
PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA AMBIENTAL
INSTITUTO DE FÍSICA
UNIVERSIDADE FEDERAL DE MATO GROSSO

**ARQUITETURA DE DADOS PARA
GERENCIAMENTO DE INFORMAÇÕES
MICROMETEOROLÓGICAS**

Allan Gonçalves de Oliveira

Prof. Dr Josiel Maimone de Figueiredo

Cuiabá (MT) - 2011

PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA AMBIENTAL
INSTITUTO DE FÍSICA
UNIVERSIDADE FEDERAL DE MATO GROSSO

**ARQUITETURA DE DADOS PARA
GERENCIAMENTO DE INFORMAÇÕES
MICROMETEOROLÓGICAS**

Allan Gonçalves de Oliveira

Dissertação apresentada ao Programa de Pós-Graduação em Física Ambiental da Universidade Federal de Mato Grosso, como parte dos requisitos para obtenção do título de Mestre em Física Ambiental.

Josiel Maimone de Figueiredo

Cuiabá, 12 de dezembro de 2011.

FICHA CATALOGRÁFICA

O48a Oliveira, Allan Gonçalves de.
Arquitetura de dados para gerenciamento de informações micrometeorológicas / Allan Gonçalves de Oliveira. – 2011.
xiii, 73 f. : il. color.

Orientador: Prof. Dr. Josiel Maimone de Figueiredo.
Dissertação (mestrado) – Universidade Federal de Mato Grosso, Instituto de Física, Pós-Graduação em Física Ambientel, 2011.

Bibliografia: f. 52-61.
Inclui apêndice.

1. Variáveis micrometeorológicas – Armazenamento de dados. 2. Variáveis micrometeorológicas – Dados – Controle de acesso. 3. Arquitetura de dados - Micrometeorologia. I. Título.

CDU – 551.5:004.62

Ficha elaborada por: Rosângela Aparecida Vicente Söhn – CRB-1/931

UNIVERSIDADE FEDERAL DE MATO GROSSO
INSTITUTO DE FÍSICA
Programa de Pós-Graduação em Física Ambiental

FOLHA DE APROVAÇÃO

**TÍTULO: ARQUITETURA DE DADOS PARA GERENCIAMENTO
DE INFORMAÇÕES MICROMETEOROLÓGICAS**

AUTOR: ALLAN GONÇALVES DE OLIVEIRA

Dissertação de Mestrado defendida e aprovada em 31 de outubro de 2011, pela comissão julgadora:

Prof. Dr. Josiel Maimone de Figueiredo
Orientador
Instituto de Computação – UFMT

Profa. Dra. Claudia Aparecida Martins
Examinadora Interna
Instituto de Computação – UFMT

Prof. Dr. Elmo Batista de Faria – Examinador Externo
Instituto de Computação – UFMT

Dedicatória

Dedico este trabalho primeiramente a Deus, como forma de agradecimento e louvor, primeiramente por tudo que Ele É, por tudo que me proporciona, dando perfeitas condições de seguir cada dia.

Dedico aos meus Pais e irmãos, como resultado de nossos projetos de vida construídos juntos, através das nossas experiências, sonhos e lutas. Dedico a eles, não por serem família, mas por serem a Minha família, a MELHOR família que Deus poderia me dar.

Dedico a minha noiva Bianca, que com carinho tem ajudado e me sustentado nos momentos difíceis e tem servido de inspiração para novos projetos.

Dedico ao Amigo Tião, que a muitos anos faz parte da família, apoiando em tudo que pode, incentivando e compartilhando de momentos felizes em nossas vidas.

Dedico também ao meu orientador professor Josiel Maimone de Figueiredo. A ele, dedico este trabalho como resultado do trabalho que ele plantou, a 5 anos e meio, quando me convidou para fazer parte do seu projeto de pesquisa e acreditou em meu potencial e ajudou a fortalecer os pontos fracos.

Agradecimentos

Agradeço primeiramente a Deus, por me conceder perfeitas condições de realizar este trabalho, proporcionado saúde física e mental e permitindo-me contar com a presença de pessoas importantes em minha vida as quais são citadas a baixo:

Aos meus pais Jéssé de Oliveira e Márcia Vitória Gonçalves de Oliveira, por serem pais maravilhosos, presentes, compreensivos e por todo apoio que me deram, pra realizar não só este, mas muitos outros projetos em minha vida.

Aos meus irmãos, Renan, Isis e o Marquinho, pelo apoio, carinho, confiança e por todas as experiências que vivemos juntos.

A minha noiva Bianca, por ter entrado em minha vida, passando a fazer parte dos meus projetos e sendo motivo de muitos outros. Pela compreensão, amizade e confiança.

Ao meu orientador Professor Josiel Maimone de Figueredo, por todos esses anos de orientação, desde a iniciação científica até o mestrado. Nesses anos além da orientação nos trabalhos acadêmicos foi também amigo.

A professorar Claudia Aparecida Martins por fazer parte desse projeto, como co-orientadora e por várias horas de ajuda com assuntos diversos.

Ao professor Nelcileo Vírgilio Araujo, pela amizade, pela força também na área profissional, principalmente quanto a UAB-IFMT.

Ao professor José de Souza Nogueira, que, como coordenador do PPGFA e junto à sua esposa professora Marta Cristina de Jesus Albuquerque Nogueira matém o programa em pleno funcionamento e buscando cada dia melhorar mais.

Ao Amigo Paraná (professor José Nogueira), que nesse período se mostrou realmente amigo, prestativo e que tem também participação na construção deste trabalho.

Aos colegas de mestrado, Jonathan, Leone, Thiago Rangel e Thiago Meirelles colegas de classe que se tornaram companheiros de pesquisa e amigos.

Aos colegas Thiago Meirelles e Raphael Souza pela amizade e pela ajuda com o desenvolvimento desse trabalho.

A todos os colegas do mestrado (2010), Adilson, André, Amanda, Edson (Vovô), Jonathan, Leone, Márcia, Thiago Meirelles e Thiago Rangel, pelo companheirismo nesses quase dois anos juntos.

Ao Cesário e Soilce pelos serviços administrativos prestados, que com dedicação ajudam todos os integrantes do programa naquilo que podem.

A todos os integrantes do PPGFA que, de alguma forma contribuíram com este trabalho, trocando informações, ajudando, mostrando os caminhos.

Ao meu amigo Luis, colega de graduação, que mesmo longe em outro programa de pós-graduação, quando preciso me ajuda no que for necessário.

A todos, muito obrigado!

Epígrafe

Um homem que nunca muda de
opinião, em vez de demonstrar a
qualidade da sua opinião demonstra a
pouca qualidade da sua mente.

Marcel Achard

OLIVEIRA, A. G. **Arquitetura de Dados Para Gerenciamento de Informações Científicas**. 2011. 86p. Dissertação (Mestrado em Física Ambiental). Instituto de Física, Universidade Federal de Mato Grosso, Cuiabá, 2011.

RESUMO

Pesquisas envolvendo dados científicos comumente se deparam com dificuldades quanto ao armazenamento adequado dos dados coletados. Essas pesquisas estão inseridas num contexto em que vários usuários especialistas manipulam os dados com o intuito de analisar comportamento, realizar previsões, gerar modelagens, entre outros. Nessas manipulações, diversos cálculos são realizados e os resultados dos mesmos servem como entrada para novos cálculos com novos resultados para novas entradas e assim sucessivamente. Além disso, esses especialistas estão também inseridos em projetos nos quais dados de um projeto devem ser manipulados apenas por especialistas vinculados ao mesmo. Nesse sentido, o objetivo deste trabalho foi fornecer uma arquitetura de dados para o gerenciamento de grande quantidade de dados científicos com aplicação principal sobre variáveis micrometeorológicas. A arquitetura suporta controle e armazenamento dos dados brutos e suas informações complementares, como: Sensor, tipo e local. São contempladas também as informações gerenciais, que servem principalmente para compartilhamento adequado dos dados. Essa arquitetura suporta ainda o armazenamento e execução automatizada de validações relacionadas a um tipo de dado. Os testes realizados para validação da arquitetura mostram que os objetivos foram alcançados e ainda suportam extensões de funcionalidades que possam vir a contribuir ainda mais com o contexto científico.

Palavras Chave: Banco de Dados, Dados Científicos, Dados Micrometeorológicos

OLIVEIRA, A. G. **Data Model for scientific information management**. 2011. 86p.
Dissertação (Mestrado em Física Ambiental). Instituto de Física, Universidade Federal de Mato Grosso, Cuiabá, 2011.

ABSTRACT

Researches related to scientific data often have difficulties with proper storage of the collected data. These researches are included in a context where various expert users manipulate the data in order to analyze behavior, make predictions, generate models, among others. All these processing include calculations in which the results are used as input for new calculations with new results to new entries and so on. Moreover, these scope of the researches are also included in projects in which data from one project should be handled only by specialists related to it. Thus, the aim of this work was to provide a data architecture for managing large amounts of scientific data with the main application in micrometeorological variables. The architecture supports storage and control of their raw data and information such as: sensor, type and location. They include also the management information, which mainly purpose is to control appropriate sharing of data between project members. This architecture also supports the storage and automated execution of validation related to a data type. Tests for validation of the architecture showed that the objectives were achieved and still support extensions of features that may further contribute to the scientific context.

Keywords: Database, Scientific Data, Meteorological Data

Sumário

Dedicatória	iv
Agradecimentos	v
Epígrafe	vii
1 Introdução	1
1.1 Problemática	1
1.2 Justificativa	5
1.3 Objetivo Geral	6
1.4 Objetivos Específicos	6
1.5 Contextualização do Trabalho	6
1.5.1 Gaiia	7
1.6 Organização do Trabalho	7
2 Fundamentação Teórica	9
2.1 Arquiteturas de tratamento de dados	10
2.2 Banco de Dados	12
2.2.1 Modelo Relacional	13
2.3 Dados Científicos	16
2.4 Arquitetura Corporativa de Software	19
2.4.1 Enterprise Resource Planning - ERP	19
2.4.2 Descoberta de Conhecimento em Banco de Dados	20
2.5 Considerações Sobre o Capítulo	22

3	Material e Método	24
3.1	Materiais	24
3.1.1	Sistema Gerenciador de Banco de Dados	25
3.1.2	Modelagem dos dados	25
3.1.3	Mapeamento das Classes e Interface	25
3.2	Métodos	26
3.3	Considerações Sobre o Capítulo	27
4	Resultados e Discussão	29
4.1	Definição da Arquitetura	29
4.2	Dados Brutos	30
4.3	Restrições e Composições	32
4.4	Informações Gerenciais	33
4.4.1	Usuários e Permissões	33
4.4.2	Gerenciamento de Sensores	34
4.5	Lógica da Validação e Composição de Dados	35
4.5.1	Funcionamento da Validação	36
4.5.2	Composição	37
4.6	Ontologias e Visões	37
4.6.1	Ontologias	38
4.6.2	Visões	38
4.7	Estudo de Caso	40
4.7.1	Centralização dos Dados	40
4.7.2	Controle de Acesso	41
4.7.3	Rastreabilidade de Informações	43
4.7.4	Execução da Validação	44
4.7.5	Acesso as visões e exportação	46
4.8	Considerações Sobre o Capítulo	48
5	Conclusão	49
5.1	Trabalhos Futuros	50
	Bibliografia	52
A	Códigos de criação das tabelas	62

Lista de Tabelas

1	Exemplo de tabela de dados brutos.	39
2	Exemplo de visão.	40

Lista de Figuras

1	(a) Fluxo de geração de dados comerciais. (b) Fluxo de geração de dados científicos.	10
2	Composição do modelo relacional.	14
3	Processo de KDD. Adaptado de Fayyad (1996)	22
4	Arquitetura do sistema.	26
5	Arquitetura dos dados.	30
6	Modelagem que contempla o armazenamento dos dados brutos.	31
7	Modelagem das restrições.	32
8	Modelagem das regras de acesso aos dados.	34
9	Modelagem do gerenciamento dos sensores.	35
10	Diagrama do fluxo de validação dos dados.	37
11	Diagrama do fluxo composição de dados.	38
12	Modelagem das ontologias.	39
13	Tela de Importação de Dados.	41
14	Tela de Cadastro de Organizações.	42
15	Diagrama de Atividades do fluxo de controle de acesso aos dados.	42
16	Informações detalhadas na tela Dados.	43
17	Informações detalhadas na tela Sensors.	44
18	Tela de criação das regras de validação de dados.	45
19	Árvore de execução gerada na validação dos dados de temperatura.	46
20	Tela que exibe os dados de uma visão.	47
21	Tela que exibe o processo de exportação dos dados de uma visão.	47

Capítulo 1

Introdução

1.1 Problemática

O gerenciamento correto da informação é com certeza um fator importante para garantir confiabilidade de qualquer organização. Nas aplicações com interesses financeiros o domínio da informação garante competitividade no mercado, em instituições de pesquisa garante confiabilidade nos seus resultados apresentados à sociedade.

Esse gerenciamento das informações passa inevitavelmente pelo armazenamento adequado das mesmas. Em qualquer tarefa que se use dados coletados para tomada de decisões e análise de resultados, o armazenamento é necessário, em todas as fases do processo, principalmente em pesquisa científica (BUNEMAN et al., 2004).

Problemas em relação ao armazenamento surgem quando há a necessidade de compartilhamento dos dados entre vários usuários, além das questões de armazenamento e controle dos aspectos físicos de hardware. O problema do uso concorrente dos dados pelos usuários pode levar a situações nas quais alterações e exclusões indevidas sejam de difícil detecção.

Após realizado o armazenamento, a manipulação e o processamento das informações é importante para que os usuários consigam obter os resultados esperados. Nas fases de processamento, o início ocorre com a coleta dos dados, o que em aplicações comerciais é mais simples, pois os dados são cadastrados diretamente por um usuário do sistema, que

controla o armazenamento da informação. Em aplicações científicas, nas quais geralmente os dados são coletados por sensores eletrônicos, o fluxo é mais complexo, com várias etapas de armazenamento e de interações do usuário. Nesse processamento, normalmente busca-se extrair informações que tragam significado útil ao usuário, o que aumenta a complexidade, no sentido de que a utilidade é dependente do contexto do usuário e do contexto das informações em si (FIGUEIREDO, 2005). Já na maioria das aplicações puramente comerciais, como nos balanços financeiros e controle de estoque o processamento já é bem definido e consagrado.

Com o avanço na tecnologia de armazenamento dos dados e capacidade de processamento dos computadores atuais, tornou-se mais barato armazenar grandes quantidade de dados, com isso, os dados podem ser disponibilizados *online* e podem ser utilizados para simulações computacionais.

Contudo, em aplicações científicas os dados geralmente estão disponíveis na forma bruta, como imagens ruidosas, dados de malha não estruturada com variáveis físicas em cada ponto da malha, ou como a saída de sensores diferentes observando a mesma cena. Para manipular essas estruturas de dados complexas é necessário o uso de soluções sofisticadas de computação. No entanto, nem sempre os cientistas têm acesso a ferramentas computacionais mais sofisticadas e eficientes para tratamento dos seus dados (GRAY et al., 2005). Isso porque a interdisciplinaridade é operacionalmente difícil de ser alcançada com a necessidade de especialistas de áreas diferentes trabalharem juntos com um mesmo objetivo e complementando seus conhecimentos.

Nesse contexto, o uso das ferramentas computacionais por cientistas de outras áreas gera uma complexidade adicional às pesquisas, nas quais os dados são usados principalmente com o objetivo de demonstrar modelagens e representações da realidade e, nesse sentido usar ferramentas computacionais adicionais torna o processo como um todo mais complexo. Gray et al. (2005) ao perguntar para um conjunto de cientistas o motivo de não usarem um gerenciador de banco de dados para armazenarem seus dados obteve as seguintes respostas:

- Não percebem vantagem no uso diante do custo de aprendizagem de novas ferramentas;

- Impossibilidade de acessar os dados através de suas ferramentas tradicionais de manipulação;
- Exigência de um administrador para o banco de dados.

Atualmente o cenário tem mudado, de acordo com Benioff e Lazowska (2005) a computação revolucionou a pesquisa científica, sendo hoje reconhecida como o "terceiro pilar" a sustentar tal pesquisa, junto com os pilares da teoria e da experimentação. O primeiro motivo que se pode observar, como motivação para a utilização de técnicas sofisticadas de computação em diversas áreas, é a quantidade de dados gerados nos últimos tempos. O desenvolvimento contínuo de novas tecnologias de sensores tem aumentado como de forma nunca vista a quantidade de dados coletados. Esse desenvolvimento fez com que o tratamento de grande quantidade de dados se tornasse um dos desafios da pesquisa científica em computação segundo a Sociedade Brasileira de Computação (SBC)(LUCENA et al., 2006).

Entretanto, não é apenas a quantidade de dados que tem mudado esse cenário, mas também a variedade de tipos de dados como as geradas pela variedade de sensores coletores de dados com características diferentes, o que torna os dados científicos naturalmente mais complexos. Com sensores cada vez mais modernos e precisos, as fontes de dados tornaram-se mais variadas. Quando não há fontes diferentes, pode-se ter a mesma fonte de dados com observações diferentes, como por exemplo, as imagens obtidas por sensoriamento remoto, onde a mesma imagem pode ser obtida em diferentes frequências e resoluções, há sensores que medem o mesmo fenômeno e os digitalizam de forma diferente.

Essa variedade de informações exige também que cada tipo seja tratado de forma específica. Por exemplo, formas diferentes de tratamento devem ser dadas para dados temporais, imagens e sequências genômicas. Isso significa que as funcionalidades tradicionalmente implementadas nos Sistemas Gerenciadores de Banco de Dados (SGBDs) para garantir a eficiência na manipulação dos dados não são suficientes para atender à demanda gerada por esses novos tipos de dados.

No contexto do Programa de Pós-Graduação em Física Ambiental (PGFA) da Universidade Federal de Mato Grosso (UFMT) o cenário exposto anteriormente é facilmente

notado. O programa desenvolve suas pesquisas principalmente através da coleta e análise de dados micrometeorológicos. Dados esses com comportamento altamente não lineares e não previsíveis. Para análise de variáveis micrometeorológicas deve-se coletar esses dados em um período de tempo muito pequeno, na maioria das vezes em ordem de segundos, com isso, percebe-se claramente que grande quantidade de dados são gerados.

Além da quantidade de dados, como existem muitas variáveis, conseqüentemente, há também diversos sensores que captam esses dados. Essa diversidade de sensores e de variáveis leva a inevitável complexidade desses dados. Diante da complexidade e quantidade das informações, o processo operacional envolvido na análise desses dados se torna também complexo e muitas vezes erros são cometidos. O tratamento dos dados após a coleta passa por um processo de limpeza, onde dados incorretos, devido a falha de equipamento, erro de coleta ou simplesmente os *outliers* são retirados para não inserirem erros na análise. Por ser um processo manual, esse processo de limpeza, é demasiadamente demorado e complicado, diante dos problemas já expostos anteriormente.

No PGFA além do processo de análise dos dados coletados em busca de informações úteis, outro problema notado é o armazenamento confiável desses dados. Na grande maioria das vezes, cada pesquisador mantém os dados em planilhas eletrônicas no computador pessoal, sem uma prática sistemática de armazenamento e garantia de integridade dos dados. Esse problema se acentua quando esses dados têm de ser pré-processados, sem o armazenamento adequado, e sem uma ferramenta de versionamento que garanta o controle de versões dos dados, por exemplo, um erro pode ser propagado e ainda ocasionar retrabalho.

Além do armazenamento, o compartilhamento dos dados torna-se falho quando não se tem uma forma confiável de armazenamento dos dados. Como as pesquisas são desenvolvidas em grupos, inevitavelmente as informações devem ser compartilhadas, e nesse ponto faz-se extremamente necessário que haja formas de se controlar o compartilhamento, garantir confiabilidade dos dados e ainda um mecanismo de validação automática dos dados.

Diante do exposto, faz-se necessário a utilização de mecanismos computacionais que permitam a centralização dos dados, bem como seu gerenciamento baseado em usuários, projetos e validação automática.

1.2 Justificativa

Grande parte da problemática apresentada se aplica às variáveis micrometeorológicas e o contexto do PGFA tem nessas variáveis o principal campo de estudo. O PGFA tem alcançado resultados positivos e extremamente importantes para o Estado, principalmente porque encontra-se no Estado de Mato Grosso três dos principais biomas do país, a Floresta Amazônica, Cerrado e Pantanal.

Os resultados alcançados são frutos do esforço conjunto dos pesquisadores e da constante busca por resultados cada vez mais precisos e confiáveis. Para que se alcance resultados confiáveis, é imprescindível que se tenha dados também confiáveis. A garantia de dados confiáveis passa por métodos sistemáticos desde a calibração dos equipamentos, coleta, armazenamento e processamento dos dados. No entanto, à medida que se aumenta a quantidade de dados coletados, o simples armazenamento dos mesmos em arquivos passa a ser factível de erros e até problemas como perda dos dados.

Perda de informações e problemas com versionamentos são comuns quando se trabalha com arquivos. Além dos problemas citados, há dificuldade em se trabalhar com compartilhamento dos dados, e este, aumenta consideravelmente na proporção que aumenta o número de usuários.

Diante do exposto, torna-se necessário recorrer a técnicas sofisticadas de computação que proporcionem armazenamento e gerenciamento adequado desses dados. Técnicas que permitam a modelagem adequada das regras de negócios envolvidas no processamento desses dados e que os centralizem, em uma base de dados única e robusta, permitindo compartilhamento, replicação e validação dos dados.

Além do armazenamento, é necessário que se tenha uma forma eficiente de executar o pré-processamento, que é a fase em que os dados são tratados de forma a serem melhores interpretados. Essa fase é indispensável, principalmente devido ao comportamento estocástico das variáveis medidas, não é incomum que se tenha *outliers* e que eles sejam analisados, ou ainda que haja falhas na obtenção dos dados.

1.3 Objetivo Geral

O objetivo principal deste trabalho é fornecer um ambiente computacional para o gerenciamento de grande quantidade de dados científicos com aplicação principal sobre variáveis micrometeorológicas.

1.4 Objetivos Específicos

Para se alcançar o objetivo geral, alguns objetivos específicos devem antes ser alcançados, são eles:

- Estudar o comportamento dos dados micrometeorológicos.
- Modelar o armazenamento dos dados.
- Modelar o controle de acesso aos dados.
- Modelar o armazenamento das restrições de cada tipo de dado.
- Automatizar o processo de validação dos dados.
- Trazer para o ambiente dos dados científico a integração e flexibilidade providas pela tecnologia de *Enterprise Resource Planning e Business Intelligence*.
- Avaliar a implementação.

1.5 Contextualização do Trabalho

Este trabalho se insere no contexto do Grupo de Pesquisa em Gerenciamento e Armazenamento Inteligente de Imagens Ambientais (GAIIA) que se dedica a aplicar técnicas de computação ao tratamento e armazenamento de dados ambientais (OLIVEIRA et al., 2008), (OLIVEIRA; FIGUEIREDO, 2009), (OLIVEIRA et al., 2010), (OLIVEIRA et al., 2010), (VENTURA et al., 2010).

1.5.1 Gaiia

Uma das principais frentes de pesquisa do GAIIA se dedica a implementação de uma arquitetura de componentes de software que dê suporte ao gerenciamento e armazenamento mais eficiente no tocante as técnicas computacionais envolvidas no processo de controle ambiental utilizando dados obtidos de sensores variados, como sensores orbitais que geram imagens de satélites.

A composição dos componentes de software gera um ambiente integrado denominado Environmental Resource Planning (EnvRP), o nome deriva do Enterprise Resource Planning (ERP), no qual um banco de dados robusto é utilizado de forma que as informações sejam todas inter-relacionadas e bem mapeadas, buscando-se eficiência no armazenamento e recuperação dos dados. Assim, este trabalho é focado na estruturação da base de dados que compõem o EnvRP para dar suporte a implementação dos módulos que devem compor o sistema. Além de um sistema para gerenciamento de dados ambientais com as características dos ERPs, como resultado deste trabalho, a arquitetura de dados elaborada fornece também suporte para que outras técnicas de Banco de Dados, bem como de Inteligência Artificial possam ser aplicadas aos dados científicos.

1.6 Organização do Trabalho

O documento está organizado com a seguinte estrutura:

- **Capítulo 2:** Nesse capítulo é apresentada a fundamentação teórica necessária para o entendimento e contextualização deste trabalho. O capítulo é organizado iniciando-se com a abordagem introdutória a respeito das diferenças entre as arquiteturas de tratamento dos dados comerciais e os dados científicos. Nesse contexto, são apresentados os conceitos sobre os banco de dados relacionais, Enterprise Resource Planning e Business Intelligence.
- **Capítulo 3:** Apresenta os materiais e a metodologia utilizada no desenvolvimento do trabalho.

- **Capítulo 4:** Nesse capítulo são apresentados e discutidos os resultados do trabalho desenvolvido de acordo a metodologia apresentada no Capítulo 3.
- **Capítulo 5:** Nesse capítulo é mostrada a conclusão e os possíveis trabalhos futuros relacionados a este trabalho.

Capítulo 2

Fundamentação Teórica

*E*m qualquer área do conhecimento um dos bens mais importante são os dados. São eles que dão significados a toda informação importante dentro de uma empresa, instituição acadêmica ou órgão governamental. Segundo Oliveira, Rodrigues e Rangel (2004), tanto organizações públicas como privadas passaram a perceber o valor dos dados, e a considerá-los como um bem importante para a competitividade.

A forma do tratamento e armazenamento dos dados está diretamente relacionada com a natureza dos dados em si. Nesse contexto, é preciso um entendimento sobre as propriedades dos dados e sua organização de forma a poderem ser manipulados pelos softwares e armazenados pelo hardware. Assim, é preciso compreender os componentes e a arquitetura do fluxo dos dados, bem como a forma de organização e processamento em cada etapa.

Este capítulo faz uma breve abordagem a respeito das diferenças existentes entre as arquiteturas de manipulação de dados comerciais e de dados científicos, com a abordagem dos conceitos envolvidos no modelo relacional (Seção 2.2.1) de dados e ainda características específicas dos dados científicos e, como as questões da arquitetura corporativa de software com foco nos ERPs e na descoberta de conhecimento em base de dados podem ser aplicados em um ambiente de gerenciamento de dados científicos.

2.1 Arquiteturas de tratamento de dados

A organização dos dados é a primeira etapa para que sua manipulação ocorra de forma adequada e atenda às demandas da organização. Essa organização dos dados tem como etapas a modelagem e a implementação dentro da arquitetura do Sistema Gerenciador de Banco de Dados (SGBD) escolhido.

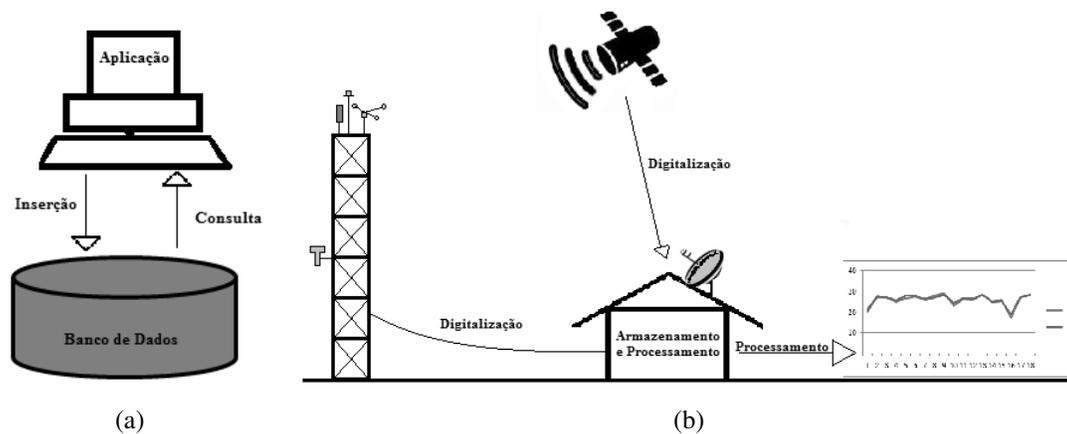


Figura 1: (a) Fluxo de geração de dados comerciais. (b) Fluxo de geração de dados científicos.

A modelagem e implementação dependem do contexto no qual os dados estão sendo manipulados, assim é preciso entender a arquitetura do ambiente computacional. Normalmente, a arquitetura de ambientes computacionais comerciais possui características na forma de acesso às informações armazenadas no SGBD conforme ilustrada na Figura 1 (a), na qual um software denominado Aplicação se comunica um SGBD, que é outro software que controla o armazenamento e o acesso aos dados. É importante destacar que nessa arquitetura geralmente os dados novos da Aplicação são inseridos diretamente por um usuário e esses dados são categóricos, bem comportados e com estrutura definida.

Entretanto, em outros contextos, em que os dados são complexos, a manipulação é mais variada, com muitos outros componentes na arquitetura e, os dados são inviáveis de serem inseridos diretamente pelo usuário. Exemplos desses dados são o histograma de uma imagem médica, dos dados oriundos de sensores orbitais, dos resultados de uma

simulação, dos dados de sensores de temperatura, entre outros tipos de dados. A Figura 1(b) mostra uma arquitetura que contempla o fluxo de obtenção dos dados complexos. Nesse contexto, é necessário identificar que a manipulação dos dados pode variar muito, pois o processo é iniciado por sensores que fazem a medição de fenômenos reais e, esse processo de digitalização envolve dispositivos com algoritmos embutidos de processamento de sinais que entre outros objetivos tentam diminuir o ruído existente na captação de forma a separar somente os sinais considerados importantes.

Um exemplo representativo é o processo de aquisição de imagens de Sensoriamento Remoto (SR), no qual são feitas medidas da radiação solar refletida pelos objetos na Terra (OLIVEIRA, 2008). Essa reflexão é captada pelos sensores em diversas faixas de frequências denominadas Bandas Espectrais, seu resultado é então digitalizado para formação dos *pixels* das imagens. A intensidade dos raios refletidos define o valor do *pixel*, esses valores podem variar de acordo com a resolução radiométrica do sensor, que na grande maioria são de 8 bits, o que gera valores de pixel variando de 0 a 255.

Após a digitalização ocorre a primeira etapa de armazenamento dos dados no próprio dispositivo, o que pode ser específico, pois cada sensor pode digitalizar os dados utilizando protocolos diferentes. Finalizada a digitalização e armazenamento local do dado, é necessário transmiti-lo para outros dispositivos, etapas que podem se repetir até que o dado seja armazenado no servidor central, no qual ocorre a integração das informações e, finalmente o usuário interage com o objetivo de analisá-lo.

Após feita a captação e o armazenamento dos dados, sua manipulação que vai determinar as informações e os conhecimentos a serem gerados. Manipulações simples, como visualização permitem obter o valor direto dos dados, como no caso da cotação de uma moeda em uma determinada data. Já manipulações complexas geram valores que não são simples de gerar, como a identificação de comportamentos e tendências nos dados.

Para que a manipulação dos dados seja feita de forma adequada, é necessário contextualizá-la ao tipo de dado, ou seja, dependendo das propriedades dos dados um conjunto de manipulações deve ser aplicado. Essas propriedades determinam qual(is) algoritmo(s) pode(m) ser utilizado(s), qual(is) organização(ões) em disco, qual(is) método de acesso e assim sucessivamente. Dessa forma, letras e números possuem características simples enquanto imagens, informações geográficas e vídeos têm propriedades complexas (ADALI;

SAPINO; SUBRAHMANIAN, 1999; ADALI; SAPINO; SUBRAHMANIAN, 2000; ATNAFU; BRUNIE; KOSCH, 2001; KIRANYAZ et al., 2003). Aplicações comerciais normalmente manipulam somente dados em formatos simples.

Uma das propriedades que pode existir em todos os tipos de dados é a dimensão tempo, ou seja, a maioria dos bancos de dados atuais descrevem somente estado atual da informação e não suas mudanças sofridas com o tempo (JOHNSTON; WEIS, 2010). Com a maioria das técnicas de tratamento de dados tendendo a tratar os dados sem considerar a dimensão temporal, com os dados sendo considerados como uma coleção não ordenada de eventos (ANTUNES; OLIVEIRA, 2001). Contudo, diversos contextos demandam a manipulação de informações históricas e dados relativos a estados da variável, nesses contextos os Banco de Dados são denominados Temporais, pois permitem armazenar todos os estados de uma aplicação, registrando sua evolução com o passar do tempo. Esses controles sobre a dimensão tempo incluem manipulação de datas, períodos, duração de validade de informações e intervalos temporais; o que influencia os dados em diferentes níveis: na modelagem de dados, na linguagem de recuperação e manipulação de dados, no nível de implementação do SGBD.

Além da manipulação das propriedades dos dados, seja no contexto comercial quanto científico, seja considerando ou não a dimensão tempo, há a necessidade de se interpretar de forma eficiente esses dados. Para Ye (2003) a complexidade dos dados científicos é o resultado do avanço da tecnologia, como o melhoramento de sensores, capacidade de armazenar dados e poder computacional. Sendo que a coleta desses dados está acontecendo muito mais rapidamente do que a capacidade humana de análise manual, por isso é necessário utilizar da tecnologia para que os pesquisadores possam encontrar as informações úteis em uma imensa quantidade de dados (KAMATH, 2006).

2.2 Banco de Dados

A gestão adequada dos dados é um fator de vital importância para a sobrevivência de qualquer organização, seja ela pública ou privada, com ou sem interesses comerciais. Esse processo de gestão dos dados, tradicionalmente, tem como objetivo organizar os dados para que os mesmos possam ser armazenados em algum dispositivo para posteriormente

serem recuperados.

Segundo Date (2000), um sistema de banco de dados pode ser interpretado como um sistema computadorizado de armazenamento e recuperação de registros, o que significa que esse tipo de sistema não se resume apenas ao armazenamento dos registros, mas oferece também a possibilidade de manipulação e gerenciamento desses registros.

A manipulação adequada desses registros é que permite a execução do processo de interpretação de um fenômeno, análise de ocorrências em um determinado período, otimização e verificação de rotinas e resultados. Essa manipulação pode acontecer sob a forma de alterações, inserções e exclusões de dados. De acordo com Silberschatz, Korth e Sudarshan (1999), um Sistema Gerenciador de Banco de Dados (SGBD) é constituído por um conjunto de dados integrados a um conjunto de programas para acesso a esses dados

É responsabilidade dos SGBDs garantir o armazenamento correto, seguro e eficiente dos dados. Com isso, fica praticamente garantida a otimização e a segurança da recuperação das informações, fator determinante do sucesso de uma aplicação de armazenamento de informações.

O primeiro SGBD comercial surgiu na década de 1960 com base nos primitivos sistemas de arquivos disponíveis na época, os quais não controlavam o acesso concorrente por vários usuários ou processos. Esse modelo é utilizado ainda hoje, a partir do momento em que os dados são armazenados em estruturas simples de arquivos, planilhas eletrônicas por exemplo.

Os SGBDs evoluíram desses sistemas de arquivos de armazenamento em disco, criando novas estruturas de dados com o objetivo de armazenar informações. Com o tempo, os SGBDs passaram a utilizar diferentes formas de organização e representação, ou modelos de dados, para descrever a estrutura das informações contidas em seus bancos de dados. Na seção a seguir é apresentado o modelo relacional de dados, que é o modelo utilizado neste trabalho.

2.2.1 Modelo Relacional

O Modelo Relacional criado por Edgar Frank Codd em 1970 baseia-se nas relações matemáticas da teoria dos conjuntos. Codd buscava um modelo que resolvesse parte dos pro-

blemas existentes nos modelos utilizados, o Hierárquico e de Redes. Segundo Codd (1970) os usuários de banco de dados de grandes dimensões não devem saber os detalhes físicos de armazenamento dos dados, ou seja, esses detalhes devem estar transparente para o usuário. Essa transparência é necessária porque muitas vezes, mudanças na representação de dados, atualização de relatório e crescimento natural nos tipos de informações armazenadas podem acontecer.

A principal proposição do modelo relacional é que todos os dados são representados como relações matemáticas, isto é, um subconjunto do produto Cartesiano de n conjuntos. O banco de dados é representado com uma coleção de relações, cada relação representa uma tabela de valores, informalmente, costuma-se chamar uma relação apenas de Tabela.

Apesar de serem chamadas comumente de tabelas, algumas características tornam as relações diferentes de um arquivo ou uma tabela comum. Uma relação é definida como um conjunto de tuplas. Matematicamente, os elementos de um conjunto não têm ordem entre eles, portanto, tuplas em uma relação não tem qualquer ordem particular. No entanto, em um arquivo, os registros são armazenados fisicamente no disco (ou memória), então há sempre uma ordem entre os registros (ELMASRI; NAVATHE, 2003).

No modelo relacional, cada linha na tabela representa um fato que geralmente corresponde a uma entidade do mundo real ou relacionamento. Os nomes das colunas das tabelas são usados para ajudar na interpretação do significado dos valores em cada linha.



Figura 2: Composição do modelo relacional.

A Figura 2 ilustra os componentes de uma relação, formalmente a terminologia correta para cada linha de uma relação é Tupla, e cada coluna representa um atributo. Para cada atributo existe um domínio de valores possíveis, na subseção a seguir são definidos mais

precisamente cada um desses termos, Relação, Tupla, Atributo e Domínio.

Domínio, Atributo, Tuplas e Relações

Um domínio D é um conjunto de valores atômicos, ou seja, valores indivisíveis. Um domínio é especificado por um tipo de dado, ou seja, o tipo determina quais valores constituem o domínio. Também é útil especificar um nome para o domínio, para ajudar na interpretação de seus valores. Alguns exemplos de domínios são:

- `numero_telefone`: O conjunto de números de 10 dígitos válidos para um telefone no Brasil.
- `cpf`: O conjunto de números de 11 dígitos válidos para um CPF no Brasil.
- `numero_rga`: O conjunto de números de 4 dígitos válidos para representar o registro de um aluno de pós-graduação na UFMT.

Os exemplos anteriores são definições lógicas de domínios, um tipo de dado ou um formato pode também ser especificado para cada domínio. Como exemplo, o domínio `numero_telefone` pode ser declarado como uma cadeia de caracteres da forma $(dd) dddd dddd$, onde d é um valor numérico (decimal) e os dois primeiros dígitos formam um código de área válido para um telefone. Além do nome, tipo de dado e do formato, um domínio pode ainda ter informações adicionais que auxiliem na interpretação e validação dos valores, como exemplo, um domínio `Temperatura` pode ter uma unidade de medida, como graus *Celsius*, *Fahrenheit*, *Kelvin* e ainda uma faixa de valores possíveis, como -20 até 100 graus Celsius.

Um esquema relacional R , denotado por $R(A_1, A_2, \dots, A_n)$, é composto por um nome da relação R e por um conjunto de atributos A_1, A_2, \dots, A_n . Cada atributo A_i é o nome de um papel desempenhado por um domínio D em um esquema relacional R . O domínio D é chamado de domínio de A_i e é denotado por $\text{dom}(A_i)$. Um esquema relacional é usado para descrever uma relação, R é chamado de nome desta relação. O grau de uma relação é o número de atributos n de um esquema relacional.

O esquema relacional mostrado na Figura 2 por exemplo é uma relação de grau 4 que descreve um aluno da seguinte forma: $ALUNO(RGA, NOME, CPF, TELEFONE)$, a

mesma relação pode ser descrita usando o tipo de dado para cada atributo: *ALUNO*(*RGA*: numérico, *NOME*: string, *CPF*: numérico, *TELEFONE*: numérico)

Uma relação r de um esquema relacional $R((A_1), A_2, \dots, A_n)$, denotado por $r(R)$, é um conjunto de n -tuplas $r = t_1, t_2, \dots, t_n$. Cada n -tupla t é uma lista ordenada de n valores $t = \{v_1, v_2, \dots, v_n\}$, onde cada valor v_i , $1 \leq i \leq n$, é um elemento do $\text{dom}(A_i)$ ou é um valor especial, como *null*. O i -ésimo valor na tupla t , que corresponde ao atributo A_i , é referido como $t[A_i]$.

A definição de uma relação pode ser reescrita mais formalmente como segue. Uma relação $r(R)$ é uma relação matemática de grau n sobre os domínios $\text{dom}(A_1)$, $\text{dom}(A_2)$, ..., $\text{dom}(A_n)$, que é um subconjunto do produto cartesiano dos domínios que definem R :

$$r(R) = (\text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n)) \quad (1)$$

O produto cartesiano especifica todas as combinações possíveis de valores dos domínios subjacentes. Assim, se chamarmos o número total de valores, ou cardinalidade, de um domínio D por $|D|$ (assumindo que todos os domínios são finitos), o número total de tuplas do produto cartesiano é:

$$|\text{dom}(A_1)| \times |\text{dom}(A_2)| \times \dots \times |\text{dom}(A_n)| \quad (2)$$

De todas essas possíveis combinações, o estado de uma relação em um dado momento, o estado atual, reflete apenas tuplas válidas que representam um estado particular no mundo real. Em geral, como o estado do mundo real muda, o mesmo acontece com a relação, ao ser transformada em outro estado da relação.

2.3 Dados Científicos

A definição de dados científicos está condicionada à definição de ciência, que para Pfaltz (2007), está fundamentalmente relacionada à observação de fenômenos reais e posterior gravação dessas observações. Observações estas, que são feitas através de instrumentos científicos e simulações computacionais que geram quantidades grandes de dados para

serem armazenados (GRAY et al., 2005; FAYYAD; HAUSSLER; STOLORZ, 1996; JACOBSON; BERKIN; ORTON, 1994).

Pode-se observar em Gray et al. (2005) e (BOSE; FREW, 2005) algumas divisões dos dados científicos feitos em ((NASA), 1986) em relação ao processamento. Eles são divididos em 3 níveis:

Nível 0 São os dados brutos vindos direto do equipamento de medida sem nenhum processamento;

Nível 1 Neste nível os dados são processados de alguma forma, como mudança de resolução e calibragem;

Nível 2 São dados derivados do nível 1, geralmente são combinações de dados, resultantes por exemplo de alguma modelagem.

Esses níveis podem ainda aumentar de acordo com o grau de tratamento que é aplicado para os dados, esses níveis reforçam o questionamento levantando quanto a complexidade dos dados científicos.

Uma característica dos dados científicos é a sua dimensão temporal, na qual, geralmente, os valores estão relacionados a um instante de tempo.

Muitas aplicações de banco de dados são por natureza temporais. Como exemplo, temos as aplicações comerciais, registros médicos e a maioria das aplicações científicas, como monitoramento de recursos hídricos. Aplicações como essas, que armazenam informações dos estados do mundo real, ou seja, considerando a dimensão tempo são chamados de Banco de Dados Temporais e permitem armazenar e recuperar todos os estados de um dado ao longo do tempo, diferentemente dos Bancos de Dados Operacionais, em que a realidade é representada apenas pelo estado atual de um dado (EDELWEISS; OLIVEIRA, 1994; OZSOYOGLU; SNODGRASS, 1995).

Dados temporais são dificilmente expressos pelos modelos de dados padrão (não temporais), fazendo com que problemas que utilizam informações temporais sejam normalmente resolvidos pelos desenvolvedores das aplicações, acima do nível do SGBD, o que pode elevar a complexidade de implementação de um sistema, e ainda levar ao surgimento de soluções proprietárias. Os banco de dados tradicionais diferem dos banco de dados

temporais na forma de execução das consultas, sobre os dados temporais, elas utilizam normalmente os operadores de desigualdade, enquanto que consultas tradicionais utilizam normalmente o operador de igualdade (JENSEN; SNODGRASS, 1997).

Para Mariote (2008) podem ser identificadas duas grandes linhas no que diz respeito ao tratamento de atributos temporais. A primeira defende que um atributo temporal seja visto como qualquer outro, para Shoshani e Kawagoe (1986) e Parent, Spaccapietra e Zimányi (1999), qualquer atributo pode ser visto como uma série de dados, o que possibilita uma simples definição de operadores, e que esses dados sejam tratados da mesma forma que os demais elementos do banco. A segunda defende que um atributo temporal seja visto de forma diferente dos demais, para Erwig et al. (1998), esses atributos devem ser tratados de maneira distinta dos demais. Algumas técnicas e conceitos, como a normalização e as dependências funcionais, apresentam características diferentes para dados temporais.

Um dado pode ser caracterizado como temporal quando apresenta mais de um estado ao longo do tempo. Um estado pode ser compreendido como um valor associado a um objeto durante certo intervalo de tempo. Um objeto tem seu estado alterado por meio de um evento, que pode ser visto como um fenômeno instantâneo (JENSEN; SNODGRASS, 1996).

Em uma análise mais simples, um objeto temporal pode ser visto como um conjunto de atributos que podem assumir diferentes valores em intervalos de tempo distintos. Em um determinado instante de tempo, porém, eles possuem apenas um valor. Uma sequência de dados pode ser vista da seguinte forma: seja t uma tabela de entradas do tipo $\langle id, (\text{tempo}, \text{valor})^* \rangle$, onde o asterisco indica repetição, ou seja, uma mesma identificação id pode assumir mais de um valor ao longo do tempo. A tupla $(\text{tempo}, \text{valor})$ representa uma sequência temporal, e está associada à chave id .

No contexto dos dados temporais temos as *séries temporais*, que são valores associados a um objeto, sendo que o objeto não possui dois valores distintos no mesmo instante de tempo. As principais diferenças entre séries temporais e sequências de dados são relativas ao volume de dados e ao conjunto de operações necessário para realizar todas as manipulações desejadas com os dados brutos (RAFIEI; MENDELZON, 1997).

2.4 Arquitetura Corporativa de Software

Um sistema de informação pode ser definido tecnicamente como um conjunto de componentes inter-relacionados que coletam (ou recuperam), processam, armazenam e distribuem informações destinadas a apoiar as tomadas de decisão, a coordenação e o controle de uma organização (LAUDON; LAUDON, 2007; ANDRADE, 2010). Dentre os vários sistemas de informações disponíveis o *Enterprise Resource Planning* tem por principal característica integrar todos os módulos de um sistema corporativo.

2.4.1 Enterprise Resource Planning - ERP

O ERP é um conjunto de sistemas que tem como objetivo agregar e estabelecer relações de informação entre todas as áreas de uma companhia. Surgiu como sucessor do MRPs (*Material Requirement Planning* ou planejamento das requisições de materiais) e do MRP II (*Manufacturing Resource Planning* ou planejamento dos recursos de manufatura)(FERREIRA, 2004; MCCLELLAN, 200; LAUDON; LAUDON, 2000). Além do planejamento dos recursos de produção, os ERPs agregaram controles dos demais recursos empresariais, como recursos humanos, vendas e finanças (LAUDON; LAUDON, 2000).

Esse tipo de sistema teve sua origem da necessidade das empresas em gerir suas estruturas organizacionais, cada vez mais complexas e dinâmicas(RICCIO, 2001). Keller (1999) cita que no final dos anos 80, muitas das empresas de manufatura diziam-se insatisfeitas com a geração de software empresariais da época, que ofereciam pouca flexibilidade e interfaces pouco amigáveis.

Os ERPs passaram então a ser visto como algo diferente e inovador, o que despertou interesse de milhares de empresas (RICCIO, 2001). Com essa popularidade, a implementação dos ERPs passou a fazer parte das decisões estratégicas tomadas pelas grandes empresas em todo o mundo(CURRIE; GALLIERS, 1999; LUCAS, 1999; WILKINSON et al., 1999).

Uma das principais vantagens da utilização de um ERP segundo Kale (2000) é a integração de todos os dados em um único banco de dados, garantindo a integração entre os sistemas e todos os usuários. É importante entender que, nesse caso, trata-se de um sistema de informações integrado e não uma empresa integrada, para Souza e Zwicker

(2003), a utilização de um sistema de informação integrado objetiva a empresa integrada. Outras vantagens ainda podem ser citadas:

- Padroniza os processos, aumentando sua eficiência;
- Provê a habilidade de conhecer e implementar melhores práticas globais;
- Provê um instrumento para melhoria de qualidade e padronização de esforços, incluindo Controle de Qualidade, Análise de Qualidade e *Total Quality Management* - TQM (CALARGE; LIMA, 2001; WANG, 1998; RAVICHANDRAN; RAI, 1999);
- Implementa práticas do tipo "*best in class*" - melhores práticas - e oferece meios para a execução de "*benchmarking*";
- Permite que envolvidos no controle operacional acessem os dados necessários para decisões rotineiras, oferecendo a cada um maior conhecimento e domínio sobre suas atividades.

A flexibilidade é alcançada através da padronização das configurações do sistema baseada em relações no banco de dados. Um exemplo é o Adempiere ERP Adempiere (2011), um ERP de código aberto que tem como principal característica a flexibilidade. Suas janelas são geradas a partir de configurações baseadas em tabelas no banco de dados. Existe um conjunto de tabelas responsáveis por formarem o dicionário de aplicação. Este dicionário guarda informações de criação das janelas baseadas em configurações da organização e nível de usuário. Dentre as configurações possíveis através do dicionário de aplicação está a internacionalização do sistema, que permite que vários idiomas sejam configurados.

2.4.2 Descoberta de Conhecimento em Banco de Dados

O avanço na capacidade de se obter e armazenar dados têm colocado como desafio a implementação de técnicas que consigam mensurar e descobrir padrões relevantes nesses dados. Dessa forma, segundo Gonçalves, Pacheco e Morales (2001) uma nova perspectiva é apresentada na análise de dados, na qual a análise é vista com um caráter exploratório. Essa perspectiva, ou forma de análise caracteriza a chamada Descoberta de Conhecimento

em Banco de Dados (*Knowledge Discovery on Data Base - KDD*), que segundo Fayyad (1998), é o processo de identificar estruturas que representem informação válida, inédita, potencialmente útil e essencialmente compreensível, em uma massa de dados. O termo estrutura aqui representa qualquer padrão, ou modelo que propicie uma representação parcimoniosa de um sub-conjunto interessante da coleção de dados.

De forma mais simplista, KDD representa o processo de transformação de dados de baixo nível em conhecimento de alto nível (GOEBEL; GRUENWALD, 1999). Esse processo ou visão de análise de dados baseia-se nas tendências de avalanche de formação de dados e o questionamento sobre o significado e conteúdo desses dados (HAIR et al., 2009).

Incluso ao processo KDD está a Mineração de Dados (MD) (*Data Mining*), que consiste na exploração e análise, por meios automáticos ou semi-automáticos, de grandes quantidades de dados para descobrir modelos e regras significativas. A proposta da Mineração de Dados consiste em proporcionar uma perspectiva nova ou, mais precisamente, uma evolução nos processos de análise, permitindo a descoberta de novos padrões ou a validação de padrões conhecidos.

Tais análises são geralmente efetuadas em grandes quantidades de dados, e isso pode se tornar um problema. De acordo com Hinneburg e Keim (1999), técnicas e algoritmos de busca de dados que se mostram adequados para conjuntos de dados pequenos podem não o ser quando o volume de dados cresce ordens de grandeza, tanto para o número de atributos (dimensões) envolvidos, quanto para o número de itens tratados.

Também é importante reconhecer que num processo de MD o objetivo da procura nem sempre é claramente definido. Por exemplo, é frequente a busca de algum padrão no relacionamento entre diversos atributos, ou a busca de agrupamentos (*clusters*), mas quais atributos, ou que padrão seria esse não é conhecido inicialmente (HINNEBURG; KEIM, 1999).

Segundo Fayyad (1998) e Fayyad (1996), o processo de descoberta de informação em bases de dados segue as seguintes etapas (Figura 3):

- Desenvolvimento da base de dados;
- Seleção/amostragem de dados;
- Pré-processamento e limpeza de dados;

- Transformação/redução de dados;
- Mineração de dados;
- Criação de modelos e definição de padrões;
- Avaliação da usabilidade dos padrões definidos;
- Visualização de dados;
- Extração de conhecimento.

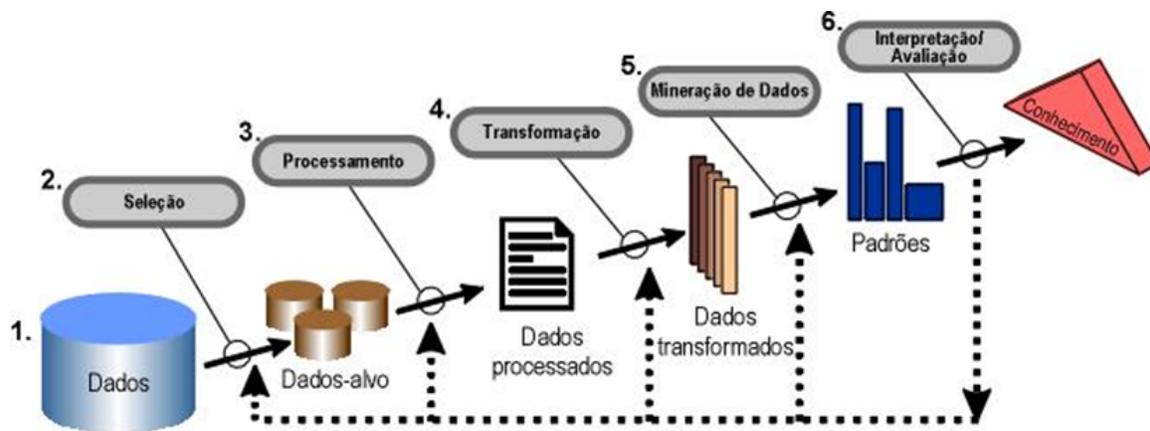


Figura 3: Processo de KDD. Adaptado de Fayyad (1996)

Com as definições de Descoberta de Conhecimento em Base de Dados e Mineração de Dados, percebe-se que são técnicas extremamente úteis ao contexto dos dados científicos, que são colhidos de forma bruta e necessitam de um nível grande de pré-processamento para transformar esses dados de uma forma que seja possível a detecção de padrões (KAMATH, 2006).

2.5 Considerações Sobre o Capítulo

Neste capítulo é apresentada a fundamentação teórica para melhor entendimento do restante do trabalho e contextualização da problemática apresentada. É importante destacar

que existe grande diferença entre o tratamento que deve ser feito para os dados científicos em relação aos comerciais, o que demonstra que este tratamento é fruto da complexidade inerente ao comportamento dos dados científicos, que tem entre suas características a dimensão temporal. Essas características e dimensões são importantes para que modelos de dados, como o relacional possa ser utilizado de forma adequada.

Além do modelo relacional, vale ressaltar que outros modelos surgiram juntamente com a evolução dos tipos de dados. De acordo com Silberschatz, Korth e Sudarshan (2001), à medida que as bases de dados foram sendo utilizadas em um âmbito maior de aplicações, as limitações impostas pelo modelo relacional emergiram como obstáculos. Como resultado, pesquisadores da área de bancos de dados inventaram novos modelos de dados que superassem as restrições do modelo relacional. Chaudhri e Zicari (2001) dizem ainda que a utilização de dados altamente estruturados, transações longas, dados em multimídia e operações fora do padrão, fazem com que novos modelos de dados sejam utilizados. Entretanto, vale destacar o que diz Pfaltz (2007), que os bancos de dados relacionais podem parecer bastante adequado para o contexto científico, pois as linhas podem denotar fenômenos ou objetos, e as colunas podem denotar seus atributos.

O uso de SGBDs pode ser feito para integrar todas os dados, regras e restrições, como a abordagem realizada pelos ERPs, cujos conceitos que formam a base para criação da arquitetura de dados proposta neste trabalho.

O capítulo abordou ainda os conceitos de KDD e MD que formam um processo de análise de dados para se extrair informações úteis dos dados. Como mostrado, o processo de KDD tem como fase inicial o uma base dados para que possa ser trabalhada, nesse sentido, a integração das informações proporcionada pelos ERPs pode ser melhor explorada com o uso do KDD. Assim, o próximo capítulo aborda o material e métodos utilizados para se formular a arquitetura de dados proposta neste trabalho. Esta arquitetura possibilita que o KDD, bem como outras técnicas computacionais, possam ser utilizadas nas pesquisas envolvendo os dados micrometeorológicos.

Capítulo 3

Material e Método

O contexto deste trabalho envolve a manipulação de dados micrometeorológicos usados nos estudos do PGFA, como pesquisas envolvendo emissividade atmosférica Curado et al. (2011) Rodrigues et al. (2011b) Rodrigues et al. (2011c), comportamento das variáveis do solo Novais et al. (2011a) Novais et al. (2011b), balanço de energia Rodrigues et al. (2011a) e evapotranspiração (VENTURA et al., 2010). Nesse sentido buscou-se criar a arquitetura de um ambiente que contemple as soluções para o armazenamento, compartilhamento, validação, análise e visualização dos dados. Com isso, o ambiente sugerido contempla uma solução genérica para manipulação de dados científicos baseado nos conceitos de uma plataforma ERP aplicado a dados ambientais.

A seguir são apresentados os materiais e métodos utilizados para construção desse ambiente.

3.1 Materiais

Os materiais escolhidos seguiram alguns critérios de escolha, o principal é a escolha de ferramentas baseadas no conceito de software livre, principalmente pela possibilidade de acesso ao código fonte quando necessário, e pela gama de informações disponíveis na rede através de fóruns e tutoriais online.

3.1.1 Sistema Gerenciador de Banco de Dados

O SGBD utilizado neste trabalho foi o PostgreSQL 9.0Postgres (2011), por ser um SGDB de código aberto e com comunidade de desenvolvedores ativa, o que facilita a busca por informações e resolução de possíveis problemas. O PostgreSQL conta ainda com a possibilidade de implementação de procedimentos armazenados em diversas linguagens procedimentais e expansão de suas funcionalidades para suportar novos requisitos, como o PostGis para manipulação de dados georeferenciados

3.1.2 Modelagem dos dados

Para elaboração do modelo conceitual da arquitetura de dados proposta, adotou-se a *Unified Modelling Language* (UML) que define um conjunto básico de diagramas e notações que permitem representar as múltiplas perspectivas (estruturais / estáticas e comportamentais / dinâmicas) do sistema sobre análise e desenvolvimento Costa (2001), Liu et al. (2004) e tem por principal vantagem reduzir os esforços na construção de modelos no desenvolvimento de software e facilitar a manutenção e reutilização durante o ciclo de vida do software(GROUP, 2011). Neste trabalho foram utilizados os Diagramas de Classe e Diagrama de Atividade (BOOCH; RUMBAUGH; JACOBSON, 1999; JACOBSON; BOOCH; RUMBAUGH, 1999).

3.1.3 Mapeamento das Classes e Interface

Devido à necessidade de se gerar um ambiente funcional para uso via Web, optou-se por utilizar uma tecnologia multiplataforma, sem o vínculo a nenhum sistema operacional para funcionamento, assim, foi escolhida a tecnologia Java Java (2011) para ambientes corporativos. Já para o mapeamento das classes, bem como para a construção da interface gráfica do sistema, foi utilizado o Jboss Seam (SEAM, 2011). O JBoss Seam é um *framework* para desenvolvimento de aplicações *Java Enterprise Edition* (JEE) que integra, facilmente, tecnologias como *Asynchronous JavaScript and XML* (AJAX), *JavaServer Faces* (JSF), *Java Persistence API* (JPA), *Enterprise Java Beans* (EJB 3.0) e *Business Process Management* (BPM)(FLAVINHA, 2008). Dessa forma, o Jboss Seam permite eliminar a complexidade

em níveis de arquitetura e API (*Application programming interface*), oferecendo ao desenvolvedor total controle sobre a implementação da lógica de negócio sem se preocupar com a exposição das informações e/ou configuração excessiva de arquivos XML (*Extensible Markup Language*), dispondo de anotações para classes Java e componentes bem definidos para a camada de apresentação (FLAVINHA, 2008; FARLEY, 2008).

O principal benefício de sua arquitetura é o uso dos componentes EJB como *beans* acoplados direto nas páginas JSF (Figura 4). Basicamente EJB é uma arquitetura de componentes multi-plataforma para o desenvolvimento de aplicações Java, distribuídas, escaláveis e orientadas a objetos. Como o JBoss Seam fornece uma ponte entre o modelo de componentes do JSF e o modelo do EJB, isso permite que o EJB seja utilizado diretamente como um *managed bean* JSF, eliminando a necessidade de utilizar fachadas (façade) extras de *beans*, uma vez que é necessário apenas um único EJB. Dentro do contexto deste trabalho o uso de EJB tem a função de facilitar a componentização das funcionalidades.

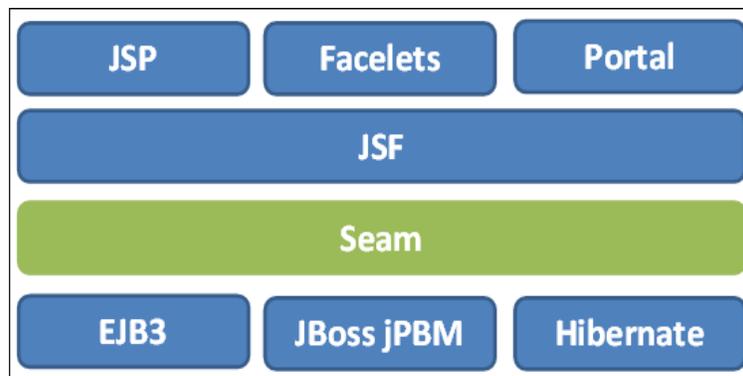


Figura 4: Arquitetura do sistema.

3.2 Métodos

A metodologia seguida para elaboração da arquitetura proposta foi a aplicação dos principais conceitos dos ERPs no contexto dos dados ambientais. Assim, foi criada uma base única com integração de todas as informações necessárias. Para o controle de qualidade dos dados foi feita uma modelagem para armazenamento das regras que controlam os da-

dos e para validação automática dos dados a partir dessas regras. Para isso, foi criado um interpretador que lê as regras e as aplica no conjunto de dados exportados.

Para a validação da arquitetura de dados foram definidos testes do sistema. Para Harold, Hierons e Danicic (2000) o teste é uma das atividades mais onerosas do processo de desenvolvimento. O teste consiste em uma atividade dinâmica que visa revelar defeitos do sistema desenvolvido e testar suas funcionalidades. O processo é composto principalmente por quatro etapas: planejamento de teste, projeto de caso de teste, execução e avaliação dos resultados dos teste (MYERS, 1979; BEIZER, 1990; MALDONADO, 1991; PRESSMAN, 2001). Após planejados e projetados os testes foram executados e seus resultados avaliados, conforme apresentado na Seção 4.7.

Para este trabalho foram planejados três testes:

- Cadastros Básicos do Sistema - são os cadastros necessários ao funcionamento dos outros módulos e inclui dados como organização, usuário e tipos de dados;
- Importação de Dados - foi testada a importação de um conjunto real de dados da torre micrometeorológica do PGFA localizada em um sítio experimental no município de Cuiabá denominado Fazenda Miranda;
- Exportação com validação de dados - A exportação foi testada com o mesmo conjunto de dados e com o cadastro e teste de uma validação de valores de temperatura, nesta validação foram testados os limites da temperatura, classificando-as como: Muito Alta, Muito Baixa, Normal;

3.3 Considerações Sobre o Capítulo

Este capítulo apresentou o material e metodologia adotada neste trabalho. O uso do PostgreSQL como sistema gerenciador de banco de dados nos permite ter uma ferramenta totalmente manipulável e robusta para manipulação de grandes quantidades de dados e usuários.

A modelagem das tabelas que compõem a arquitetura proposta se deu através do Diagrama de Classes da UML, essas Classes foram mapeadas para o modelo relacional através

do JPA. Dessa forma pode-se utilizar das vantagens do modelo orientado a objetos, como representação de classes e seus atributos e, o armazenamento dos dados no modelo relacional de dados. Neste trabalho optou-se pelo idioma inglês para nomenclatura das classes, atributos e os relacionamentos para que o mesmo possa se tornar um projeto *opensource* (código aberto) no qual colaboradores de diversas partes possam colaborar.

A utilização do Jboss Seam para implementação da interface com o usuário e construção da camada de manipulação dos dados, permite que através do acoplamento dos componentes EJBs diretamente nas páginas JSF, seja possível representar a integração da arquitetura dos ERPs na interface com o usuário. O JBoss Seam fornece ainda outras funcionalidades úteis, como a possibilidade de internacionalização da aplicação, o que consiste em se ter uma interface disponível em diversos idiomas.

O próximo capítulo apresenta a modelagem da arquitetura de dados proposta como resultado deste trabalho. São apresentados também testes com dados reais de uma estação micrometeorológica.

Capítulo 4

Resultados e Discussão

O contexto deste trabalho envolve principalmente a criação de um ambiente que permite o gerenciamento de dados científicos e, o enfoque principal é a aplicação direta aos dados de estudos do PGFA. Nesse sentido, este capítulo apresenta os resultados deste trabalho que envolvem a modelagem da arquitetura de dados proposta, bem como as consequências de uso dessa modelagem para solução dos aspectos de flexibilidade, concorrência, centralização e validação. As modelagens são apresentadas por meio de diagramas de classes e suas descrições. Além das modelagens é apresentado também o estudo de caso com os dados de uma estação micrometeorológica.

4.1 Definição da Arquitetura

Para alcançar o objetivo de desenvolver um ambiente que possua as funcionalidades de uma plataforma ERP aplicados no contexto de dados científicos, a arquitetura do sistema foi dividida em grupos de funcionalidades. Essas funcionalidades, conforme ilustrado na Figura 5, foram divididas em três grupos que se relacionam com o armazenamento dos dados dos sensores, com as informações gerenciais e com os metadados para composição das ontologias.

No grupo *Dados*, tem-se o controle dos dados captados pelos sensores, o que envolve o armazenamento dos valores das medidas obtidas pelos sensores, das informações dos



Figura 5: Arquitetura dos dados.

equipamentos que realizaram a captação e, das informações que compõem esses dados como: Tipo, Local de medição, Composição que descreve valores derivados um do outro e ainda as Restrições que controlam a qualidade e validade dos dados.

No grupo *Gerenciais* são armazenados os dados relacionados com as informações gerenciais como: Organização, Projetos de Pesquisa, Usuário e Permissões.

No grupo *Ontologias* ocorrem os controles das ontologias, que com as definições armazenadas nos metadados acessam os outros dois grupos e geram as visões. Os metadados são informações que definem a ligação entre as informações gerenciais e dos dados formando os contextos com que os usuários visualizam e processam os dados, por exemplo a configuração dos *labels* (descrição dos campos) que devem aparecer na tela para cada tipo de dado e em cada contexto.

As próximas seções detalham cada grupo da arquitetura apresentada.

4.2 Dados Brutos

O grupo dos Dados Brutos se relaciona ao armazenamento dos dados medidos pelos sensores e envolve um conjunto de tabelas que possuem toda informação necessária para o

entendimento e avaliação dos valores captados pelos sensores. A modelagem é mostrada na Figura 6, onde se pode observar 7 (sete) tabelas.

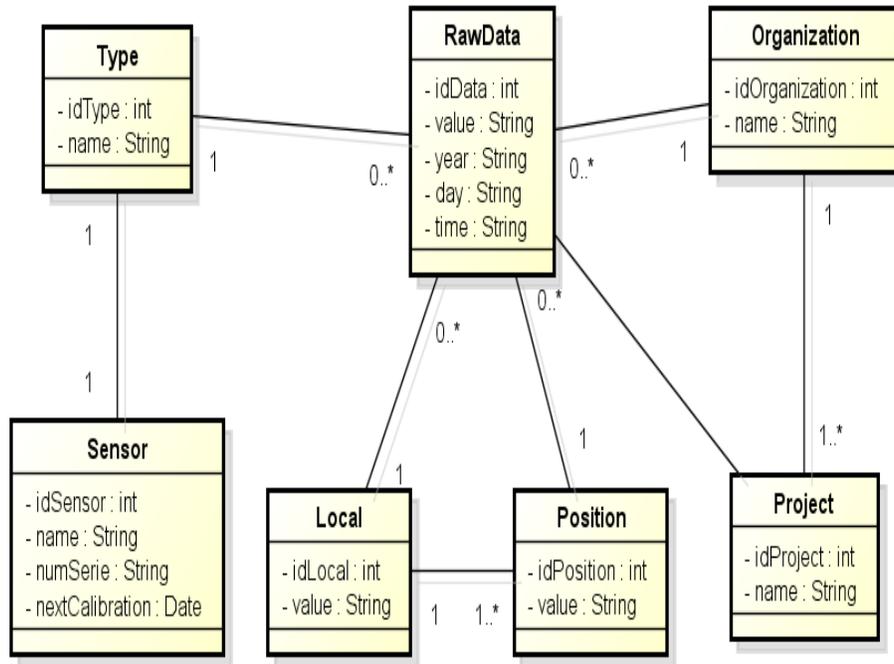


Figura 6: Modelagem que contempla o armazenamento dos dados brutos.

Na tabela *RawData* são armazenados os valores medidos (*value*), o momento da medição (*year*, *day* e *time*), que juntos definem a identificação temporal do valor medido, o que é útil em aplicações onde dados são coletados e analisados em função do tempo. É importante ressaltar que o atributo *value* é definido como *string*, o que permite que qualquer valor possa ser armazenado feita sua conversão para esse tipo. As associações da tabela *Data* com as tabelas *Organization* e *Project* definem respectivamente, a qual organização e projeto está associado determinado valor.

A tabela *Local* armazena os locais (ou os sítios experimentais) onde são coletados dados. Um mesmo local pode haver vários equipamentos realizando as medições, sendo que cada um é armazenado na tabela *Position*, dessa forma, cada local pode ter subdivisões que fornecem maiores detalhes na identificação da medida.

Na tabela *Type* são armazenados os tipos de dados medidos, que definem um tipo

específico de dado e pode estar associado a um sensor, por exemplo o tipo Temperatura está associado ao Termohigrometro .

As tabelas *Organization* e *Project* são identificações da instituição a qual pertence a medida e ao projeto (ou subdivisão) dessa instituição. Essas informações tornam-se úteis principalmente para o controle de acesso e compartilhamento dos dados como é mostrado na Seção4.4.1.

4.3 Restrições e Composições

As restrições são regras referentes ao valor de um determinado tipo de variável. Essas regras podem definir um controle de qualidade para interpretação de um fenômeno ou apenas os limites aceitáveis de valores. Por exemplo, poder-se-ia definir valores máximos e mínimos de temperatura do ar em °C para a cidade de Cuiabá. Dessa forma, uma restrição pode ser aplicada isoladamente ou demandar a aplicação de outras restrições (composição).

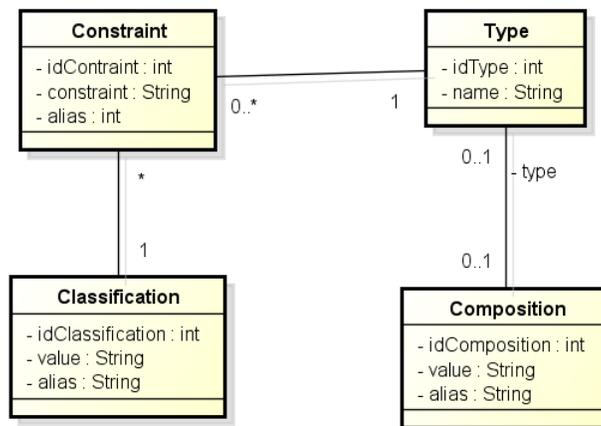


Figura 7: Modelagem das restrições.

A modelagem das restrições é mostrada na Figura 7, a tabela *Constraint* guarda a expressão matemática ou lógica que define a restrição. A execução da expressão é feita por um interpretador que retorna um valor, caso o valor da expressão seja considerado positivo para a regra definida, então o valor é marcado com uma classificação verdadeira/positiva

(*idClassificationTrue*) caso contrário ele é marcado com a classificação falsa/negativa (*idClassificationFalse*).

A Figura 7 mostra ainda a tabela *Composition*, que define as composições, que são valores calculados a partir de outras variáveis. A coluna *value* guarda a expressão que calcula o valor da variável, o cálculo é feito pelo mesmo interpretador responsável pela execução das restrições. Como exemplos de Composições pode-se tomar os valores de fluxos de energia, que geralmente são calculados por modelos matemáticos baseados em outras variáveis medidas, como Saldo de Radiação, Temperatura, Umidade do Ar e Velocidade do Vento.

4.4 Informações Gerenciais

Informações Gerenciais são todas as informações referentes ao controle dos dados. Essas informações permitem gerenciar o acesso aos dados controlando as permissões por organizações, projetos e usuário. O objetivo dessa parte da arquitetura é ainda gerenciar as informações dos sensores, para que possam ser utilizadas para interpretação de uma possível falha ou garantia de qualidade dos valores medidos, através do histórico de calibrações e pessoa responsável pela manutenção do equipamento.

4.4.1 Usuários e Permissões

Os usuários são pessoas que acessam o sistema para incluir ou requisitar informações. A Figura 8 mostra a modelagem do cadastro de usuários e o controle de acesso aos dados. O controle é feito através de listas de acessos a organizações, projetos e aos tipos de dados.

Cada usuário tem acesso a uma ou mais organizações assim como a projetos. A tabela *Permission*, controla o acesso aos dados, este acesso pode ser controlado por usuário, organização ou projeto.

O controle é feito para cada tipo de dado (*idType*) e para um determinado período (*startTime* e *endTime*).

O controle de permissões feito através de usuários, organização e projeto permite o compartilhamento de dados entre pessoas de projetos e organizações diferentes, podendo

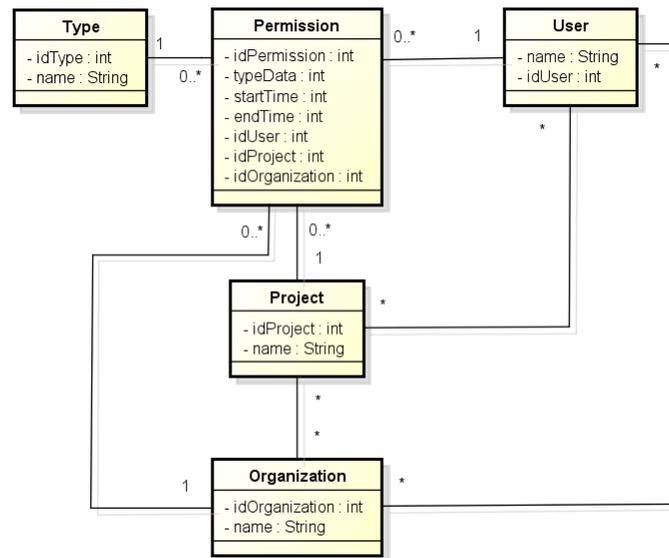


Figura 8: Modelagem das regras de acesso aos dados.

ser dada permissão para o usuário de apenas um conjunto de dados através dos atributos *startTime* e *endTime*.

4.4.2 Gerenciamento de Sensores

A modelagem do gerenciamento dos sensores foi feita com o objetivo de monitorar a qualidade dos dados, pois controlando o comportamento de cada sensor é possível identificar possíveis falhas de leitura, essa modelagem é mostrada na Figura 9.

A tabela *Sensors* organiza as informações de cada sensor e pode estar relacionada a vários tipos de dados, pois existem sensores que medem diferentes tipos de variáveis. A tabela *Calibration* guarda as datas das calibrações feitas em cada sensor o que permite manter o histórico de calibrações e essa informação ser utilizada para interpretação e validação de dados.

Essa tabela guarda também a informação de qual usuário do sistema, pertencente a uma organização (e um projeto) fez a calibração do equipamento. A tabela *Manufacturer* armazena informações do fabricante do sensor.

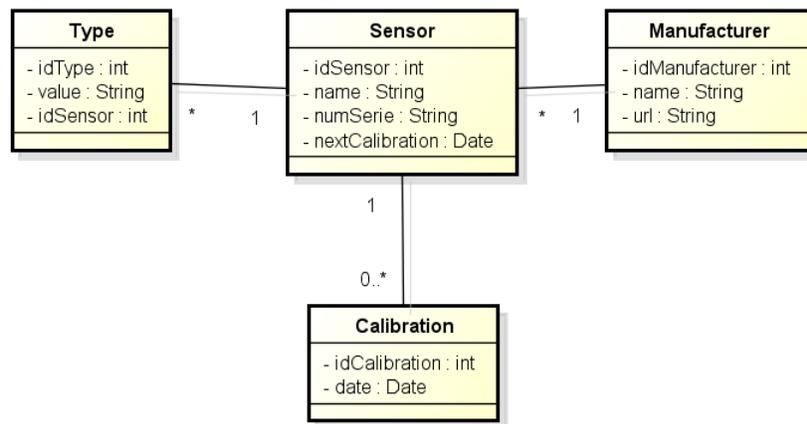


Figura 9: Modelagem do gerenciamento dos sensores.

4.5 Lógica da Validação e Composição de Dados

Para que os dados possam ser validados, é preciso estabelecer que tipos de validações são importantes para o contexto deste trabalho, bem como o momento em que as validações são ativadas. Os tipos de validação envolvem determinar as propriedades do domínio dos dados, o que tradicionalmente pode ser implementado em um banco de dados utilizando gatilhos e procedimentos armazenados. Contudo esse tipo de abordagem envolve a implementação da validação para cada novo dado, o que significa a necessidade de conhecimento de programação por parte do usuário.

Para permitir que as validações sejam inseridas e aplicadas sem a necessidade de criação de novos gatilhos e procedimentos, a validação e composição de dados foi implementada utilizando um interpretador próprio, que recebe uma expressão matemática ou lógica e retorna seu valor.

Dessa forma, uma validação é inserida no banco de dados como uma *string* que referencia os campos, valores e condições a serem obedecidos. Foi definida uma sintaxe para as expressões matemáticas e lógicas para funcionamento do interpretador. Os seguintes operadores são permitidos:

- + : Adição;
- - : Subtração;

- / : Divisão Real;
- * : Multiplicação.
- > : Maior;
- < : Menor;
- >= : Maior ou igual;
- => : Menor ou igual;
- <> : Diferente.

Além dos operadores aritméticos e condicionais, foram definidos também operadores especiais, para controle de fluxo e delimitadores, a saber:

- (): Delimitadores;
- " : Delimitador de *string*.
- ; : Controle de fluxo para operações lógicas. Ex: (SE(4>2;"verdadeiro";"falso"))

4.5.1 Funcionamento da Validação

A validação dos dados consiste em executar uma operação que se baseia em alguma condição e então classificar os dados de acordo com esta condição. O interpretador recebe uma *string* contendo a expressão que valida um único valor. A Figura 10 mostra como é o fluxo interno de validação dos dados.

O processo de validação é executado no momento de exportação de um conjunto de dados, primeiramente é verificado se o tipo de dado exportado contém uma validação cadastrada, se sim, então o processo ilustrado na Figura 10 é acionado. Para cada valor do conjunto de dados requisitado o processo é executado.

No início do processo de validação a aplicação ativa o interpretador que verifica quais campos estão presentes na condição e busca esses valores para comparação. Após buscar os valores o interpretador executa a operação e retorna o valor da comparação, se esse

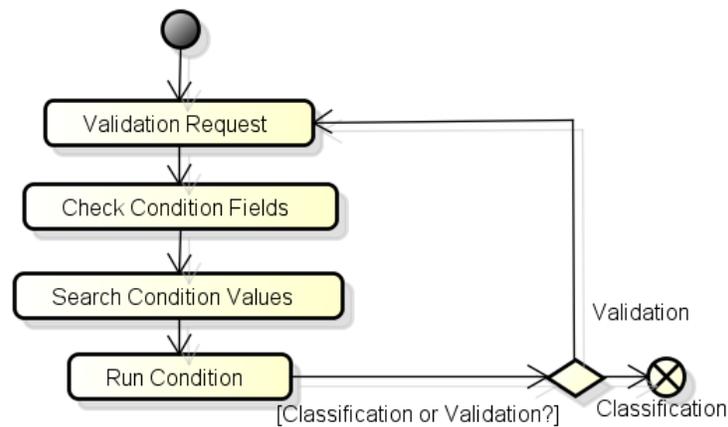


Figura 10: Diagrama do fluxo de validação dos dados.

valor for uma nova comparação então o processo é reiniciado recursivamente, caso seja uma classificação, então o valor da classificação é escrito no arquivo de exportação dos dados.

4.5.2 Composição

A composição é a descrição de um valor que é calculado a partir de outros valores já existentes. Portanto, é o resultado de uma expressão matemática. O mesmo interpretador responsável pela validação, é quem executa o cálculo das composições, mudando apenas o fluxo de execução, como é mostrado na Figura 11.

4.6 Ontologias e Visões

Apesar da complexidade de toda a arquitetura de dados definida, um dos objetivos a serem alcançados é a de facilitar a manipulação dos dados pelo usuário comum. Nesse sentido, optou-se pelo encapsulamento de toda essa complexidade através do uso de ontologias e visões.

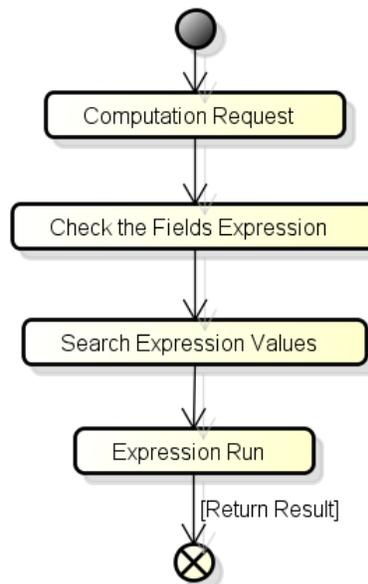


Figura 11: Diagrama do fluxo composição de dados.

4.6.1 Ontologias

As ontologias são utilizadas com o intuito de garantir a flexibilidade e esconder a complexidade das informações, assim uma ontologia realiza o agrupamento dos metadados para montagem do contexto e, é esse contexto que o usuário final vai manipular os dados. Assim, conforme mostrado na Figura 12, as ontologias servem para definir quais *labels* serão utilizados para cada tipo de dado de acordo com as ontologias pertencentes a cada projeto. É possível ainda associar as regras de validação do tipo de dado à ontologia, o que permite por exemplo, que usuários diferentes usem o mesmo tipo de dado com validações distintas.

4.6.2 Visões

Uma visão é uma tabela única derivada de outra tabela, que pode ser uma tabela básica ou uma visão previamente definida. Uma visão não existe de forma física, ela é considerada uma tabela virtual, em contraste com as tabelas básicas, cujas tuplas são realmente armazenadas no banco de dados (ELMASRI; NAVATHE, 2005; HEUSER, 1998).

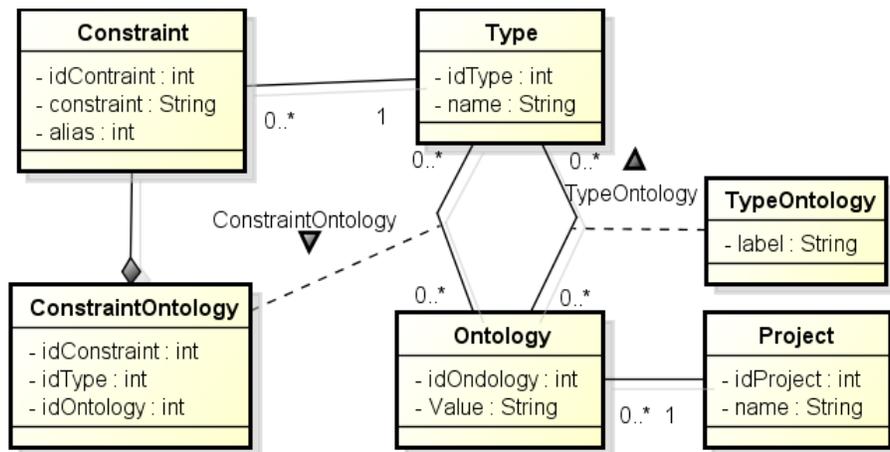


Figura 12: Modelagem das ontologias.

Com o intuito de melhorar o desempenho na execução das consultas são criadas visões materializadas que particionam os dados baseado no projeto ao qual os dados estão vinculados. Assim, para cada novo projeto criado, o sistema avalia o conjunto de visões a partir da ontologia à qual o projeto está vinculado e gera as visões para esses dados.

As visões foram utilizadas para definir tabelas personalizadas para o usuário. A tabela onde foram armazenados os dados brutos armazenam os dados de todos os tipos de sensores, assim, uma linha da tabela contém dados de apenas um tipo de variável. Dessa forma as visões foram utilizadas para fornecer visualização personalizada para o usuário, visões foram geradas para criar tabelas virtuais, onde para cada linha fosse possível visualizar dados de diferentes variáveis. As Tabelas 1 e 2 mostram valores na tabela onde são armazenados os dados e uma visão respectivamente.

Tabela 1: Exemplo de tabela de dados brutos.

Ano	Dia	Hora	Tipo	Valor
2011	121	12:00	Temperatura	35
2011	121	12:00	Umidade	63
2011	121	12:00	Vento	5
2011	121	13:00	Temperatura	36
2011	121	13:00	Umidade	60
2011	121	13:00	Vento	4

Tabela 2: Exemplo de visão.

Ano	Dia	Hora	Temperatura	Umidade	Vento
2011	121	12:00	35	63	5
2011	121	13:00	36	60	4

4.7 Estudo de Caso

Para validar a arquitetura de dados proposta, foram feitos testes referentes a 4 (quatro) principais funcionalidades, a saber: Centralização dos Dados, Controle de Acesso, Rastreabilidade das Informações e Validação dos Dados. O funcionamento do ambiente prevê o cadastramento das informações básicas do contexto, com o cadastramento dos sensores, das informações gerenciais e a definição da ontologia a ser trabalhada. Posteriormente os dados podem ser importados e manipulados.

4.7.1 Centralização dos Dados

A centralização dos dados foi modelada como apresentada na seção 4.2. Todos os dados medidos por sensores são armazenados na tabela *Data* e esta contém ligações com outras informações referentes ao valor medido. A tela de alimentação da tabela *Data* é mostrada na Figura 13, nessa tela é feito o *upload* de um arquivo texto contendo os dados coletados.

A primeira informação a ser configurada nessa tela é o local de onde vieram os dados, através do campo *Station*, posteriormente deve-se selecionar o delimitador dos campos no arquivo, através do campo *Delimiter*. Abaixo desses dois campos há o campo para seleção do arquivo, quando o arquivo é selecionado é preenchida a área *First Row* onde é possível ver a primeira linha do arquivo selecionado para que o usuário possa configurar a importação. A tabela toda mostra apenas informações da primeira linha do arquivo, cada coluna do arquivo é então representada em uma linha da tabela *First Row* para configuração individual.

A tabela *First Row* contém 3 (três) colunas, na primeira há um *checkbox* no qual é possível configurar se a coluna será ignorada ou não. A segunda coluna contém o va-

The screenshot shows a 'Data Import' window with the following settings:

- Settings:**
 - Estation:** Fazenda Miranda
 - Delimiter:** ,
 - File:** TesteImportacao.csv
- First Row:**

Import? Yes or No	Valor	Height/Type
<input checked="" type="checkbox"/>	121	Julian Day
<input checked="" type="checkbox"/>	00:00:00	Hours
<input checked="" type="checkbox"/>	-50.63	7m NET
<input checked="" type="checkbox"/>	-47.67	7m GMedio
<input checked="" type="checkbox"/>	20.62	2m Temp1
<input checked="" type="checkbox"/>	95.56	7m Temp2
<input checked="" type="checkbox"/>	23.86	7m Umid2
<input type="checkbox"/>	76.90	
<input type="checkbox"/>	103	

Figura 13: Tela de Importação de Dados.

lor medido, na terceira coluna (*Height/Type*) é possível configurar a altura da medição (primeiro campo) e o tipo de dado (segundo campo).

4.7.2 Controle de Acesso

O controle de acesso aos dados faz parte do módulo de informações gerenciais da arquitetura (parte b da Figura 5) e foi modelado como mostrado na seção 4.4.1

O controle de acesso aos dados parte primeiramente do cadastro das organizações (Figura 14). Logo após podem ser cadastrados projetos e usuários ligados a organização. Com organizações, projetos e usuários cadastrados, pode-se gerar então listas de permissões de acessos aos dados, essas listas podem ser baseadas apenas no projeto ou no usuário. Dessa forma, um usuário que não faz parte de uma determinada organização que detém os direitos de acesso aos dados, pode receber permissão de acesso aos dados, desde que configurada por algum usuário que tenha tal permissão.

O diagrama de atividades mostrado na Figura 15 mostra como acontece o processo de controle de acesso aos dados.

Organization Details		
Idorganization	2	
Name	Pós-Graduação em Física Ambiental - PGFA	
<input type="button" value="Edit"/> <input type="button" value="Done"/>		
Projects		
Idproject	Name	Action
1	FAPEMAT	Select
<input type="button" value="Add project"/>		

Figura 14: Tela de Cadastro de Organizações.

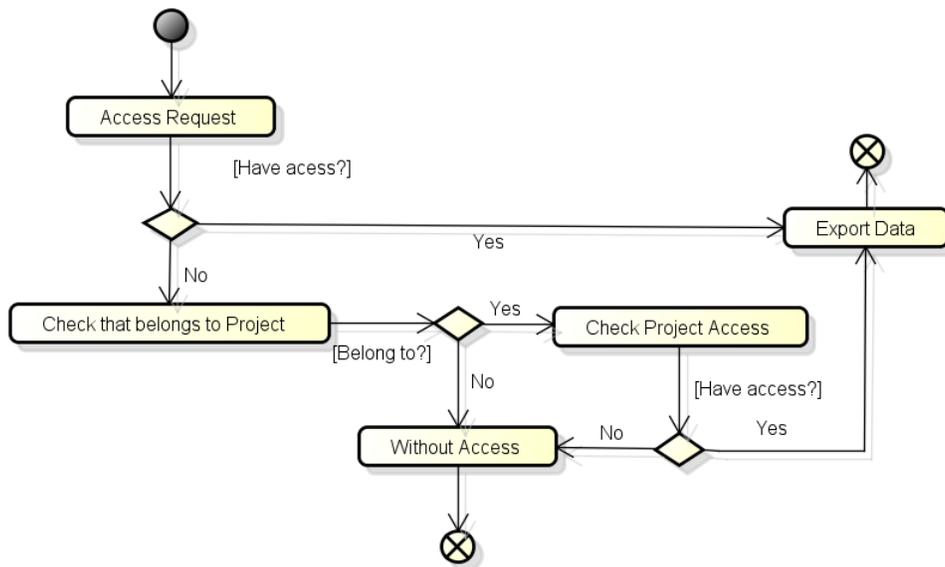


Figura 15: Diagrama de Atividades do fluxo de controle de acesso aos dados.

Quando é feita a requisição de acesso aos dados primeiro é verificado se o usuário tem permissão de acesso a aquele conjunto de dados requisitados, se sim, ele pode ter acesso. Caso o usuário não tenha permissão de usuário, é então verificado se ele faz parte de algum projeto. Se o usuário não pertencer a nenhum projeto então ele não pode acessar os dados. Pertencendo a algum projeto, verifica-se então se o projeto tem acesso aos dados, caso tenha, o usuário então tem acesso e pode exportar os dados solicitados.

4.7.3 Rastreabilidade de Informações

A rastreabilidade das informações é extremamente útil para controle de qualidade dos dados. A validação dessa funcionalidade pode ser observada em todas as telas do sistema. A Figura 16 por exemplo, mostra detalhes da tela *Data* onde é possível ver todas as informações referentes a um valor medido, na parte inferior da figura pode-se observar 6 (seis) abas onde é possível acessar todas as informações. Essa característica é presente em todas as telas, o que possibilita o rastreamento rápido de informações para tomada de decisões.



The screenshot shows a web interface titled "Data Details". It displays a data record with the following fields:

Date	
Day j	121
Time	18:00:00
Value	-57.46
Year	2008

Below the data record are two buttons: "Edit" and "Done". Underneath these are six tabs: "Local", "Object", "Organization", "Project", "Sensor", and "Type". At the bottom, there is a table with the following data:

Id local	Name	Organization idorganization	Action
1	Fazenda Miranda	1	View

Figura 16: Informações detalhadas na tela Dados.

Clicando na aba *Sensor*, por exemplo, é possível acessar a tela de informações do sensor (Figura 17) que coletou essa informação. Com isso, informações úteis como, por exemplo, as datas de calibração do sensor podem ser acessadas.

Todas as telas do sistema são geradas a partir do mapeamento das tabelas no banco

de dados e seus relacionamentos, assim, em qualquer tela do sistema é possível acessar informações relacionadas através das abas que ficam na parte inferior da tela.

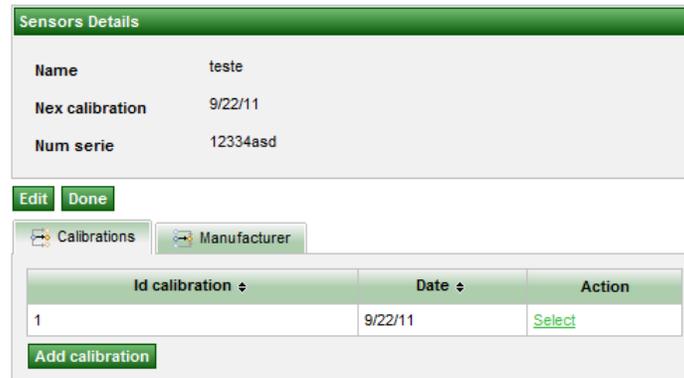


Figura 17: Informações detalhadas na tela Sensors.

4.7.4 Execução da Validação

A validação dos dados foi feita utilizando um interpretador, que recebe uma *string* contendo uma expressão lógica e retorna o seu resultado. Para execução correta da expressão, algumas regras devem ser seguidas, as tabelas *Type*, *Classification* e *Constraint* têm um campo chamado *Alias*, onde deve ser armazenado um nome para cada registro nessas tabelas. O campo *Alias* é utilizado pelo interpretador para decidir se busca um valor de um tipo de dado, uma classificação ou uma outra restrição (*constraint*). A possibilidade de execução de uma restrição chamada por outra através do campo *alias*, permite a simulação de novos operadores baseados nos 4 (quatro) operadores definidos na sintaxe do interpretador. Para que isso seja possível, foi definido um padrão de nomenclatura dos *alias*, no qual todos devem ter o prefixo que identifica a tabela correspondente, são eles:

- *T_* : *Type*;
- *C_* : *Classification*;
- *CT_* : *Constraint*.

A Figura 18 mostra o exemplo de cadastro de uma regra hipotética de validação para definir se a temperatura é normal, muito alta ou muito baixa.

Edit Constraint

Alias *

Expression

* required fields

Save
Delete
Cancel

Type

Id type	Alias	Udm	Value
1	T_Temp	Celcius	Temp

Change type

Classification Search Results (3)

Alias	Value	Action
C_TMax	Muito Alta	Select
C_TMin	Muito Baixa	Select
C_TNormal	Normal	Select

Constraint Search Results (2)

Alias	Expression	Type	Action
CT_LimitT	SE(T_Temp>49;C_TMax;CT_LimitMinT)	Temp	Select
CT_LimitMinT	SE(T_Temp<10;C_Tmin;C_Normal)	Temp	Select

Figura 18: Tela de criação das regras de validação de dados.

Para a validação do exemplo mostrado, uma temperatura é "Muito Alta" se o seu valor é maior que 49°, "Muito Baixa" se for menor que 10° e normal estiver entre 10° e 49°.

Pode-se observar na Figura 7 duas áreas na parte inferior, a primeira, lista as classificações disponíveis para serem utilizadas, a segunda mostra outras regras cadastradas para serem utilizadas em conjunto, como é mostrado no exemplo. O cadastro ilustrado é o da regra com *alias* *CT_LimitT*, a expressão é: $SE(T_Temp > 49; C_TMax; CT_LimitMinT)$, essa regra utiliza o tipo temperatura (*T_Temp*), e caso a comparação seja verdadeira o valor é classificado como "Muito Alto" (*C_Tmax*), senão, é chamado outra regra de validação, *CT_LimitMinT*, que tem a expressão: $SE(T_Temp < 10; C_Tmin; C_Normal)$.

Essa última compara se a temperatura é menor que 10° , ($T_Temp < 10$), caso seja verdadeiro, o valor é classificado como "Muito Baixo" (C_Tmin), senão, é classificado como "Normal" ($C_TNormal$). A árvore de execução gerada pela chamada recursiva do método responsável pela execução da expressão é mostrada na Figura 19.

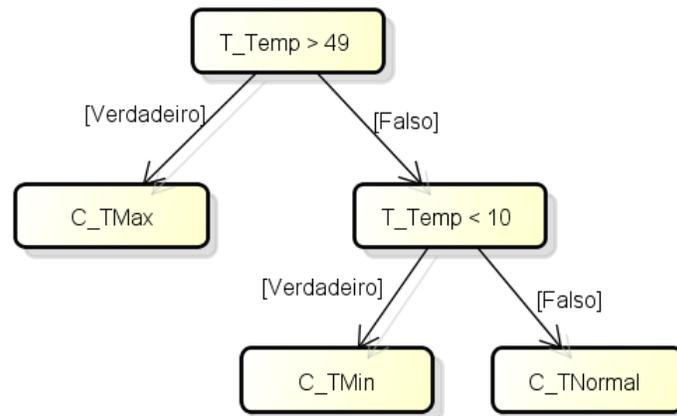


Figura 19: Árvore de execução gerada na validação dos dados de temperatura.

4.7.5 Acesso as visões e exportação

Depois de criada as visões, uma janela (Figura 20) é gerada para acesso a essa visão. Essa janela contém na parte superior as configurações de pesquisa, para visualização dos dados por ano, *InitialDay* (Dia Inicial) e *LastDay* (Último dia).

Na área central da janela são visualizados os dados de acordo com a configuração da visão. No caso da Figura 20 essa visão foi configurada com o nome da estação micrometeorológica do teste mostrado na Seção 4.7.1 Fazenda Miranda. No referido teste, foram importados valores de Saldo de Radiação (*Net*), Fluxo Médio de Calor no Solo (*Gmedio*), Temperatura 1 (*Temp1*), Temperatura 2 (*Temp2*), Umidade 1 (*Umid2*) e Umidade 2. Na criação da visão foram selecionados todos os valores menos a *Temp2* e *Umid2*.

Na parte inferior da janela pode-se observar o botão *Export this data set*, que permite exportar este conjunto de dados para um documento texto. Quando acionado esse botão, é feita a verificação se existe regras de validação para os tipos de dados contidos na visão,

Dados Fazenda Miranda Search Filter

Year:

Initial Day:

Last Day:

Match: All Any

Search **Reset**

Dados Fazenda Miranda Search Results (33451)

Year	Day	Time	Gmedio	Net	Temp	Umid
2009	273	10:00:00	236.84	568.31	21.9	57.2
2009	273	10:30:00	252.15	611.27	22.87	54.65
2009	273	11:00:00	269.6	650.7	23.82	52.4
2009	273	11:30:00	278.94	685.21	24.8	50.31
2009	273	12:00:00	263.42	695.07	25.42	49.26

[Next Page](#) [Last Page](#)

Export this data set

Figura 20: Tela que exibe os dados de uma visão.

se sim, é lançado um questionamento ao usuário se deseja ou não utilizar as validações (Figura 21), caso o usuário escolha a opção *Yes* então o processo descrito na Seção 4.5.1 é executado, caso contrário a exportação segue sem validação nenhuma.

Confirmation

This data set contains associated validation rules. Would you like to use them?

Yes **No**

Figura 21: Tela que exibe o processo de exportação dos dados de uma visão.

4.8 Considerações Sobre o Capítulo

Este capítulo apresentou os resultados alcançados neste trabalho. Foi apresentada a visão geral da arquitetura de dados proposta bem como seu detalhamento através dos diagramas de classe. O estudo de caso com utilização de dados reais mostrou a aplicabilidade da arquitetura proposta.

No capítulo seguinte são apresentadas as conclusões a respeito do trabalho desenvolvido e as sugestões de trabalhos futuros.

Capítulo 5

Conclusão

Este trabalho apresentou a modelagem de uma arquitetura de dados para gerenciamento de dados científicos cujo enfoque principal é manipular dados reais, obtidos de sensores micrometeorológicos, envolvidos no contexto Programa de Pós-Graduação em Física Ambiental.

A modelagem do armazenamento desse tipo de dado é não trivial devido a complexidade inerente ao tipo de dado, seu comportamento e forma de manipulação, que envolve muitos usuários e vários domínios de aplicação diferentes. Assim, essa modelagem busca solucionar os problemas comumente encontrados na manipulação de dados em contextos de projetos de pesquisa, ou seja, o uso compartilhado dos dados por uma equipe de especialistas que buscam gerar novos conhecimentos com o entendimento do comportamento dos dados.

A validação da arquitetura foi feita com a implementação de uma interface com o usuário baseada no ambiente web, de forma a mostrar que o armazenamento dos dados brutos e suas informações auxiliares permitem embutir nos dados as regras e validações relacionadas com cada contexto da pesquisa na qual a aplicação científica está relacionada.

Em relação aos aspectos de centralização e integração das informações em uma única base de dados, este trabalho permite que os dados sejam analisados de forma mais adequada, pois o tratamento é uniformizado para todas as manipulações permitindo que seja feito melhor controle de qualidade dos resultados das análises. Um aspecto importante na uniformização do tratamento dos dados é que o controle de acesso dos usuários per-

mite que os dados sejam compartilhados sem que ocorra problemas de sigilo e alterações inadequadas.

A arquitetura de armazenamento das regras de validação e o interpretador de expressões lógicas e matemáticas possibilitam a verificação automatizada dos dados, a partir do cadastro das regras de verificação de cada tipo de dado. A verificação automatizada permite que o conhecimento gerado pelas modelagens computacionais possam ser embutidas no processamento dos dados e os mesmos serem validados automaticamente, ou seja, essa funcionalidade pode trazer grandes benefícios do ponto de vista de qualidade na interpretação de fenômenos, visto que essas validações podem por exemplo representar regras físicas de comportamento do fenômeno.

Os resultados mostram ainda que o modelo relacional de dados pode ser utilizado para armazenamento e recuperação de dados científicos, o que confirma a hipótese levantada por Pfaltz (2007) de que os bancos de dados relacionais podem parecer bastante adequados para o contexto científico, pois as linhas podem denotar fenômenos ou objetos, e as colunas podem denotar seus atributos. No entanto, isso foi possível devido a arquitetura proposta, na qual utilizou dos conceitos dos ERPs, onde a integração das informações permite que os dados sejam interpretados desde a visão pontual do seu valor até a visualização de informações que compõem o contexto desse dado.

A modelagem a partir dos conceitos dos ERPs permite também que as técnicas de KDD possam ser aplicadas nas análises dos dados, o que tem especial importância no contexto dos dados científicos, pois a quantidade de dados aumenta em ordens de grandeza a cada ano.

De forma geral, através do estudo de caso, percebe-se que arquitetura alcançou os objetivos propostos e gera grandes benefícios ao PGFA e, a comunidade científica em geral. Além disso, as funcionalidades podem ser estendidas para que mais domínios científicos possam utilizá-la, como no caso de pesquisas envolvendo dados de desempenho de atletas.

5.1 Trabalhos Futuros

A arquitetura de dados gerada neste trabalho torna o uso das ferramentas computacionais de banco de dados mais efetiva no ambiente científico. Contudo para que o ambiente En-

vRP seja disponibilizado com todas as suas funcionalidades, vários aspectos novos devem ser adicionados ao ambiente. Nesse contexto, as sugestões de direcionamento de ações e de trabalhos futuros são destacados a seguir.

Um aspecto importante a ser adicionado ao ambiente é o versionamento dos dados. Dessa forma, a base de dados passa a suportar além dos dados brutos, os dados trabalhados e modelados pelos pesquisadores, com controle sobre as alterações e resultados gerados por cada manipulação.

Além do versionamento, a replicação de um conjunto de dados dentro da mesma base, desde que parametrizada por projeto ou usuário, permite que diferentes pesquisadores trabalhem na mesma base de dados com diferentes perspectivas ou formas de interpretação sem que os dados de um interfiram no outro.

A manipulação de grande quantidade de dados é suportada pelo SGBD escolhido para validar a proposta. Contudo no contexto do PGFA, é necessário incluir no ambiente funcionalidades de processamento paralelo e em *cluster*, pois as modelagens computacionais realizadas pelos pesquisadores têm suas validações normalmente baseadas em processamentos desse tipo. Com isso, é necessário estender as funcionalidades do ambiente de forma a contemplar a aplicação de processamento paralelo e distribuído.

As funcionalidades relacionadas com o processo de KDD devem ser implementadas de forma a possibilitar que novos algoritmos e formas de visualização sejam disponibilizadas ao pesquisador.

Em relação ao ambiente EnvRP como um todo, é preciso construir uma interface com o usuário que contemple todas as funcionalidades e facilite a interação de diversos perfis de usuários. O que envolve a criação de interfaces específicas para o usuário administrador do sistema, o usuário criador de ontologias e o usuário analista.

Referências Bibliográficas

ADALI, S.; SAPINO, M. L.; SUBRAHMANIAN, V. S. A multimedia presentation algebra. In: *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 1999. (SIGMOD '99), p. 121–132. ISBN 1-58113-084-8. Disponível em: <<http://doi.acm.org/10.1145/304182.304193>>. 12

ADALI, S.; SAPINO, M. L.; SUBRAHMANIAN, V. S. An algebra for creating and querying multimedia presentations. *Multimedia Syst.*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 8, p. 212–230, October 2000. ISSN 0942-4962. Disponível em: <<http://dl.acm.org/citation.cfm?id=782567.782579>>. 12

ADEMPIERE. *Página Oficial do Adempiere ERP*. 2011. Disponível em: <<http://www.adempiere.org/>>. 20

ANDRADE, A. C. *Sistema de Informação para Gerenciamento Financeiro para a Pró-Reitoria de Ensino de Pós-Graduação da UFMT*. Monografia (Bacharelado em Ciência da Computação) — Universidade Federal de Mato Grosso, 2010. 19

ANTUNES, C. M.; OLIVEIRA, A. L. Temporal data mining: an overview. In: *Workshop on Temporal Data Mining*. [S.l.: s.n.], 2001. 12

ATNAFU, S.; BRUNIE, L.; KOSCH, H. Similarity-based algebra for multimedia database systems. In: *Proceedings of the 12th Australasian database conference*. Washington, DC, USA: IEEE Computer Society, 2001. (ADC '01), p. 115–122. ISBN 0-7695-0966-5. Disponível em: <<http://dl.acm.org/citation.cfm?id=545538.545552>>. 12

- BEIZER, B. *Software testing techniques*. 2. ed. New Yourk: Van Nostrand Reinhold Company, 1990. 27
- BENIOFF, M. R.; LAZOWSKA, E. D. *Computational Science: Ensuring America's Competitiveness*. [S.l.], Junho 2005. Disponível em: <http://www.nitrd.gov/pitac/reports-20050609_computational/computational.pdf>. 3
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *The Unified Modelling Language User Guide*. [S.l.]: Addison-Wesley Professional, 1999. ISBN 0-201-57168-4. 25
- BOSE, R.; FREW, J. Lineage retrieval for scientific data processing: a survey. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 37, p. 1–28, March 2005. ISSN 0360-0300. Disponível em: <<http://doi.acm.org/10.1145/1057977.1057978>>. 17
- BUNEMAN, P.; KHANNA, S.; TAJIMA, K.; TAN, W.-C. Archiving scientific data. *ACM Trans. Database Syst.*, ACM, New York, NY, USA, v. 29, p. 2–42, March 2004. ISSN 0362-5915. Disponível em: <<http://doi.acm.org/10.1145/974750.974752>>. 1
- CALARGE, F. A.; LIMA, P. C. Da abordagem do tqm (total quality management) ao gqm (global quality management): a inserção e utilização da metodologia do projeto axiomático no desenvolvimento de modelos de gestão sistêmica da qualidade. *Gestão e Produção*, scielo, v. 8, p. 196 – 213, 08 2001. ISSN 0104-530X. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttextpid=S0104-530X2001000200007nrm=iso>. 20
- CHAUDHRI, A. B.; ZICARI, R. *Succeeding with Object Databases: A Practical Look at Today's Implementations with Java and XML*. [S.l.]: Willey, 2001. 23
- CODD, E. F. A relational model of data for large shared data banks. *Commun. ACM*, ACM, New York, NY, USA, v. 13, p. 377–387, June 1970. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/362384.362685>>. 14
- COSTA, C. A. A aplicação da linguagem de modelagem unificada (uml) para o suporte ao projeto de sistemas computacionais dentro de um modelo de referência. *Gestão e Produção*, v. 8, n. 1, p. 19–36, abril 2001. 25

- CURADO, L. F. A.; RODRIGUES, T. R.; NOVAIS, J. W. Z.; OLIVEIRA, A. G.; VENTURA, T. M.; MUSIS, C. R. D.; NOGUEIRA, J. S. Adjustment of the brunt's equation parameters for the northern brazilian pantanal. *Journal of Ecology and The Natural Environment*, v. 3, p. 157–162, 2011. 24
- CURRIE, W.; GALLIERS, B. *Rethinking Management Information Systems*. [S.l.]: Oxford, 1999. 19
- DATE, J. C. *Introdução a Sistemas de Bancos de Dados, tradução ao da 7a. edição americana*. [S.l.: s.n.], 2000. 13
- EDELWEISS, N.; OLIVEIRA, J. P. M. de. *Modelagem de aspectos temporais de sistemas de informação*. Recife: Livro Texto da Escola de Computação, p. 162, 1994. 17
- ELMASRI, R.; NAVATHE, S. B. *Fundamentals of database systems 4th ed.* [S.l.]: Pearson Addison Wesley, 2003. 14
- ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados*. São Paulo: Pearson Addison Wesley, 2005. 38
- ERWIG, M.; GIITING, R. H.; SCHNEIDER, M.; VAZIRGIANNIS, M. Abstract and discrete modeling of spatio-temporal data types. *ACM GIS 98*, 1998. 18
- FARLEY, J. *Projetos práticos com Jboss Seam*. 1. ed. [S.l.]: Ciência Moderna, 2008. 26
- FAYYAD, U.; HAUSSLER, D.; STOLORZ, P. Mining scientific data. *Commun. ACM*, ACM, New York, NY, USA, v. 39, p. 51–57, November 1996. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/240455.240471>>. 17
- FAYYAD, U. M. Data mining and knowledge discovery: Making sense out of data. *IEEE Expert: Intelligent Systems and Their Applications*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 11, p. 20–25, October 1996. ISSN 0885-9000. Disponível em: <<http://dl.acm.org/citation.cfm?id=631329.631421>>. xiii, 21, 22
- FAYYAD, U. M. Mining Databases: Towards Algorithms for Knowledge Discovery. *IEEE Data Engineering Bulletin*, v. 21, p. 39–48, 1998. 21

FERREIRA, A. C. *Um Sistema de Gestão Empresarial Baseado no conceito de Enterprise Resource Planning -ERP*. Monografia (Bacharelado em Análise de Sistemas) — Universidade Católica de Pelotas, Escola de Informática, 2004. 19

FIGUEIREDO, J. M. *Formalização do domínio imagem para buscas por conteúdo em SGBDs relacionais*. Tese (Doutorado) — Universidade de São Paulo (USP), 2005. 2

FLAVINHA, F. A. Jboss seam simplicidade e produtividade no desenvolvimento de aplicações web. *o JavaMagazine*, v. 58, 2008. 25, 26

GOEBEL, M.; GRUENWALD, L. A survey of data mining and knowledge discovery software tools. *SIGKDD Explor. Newsl.*, ACM, New York, NY, USA, v. 1, p. 20–33, June 1999. ISSN 1931-0145. Disponível em: <<http://doi.acm.org/10.1145/846170.846172>>. 21

GONÇALVES, A. L.; PACHECO, R. C. S.; MORALES, A. B. T. Utilização de técnicas de mineração de dados em bases de c t: uma análise dos grupos de pesquisa no brasil. In: *XXI ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO – ENEGEP, Salvador–Bahia*. Rio de Janeiro–RJ: Anais do XXI Encontro Nacional de Engenharia de Produção – XXXI ENEGEP, 2001. 20

GRAY, J.; LIU, D. T.; NIETO-SANTISTEBAN, M.; SZALAY, A.; DEWITT, D. J.; HEBER, G. Scientific data management in the coming decade. *SIGMOD Rec.*, ACM, New York, NY, USA, v. 34, p. 34–41, December 2005. ISSN 0163-5808. Disponível em: <<http://doi.acm.org/10.1145/1107499.1107503>>. 2, 17

GROUP, O. M. *Unified Modeling Language Specification*. Object Management Group. 2011. Disponível em: <<http://www.omg/technology/documents/>>. 25

HAIR, J. F.; BLACK, W. C.; BABIN, B. J.; ANDERSON, R. E. *Multivariate Data Analysis*. 7. ed. [S.l.]: Prentice Hall, 2009. 21

HARROLD, M. K.; HIERONS, R.; DANICIC, S. The relationship betewen program dependnce and mutation analysis. In: *Mutation 2000 Symposium*. [S.l.]: CA: Kluwer Academic Publishers, 2000. p. 61–72. 27

- HEUSER, C. A. *Projeto de Banco de Dados*. [S.l.]: Instituto de Informática da UFMG, 1998. 38
- HINNEBURG, A.; KEIM, D. A. Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In: . [S.l.]: Morgan Kaufmann, 1999. p. 506–517. 21
- JACOBSON, A. S.; BERKIN, A. L.; ORTON, M. N. Linkwinds: interactive scientific data analysis and visualization. *Commun. ACM*, ACM, New York, NY, USA, v. 37, p. 42–52, April 1994. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/175276.175280>>. 17
- JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. *The Unified Software Development Process*. 1. ed. [S.l.]: Addison-Wesley Professional, 1999. 25
- JAVA. *Página oficial do Java*. Outubro 2011. Disponível em: <<http://www.java.com>>. 25
- JENSEN, C. S.; SNODGRASS, R. T. Semantics of time-varying information. *Information Systems*, 1996. 18
- JENSEN, C. S.; SNODGRASS, R. T. *Semantics of time-varying attributes and their use for temporal database design. Technical report*. [S.l.]: Time Center, 1997. 18
- JOHNSTON, T.; WEIS, R. *Managing Time in Relational Databases: How to Design, Update and Query Temporal Data*. [S.l.]: Morgan Kaufmann, 2010. 12
- KALE, V. *Implementing SAP R/3: The Guide for Business and Technology Managers*. 1. ed. [S.l.]: Sams, 2000. ISBN 0672317761. 19
- KAMATH, C. Mining science data. In: NSTITUTE OF PHYSICS PUBLISHING. *Journal of Physics: Conference Series 46*. [S.l.], 2006. p. 500-5004. 12, 22
- KELLER, E. L. Manufacturing systems. *Leassons Leanerd*, v. 17, n. 11, p. 44–50, 1999. 19

KIRANYAZ, S.; CAGLAR, K.; GULDOGAN, E.; GULDOGAN, O.; GABBOUJ, M. Muvis: A content-based multimedia indexing and retrieval framework. In: *Proc. of the Seventh International Symposium on Signal Processing and its Applications, ISSPA 2003*. [S.l.: s.n.], 2003. p. 1–8. 12

LAUDON, K. C.; LAUDON, J. P. *Management Information Systems - Organizaton and Tecnoology In the Network Enterprise*. 6. ed. USA: Prentice Hall, 2000. 19

LAUDON, K. C.; LAUDON, J. P. *Sistemas de informação gerenciais: administrando a empresa digital*. 5. ed. [S.l.]: Pearson, 2007. 19

LIU, J.; LIU, Z.; HE, J.; LI, X. Linking uml models of design and requirement. In: *Software Engineering Conference*. [S.l.]: IEEE/IET Electronic Library (IEL), 2004. p. 329 – 338. 25

LUCAS, H. C. *Information Technology and the Productivity Paradox: Assessing the Value of Investing in IT*. [S.l.]: Oxford University Press, USA, 1999. ISBN 0195121597. 19

LUCENA, C. J. P. de; MEDEIROS, C. B.; L., C. L.; MALDONADO, J.; ALMEIDA, V. A. F. *Grandes Desafios da Pesquisa em Computação no Brasil 2006 a 2016*. [S.l.], 2006. Disponível em: <http://www.sbc.org.br/index.php?option=com_jdownloadsItemid=195task=finishcid=11catid=50>. 3

MALDONADO, J. C. *Crítérios potenciais usos: Uma contribuição ao teste estrutural de software*. Tese (Doutorado) — DCA/FEE/UNICAMP, Campinas/SP, 1991. 27

MARIOTE, L. E. *Mineração de séries temporais de dados de sensores*. Dissertação (Dissertação) — UNICAMP, 2008. 18

MCCLELLAN, M. *Enterprise Resource Planning (ERP)*. Vancouver, WA, 200. 19

MYERS, G. J. *The art of software testing*. New Yourk: Wiley, 1979. 27

(NASA), N. A. S. A. *Report of the EOS Data Panel, Earth Observing System Data and Information System. Technical Memorandum 87777, National Aeronautics Space Administration (NASA. 2a. ed. Washington, DC, 1986. 17*

NOVAIS, J. W. Z.; CURADO, L. F. A.; RODRIGUES, T. R.; OLIVEIRA, A. G.; PAULO, S. R.; NOGUEIRA, J. S. Azonalidade da condutividade térmica do solo no norte do pantanal matogrossense. In: *XVII Congresso Brasileiro de Agrometeorologia – Guarapari/ES*. Vitória-ES: DCM/Incaper,, 2011. v. 17. 24

NOVAIS, J. W. Z.; CURADO, L. F. A.; RODRIGUES, T. R.; OLIVEIRA, A. G.; PAULO, S. R.; NOGUEIRA, J. S. Variabilidade sazonal horária do fluxo de calor no solo no norte do pantanal. In: *XVII Congresso Brasileiro de Agrometeorologia – Guarapari/ES*. Vitória-ES: DCM/Incaper,, 2011. v. 17. 24

OLIVEIRA, A. G. *Uma arquitetura de componentes para gestão ambiental*. Monografia (Bacharelado em Ciência da Computação) — Universidade Federal de Mato Grosso, 2008. 11

OLIVEIRA, A. G.; FIGUEIREDO, J. M. Arquitetura de componentes e serviços para gestão ambiental. In: *XVII SEMINÁRIO DE INICIAÇÃO CIENTÍFICA*. Cuiabá/MT: CADERNO DE RESUMOS DO XVII SEMINÁRIO DE INICIAÇÃO CIENTÍFICA, 2009. 6

OLIVEIRA, A. G.; LIBERATTI, G.; RODRIGUES, R.; MARTINS, J. M. F. . C. A.; BONFANTE, A. G. Uma arquitetura de componentes para processamento de imagens ambientais. In: *3º Simpósio de Geotecnologias no Pantanal, 2010, Cáceres/MT*. São José dos Campos – SP: Anais do III Simpósio de Geotecnologias no Pantanal. Embrapa Informática Agropecuária, 2010. p. 885–893. 6

OLIVEIRA, A. G.; MARTINS, C. A.; BONFANTE, A. G.; FIGUEIREDO, J. M. Sistemas de informação geográfica e o uso de tecnologias livres. In: *Mato Grosso Digital 2008*. Cuiabá/MT: [s.n.], 2008. 6

- OLIVEIRA, A. G.; VENTURA, T. M.; FIGUEIREDO, J. M.; MARTINS, C. A.; BONFANTE, A. G. Gerenciamento e armazenamento de dados micrometeorológico. In: *Escola Regional de Informática ERIMT*. Cuiabá/MT: [s.n.], 2010. 6
- OLIVEIRA, P. J.; RODRIGUES, F.; RANGEL, P. Limpeza de dados uma visão geral. In: PROCEEDINGS OF DATA GADGETS 2004 WORKSHOP BRINGING UP EMERGING SOLUTIONS FOR DATA WAREHOUSING SYSTEMS. *SIMPÓSIO DOUTORAL DO DEPARTAMENTO DE INFORMÁTICA*. Malaga/Espanha, 2004. p. 39-51. 9
- OZSOYOGLU, G.; SNODGRASS, R. Temporal and real-time databases: a survey. *Knowledge and Data Engineering, IEEE Transactions on*, v. 7 Issue:4, 1995. 17
- PARENT, C.; SPACCAPIETRA, S.; ZIMÁNYI, E. Spatio-temporal conceptual models: data structures + space + time. In: *Proceedings of the 7th ACM international symposium on Advances in geographic information systems*. New York, NY, USA: ACM, 1999. (GIS '99), p. 26–33. ISBN 1-58113-235-2. Disponível em: <[http://doi.acm.org/10.1145-320134.320142](http://doi.acm.org/10.1145/320134.320142)>. 18
- PFALTZ, J. L. What constitutes a scientific database? In: *19th International Conference on Scientific and Statistical Database Management, SSDBM*. Banff, Canada: IEEE Computer Society, 2007. p. 2. 16, 23, 50
- POSTGRES. *Página oficial do sistema gerenciador de banco de dados Postgresql*. 2011. Disponível em: <<http://www.postgresql.org/>>. 25
- PRESSMAN, R. S. *Software engineering - a practitioner's approach*. 5. ed. [S.l.]: McGraw-Hill, 2001. 27
- RAFIEI, D.; MENDELZON, A. Similarity-based queries for time series data. *SIGMOD Rec.*, ACM, New York, NY, USA, v. 26, p. 13–25, June 1997. ISSN 01635808. Disponível em: <<http://doi.acm.org/10.1145/253262.253264>>. 18
- RAVICHANDRAN, T.; RAI, A. Total quality management in information systems development: key constructs and relationships. *J. Manage. Inf. Syst.*, M. E. Sharpe, Inc.,

Armonk, NY, USA, v. 16, p. 119–155, December 1999. ISSN 0742-1222. Disponível em: <<http://dl.acm.org/citation.cfm?id=1195835.1195843>>. 20

RICCIO, E. L. *EFEITOS DA TECNOLOGIA DE INFORMAÇÃO NA CONTABILIDADE: Estudo de Casos de Implementação de Sistemas Empresariais Integrados - ERP*. Tese (Doutorado) — Faculdade de Economia Administração e Contabilidade, Universidade de São Paulo (USP), 2001. 19

RODRIGUES, T. R.; CURADO, L. F. A.; NOVAIS, J. W. Z.; OLIVEIRA, A. G.; PAULO, S. R.; NOGUEIRA, J. S. Distribuição sazonal dos componentes do balanço de energia no norte do pantanal. In: *XVII Congresso Brasileiro de Agrometeorologia – Guarapari/ES*. Vitória-ES: DCM/Incaper,, 2011. v. 17. 24

RODRIGUES, T. R.; CURADO, L. F. A.; NOVAIS, J. W. Z.; OLIVEIRA, A. G.; PAULO, S. R.; NOGUEIRA, J. S. Sazonalidade da emissividade atmosférica no norte do pantanal mato-grossense. In: *XVII Congresso Brasileiro de Agrometeorologia – Guarapari/ES*. Vitória-ES: DCM/Incaper,, 2011. v. 17. 24

RODRIGUES, T. R.; VENTURA, T. M.; OLIVEIRA, A. G.; CURADO, L. F. A.; Z. NOVAIS, J. W.; FIGUEIREDO, J. M.; MARTINS, C. A. Determinação dos parâmetros da equação de brunt utilizando algoritmo genético. In: *XVII Congresso Brasileiro de Agrometeorologia – Guarapari/ES*. Vitória-ES: DCM/Incaper,, 2011. v. 17. 24

SEAM, J. *Página Oficial do Framework Jboss Seam*. 2011. Disponível em: <<http://seamframework.org/>>. 25

SHOSHANI, A.; KAWAGOE, K. Temporal data management. *Twelfth International Conference on Very Large Data Bases*, p. 79-88, 1986. 18

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. *Sistema de Banco de Dados 3th edica ao*. S ao Paulo: MAKRON Books Ltda, 1999. 13

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. *Database System Concepts, 4th edition*. [S.l.]: McGraw-Hill, 2001. 23

- SOUZA, C. A.; ZWICKER, R. Sistemas erp: estudos de casos multiplos em empresas brasileiras. In: _____. *Sistemas ERP no Brasil*. 1. ed. [S.l.]: Atlas, 2003. 20
- VENTURA, T. M.; OLIVEIRA, A. G.; FIGUEIREDO, J. M.; MARTINS, C. A.; BONFANTE, A. G. Estimativa de evapotranspiração utilizando redes neurais. In: *Escola Regional de Informática ERIMT*. Cuiabá/MT: [s.n.], 2010. 6, 24
- WANG, R. Y. A product perspective on total data quality management. *Commun. ACM*, ACM, New York, NY, USA, v. 41, p. 58–65, February 1998. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/269012.269022>>. 20
- WILKINSON, J. W.; CERULLO, M. J.; RAVAL, V.; WING, B. W. on. *Accounting Information Systems: Essential Concepts and Applications*. 4. ed. [S.l.]: Wiley, 1999. 19
- YE, N. *The Handbook of Data Mining*. Mahwah, NJ: LAWRENCE ERLBAUM ASSOCIATES, 2003. 12

Apêndice A

Códigos de criação das tabelas

```
-- Table: calibration

-- DROP TABLE calibration;

CREATE TABLE calibration
(
  idcalibration serial NOT NULL,
  date date NOT NULL,
  idsensor integer,
  CONSTRAINT calibration_pkey PRIMARY KEY (idcalibration),
  CONSTRAINT fk54b799ea4fa4ad7e FOREIGN KEY (idsensor)
    REFERENCES sensors (idsensor) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT fk54b799eab89d15b1 FOREIGN KEY (idsensor)
    REFERENCES sensors (idsensor) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
```

```
ALTER TABLE calibration OWNER TO pgfa;

-- Table: calibration

-- DROP TABLE calibration;

CREATE TABLE calibration
(
    idcalibration serial NOT NULL,
    date date NOT NULL,
    idsensor integer,
    CONSTRAINT calibration_pkey PRIMARY KEY (idcalibration),
    CONSTRAINT fk54b799ea4fa4ad7e FOREIGN KEY (idsensor)
        REFERENCES sensors (idsensor) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT fk54b799eab89d15b1 FOREIGN KEY (idsensor)
        REFERENCES sensors (idsensor) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE calibration OWNER TO pgfa;

-- Table: calibration

-- DROP TABLE calibration;

CREATE TABLE calibration
(
```

```

idcalibration serial NOT NULL,
date date NOT NULL,
idsensor integer,
CONSTRAINT calibration_pkey PRIMARY KEY (idcalibration),
CONSTRAINT fk54b799ea4fa4ad7e FOREIGN KEY (idsensor)
    REFERENCES sensors (idsensor) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fk54b799eab89d15b1 FOREIGN KEY (idsensor)
    REFERENCES sensors (idsensor) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE calibration OWNER TO pgfa;

-- Table: calibration

-- DROP TABLE calibration;

CREATE TABLE calibration
(
    idcalibration serial NOT NULL,
    date date NOT NULL,
    idsensor integer,
    CONSTRAINT calibration_pkey PRIMARY KEY (idcalibration),
    CONSTRAINT fk54b799ea4fa4ad7e FOREIGN KEY (idsensor)
        REFERENCES sensors (idsensor) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT fk54b799eab89d15b1 FOREIGN KEY (idsensor)

```

```
REFERENCES sensors (idsensor) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE calibration OWNER TO pgfa;

-- Table: manufacturer

-- DROP TABLE manufacturer;

CREATE TABLE manufacturer
(
    idmanufacturer serial NOT NULL,
    "name" character varying(255),
    url character varying(255),
    CONSTRAINT manufacturer_pkey PRIMARY KEY (idmanufacturer)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE manufacturer OWNER TO pgfa;

-- Table: "object"

-- DROP TABLE "object";

CREATE TABLE "object"
```

```

(
  idobject serial NOT NULL,
  "name" character varying(255) NOT NULL,
  idlocal integer,
  CONSTRAINT object_pkey PRIMARY KEY (idobject),
  CONSTRAINT fkc300a33f30aea87e FOREIGN KEY (idlocal)
    REFERENCES "local" (idlocal) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT fkc300a33fe063cf8b FOREIGN KEY (idlocal)
    REFERENCES "local" (idlocal) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE "object" OWNER TO pgfa;

```

```
-- Table: "object"
```

```
-- DROP TABLE "object";
```

```

CREATE TABLE "object"
(
  idobject serial NOT NULL,
  "name" character varying(255) NOT NULL,
  idlocal integer,
  CONSTRAINT object_pkey PRIMARY KEY (idobject),
  CONSTRAINT fkc300a33f30aea87e FOREIGN KEY (idlocal)
    REFERENCES "local" (idlocal) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,

```

```
        CONSTRAINT fkc300a33fe063cf8b FOREIGN KEY (idlocal)
            REFERENCES "local" (idlocal) MATCH SIMPLE
            ON UPDATE NO ACTION ON DELETE NO ACTION
    )
WITH (
    OIDS=FALSE
);
ALTER TABLE "object" OWNER TO pgfa;

-- Table: "object"

-- DROP TABLE "object";

CREATE TABLE "object"
(
    idobject serial NOT NULL,
    "name" character varying(255) NOT NULL,
    idlocal integer,
    CONSTRAINT object_pkey PRIMARY KEY (idobject),
    CONSTRAINT fkc300a33f30aea87e FOREIGN KEY (idlocal)
        REFERENCES "local" (idlocal) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT fkc300a33fe063cf8b FOREIGN KEY (idlocal)
        REFERENCES "local" (idlocal) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE "object" OWNER TO pgfa;
```

```
-- Table: project

-- DROP TABLE project;

CREATE TABLE project
(
  idproject serial NOT NULL,
  "name" character varying(255) NOT NULL,
  idorganization integer,
  CONSTRAINT project_pkey PRIMARY KEY (idproject),
  CONSTRAINT fked904b19338443be FOREIGN KEY (idorganization)
    REFERENCES organization (idorganization) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT fked904b19f2748411 FOREIGN KEY (idorganization)
    REFERENCES organization (idorganization) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE project OWNER TO pgfa;

-- Table: project_organization

-- DROP TABLE project_organization;

CREATE TABLE project_organization
(
```

```
projects_idproject integer NOT NULL,
organization_idorganization integer NOT NULL,
CONSTRAINT fk6e0e9c992e93989d FOREIGN KEY (organization_idorganization)
    REFERENCES organization (idorganization) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fk6e0e9c9957b09022 FOREIGN KEY (projects_idproject)
    REFERENCES project (idproject) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fk6e0e9c996fa3584a FOREIGN KEY (organization_idorganization)
    REFERENCES organization (idorganization) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fk6e0e9c99c0a8f855 FOREIGN KEY (projects_idproject)
    REFERENCES project (idproject) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE project_organization OWNER TO pgfa;

-- Table: project_user

-- DROP TABLE project_user;

CREATE TABLE project_user
(
    project_idproject integer NOT NULL,
    user_iduser bigint NOT NULL,
    CONSTRAINT fk3801613174fc4801 FOREIGN KEY (project_idproject)
        REFERENCES project (idproject) MATCH SIMPLE
```

```

        ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fk3801613190ec5fe2 FOREIGN KEY (user_iduser)
    REFERENCES "user" (iduser) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fk38016131b79fa335 FOREIGN KEY (user_iduser)
    REFERENCES "user" (iduser) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fk38016131ddf4b034 FOREIGN KEY (project_idproject)
    REFERENCES project (idproject) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE project_user OWNER TO pgfa;

```

```
-- Table: regra
```

```
-- DROP TABLE regra;
```

```

CREATE TABLE constraint
(
    idConstranint serial NOT NULL,
    constraint character varying(255),
    idtype integer,
    alias character varying(20),
    CONSTRAINT regra_pkey PRIMARY KEY (idregra),
    CONSTRAINT fk675f1c364b25fcc FOREIGN KEY (idtype)
        REFERENCES "type" (idtype) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,

```

```

CONSTRAINT fk675f1c38b65a31f FOREIGN KEY (idtype)
    REFERENCES "type" (idtype) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT constraint_alias_key UNIQUE (alias)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE constraint OWNER TO pgfa;

-- Table: sensors

-- DROP TABLE sensors;

CREATE TABLE sensors
(
    idsensor serial NOT NULL,
    "name" character varying(255),
    numserie character varying(255),
    nextcalibration date NOT NULL,
    idmanufacturer integer,
    idtype integer,
    idlocal integer,
    CONSTRAINT sensors_pkey PRIMARY KEY (idsensor),
    CONSTRAINT fk760a23f930aea87e FOREIGN KEY (idlocal)
        REFERENCES "local" (idlocal) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT fk760a23f964b25fcc FOREIGN KEY (idtype)
        REFERENCES "type" (idtype) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,

```

```

CONSTRAINT fk760a23f97b26f14d FOREIGN KEY (idmanufacturer)
    REFERENCES manufacturer (idmanufacturer) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fk760a23f98b65a31f FOREIGN KEY (idtype)
    REFERENCES "type" (idtype) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fk760a23f9bc36b0fa FOREIGN KEY (idmanufacturer)
    REFERENCES manufacturer (idmanufacturer) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fk760a23f9e063cf8b FOREIGN KEY (idlocal)
    REFERENCES "local" (idlocal) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE sensors OWNER TO pgfa;

-- Table: "type"

-- DROP TABLE "type";

CREATE TABLE "type"
(
    idtype serial NOT NULL,
    "value" character varying(255) NOT NULL,
    udm character varying(10),
    alias character varying(20),
    CONSTRAINT type_pkey PRIMARY KEY (idtype)
)

```

```
WITH (  
    OIDS=FALSE  
);  
ALTER TABLE "type" OWNER TO pgfa;  
  
-- Table: "user"  
  
-- DROP TABLE "user";  
  
CREATE TABLE "user"  
(  
    iduser bigserial NOT NULL,  
    "name" character varying(255) NOT NULL,  
    "password" character varying(10) NOT NULL,  
    username character varying(255) NOT NULL,  
    CONSTRAINT user_pkey PRIMARY KEY (iduser)  
)  
WITH (  
    OIDS=FALSE  
);  
ALTER TABLE "user" OWNER TO pgfa;
```