

ANDREWS FERNANDO DE SOUSA

**PROJETO DA BASE DE DADOS DE
UM SISTEMA WEB PARA CONTROLE
DE ENSAIOS E CALIBRAÇÃO DE
LABORATÓRIOS EM
CONFORMIDADE COM A NORMA
ISO/IEC 17025**

Trabalho de Conclusão de Curso
apresentado à Escola de Engenharia de São
Carlos, da Universidade de São Paulo
Curso de Engenharia de Computação

ORIENTADOR: Professor Doutor Carlos Dias Maciel

São Carlos

2008

Resumo do Projeto de Formatura apresentado à EESC-USP como parte dos requisitos necessários para a obtenção da conclusão do curso de Engenharia de Computação.

PROJETO DA BASE DE DADOS DE UM SISTEMA WEB PARA CONTROLE DE ENSAIOS E CALIBRAÇÃO DE LABORATÓRIOS EM CONFORMIDADE COM A NORMA ISSO/IEC 17025

Andrews Fernando de Sousa

11 / 2008

Orientador: Prof. Dr. Carlos Dias Maciel

Área de Concentração: Ensaios e calibração em laboratórios

Palavras-chave: Norma ISSO/IEC 17025, Banco de Dados, MySQL, Modelo Entidade-Relacionamento, Modelo Relacional.

RESUMO

A norma ISO/IEC 17025 foi criada com intuito de facilitar o processo de ensaios e calibração aplicáveis a todas as operações realizadas em laboratório e serve como guia na implementação da garantia da qualidade. Essa norma traz tanto os requisitos técnicos quanto os gerenciais para laboratórios que desejam fornecer resultados tecnicamente válidos. Para isso implementou-se um sistema web dentro dos requisitos exigidos pela norma ISO/IEC 17025 para manter o Laboratório de Instrumentação e Microeletrônica dentro dos padrões de qualidade. O sistema contará com um banco de dados onde serão guardadas todas as informações utilizadas nos processos de calibração e ensaios, automatizando o processo de determinação dos resultados. O projeto envolve a construção do modelo lógico de banco de dados, do modelo físico, das técnicas de controle de dependência de dados e dos métodos de consulta para uma melhor visualização e futura implementação da base de dados do sistema.

Resumo do Projeto de Formatura apresentado à EESC-USP como parte dos requisitos necessários para a obtenção da conclusão do curso de Engenharia de Computação.

**PROJETO DA BASE DE DADOS DE UM SISTEMA WEB PARA
CONTROLE DE ENSAIOS E CALIBRAÇÃO DE LABORATÓRIOS
EM CONFORMIDADE COM A NORMA ISSO/IEC 17025**

Andrews Fernando de Sousa

11 / 2008

Orientador: Prof. Dr. Carlos Dias Maciel

Área de Concentração: Ensaios e calibração em laboratórios

Palavras-chave: Norma ISSO/IEC 17025, Banco de Dados, MySQL, Modelo Entidade-Relacionamento, Modelo Relacional.

ABSTRACT

The standard ISO/IEC 17025 was created with the intention to make easier the assay and calibration process and to guide the laboratory to implement a quality service. This standard brings all the technical and management requirements to the labs that wish validate the results technically. With this reason, a system was developed to grant some requirements from the standard to keep the Laboratório de Instrumentação e Microeletrônica inside the quality standards. The system has a data bank that will keep all the information of the patterns used in the assay and calibration processes. This project has a logic model data bank, physical model, control techniques of data dependences and queries methods.

1. INTRODUÇÃO

1.1. MOTIVAÇÃO

Internacionalmente, o processo de padronização das atividades dos laboratórios de ensaio e calibração teve início com a publicação da ISO/IEC Guia 25 em 1978. Na Europa, vigorava a EN 45001.

Tanto a ISO Guia 25 como a EN 45001 não possuíam um nível de detalhamento suficiente para permitir uma aplicação/interpretação consistente e sem ambigüidades, não conseguindo garantir boa qualidade dos ensaios e calibrações realizadas pelos laboratórios. Para suprir essas lacunas, a ISO revisou a ISO Guia 25 e desse trabalho resultou a norma ISO/IEC 17025 – Requisitos gerais para a competência de laboratórios de ensaio e calibração, publicada internacionalmente no ano de 2000. No Brasil, foi publicada pela ABNT a NBR/ISO/IEC 17025 em janeiro de 2001 [2].

A velocidade com que a tecnologia avança nos transporta para a necessidade de gerenciar os processos de ensaio e calibração e automatizar a execução das tarefas, aumentando a confiabilidade dos processos e tornando mais eficiente e segura a apresentação dos dados.

A criação de um sistema para automatização dos laboratórios, onde se tem um maior controle das informações obtidas através dos experimentos realizados economiza tempo, evita erros de pesquisa, oferece flexibilidade, economia de recursos humanos e financeiros, segurança, conformidade e confiabilidade ao laboratório responsável pelos ensaios e calibrações. Esse sistema deve estar em conformidade com os requisitos da norma ISO/IEC 17025, para garantir toda a qualidade necessária para a validação técnica dos resultados.

A existência de uma base de dados é essencial para evitar problemas de perdas de dados quando, por exemplo, ocorrer queda de energia elétrica. Esta também facilita as consultas, buscas, análises, e o tratamento de informações dos experimentos já realizados, garantindo maior eficiência e desempenho no gerenciamento das informações provenientes dos ensaios. Uma vez que esse gerenciamento se torna cada vez mais complicado devido ao aumento do número de pessoas no laboratório.

Visando facilitar o tratamento e gerenciamento das informações e garantir a qualidade dos ensaios e calibrações realizadas pelo Laboratório de Instrumentação e Microeletrônica, surge a necessidade da criação e desenvolvimento desse projeto.

1.2.OBJETIVO

O projeto foi desenvolvido como Trabalho de Conclusão de Curso e tem por objetivo projetar uma ferramenta web para controle dos ensaios e calibrações realizadas no Laboratório de Instrumentação e Microeletrônica.

O projeto como um todo foi dividido em duas partes, sendo desenvolvido em parceria com outro estudante de graduação do curso de Engenharia de Computação, Coriolano Martins da Silva Neto. Mais especificamente, a parte que corresponde à criação da documentação necessária para a implementação do sistema.

O vigente documento é responsável pela criação de uma base de dados para o sistema a ser criado, mais precisamente do modelo lógico e do modelo físico do banco de dados, das técnicas de controle de dependência de dados e dos métodos de consulta das informações.

1.3.MATERIAIS E MÉTODOS

O projeto da base de dados foi realizado após o levantamento de requisitos para o sistema junto ao Laboratório de Instrumentação e Microeletrônica. Foram considerados somente os aspectos mais importantes.

Para representação do modelo lógico, entidade-relacionamento, foi utilizada uma ferramenta para criação de diagramas chamada Dia. As consultas ao banco de dados foram realizadas utilizando o MySQL.

1.3.1. Dia

O Dia (Fig.1) é um aplicativo criado em GTK+ para a criação de diagramas, e executado em Linux, Unix e Windows. Dia é um software livre lançado sob a licença GPL e inspirado pelo software comercial para Windows, o Visio, mas é mais indicado para a criação de diagramas informais e uso casual. Também pode ser usado para se desenhar diferentes tipos de diagramas.

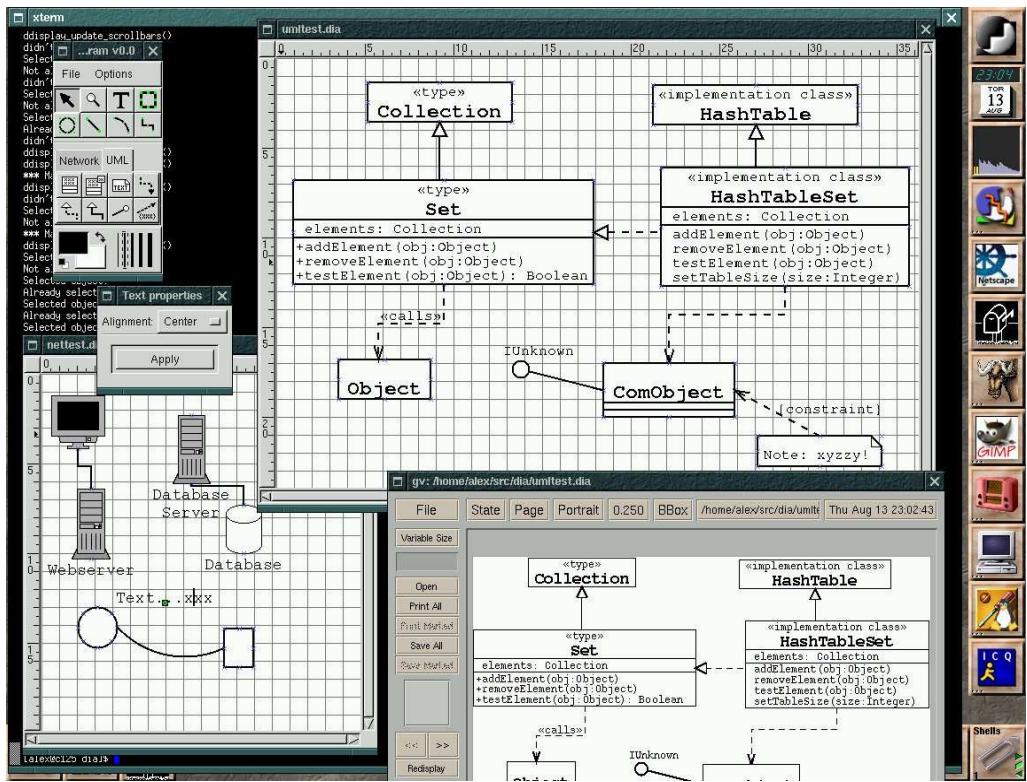


Figura 1 – “Screenshot” da tela de desenvolvimento do Dia [3].

O Dia possui atualmente objetos especiais para ajudar a desenhar diagramas de modelos entidade-relacionamento como pode ser observado na Fig.2, diagramas UML que podem ser observados na Fig.3, fluxogramas, diagramas de redes e muitos outros.

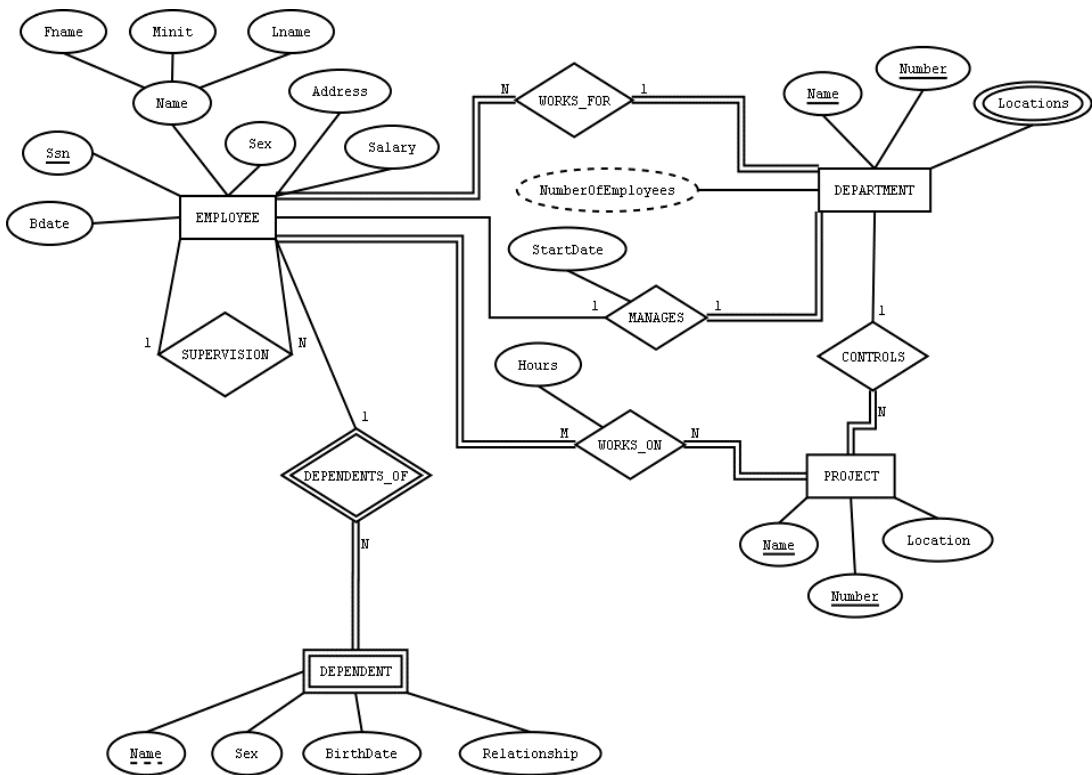


Figura 2 - Exemplo de um diagrama entidade-relacionamento feito com o Dia

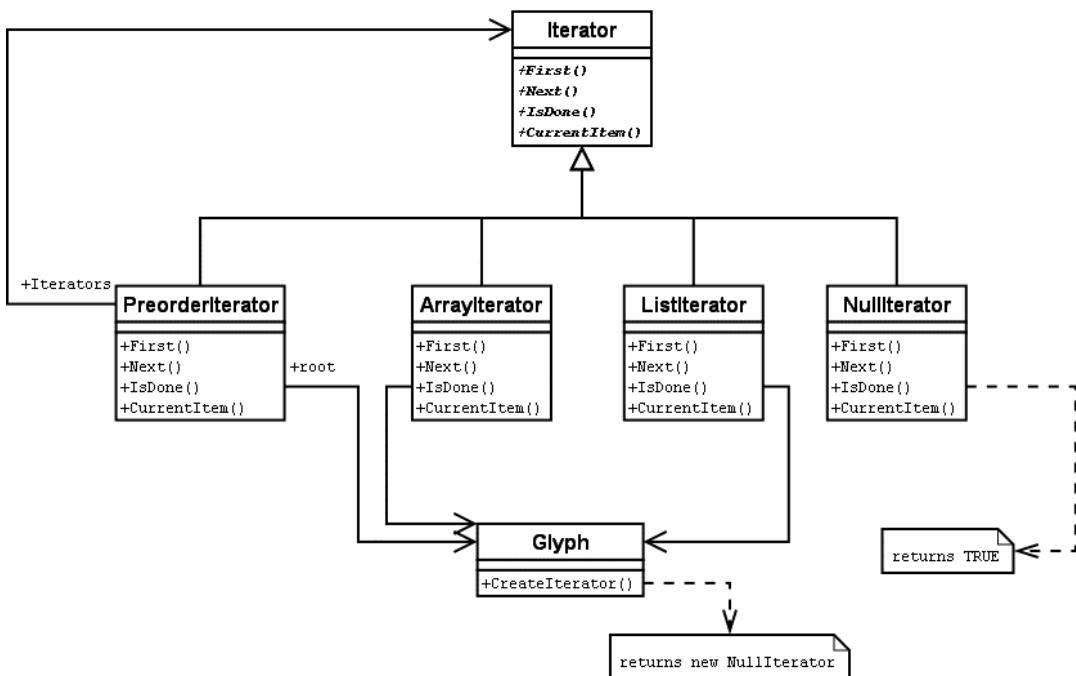


Figura 3 - Exemplo de UML feito pelo Dia

Os diagramas podem ser exportados para diferentes formatos, incluindo EPS, SVG, XFIG, WMF e PNG.

1.3.2. MySQL

O MySQL é um servidor robusto de banco de dados SQL (Structured Query Language – Linguagem Estruturada para Pesquisas) muito rápido, multi-tarefa e multi-usuário. O servidor MySQL pode ser usado em programa com alta carga e missão crítica como também pode ser utilizado para aplicações em massa. MySQL é uma marca registrada da MySQL AB [4].

O MySQL se tornou o banco de dados “*open source*” mais popular do mundo porque possui alta performance, confiabilidade, consistência e é fácil de usar. Atualmente, é usado em milhões de instalações em todos os continentes. Se tornou a escolha de uma nova geração de aplicações que utilizam o modelo LAMP (Linux, Apache, MySQL e PHP).

Segue abaixo algumas linguagens de acesso ao MySQL:

- C, C++, Java
- PHP, Python, Pearl
- Ruby
- .NET, Oracle Forms, Microsoft ASP
- Delphi, Kylix
- Lisp, Lasso, Pike

O MySQL funciona em muitas plataformas, utilizando threads nativas. Algumas delas estão listadas abaixo:

- Windows (95, 98, 2000, XP);
- Linux (x86, PPC, SPARC);
- Sun Solaris 2.5+, 32 e 64;
- Apple MacOS X;
- Netware;
- IBM OS/2;
- Silicon Graphics Irix 6.x;
- FreeBSD 3+;

- OpenBSD, NetBSD.

Utilizando threads emuladas pode ser executado para:

- BSDI 2.x
- OpenBSD 2.x
- Sun SunOS 4

O MySQL foi desenvolvido, desde 2000, já com suporte para arquiteturas 64 bits. Arquiteturas 64 bits são recomendadas para computadores quem possuem mais de 4GB de RAM.

Entre os usuários do banco de dados MySQL estão: NASA, Friendster, Banco Bradesco, Dataprev, HP, Nokia, Sony, U.S Army, Alcatel, Slashdot e outros.

O MySQL é somente o banco de dados, necessitando também de um software que interaja com o usuário para poder guardar as informações de um site web ou um software.

Abaixo o exemplo de código SQL para o MySQL:

```

DROP TABLE IF EXISTS carro;
CREATE TABLE carro (nome CHAR(20), marca CHAR(20), cor CHAR(10));
INSERT INTO carro (nome, marca, cor) VALUES ('fusca', 'VW', 'prata'),
                                             ('a3', 'audi', 'verde'),
                                             ('Ka', 'ford', 'preto');
SELECT * FROM carro;

```

Resultando em:

Nome	Marca	Cor
Fusca	VW	prata
A3	audi	verde
Ka	Ford	preto

Figura 4 - Exemplo de código SQL usando MySQL

As consultas nesse projeto foram feitas em SQL utilizando-se o servidor de banco de dados MySQL.

1.4.ORGANIZAÇÃO DA MONOGRAFIA

Após este primeiro capítulo introdutório onde foram apresentadas as motivações para o desenvolvimento do trabalho, o seu objetivo e os materiais e métodos envolvidos na sua construção, o segundo capítulo apresenta uma explicação dos conceitos teóricos envolvidos no projeto. No terceiro capítulo são descritos o estado atual do projeto, os métodos e implementações envolvidos e as atividades realizadas durante o desenvolvimento. No quarto capítulo são apresentados os resultados obtidos durante as simulações e testes efetuados com o intuito de validar o trabalho. No quinto capítulo é feita uma conclusão e as considerações finais, bem como os possíveis trabalhos futuros. Finalmente, no sexto capítulo são apresentadas as obras bibliográficas consultadas para a realização do projeto.

2. FUNDAMENTOS TEÓRICOS

2.1.NBR ISO/IEC 17025

A ISO 17025 foi criada prioritariamente para atender e padronizar os serviços prestados por laboratórios que visa estabelecer ferramentas de qualidade para o setor laboratorial.



Figura 5 - Fluxo de trabalho em um laboratório de ensaios [5]

Antigamente, a padronização dos laboratórios era implementada de acordo com a ISO Guia 25, que não era aceita na Europa onde se utilizava o EN 45001. De fato, tanto a ISO Guia 25 como o EN 45001 não eram suficientemente detalhados e por isso, após revisarem essas duas normas, foi publicado internacionalmente no início do ano 2000 a ISO/IEC 17025, que inclui todos os requisitos de gerenciamento da ISO 9001 [6].



Figura 6 - NBR ISO/IEC 17025 [6]

No Brasil, a ABNT divulgou a ISO/IEC 17025 em janeiro de 2001, para especificar os requisitos que atestam competências em ensaios e calibrações. A norma se aplica a laboratórios considerados como de primeira, segunda ou terceira parte:

- Primeira Parte – o fornecedor realiza ensaio em seu próprio produto.
- Segunda Parte – o cliente realiza ensaio no produto do fornecedor.
- Terceira Parte - o laboratório que realiza o ensaio não possui interesse no produto.

Esta norma não cobre requisitos de segurança e regulamentos sobre a operação de laboratórios mas cobre a competência técnica que não é coberta pela norma ISO 9001. Ela é dividida em 14 exigências de gerenciamento e 10 técnicas.

2.1.1. Requisitos Gerenciais

Organização

- Estabelecer responsabilidades e estrutura organizacional;
- Gerenciar as atividades realizadas nas instalações permanentes, temporárias, em campo e móvel;
- Ter uma gerência técnica e um gerente da qualidade;

- Proteger as informações confidenciais e direitos de propriedade do cliente;
- Estabelecer medidas para que a gerência e o pessoal do laboratório estejam livres de quaisquer pressões e influências indevidas, que possam afetar a qualidade de seus trabalhos.

Sistemas da qualidade

- Estabelecer, implementar e manter um sistema da qualidade, documentado na extensão necessária para assegurar a qualidade de ensaios e/ou calibrações;
- Elaborar manual da qualidade que inclua pelo menos: políticas e objetivos da qualidade, responsabilidades do GT (Gerenciamento Técnico) e GQ (Gerenciamento da Qualidade) e estrutura da documentação do SQ (Sistema da Qualidade).

Controle dos Documentos

- Controlar todos os documentos que fazem parte do seu SQ (Sistema da Qualidade), assegurando análise crítica e aprovação por pessoal competente e a sua disponibilidade ao pessoal.

Análise Crítica dos Pedidos, Propostas e Contratos

- Documentar e entender os requisitos solicitados pelos clientes;
- Ter capacidade e recursos para atender os requisitos do cliente;
- Selecionar o método de calibração ou ensaio mais apropriado.

Subcontratação de Ensaios e Calibrações

- Assegurar que o subcontratado seja competente e aceito pelo cliente.

Aquisição de Serviços e Suprimentos

- Assegurar que os serviços e suprimentos adquiridos, que possam afetar a qualidade dos ensaios ou calibrações, estejam de acordo com as especificações.

Atendimento ao Cliente

- Oferecer cooperação ao cliente e permitir que este monitore o desempenho do laboratório em relação ao trabalho realizado.

Reclamações

- Solucionar as reclamações de clientes recebidas.

Controle dos Trabalhos de Ensaios ou Calibração não Conforme

- Tomar ações imediatas quando qualquer aspecto de seu trabalho de ensaio e/ou calibração não estiverem em conformidade com seus próprios procedimentos ou com os requisitos acordados com os clientes.

Ação Corretiva

- Implementar ações corretivas para eliminar as causas de uma determinada não-conformidade, evitando a sua reincidência.

Ação Preventiva

- Implementar ações para prevenir a ocorrência de não-conformidades e buscar a melhoria contínua.

Registros

- Manter registros técnicos e da qualidade legíveis de forma a permitir que as informações contidas nestes possam ser recuperadas a qualquer momento.

Auditorias Internas

- Verificar periodicamente se as suas atividades continuam a atender os requisitos do Sistema da Qualidade e da NBR ISO/IEC 17025.

Análise Crítica pela Gerência

- Assegurar a contínua adequação e eficácia do sistema da qualidade e das atividades de ensaio e/ou calibração e introduzir mudanças ou melhorias necessárias.

2.1.2. Requisitos Técnicos

Generalidades

- Fatores que determinam a confiabilidade dos resultados de ensaio e calibração.

Pessoal

- Assegurar a competência do pessoal para operar equipamentos, realizar ensaios e calibrações, analisar e aprovar resultados.

Acomodações e Condições Ambientais

- Verificar as instalações e monitorar as condições ambientais de forma a evitar que os resultados dos ensaios e calibrações sejam invalidados.

Métodos de Ensaio e Calibração e Validação de Métodos

- Assegurar que somente métodos apropriados sejam utilizados: amostragem, transporte, preparação dos itens, operação de equipamentos, ensaio, calibração, análise de dados, incerteza de medição.

Equipamentos

- Assegurar que os equipamentos e softwares utilizados pelo laboratório atendam à exatidão requerida e às especificações dos ensaios e calibrações.

Rastreabilidade da Medição

- Assegurar a rastreabilidade ao SI (Sistema Internacional de Unidades) dos seus próprios materiais de referência e dos padrões e instrumentos de medir que tiverem efeito significativo sobre os resultados dos ensaios e calibrações;
- Quando a rastreabilidade ao SI não for possível, utilizar materiais de referência certificados, métodos e padrões condensados e participar de comparações interlaboratoriais.

Amostragem

- Ter planos e procedimentos para amostragem que assegurem a validade e a aplicabilidade dos resultados do ensaio e calibração.

Manuseio de Itens de Ensaio e Calibração

- Transportar, receber, armazenar, preparar, proteger, reter e remover os itens de ensaio e calibração de forma a assegurar sua segurança e integridade e sua adequação para o ensaio e calibração.

Garantia da Qualidade de Resultados de Ensaio e Calibração

- Monitorar a validade dos resultados de ensaio e calibração por meio de atividades de controle da qualidade: comparações interlaboratoriais ou intralaboratoriais, reensaio ou recalibração de itens retidos, etc

Apresentação de Resultados

- Relatar os resultados dos ensaios e calibrações com exatidão, clareza, sem ambigüidade, incluindo toda a informação solicitada pelo cliente e necessária a sua interpretação.

Atualmente, o Inmetro a utiliza como base para a certificação laboratorial (acreditação), pois ela estabelece requisitos gerenciais para a implementação de um Sistema de Gestão da Qualidade (SGQ) em laboratórios de ensaio e calibração. Por esse motivo, muitos laboratórios têm procurado implantar a ISO/IEC 17025 que atende a NBR ISO 9001 como uma forma de melhorar a performance laboratorial, promover uma maior competitividade e de adquirir reconhecimento nacional e internacional [6].

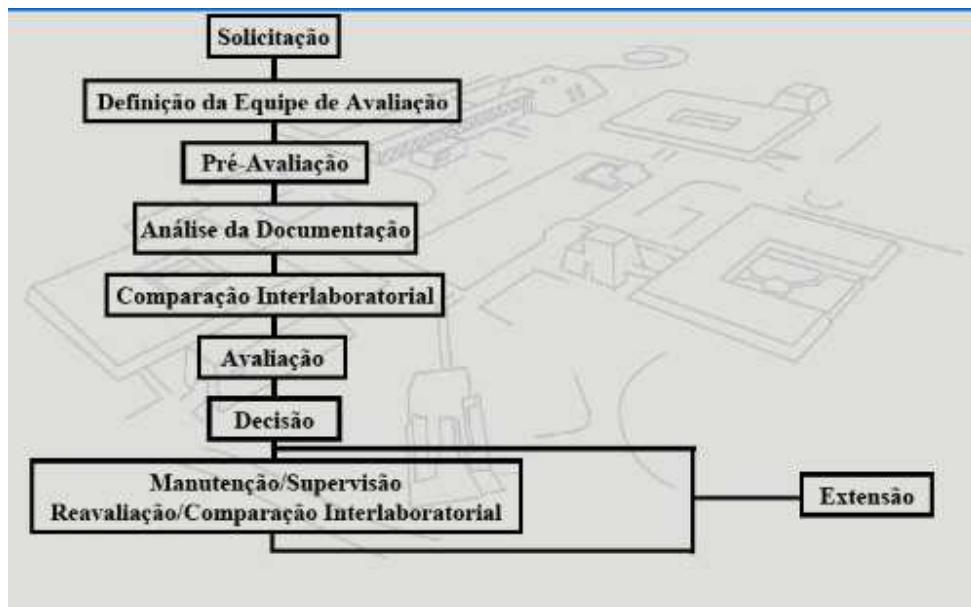


Figura 7: Acreditação laboratorial [6]

Os laboratórios que desejam demonstrar sua competência perante o Inmetro devem implementar um SGQ que esteja de acordo com a norma, cuja filosofia principal é a melhoria contínua, padronização e manutenção do conhecimento no sistema por meio da utilização de procedimentos, registros e instruções de trabalho [6].

2.2. BANCO DE DADOS

Um banco de dados pode ser definido como um conjunto de dados devidamente relacionados. Por dados podemos compreender como fatos conhecidos que podem ser armazenados e que possuem um significado implícito. Um banco de dados possui as seguintes propriedades:

- Um banco de dados é uma coleção lógica coerente de dados com um significado inerente; uma disposição desordenada dos dados não pode ser referenciada como um banco de dados;

- Um banco de dados é projetado, construído e preenchido com dados para um propósito específico; um banco de dados possui um conjunto pré definido de usuários e aplicações;
- Um banco de dados representa algum aspecto do mundo real, o qual é chamado de “mini-mundo”; Qualquer alteração feita no mini-mundo é refletida no banco de dados.

Um banco de dados é mantido e gerenciado por um SGBD (Sistema Gerenciador de Banco de Dados). O SGBD permite aos usuários criarem e manipularem banco de dados de propósito geral. O conjunto formado pelo banco de dados mais as aplicações que manipulam o mesmo é chamado de Sistema de Banco de Dados.

Uma característica importante do SGBD é que ele mantém não somente os dados mas também a forma que eles são armazenados, contendo uma descrição completa do banco de dados. Essas informações são armazenadas em catálogos pelo SGDB e são chamadas de “Meta Dados”. Assim, qualquer alteração que deva ser feita no banco de dados será realizada nos catálogos e com isso evita-se alterar os programas que estão utilizando a base de dados. A figura abaixo mostra um ambiente de sistema de banco de dados.

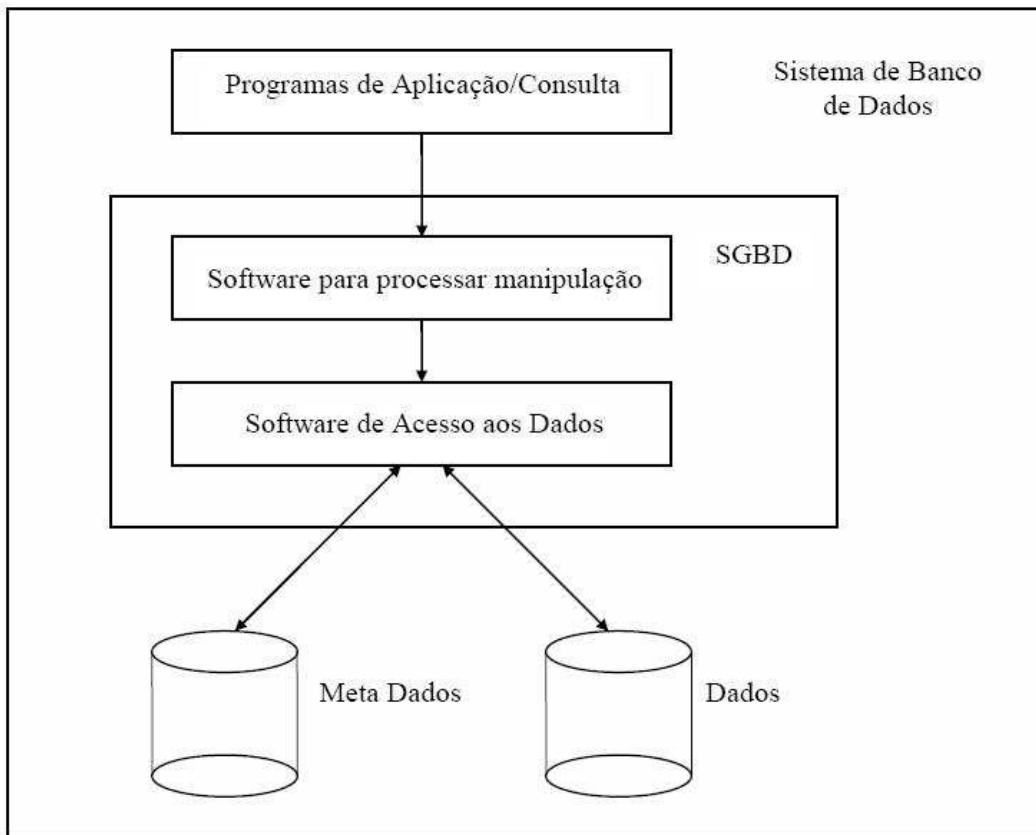


Figura 8 - Um ambiente de Sistema de Banco de Dados [7]

Um SGBD deve fornecer ao usuário uma representação conceitual dos dados, abstraindo a maneira como as informações são armazenadas. Um “modelo de dados” é utilizado para fornecer essa representação conceitual utilizando conceitos lógicos como objetos, suas propriedades e seus relacionamentos. A estrutura detalhada e a organização de cada arquivo são descritas nos catálogos.

Para um grande banco de dados, existe um grande número de pessoas envolvidas, desde o projeto, uso e até manutenção.

2.2.1. Conceitos e Arquitetura de um SGBD

Uma das principais características da abordagem banco de dados é que a mesma fornece alguns níveis de abstração de dados, omitindo ao usuário final, detalhes de como os dados estão armazenados. Podemos utilizar um modelo de dados para descrever a estrutura lógica e física de um banco de dados. Os modelos de dados podem ser, basicamente, de dois tipos:

- Alto nível: ou modelo de dados conceitual, que fornece uma visão mais próxima de como os usuários visualizam os dados realmente;
- Baixo nível: ou modelo de dados físico, que fornece uma visão mais detalhada do modo como os dados estão realmente armazenados no computador.

O SGBD trabalha com um esquema de arquitetura de três níveis (Fig.9), na qual sua principal meta é separar as aplicações do usuário do banco de dados físico. Os esquemas podem ser definidos como:

- Nível interno: descreve a estrutura de armazenamento físico do banco de dados; utiliza um modelo de dados e descreve detalhadamente os dados armazenados e os caminhos de acesso ao banco de dados;
- Nível conceitual: descreve a estrutura do banco de dados como um todo; é uma descrição global do banco de dados que não fornece detalhes do modo como os dados estão fisicamente armazenados;
- Nível externo: ou esquema de visão, o qual descreve as visões do banco de dados para um grupo de usuários; cada visão descreve quais porções do banco de dados um grupo de usuários terá acesso.

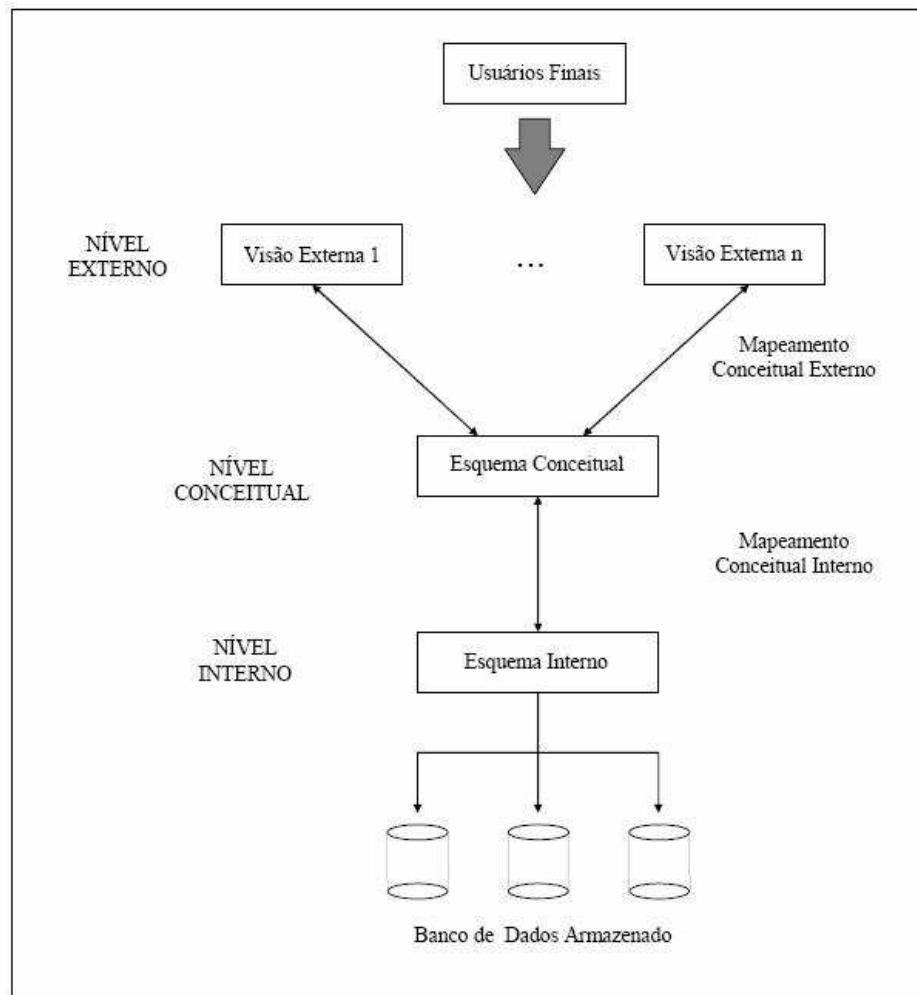


Figura 9 - Arquitetura Três esquemas [7]

Com a arquitetura três esquemas (Fig.9) consegue-se uma característica muito importante, independência de dados, que pode ser definida como a capacidade de se alterar um esquema em um nível sem precisar alterar um nível superior. Existem dois tipos de independência de dados:

- Independência de dados lógica: capacidade de alterar o esquema conceitual sem ter que alterar o esquema externo ou as aplicações do usuário;
- Independência de dados física: capacidade de alterar o esquema interno sem ter que alterar o esquema conceitual, o esquema externo ou as aplicações dos usuários.

O SGBD é um sistema complexo, formado por um conjunto muito grande de módulos. A Fig10 mostra o esquema simplificado da estrutura de funcionamento de um SGBD.

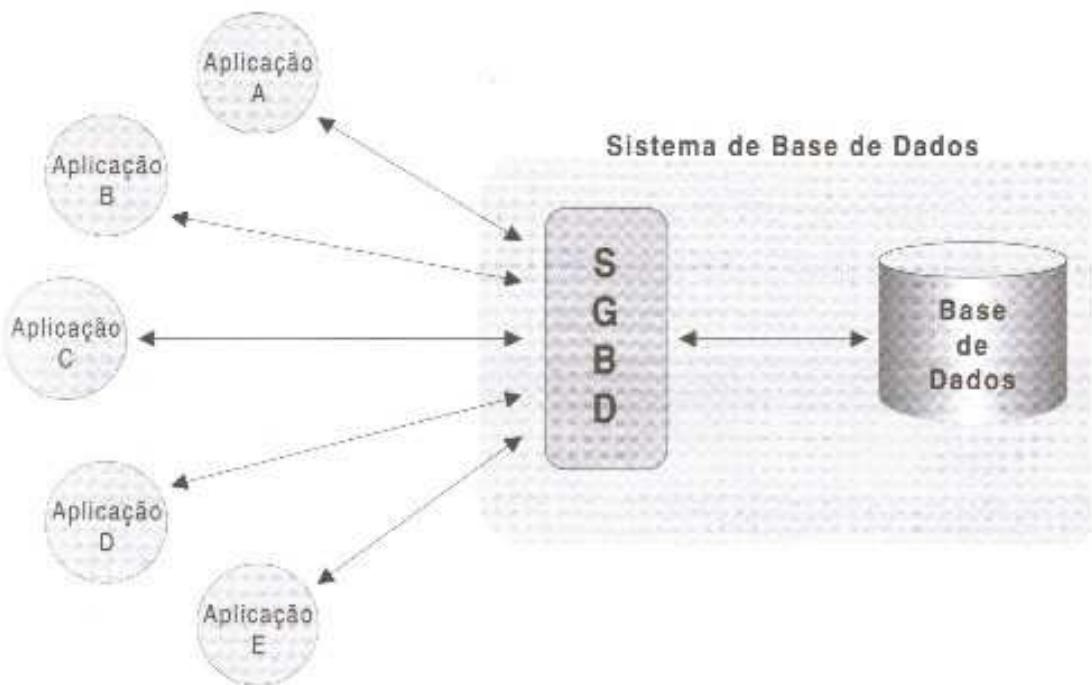


Figura 10 - Esquema da estrutura de funcionamento de um SGBD [8]

O principal critério para se classificar um SGBD é baseado no modelo de dados no qual ele é baseado. Os SGBD contemporâneos, em sua grande maioria, são classificados como de modelo relacional, modelos conceituais, modelos orientados a objetos, modelos de usuários, modelos de localização e modelos de ambiente [7].

2.2.2. Modelo Entidade Relacionamento

O modelo Entidade-Relacionamento é um modelo de dados conceitual de alto nível, cujos conceitos foram projetados para estarem o mais próximo possível da visão que o usuário tem dos dados, não se preocupando como estes dados estarão realmente

armazenados. O modelo ER é utilizado principalmente durante a fase de projeto do banco de dados. A Fig.11 apresenta um exemplo de um modelo ER.

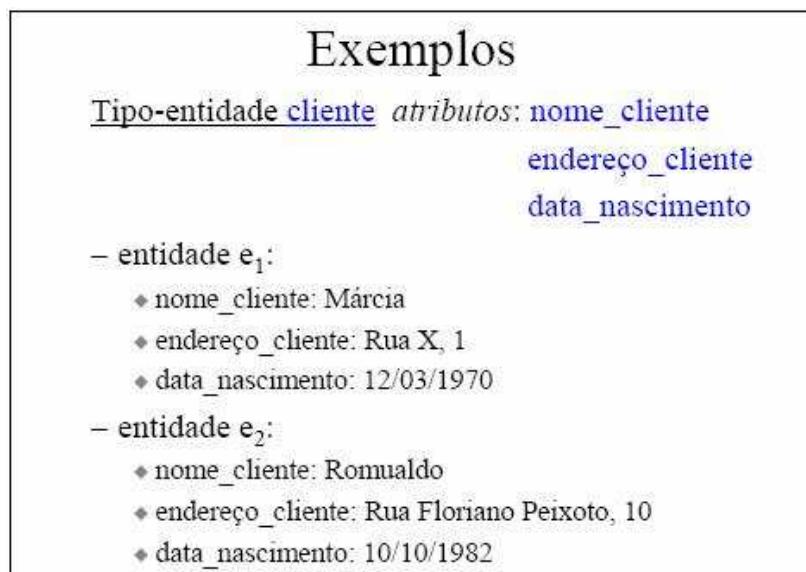


Figura 11 - Exemplos de entidade e atributos

O objeto básico tratado pelo modelo ER é a “entidade”, que pode ser definida como um objeto do mundo real, concreto ou abstrato e que possui existência independente. Cada entidade possui um conjunto particular de propriedades chamado atributos. Os atributos podem ser divididos entre simples, que não podem ser separados entre si, e compostos, que podem ser subdivididos em outras partes com significados independentes entre si.

Os atributos que podem assumir apenas um determinado valor em uma mesma instância é denominado “atributo simplesmente valorado”, enquanto que aqueles que podem assumir mais de um valor em determinada instância são chamados de “atributos multivalorados”. O atributo que é gerado a partir de outro atributo é chamado de “atributo derivado”.

Tipo entidade é o conjunto de todas as entidades que são similares entre si, ou seja, possuem o mesmo conjunto de atributos mas com valores diferentes. Cada tipo entidade é identificada por seu nome e pelo conjunto de atributos que definem suas propriedades.



Figura 12 - Exemplo de entidade e atributo

Uma restrição muito importante para as entidades de um tipo entidade é o “atributo chave”. Esse atributo chave serve para identificar uma entidade de forma única. Muitas vezes uma chave pode ser formada pela composição de um ou mais atributos. Uma entidade também pode ter mais de um atributo chave.

É importante conhecer também os relacionamentos entre os tipos entidades. Um “tipo relacionamento” R entre n entidades E_1, E_2, \dots, E_m é um conjunto de relações entre entidades deste tipo. Cada instância de relacionamento r_1 em R é uma associação de entidades, onde a associação inclui uma entidade de cada tipo entidade participante no tipo relacionamento. Isso que significa que essas entidades estão relacionadas de alguma forma no mini-mundo.

A Fig.13 apresenta um exemplo de relacionamento entre dois tipos entidade (empregado e departamento) e o relacionamento entre eles (trabalha para). Observa-se que para cada relacionamento, participam apenas uma entidade de cada tipo entidade, porém, uma entidade pode participar de mais de um relacionamento.

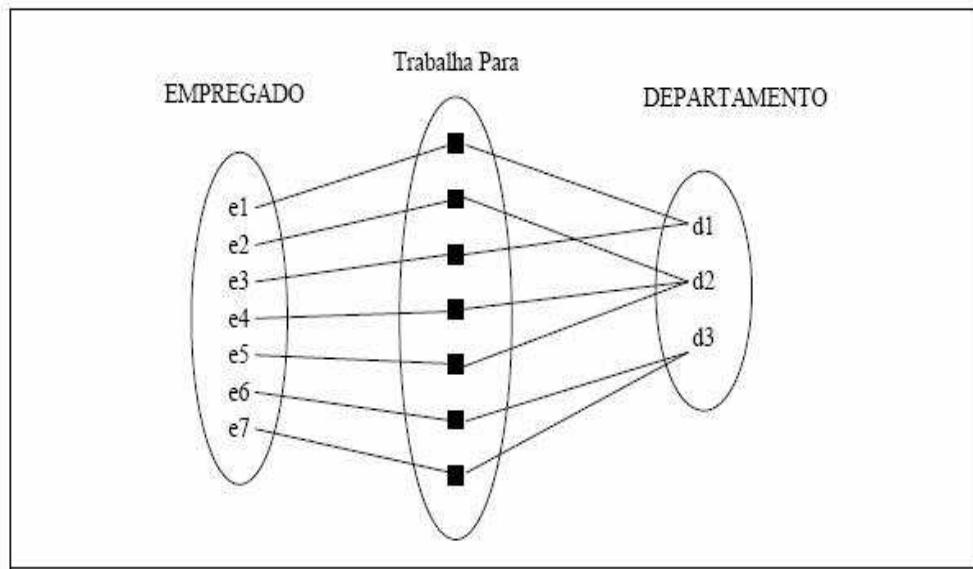


Figura 13 - Exemplo de um relacionamento_[7]

Neste exemplo (Fig.13) existe um relacionamento de grau 2 (binário), ou seja, somente dois tipos entidade participam do mesmo. O grau de um relacionamento é ilimitado, porém, a partir do grau 3 (ternário), a compreensão e a dificuldade de se desenvolver a relação corretamente se tornam extremamente complexas.

Um relacionamento pode ser recursivo, que seria o caso quando há o relacionamento entre entidades do mesmo tipo entidade. Por exemplo, na Fig.14 onde apresenta um relacionamento recursivo.

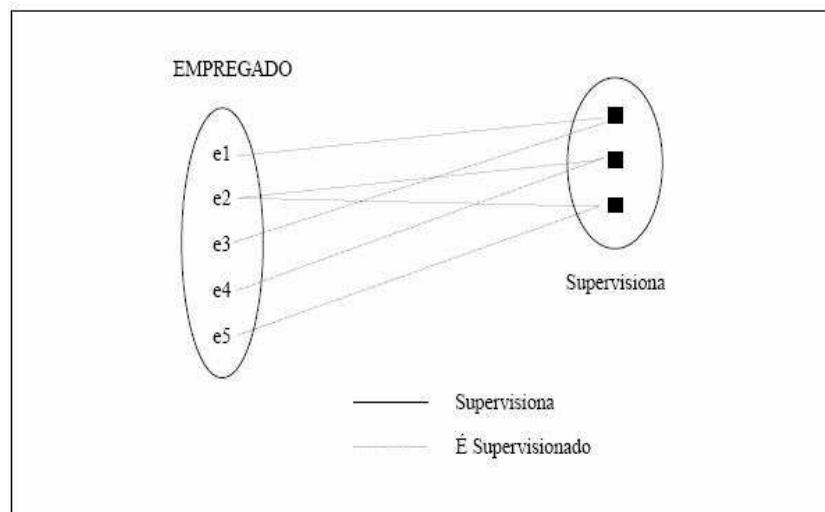


Figura 14 - Relacionamento recursivo_[7]

Os relacionamentos podem apresentar certas restrições que limitam as possíveis combinações das entidades participantes. Essas restrições são impostas pelo estado dessas entidades no mini-mundo. Por exemplo, quando um empregado pode gerenciar um departamento e um departamento pode ser gerenciado por apenas um empregado. A este tipo de restrição damos o nome de cardinalidade. Ela indica o número de relacionamentos que uma entidade pode participar. A cardinalidade pode ser 1:1, 1:N, M:N. No exemplo citado, a cardinalidade é de 1:1 pois um empregado só pode gerenciar um departamento e um departamento só pode ser gerenciado por um empregado.

No exemplo da Fig13 tem-se uma cardinalidade 1:N pois um empregado pode trabalhar em apenas um departamento mas um departamento pode ter N empregados.

Outra restrição muito importante é a participação. Podendo ela ser parcial ou total. Olhando para o exemplo de que um empregado gerencia um departamento e um departamento é gerenciado por um empregado somente, percebe-se que nem todo empregado é gerente de um departamento (participação parcial), ou seja, há empregado que não gerencia nenhum departamento, mas todo departamento é gerenciado por um empregado (participação total).

Algumas vezes torna-se necessário armazenar um atributo no tipo relacionamento, por exemplo, se quiser saber que dia tal empregado começou a gerenciar o departamento é difícil estabelecer a qual entidade pertence o atributo pois ele é definido pela existência do relacionamento. Quando a cardinalidade for 1:1 podemos colocar o atributo em um tipo entidade, de preferência aquele que tiver participação total, isso porque no caso nem todo empregado irá participar do relacionamento. Caso a cardinalidade seja 1:N então podemos colocar o atributo no tipo entidade com participação N. Porém, se a cardinalidade for N:M, então o atributo deverá ficar no relacionamento.

Um tipo entidade fraca é quando a entidade não tem o atributo chave por si só. Implica que não podemos distinguir algumas entidades por que as combinações dos valores de seus atributos podem ser idênticas. As entidades fracas precisam estar relacionadas com uma entidade pertencente ao tipo entidade proprietária. Este relacionamento é chamado relacionamento identificador.

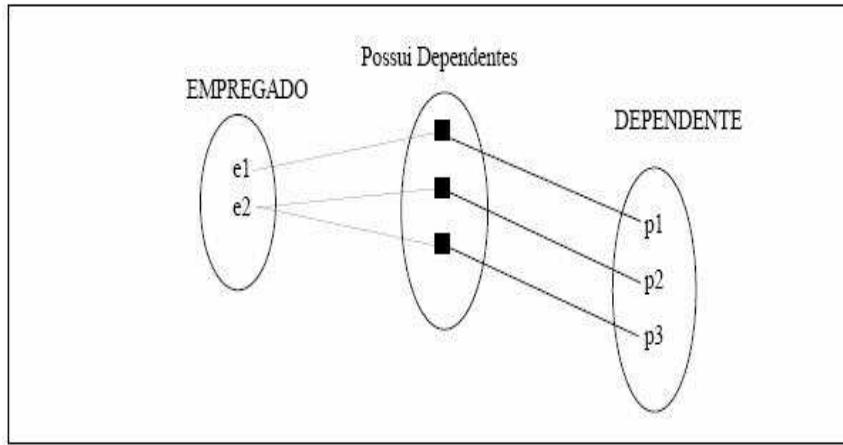


Figura 15 - Relacionamento entidade fraca (dependente) [7]

O tipo entidade DEPENDENTE é uma entidade fraca pois não possui um método de identificar uma entidade única. Por outro lado, EMPREGADO possui um atributo chave, o número do RG por exemplo. Porém, um dependente de 5 anos de idade não possui necessariamente um documento. Desta forma, é uma entidade fraca. Essa entidade possui uma chave parcial que juntamente com a chave primária da entidade proprietária forma uma chave primária composta. Neste exemplo, a chave primária de empregado é o RG, a do dependente é o nome, pois dois irmãos não possuem o mesmo nome. Desta forma a chave primária desta entidade fica sendo o RG do pai ou da mãe mais o nome do dependente.

O Diagrama Entidade Relacionamento é composto por um conjunto de elementos gráficos que visa representar todos os objetos do modelo Entidade Relacionamento tais como entidades, atributos, atributos chaves, relacionamentos, restrições estruturais, etc.

O diagrama ER fornece uma visão lógica do banco de dados, fornecendo um conceito mais generalizado de como estão estruturados os dados de um sistema [7].

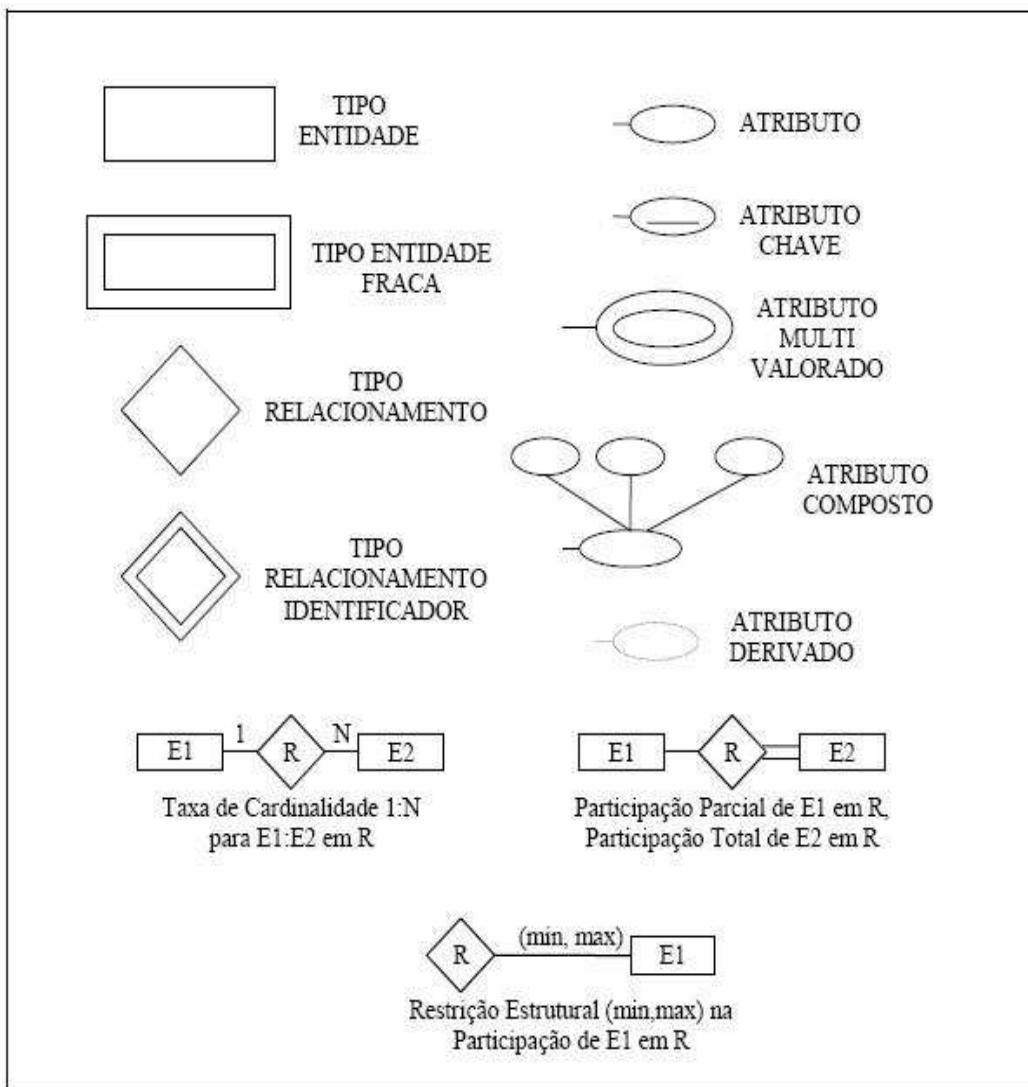


Figura 16 - Objetos que compõem o diagrama ER [7]

Abaixo segue um exemplo de diagrama ER.

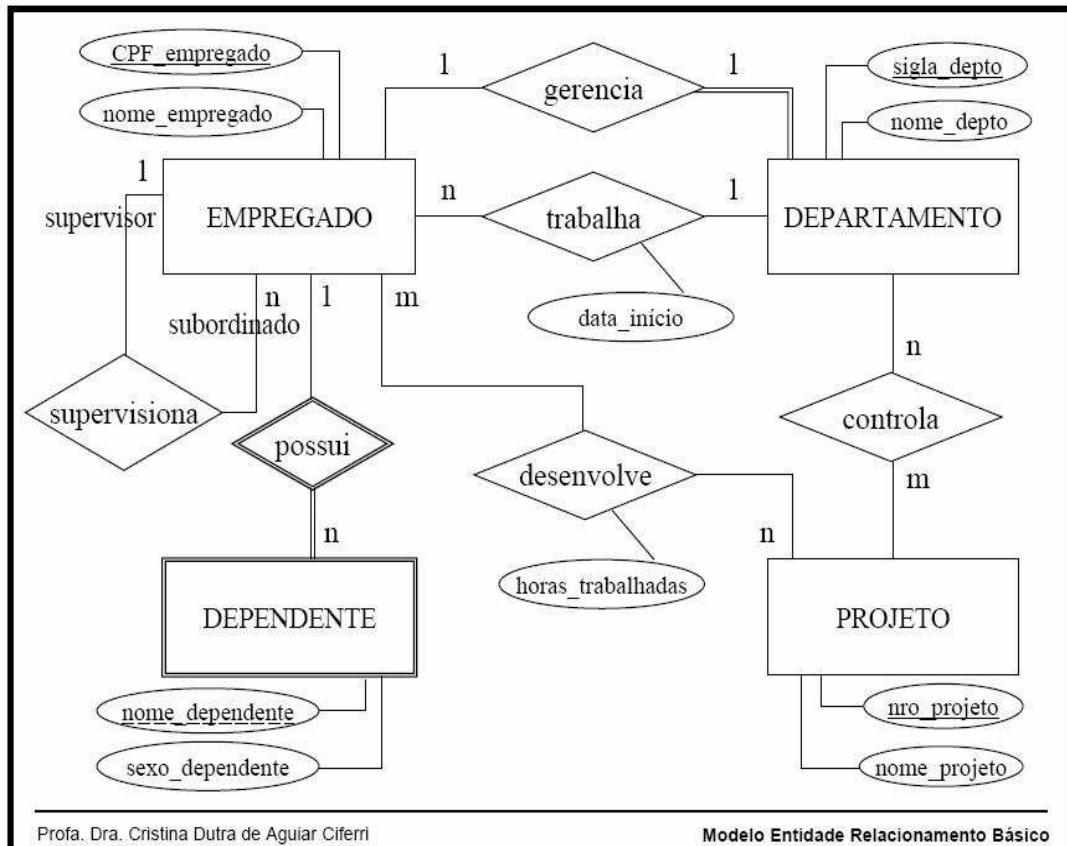


Figura 17 - Exemplo de diagrama ER [9]

2.2.3. Modelo Relacional

O modelo Relacional foi criado por Codd_[10] em 1970 e tem por finalidade apresentar os dados como uma coleção de relações, onde cada relação é representada por uma tabela, ou falando de forma mais direta, um arquivo.

Quando uma relação é pensada como uma tabela de valores, cada linha nesta tabela representa uma coleção de dados relacionados. Estes valores podem ser interpretados como sendo fatos descrevendo uma instância de uma entidade ou de um relacionamento.

Na terminologia do modelo relacional, cada tabela é chamada de relação; uma linha de uma tabela é chamada de tupla; o nome de cada coluna é chamado de atributo; o tipo de dado que descreve cada coluna é chamado de domínio [7].

Um domínio D é um conjunto de valores atômicos, sendo que por atômico podemos compreender que cada valor do domínio é indivisível. Durante a especificação do

domínio é importante destacar o tipo, o tamanho e a faixa do atributo que está sendo especificado. Por exemplo:

Coluna	Tipo	Tamanho	Faixa
RG	Numérico	10,0	03000000-25999999
Nome	Caracter	30	a-z, A-Z
Salário	Numérico	5,2	00100,00-12999,99

Figura 18 - Especificação do domínio do atributo. (Modelo Relacional)

Um esquema de relação R, denotado por $R(A_1, A_2, \dots, A_n)$, onde cada atributo A_i é o nome do papel desempenhado por um domínio D no esquema relação R, onde D é chamado domínio de A_i e é denotado por $dom(A_i)$. O grau de uma relação R é o número de atributos presentes em seu esquema de relação.

A instância r de um esquema relação denotado por $r(R)$ é um conjunto de n-tuplas $r = [t_1, t_2, \dots, t_n]$ onde os valores de $[t_1, t_2, \dots, t_n]$ devem estar contidos no domínio D. O Valor nulo também pode fazer parte do domínio de um atributo e representa um valor não conhecido para uma determinada tupla.

Uma relação pode ser definida como um conjunto de tuplas distintas. Isto implica que a combinação dos valores dos atributos em uma tupla não pode se repetir na mesma tabela.

Existirá sempre um subconjunto de atributos em uma tabela que garantem que não haverá valores repetidos para as diversas tuplas da mesma, garantindo que $t_1[SC] \neq t_2[SC]$.

SC é chamada de superchave de um esquema de relação. Toda relação possui ao menos uma superchave - o conjunto de todos os seus atributos. Uma chave C de um esquema de relação R é uma superchave de R com a propriedade adicional que removendo qualquer atributo A de C, resta ainda um conjunto de atributos C' que não é uma superchave de R. Uma chave é uma superchave da qual não se pode extrair atributos. Por exemplo, o conjunto: $(RA, Nome, Endereço)$ é uma superchave para estudante, porém, não é uma chave pois se tirarmos o campo *Endereço* continuaremos a ter uma superchave. Já o conjunto $(Nome da Revista, Volume, No da Revista)$ é uma

superchave e uma chave, pois qualquer um dos atributos que retirarmos, deixaremos de ter uma superchave, ou seja, *(Nome da Revista, Volume)* não identifica uma única tupla.

Em outras palavras, uma superchave é uma chave composta, ou seja, uma chave formada por mais que um atributo (Tab1).

Tabela 1 - Exemplo de superchave

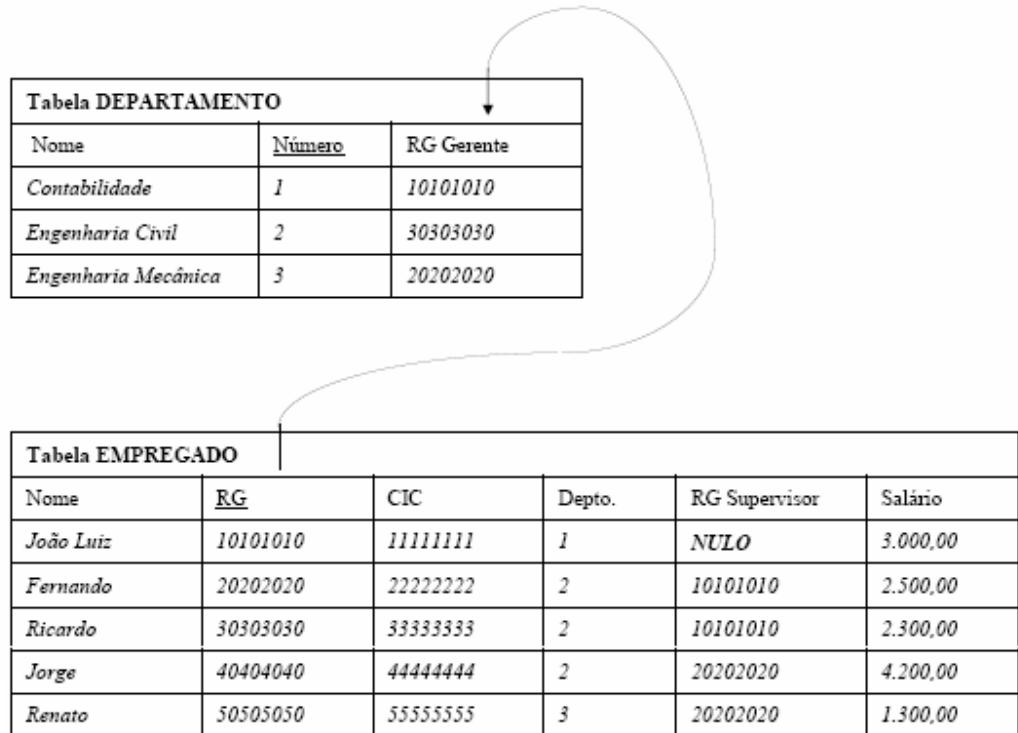


Tabela DEPENDENTES				
<u>RG Responsável</u>	<u>Nome Dependente</u>	Dt. Nascimento	Relação	Sexo
10101010	Jorge	27/12/86	Filho	Masculino
10101010	Luiz	18/11/79	Filho	Masculino
20202020	Fernanda	14/02/69	Conjuge	Feminino
20202020	Angelo	10/02/95	Filho	Masculino
30303030	Fernanda	01/05/90	Filho	Feminino

Quando uma relação possui mais que uma chave (não confundir com chave composta) - como, por exemplo, RG e CIC para empregados - cada uma destas chaves são chamadas de chaves candidatas. Uma destas chaves candidatas deve ser escolhida como chave primária.

Uma chave estrangeira CE de uma tabela R1 em R2 ou vice-versa, especifica um relacionamento entre as tabelas R1 e R2. Como pode-se observar na figura abaixo.

Tabela 2 - Exemplo de chave estrangeira



2.2.4. Dependência Funcional e Normalização

Dependência funcional é uma restrição entre dois conjuntos de atributos de uma base de dados. Esta é representada por $X \Rightarrow Y$ (lê-se: Y é funcionalmente dependente de X, ou X infere sobre Y) que são dois conjuntos de atributos e ainda são subconjuntos de R , pense em $R = \{A_1, A_2, \dots, A_n\}$. X e Y especificam uma restrição nas tuplas que podem compor uma instância relação r de R , ou seja, os valores do componente Y em uma tupla em r depende de, ou é determinada pelos valores do componente X .

O processo de normalização pode ser visto como o processo no qual são eliminados esquemas de relações (tabelas) não satisfatórios, decompondo-os através de seus atributos em esquemas de relações menos complexas mas que satisfazem as propriedades desejadas. Como foi proposto por Codd^[10], deve-se fazer uma bateria de testes para verificar se o esquema de relação está na 1^a, 2^a e 3^a formas normais. Estas três formas normais são baseadas em dependências funcionais dos atributos do esquema de relação.

A 1^a forma normal (1FN) nos diz que devemos eliminar grupos de dados repetitivos da estrutura. Para isso, deve-se localizar os atributos multivalorados, ou seja, somente são permitidos atributos atômicos. No caso, por exemplo, de um tipo entidade nota fiscal, temos como atributos produto, quantidade, valor unitário e total como grupo de dados multivalorado. Deve-se separar esse grupo em uma nova entidade, dar um nome ao grupo, levar a chave da nota fiscal (para manter a relação entre as entidades) e estabelecer uma chave para o novo grupo. Essa chave pode ser composta pela chave da nota fiscal e por mais de um atributo existente. É possível, quando não se localizar um atributo adequado, criá-la.

A 2^a forma normal (2FN) diz que quando somente houver grupos de dados na 1FN (e jamais antes disso), deve-se localizar dados que não dependem única e exclusivamente da chave da entidade. Por exemplo, o cliente está relacionado à nota fiscal, mas não depende dela. O cliente existe, mesmo que não exista a nota fiscal. Por esse motivo dizemos que o cliente não depende da nota fiscal e, por isso, deve ter os dados separados em uma nova entidade. Ao se identificar os grupos independentes, deve-se separá-los em uma nova entidade, uma para cada grupo independente. Feito isso, dá-se um nome à entidade e estabelece-se uma chave para o novo grupo. Caso não haja um bom atributo para ser a chave, deve-se criá-lo.

A 3^a forma normal (3FN) é aplicada depois que os grupos de dados estiverem todos na 2FN (e jamais antes disso). Aí então localizam-se os atributos com dependência transitiva. Dependência transitiva ocorre quando um dado pode ser obtido através de outro, exceto a chave. Por exemplo, $X \Rightarrow Y$, em uma tabela T, existe um conjunto de atributos Z que não é um conjunto de chaves de T e as dependências $X \Rightarrow Z$, $Z \Rightarrow Y$ são válidas. No exemplo da nota fiscal, temos o valor total de um item e o valor total da nota fiscal como sendo atributos que podem ser obtidos através de cálculos matemáticos, por isso devem ser excluídos [7].

Com os dados normalizados é possível criar as tabelas.

2.2.5. Álgebra Relacional

A álgebra relacional é uma coleção de operações que são utilizadas para manipular as relações. Essas operações são utilizadas para selecionar tuplas de relações individuais e para combinar tuplas relacionadas de relações diferentes para especificar uma consulta em um

determinado banco de dados. O resultado de cada operação é uma nova operação, a qual também pode ser manipulada pela álgebra relacional.

A Operação Select

Esta operação é utilizada para selecionar um subconjunto de tuplas de uma relação, que devem satisfazer uma condição de seleção.

A forma geral de uma operação select é:

$$\sigma_{<\text{condição de seleção}>}^{<\text{nome da relação}>}$$

Onde a letra sigma representa a operação de seleção, a condição de seleção é uma expressão booleana aplicada sobre os atributos da relação e nome da relação é o nome da relação sobre a qual será aplicada a operação select.

Exemplo:

$$\text{Consulta1} = \sigma_{\text{salário} < 2500,00}^{(\text{EMPREGADO})}$$

Essa consulta retorna as linhas da tabela EMPREGADO onde a coluna salário apresenta valor menor do que 2500,00 .

A Operação Project

A operação project seleciona um conjunto determinado de colunas de uma relação. A forma geral de uma operação project é:

$$\pi_{<\text{lista de atributos}>}^{<\text{nome da relação}>}$$

A letra grega pi representa a operação project, a lista de atributos representa a lista de atributos que o usuário deseja selecionar, e nome da relação representa a relação sobre a qual a operação project será aplicada.

Exemplo:

$$\text{Consulta2} = \pi_{<\text{nome, data nascimento}>}^{(\text{DEPENDENTES})}$$

Essa consulta retorna as colunas nome e data nascimento da tabela DEPENDENTES.

As operações select e project podem ser utilizadas de forma combinada , permitindo que apenas determinadas colunas de determinadas tuplas possam ser selecionadas.

Poe exemplo:

$$\text{Consulta3} = \pi_{\langle \text{nome}, \text{RG}, \text{salário} \rangle} (\sigma_{\text{salário} < 2500,00} (\text{EMPREGADO}))$$

Essa consulta retorna somente as colunas nome, RG e salário com as tuplas onde o salário for menor do que 2500,00 da tabela EMPREGADO.

Operações Matemáticas

Levando em consideração que as relações podem ser tratadas com conjuntos, podemos então aplicar um conjunto de operações matemáticas sobre as mesmas. Como por exemplo: união (U) , intersecção (\cap), e diferença (-). Estas operações podem ser aplicadas sobre mais de uma tabela, contanto que as tuplas seja exatamente do mesmo tipo. Elas são definidas da seguinte forma:

- união - o resultado desta operação representada por $R \cup S$ é uma relação T que inclui todas as tuplas que se encontram em R e todas as tuplas que se encontram em S ;
- intersecção - o resultado desta operação representada por $R \cap S$ é uma relação T que inclui as tuplas que se encontram em R e em S ao mesmo tempo;
- diferença - o resultado desta operação representada por $R - S$ é uma relação T que inclui todas as tuplas que estão em R mas não estão em S .

Produto Cartesiano

O produto cartesiano é uma operação binária que combina todas as tuplas de duas tabelas. Diferente da operação união, o produto cartesiano não exige que as tuplas da tabela possuam exatamente o mesmo tipo. Ele permite então a consulta entre tabelas relacionadas utilizando uma condição de seleção apropriada. O resultado de um produto cartesiano é um tabela resultante da combinação das tuplas das tabelas sobre as quais aplicou-se a operação.

O formato geral de uma operação produto cartesiano entre duas tabelas R e S é:

$R \times S$

Esta operação não é muito utilizada por não oferecer um resultado otimizado.

Operação Junção

A operação junção atua de forma similar a operação produto cartesiano, porém a tabela resultante conterá apenas a combinação das tuplas que se relacionam de acordo com uma determinada condição de junção.

A forma geral de uma operação de junção entre duas tabelas R e S é a seguinte:

R \times _{<condição de junção>} S

2.2.6. SQL

SQL (Structured Query Language) é uma linguagem de pesquisa declarativa para banco de dados relacional. Muitas das características originais do SQL foram inspiradas na álgebra relacional. Decorrente de sua facilidade de uso e simplicidade, a linguagem SQL é um grande padrão de banco de dados. Ela se diferencia de outras linguagens pois especifica a forma do resultado e não o caminho pra chegar até ele [6]. Os principais comandos SQL estão listados abaixo.

Comando CREATE TABLE

Permite ao usuário criar uma nova tabela (ou relação). Para cada atributo da relação é definido um nome, um tipo, máscara e algumas restrições. A restrição “not null” significa que o atributo deve obrigatoriamente ser preenchido. Se não for especificado é admitido que o atributo possa assumir um valor nulo, por default. Por exemplo para se criar uma tabela empregados o seguinte comando deverá ser utilizado:

```
CREATE TABLE metodo (
    data          DATE        NOT NULL,
    titulo        CHAR (30)
    descrição    CHAR(100)
    file          CHAR (30)
    PRIMARY KEY (titulo)
)
```

Comando DROP TABLE

O comando drop table permite a exclusão uma tabela (relação) em um banco de dados.

Por exemplo para eliminarmos a tabela empregados, o comando a ser utilizado é o seguinte:

```
DROP TABLE empregados;
```

Se a chave da tabela empregados for usada como chave estrangeira ou chave primária composta em outras tabelas, deve-se corrigi-las. Este processo não é assim tão simples pois implica na alteração do projeto físico de diversas tabelas.

Comando ALTER TABLE

O comando alter table permite que o usuário faça a inclusão de novos atributos em uma tabela. A forma geral para o comando é a seguinte:

```
ALTER TABLE <nome_tabela> ADD <nome_coluna> <tipo_coluna>
```

Para o comando alter table não é permitido utilizar a restrição not null, porque assim que uma coluna é criada, todas as tuplas da tabela para aquela coluna receberão o valor null.

Consultas em SQL

Em SQL pode-se fazer consultas, inserir e atualizar dados em uma ou mais tabelas. Para isso utiliza-se os comandos SELECT , INSERT INTO e UPDATE.

Com o comando SELECT conseguimos selecionar tuplas em uma ou mais tabelas. Sua forma básica é a seguinte:

```
SELECT <lista de atributos>
FROM <lista de tabelas>
WHERE <condições>;
```

Por exemplo, para selecionar o nome e o rg dos funcionários que trabalham no departamento 2 e que ganham mais do que R\$2500,00 na tabela empregados, utilizamos o seguinte comando:

```
SELECT nome, rg
FROM empregados
WHERE departamento = 2 AND salário > 2500;
```

Diferente de álgebra relacional, a operação SELECT permite a geração de tuplas duplicadas como resultado de um expressão. Para evitar isso utiliza-se o especificador DISTINCT.

```
SELECT DISTINCT nome
FROM empregados
WHERE departamento = 2;
```

Se quisermos ordenar as tuplas de forma ordenada, usa-se o comando ORDER BY.

```
SELECT nome, rg
FROM empregados
ORDER BY nome;
```

Inserções em SQL

Para elaborar inserções em SQL, utiliza-se o comando INSERT INTO. A forma geral para o comando INSERT INTO é:

```
INSERT INTO <nome da tabela> <(lista de colunas)>
VALUES <(lista de valores)>;
```

Por exemplo:

```
INSERT INTO empregados (nome, rg, salário)
VALUES ('Jose Silva', 1234567, 3600);
```

Se todas as colunas forem preenchidas no comando INSERT INTO não é necessário informar a lista de colunas, mas se somente alguns atributos forem inseridos deve-se então dizer em quais as colunas os valores serão inseridos.

Atualizações em SQL

Para realizar atualizações nas tabelas é utilizado o comando UPDATE. Sua forma básica é a seguinte:

```
UPDATE <tabela>
SET <coluna> = <expressão>
WHERE <condição>
```

Por exemplo, para atualizar o salário de todos os empregados que trabalham no departamento 2 para R\$3000,00 o seguinte código é utilizado:

```
UPDATE empregados
SET salário = 3000,00
WHERE departamento = 2;
```

Eliminar registros em SQL

Para eliminar um tupla de uma tabela, utiliza-se o comando delete. A forma básica do comando delete é a seguinte:

```
DELETE FROM <tabela>
WHERE <condição>;
```

Se quisermos eliminar os registros nos quais os empregados que trabalham no departamento 2 ganham mais de R\$3500,00 devemos utilizar o seguinte comando:

```
DELETE FROM empregados
WHERE departamento = 2 AND salário > 3500;
```

3. MÉTODOS E IMPLEMENTAÇÕES

3.1.PROJETO

A base de dados desenvolvida neste projeto guardará todas as informações dos padrões para controle de dados e métodos utilizados nas atividades realizadas pelo Laboratório de Instrumentação e Microeletrônica, como também os dados e resultados de ensaios e calibrações para posterior consulta e estudo dos mesmos.

O projeto do banco de dados envolve a construção de um modelo lógico, um modelo físico, envolve também as técnicas de controle e dependência de dados e métodos de consulta e criação de dados. A existência da mesma, garante maior segurança dos dados das análises realizadas, uma vez que os usuários deverão possuir uma senha para acesso das informações. Com isso evita-se a perda de dados acidentalmente, ou por queda de energia ou mesmo por simplesmente não saber a localização das informações para um dado experimento.

A idéia do projeto é passar todas as informações e materiais necessários para uma futura implementação do sistema como um todo, tanto a parte da base de dados como a aplicação para controlá-la.

3.2.EQUIPE

O trabalho foi desenvolvido junto com outro aluno da graduação, Coriolano Martins da Silva Neto, que ficou responsável por toda a parte de modelagem da aplicação que deverá interagir com a base de dados modelada aqui. As atividades realizadas pela outra parte do projeto é essencial para a realização deste, pois a modelagem da base de dados depende das informações obtidas pelo Coriolano. Suas atividades são resumidas da seguinte forma (Fig19):

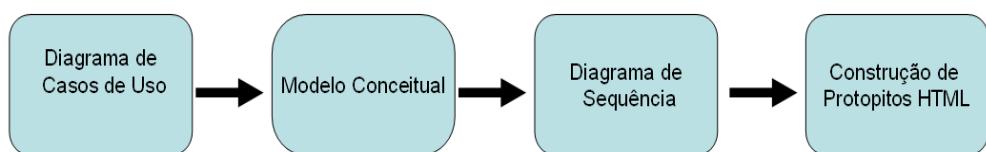


Figura 19 - Atividades realizadas pela outra parte da equipe

De posse do modelo criado, pode-se então modelar toda a base de dados que deverá interagir com as páginas dinâmicas para posterior exibição aos usuários através dos documentos HTML que serão exibidos pelo navegador web.

3.3. DESCRIÇÃO DAS ATIVIDADES REALIZADAS

Todas as etapas, desde a modelagem até o projeto físico da base de dados é mostrada na Fig20.

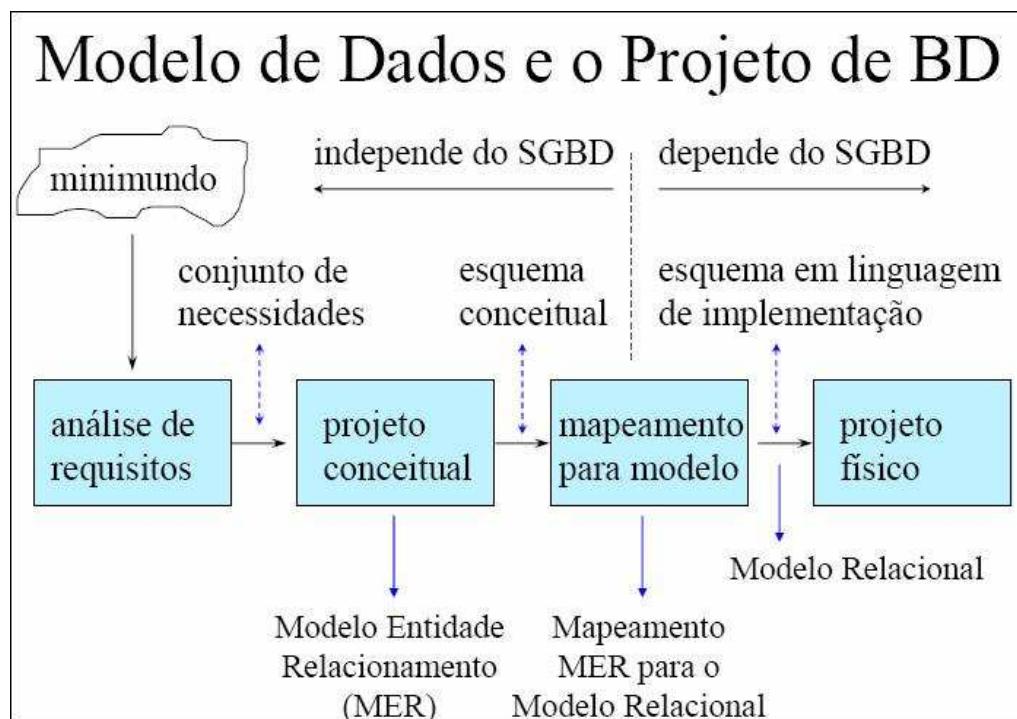


Figura 20 - Etapas do projeto de um banco de dados

A primeira etapa consiste do levantamento de requisitos do sistema, ou seja, todas as informações relevantes da aplicação que consistem em descrever e dar uma idéia geral de como será o sistema. Esses requisitos são levantados a partir das necessidades do Laboratório de Instrumentação e Microeletrônica, cobrindo todas as necessidades do mesmo.

Após essa etapa já é possível a criação do modelo Entidade-Relacionamento, que descreve e especifica todo o esquema conceitual da base de dados. O MER diz como as entidades se relacionam, como elas estão ligadas umas as outras e todos os atributos das mesmas. De posse do modelo Entidade-Relacionamento podemos criar o diagrama Entidade-Relacionamento.

A próxima fase é o mapeamento do modelo ER para o modelo Relacional, para a posterior construção do modelo físico do banco de dados, ou seja, a partir do modelo lógico chega-se ao modelo físico. Onde os dados serão representados como uma coleção de relações, e cada relação é representada por uma tabela.

A fase seguinte consiste na verificação dos dados para a eliminação das dependências, utilizando para isso a normalização, abordando a 1^a, 2^a e 3^a Formas Normais.

Por fim, para a elaboração de consultas utilizaremos a álgebra relacional, que é uma forma canônica para as linguagens de consulta, e SQL, que a linguagem mais usada para interagir com a base de dados.

3.3.1. Projeto Conceitual Utilizando Modelo ER

Após o levantamento de requisitos, das necessidades do Laboratório de Instrumentação e Microeletrônica, gera-se uma descrição mais detalhada dos tipos de entidades envolvidas com seus respectivos atributos:

- ENSAIO: ID, data, assunto, descrição, metodo, file (caminho do diretório onde estão os arquivos utilizados no ensaio), equipamentos (multivalorado e composto), responsável;
- COLABORADOR: nome, usp (número de identificação), curso, data_ingresso, senha, tipo_usuario (gerente ou usuário), laboratório, email;
- METODO: data, titulo, descrição, file;
- LABORATORIO: nome, cnpj, responsável_geral, equipamentos (multivalorado e composto);

Foram identificados também os tipos relacionamentos, com as razões de cardinalidade, restrição de participação e atributos:

- COLABORADOR **pertence a** LABORATORIO:

Razão de cardinalidade (1:1), pois:

- 1 colaborador pertence a **1** laboratório;
- 1 laboratório tem **1** colaborador.

Restrição de participação:

- do COLABORADOR: parcial, pois o COLABORADOR não precisa ter um LABORATÓRIO para que ele exista no banco de dados;
- do LABORATÓRIO: parcial, pois o LABORATÓRIO não precisa ter um COLABORADOR para que ele exista no banco de dados.

Atributos: não há.

- **COLABORADOR realiza ENSAIO:**

Razão de cardinalidade (1:N), pois:

- 1 colaborador realiza **N** ensaios;
- 1 ensaio é realizado por **1** colaborador.

Restrição de participação:

- do COLABORADOR: parcial, pois o COLABORADOR não precisa realizar um ENSAIO para que ele exista no banco de dados;
- do ENSAIO: total, pois o ENSAIO precisa ter um COLABORADOR para que ele exista no banco de dados.

Atributos: não há.

- **ENSAIO possui METODO:**

Razão de cardinalidade (1:1), pois:

- 1 ensaio possui **1** metodo;
- 1 metodo é de **1** ensaio.

Restrição de participação:

- do ENSAIO: total, pois o ENSAIO precisa possuir um METODO para que ele exista no banco de dados;
- do METODO: parcial, pois o METODO não precisa pertencer a um ENSAIO para que ele exista no banco de dados.

Atributos: não há.

O próximo passo é elaborar o diagrama Entidade-Relacionamento, a ênfase está em representar esquemas em vez de instâncias.

Os tipos entidades são: COLABORADOR, ENSAIO, METODO, LABORATÓRIO, todos são mostrados dentro de retângulos.

Os tipos relacionamentos são: realiza, pertence, possui, todos são mostrados dentro de losangos interligados aos tipos de entidades participantes.

Atributos são mostrados em elipses conectadas aos tipos de entidades as quais estão relacionadas. Os componentes de um atributo composto são também representados em elipses porém são conectados à outra elipse do atributo ao qual ela pertence. Os atributos chaves são sublinhados, e atributos multivalorados são elipses duplas. Abaixo segue o diagrama ER do banco de dados do projeto em questão (Fig.21).

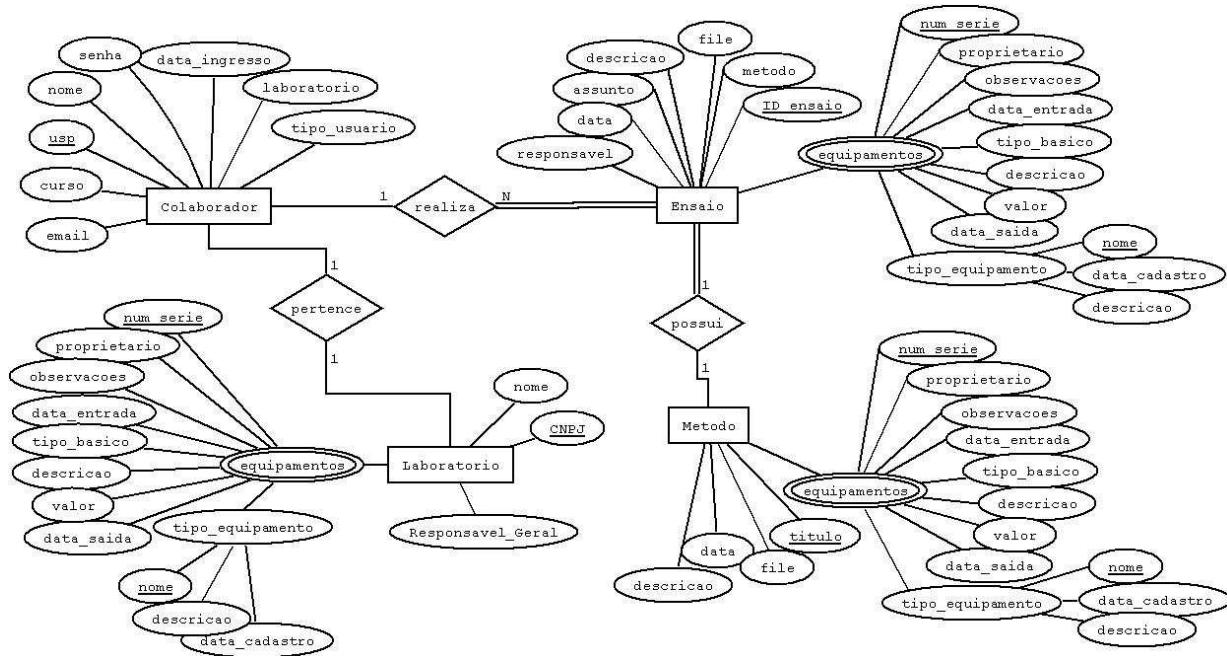


Figura 21 - Diagrama ER da base de dados do projeto

3.3.2. Modelo Relacional

Para mapearmos o modelo ER para o modelo Relacional deve-se fazer o seguinte:

Para cada entidade no modelo ER é criada um tabela T1 que inclua todos os atributos de E. Para cada atributo composto são inseridos apenas os atributos simples de cada um. Um dos atributos chaves de E deve ser escolhido como chave primaria de T1.

Para cada relacionamento regular com cardinalidade 1:1 entre entidades E1 e E2 , que geraram as tabelas T1 e T2 respectivamente, devemos escolher a chave primária de uma das relações e inserir como chave estrangeira na outra. É interessante que a chave do lado com participação parcial seja inserida como chave estrangeira no lado que tem participação total;

Para cada relacionamento regular com cardinalidade 1:N, entre E1 e E2 que geraram T1 e T2, deve-se inserir a chave primária de T1 como chave estrangeira em T2;

Para cada atributo multivalorado A1, cria-se uma tabela T1, contendo o atributo multivalorado A1, e contendo também o atributo chave C da tabela que representa a entidade ou relacionamento que contém A1. Se A1 for composto então T1 deverá conter todos os atributos de A1.

De acordo com as instruções de mapeamento do modelo ER para o modelo Relacional e após as verificações de dependência funcional e modificações necessárias para realizar a normalização temos as seguintes tabelas:

COLABORADOR (USP, nome, curso, data_ingresso, senha, tipo_usuario, laboratório, email)

ENSAIO (ID_ensaio, data_inicial, data_final, assunto, descrição, metodo, file, responsável)

METODO (data, titulo, descrição, file)

LABORATÓRIO (CNPJ, nome, responsável_geral)

EQUIPAMENTOS (numero_serie, ID_ensaio, CNPJ_laboratorio, tipo_basico, descrição, proprietário, data_entrada, data_saida, observações, valor)

TIPO_EQUIPAMENTO (nome, descrição, dataCadastro)

3.3.3. Normalização e Dependências Funcionais

Para eliminarmos os esquemas de relações não satisfatórios utilizamos as regras de normalização. No caso da base de dados em questão percebemos que o atributo equipamento era multivalorado e composto.

Para resolver o problema, primeiro pelo fato de ser multivalorado e estar presente nas entidades LABORATÓRIO e ENSAIO, criamos uma tabela EQUIPAMENTOS que contém as chaves primárias de ambas as entidades que são: CNPJ_laboratorio e ID_ensaio. Como o atributo equipamento era composto, seus atributos foram adicionados a tabela EQUIPAMENTOS. Detalhe para o atributo composto tipo_equipamento, que era uma

atributo de equipamento, foi criada uma outra tabela chamada TIPO_EQUIPAMENTO contendo seus atributos e ainda foi acrescentado na tabela EQUIPAMENTOS mais um atributo chamado tipo_basico que faz referência ao atributo nome da tabela TIPO_EQUIPAMENTO.

Após a normalização o modelo ER ficaria da seguinte forma (Fig22):

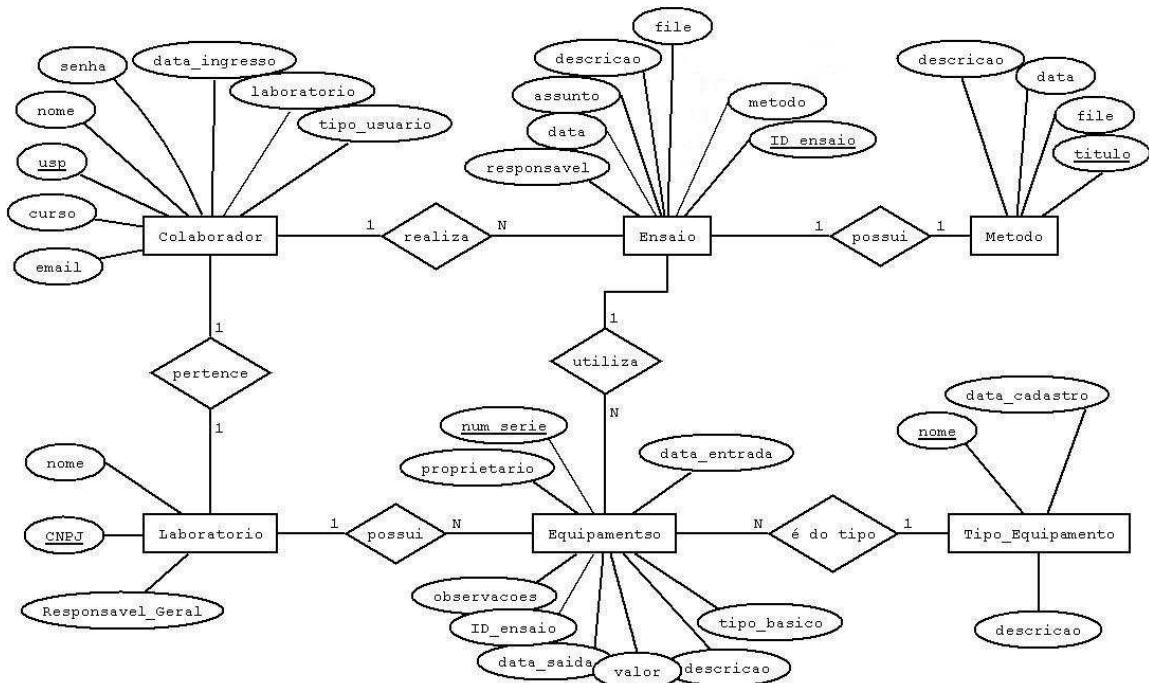


Figura 22 - MER apóis normalização

3.3.4. Comandos SQL para Criação das Tabelas

Para criar as tabelas do banco de dados em questão, segue abaixo a lista de comandos a serem utilizados em SQL.

Comando para criar a tabela COLABORADOR com seus respectivos atributos:

```
CREATE TABLE COLABORADOR (
    USP           INTEGER,
    nome          CHAR (30),
    data_ingresso DATE,
    senha         VARCHAR      NOT NULL,
```

```
tipo_usuario      CHAR(10)      NOT NULL,  
laboratório      VARCHAR,  
email            CHAR(30),  
PRIMARY KEY (USP)  
)
```

Se quisermos que um atributo incremente seu valor automaticamente para cada novo registro devemos utilizar o comando AUTO_INCREMENT.

As outras tabelas seguem a mesma lógico, por isso o código será omitido.

4. RESULTADOS

Após todo desenvolvimento teórico para criação da base de dados, foi desenvolvido um protótipo para o sistema.

Este protótipo tem como objetivo mostrar como será a navegação e quais as consultas, inserções, atualizações e remoções, ou seja, quais as operações serão realizadas no banco de dados.

Aqui serão apresentados os documentos criados, para em seguida serem mostradas as operações em cima da base de dados para os casos escolhidos.

4.1. ATRIBUIÇÕES AO USUÁRIO

A página inicial da ferramenta é onde o usuário realiza sua autenticação (Fig.23).

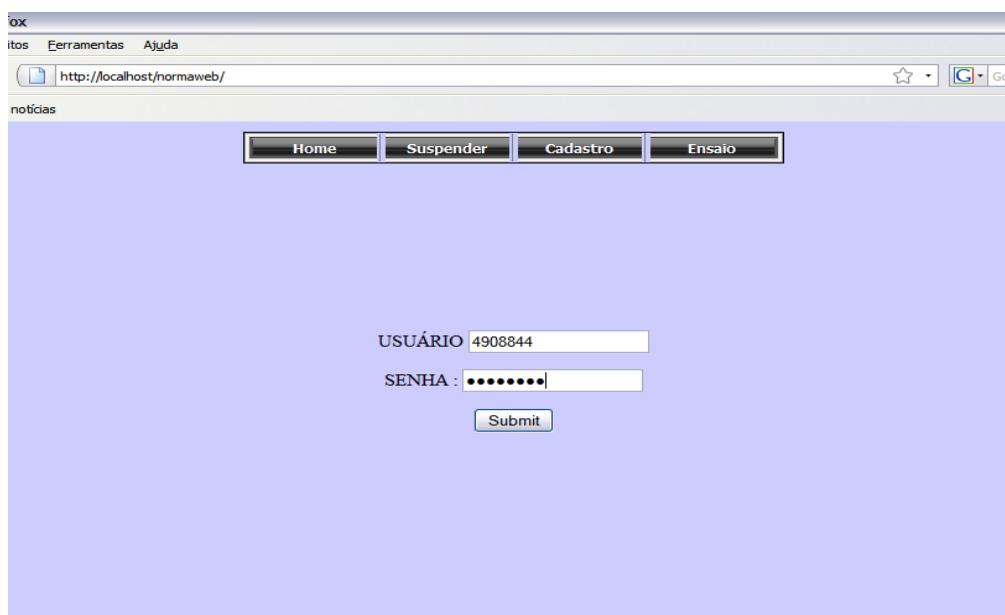


Figura 23 - Página inicial do sistema

Depois de entrar com informações pessoais de senha e nome de usuário, que por sugestão será o numero USP, o menu superior passará a ter seus links liberados. Antes disso, qualquer clique na parte superior não surtirá nenhum efeito na aplicação [11].

As operações realizadas na base de dados usando álgebra relacional são:

Consulta1 = $\pi_{\text{senha}} (\sigma_{\text{USP} = \text{usuario}} (\text{COLABORADOR}))$

Este comando retorna o valor da senha que está cadastrada na base de dados para o numero de usuário inserido no campo usuário. De posse da senha cadastrada, compara-se o valor da senha digitada pelo usuário no momento do login com a senha que está cadastrada na base de dados.

O equivalente a esses comandos utilizando-se SQL é:

```
SELECT senha
FROM COLABORADOR
WHERE USP = usuário;
```

Logo após o processo de reconhecer o usuário, o sistema entra na pagina inicial de um usuário comum (Fig24). Nessa janela o usuário pode editar algumas informações básicas sobre o seu perfil, e ao clicar no botão “Salvar”, carregará automaticamente as informações no banco de dados [11].

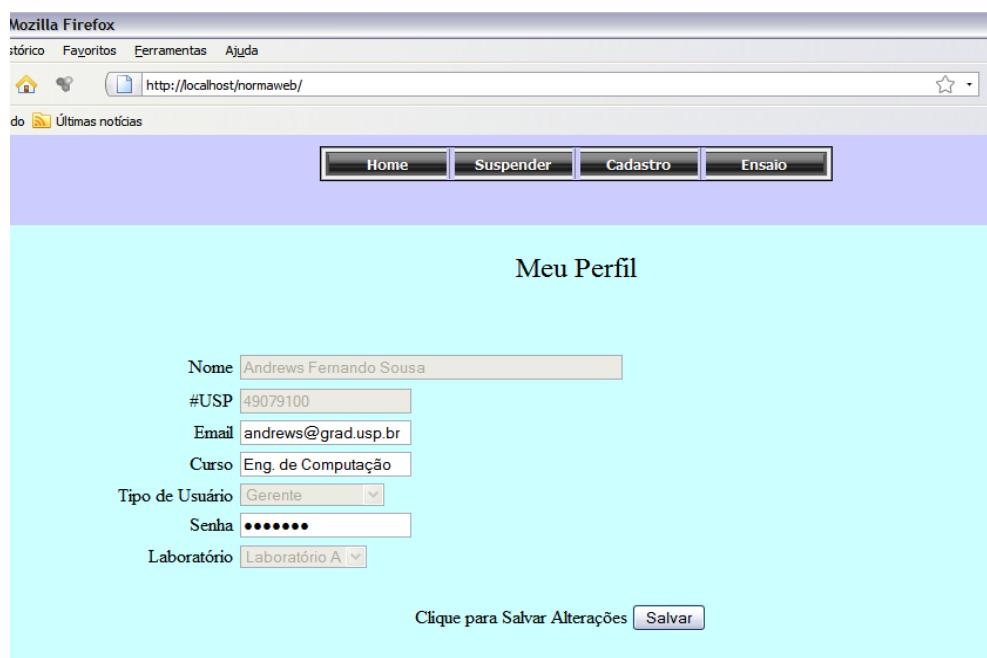


Figura 24 - Página inicial do usuário

Após o usuário clicar no botão salvar as operações realizadas são:

UPDATE COLABORADOR

SET email = Email, curso = Curso, senha = Senha

WHERE USP = Usuário ;

O comando acima atualiza os dados para o usuário que está logado no sistema, considerando que Email, Curso e Senha contêm os valores inseridos pelo usuário.

Para se criar um novo ensaio, o formulário padrão, que possui todos os atributos necessários para a entrada de um novo experimento na base de dados, é utilizado.

Preocupando-se com a integridade dos dados, os campos métodos, equipamentos e laboratórios são preenchidos após uma busca da base de dados. Com isto o usuário poderá escolher somente métodos cadastrados no sistema. O mesmo ocorre para equipamentos e laboratório.

As consultas para preencher os campos métodos, equipamentos e laboratórios presentes no formulário da Fig.25 são listadas abaixo.

Em álgebra relacional:

Consulta2 = π_{titulo} (METODO)

Consulta3 = $\pi_{\text{descricao}}$ (EQUIPAMENTOS)

Consulta4 = π_{nome} (LABORATORIO)

Em SQL:

```
SELECT titulo  
FROM METODO
```

Os comandos são os mesmos para equipamentos e laboratório, segue o modelo e por isso serão omitidos.

Untitled Document - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://localhost/normaweb/

Mais visitados Guia rápido Últimas notícias

Home Suspender Cadastro Ensaio

Ensaio
Criar
Procurar

Novo Ensaio

Assunto: Ensaio Geral de Teste

Método: Método X

Arquivos Relacionados:

Equipamento 1: Equipamento A

Equipamento 2: Equipamento B

Equipamento 3: Equipamento C

Laboratório: Laboratório A

Descrição: Ensaio para demonstração de entrada.

Clique para cadastrar >

Figura 25 - Cadastro de ensaio

No campo arquivos relacionados será guardado na base de dados o caminho escolhido pelo usuário. Por isso existe na tabela ENSAIO um atributo chamado file, que guardará o caminho especificado.

Ajuda

ost/normaweb/

Home Suspender Cadastro Ensaio

Ensaio
Criar
Procurar

Consultar por Ensaio

Data Inicial: 01/01/2001

Data Final:

USP do Autor:

Código do Ensaio:

Figura 26 - Consultar ensaio

Para os filtros da Fig.26, deverá ser controlado via código qual o tipo de consulta a ser realizado pelo banco de dados. Para cada combinação de campos deverá ser utilizada uma combinação de comandos em SQL. Segue abaixo um exemplo, considerando que o usuário preencheu todos os campos. Para os outros casos basta eliminar dos comando SQL a parte correspondente que não foi preenchida pelo usuário no formulário.

Os comandos são os seguintes, considerando que o resultado seja único, ou seja, a busca retornará somente um ensaio:

```
SELECT assunto, metodo, file, descrição, ID_ensaio
FROM ENSAIO
WHERE data_inicial = 'Data Inicial' AND data_final = 'Data Final' AND responsável = '#USP
do Autor' AND ID_ensaio = 'Código do Ensaio';
```

De posse do ID_ensaio que é o código do ensaio, pode-se realizar uma consulta na tabela EQUIPAMENTOS e verificar quais equipamentos estão sendo utilizados para o ensaio com o ID_ensaio dado. A consulta seria da seguinte forma:

```
SELECT descrição
FROM EQUIPAMENTOS
WHERE ID_ensaio = ID_ensaio;
```

Note que o ID_ensaio do lado direito da igualdade veio da consulta anterior. Com isso temos todas as informações que serão exibidas ao usuário após a consulta.

4.2. Atribuições do Gerente

Primeiramente devemos frisar que todos os gerentes herdam naturalmente as características de usuários comuns, além de ganhar algumas funções extras.

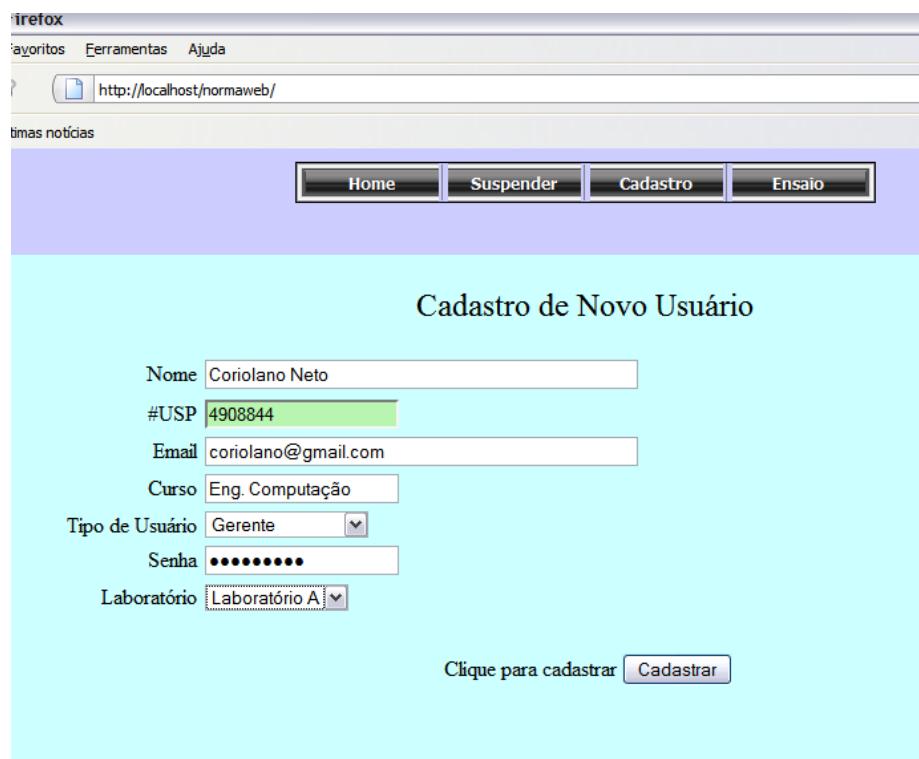
A principal função é a de gerenciamento de usuários, equipamentos e métodos. No menu superior chamado Cadastro, o gerente encontra a opção de adicionar um novo colaborador ao laboratório. Clicando em Novo Usuário, ele é levado para a página de cadastros. Essa página possui o mesmo formato da página inicial dos usuários (Fig.24), porém não existem campos impossibilitados para edição.

Além disso, ao clicar no botão cadastrar, um novo perfil de acesso é criado no sistema e não apenas edita-se um antigo, como ocorria no caso anterior.

Para a inserção de um novo usuário na base de dados utiliza-se os seguintes comando SQL, utilizando os dados da Fig27:

```
INSERT INTO COLABORADOR (USP, nome, curso, data_ingresso, senha, tipo_usuario, laboratório, email)
```

```
VALUES (4908844, 'Coriolano Neto', 'Eng. Computação', 'data do dia', 'senha', 'Gerente', 'Laboratório A', 'coriolano@gmail.com');
```



Nome: Coriolano Neto
#USP: 4908844
Email: coriolano@gmail.com
Curso: Eng. Computação
Tipo de Usuário: Gerente
Senha: *****
Laboratório: Laboratório A

Clique para cadastrar

Figura 27 - Cadastro de usuário

Como as outras operações são basicamente iguais ao cadastro de usuários os comandos SQL serão omitidos.

4.3. DIFICULDADES E LIMITAÇÕES

Durante o desenvolvimento foram encontradas dificuldades na parte de levantamento de requisitos devido ao fato de que cada laboratório possui métodos e práticas diferentes, o que dificultou a criação de um sistema para ser utilizado em qualquer laboratório.

Outra dificuldade foi o tempo demandado para se estudar e entender os requisitos exigidos pela norma ISO/IEC 170254 para garantir a qualidade do Laboratório de Instrumentação e Microeletrônica e aplicar os requisitos exigidos pela mesma no desenvolvimento do sistema.

A parte da criação da base de dados sofre algumas limitações pelo fato de depender da outra parte do projeto. Alterações futuras na base de dados poderão ser feitas para adequar mais precisamente as necessidades do laboratório em questão, mas deve-se tomar os devidos cuidados para não ser preciso alterar a parte física da base de dados, pois seriam mudanças bem mais complexas.

5. CONCLUSÃO

Pode-se concluir que a utilização de um sistema para as atividades de um laboratórios de ensaio ou calibração é fundamental para a implementação dos sistemas de gestão da qualidade, previstos pela norma ISO/IEC 17025.

As ferramentas informatizadas tem sido capaz de manter todas as exigências encontradas na norma e maximizar os resultados. Contudo, para se desenvolver um sistema para tal fim, o tempo gasto e os recursos desprendidos tem sido um fator negativo, pois quanto mais complexos maior o tempo para se recuperar o investimento.

O trabalho realizado obteve sucesso em planejar e projetar um sistema para o controle da qualidade dos ensaios e calibrações realizados pelo Laboratório de Instrumentação e Microeletrônica, pois atende aos requisitos da norma para controle de documentos e mantém a integridade dos dados e resultados obtidos em ensaios e calibrações. Porém, necessita ser implementado, adicionando mais recursos e funcionalidades específicas para o laboratório em questão. Adequando às necessidades do mesmo.

6. BIBLIOGRAFIA

- [1] CARVALHO, Alexandre Dias de – ABNT NBR ISSO/IEC 17025:2005. Chefe da equipe de avaliação de laboratórios de ensaios. adcarvalho@inmetro.gov.br
- [2] BICHO, Galdino Guttmann e VALLE, Benjamin. Guia 25 - A ISO dos Laboratórios. Revista Banas Qualidade. São Paulo, nº 85, junho/99
- [3] Dia's Homepage – website – <http://live.gnome.org/Dia>
- [4] MySQL – website – <http://www.mysqlbrasil.com.br/>
- [5] MARCON, Felipe – LIMS Para Atendimento da Norma ISO/IEC 17025
- [6] Sociedade Brasileira de Metrologia – website - <http://www.metrologia.org.br>
- [7] UNIMAR, Universidade de Marília – Apostila Banco de Dados.
- [8] Espaço Info – website – <http://www.espacoinfo.net>
- [9] CIFERRI, C. D. de Aguiar. Disciplina Banco de Dados disponível na internet pelo site <http://coteia.icmc.usp.br/mostra.php?ident=348>
- [10] CODD, Edgar Frank. Relational Model of Data for Large Shared Data Banks. Revista AMC. Vol 13, No 6.
- [11] SILVA NETO, Coriolano Martins da – Projeto de Sistema web para Controle de Ensaios Laboratoriais em Conformidade com a Norma ISO/IEC 17025.