servidor de banco de dados, embora hoje customizada para operar automaticamente, pode operar manualmente, nos moldes da habilitação dos *firewalls*. Atualmente este processo, denominado *vsftpd*, por motivos de controle, é ativado/desativado sob demanda, pelo administrador do banco de dados, na ocasião da transferência de uma massa de dados a carregar. A habilitação deste é de fundamental importância, e finaliza os procedimentos de conectividade ao banco de dados FALCAO.

5 RESULTADOS

Embora o resultado deste trabalho seja observado majoritariamente através da sua etapa final, ou seja, a disponibilização da informação através da aplicação *Web SACD*, etapas anteriores tais como a transferência e consistência criteriosas dos dados, a adequação do SGBD à carga dos dados, e a carga dos dados propriamente dita, precisaram ocorrer com sucesso para que a etapa de disponibilização também pudesse assim ocorrer.

Dentre estas etapas, destaca-se a adequação do SGBD à carga dos dados, etapa esta integralmente ligada à eficiência do modelo lógico de dados e que, por si só, justifica o cunho científico deste trabalho, conforme destacado em artigo recentemente publicado **NETO e ANDRADE [2007]**³¹.

A seguir são encontrados os resultados obtidos com estas implementações, etapa por etapa.

5.1 O SERVIDOR FALCAO

O servidor de banco de dados FALCAO, núcleo tecnológico deste trabalho, está completamente implementado e operante.

A TAB. 2, a seguir, contém a configuração atual do *hardware* do servidor de banco de dados FALCAO e sua comparação com o *hardware* mínimo, recomendado pelo fabricante do SGBD *ORACLE* 10G, conforme referência 19, quando operando sob sistema operacional Linux Fedora Core 4.

TABELA 2– Comparativo de *Hardware* do servidor FALCAO

<i>Hardware</i> mínimo recomendado pela <i>ORACLE</i>	Configuração do servidor FALCAO
512 MB de memória RAM	710 MB de memória RAM
1 GB para a partição swap	3 GB para a partição swap
400 MB no diretório /tmp	19 GB no diretório /tmp
2.1 GB para o <i>software ORACLE</i> Database 10G e o banco de dados	120 GB para o <i>software ORACLE</i> Database 10G e o banco de dados

Atualmente, o banco de dados FALCAO armazena informações referentes aos anos 2003, 2004, 2005, 2006, 2007 e o período Janeiro - Março de 2008, somando aproximadamente 1 GB em informações carregadas, e está dimensionado para receber arquivos da mesma ordem, com tolerância de dez por cento na variação de seu volume, ao longo dos próximos quatro anos.

Desta forma, ao final de 2011, teremos um histórico de 2GB em dados carregados, o que corresponde, aproximadamente, a vinte e seis milhões de linhas carregadas e disponíveis.

A FIG. 9 mostra graficamente, através da ferramenta gratuita denominada *Spotlight*, **EZ GOAL IT NETIX**³², o banco de dados FALCAO em operação. Através desta monitoração, é possível observar o comportamento do banco quando submetido a situações diversas, como cargas e acessos concorrentes, permitindo concluir mais rapidamente os ajustes físicos específicos, quer corretivamente, quer preditivamente.



FIGURA 9 - Monitoração do banco de dados FALCAO através do Spotlight

5.1.1 Afinidade das peculiaridades do SGBD Oracle 10G com este trabalho

A medida que novas versões do produto banco de dados *ORACLE* foram lançadas no mercado, saltos qualitativos foram dados. Pode-se citar dois altamente significativos, ligados diretamente a este trabalho: A introdução da linguagem *JAVA* ao banco de dados, da versão 8 para a versão 8i e a inovação trazida com o conceito de *grid*, na versão 10G, versão esta que apóia este trabalho. O *ORACLE* 10G é o primeiro banco de dados desenvolvido para computação *grid*, o que implica em um método mais flexível e com a melhor relação custo/ benefício para o gerenciamento da informação.

Embora o conceito de computação *grid* não seja novo, ele prevê um cenário em que todos os sistemas computacionais de uma organização, e eventualmente os de parceiros desta, sejam compartilhados, de modo a criar um grupo de recursos dinâmicos e um poderoso supercomputador virtual, formado por computadores particularmente limitados sob o aspecto processamento.

Essa idéia, que há alguns anos vem ganhando forma no mundo acadêmico, fundamentalmente em universidades e centros de pesquisa, só agora começa a surgir como opção para o mercado corporativo. As ofertas, embora ainda não bem aceitas pelas corporações, estão desembarcando no mercado através de fabricantes de *hardware*, desenvolvedores de *software* e fornecedores de unidades de discos. O *Enterprise Grid Computing*, rótulo comercial do fabricante *ORACLE* para a tecnologia *grid*, consiste na coordenação simultânea de vários servidores. Esses servidores funcionam integrados, como um grande recurso de computação, tendo como base altos níveis de automações de gerenciamento e padronização de *software*.

Desta forma, a tecnologia atualmente implementada no banco de dados FALCAO, somada às customizações peculiares deste trabalho, refletem, sobretudo, por conta das propriedades do produto *ORACLE* 10G, o excelente desempenho do banco de dados FALCAO, conseqüentemente observado na aplicação SACD, mesmo utilizando *hardware* discreto em ambos servidores.

Além disso, a tecnologia encontrada no produto *ORACLE* 10G permite, caso se mostre viável, a expansão do banco de dados FALCAO para arquiteturas mais sofisticadas, como por exemplo arquiteturas baseadas em métodos de tolerância a falha, *fail over*, incorporados por meio da redundância interna ou externa de componentes e/ou servidores.

5.1.2 Otimização de acesso aos dados versus Desempenho da aplicação SACD

Planos de execução sintetizam o caminho através do qual o banco de dados, utilizando-se de suas estatísticas, captura a informação.

Então, otimizar acesso aos dados traduz-se em fazer com que o otimizador do banco de dados, por conta de suas propriedades intrínsecas e configurações peculiares encontradas neste trabalho, escolha os melhores planos de execução e, por consequência desta escolha, gaste o menor tempo para interagir com o banco de dados, disponibilizando a informação solicitada rapidamente.

No caso do banco de dados FALCAO, este acesso foi exaustivamente testado e ajustado para que hoje, mesmo estando configurado para múltiplas conexões simultâneas e não utilizando os recursos da tecnologia *grid*, ofereça um tempo de resposta adequado a todas as estações de trabalho espalhadas pelo IPEN. Vale destacar que este desempenho se mantém mesmo quando operando o processo da carga de dados, ou seja, os acessos cotidianos em nada serão prejudicados, incluindo acessos simultâneos e consultas mais elaboradas freqüentemente mais dispendiosas ao sistema, quando novos dados estiverem sendo carregados pelo processo de carga de dados.

5.2 A TRANSFERÊNCIA DOS DADOS

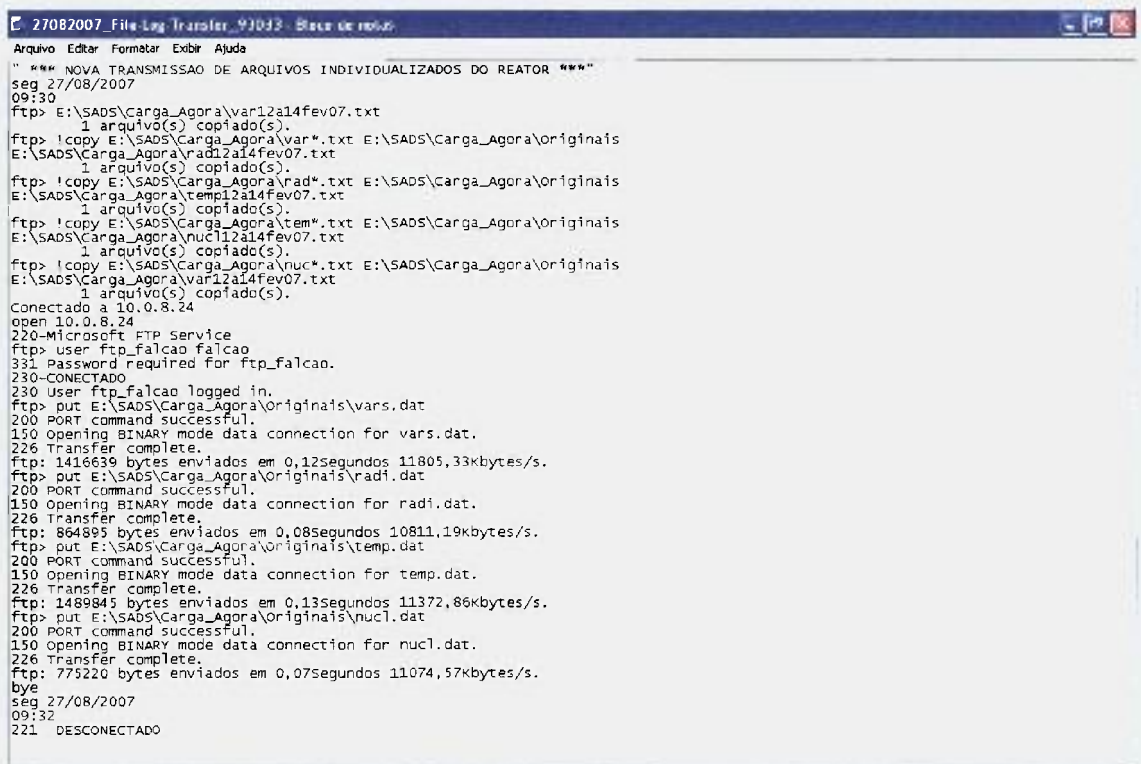
O processo de transferência dos dados está implementado e operando com êxito, estando também preparado para operar com freqüências de geração distintas, ou seja, maiores ou menores do que as freqüências hoje aplicadas.

Seu êxito ocorre efetivamente quando o arquivo no destino corresponde ao arquivo da origem, no que tange a denominação e ao conteúdo, integralmente.

Atualmente, este processo controla a transferência de aproximadamente 4,5 MB de dados por semana, em uma única transmissão, gastando para isso aproximadamente dois minutos, a uma taxa de transferência efetiva aproximada de 37,5 KB/s. Pode-se também, caso o sistema seja programado para tal, agendar transmissões em intervalos menores dos intervalos atualmente aplicados, transmitir-se a mesma massa de informações por semana, porém com freqüência maior e conseqüentemente, em arquivos menores. Assim, considerando que a semana operacional do reator IEA-R1 possui três dias, poder-se-ia transmitir diariamente volumes menores de dados com freqüência mais alta, totalizando o mesmo volume hoje transmitido em uma única vez.

Por exemplo, transmitir-se 187,5 KB a cada três horas totalizaria 4,5 MB hoje transferidos semanalmente de uma única vez. Com isso, menos recursos da rede interna do IPEN são utilizados e a informação é disponibilizada ao pesquisador quase instantaneamente após ser gerada.

A FIG. 10 mostra o conteúdo do arquivo que registra as etapas atuais de transferência, denominado *File-Log-Transfer*, cuja programação é encontrada no Apêndice B, contendo a denominação do próprio arquivo de transferência, a data da transferência, a denominação dos arquivos transferidos, o volume transferido por cada arquivo e indiretamente, o tempo total gasto na transferência.



```

270B2007_File-Log-Transfer_93043 - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
" *** NOVA TRANSMISSAO DE ARQUIVOS INDIVIDUALIZADOS DO REATOR ***"
seg 27/08/2007
09:30
ftp> E:\SADS\Carga_Agora\var12a14fev07.txt
1 arquivo(s) copiado(s).
ftp> !copy E:\SADS\Carga_Agora\var*.txt E:\SADS\Carga_Agora\originais
E:\SADS\Carga_Agora\radi12a14fev07.txt
1 arquivo(s) copiado(s).
ftp> !copy E:\SADS\Carga_Agora\radi*.txt E:\SADS\Carga_Agora\originais
E:\SADS\Carga_Agora\temp12a14fev07.txt
1 arquivo(s) copiado(s).
ftp> !copy E:\SADS\Carga_Agora\tem*.txt E:\SADS\Carga_Agora\originais
E:\SADS\Carga_Agora\nucl12a14fev07.txt
1 arquivo(s) copiado(s).
ftp> !copy E:\SADS\Carga_Agora\nuc*.txt E:\SADS\Carga_Agora\originais
E:\SADS\Carga_Agora\var12a14fev07.txt
1 arquivo(s) copiado(s).
Conectado a 10.0.8.24
open 10.0.8.24
220-Microsoft FTP Service
ftp> user ftp_falcao falcao
331 Password required for ftp_falcao.
230-CONECTADO
230 User ftp_falcao logged in.
ftp> put E:\SADS\Carga_Agora\Originais\vars.dat
200 PORT command successful.
150 Opening BINARY mode data connection for vars.dat.
226 Transfer complete.
ftp: 1416639 bytes enviados em 0,125segundos 11805,33kbytes/s.
ftp> put E:\SADS\Carga_Agora\Originais\radi.dat
200 PORT command successful.
150 Opening BINARY mode data connection for radi.dat.
226 Transfer complete.
ftp: 864895 bytes enviados em 0,085segundos 10811,19kbytes/s.
ftp> put E:\SADS\Carga_Agora\Originais\temp.dat
200 PORT command successful.
150 Opening BINARY mode data connection for temp.dat.
226 Transfer complete.
ftp: 1489845 bytes enviados em 0,133segundos 11372,86kbytes/s.
ftp> put E:\SADS\Carga_Agora\Originais\nucl.dat
200 PORT command successful.
150 Opening BINARY mode data connection for nucl.dat.
226 Transfer complete.
ftp: 775220 bytes enviados em 0,075segundos 11074,57kbytes/s.
bye
seg 27/08/2007
09:32
221 DESCONECTADO

```

FIGURA 10 – Listagem do arquivo que registra o processo de transferência dos dados

5.3 A CARGA DOS DADOS

Assim como na customização do servidor de banco de dados e no processo de transferência dos dados, após exaustivos testes, o processo de carga também está totalmente implementado e operante. O processo de carga resume-se em gravar as informações dos arquivos oriundos do reator IEA-R1 nas tabelas do banco de dados FALCAO, segundo critérios citados anteriormente e em seguida, sinalizar o término do processo, obtendo êxito neste, ou não.

Logo, é necessário que este processo verifique e registre todas as etapas relacionadas as partes envolvidas, ou seja, arquivos e tabelas.

A FIG. 11 apresenta o registro de todos os detalhes da carga do arquivo de temperaturas, denominado controle_carga_reator_temperatura.log. Sua programação bem como detalhes da carga dos arquivos de vazões, radiações e grandezas nucleares são encontrados no Apêndice B.

```

controle_carga_reator_temperatura
Arquivo Editar Formatar Exibir Ajuda
SQL*Loader: Release 10.1.0.2.0 - Production on Seg Set 03 09:29:41 2007
Copyright (c) 1982, 2004, Oracle. All rights reserved.

  Column Name          Position   Len  Term Enc Datatype
-----
GERACAO                1:14      14           CHARACTER
T1                     24:27      4           CHARACTER
T2                     29:32      4           CHARACTER
T3                     34:37      4           CHARACTER
T4                     39:42      4           CHARACTER
DT1                    44:47      4           CHARACTER
T6                     49:52      4           CHARACTER
T7                     54:57      4           CHARACTER
T8                     59:62      4           CHARACTER
T9                     64:67      4           CHARACTER
DT2                    72:76      5           CHARACTER

7455 Rows successfully loaded.
1 Row not loaded due to data errors.
0 Rows not loaded because all WHEN clauses were failed.
0 Rows not loaded because all fields were null.

Space allocated for bind array:          13312 bytes(64 rows)
Read  buffer bytes: 1048576

Total logical records skipped:          0
Total logical records read:             7456
Total logical records rejected:         1
Total logical records discarded:        0

Run began on Seg Ago 27 09:29:41 2007  Elapsed time was:    00:00:01.08
Run ended on Seg Ago 27 09:29:42 2007  CPU time was:        00:00:00.19

```

FIGURA 11 - Registro da carga do arquivo de temperatura no banco de dados FALCAO

Em seus respectivos conteúdos, estão gravados os detalhes relacionados a data exata da carga, o tempo gasto para efetuá-la, a estrutura da tabela que recebe os dados, a quantidade de linhas carregadas, os erros referentes a eventuais inconsistências no conteúdo do arquivo carregado, o total destes erros, os problemas com a carga em si, caso ocorram, e os recursos do servidor de banco de dados utilizados no período da carga, recursos estes configuráveis e necessários para que a carga ocorra com sucesso.

5.3.1 Procedimento periódico de garantia de recuperação dos dados

Hoje, o banco de dados FALCAO está inserido numa arquitetura computacional simples, que não contempla redundância, ou seja, não existe sistema reserva operando em conjunto com ele. Desta forma, sobretudo para esta arquitetura, tão importante quanto a carga dos dados, é a existência de um procedimento eficiente que garanta a cópia periódica das informações contidas no banco, assim como sua recarga, em caso de eventuais falhas.

O mecanismo de cópia e recuperação, *back-up* e *recover*, intrínseco ao produto *ORACLE* 10G, denominado *RMAN* é utilizado neste trabalho. A frequência com que devem ocorrer foi configurada, os mecanismos foram testados e hoje se encontram operando com sucesso.

Atualmente este procedimento ocorre de duas formas, ou seja, uma cópia total dos dados ocorre a cada quatro semanas corridas, nos dias em que o reator IEA-R1 não opera, enquanto que mecanismos de cópias incrementais, complementares ao procedimento anterior, ocorrem enquanto o banco de dados FALCAO permanecer ativo, ou seja, são automáticos, embora configuráveis. Desta forma, hoje esta implementação confere ao banco de dados FALCAO a segurança necessária aliada a certeza de que seus dados podem ser recuperados na íntegra.

Para o volume atual de dados carregados, esta recuperação ocorreria em torno de quatro horas.

5.4 A APLICAÇÃO SACD

Da mesma forma que as etapas anteriores, esta etapa está completamente implementada e disponível.

A FIG. 12 mostra a tela inicial da aplicação SACD, tela *logon*, atrelando sua utilização a um usuário autorizado, seguido de um código particular de acesso.

A validação do usuário no banco de dados precede quaisquer outros procedimentos.



FIGURA 12 - Tela de autenticação - SACD

A FIG. 13 mostra a tela principal da aplicação SACD, da qual é possível visualizar-se as funcionalidades do sistema.



FIGURA 13 - Tela principal da aplicação SACD

A FIG. 14 apresenta a tela inicial da área termo-hidráulica e suas opções.



FIGURA 14 - Tela inicial termo-hidráulica

A FIG.15 apresenta a funcionalidade através da qual é possível visualizar-se o gráfico das temperaturas do circuito primário do reator IEA-R1.

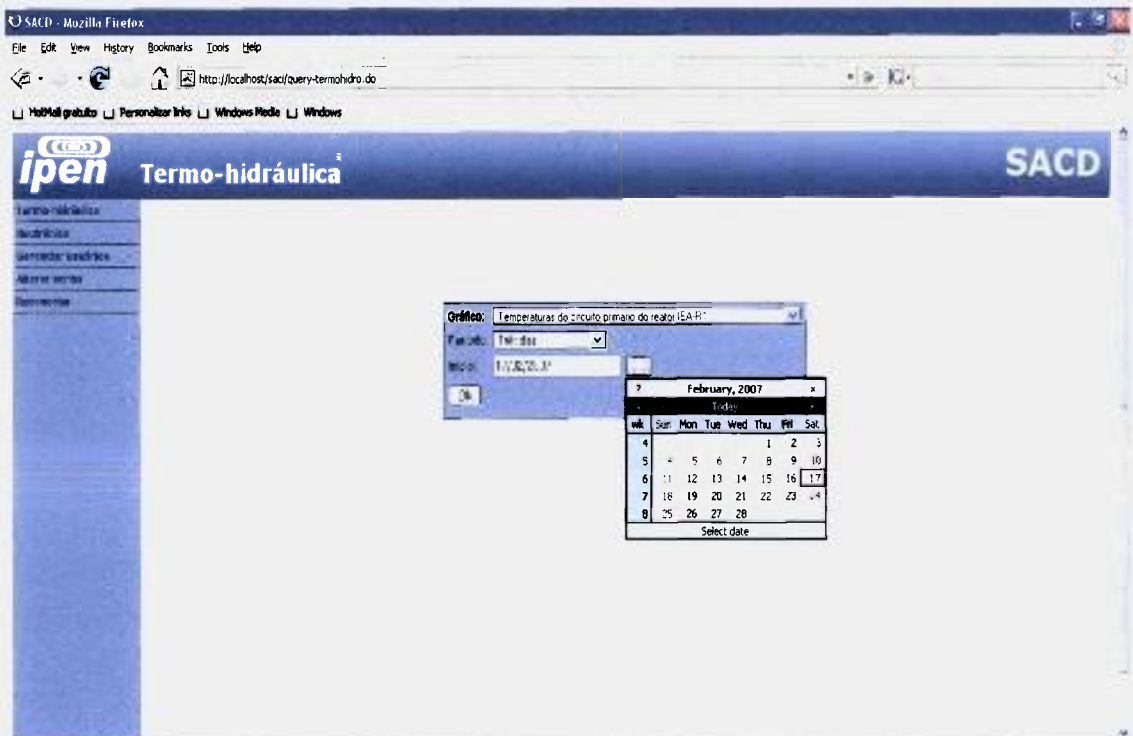


FIGURA 15 - Tela inicial termo-hidráulica - detalhamento

6 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Para que haja maior clareza na análise e discussão dos resultados obtidos, e sobretudo enfatizando os objetivos principais deste trabalho, comparar-se-á o cotidiano dos pesquisadores que se utilizam dos valores das variáveis monitoradas no reator IEA-R1, antes e após a implementação deste trabalho.

6.1 COTIDIANO DAS ATIVIDADES SEM A UTILIZAÇÃO DOS RECURSOS GERADOS POR ESTE TRABALHO

A seguir um exemplo típico de utilização normal, representado pelas FIG.16, 17 e 18. A FIG. 18 representa a etapa de um procedimento termo-hidráulico cotidiano, por parte dos pesquisadores do reator IEA-R1, onde parte dos dados gerados pelo sistema SAD são manualmente transferidos de um arquivo tipo texto, FIG. 16, para uma planilha eletrônica, FIG. 17, para que nesta, possam ser posteriormente formatados.

Date*Time	Con	ope	T1	T2	T3	T4	OT	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21
01/05 11:00 211	25.4	25.9	26.0	24.4	-1.6	23.9	5.0	25	24.2	25.0	24.3	23.0	24.3	24.4	1.6	22.5	24.9	26.6	26.4	27.1	27.3		
01/05 11:01 211	25.5	26.0	26.0	24.4	-1.6	23.9	5.0	25	24.2	24.9	24.3	23.9	24.3	24.4	1.6	22.6	24.9	26.6	26.4	27.1	27.3		
01/05 11:02 211	25.5	25.9	26.0	24.4	-1.6	23.9	5.0	25	24.3	24.9	24.4	23.8	24.2	24.5	1.6	22.5	24.9	26.6	26.4	27.1	27.3		
01/05 11:03 211	25.5	25.9	26.0	24.4	-1.6	23.9	5.0	25	24.2	25.0	24.3	23.9	24.3	24.5	1.6	22.6	24.9	26.6	26.4	27.1	27.4		
01/05 11:04 211	25.5	25.9	26.0	24.4	-1.6	23.9	5.0	25	24.3	25.0	24.4	23.9	24.3	24.4	1.6	22.5	24.9	26.6	26.4	27.1	27.4		
01/05 11:05 211	25.5	25.9	26.0	24.4	-1.6	23.8	5.0	25	24.3	25.0	24.4	23.9	24.3	24.4	1.6	22.5	24.9	26.6	26.4	27.1	27.4		
01/05 11:06 211	25.5	25.9	26.0	24.4	-1.6	23.8	5.0	25	24.2	25.0	24.4	23.9	24.3	24.4	1.6	22.5	24.9	26.6	26.4	27.1	27.4		
01/05 11:07 211	25.5	25.9	26.0	24.4	-1.6	23.8	5.0	25	24.2	24.9	24.3	23.9	24.3	24.5	1.6	22.5	24.9	26.6	26.4	27.1	27.4		
01/05 11:08 211	25.5	25.9	26.0	24.4	-1.6	23.8	5.0	25	24.3	24.9	24.3	23.9	24.3	24.5	1.6	22.6	24.9	26.6	26.4	27.1	27.4		
01/05 11:09 211	25.5	25.9	26.0	24.4	-1.6	23.8	5.0	25	24.2	25.0	24.4	23.9	24.3	24.4	1.6	22.7	24.9	26.6	26.4	27.1	27.4		
01/05 11:10 211	25.5	25.9	26.0	24.4	-1.6	23.8	5.0	25	24.3	25.0	24.4	23.9	24.3	24.5	1.6	22.8	24.9	26.6	26.4	27.1	27.4		
01/05 11:11 211	25.5	26.0	26.0	24.4	-1.6	23.8	5.0	25	24.2	25.0	24.4	23.9	24.3	24.5	1.6	23.0	24.9	26.6	26.4	27.1	27.4		
01/05 11:12 211	25.5	26.0	26.0	24.4	-1.6	23.8	5.0	25	24.3	25.0	24.4	23.9	24.3	24.4	1.6	23.0	24.9	26.6	26.4	27.1	27.5		
01/05 11:13 211	25.5	26.0	26.0	24.4	-1.6	23.8	4.9	25	24.3	24.9	24.4	23.9	24.3	24.4	1.6	23.0	24.9	26.6	26.4	27.1	27.4		
01/05 11:14 211	25.5	26.0	26.0	24.4	-1.6	23.8	5.0	25	24.3	25.0	24.4	23.9	24.3	24.4	1.6	23.1	24.9	26.6	26.4	27.1	27.4		
01/05 11:15 211	25.5	26.0	26.0	24.4	-1.6	23.8	5.0	25	24.2	25.0	24.4	23.9	24.3	24.4	1.6	23.0	24.9	26.6	26.4	27.1	27.5		
01/05 11:16 211	25.5	26.0	26.0	24.4	-1.6	23.8	4.9	25	24.2	25.0	24.4	23.9	24.3	24.4	1.6	23.0	24.9	26.6	26.4	27.1	27.4		
01/05 11:17 211	25.5	26.0	26.0	24.4	-1.6	23.8	5.0	25	24.3	25.0	24.4	23.9	24.3	24.4	1.6	23.0	24.9	26.6	26.4	27.1	27.5		
01/05 11:18 211	25.5	25.9	26.0	24.4	-1.6	23.9	5.0	25	24.2	25.0	24.4	23.9	24.3	24.4	1.6	23.3	24.9	26.6	26.4	27.1	27.5		
01/05 11:19 211	25.5	26.0	26.0	24.4	-1.6	23.8	4.9	25	24.2	25.0	24.4	23.9	24.3	24.5	1.6	23.3	24.9	26.6	26.4	27.1	27.5		
01/05 11:20 211	25.5	26.0	26.0	24.4	-1.6	23.8	5.0	25	24.3	25.0	24.4	23.9	24.3	24.4	1.6	23.3	24.9	26.6	26.4	27.1	27.5		
01/05 11:21 211	25.5	25.9	26.0	24.4	-1.6	23.8	4.9	25	24.2	25.0	24.4	23.9	24.3	24.5	1.6	23.2	24.9	26.6	26.4	27.1	27.5		
01/05 11:22 211	25.5	26.0	26.0	24.4	-1.6	23.8	5.0	25	24.3	25.0	24.4	23.9	24.3	24.5	1.6	23.2	24.9	26.6	26.4	27.1	27.5		
01/05 11:23 211	25.5	25.9	26.0	24.4	-1.6	23.9	5.0	25	24.2	25.0	24.3	23.0	24.3	24.4	1.6	23.2	24.9	26.6	26.4	27.1	27.5		

FIGURA 16 - Arquivo tipo txt gerado pelo sistema SAD

6.2 COTIDIANO DAS ATIVIDADES UTILIZANDO OS RECURSOS GERADOS POR ESTE TRABALHO

O gráfico da FIG. 19, embora seja equivalente ao gráfico manualmente gerado da FIG. 18, permite uma análise mais eficiente e é confeccionado e disponibilizado aos pesquisadores em tempo hábil sensivelmente menor. Para a infraestrutura utilizada este tempo é de aproximadamente 1,5 minutos. Nesta, os dados são oriundos do banco de dados FALCAO.

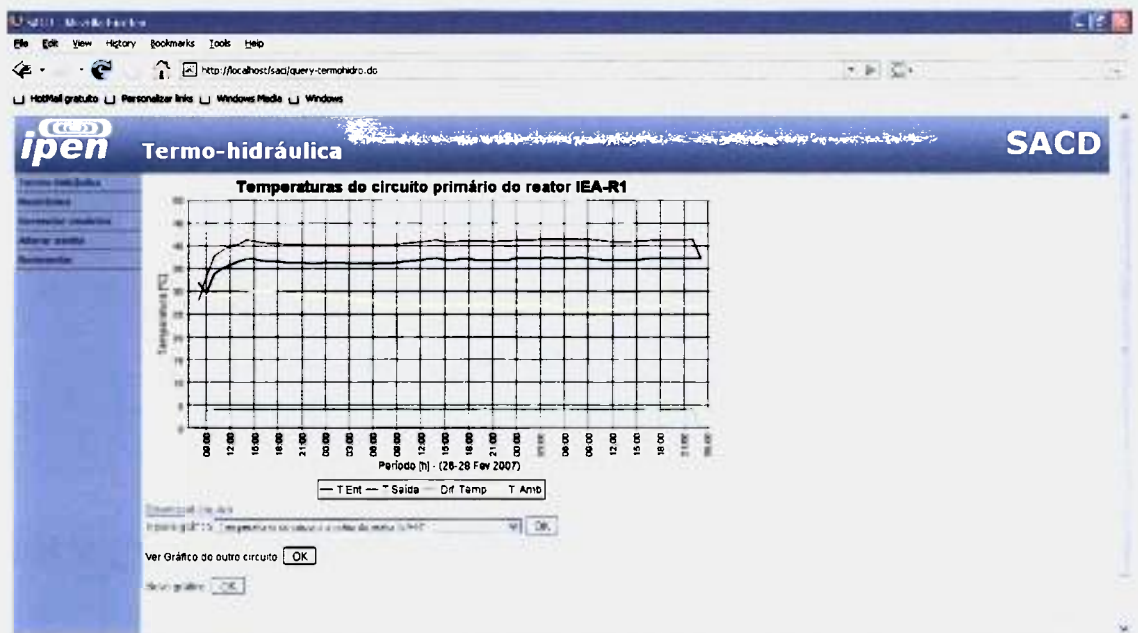


FIGURA 19- Temperatura - circuito primário do reator IEA-R1 via SACD

Os dados utilizados para a confecção do gráfico da FIG. 19, também podem ser automaticamente disponibilizados na forma de arquivo texto, ou em forma de planilha eletrônica, através da opção Download arquivo, caso o pesquisador além do gráfico, necessite dos dados em alguma destas formas.

Estas opções são apresentadas pelas planilhas apresentadas nas FIG. 20 e 21, respectivamente.

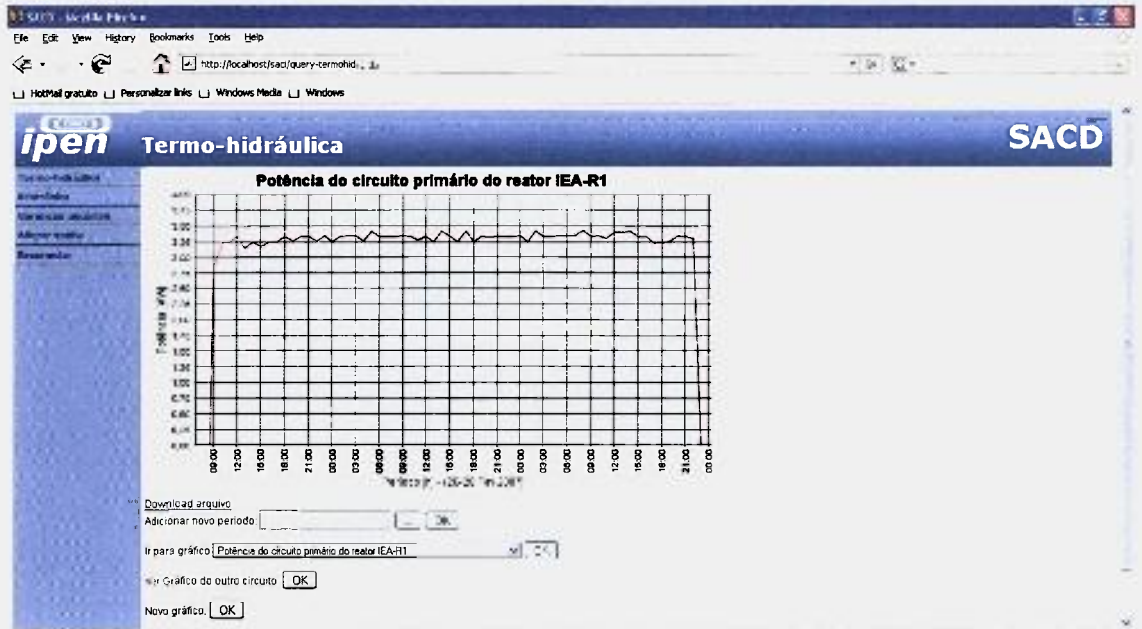


FIGURA 22 - Potência do circuito primário do reator IEA-R1 via SACD

A FIG. 23, gerada instantaneamente, permite a comparação do mesmo fenômeno em períodos distintos, reduzindo ainda mais o tempo gasto, se comparado ao procedimento manual outrora aplicado. Neste exemplo, apresenta-se a potência do circuito primário do reator IEA-R1 nos períodos 26 de fevereiro a 1 de março de 2007 e 12 a 15 de fevereiro de 2007.

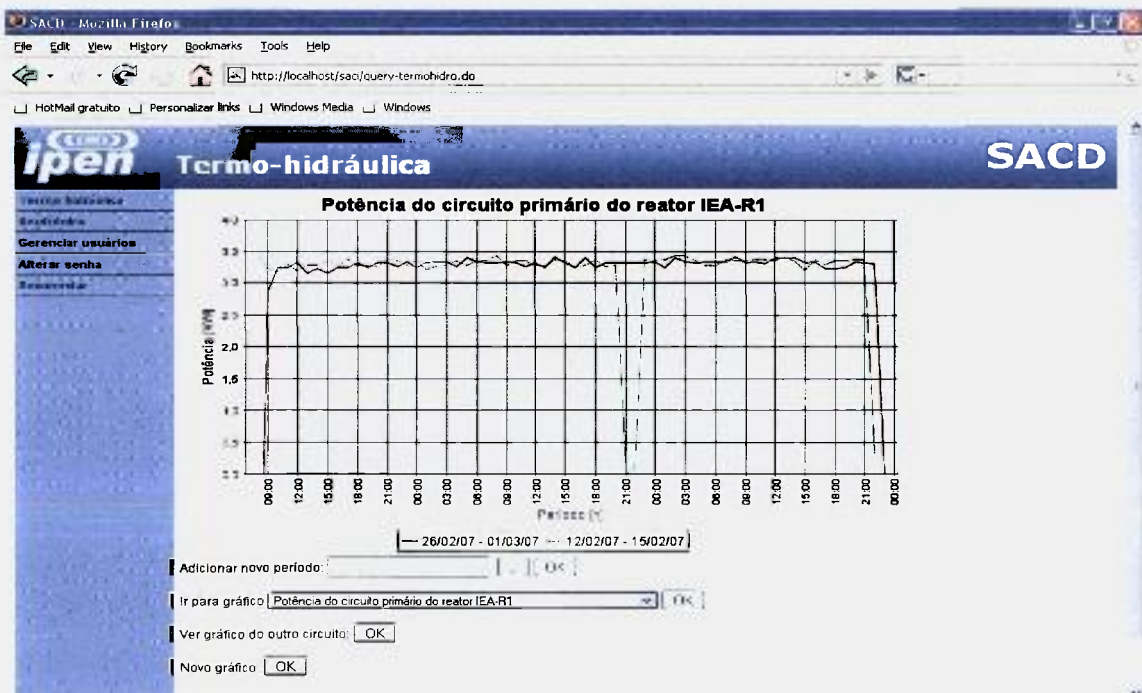


FIGURA 23 - Potência do circuito primário do IEA-R1 - períodos distintos via SACD

Analogamente, a FIG. 24 também se refere às potências, porém apresenta para o mesmo período, potências distintas. Neste exemplo, apresentam-se as potências dos circuitos primários e secundários do reator IEA-R1 no período de 26 a 28 de fevereiro de 2007.

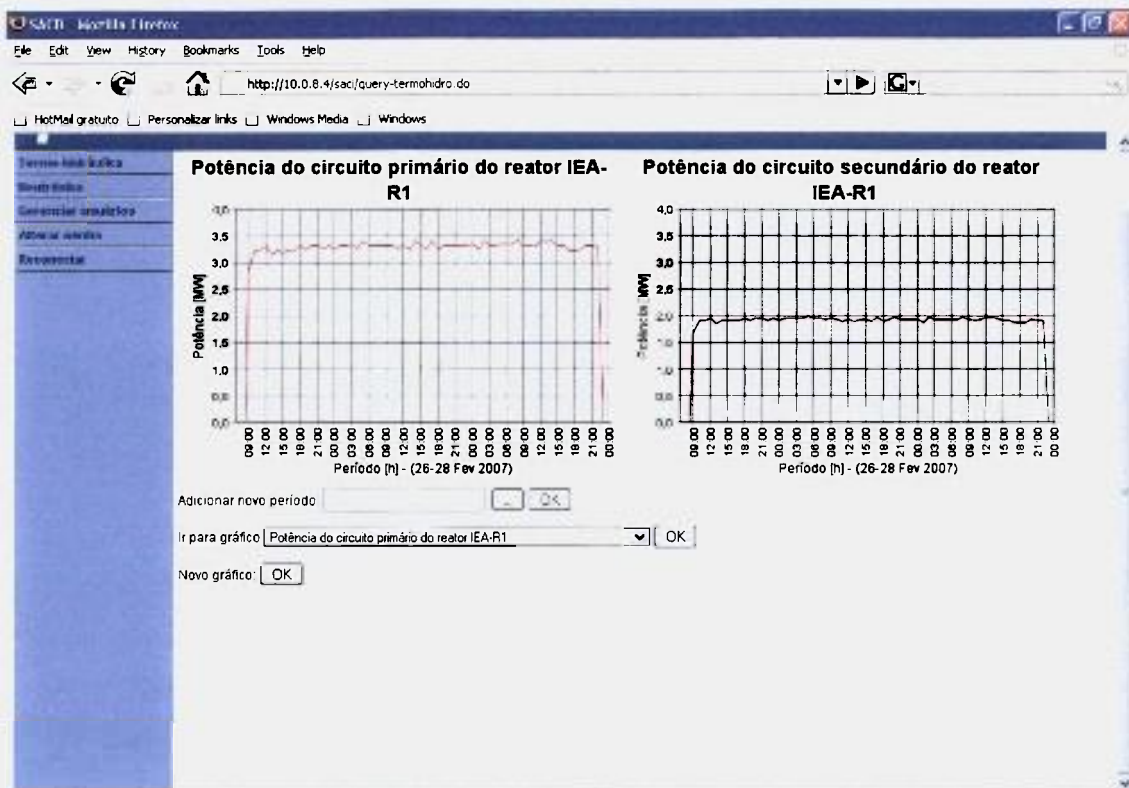


FIGURA 24- Potência dos circuitos primário e secundário, para o mesmo período, via SACD

A potência é função direta da vazão e da variação de temperatura. No circuito secundário, exceto as pequenas perdas admissíveis, a potência deve coincidir com a potência do circuito primário. Na FIG. 24 observa-se que a potência no circuito primário é de aproximadamente 3,4 MW.

Já a potência observada para o circuito secundário apresenta uma diferença de aproximadamente 44 % em relação ao circuito primário, diferença que não pode ser associada a perdas admissíveis. Esta aparente incoerência se deve sobretudo a existência de imprecisões dos instrumentos ligados a leitura da grandeza temperatura, o que pode ser observado através da variação não significativa das grandezas T1, T2, T3 e T4, FIG. 25, referentes às medidas de temperatura na água da piscina. Cabe lembrar que a diferença entre as potências dos dois circuitos, se deve exclusivamente a problemas/imprecisão dos instrumentos utilizados na medição da temperatura, ou seja, este erro está

diretamente ligado a leitura da informação e não ao seu registro, transferência, armazenagem, processamento ou disponibilização.

Temp_26_fev_08.txt

Date	Time	Con	Ope	T1	T2	T3	T4	DT1	T6	T7	T8	T9
26/02/07	08:06	10	10	25,1	25,9	25,6	24,8	-0,8	24,4	26,7	26,4	25,5
26/02/07	08:07	10	10	25,2	25,9	25,6	24,8	-0,8	24,4	26,6	26,4	25,4
26/02/07	08:07	10	10	25,2	25,9	25,6	24,8	-0,8	24,4	26,6	26,5	25,5
26/02/07	08:08	10	10	25,1	25,9	25,6	24,7	-0,9	24,4	26,6	26,4	25,4
26/02/07	08:08	10	10	25,2	25,9	25,6	24,7	-0,9	24,3	26,6	26,4	25,5
26/02/07	08:09	10	10	25,2	25,9	25,6	24,8	-0,8	24,4	26,6	26,4	25,5
26/02/07	08:09	10	10	25,2	25,9	25,6	24,8	-0,8	24,4	26,7	26,4	25,5
26/02/07	10:19	10	10	25,1	25,9	25,6	24,8	-0,8	24,4	26,7	26,4	25,4
26/02/07	10:20	10	10	25,1	25,9	25,7	24,8	-0,9	24,4	26,6	26,4	25,4
26/02/07	10:21	10	10	25,1	25,9	25,7	24,8	-0,9	24,3	26,6	26,4	25,5
26/02/07	10:21	10	10	25,1	25,9	25,6	24,8	-0,8	24,4	26,6	26,4	25,5
26/02/07	10:22	10	10	25,2	25,9	25,6	24,8	-0,8	24,3	26,6	26,4	25,4
26/02/07	10:22	10	10	25,1	25,9	25,6	24,8	-0,8	24,4	26,6	26,4	25,4
26/02/07	10:23	10	10	25,1	25,9	25,6	24,8	-0,8	24,4	26,6	26,4	25,5
26/02/07	14:16	10	10	25,2	25,9	25,6	24,8	-0,8	24,4	26,7	26,4	25,4
26/02/07	14:17	10	10	25,2	25,9	25,6	24,8	-0,8	24,4	26,6	26,4	25,4
26/02/07	14:17	10	10	25,2	25,9	25,5	24,8	-0,7	24,3	26,6	26,4	25,4
26/02/07	14:18	10	10	25,1	25,9	25,6	24,8	-0,8	24,4	26,6	26,4	25,5
26/02/07	14:18	10	10	25,1	25,9	25,6	24,7	-0,9	24,3	26,6	26,4	25,4
26/02/07	14:19	10	10	25,1	25,9	25,6	24,7	-0,9	24,4	26,6	26,4	25,4
26/02/07	14:19	10	10	25,1	25,8	25,5	24,7	-0,8	24,4	26,6	26,4	25,4

FIGURA 25- Temperatura da água da piscina do reator IEA-R1

Outro exemplo de utilização do sistema SACD é evidenciado em um procedimento ligado a neutrônica, mais especificamente na análise do comportamento neutrônico do reator nuclear IEA-R1. Nesta análise, um dos parâmetros mais importantes é o valor do fluxo de nêutrons no núcleo do reator.

Uma avaliação correta deste fluxo de nêutrons é necessária para uma boa estimativa da distribuição espacial da potência do reator e de outros parâmetros de igual relevância para uma operação segura, tais como a margem de desligamento do reator, valor de barras de segurança e controle, curvas envelope para os fatores de potência, curva S etc.

Desde os primórdios da engenharia nuclear, esta tem sido uma das principais tarefas da área de física dos reatores. Desde então vários métodos foram desenvolvidos com o propósito de calcular o fluxo de nêutrons, levando em consideração a complexidade da geometria e da composição do reator.

Tão importante quanto a avaliação do fluxo de nêutrons, é a estimativa do tempo de operação do reator associado ao controle efetivo de desligamento deste, caso necessário.

Este controle, diretamente ligado as barras absorvedoras, segurança e controle, quantifica, por processo empírico iterativo, a capacidade de cada barra em desligar o reator. No IEA-R1, estas barras somam quatro, distribuídas entre três de segurança e uma de controle.

Desta forma, sabendo-se a capacidade que cada barra possui de desligar o reator, associa-se o somatório destas capacidades à capacidade do arranjo absorvedor como um todo, incluindo neste um coeficiente de segurança de valor próximo a dois.

A esta capacidade conjunta denomina-se Reatividade do arranjo absorvedor ou simplesmente Reatividade. Esta capacidade, obtida para cada barra absorvedora na ocasião do rearranjo do combustível nuclear, varia de barra para barra e é função direta de suas posições.

A FIG. 26, também coloquialmente conhecida por curva *S*, mostra esta relação. Nota-se que as barras absorvedoras têm seus cursos variando da abscissa 0 [mm], posição correspondente a barra totalmente inserida, condição subcrítica do reator, à abscissa 1000 [mm], posição correspondente a barra totalmente retirada, condição supercrítica do reator.

Com a “queima” do combustível nuclear, as barras tendem gradativamente à abscissa mil, partindo da abscissa correspondente a condição crítica, ponto de partida do reator, compreendido entre 0 [mm] e 1000 [mm], e definido para cada barra na ocasião do reabastecimento do reator, através de um processo denominado calibração.

Assim, sabendo-se a reatividade de cada barra, através da função matemática que interpola a curva *S*, as posições iniciais destas barras, através das respectivas condições críticas, a posição instantânea de cada barra e a potência de operação real do reator no momento da consulta, calcula-se o exato valor do excesso de reatividade do arranjo, E_r , e, conseqüentemente, o tempo restante de operação do reator. O banco FALCAO armazena todas estas informações, fornecendo subsídios para estes cálculos, instantaneamente.

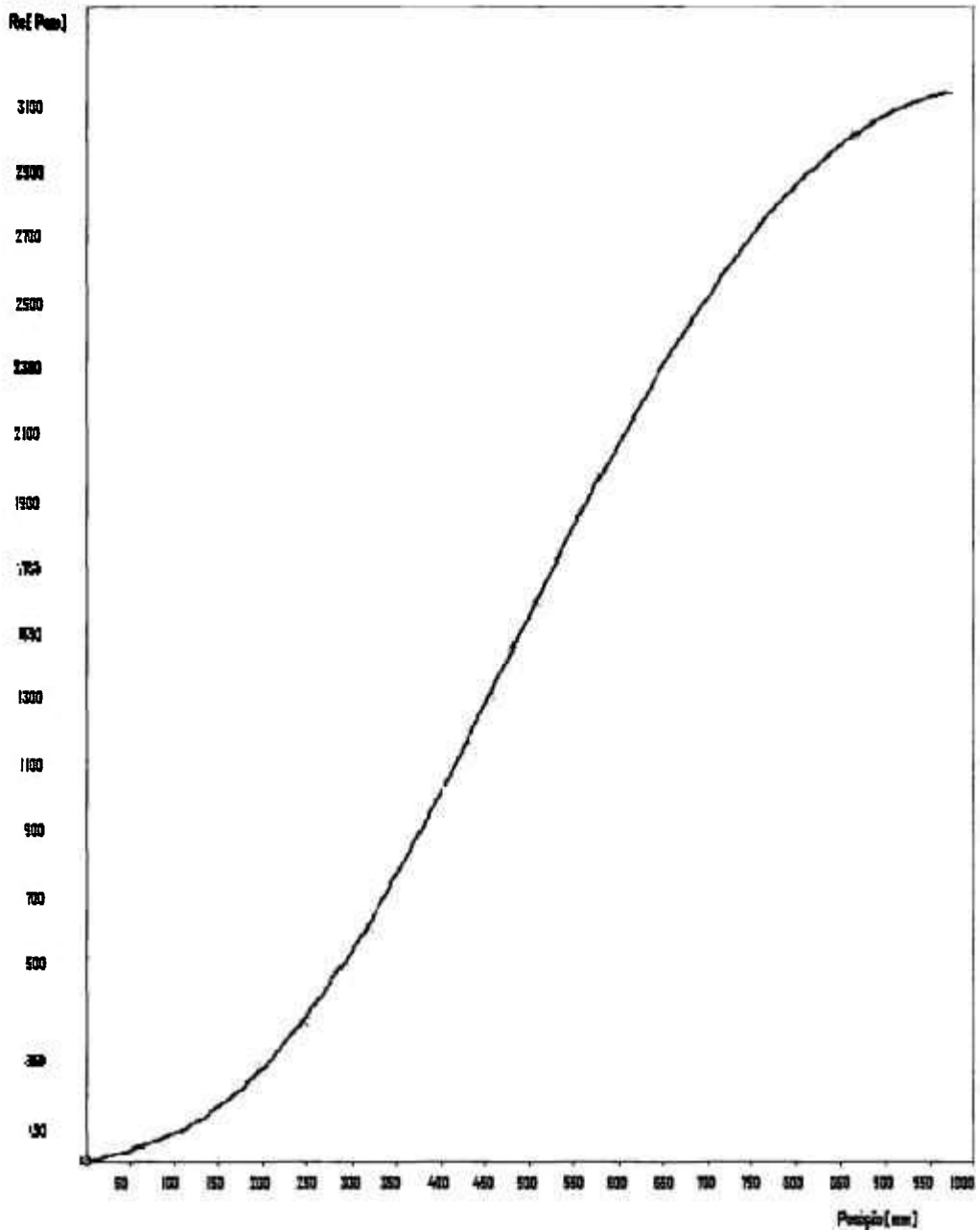


FIGURA 26 – Função reatividade

A FIG. 27 mostra a reatividade da barra de segurança número 2, correspondente ao arranjo número 234 do reator IEA-R1. Traz também a posição crítica da barra, coordenada X do ponto crítico, P_c , e o Er desta, obtido pelo módulo da diferença entre a coordenada y do ponto crítico e a máxima reatividade da barra, coordenada y do ponto P_m , expressa em [Pcm], partes por cem mil.

Então, para a barra da FIG. 27, o Er vale aproximadamente 1444 [Pcm]. Este valor representa a parcela desta barra na composição do Er do arranjo total, imediatamente após o reabastecimento.

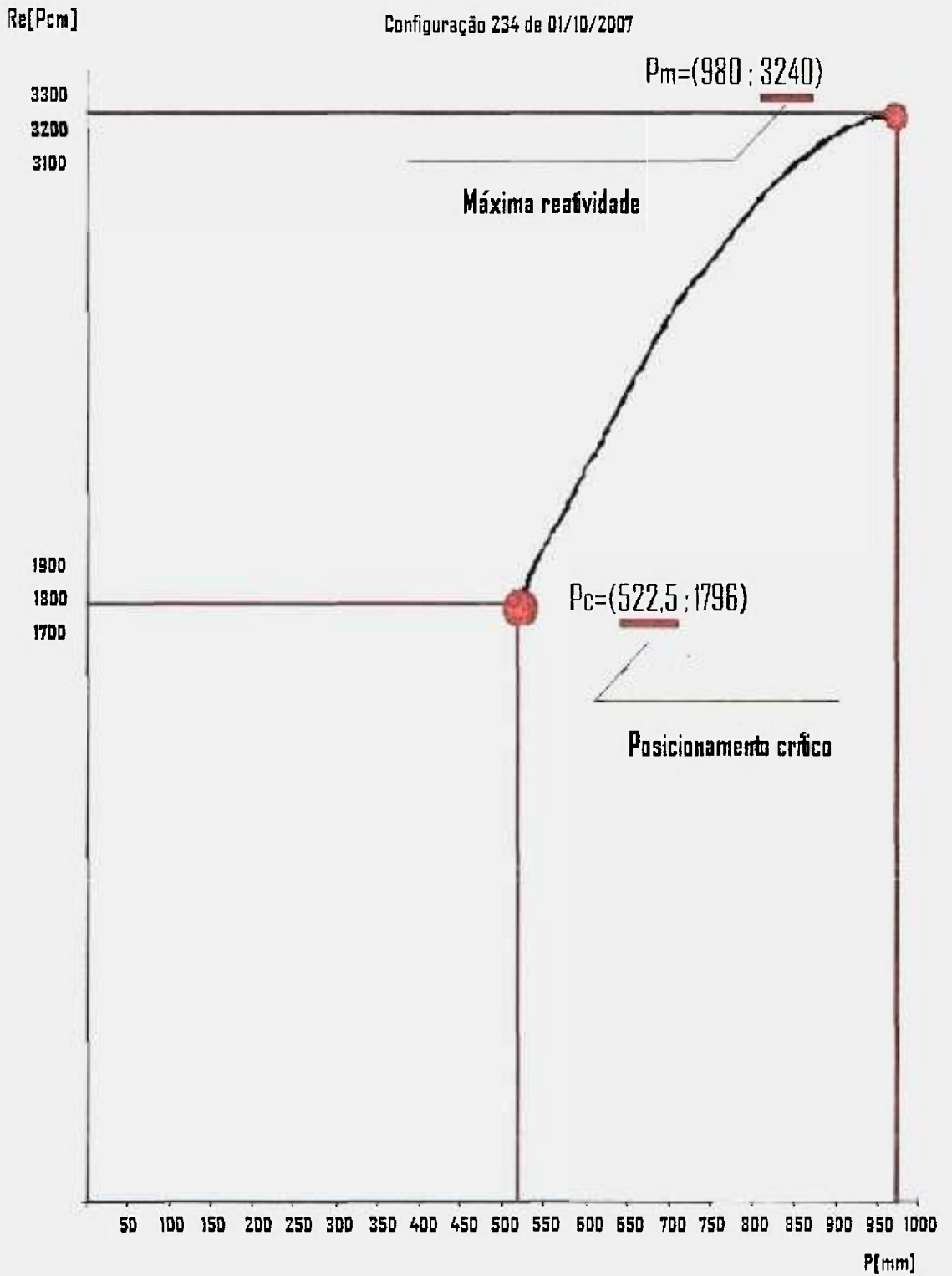


FIGURA 27 – Posição crítica na curva S

Até então inteiramente manual, o cálculo do Er era baseado exclusivamente na curva S, estimando-se neste um período fixo de operação do reator e atribuindo a este período um valor também fixo de potência de operação: 40 h

semanais a 4 MW de potência. Com o sistema SACD, o cálculo do Er , além de instantâneo, tornou-se mais preciso.

A FIG. 28 apresenta o Er do arranjo 234, instantes após a partida do reator e conseqüentemente, ainda desconsiderando a influência das sessões de choque geradas pelo fator Xenônio, também conhecido por “veneno”.

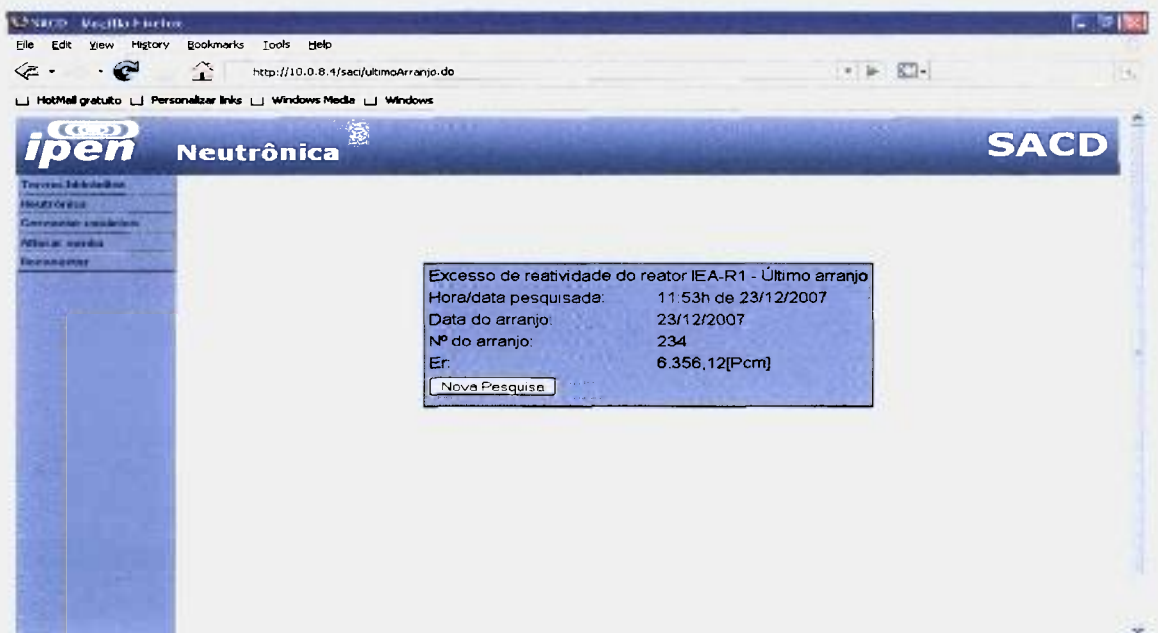


FIGURA 28 – Er inicial do arranjo 234, via o sistema SACD

A FIG. 29 apresenta o Er do arranjo 234, 3 dias após a ligação do reator. Os valores obtidos mostram o Er já considerando a “queima do veneno”. Nesta tela pode-se observar em detalhes os valores do Er de cada barra, assim como as respectivas posições destas.

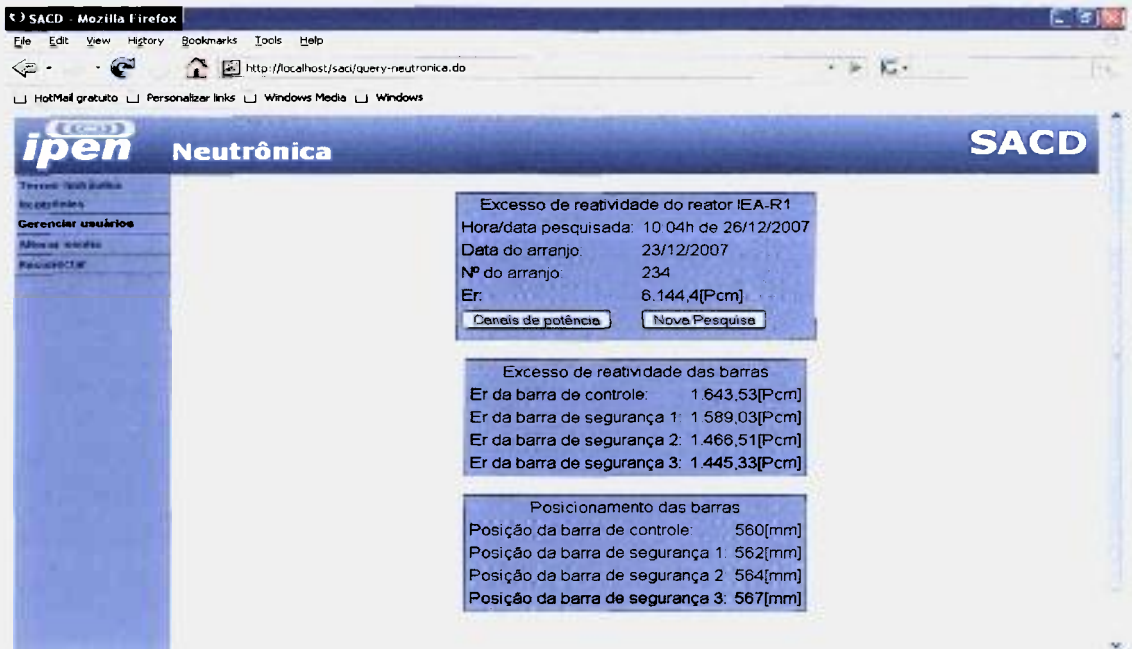


FIGURA 29 – Er do arranjo 234, em 26/12/2007

A FIG. 30 apresenta os valores de corrente referentes aos canais de potência, primário e secundário, do reator IEA-R1.

Estas informações são úteis caso seja necessário concluir-se, através destas grandezas, a cerca da potência dos circuitos primário e secundário do reator.

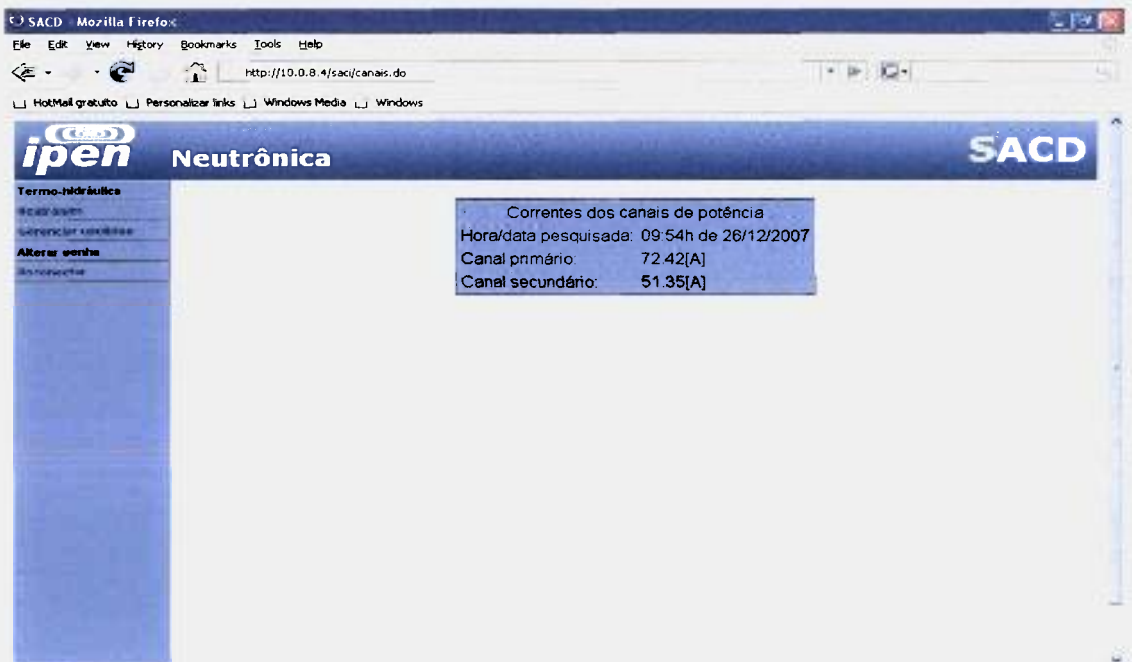


FIGURA 30 - Valores de corrente dos canais de potência do reator IEA-R1, em 26/12/2007

Como dito anteriormente, a semana útil do reator tem menos de 40 h e a potência média de operação oscila em torno de 3,3 MW, como mostra a FIG. 22.

Logo, nota-se um superdimensionamento do cálculo da ordem de 17,5 % na parcela referente a potência de operação e 60 % na parcela referente ao tempo de operação, o que resulta em um subdimensionamento substancial no cálculo da autonomia do reator. Em termos práticos, significa que o reator poderia operar por mais tempo antes de ser desligado para reabastecimento.

A operação de reabastecimento é uma operação dispendiosa e trabalhosa.

Com a utilização do SACD obtém-se por conta da precisão associada ao cálculo, um melhor aproveitamento de recursos, tanto humanos, quanto materiais, quando da estimativa de reabastecimento do reator IEA-R1.

7 CONCLUSÕES

A proposta de implementação do banco de dados FALCAO ocorreu com o êxito esperado. Os pesquisadores ligados a área de termo-hidráulica, cujas demandas foram bem definidas, hoje têm disponível em seu cotidiano de pesquisas, através da aplicação SADC, uma fonte de referência histórica de dados, quer para geração de relatórios, quer para comparações gráficas. Os pesquisadores da área de neutrônica, da mesma forma que seus colegas da área de termo-hidráulica, utilizam as informações do banco de dados FALCAO em seu cotidiano técnico para apoio a decisões estratégicas, através da análise de gráficos e cálculos dinâmicos. Aplicações como, por exemplo, análises de transientes, poderiam ser melhor viabilizadas caso a frequência de coleta dos dados, fator que independe do banco FALCAO e sim, exclusivamente do sistema SAD, fosse maior do que as frequências hoje aplicadas.

O banco de dados FALCAO se mostra indispensável ao cotidiano de uma determinada parcela de pesquisadores do IPEN, por conta da substancial melhoria dos processos atuais, quando comparados aos processos anteriores ao banco de dados FALCAO. O fator mais significativo de melhoria é o menor tempo gasto na aquisição de informações, associado à versatilidade e confiabilidade que as funcionalidades do sistema SADC fornecem.

7.1 A PROPOSTA ATUAL

Nas dependências do CEN, o banco de dados FALCAO se encontra disponível e operante, após ter sido exaustivamente testado pelos processos de transmissão, carga e disponibilização de dados, para atingir o estágio operacional que se encontra. Atualmente o banco FALCAO disponibiliza informações referentes aos últimos cinco anos de operação do reator IEA-R1, estando também devidamente dimensionado para receber dados de mesma ordem ao longo dos próximos quatro anos (até 2011).

7.2 CONTINUIDADE DO TRABALHO

Sem dúvida, existe muito a se fazer no sentido de colaborar com a automação de processos complexos, particularmente processos nucleares, como o processo que justifica a existência deste estudo.

Um sistema gerenciador de banco de dados que, além de armazenar e disponibilizar informações, como o sistema deste trabalho, também pudesse interagir com estas com o propósito de tomar decisões, justificaria a continuidade deste trabalho.

Assim, conclui-se inequivocamente a viabilidade deste trabalho, quer pelo cunho científico deste, quer pela eficiência com que demandas atuais são atendidas.

Sua utilidade e necessidade são também respectivamente observadas por permitirem que novas pesquisas ligadas ao reator IEA-R1 sejam desenvolvidas em tempos mais reduzidos, com maior confiabilidade e também por manterem os usuários destas informações atualizados em relação a suas tendências.

APÊNDICE A – ARQUIVOS RESPONSÁVEIS PELA GERAÇÃO DAS TABELAS, RESTRIÇÕES, PARÂMETROS DE LOGON E RELACIONAMENTOS DO BANCO DE DADOS FALCAO

Conexão ao banco de dados Oracle

Utilizando o programa SQL-Plus, nativo do banco de dados *Oracle* e necessariamente o usuário *system*, administrador do banco de dados, conecta-se no banco de dados para execução dos comandos seguintes.

```
conn system@falcao/<senha>
```

Criação do usuário proprietário das tabelas do banco de dados FALCAO

```
create user cen identified by <senha>
default tablespace sacd_dados
temporary tablespace temp;
grant connect,resource,create session to cen;
```

Criação das tabelas temporárias

Tabela de vazões:

```
create table cen.tab_temporaria_vazao
(
geracao timestamp,
F1M3 number(6,2),F23 number(6,2),F2M3 number(6,2),
F3M3 number(6,2),C1 number(6,2),C2 number(6,2),
L1 number(6,2),F1 number(6,2),F2 number(6,2),
F3 number(6,2),F4 number(6,2),DP number(6,2),
PR number(6,2),PRPA number(6,2),PRPB number(6,2),
PRSA number(6,2),PRSB number(6,2),NBPA number(6,2),
NBPB number(6,2)
);
```

Tabela de temperaturas:

```
create table cen.tab_temporaria_temperatura
(
Geracao timestamp,
T1 number(6,2),T2 number(6,2),T3 number(6,2),
T4 number(6,2),DT1 number(6,2),T6 number(6,2),
T7 number(6,2),T8 number(6,2),T9 number(6,2),
DT2 number(6,2),T10 number(6,2),T11 number(6,2),
T12 number(6,2),DT3 number(6,2),T13 number(6,2),
T14 number(6,2),T15 number(6,2),T16 number(6,2),
DT4 number(6,2),T17 number(6,2),T18 number(6,2),
DT5 number(6,2),T19 number(6,2),T20 number(6,2),
DELTA number(6,2),T21 number(6,2),DELTAB number(6,2),
T22 number(6,2),T23 number(6,2),T24 number(6,2)
);
```

Tabela de radiações:

```
create table cen.tab_temporaria_radiacao
(
geracao timestamp,
MA1 number(4,2), MA2 number(4,2), MA3 number(4,2),
MA4 number(4,2), MA5 number(4,2), MA6 number(4,2),
MA7 number(4,2), MA8 number(4,2), MA9 number(4,2),
MD1 number(5), MD2 number(5), MD3 number(5),
MD4 number(5), RES number(5), MD6 number(8,6)
);
```

Tabela de grandezas nucleares:

```
create table cen.tab_temporaria_nucleares
(
geração timestamp,
N1 number(6,2), N2 number(6,2), N3 number(6,2),
N4 number(6,2), N5 number(6,2), N6 number(6,2),
N7 number(6,2), N8 number(6,2), Z1 number(6,2),
Z2 number(6,2), Z3 number(6,2), Z4 number(6,2),
PERIOD3 number(6,2)
);
```

Criação das tabelas efetivas

Tabela de parâmetros:

```
Create table cen.tab_geral_001T
(
pkni_001T_id_comp number (3),
atsv_001T_nome_componente varchar2(7),
atni_001T_nome_grp_comp number(2),
atsv_001T_desc_comp varchar2(180)
);
```

```
alter table cen.tab_geral_001T add constraint tab_geral_l_pk11 primary key
(pkni_001T_id_comp) using index storage (initial 131072 next 131072
minextents 1 maxextents 4096 ) tablespace sacd_index;
```

Tabela de vazões:

```
create table cen.tab_vazao_005T
(
pkni_005T_id number (35),
pkdt_005T_tempo timestamp,
atsv_005T_geracao timestamp,
atni_005T_id_comp number (3) not null,
atnd_005T_valor_vazao number (6,2)
)
;
alter table cen.tab_vazao_005T add constraint tab_vazao_l_pk51 primary key
(pkni_005T_id,pkdt_005T_tempo) using index storage (initial 131072 next
131072 minextents 1 maxextents 4096) tablespace sacd_index;
```

Tabela de temperaturas:

```
create table cen.tab_temperatura_004T
(
pkni_004T_id          number (35),
pkdt_004T_tempo      timestamp,
atsv_004T_geracao    timestamp,
atni_004T_id_comp    number (3) not null,
atnd_004T_valor_temperatura number (6,2)
);

alter table cen.tab_temperatura_004T
add constraint tab_temperatura_l_pk41 primary key
(pkni_004T_id,pkdt_004T_tempo) using index storage (initial 131072 next
131072 minextents 1 maxextents 4096 ) tablespace sacd_index;
```

Tabela de radiações:

```
create table cen.tab_radiacao_003T
(
pkni_003T_id          number (35) ,
pkdt_003T_tempo      timestamp ,
atsv_003T_geracao    timestamp,
atni_003T_id_comp    number (3) not null,
atnd_003T_valor_radiacao number (11,6)
)
;

alter table cen.tab_radiacao_003T add constraint tab_radiacao_l_pk31 primary
key (pkni_003T_id,pkdt_003T_tempo) using index storage (initial 131072 next
131072 minextents 1 maxextents 4096 ) tablespace SACD_INDEX;
```

Tabela de grandezas nucleares:

```
create table cen.tab_nuclear_002T
(
pkni_002T_id          number (35),
pkdt_002T_tempo      timestamp,
atsv_002T_geracao    timestamp,
atni_002T_id_comp    number (3) not null,
atnd_002T_valor_nuclear number (6,2)
);

alter table cen.tab_nuclear_002T add constraint tab_nuclear_l_pk21 primary key
(pkni_002T_id,pkdt_002T_tempo) using index storage ( initial 131072 next
131072 minextents 1 maxextents 4096) tablespace sacd_index;
```

Tabela de usuários:

```
create table cen.tab_usuario
(
pkni_id                number(10) not null,
atsv_login             varchar2(24),
atsv_senha            varchar2(200),
atni_admin             number(1)
atsv_nome              varchar2(100)
);
```

```
alter table cen.tab_usuario add constraint pk_tab_usuario primary key (pkni_id)
using index storage ( initial 131072 next 131072 minextents 1 maxextents 4096)
tablespace sacd_index;
```

```
create table cen.tab_polinomio
(
pkdt_dt_arranjo       date,
pkni_num_arranjo      number(5),
pkni_tipo_barra       number(2),
atsv_polinomio_barra varchar2(80),
atni_r_quadrado       number(9,8)
);
```

```
alter table cen.tab_polinomio add constraint pk_polinomio primary key
(pkdt_dt_arranjo,pkni_num_arranjo,pkni_tipo_barra) using index storage
( initial 131072 next 131072 minextents 1 maxextents 4096)
tablespace sacd_index;
```

```
alter table cen.tab_polinomio add constraint fk_nuclear foreign
key(pkni_tipo_barra) references cen.tab_geral_001t(pkni_001t_id_comp);
```

```
create table cen.tab_curva_s
(
fkdt_dt_arranjo       date,
pkni_num_arranjo      number(5),
pkni_tipo_barra       number(2),
Atnd_posicao_barra     number(8,2),
Atnd_reat_barra       number(8,2)
);
```

```
alter table cen.tab_curva_s add constraint pk_curva_s primary key
(fkdt_dt_arranjo,pkni_tipo_barra,pkni_num_arranjo) using index storage
(initial 131072 next 131072 minextents 1 maxextents 4096)
tablespace sacd_index;
```

```
alter table cen.tab_curva_s add constraint fk_cur_pol foreign key
(fkdt_dt_arranjo,pkni_num_arranjo,pkni_tipo_barra)
references
cen.tab_polinomio (pkdt_dt_arranjo,pkni_num_arranjo,pkni_tipo_barra);
```

```
alter table cen.tab_curva_s add constraint ck_tp_barra check
( pkni_tipo_barra in ('9','10','11','12'));
```

Criação das seqüências:

```
create sequence cen.tab_vazao_seq nomaxvalue nocycle nocache;  
create sequence cen.tab_temperatura_seq nomaxvalue nocycle nocache;  
create sequence cen.tab_radiacao_seq nomaxvalue nocycle nocache;  
create sequence cen.tab_nuclear_seq nomaxvalue nocycle nocache;  
create sequence cen.tab_usuario_seq nomaxvalue nocycle nocache;
```

Inserção dos parâmetros de logon:

```
insert into cen.tab_usuario  
values (1,'SALIM','81dc9bdb52d04dc20036dbd8313ed055',1);  
commit;
```

Relacionamentos

```
alter table cen.tab_nuclear_002T add constraint fk_11  
foreign key(atni_002T_id_comp)  
references cen.tab_geral_001T(pkni_001T_id_comp)  
;  
alter table cen.tab_radiacao_003T add constraint fk_21  
foreign key (atni_003T_id_comp)  
references cen.tab_geral_001T(pkni_001T_id_comp)  
;  
alter table cen.tab_temperatura_004T add constraint fk_31  
foreign key (atni_004T_id_comp)  
references cen.tab_geral_001T(pkni_001T_id_comp)  
;  
alter table cen.tab_vazao_005T add constraint fk_41  
foreign key (atni_005T_id_comp)  
references cen.tab_geral_001T(pkni_001T_id_comp)  
;  
;
```

APÊNDICE B - FONTES DOS PROGRAMAS QUE CONSISTEM, TRANSMITEM, CARREGAM E DISPONIBILIZAM OS DADOS ORIUNDOS DO REATOR IEA-R1 PARA O BANCO DE DADOS FALCAO, NO CEN

1 - Programa de consistência e transferência dos arquivos

Rotina Transfere Dados IEA-R1

```

SET DD=%DATE:~7,2%
SET MM=%DATE:~4,2%
SET YYYY=%DATE:~10,4%
SET HOURS=%TIME:~0,2%
SET MINS=%TIME:~3,2%
SET SECS=%TIME:~6,2%
SET LOGNAME1=%MM%%DD%%YYYY%
SET LOGNAME1=%LOGNAME1:=%
SET LOGNAME2=%HOURS%%MINS%%SECS%
SET LOGNAME2=%LOGNAME2:=%
echo " *** NOVA TRANSMISSAO DE ARQUIVOS INDIVIDUALIZADOS DO REATOR ***"
    >> B:\SAD2006\Logs_SAD\%LOGNAME1%_2_File-Log-Transfer_%LOGNAME2%.Log
echo " *** NA AUSENCIA DE UM DOS 4 ARQUIVOS, OS ARQUIVOS EXISTENTES SERAO TRANSFERIDOS ***"
    >> B:\SAD2006\Logs_SAD\%LOGNAME1%_2_File-Log-Transfer_%LOGNAME2%.Log
echo " *** ESTA BAT ESTA PROGRAMADA PARA SER STARTADA DA ESTACAO DELVONEI-NET1 E EXECUTADA
NA ESTACAO WRICCI-NET***"
    >> B:\SAD2006\Logs_SAD\%LOGNAME1%_2_File-Log-Transfer_%LOGNAME2%.Log
echo " *** CASO SEJA STARTADA DA ESTACAO WRICCI-NET DIRETAMENTE, SE FAZ NECESSARIO TROCAR B:
POR E: E M: POR C: ***"
    >> B:\SAD2006\Logs_SAD\%LOGNAME1%_2_File-Log-Transfer_%LOGNAME2%.Log
date /t >> B:\SAD2006\Logs_SAD\%LOGNAME1%_2_File-Log-Transfer_%LOGNAME2%.Log
time /t >> B:\SAD2006\Logs_SAD\%LOGNAME1%_2_File-Log-Transfer_%LOGNAME2%.Log
if not exist B:\SAD2006\var*.txt goto:error1
:proximo1
if not exist B:\SAD2006\rad*.txt goto:error2
:proximo2
if not exist B:\SAD2006\item*.txt goto:error3
:proximo3
if not exist B:\SAD2006\nuc*.txt goto:error4
:proximo4
echo !copy B:\SAD2006\var*.txt M:\          >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo !copy B:\SAD2006\rad*.txt M:\        >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo !copy B:\SAD2006\nuc*.txt M:\        >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo !ren M:\var*.txt vars.dat            >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo !ren M:\rad*.txt radi.dat            >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo !ren M:\item*.txt temp.dat           >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo !ren M:\nuc*.txt nucl.dat            >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo open 10.0.8.124                      >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo user oracle ipen2006                 >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo bin                                   >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo put M:\vars.dat                      >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo put M:\radi.dat                     >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo put M:\temp.dat                     >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo put M:\nucl.dat                     >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo put B:\SAD2006\Logs_SAD\%LOGNAME1%_2_File-Log-Transfer_%LOGNAME2%.Log >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo !ren M:\vars.dat %LOGNAME1%_vars-Transmitido_%LOGNAME2%.txt
    >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo !ren M:\radi.dat %LOGNAME1%_radi-Transmitido_%LOGNAME2%.txt
    >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo !ren M:\temp.dat %LOGNAME1%_temp-Transmitido_%LOGNAME2%.txt
    >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo !ren M:\nucl.dat %LOGNAME1%_nucl-Transmitido_%LOGNAME2%.txt
    >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo !copy B:\SAD2006\Logs_SAD\%LOGNAME1%_2_File-Log-Transfer_%LOGNAME2%.Log M:\Transferidos
    >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo !copy B:\SAD2006\Logs_SAD\%LOGNAME1%_2_File-Log-Transfer_%LOGNAME2%.Log M:\Originais
    >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo !del M:\%LOGNAME1%_vars-Transmitido_%LOGNAME2%.txt
    >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo !del M:\%LOGNAME1%_radi-Transmitido_%LOGNAME2%.txt
    >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo !del M:\%LOGNAME1%_temp-Transmitido_%LOGNAME2%.txt
    >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo !del M:\%LOGNAME1%_nucl-Transmitido_%LOGNAME2%.txt
    >>B:\SAD2006\Logs_SAD\File_Parameter.dat

```

```
echo !del B:\SAD2006\var*.txt >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo !del B:\SAD2006\rad*.tx >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo !del B:\SAD2006\tem*.txt >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo !del B:\SAD2006\nuc*.txt >>B:\SAD2006\Logs_SAD\File_Parameter.dat
echo bye >>B:\SAD2006\Logs_SAD\File_Parameter.dat
ftp -n -s:B:\SAD2006\Logs_SAD\File_Parameter.dat
>> B:\SAD2006\Logs_SAD\%LOGNAME1%\_2_File-Log-Transfer_%LOGNAME2%.Log
goto :end
:error1
echo "Arquivo var*.txt ainda indisponível"
>> B:\SAD2006\Logs_SAD\%LOGNAME1%\_2_File-Log-Transfer_%LOGNAME2%.Log
goto proximo1
:error2
echo "Arquivo rad*.txt ainda indisponível"
>> B:\SAD2006\Logs_SAD\%LOGNAME1%\_2_File-Log-Transfer_%LOGNAME2%.Log
goto proximo2
:error3
echo "Arquivo temp*.txt ainda indisponível"
>> B:\SAD2006\Logs_SAD\%LOGNAME1%\_2_File-Log-Transfer_%LOGNAME2%.Log
goto proximo3
:error4
echo "Arquivo nucl*.txt ainda indisponível"
>> B:\SAD2006\Logs_SAD\%LOGNAME1%\_2_File-Log-Transfer_%LOGNAME2%.Log
goto proximo4
:end exit
```

2 - Programas de carga de dados

Os quatro grupos de arquivos, ou seja, arquivos de temperaturas, grandezas nucleares, radiações e vazões, a menos do arranjo físico, que inclui o total de colunas envolvidas e é peculiar de cada arquivo, possuem os processos de carga de dados idênticos. Abaixo, em detalhes, o processo de carga do arquivo de temperaturas

Rotina Carrega Dados IEA-R1

```
ORACLE_SID=ipen export ORACLE_SID
ORACLE_HOME=/home/oracle/oracle/product/10.2.0/db_1 export ORACLE_HOME
PATH=$PATH:$ORACLE_HOME/bin:/home/oracle/oracle/product/10.2.0/db_1/bin/ export PATH
ORACLE_OWNER=oracle export ORACLE_OWNER
NLS_LANG=AMERICAN_BRAZIL.WE8ISO8859P1 export NLS_LANG
LANGUAGE=AMERICAN_BRAZIL.WE8ISO8859P1 export LANGUAGE
ORACLE_BASE=/home/oracle export ORACLE_BASE
LD_LIBRARY_PATH=/home/oracle/oracle/product/10.2.0/db_1/bin export LD_LIBRARY_PATH
sqlldr carga/carga control=controle_carga_reator_temperatura.ctl
load Data
infile 'F:\SAD\SAD2007\Temperatura\temp.dat'
badfile 'F:\SAD\SAD2007\Temperatura\temp_bad.bad'
discardfile 'F:\SAD\SAD2007\Temperatura\temp_discard.dsc'
discardmax 999
append
into table cen.tab_temporaria_temperatura
(
geracao          POSITION(01:14) char,
t1               POSITION(24:27) decimal external,
t2               POSITION(29:32) decimal external,
t3               POSITION(34:37) decimal external,
t4               POSITION(39:42) decimal external,
dt1              POSITION(44:47) decimal external,
t6               POSITION(49:52) decimal external,
t7               POSITION(54:57) decimal external,
t8               POSITION(59:62) decimal external,
t9               POSITION(64:67) decimal external,
dt2              POSITION(72:76) decimal external,
t10              POSITION(78:81) decimal external,
t11              POSITION(83:86) decimal external,
t12              POSITION(88:91) decimal external,
dt3              POSITION(96:100) decimal external,
t13              POSITION(102:105) decimal external,
t14              POSITION(107:110) decimal external,
t15              POSITION(112:115) decimal external,
t16              POSITION(117:120) decimal external,
dt4              POSITION(125:129) decimal external,
t17              POSITION(131:134) decimal external,
t18              POSITION(136:139) decimal external,
dt5              POSITION(144:148) decimal external,
t19              POSITION(150:153) decimal external,
t20              POSITION(155:158) decimal external,
deltaa           POSITION(163:167) decimal external,
t21              POSITION(169:172) decimal external,
deltab           POSITION(177:181) decimal external,
t22              POSITION(183:186) decimal external,
t23              POSITION(189:191) decimal external,
t24              POSITION(194:196) decimal external
)insert into cen.tab_temperatura_004T
(select tab_temperatura_seq.NEXTVAL,sysdate,geracao,'29','t1 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq.NEXTVAL,sysdate,geracao,'30','t2 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq.NEXTVAL,sysdate,geracao,'31','t3 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq.NEXTVAL,sysdate,geracao,'32','t4 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq.NEXTVAL,sysdate,geracao,'33','dt1 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
```

```

(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'34',t6 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'35',t7 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'36',t8 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'37',t9 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'38',dt2 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'39',t10 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'40',t11 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'41',t12 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'42',dt3 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'43',t13 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'44',t14 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'45',t15 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'46',t16 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'47',dt4 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'48',t17 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'49',t18 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'50',dt5 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'51',t19 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'52',t20 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'53',deltaa from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'54',t21 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'55',deltab from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'56',t22 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'57',t23 from cen.tab_temporaria_temperatura);
commit;
insert into cen.tab_temperatura_004T
(select tab_temperatura_seq,NEXTVAL,sysdate,geracao,'58',t24 from cen.tab_temporaria_temperatura);
commit;
delete from cen.tab_temporaria_temperatura where t1 <> '1234123'; commit;exit exit

```

3 – Programa fonte¹ da aplicação SACD

main.jsp

```
<%@ page language="Java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="f" uri="http://Java.sun.com/jsf/core"%>
<%@ taglib prefix="h" uri="http://Java.sun.com/jsf/html"%>
<%@ taglib prefix="t" uri="http://myfaces.apache.org/tomahawk"%>
<f:view> <f:loadBundle basename="br.sacd.resources.ApplicationMessages" var="msg" />
  <jsp:include page="/includes/header.jsp" />
  <jsp:include page="/includes/menu.jsp" />
  <t:div id="panel">
    <h:form id="mainForm">
      <t:panelGrid id="p2" styleClass="caixa">
        <t:panelGrid id="p3" columns="1">
          <t:panelGrid id="p4" columns="2">
            <t:outputLabel for="grafico"
value="#{msg['label.grafico']}" />
            <t:selectOneMenu id="grafico"
value="#{mainBean.tipoGrafico}" styleClass="big">
              <t:selectItems var="grafico"
itemLabel="#{grafico.nome}" itemValue="#{grafico.id}"
value="#{mainBean.tiposGraficos}"
              />
            </t:selectOneMenu>
            <t:outputLabel for="inicio"
value="#{msg['label.data.inicio']}" />
            <t:inputCalendar id="inicio" required="true"
value="#{mainBean.inicio}" renderAsPopup="true"
              renderPopupButtonAsImage="true" styleClass="small"
            />
            <t:outputLabel for="fim"
value="#{msg['label.data.fim']}" />
            <t:inputCalendar id="fim" required="true"
value="#{mainBean.fim}" renderAsPopup="true"
              renderPopupButtonAsImage="true" styleClass="small"
            />
          </t:panelGrid>
          <t:message id="msgInicio" for="inicio" />
          <t:message id="msgFim" for="fim" />
        </t:panelGrid>
        <h:commandButton id="submit" value="#{msg['label.botao.ok']}"
action="#{mainBean.submit}" />
      </t:panelGrid>
    </h:form>
  </t:div>
</jsp:include page="/includes/footer.jsp" /></f:view>
```

Login.jsp

```
<%@ page language="Java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="f" uri="http://Java.sun.com/jsf/core"%>
<%@ taglib prefix="h" uri="http://Java.sun.com/jsf/html"%>
<%@ taglib prefix="t" uri="http://myfaces.apache.org/tomahawk"%>
<f:view>
  <f:loadBundle basename="br.sacd.resources.ApplicationMessages" var="msg" />
  <jsp:include page="/includes/header.jsp" />
  <t:div id="login">
    <h:form>
      <t:panelGrid id="caixa" styleClass="caixa">
        <h:messages errorClass="error" />
        <t:panelGrid id="caixaLayout" columns="2">
          <h:outputLabel for="usuario" value="Usuário" />
          <h:inputText id="usuario" value="#{loginBean.usuario}"
styleClass="medium"/>
          <h:outputLabel for="senha" value="Senha" />
          <h:inputSecret id="senha" value="#{loginBean.senha}"
styleClass="medium"/>
        </t:panelGrid>
        <h:commandButton id="submit" value="OK"
action="#{loginBean.submit}" />
      </t:panelGrid>
    </h:form>
  </t:div>
  <jsp:include page="/includes/footer.jsp" />
</f:view>
```

¹ O código completo do sistema SACD e os demais códigos pertinentes à este trabalho, encontram-se no CD anexo.

Graficos.jsp

```

<%@ page language="Java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-
8859-1"%>
<%@ taglib prefix="f" uri="http://Java.sun.com/jsf/core"%>
<%@ taglib prefix="h" uri="http://Java.sun.com/jsf/html"%>
<%@ taglib prefix="t" uri="http://myfaces.apache.org/tomahawk"%>
<%@ taglib uri="https://ajax4jsf.dev.java.net/ajax" prefix="a4j"%>
<f:view>
  <f:loadBundle basename="br.sacd.resources.ApplicationMessages" var="msg" />
  <jsp:include page="/includes/header.jsp" />
  <jsp:include page="/includes/menu.jsp" />
  <t:div id="panel">
    <t:panelGrid id="p1" columns="1">
      <t:graphicImage value="/grafico?rand=#{mainBean.random}" />
      <h:form id="periodoForm">
        <h:dataTable id="periodList" var="period"
value="#{mainBean.grafico.otherPeriods}">
          <h:outputText value="Periodo" />
          <h:outputText value="#{period}" />
        </h:dataTable>
        <t:panelGrid id="p2" columns="3"
rendered="#{mainBean.multiplosPeriodos}">
          <t:outputLabel for="periodo" value="Adicionar novo periodo:"
/>
          <t:inputCalendar id="periodo" value="#{mainBean.periodo}"
renderAsPopup="true"
          renderPopupButtonAsImage="true" styleClass="small"
          />
          <h:commandButton id="submit1"
value="#{msg['label.botao.ok']}"
          action="#{mainBean.novoPeriodo}"
          />
        </t:panelGrid>
      </h:form>
      <h:form id="relForm">
        <t:panelGrid id="p3" columns="3">
          <t:outputLabel for="grafico"
value="#{msg['label.grafico.relacionado']}" />
          <t:selectOneMenu id="grafico" value="#{mainBean.tipoGrafico}"
styleClass="big">
            <t:selectItems var="grafico" itemLabel="#{grafico.nome}"
itemValue="#{grafico.id}"
            value="#{mainBean.tiposGraficosRelacionados}"
            />
          </t:selectOneMenu>
          <h:commandButton id="submit" value="#{msg['label.botao.ok']}"
action="#{mainBean.submit}" />
        </t:panelGrid>
      </h:form>
      <h:form id="NovoForm">
        <t:panelGrid id="p3" columns="2">
          <t:outputLabel for="submit2" value="Novo Gráfico" />
          <h:commandButton id="submit2"
value="#{msg['label.botao.ok']}" action="#{mainBean.novoGrafico}" />
        </t:panelGrid>
      </h:form>
    </t:panelGrid>
  </t:div>
  <jsp:include page="/includes/footer.jsp" />
</f:view>

```

Erros.jsp

```

<%@ taglib prefix="f" uri="http://Java.sun.com/jsf/core"%>
<%@ taglib prefix="h" uri="http://Java.sun.com/jsf/html"%>
<%@ taglib prefix="t" uri="http://myfaces.apache.org/tomahawk"%>
<f:subview id="error">
    <f:loadBundle basename="br.sacd.resources.ApplicationMessages" var="msg" />
    <t:document>
        <t:documentHead>
            <t:htmlTag value="title">
                <h:outputText value="#{msg['titulo.erro']}" />
            </t:htmlTag>
            <t:stylesheet path="css/cen.css" />
        </t:documentHead>
        <t:documentBody>
            <t:div id="main">
                <t:div id="topo">
                    <h:graphicImage value="imagens/logo_ipen.jpg" style="float:
left" />
                    <f:verbatim>
                        <span class="titulo">SACD</span>
                    </f:verbatim>
                </t:div>
                <t:htmlTag value="p"><h:outputText
value="#{msg['mensagem.erro']}" /></t:htmlTag>
                <h:form>
                    <h:inputTextarea style="width: 99%;" rows="10" readonly="true"
value="#{errorBean.stackTrace}" />
                </h:form>
                </t:div>
            </t:documentBody>
        </t:document>
    </f:subview>

```

ANEXO A – ASPECTOS BÁSICOS DA LINGUAGEM SQL APLICADOS A ESTE TRABALHO

Estrutura da linguagem sql

Fundamentada na teoria dos conjuntos, é através dos comandos da linguagem Structure Query Language, SQL, que interagimos com os bancos de dados relacionais. A linguagem SQL, normalizada pela norma American National Standards Institute, ANSI, é subdividida em quatro grupos, a saber:

DML – Data Manipulation Language (insert, update e delete).

DDL – Data Definition Language (alter, drop, create, commit, rollback).

DCL – Data Control Language (grant, revoke).

DRL – Data Retrive Language (select).

Estrutura sintática dos comandos de manipulação de dados

Insert into [tabela]

Values [valor1, valor2,...];

Update [tabela]

Set [coluna = valor,...]

Where [condição];

Delete from [tabela]

Where [condição];

Select [coluna,...]

From [tabela]

Where [condição]

Group by [coluna,...]

Having [condição]

Order by [coluna,...];

Cláusula group by:

Pode ser usada para dividir as linhas dentro da tabela em pequenos grupos, podendo ser também utilizada para retornar informações sumarizadas por cada grupo formado.

Estrutura do comando com a cláusula group by:

```
Select coluna1,função_grupo,coluna2
From tabela
[where      condição]
[group by   expressões do group by]
[order by   coluna(s)];
```

Expressões com group by especificam as colunas cujos valores determinarão a base para o agrupamento das linhas. As colunas que não utilizarem funções de grupo deverão ser declaradas na cláusula group by.

Exemplo:

```
Select cargo,max(salario)
From cen.pagamento
Group by cargo;
```

Cláusula having

Pode-se usar a cláusula having para especificar quais os grupos serão mostrados. O uso da cláusula having requer obrigatoriamente o uso anterior da cláusula group by, ou seja, a cláusula having serve de filtro ao conteúdo agrupado.

Estrutura do Comando com a cláusula having

```
Select coluna,função_grupo,coluna
From tabela
[Where      condição]
[Group by   expressões do group by]
[Having     condições de grupo]
[Order by   coluna(s)];
```

ANEXO B – PROCEDIMENTOS PARA INSTALAÇÃO E CUSTOMIZAÇÃO DOS PRODUTOS *ORACLE 10G*, *JAVA 5.0* e *TOMCAT 5.5*

SGBD *ORACLE 10G*

Através do usuário administrador do sistema operacional, *root*, cuja utilização é indicada pelo símbolo #, executa-se:

```
# gunzip ship.db.cpio.gz
# cd/usr/sbin
# groupadd dba
# useradd -g oinstall -G dba oracle
```

Com o sistema operacional Linux Fedora 4 instalado, a instalação do SGBD *Oracle 10G* é iniciada utilizando o usuário *oracle*, grupo *dba*, editando-se o arquivo */home/oracle/.bash_profile* e anexando a ele as informações abaixo. A utilização do usuário *oracle* é indicada pelo símbolo \$.

```
$umask 022
$PATH=/bin: /usr /bin: /usr /local/bin; /usr /X11R6/bin
$LD_LIBRARY_PATH=/usr/lib: /usr/x11R6/lib
$ORACLEBASE=/u01/app/oracle
$ORACLEHOME=$ORACLE_BASE/product/10 .1. 0/db_1
$ORACLESID=exemplo
$LD_LIBRARY_PATH=$ORACLEHOME/jdk/fre/lib/i386: $ORACLE_HOME/jdk/j
$re/lib/i 386/server: $ORACLEHOME/rdbms/lib: $ORACLEHOME/lib:
$$LD_LIBRARY_PATH
$PATH=$ORACLEHOME/bin: $PATH
$export PATH LD_LIBRARY_PATH
$export ORACLEBASE ORACLEHOME ORACLE_SID
```

Agora, cria-se a estrutura de diretórios e suas respectivas permissões de acesso:

```
# mkdir -p /u01/app/oracle
# chown -R oracle:oinstall/u01/app
# chmod -R 775 /u01/app
```

Editamos o arquivo `/etc/sysctl.conf` adicionando as seguintes linhas:

```
kernel.sem = 250 32000 100 128
kernel.shmall = 2097152
kernel.shmmax = 2147483648
kernel.shmmni = 4096
fs.file-max = 65536
neto.ipv4.ip_local_porchrange = 1024 65000
```

Executa-se o comando a seguir, para ajustar os parâmetros do kernel:

```
# sysctl -p
```

Conectado como usuário `root`, executa-se o comando seguinte para ativar a interface gráfica:

```
# startx
```

Com isto providenciado, passa-se a instalação do banco propriamente dita.

```
# su - oracle
```

```
$ cd /Install/Disk/ $ ./runInstaller
```

Surgindo a tela inicial do *Universal Installer*, FIG. 31, basta seguir a seqüência de telas e, ao final, ter-se-á o *software* instalado e uma base de dados de exemplo criada, cuja denominação, escolhida ao longo do processo de instalação, independe do nome do servidor que a abriga.



FIGURA 31 - Tela do instalador do SGBD Oracle 10G

Conectando ao banco de dados via sistema operacional com o usuário sysdba:

```
sqlplus /no log
```

```
SQL*plus: Release 10.1.0.2.0 - production on Suo Mar 7 15:10:23 2004
copyright (c) 1982, 2004, Oracle. All rights reserved
```

```
SQL>conn / as sysdba
connected to instance
```

```
Ativando o banco de dados
SQL>startup
```

Neste ponto, o banco de dados *ORACLE 10G* está instalado e pronto para ser utilizado, apenas necessitando de alguns ajustes para que conexões via estações remotas sejam possíveis.

Para isso, ativa-se o *listener*, como segue:

```
$ lsnrctl stop
$ lsnrctl start
$ lsnrctl status
```

A FIG. 32 confirma a ativação:

```
Prompt de comando (2)
Microsoft Windows XP [versão 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

F:\Documents and Settings\Falcao>lsnrctl status

LSNRCTL for 32-bit Windows: Version 10.1.0.2.0 - Production on 02-SET-2007 09:47:27

Copyright (c) 1991, 2004, Oracle. All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC)))
STATUS of the LISTENER
-----
Alias                     LISTENER
Version                   TNSLSNR for 32-bit Windows: Version 10.1.0.2.0 - Produ
ction
Start Date                02-SET-2007 09:46:40
Uptime                    0 days 0 hr. 0 min. 48 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   C:\oracle\product\10.1.0\Db_1\network\admin\listener.o
ra
Listener Log File         C:\oracle\product\10.1.0\Db_1\network\log\listener.log

Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(PIPENAME=\\.\pipe\EXTPROCipc)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=serverfalcao)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=serverfalcao)(PORT=8080))(Presentati
on=HTTP)(Session=RAW))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=serverfalcao)(PORT=2100))(Presentati
on=FTP)(Session=RAW))
Services Summary...
Service "PLSExtProc" has 1 instance(s).
  Instance "PLSExtProc", status UNKNOWN, has 1 handler(s) for this service...
Service "ipen" has 1 instance(s).
  Instance "ipen", status READY, has 1 handler(s) for this service...
Service "ipenXDB" has 1 instance(s).
  Instance "ipen", status READY, has 1 handler(s) for this service...
The command completed successfully

F:\Documents and Settings\Falcao>
```

FIGURA 32 - Estado de ativação do *listener*

Linguagem JAVA 5.0

Instalação do produto JAVA

Neste trabalho o produto *JAVA* versão 5.0 é utilizado.

O download do arquivo *JDK 1_5_0_11-windows-i586-p.exe* está disponível gratuitamente no endereço <http://Java.sun.com/Javase/downloads/index.jsp>.

A execução do arquivo *JDK 1_5_0_11-windows-i586-p.exe*, sob sistema operacional Windows XP Service Pack 2, resultará na tela representada pela FIG. 33.



FIGURA 33 - Tela inicial da instalação do JAVA 5.0

Feito isso, escolhe-se o diretório destino para a instalação do produto *JAVA* e aguarda-se o término da instalação, finalizando-a na opção *Finish*, como mostra a FIG. 34.

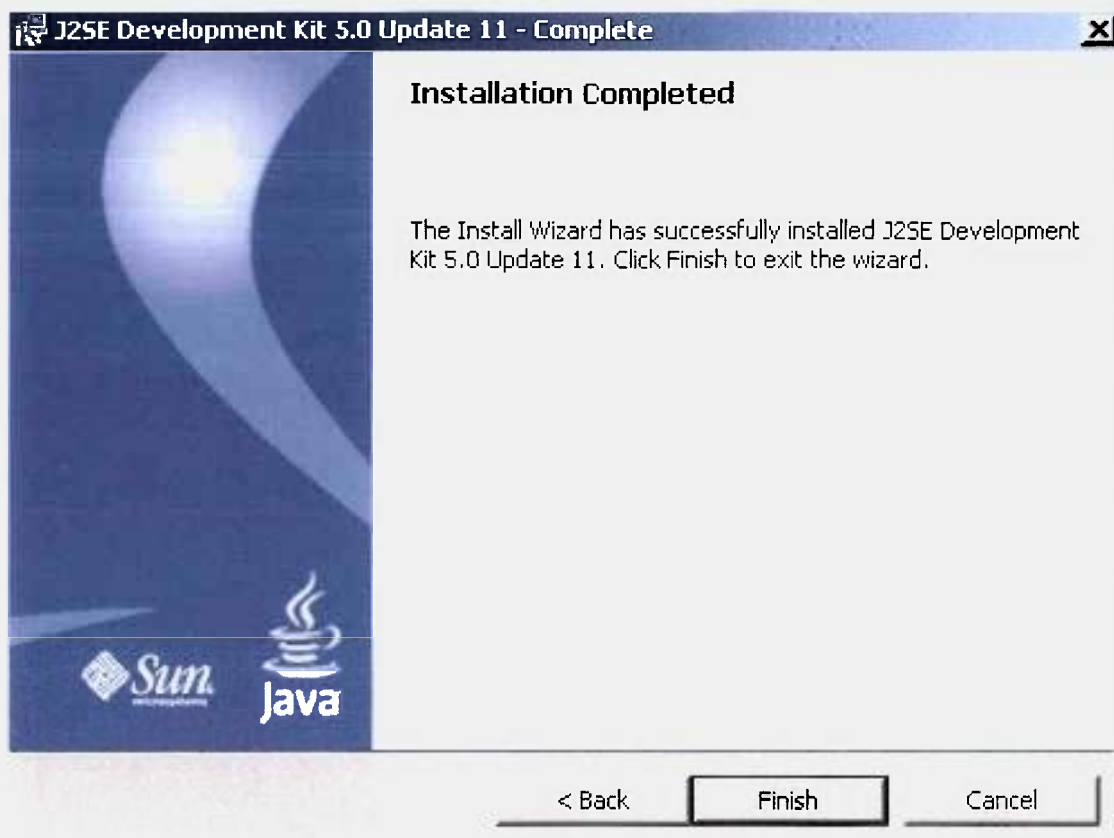


FIGURA 34 - Tela final da instalação do JAVA 5.0

Instalação do produto TOMCAT

Neste trabalho utiliza-se a versão 5.5.23 do servidor de aplicação *TOMCAT*. O download do arquivo `apache-tomcat-5.5.23.exe` está disponível gratuitamente no endereço <http://ftp.unicamp.br/pub/apache/tomcat/tomcat-5/v5.5.23/bin>.

A execução do arquivo `apache-tomcat-5.5.23.exe`, sob sistema operacional Windows XP Service Pack 2, resultará na tela representada pela FIG. 35.



FIGURA 35- Tela inicial da instalação do *TOMCAT 5.5.23*

Feito isso, após optar-se pelas configurações Service Startup, Start Menu Items e Webapps, opta-se pela porta 80, dá-se continuidade a instalação e finaliza-se, como mostra a FIG. 36.

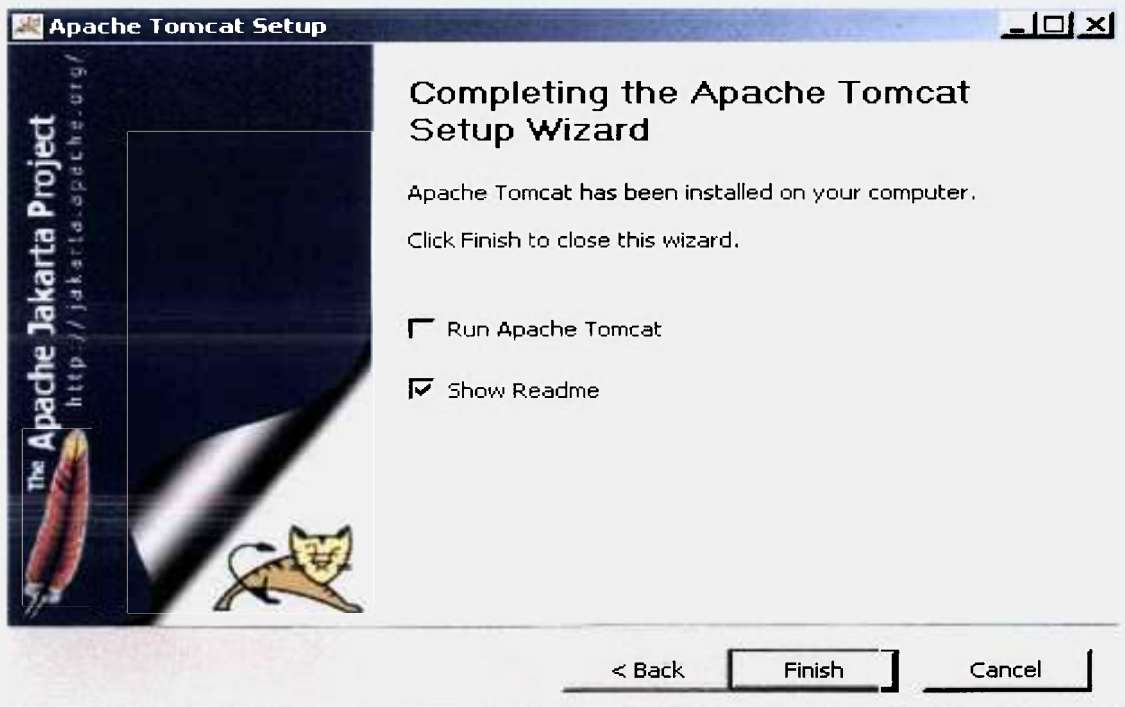


FIGURA 36 - Tela final da instalação do *TOMCAT 5.5.23*

REFERÊNCIAS BIBLIOGRÁFICAS

1. **CÂNDIDO, F., MVS/VSAM. São Paulo, S.P.: Manual técnico – T-Systems, 2007.**
2. **IBM Corp. and Fund. Software, IMS / ESA Performance Analyzer San Jose, CA.: Ed. IBM, 1998.**
3. **ALVES, W.P. Fundamentos de bancos de dados. n.1, p.28-36, São Paulo, S.P.: Ed. ÉRICA, 2004.**
4. **BARROS, C.T.M.A. Um banco de dados para fins de marketing: a experiência do CIN. Ci. Inf., Brasília, v.25, n.3, p.438-444, set./dez., 1996**
5. **COMPASS (Computer Aided Search System), User Guide – Bruweel International Company, Dinamarca, 1997.**
6. **RIBEIRO, A.C.O. Impacto da especialização de dados de falha em análises probabilísticas de segurança de plantas de processo. 2005. Dissertação (Mestrado) – Universidade Federal do Rio de Janeiro, Rio de Janeiro. Orientador: Paulo Fernando Ferreira Frutuoso e Melo.**
7. **CARVALHO, E.N. Uma revisão crítica do emprego de banco de dados de falhas em análises probabilísticas de segurança de plantas nucleares e químicas. 2007. Dissertação (Mestrado) – Universidade Federal do Rio de Janeiro, Rio de Janeiro. Orientador: Paulo Fernando Ferreira Frutuoso e Melo.**
8. **BENEVENUTTI, E.L. Metodologia para monitoração e diagnóstico de vibração das bombas moto-operadas do circuito primário de refrigeração do Reator IEA-R1. 2004. Dissertação (Mestrado) - Instituto de Pesquisas Energéticas e Nucleares – IPEN-CNEN-SP, São Paulo. 220 p. Orientador: Daniel Kao Sun Ting.**
9. **GONÇALVES, I.M.P. Monitoração e diagnóstico para detecção de falhas de sensores utilizando a metodologia GMDH. 2006. Tese (Doutoramento) - Instituto de Pesquisas Energéticas e Nucleares – IPEN-CNEN-SP, São Paulo. Orientador: Daniel Kao Sun Ting.**
10. **BOARATTI, M.F.G. Modelagem analítica da propagação de ondas de tensão em tubos de parede fina visando a localização de uma fonte pontual harmônica em sua superfície. 2006. Tese (Doutoramento) - Instituto de Pesquisas Energéticas e Nucleares – IPEN-CNEN-SP, São Paulo. Orientador: Daniel Kao Sun Ting.**
11. **Elipse HMI/SCADA solutions. Disponível em: <http://www.elipse.com.br/produto_apresentacao.aspx?id=2&idioma=1>. Acesso em 5 Mai. 2008.**
12. **Fluido dinâmica computacional e suas aplicações. Disponível em: <<http://br.monografias.com/trabalhos/fluidodinamica/fluidodinamica.shtml>>. Acesso em 5 Mai. 2008.**

13. **ANSYS, Code Manual, Release 5.4(1997), ANSYS Inc. Houston, USA.**
14. **FLUENT. Disponível em <<http://www.fluent.com>> Acesso em 29 Mai. 2008.**
15. **PHOENICS. Disponível em <<https://www.in2itive.biz/cham/login.aspx?ReturnUrl=%2fcham%2fDefault.aspx>> Acesso em 29 Mai. 2008.**
16. **CFX. Disponível em <<http://www.ansys.com/products/cfx.asp>> Acesso em 29 Mai. 2008.**
17. **Yu, Y., Lee, S. E Zhang, Z.L. (2004) Leopard: A locality-aware peer-to-peer system with no host spot, Tech. Report CSE Dept., U of Minnesota, 2004.**
18. **FOWLER, T.B., VONDY, D.R., CUNNINGHAM, G.W., Nuclear reactor Core Analysis Code: CITATION, ORNL-TM_2496, Rev.2, July 1971.**
19. **BRIESMEISTER, J.F. MCNP – A General Monte Carlo N- Particle Transport Code. RSICC Code package CCC-660, (1997).**
20. **FANDERUFF, D., Oracle 8i São Paulo, S.P.: Ed. Makron Books, 2000.**
21. **TEOREY, T., LIGHTSTONE,S., NADEAU,T., Projeto e modelagem de banco de dados. Rio de Janeiro, R.J.: Ed. CAMPUS, 2007.**
22. **CHEN, P. Modelagem de dados: A abordagem entidade. São Paulo, S.P.: Ed. Makron, 1990.**
23. **MECENAS, I., OLIVEIRA, V. Banco de dados – Do modelo conceitual à implementação física. n.3, p.36-40, Rio de Janeiro, R.J.: Ed. ALTA BOOKS, 2005.**
24. **MARTINS, V.W. Uso de formas nomais no estudo de dinâmica caótica em sistemas hamiltonianos. 1994. Tese (Doutoramento) - Instituto de Física Gleb Wataghin - UNICAMP- CAMPINAS, São Paulo. Orientador: Ozório de Almeida, Alfredo M.**
25. **SOARES, S.P.M., Dominando Erwin. São Paulo, S.P.: Ed. Ciência Moderna, 2003.**
26. **EMBARCADERO. Disponível em <<http://www.downloaddatabase.com/databasedevelopment/download-embarcadero-sql-debugger.htm>> Acesso em 5 Mai. 2008.**
27. **SERSON, R.R. Oracle 10G Database - Guia do DBA. São Paulo, S.P.: Ed. Novatec, p. 16, 2004.**
28. **DANESH, A. Dominando o Linux Red Hat. São Paulo, S.P.: Ed. Makron Books, 2000.**
29. **JANDL J., P. Introdução ao Java. São Paulo, S.P.: Ed. Berkeley, 2002.**
30. **MCDANIEL, J. Toad pocket reference for Oracle. New York, N.I. Ed. Oreilly & Assoc., 2002.**

31. **NETO, J.G.; ANDRADE, D.A.** “**FALCAO** – A relational database to storing the variables monitored in the research reactor IEA-R1” **Artigo publicado no International Atlantic Nuclear Conference – INAC, Santos – SP, 2007.**
32. **EZ Goal IT Netix.** Disponível em: <
<http://www.ezgoal.com/house/publisher.asp?qu=Quest+Software&Software>>. Acesso em 20 Mar. 2008.