



---

Universidade Federal de Campina Grande - UFCG  
Centro de Engenharia e Informática - CEEI  
Departamento de Engenharia Elétrica - DEE

---

Trabalho de conclusão de curso

PROJETO: **Banco de Dados para projeto do  
simulador para treinamento de  
operadores de subestação**

Aluno: Bruno Almeida de Souza

Orientadora: Fátima Vieira

Campina Grande PB  
Dezembro 2008



---

Universidade Federal de Campina Grande - UFCG  
Centro de Engenharia e Informática - CEEI  
Departamento de Engenharia Elétrica - DEE

---

**PROJETO: Banco de Dados para projeto do  
simulador para treinamento de  
operadores de subestação**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, em cumprimento parcial às exigências para obtenção do Grau de Engenheiro Eletricista.

---

Bruno Almeida de Souza

Aluno

---

Fátima Vieira

Professora

**Examinador:**

**Universidade Federal de Campina Grande - UFCG**

---

Campina Grande PB

Dezembro de 2008



Biblioteca Setorial do CDSA. Fevereiro de 2021.

Sumé - PB

## Dedicatória

Dedico este trabalho aos meus pais Manoel Leondio e Berenice, pelo amor, esforço, companheirismo, por terem abraçado esse sonho comigo e pelo sacrifício de manter meus estudos de forma tão heróica.

Aos meus amigos Francisco Sergio, André Luis e Raffael Carvalho, e a tantos outros que com companheirismo e fidelidade sempre compartilharam das minhas vitórias e momentos difíceis.

E por fim, agradeço imensamente à minha companheira, amiga, conselheira e eterna namorada, Sonia, por ter galgado ao meu lado cada degrau da minha juventude, por ter me suportado e por sempre demonstrar seu amor da forma mais sincera e intensa durante todos esses anos.

## **Agradecimentos**

Agradeço a todos os professores, funcionários e monitores da Unidade Acadêmica de Engenharia Elétrica pela dedicação, pelo trabalho árduo e profissionalismo demonstrado em suas funções.

Em especial, agradeço a Prof. Dr. Fátima Vieira pela paciência, confiança e por ter se disponibilizado em me orientar durante esse trabalho, por ter acreditado em mim e ter me ensinado com maestria durante o projeto SMS.

Aos professores Damásio Fernandes Júnior e Leimar de Oliveira pela disponibilidade e amizade durante minha graduação.

# Sumário

<b>1 - INTRODUÇÃO.....</b>	<b>6</b>
1.1 <i>POR QUE USAR BANCO DE DADOS RELACIONAL NESTE PROJETO?</i> .....	8
1.1.1 <b>O MODELO RELACIONAL DE DADOS</b> .....	8
1.2 <i>POR QUE USAR O MySQL?</i> .....	12
<b>2 - SIMULADOR PARA TREINAMENTO DE OPERADOR DE SUBESTAÇÃO DA CHESF .....</b>	<b>14</b>
2.1 <b>VISÃO GERAL</b> .....	14
2.2 <b>COMPOSIÇÃO DO SIMULADOR</b> .....	15
2.3 <b>REQUISITOS NECESSÁRIOS PARA UTILIZAÇÃO DO SIMULADOR</b> .....	15
2.4 <b>INICIALIZAÇÃO DO SIMULADOR</b> .....	15
2.5 <b>INTERAÇÃO COM O SIMULADOR</b> .....	15
<b>3 – O BANCO DE DADOS DO SIMULADOR.....</b>	<b>17</b>
3.1 <b>ESPECIFICAÇÃO</b> .....	17
3.2 <b>ESQUEMA CONCEITUAL DO BD:</b> .....	18
3.3 <b>TUPLAS</b> .....	18
3.4 <b>DICIONÁRIO DE DADOS</b> .....	20
3.5 <b>IMPLANTAÇÃO DO BD NO LIHM (MODO LOCAL E REMOTO)</b> .....	21
3.6 <b>TESTES REALIZADOS</b> .....	22
3.7 <b>IMPLEMENTAÇÃO</b> .....	24
3.7.1 <i>Ferramentas e Ambientes</i> .....	24
3.7.2 <i>Guia para manutenção do banco de dados</i> .....	24
3.8 <b>ACOPLAMENTO DO BANCO DE DADOS AO SIMULADOR</b> .....	25
<b>4 – CONCLUSÃO .....</b>	<b>26</b>
<b>5 – BIBLIOGRAFIA .....</b>	<b>27</b>
<b>ANEXO: GUIA PARA MANUTENÇÃO NO BANCO DE DADOS.....</b>	<b>28</b>
<b>ANEXO: DICIONÁRIO DE DADOS.....</b>	<b>34</b>
<i>TREINAMENTOS</i> .....	34
<i>RESULTADOS</i> .....	37
<i>CENÁRIOS ASSOCIADOS</i> .....	38
<i>INSTALAÇÕES</i> .....	39
<i>OPERADORES</i> .....	40
<i>TUTORES</i> .....	41
<i>RESULTADOS INDIVIDUAIS</i> .....	42
<i>RESULTADOS COLETIVOS</i> .....	44
<i>EQUIPAMENTOS</i> .....	46
<i>DISPOSITIVOS</i> .....	47
<i>ESTADO DO TREINAMENTO</i> .....	48
<i>TIPO DO TREINAMENTO</i> .....	49
<i>ESTADO DO DISPOSITIVO</i> .....	50
<i>ESTADO DO EQUIPAMENTO</i> .....	51
<i>PAINEL</i> .....	52
<i>RECURSOS</i> .....	53
<i>ESTADO DO PAINEL</i> .....	54
<b>GLOSSÁRIO .....</b>	<b>55</b>

# 1 - Introdução

O objetivo deste trabalho foi o desenvolvimento de um banco de dados para dar suporte a uma interface de simulação de treinamentos para operadores CHESF, onde um tutor responsável pelo treinamento poderá simular uma manobra em uma subestação, um painel, etc., ou seja poderá simular vários cenários e situações corriqueiras do sistema elétrico de potência. A grande vantagem deste simulador é que haverá um padrão para as avaliações onde o tutor poderá verificar o tempo que o operador leva para determinadas atividades, se eles seguirem os procedimentos que garantem a segurança das instalações e equipamentos. Este programa de simulação avalia operadores de várias unidades do complexo CHESF, verificando seus conhecimentos sobre o sistema. O registro de treinamentos realizados com funcionários da CHESF é atualmente realizado em formulários de papel o que dificulta sua consulta para fins de controle de qualidade. Este projeto tem como objetivo dar suporte a interface do simulador que será apresentada neste trabalho. O armazenamento dos dados diretamente em um sistema de informação evitará erros introduzidos durante as anotações nos formulários e durante sua transcrição posterior para planilhas, além de facilitar sua consulta por diferentes setores da empresa.

O banco de dados, utilizado para o Sistema Gerenciador de Banco de Dados (SGBD) **MySQL**, foi o objetivo deste trabalho. Por sua vez, o projeto da interface será concebido e implementado pelos alunos de mestrado do LIHM, assim como sua integração ao banco de dados.

Apresentaremos o modelo entidade relacional (MER) como forma de visualização da forma do banco de dados e o dicionário de dados para consulta de como funciona cada campo do banco de dados para entendermos como funciona a lógica do banco de dados.

Neste projeto utilizaremos o banco de dados relacional na terceira forma normal, para garantir um nível maior de padronização, esta forma de normalização esta mostrada no item 1.1.1.

Na figura 1 podemos observar como os usuários do BD interagem com o a interface do BD, no caso a interface do simulador. Os operadores são usuários finais eles apenas interagem com os cenários preparado pelos tutores. Os tutores são usuários sofisticados eles podem alterar o estado de um cenário desde que o programador de aplicações já tenha desenvolvido este cenário. Temos o SGBD que faz a ligação da interface com o BD.

## Visão Geral da Estrutura do Sistema BD

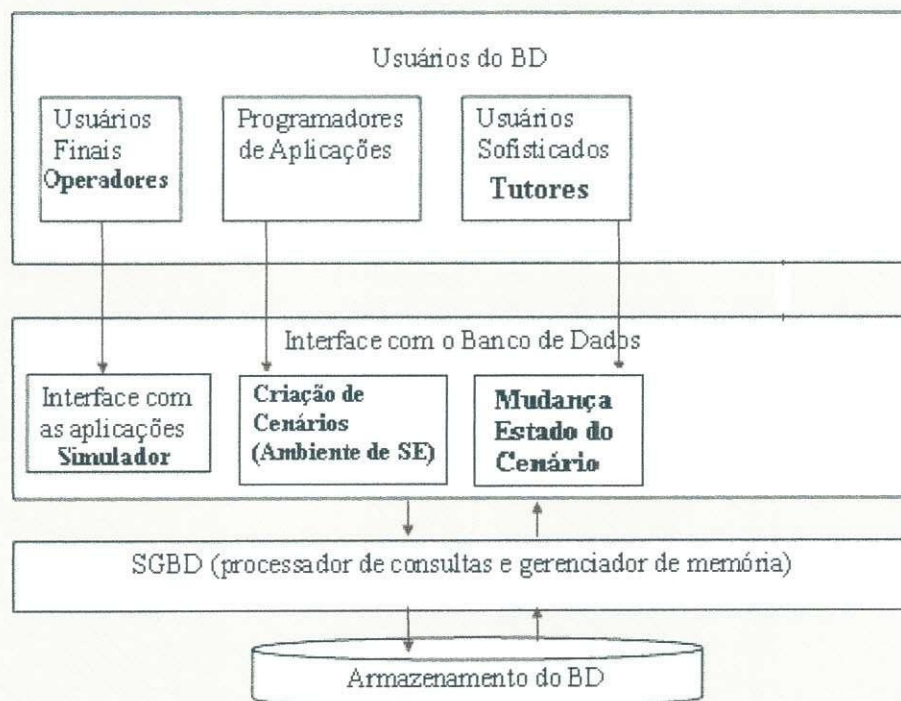


Figura 1: Visão geral da estrutura do sistema BD.



## ***1.1 Por que usar banco de dados relacional neste projeto?***

O modelo relacional apareceu devido às seguintes necessidades: aumentar a independência de dados nos sistemas gerenciadores de banco de dados; prover um conjunto de funções apoiadas em álgebra relacional para armazenamento e recuperação de dados. O Modelo relacional revelou-se ser o mais flexível e adequado para solucionar os vários problemas que se colocam no nível da concepção e implementação da base de dados. A estrutura fundamental do modelo relacional é a relação (tabela). Uma relação é constituída por um ou mais atributos (campos) que traduzem o tipo de dados armazenado. Cada instância do esquema (linha) é chamada de tuplas (registro). O modelo relacional não tem caminhos pré-definidos para se fazer acesso aos dados como nos modelos que o precederam. O modelo relacional implementa estruturas de dados organizadas em relações. Porém, para trabalhar com essas tabelas, algumas restrições precisaram ser impostas para evitar aspectos indesejáveis, como: Repetição de informação, incapacidade de representar parte da informação e perda de informação. Essas restrições são: integridade referencial, chaves e integridade de junções de relações.

No banco de dados da interface do simulador tentamos deixá-lo na terceira forma normal para aproveitar as vantagens do banco de dados relacional deixando o programador da interface mais livre com relação ao armazenamento de dados.

### **1.1.1 O Modelo Relacional de Dados**

Normalização de banco de dados e as três principais formas normais.

Os exemplos apresentados utilizarão telas do Microsoft Access e o arquivo de exemplos Northwind.mdb, o qual é instalado juntamente com o Microsoft Access. Este arquivo está disponível, por padrão, no seguinte caminho:

<C:\Arquivos de programas\Microsoft Office\Office\Samples>

Não vamos utilizar o exemplo do banco de dados no qual estamos trabalhando, pois, acreditamos que este banco de dados se apresenta de forma mais clara de ser entendida pelo usuário sem muita experiência em BD.

Porém os princípios básicos do modelo relacional aplicam-se a qualquer banco de dados baseado no modelo relacional de dados. Estes bancos de dados são algumas vezes denominados: SGBDR - Sistemas Gerenciadores de Banco de Dados Relacionais.

### **Normalização de tabelas**

Objetivo: O objetivo da normalização é evitar os problemas provocados por falhas no Projeto do Banco de Dados, bem como eliminar a "mistura de assuntos" e as correspondentes repetições desnecessárias de dados. Uma Regra de Ouro que devemos observar quando do Projeto de um Banco de Dados baseado no Modelo Relacional de dados é a de "**não misturar assuntos em uma mesma Tabela**". Por exemplo, na Tabela Clientes devemos colocar somente campos relacionados com o assunto Clientes. Não devemos misturar campos relacionados com outros assuntos, tais como Pedidos, Produtos, etc. Essa "Mistura de Assuntos" em uma mesma tabela acaba por gerar repetição desnecessária dos dados bem como inconsistência dos dados.

O Processo de Normalização aplica uma série de Regras sobre as Tabelas de um Banco de Dados, para verificar se estas estão corretamente projetadas. Embora existam 5 formas normais (ou regras de Normalização), na prática usamos um conjunto de 3 Formas Normais.

Normalmente após a aplicação das Regras de Normalização, algumas tabelas acabam sendo divididas em duas ou mais tabelas, o que no final gera um número maior de tabelas do que o originalmente existente. Este processo causa a simplificação dos atributos de uma tabela, colaborando significativamente para a estabilidade do modelo de dados, reduzindo-se consideravelmente as necessidades de manutenção. Vamos entender o Processo de Normalização na Prática, através de exemplos.

### **Primeira Forma Normal:**

"Uma Tabela está na Primeira Forma Normal quando seus atributos não contém grupos de Repetição".

Por isso dissemos que uma Tabela que possui Grupos de Repetição não está na Primeira Forma Normal. Considere a estrutura da Tabela Indicada na Próxima Figura:

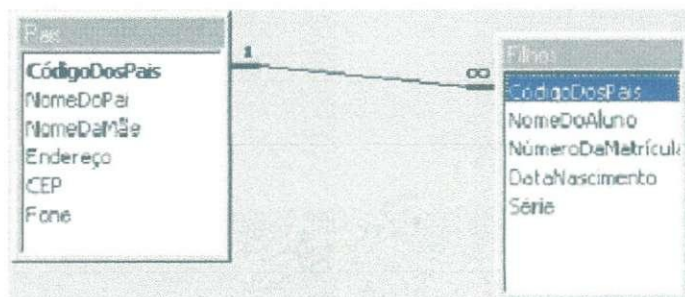
Aluno Não Normaliza...
NomeDoAluno
NúmeroDaMatricula
DataNascimento
Série
NomeDoPai
NomeDaMãe

Tabela que não está na Primeira Forma Normal

Uma tabela com esta estrutura apresentaria diversos problemas. Por exemplo, se um casal tiver mais de um filho, teremos que digitar o Nome do Pai e da Mãe diversas vezes, tantas quantos forem os filhos. Isso forma um Grupo de Repetição. Além do mais pode ser que por erro de digitação o Nome dos Pais não seja digitado exatamente igual todas às vezes, o que pode acarretar problemas na hora de fazer pesquisas ou emitir relatórios.

Este problema ocorre porque "Misturamos Assuntos" em uma mesma tabela. Colocamos as informações dos Pais e dos Filhos em uma mesma tabela. A resolução para este problema é simples: Criamos uma tabela separada para a Informação dos Pais e Relacionamos a tabela Pais com a Tabela Filhos através de um relacionamento do tipo Um para Vários, ou seja, um casal de Pais pode ter vários Filhos.

Observe na figura abaixo as duas tabelas: Pais e Filhos, já normalizadas.



Informações sobre Pais e Filhos em Tabelas Separadas

As duas tabelas Resultantes da Aplicação da Primeira Forma Normal: Pais e Filhos estão na Primeira Forma Normal, a Tabela Original, a qual misturava informações de Pais e Filhos, não estava na Primeira forma Normal

### Segunda Forma Normal:

Ocorre quando a chave Primária é composta por mais de um campo. Neste caso, devemos observar se todos os campos que não fazem parte da chave dependem

de todos os campos que compõem a chave. Se algum campo depender somente de parte da chave composta, então este campo deve pertencer à outra tabela. Observe o Exemplo Indicado na Tabela da Figura abaixo:

Cursos
NúmeroDaMatrícula
CódigoDoCurso
Avaliação
DescriçãoDoCurso

Tabela com uma Chave Primária Composta. Não está Na Segunda Forma Normal

A Chave Primária Composta é formada pela combinação dos Campos "NúmeroDaMatrícula" e "CódigoDoCurso". O Campo Avaliação depende tanto do CódigoDoCurso quanto do NúmeroDaMatrícula, porém o campo DescriçãoDoCurso, depende apenas do CódigoDoCurso, ou seja, dado o código do curso é possível localizar a respectiva descrição, independentemente do NúmeroDaMatrícula. Com isso temos um campo que não faz parte da Chave Primária e depende apenas de um dos campos que compõem a chave Primária Composta, por isso que dizemos que esta tabela não está na Segunda Forma Normal.

A Resolução para este problema também é simples: "Dividimos a Tabela que não está na Segunda Forma Normal em duas outras tabelas, conforme indicado pela figura abaixo, sendo que as duas tabelas resultantes estão na Segunda Forma Normal".

Avaliações	Cursos
NúmeroDaMatrícula	CódigoDoCurso
CódigoDoCurso	DescriçãoDoCurso
Avaliação	

Informações sobre Avaliações e Cursos em Tabelas Separadas

**Obs.:** A Distinção entre a Segunda e a Terceira forma normal, que veremos logo em seguida, muitas vezes é confusa. A Segunda Forma normal está ligada à ocorrência de Chaves Primárias compostas.

### Terceira Forma Normal:

Na definição dos campos de uma entidade podem ocorrer casos em que um campo não seja dependente diretamente da chave primária ou de parte dela, mas sim dependente de um outro campo da tabela, campo este que não faz parte da Chave Primária.

Quando isto ocorre, dizemos que a tabela não está na Terceira Forma Normal, conforme indicado pela tabela da figura abaixo:

Funcionários não n...
<b>NúmeroDaMatricula</b>
NomeFuncionário
CódigoDoCargo
DescriçãoDoCargo

Tabela com um Campo dependente de Outro campo que não a Chave Primária. Não está na Terceira Forma Normal

Observe que o Campo **DescriçãoDoCargo** depende apenas do Campo **CódigoDoCargo**, o qual não faz parte da Chave Primária. Por isso dizemos que esta tabela não está na terceira forma normal. A Solução deste problema também é simples. Novamente basta dividir a tabela em duas outras, conforme indicado pela figura a seguir. As duas tabelas resultantes estão na Terceira Forma Normal.

Funcionários	Cargos
<b>NúmeroDaMatricula</b>	<b>CódigoDoCargo</b>
NomeFuncionário	DescriçãoDoCargo
CódigoDoCargo	

Tabelas Resultantes que estão na Terceira Forma Normal

Com isso podemos concluir que como resultado do Processo de Normalização, iremos obter um número maior de tabelas, porém sem problemas de redundância e inconsistência dos dados.

## 1.2 Por que usar o MySQL?

Embora tenhamos usado o Microsoft Access para desenvolver o sistema inicial do banco de dados, a implementação no servidor do LIHM foi feita no MySQL, e foram vários os motivos desta mudança. O principal motivo é o fato de

o software ser livre seguindo a mesma especificação do simulador descrita anteriormente, também foi levado em consideração que dentre os bancos de dados livre disponíveis no mercado o MySQL é aquele que apresenta maior robustez para grande volume de dados, melhor interface gráfica e disponibilidade para trabalhar com o Linux. Para a conversão do Access para o MySQL foi usado o software BDconm, que também é um software livre que pode baixar diretamente no site da BDconm ou no [WWW.baixeaki.com.br](http://WWW.baixeaki.com.br). Este software fez a conversão de um banco de dados em Access para MySQL com todos os relacionamentos e todos os dados das tabelas.

Fonte: ([www.mysqlbrasil.com.br](http://www.mysqlbrasil.com.br))

## 2 - Simulador para Treinamento de Operador de Subestação da CHESF

### **2.1 Visão Geral**

A Companhia Hidroelétrica do São Francisco – CHESF, realiza em uma de suas subestações em Campina Grande (SE\_CGD) algumas tarefas típicas, como, por exemplo, manobras sobre dispositivos de proteção e re-energização da subestação. Devido à importância dessas atividades, é de interesse da empresa fornecer condições de capacitação e treinamento para os seus operadores. Uma forma de se fazer isso sem que haja manipulação dos equipamentos reais é o uso de uma simulação virtual do ambiente de trabalho.

O simulador proposto para o caso descrito fornece um ambiente virtual que representa todos os objetos utilizados pelo operador da subestação durante a realização de uma tarefa. Dentre estes objetos se encontram os painéis com suas chaves e mostradores para atuação sobre equipamentos em campo, tais objetos estão distribuídos e taxados de maneira análoga à sala de operações real. Desta forma, acrescentando a capacidade ao usuário de deslocar-se livremente pelo ambiente e interagir com os dispositivos possíveis, obtemos um ambiente fiel à realidade e capaz de ser usado para treinamento de operadores em diversos cenários.

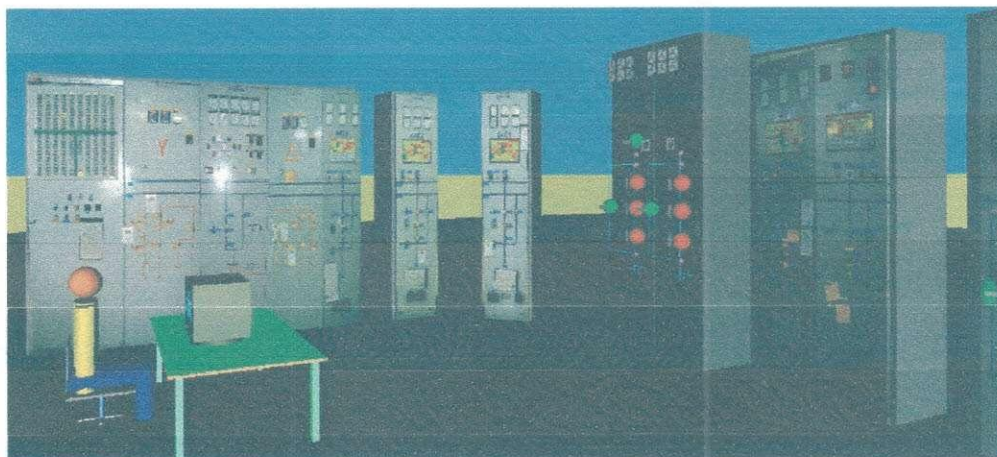


Figura 1 - Visão da sala de controle no simulador

## 2.2 Composição do Simulador

O simulador é composto de uma descrição em Realidade Virtual da sala de comando, compreende tanto painéis de controle como também uma representação do controle feito através de terminais de computadores. Esta representação é visualizada através do software FreeWRL e está ligado a programas escritos em Java que implementam a animação no simulador.

## 2.3 Requisitos Necessários para Utilização do Simulador

- Computador com sistema operacional Linux.
- Software FreeWRL.
- Máquina Virtual Java.
- Mouse.
- Auto-falantes.

## 2.4 Inicialização do Simulador

A aplicação que implementa o simulador pode ser inicializada sobre o sistema operacional Linux digitando o comando abaixo a partir de um terminal.

```
./SalaControleChesf
```

Este comando abre automaticamente o mundo virtual através do FreeWRL e executa a aplicação Java.

## 2.5 Interação com o Simulador

No ambiente representado no simulador o usuário é livre para se mover e atuar nos pontos sensíveis dos painéis, isto é, nas chaves que representam o comando dos disjuntores e seccionadores. Tudo isso é realizado através de comandos nos mouses, como é descrito nos dois tópicos a seguir.

### Deslocamento no Ambiente Virtual

O deslocamento no mundo virtual pode ser feito com mudança de pontos de vista ou através da utilização do mouse, no primeiro caso utiliza-se o menu



Navigate/Next Viewpoint de forma que a próxima vista da seqüência será exibida, isso é feito de maneira cíclica, esta funcionalidade é inerente ao software de visualização FreeWRL.

Para o deslocamento livre no ambiente virtual, com o auxílio do mouse, a interação se dá da seguinte forma:

- Posicione o cursor do mouse no centro da tela.
- Mantenha pressionado o botão esquerdo do mouse.
- Movimento o cursor na tela de tal modo que:
  - Movimento do cursor para cima equivale a caminhar para frente.
  - Movimento do cursor para baixo é equivalente a caminhar para trás.
  - Movimento do cursor para a esquerda equivale a virar-se para esquerda.
  - Movimento do cursor para a direita equivale a virar-se para direita.

### **Abertura ou Fechamento de Chaves**

A abertura ou fechamento de chaves nos painéis do simulador é feito através do mouse seguindo os passos:

- Posicione o cursor do mouse sobre a chave a ser fechada ou aberta.
- Clique com o botão esquerdo do mouse sobre a chave.

Seguindo este passos mudaremos o estado da chave.

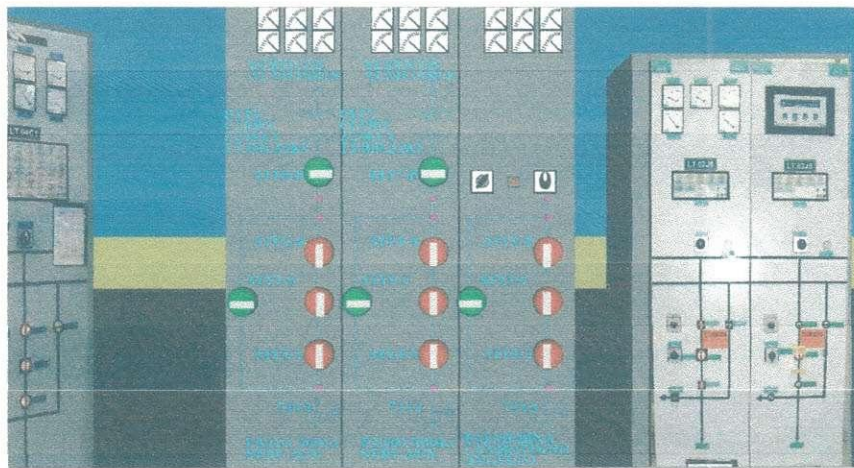


Figura 2 - Vista frontal de um dos painéis. Destaque para as chaves e suas tarjas de referência.

## 3 – O Banco de Dados do Simulador

### 3.1 Especificação

#### 3.1.1 Modelo de dados

O diagrama MER como colocado na introdução é uma forma de visualização dos relacionamentos do banco de dados. Este banco de dados foi feito inicialmente no Microsoft Access.

Para chegar a terceira forma normal como apresentada na figura 3, primeiro dividimos o banco de dados conforme tabela 1, em duas grandes partes as tabelas que seriam necessárias para atender as necessidades dos treinamentos e as tabelas que seriam necessárias para atender as necessidades de armazenamento dos cenários.

Ao dividirmos estes dois grupos criamos as tuplas que são apresentadas no item 3.3, e a partir desta procuramos normalizar o banco deixando este na terceira forma normal.

Tabelas TREINAMENTOS	Tabelas CENARIOS
Treinamentos	Cenários_Associados
Tutores	Instalações
Operadores	Equipamentos
Resultados	Dispositivos
Resultados Individuais	Painel
Resultados Coletivos	Estado Equipamento
	Estado Dispositivo
	Estado Painel

Tabela 1: Grupos de necessidades de armazenamento. Grupo 1: Tabelas relacionadas ao treinamento. Grupo 2: Tabelas relacionadas aos cenários.

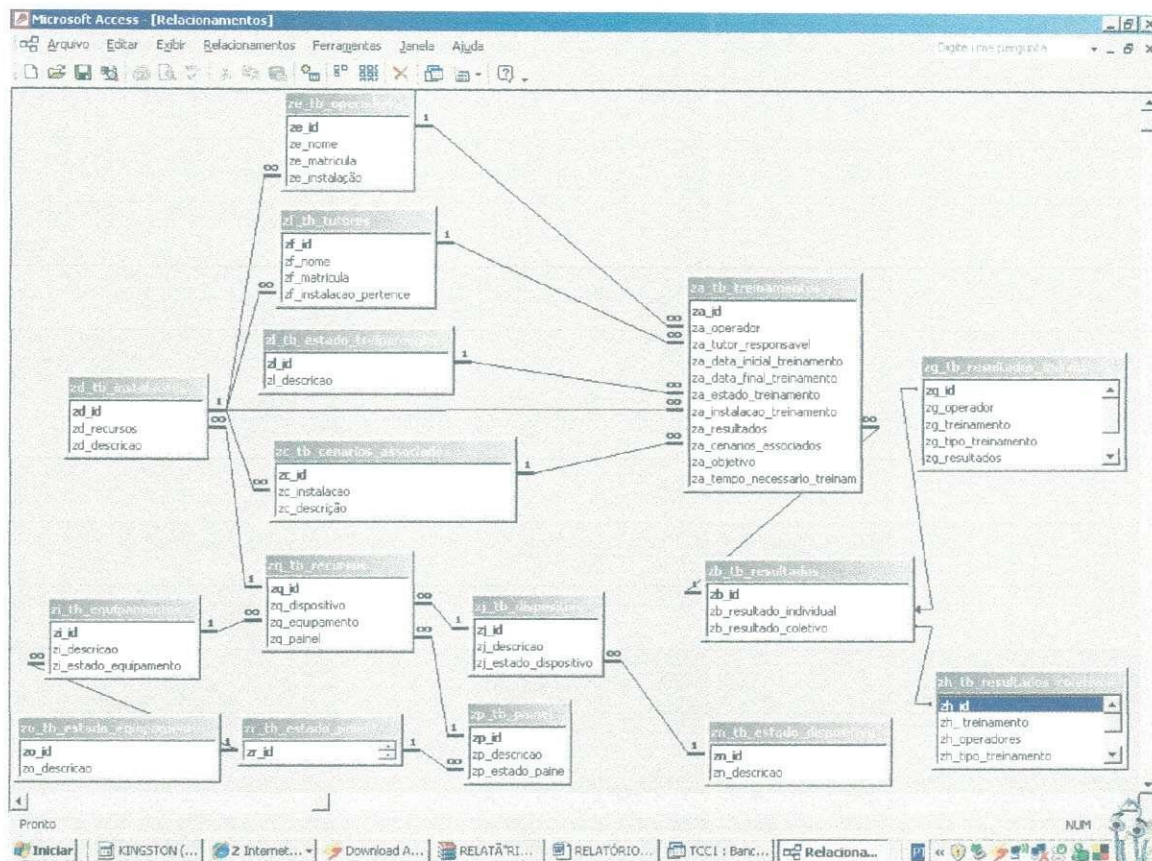


Figura 3 Diagrama MER Original

### 3.2 Esquema Conceitual do BD:

Como resultado das especificações do BD foi elaborado um Esquema Conceitual apresentado na forma de tuplas. Este esquema foi elaborado a partir do conjunto de relações-base que segue. Na notação empregada para descrever estas relações são destacadas as chaves de acesso: primária (PK- Primary Key) e secundária (FK- Foreign Key).

### 3.3 Tuplas

Cada linha formada por uma lista ordenada de colunas representa um registro, ou tupla. Os registros não precisam conter informações em todas as

colunas, podendo assumir valores nulos quando assim se fizer necessário. Resumidamente, um registro é uma instância de uma tabela, ou entidade.

A seguir as tuplas do nosso banco de dados:

za\_tb\_treinamentos(za\_id(PK), za\_cenarios\_associados(FK),  
za\_tutor\_responsavel(FK), za\_instalacao\_treinamento(FK),  
za\_data\_inicial\_treinamento, za\_data\_final\_treinamento, za\_turma\_operador(FK),  
za\_estado\_treinamento(FK), za\_resultados(FK), za\_objetivo,  
za\_tempo\_necessario\_treinamento\_minuto );

zb\_tb\_resultados(zb\_id(PK), zb\_resultado\_individual(FK),  
zb\_resultado\_coletivo(FK));

zc\_tb\_cenarios\_associados(zc\_id(PK), zc\_instalacao(FK), zc\_descricao);

zd\_tb\_instalacoes(zd\_id(PK), zd\_recursos(FK), zd\_descricao);

ze\_tb\_operadores(ze\_id(PK), ze\_nome, ze\_matricula, ze\_instalacao(FK),);

zf\_tb\_tutores(zf\_id(PK), zf\_nome, zf\_matricula, zf\_instalacao\_pertence(FK));

zg\_tb\_resultados\_individuais(zg\_id(PK), zg\_operador(FK), zg\_treinamento(FK),  
zg\_tipo\_treinamento(FK), zg\_resultados,  
zg\_tempo\_utilizado\_treinamento\_minuto);

zh\_tb\_resultados\_coletivos(zh\_id(PK), zh\_treinamento(FK), zh\_operadores(FK),  
zh\_tipo\_treinamento(FK), zh\_resultados(FK),  
zh\_tempo\_utilizado\_treinamento\_minuto);

zi\_tb\_equipamentos(zi\_id(PK), zi\_descricao, zi\_estado\_equipamento(FK));

zj\_tb\_dispositivo(zj\_id(PK), zj\_descricao, zj\_estado\_dispositivo(FK));

zl\_tb\_estado\_treinamento(zl\_id(PK), zl\_descricao);

zm\_tb\_tipo\_treinamento(zm\_id(PK), zm\_descricao);

zn\_tb\_estado\_dispositivo(zn\_id(PK), zn\_descricao);

zo\_tb\_estado\_equipamento(zo\_id(PK), zo\_descricao);

zp\_tb\_painel(zp\_id(PK), zp\_descricao);

zq\_tb\_recursos(zq\_id(PK), zq\_descricao);

zr\_tb\_estado\_painel(zr\_id(PK), zr\_descricao)

### **3.4 DICIONÁRIO DE DADOS**

A partir de agora iremos apresentar o dicionário de dados informado todas as tabelas e campos do banco de dados que tipo de variável o campo armazena e as chaves estrangeiras e primarias das tabelas este dicionário é muito importante neste trabalho, pois é a partir dele que percebemos o melhor relacionamento das tabelas.

O objetivo é conseguir fazer um banco de dados que esteja livre de anomalias, juntamente com seus principais métodos utilizados para que isso seja possível, usando a normalização, fazendo o dicionário de dados e a montagem do próprio banco.

Deste modo o dicionário de dados nos auxilia na visualização de como ficara o banco e seus relacionamentos entre tabelas.

O dicionário de dados se encontra em anexo no final deste trabalho.

### 3.5 Implantação do BD no LIHM (Modo Local e Remoto)

Uma especificação é que o banco de dados funcionasse tanto de forma local como web, neste caso o MySql com a interface PHPMYADIM funcionou plenamente, pois ao instalarmos o PHPMYADIM o programa instala um servidor Apache no PC e disponibiliza o banco de dados on line.

Colocamos o banco recém criado no servidor do LIHM e acessamos ele tanto localmente quanto remotamente através do endereço <http://150.165.61.83/phpMyAdmin/> utilizando o nome de usuário lihm e a senha disponibilizada pelos alunos de mestrado do LIHM, como mostrado na figura 4 (página de entrada do BD) e na figura 5 (BD conectado via web).

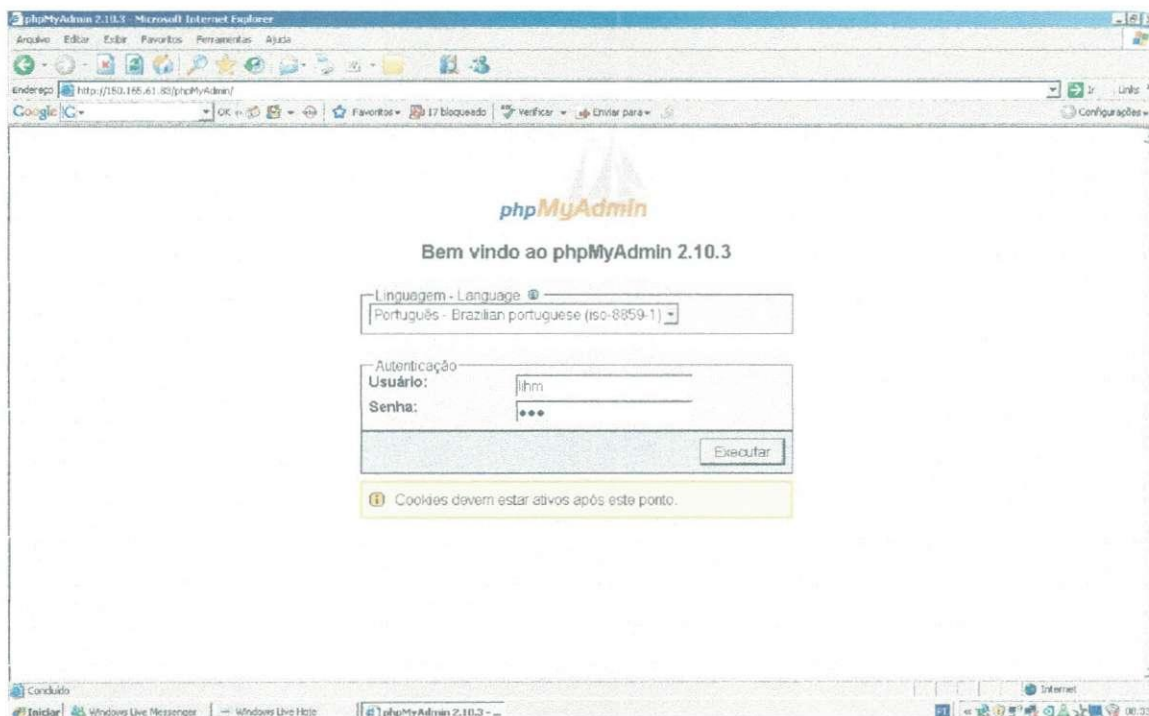
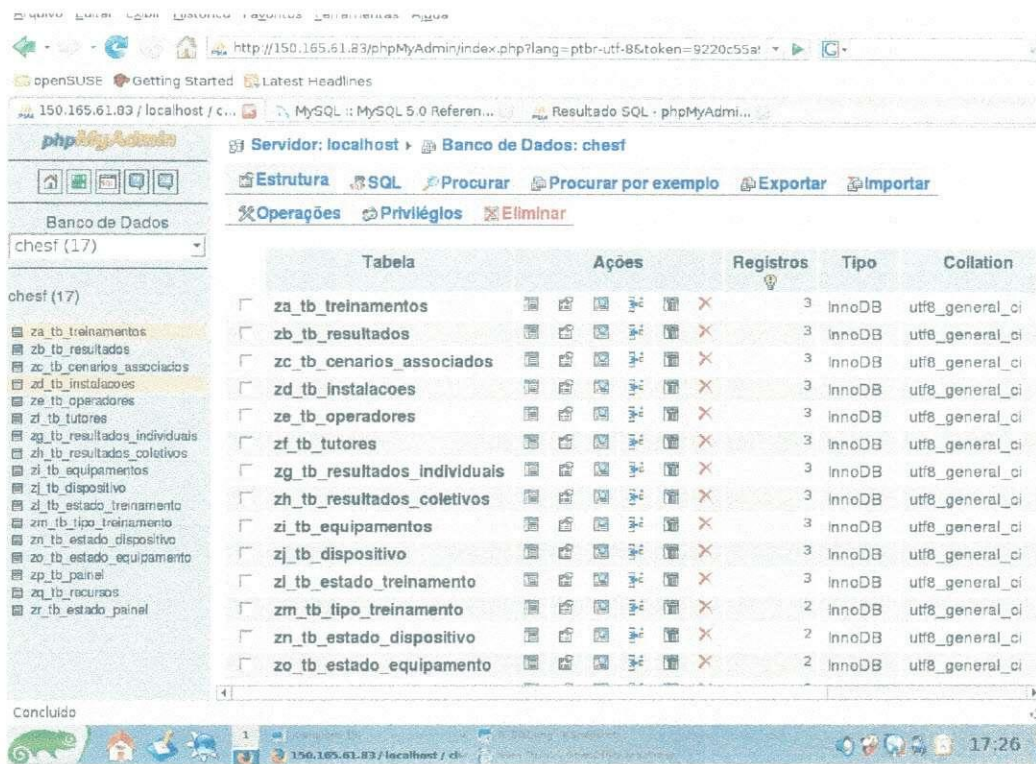


Figura 4: Tela inicial do banco de dados na web.



5 -

figura 5: BD instalado acessado remotamente via web.

### 3.6 Testes Realizados.

Após a implantação do BD no LIHM, fizemos algumas consultas para verificar se os relacionamentos criados estavam funcionando corretamente, uma das consultas é mostrada na figura 6, e o resultado obtido apresentado na figura 7, onde verificamos que os relacionamentos entre as tabelas esta funcionando corretamente.

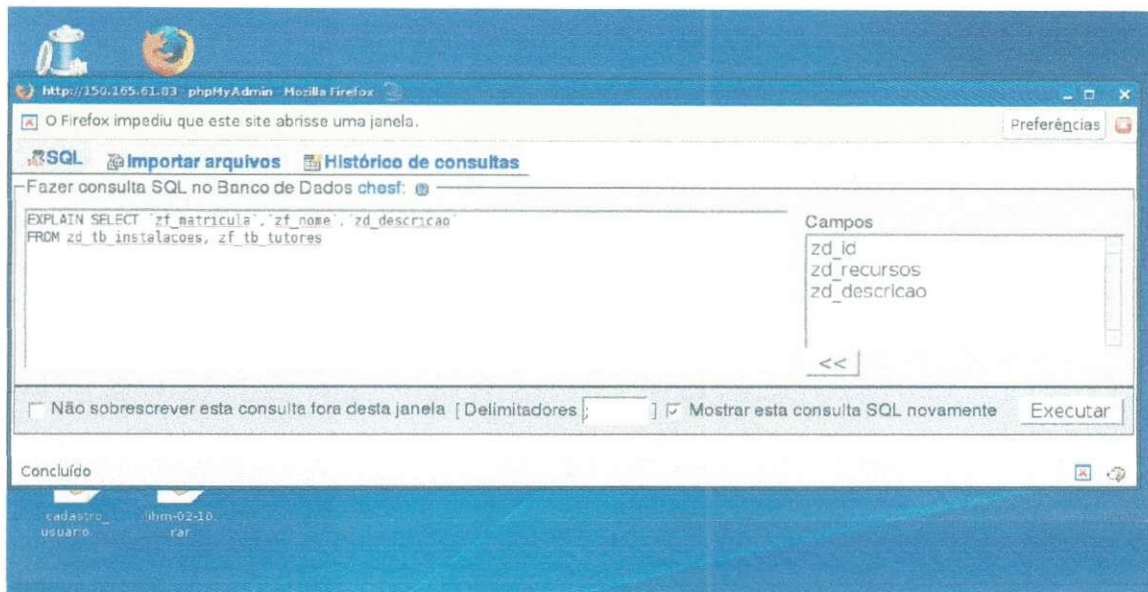


Figura 6 – Consulta no MySQL ( Verificar funcionalidade do relacionamento).

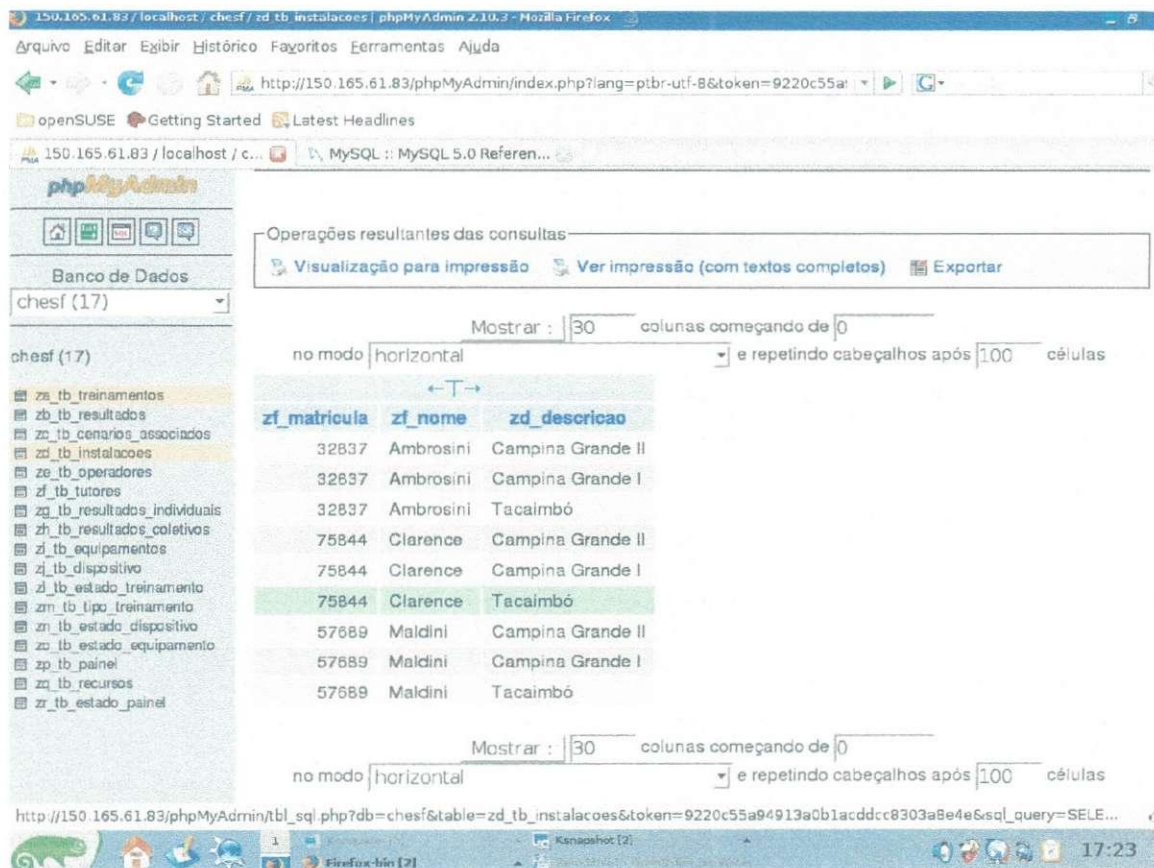




Figura 7 – Resultado da consulta no BD feita em MySql (Comprovando a funcionalidade dos relacionamentos do BD).

## **3.7 Implementação**

### *3.7.1 Ferramentas e Ambientes*

O banco de dados foi inicialmente concebido em Microsoft Access, mas devido a necessidade de se usar software livre, mudamos para o MySql. Para não reescrevermos todo o banco utilizamos o software BDconm que converte banco de dados entre estas linguagens, este software converte banco de dados como um todo para a nova linguagem, ou seja, tabelas, relacionamentos e campos que já estejam preenchidos.

O usuário do banco de dados irá perceber uma nomenclatura das tabelas sempre começando com a letra z e outra letra do alfabeto gerando uma seqüência, isto é uma codificação muito usada em banco dados relacional com grandes quantidades de tabelas e relacionamentos, pois a medida que a quantidade de tabelas e relacionamentos vão crescendo em um banco de dados se não for criada uma codificação como esta será muito difícil consultar o BD. Para seguir esta codificação basta o usuário ao criar uma nova tabela observar qual letra virá na seqüência.

### **3.7.2 Guia para manutenção do banco de dados**

Há duas maneiras de se fazer manutenções no BD, a primeira e menos usual é fazer as atualizações em Access e usar o software conversor BDconm para levar o novo banco ao servidor do LIHM. Esta forma não é recomendada pois os dados já enviados pela interface serão perdidos.

A segunda maneira é ir diretamente ao banco instalado no servidor e através da interface do PHPMYADMIN no laboratório do LIHM ou via web através do link <http://150.165.61.83/phpMyAdmin/> e fazer as manutenções necessárias no

banco de dados, a seguir faremos uma lista de procedimentos que devem ser adotadas para manutenção do banco de dados.

Em anexo está o guia de manutenção do banco de dados.

### 3.8 Acoplamento do banco de dados ao simulador

Este acoplamento será feito pelos alunos de mestrado do LIHM.

Como é bastante comum em construção de banco dados a primeira forma de um banco de dados não ter todas as informações que a interface necessita, mas sim as principais. Por isso que foi de suma importância o guia de manutenção do BD, pois este dará suporte para que outras pessoas complementem este banco chegando assim a uma versão final que atenda todas as necessidades que a interface necessite.

Nas figuras 8 e 9, temos a percepção dos níveis que cada usuário terá, podemos notar na figura 9, que os operadores e tutores ficam em um nível onde eles tem contato apenas com o simulador, podendo mudar estado das chaves ou preparar um cenário, operadores e tutores respectivamente. Já os programadores do simulador poderão criar novas subestações, novos cenários estando estes em nível mais elevado. No final poderá obter relatórios dos treinamentos tanto individualmente como coletivamente, verificando se os usuários seguiram os passos do treinamento e quanto tempo foi utilizado para completar as atividades.



Figura 8: Visão Geral da Estrutura do BD

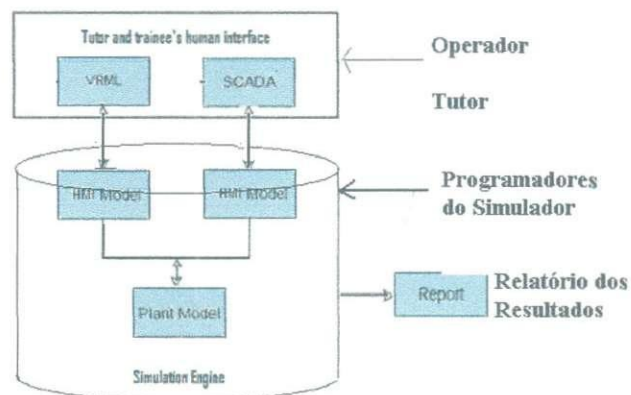


Figura 9: Níveis de utilização da aplicação

## 4 – Conclusão

Neste trabalho inicialmente foi feito um estudo de quais dados o simulador iria exigir que o banco de dados armazenasse e a melhor forma de fazer este armazenamento levando em consideração um banco de dados relacional.

O projeto realizado anteriormente no LIHM (Projeto SMS) foi de fundamental importância na realização deste trabalho, pois foi à base do conhecimento aplicado neste projeto.

Este trabalho foi uma partida para um projeto muito maior. Ele fornecerá os subsídios iniciais ao projeto maior que é o simulador, no entanto, este necessitará de complementos, pois um banco de dados nunca fica perfeito na sua forma inicial, sendo assim, fizemos um guia para manutenção do banco de dados que facilitará para que novos integrantes do projeto dêem continuidade ao mesmo de forma mais rápida sem a necessidade de refazer as mesmas pesquisas de usabilidade do MySql.

Este banco de dados dará suporte à interface de simulação de treinamentos para operadores CHESF, melhorando a forma de avaliar como os operadores enfrentam as simulações de fatos que podem ocorrer no seu dia a dia e de que forma atuarão para que minimizem as perdas para o sistema elétrico.

## 5 – Bibliografia

- Date, C. J. Banco de Dados - Fundamentos, Editora Campus, 1991
- Elmira, R. e Nava the, S.; Fundamentals of Database Systems; 2nd Edition; Benjamin/Cummings; 1994.
- Korth, H. e Silberschatz, A.; Sistema de Bancos de Dados; 2nd Edition; Makron Books; 1993.
- Notas de aula da disciplina Banco de Dados na pagina da Prof.<sup>a</sup> Fátima Vieira.
- [www.mysql.com.br](http://www.mysql.com.br)

# Anexo: Guia para manutenção no banco de dados

## Inserir um Campo

Para inserir um campo no banco de dados MySQL é necessário apenas seguir os passos a seguir.

1. Após entrar no banco seja local ou via web como foi mostrado no item 3.5 vá no banco de dados Chesf como apresentado na figura 8.

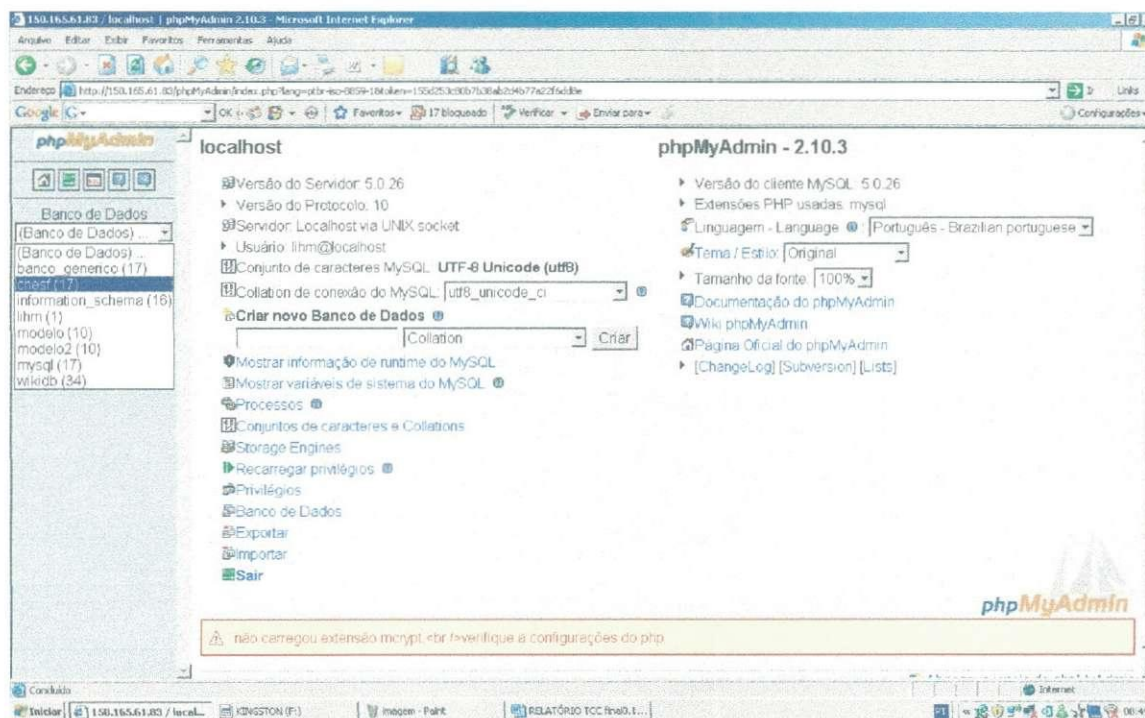


Figura 8: Banco de dados Chesf

2. Em seguida será apresentada a tela da figura 9. Onde temos todas as tabelas do banco de dados. Nesta tela que poderemos acrescentar mais tabelas como será apresentado no item 3.7.2.2.

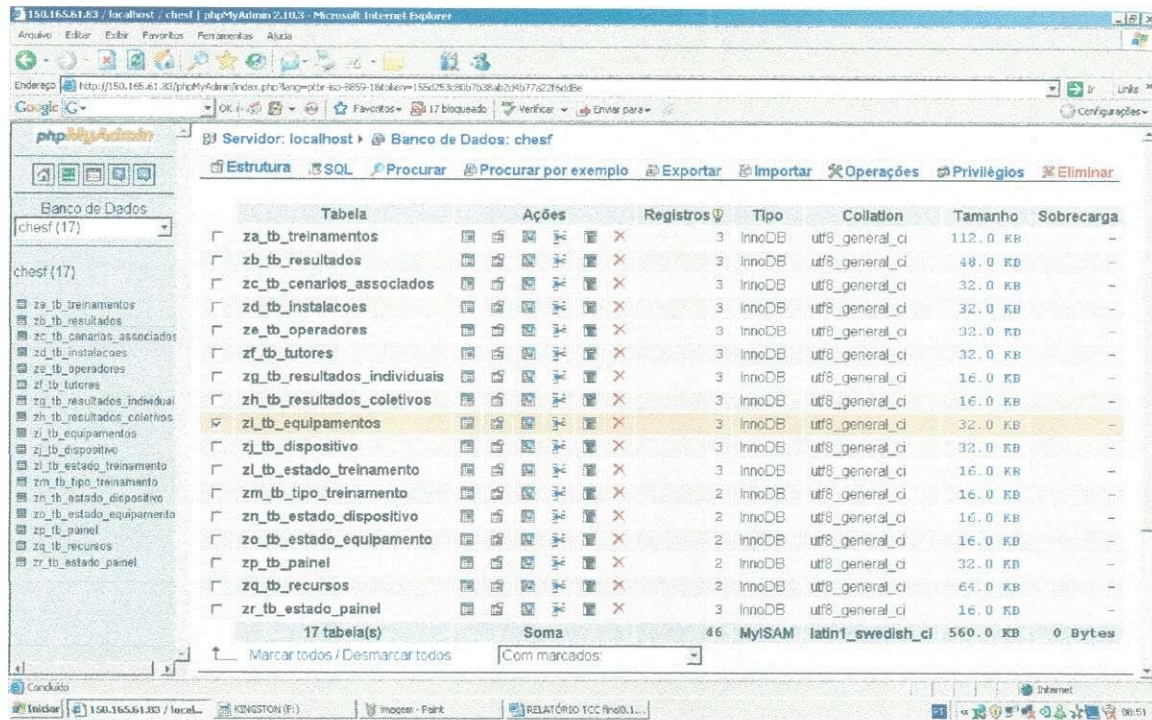


Figura 9: Tabelas do BD

3. Escolha a tabela na qual você deseja inserir um campo, no nosso caso escolheremos a tabela ze\_tb\_operadores e clique com o botão esquerdo do mouse, em seguida ira aparecer a tabela conforme a figura 10. Clique novamente com o botão esquerdo do mouse no botão executar que está marcado com uma seta vermelha na figura 10.

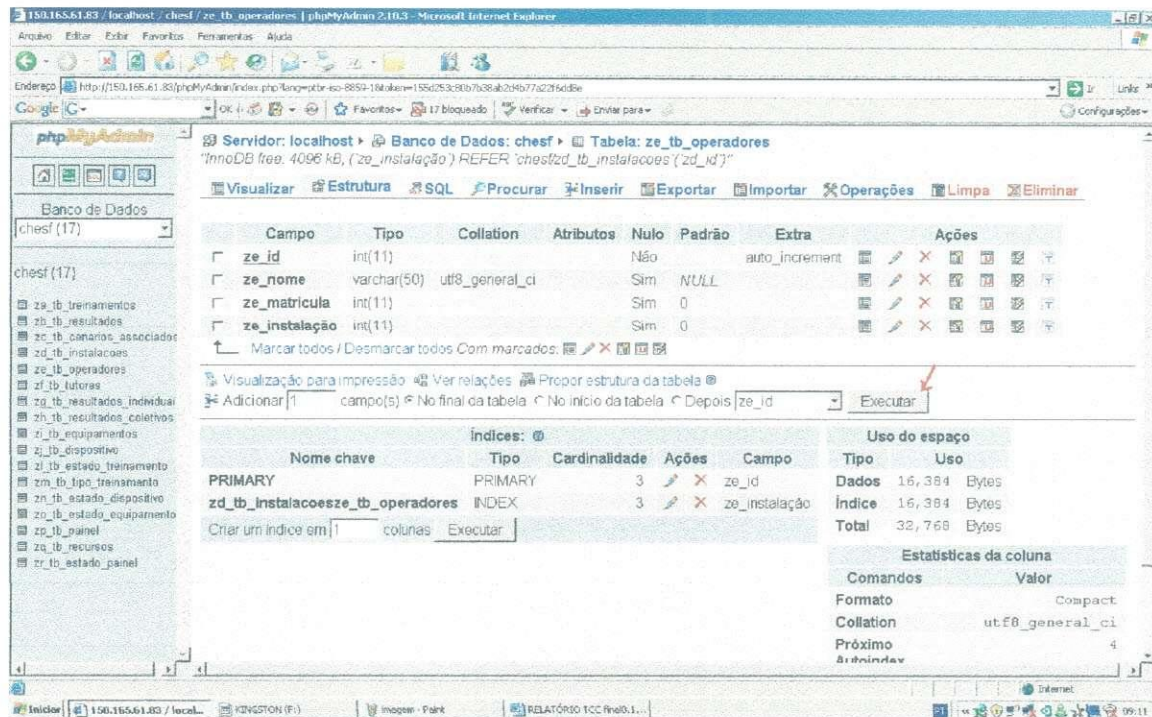


Figura 10: tabela ze\_tb\_operadores

4. Na tela apresentada na figura 11 criaremos o novo campo desejado, no item campo colocamos o nome do campo por exemplo ze\_nome, o tipo é onde definimos se aquele campo ira aceitar caracteres, números inteiros, datas, etc. Neste caso definimos texto VARCHAR, definir o tamanho que poderá armazenar aquele campo no nosso caso 100, collation tipo de caractere que armazenará podemos vamos deixar este campo em branco, pois, por definição ele usará Arial. Nulo, se deixarmos o item not null ele não aceitará valores nulos no banco de dados. Extra, nesta opção escolhemos se o número será auto incrementado automaticamente pelo banco de dados. Os ícones abaixo na figura são para identificar se o campo é chave primaria, estrangeira (no caso do MySql índice), e se o campo é único, neste caso ele não aceitará dois campo iguais nesta mesma coluna. Os comentários são para que o usuário do banco indique quais os objetivos de criar aquela coluna específica.

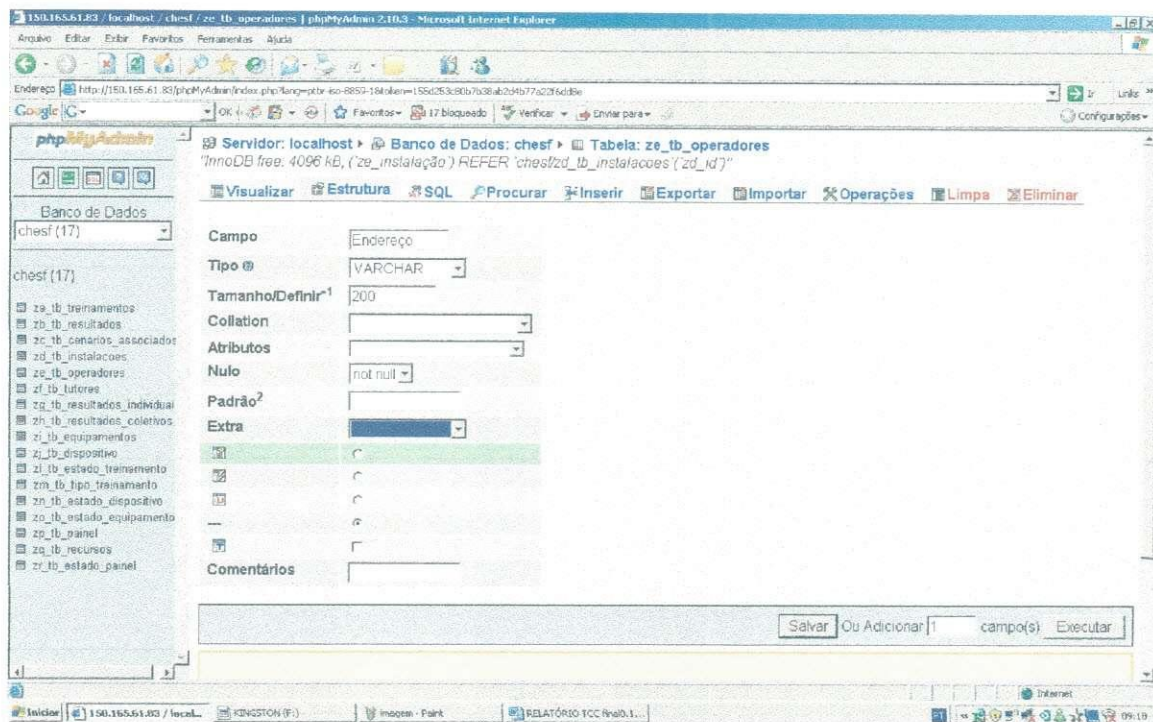


Figura 11: Tela de inserção de colunas da tabela ze\_tb\_operadores.

## Inserir uma tabela

Para inserir uma tabela no BD siga até o segundo passo do item 3.7.1.1, em seguida e na tela da figura 9, digite o nome da nova tabela e clique com o botão esquerdo do mouse no botão executar.

## Inserir um relacionamento

Para inserir um relacionamento primeiro criamos o campo como mostrado no item 3.7.1.1. Após criar o item, voltamos para a tela da figura 10 e clicamos no item ver relações, em seguida aparecerá a tela da figura 12. Nesta tela podemos verificar que o campo ze\_instalação se relaciona com o campo zd\_id, portanto as tabelas ze\_tb\_operadores esta relacionada como zd\_td\_instalações na forma de cascata.



Portanto para fazermos um relacionamento basta apenas ao chegar nesta tela escolher o campo no qual será feito o relacionamento. Conforme figura 13.

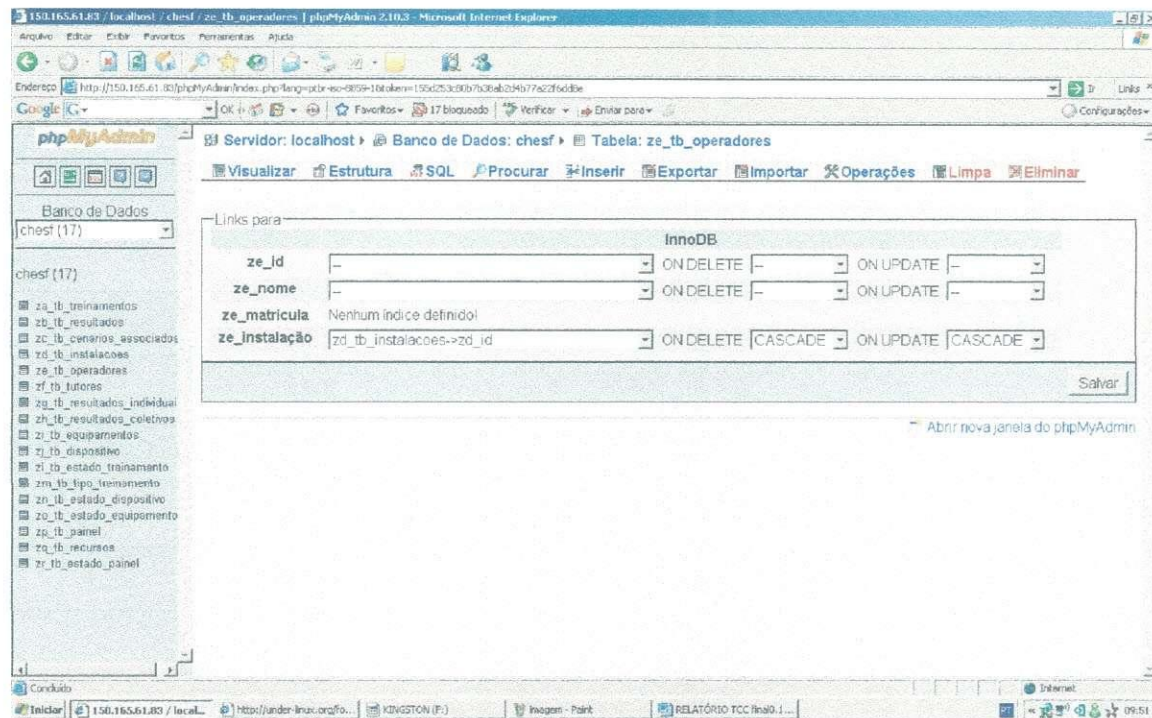


Figura 12: Tela inicial de relacionamento no BD

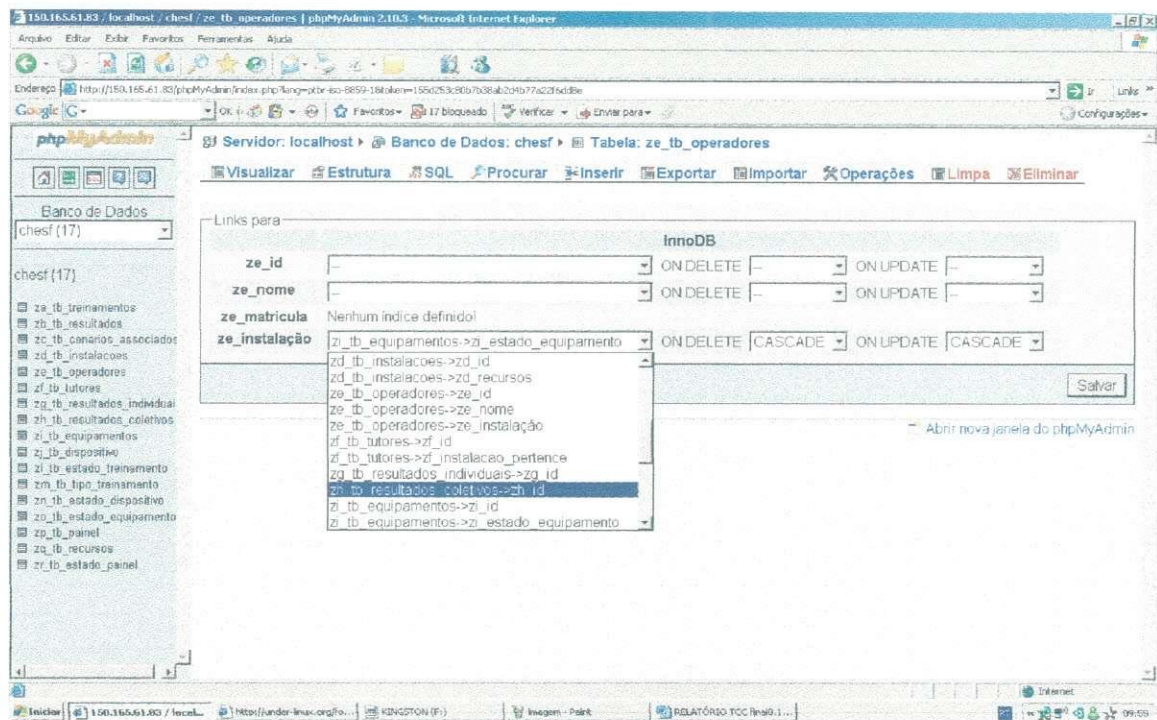


Figura 13: Como Fazer Relacionamento do BD

# Anexo: Dicionário de Dados

## **TREINAMENTOS**

### Descrição da tabela:

Informa qual é o treinamento, quem são os participantes, o tutor, data de início e término, o resultado do treinamento e o objetivo do treinamento é a tabela principal do banco, pois a maioria das consultas será feita a partir desta tabela.

### Relação-base:

za\_tb\_treinamentos (cadastro o treinamento id, cenários associados, tutor responsável, instalação, data inicial do treinamento, data final do treinamento, turma de operadores, estado do treinamento, resultados, objetivos, tempo necessário em minutos).

<b>Entidade:</b> za_tb_treinamentos
<b>Atributo:</b> za_id
<b>Descrição:</b> Id do item na tabela. <b>Domínio:</b> Números inteiros auto incrementados.
<b>Atributo:</b> za_cenarios_associados
<b>Descrição:</b> É o cenário no qual esta ou foi realizado o treinamento. <b>Domínio:</b> Aceita números inteiros.
<b>Atributo:</b> za_tutor_responsavel
<b>Descrição:</b> É o tutor responsável por ministrar o curso. <b>Domínio:</b> Números inteiros.
<b>Atributo:</b> za_instalacao_treinamento

Descrição: É a instalação sobre a qual esta sendo ministrado o curso, por exemplo, podemos simular a instalação Campina Grande II ou simular a instalação Sobradinho isto fica a cargo do curso e do tutor.

Domínio: Números inteiros.

Atributo: za\_data\_inicial\_treinamento

Descrição: É a data onde ocorreu o início do treinamento.

Domínio: Data/Hora.

Atributo: za\_data\_final\_treinamento

Descrição: É a data onde ocorreu o fim do treinamento.

Domínio: Data/Hora.

Atributo: za\_turma\_operador

Descrição: É a turma de operadores que participa do curso.

Domínio: Números inteiros.

Atributo: za\_estado\_treinamento

Descrição: É a situação na qual se encontra o treinamento. EX. Finalizado

Domínio: Números inteiros.

Atributo: za\_resultados

Descrição: É o resultado do treinamento que pode ser individual ou coletivo.

Domínio: Números inteiros.

Atributo: za\_objetivos

Descrição: É o objetivo final do curso.

Domínio: Todos os caracteres.

Atributo: za\_tempo\_necessario\_minuto

Descrição: É o máximo tempo que o operador tem para realizar o curso em minutos.

Domínio: Números inteiros.

EX:

<b>za_id</b>	<b>Za_cenarios - associados</b>	<b>za_tutor_ responsável</b>	<b>zc_instalacao_ treinamento</b>	<b>za_data_inicial_ treinamento</b>
1	3	12	5	10/05/2008
2	2	1	1	12/11/2008
3	2	8	1	20/11/2008

<b>za_data_final - treinamento</b>	<b>za_turma_ operador</b>	<b>za_estado_ treinamento</b>	<b>za_ resultados</b>	<b>za_ objetivos</b>	<b>za_tem po_ necessá rio _minuto</b>
11/05/2008	3	2	3	Operar Disjuntor 205	30
13/11/2008	2	2	2	Transferê ncia de barrament o 303	20
20/11/2008	2	1	2	Transferê ncia de barrament o 301	20

## RESULTADOS

### Descrição da tabela:

Informa quais é o foram os resultados, diferenciando entre individuais e coletivos.

### Relação-base:

zb\_tb\_resultados (cadastro dos resultados dos treinamentos id, resultado individual, resultado coletivo).

<b><u>Entidade:</u> zb_tb_resultados</b>
<b><u>Atributo:</u> zb_id</b>
<b><u>Descrição:</u> ID do resultado. <u>Domínio:</u> Números inteiros auto incrementados.</b>
<b><u>Atributo:</u> zb_resultado_individual</b>
<b><u>Descrição:</u> Chave estrangeira da tabela resultado individual. <u>Domínio:</u> Números inteiros.</b>
<b><u>Atributo:</u> zb_resultado_coletivo</b>
<b><u>Descrição:</u> Chave estrangeira da tabela resultado Coletivo <u>Domínio:</u> Números inteiros.</b>

zb_id	zb_resultado_individual	zb_resultado_coletivo
1	2	3
2	3	5
3	5	1

## **CENÁRIOS ASSOCIADOS**

### Descrição da tabela:

Informa quais foram os cenários de cada instalação.

### Relação-base:

zc\_tb\_cenarios\_associados (cadastro dos cenários dos treinamentos id, instalação, descrição).

<b>Entidade:</b> zc_tb_cenarios_associados
<b>Atributo:</b> zc_id
<b>Descrição:</b> ID do cenário associado <b>Domínio:</b> Números inteiros auto incrementados.
<b>Atributo:</b> zc_instalacao
<b>Descrição:</b> Chave estrangeira da tabela instalações. <b>Domínio:</b> Números inteiros.
<b>Atributo:</b> zc_descricao
<b>Descrição:</b> Descreve o cenário de treinamento e atividade que deve ser realizada neste cenário <b>Domínio:</b> Todos os caracteres.

zc_id	zc_instalacao	zc_descricao
1	2	Desligamento do disjuntor Principal da Subestação Campina Grande II
2	3	Transferência para barramento auxiliar da SE Recife I
3	5	Transferência para barramento auxiliar SE Tacaimbo.

## ***INSTALAÇÕES***

### Descrição da tabela:

Informa qual é a instalação sobre a qual será feito o treinamento.

### Relação-base:

zd\_tb\_instalacoes (cadastro das instalações as quais os operadores serão treinados id, recurso, descrição).

<u>Entidade:</u> <b>zd_tb_instalações</b>
<u>Atributo:</u> zd_id
<u>Descrição:</u> ID da instalação. <u>Domínio:</u> Números inteiros auto incrementados.
<u>Atributo:</u> zd_recurso
<u>Descrição:</u> Chave estrangeira da tabela recursos <u>Domínio:</u> Números inteiros.
<u>Atributo:</u> zd_descricao
<u>Descrição:</u> Descreve a instalação de treinamento. <u>Domínio:</u> Todos os caracteres.

zd_id	zd_recurso	zd_descricao
1	2	Subestação Campina Grande II
2	3	Subestação Recife I
3	5	Subestação Tacaimbo.



## **OPERADORES**

### Descrição da tabela:

Informa a matrícula, nome do operador e em qual instalação ele trabalha.

### Relação-base:

ze\_tb\_operadores (cadastro dos operadores id, nome, matrícula, instalação).

<u>Entidade:</u> <b>ze_tb_operadores</b>			
<u>Atributo:</u> ze_id			
<u>Descrição:</u> ID do operador.		<u>Domínio:</u> Números inteiros auto incrementados.	
<u>Atributo:</u> ze_nome			
<u>Descrição:</u> Nome do operador		<u>Domínio:</u> Texto	
<u>Atributo:</u> zd_matricula			
<u>Descrição:</u> Armazena a matricula do operador		<u>Domínio:</u> Números inteiros..	
<u>Atributo:</u> zd_instalacao			
<u>Descrição:</u> Chave estrangeira da tabela instalacao		<u>Domínio:</u> Números inteiros.	
ze_id	ze_nome	Ze_matricula	ze_instalacao
1	Bruno Almeida	234939309	1
2	Cristiano Ronaldo	384848480	2
3	Edson Arantes	509345990	6

## TUTORES

### Descrição da tabela:

Informa a matrícula, nome do tutor e em qual instalação ele trabalha.

### Relação-base:

zf\_tb\_tutores (cadastro dos tutores, matrícula, instalação).

<b><u>Entidade:</u> zf_tb_tutores</b>
<b><u>Atributo:</u> zf_id</b>
<b><u>Descrição:</u> ID do tutor</b> <b><u>Domínio:</u> Números inteiros auto incrementados.</b>
<b><u>Atributo:</u> zf_nome</b>
<b><u>Descrição:</u> Nome do operador</b> <b><u>Domínio:</u> Texto</b>
<b><u>Atributo:</u> zf_matricula</b>
<b><u>Descrição:</u> Armazena a matricula do tutor</b> <b><u>Domínio:</u> Números inteiros..</b>
<b><u>Atributo:</u> zf_instalacao</b>
<b><u>Descrição:</u> Chave estrangeira da tabela instalacao</b> <b><u>Domínio:</u> Números inteiros.</b>

zf_id	zf_nome	Zf_matricula	Zf_instalacao
1	Bruno Almeida	234939309	1
2	Cristiano Ronaldo	384848480	2
3	Edson Arantes	509345990	6

## **RESULTADOS INDIVIDUAIS**

### Descrição da tabela:

Informa o operador, o treinamento, tipo de treinamento e o resultado que o operador alcançou no treinamento.

### Relação-base:

zg\_tb\_resultados\_individuais (cadastro dos resultados individuais de cada operador no treinamento determinando o tipo de treinamento).

<b><u>Entidade:</u> zg_tb_resultados_individuais</b>
<b><u>Atributo:</u> zg_id</b>
<b><u>Descrição:</u> ID do resultado individual    <u>Domínio:</u> Números inteiros auto incrementados.</b>
<b><u>Atributo:</u> zg_operador</b>
<b><u>Descrição:</u> Chave estrangeira da tabela operador    <u>Domínio:</u> Números inteiros</b>
<b><u>Atributo:</u> zg_treinamento</b>
<b><u>Descrição:</u> Chave estrangeira da tabela treinamento    <u>Domínio:</u> Números inteiros.</b>
<b><u>Atributo:</u> zg_tipo_treinamento</b>
<b><u>Descrição:</u> Chave estrangeira da tabela tipo treinamento    <u>Domínio:</u> Números inteiros.</b>
<b><u>Atributo:</u> zg_resultado</b>
<b><u>Descrição:</u> Nota do operador no treinamento    <u>Domínio:</u> Números inteiros.</b>
<b><u>Atributo:</u> zg_tempo_utilizado_minuto</b>
<b><u>Descrição:</u> É o tempo que o operador realizou o curso em minutos. <u>Domínio:</u> Números inteiros.</b>

Zg_id	zg_operador	zg_treinamento	zg_tipo_ treinamento	zg_resultado	zg_tempo_ utilizado _minuto
1	2	5	1	8,00	17
2	5	5	2	7,50	10
3	1	5	2	9,50	19

## **RESULTADOS COLETIVOS**

### Descrição da tabela:

Informa o resultado do treinamento de um grupo de operados usando o critério de soma todas as notas e tirar à média.

### Relação-base:

zg\_tb\_resultados\_individuais (cadastro dos resultados coletivos dos operadores em um treinamento determinando o tipo de treinamento).

<u>Entidade:</u> zh_tb_resultados_coletivos
<u>Atributo:</u> zh_id
<u>Descrição:</u> ID do resultado coletivo <u>Domínio:</u> Números inteiros auto incrementados.
<u>Atributo:</u> zh_treinamento
<u>Descrição:</u> Chave estrangeira da tabela treinamento <u>Domínio:</u> Números inteiros
<u>Atributo:</u> zh_operador
<u>Descrição:</u> Chave estrangeira da tabela turma <u>Domínio:</u> Números inteiros.
<u>Atributo:</u> zh_tipo_treinamento
<u>Descrição:</u> Chave estrangeira da tabela tipo treinamento <u>Domínio:</u> Números inteiros.
<u>Atributo:</u> zh_resultado
<u>Descrição:</u> Nota da média dos operadores no treinamento <u>Domínio:</u> Números inteiros.
<u>Atributo:</u> zh_tempo_utilizado_minuto

Descrição: É o tempo que um conjunto de operadores realizou o curso em minutos.

Domínio: Números inteiros.

zg_id	zg_turma_ operador	Zg_treinamento	zg_tipo_ treinamento	zg_resultado	zh_tempo_ utilizado_ minuto
1	2	5	1	8,00	19
2	5	5	2	7,50	18
3	1	5	2	9,50	20

## ***EQUIPAMENTOS***

### Descrição da tabela:

Armazena qual é o equipamento pertencente a um determinado cenário.

### Relação-base:

zi\_tb\_equipamentos (cadastro dos equipamentos, descreve o equipamento e o estado do equipamento).

<u>Entidade:</u> <b>zi_tb_equipamentos</b>
<u>Atributo:</u> zi_id
<u>Descrição:</u> ID do equipamento <u>Domínio:</u> Números inteiros auto incrementados.
<u>Atributo:</u> zi_descricao
<u>Descrição:</u> Descreve o equipamento <u>Domínio:</u> Texto
<u>Atributo:</u> zi_estado_equipamento
<u>Descrição:</u> Armazena o estado do equipamento <u>Domínio:</u> Números inteiros.

zi_id	Zi_equipamento	zi_estado_equipamento
1	Transformador de força 5424	2
2	Transformador de potencial 4323	1
3	Transformador de corrente 3443	2

## ***DISPOSITIVOS***

### Descrição da tabela:

Armazena qual é o equipamento pertencente a um determinado cenário.

### Relação-base:

zj\_tb\_dispositivo (cadastro dos dispositivos, descreve o dispositivo e o estado do dispositivo).

<u>Entidade:</u> <b>zj_tb_dispositivo</b>
<u>Atributo:</u> zj_id
<u>Descrição:</u> ID do dispositivo _____ <u>Domínio:</u> Números inteiros auto incrementados.
<u>Atributo:</u> zj_descricao
<u>Descrição:</u> Descreve o dispositivo _____ <u>Domínio:</u> Texto
<u>Atributo:</u> zj_estado_dispositivo
<u>Descrição:</u> Armazena o estado do dispositivo _____ <u>Domínio:</u> Números inteiros.

zj_id	zj_dispositivo	zj_estado_equipamento
1	Disjuntor a SF6 3434	2
2	Disjuntor a óleo 3453	1
3	Disjuntor a SF6 5488	2



## RECURSOS

### Descrição da tabela:

Armazena uma lista com os recursos das instalações, estes recursos podem ser equipamentos, dispositivos e painéis.

### Relação-base:

zp\_tb\_recursos (id, dispositivos, equipamentos).

<u>Entidade:</u> <b>zq_recursos</b>
<u>Atributo:</u> zq_id
<u>Descrição:</u> ID do recurso. <u>Domínio:</u> Números inteiros auto incrementados.
<u>Atributo:</u> zq_dispositivo
<u>Descrição:</u> Chave estrangeira da tabela dispositivo.. <u>Domínio:</u> Números inteiros.
<u>Atributo:</u> zq_equipamentos
<u>Descrição:</u> Chave estrangeira da tabela equipamentos. <u>Domínio:</u> Números inteiros.
<u>Atributo:</u> zq_painel
<u>Descrição:</u> Chave estrangeira da tabela painel. <u>Domínio:</u> Números inteiros.

zq_id	Zq_dispositivos	Zq_equipamento s	Zq_painel
1	3	5	2
2	2	9	5
3	1	2	1

## ***ESTADO DO PAINEL***

### Descrição da tabela:

Armazena o estado do painel se esta em operação, manutenção ou estoque por exemplo.

### Relação-base:

zr\_tb\_estado\_painel (cadastro dos estados dos painel).

<u>Entidade:</u> <b>zr_tb_estado_painel</b>
<u>Atributo:</u> zr_id
<u>Descrição:</u> ID do estado do painel. <u>Domínio:</u> Números inteiros auto incrementados.
<u>Atributo:</u> zo_descricao
<u>Descrição:</u> Descreve o estado do painel. <u>Domínio:</u> Todos os caracteres.

zo_id	zo_descricao
1	Operação
2	Manutenção
3	Em estoque

## Glossário

No Modelo Relacional, os dados são armazenados em tabelas, entidades ou relações. Uma tabela, entidade ou relação é formada por atributos. Os atributos de uma entidade também são normalmente chamados de campo ou coluna. O conjunto de valores atribuídos aos campos, colunas ou atributos de uma tabela são chamados de linha ou registro.

Em banco de dados existe o conceito de chave primária. Chaves primárias (em inglês Primary Keys ou PK) referem-se a um dado campo ou atributo, cujos valores nunca se repetem e que podem ser usados como um índice para os demais campos da tabela do banco de dados. Nesta chave não pode haver valores nulos ou repetidos e elas garantem a unicidade do registro.

zj_id	zj_dispositivo	zj_estado Equipamento
1	Disjuntor a SF6 3434	2
2	Disjuntor a óleo 3453	1
3	Disjuntor a SF6 5488	2

Por exemplo, no banco de dados a tabela zj\_tb\_dispositivo, o campo chave primária da tabela produto é zj\_id, dessa forma, o produto Disjuntor a SF6 3434 será identificado pelo zj\_id=1, e nenhum outro dispositivo terá este mesmo código. As tabelas de um banco de dados relacional podem ser interligadas ou relacionadas através de relacionamentos definidos no momento da criação do banco de dados. É neste ponto que entra o conceito de chave estrangeira. Uma chave estrangeira é um atributo ou campo de uma tabela utilizado para criar um relacionamento entre tabelas. Uma chave estrangeira é a chave primária de uma tabela dentro de outra tabela. Por exemplo,

no banco de dados a tabela `zj_tb_dispositivo`, o campo `zj_estado_treinamento` é uma chave estrangeira que faz o relacionamento com a tabela `zo_tb_estado Equipamento`.

Ao definirmos relacionamentos, um outro conceito importante que surge é o de integridade referencial. A integridade referencial é utilizada para garantir a integridade dos dados entre as tabelas relacionadas. Por exemplo, considere um relacionamento entre as duas tabelas. Com a integridade referencial, o banco de dados não permite que seja cadastrado um produto de um estado do equipamento que ainda não foi cadastrado. Em outras palavras, ao cadastrar um dispositivo, o SGBD verifica se o código do estado do equipamento que foi digitado já existe na tabela `zo_tb_estado Equipamento`. Se não existir, o cadastro do produto não será aceito.

No modelo relacional os tipos de relacionamentos existentes são:

- Um-para-Um (1..1): Neste tipo de relacionamento, um registro na Tabela A possui um único correspondente na Tabela B, e vice-versa;
- Um-para-Muitos (1..n): neste tipo de relacionamento, um registro na Tabela A pode ter vários correspondentes na Tabela B. É o que acontece no relacionamento entre as tabelas `zj_tb_dispositivo` e `zo_tb_estado Equipamento` do banco de dados.
- Muitos-para-Muitos (n..m): Quando tabelas têm entre si relação n..m, é necessário criar uma nova tabela com as chaves primárias das tabelas envolvidas, ficando assim uma chave composta, ou seja, formada por diversos campos-chave de outras tabelas. A relação então se reduz para uma relação 1..n, sendo que o lado n ficará com a nova tabela criada.