

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

FERRAMENTA PARA CONFIGURAÇÃO DE SISTEMA
GERENCIADOR DE BANCO DE DADOS RELACIONAL
POSTGRESQL 7.3

REULY BÚSSOLO MENDES

BLUMENAU
2005

2005/1-12

REULY BÚSSOLO MENDES

**FERRAMENTA PARA CONFIGURAÇÃO DE SISTEMA
GERENCIADOR DE BANCO DE DADOS RELACIONAL
POSTGRESQL 7.3**

Trabalho de Conclusão de Curso submetido à
Universidade Regional de Blumenau para a
obtenção dos créditos na disciplina Trabalho
de Conclusão de Curso II do curso de Sistemas
de Informação — Bacharelado.

Prof. Wilson Pedro Carli - Orientador

**BLUMENAU
2005**

2005/1-12

**FERRAMENTA PARA CONFIGURAÇÃO DE SISTEMA
GERENCIADOR DE BANCO DE DADOS RELACIONAL
POSTGRESQL 7.3**

Por

REULY BÚSSOLO MENDES

Trabalho aprovado para obtenção dos créditos
na disciplina de Trabalho de Conclusão de
Curso II, pela banca examinadora formada
por:

Presidente:

Prof. Wilson Pedro Carli – Orientador, FURB

Membro:

Prof. Luiz Bianchi, FURB

Membro:

Prof. Paulo Fernando da Silva, FURB

Blumenau, 25 de Junho de 2005.

Nunca ande pelo caminho traçado, pois ele
conduz somente até onde os outros foram.

Grahan Bell

AGRADECIMENTOS

Aos meus pais e irmãs por sempre me ajudarem quando precisei de minha família.

À Roseléia Martins, por percorrer comigo quase todo o trajeto da graduação, sempre me apoiando.

Ao meu orientador, professor Wilson Pedro Carli, por acreditar na conclusão e ideal deste projeto.

Às empresas Unimed de Brusque, Teiko Soluções em TI e em especial à Tecmicro Sistemas, pelo apoio ao desenvolvimento deste projeto.

Aos meus amigos Joseane Dell Agnelo, Fernando Duarte, Patrícia Benvenuti, Juliana K. Topolski, Rafaela F. Kohler, Ana Paula Berlanda, Juciane Melssin e além da amizade e ajuda técnica de Guilherme Zirke Júnior, Márcia Righetto e Juliana da Silva.

RESUMO

Este trabalho descreve a criação de uma ferramenta *Web* para realizar configurações de pós-instalação do Sistema Gerenciador de Banco de Dados Relacional *PostgreSQL*. Esta ferramenta utiliza-se tanto de programação *Web* quanto de *scripts Linux* e está destinada para consultores e técnicos de informática que possuem pouco ou nenhum conhecimento de *Linux*, realizando configurações do *PostgreSQL* e *backup* em ambiente gráfico.

Palavras chaves: PostgreSQL; Banco de Dados; Linux; Backup

ABSTRACT

This work describes the creation of a tool web to carry through configurations of after-installation of the Relation Database Management System PostgreSQL. This tool in such a way uses of web programming how much of scripts Linux and it is destined for consultants and technician of computer science that they possess little or no knowledge of Linux, carrying through configurations of the PostgreSQL and backup in graphical environment.

Key-Words: PostgreSQL; Database; Linux; Backup.

LISTA DE ILUSTRAÇÕES

Figura 1: Caso de uso	24
Figura 2: Modelo conceitual.....	25
Figura 3: Modelo de entidade relacionamento	26
Figura 4: Navegador dos objetos no <i>Dreamweaver MX 2004</i>	31
Figura 5: Editor de comando do <i>Dreamweaver</i>	32
Figura 6: Editor de arquivo texto no <i>Linux</i>	32
Figura 7: Duplicação de base de dados	33
Figura 8: Conversão de arquivo e restauração de base de dados.....	33
Figura 9: Recriação do arquivo de configuração do SGBDR PostgreSQL.....	34
Figura 10: Menu de navegação.....	35
Figura 11: Restauração de base de dados	36
Figura 12: Duplicação de base de dados	36
Figura 13: Envio de base de dados via <i>FTP</i>	37
Figura 14: Informação para <i>backup</i>	37
Figura 15: Comandos a serem usados no agendamento.....	38
Figura 16: Agendamento de tarefas no servidor.....	39
Figura 17: Visualização de estatísticas por tamanho, retornando o arquivo para realização de ajuste.....	39
Figura 18: Estatísticas por acesso.....	40
Figura 19: Liberação de acesso.	40
Figura 20: Configuração de parâmetros do <i>PostgreSQL</i>	41
Figura 21: Serviços do servidor.....	41

LISTA DE QUADROS

Quadro 1: Dados das empresas.....	27
Quadro 2: Grupos da configuração do SGBDR PostgreSQL.....	27
Quadro 3: Configurações do SGBDR PostgreSQL.....	28
Quadro 4: Configuração de acesso remoto do SGBDR PostgreSQL.....	28
Quadro 5: Comandos para o agendamento.....	29
Quadro 6: Usuários para o agendamento.....	29
Quadro 7: Agendamento no sistema operacional.....	29

LISTA DE SIGLAS

SGBDR – Sistema Gerenciador de Banco de Dados
DML – Data Manipulation Language
DDL – Data Definition Language
PHP – Pré Hypertext Processor
TCP/IP – Transmission Control Protocol/Internet Protocol
VPN – Virtual Private Network
SQL – Structured Query Language
HTML – Hyper Text Markup Language
IBM – Internation Business Machines Corporation
ANSI – American National Standarts Institute
ISO – Internation Organization for Standardization
DBA – Database Administrator
RDBMS – Relation Database Management System
JDBC – Java Database Connectivity
TCL – Tool Command Language
PL – Procedural Language
MVCC – Multiversion Concurrency Control
WAL – Write Ahead Log
CGI – Commom Gateway Interface
NCSA – National Center for Super Computing Applications
HTTP – Hypertext Transport Protocol
SSI – Server-Side Includes
SSL – Sercured Socket Layer
FTP – File Tranfer Protocol
KB – Kilobytes
MS-DOS – Microsoft Disk Operation System
API – Application Program Interface

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 OBJETIVOS.....	14
1.2 ESTRUTURA DO TRABALHO.....	14
2 REVISÃO BIBLIOGRÁFICA	15
2.1 BANCO DE DADOS RELACIONAL	15
2.2 BANCO <i>POSTGRESQL</i>	16
2.2.1 Características do <i>PostgreSQL</i>	16
2.2.2 Dicionário de dados.....	17
2.2.3 Administração de banco de dados.....	18
2.3 TECNOLOGIAS	18
2.3.1 <i>JavaScript</i>	18
2.3.2 <i>PHP</i>	19
2.3.3 <i>HTML</i>	19
2.3.4 <i>Linux Scripts</i>	20
2.3.5 <i>Dreamweaver</i>	20
2.3.6 Servidor <i>Apache</i>	20
3 DESENVOLVIMENTO DA FERRAMENTA	22
3.1 REQUISITOS FUNCIONAIS.....	22
3.2 REQUISITOS NÃO FUNCIONAIS.....	23
3.3 ESPECIFICAÇÃO	23
3.3.1 Caso de uso	24
3.3.2 Modelo Conceitual.....	25
3.3.3 MER.....	25
3.4 DICIONÁRIO DE DADOS	26
3.5 IMPLEMENTAÇÃO	30
3.6 PRINCIPAIS FUNCIONALIDADES.....	34
3.6.1 Navegação.....	34
3.6.2 Manipulação de Base de Dados	35
3.6.3 Configuração de <i>backup</i>	37
3.6.4 Estatísticas.....	39

3.6.5 Liberação de acesso.....	40
3.6.6 Configuração de parâmetros do SGBDR	40
3.6.7 Ações no servidor.....	41
4 CONSIDERAÇÕES FINAIS	42
4.1 <i>CONCLUSÕES</i>	42
4.2 <i>EXTENSÕES</i>	42
ANEXO A – DECLARAÇÃO DA TECMICRO SISTEMAS.....	44

1 INTRODUÇÃO

Nos últimos anos percebeu-se por parte do setor empresarial, a necessidade de desenvolvimento na forma de armazenar e restaurar informações; empresas de todos os portes buscam soluções para tornar esta tarefa menos trabalhosa possível. Além de implantarem um sistema de informação que atenda suas necessidades, um fator de suma importância está na maneira que armazenam seus dados. Com o surgimento dos sistemas gerenciadores de banco de dados esta tarefa tornou-se mais ágil e confiável.

Os estudos na área de banco de dados se intensificaram nos últimos anos. Diversos trabalhos foram realizados nesta área visando conceituar juntamente com o crescimento dos sistemas de informações empresariais, estabelecendo requisitos, propondo uma arquitetura adequada e definindo *interfaces* necessárias para cada tipo de usuário.

Segundo Leite (1980), a evolução das técnicas de projeto e implementação de sistemas de informação apoiados em computadores tem possibilitado a construção de sistemas cada vez mais abrangentes e, por conseguinte, o volume de dados a ser manipulado, cada vez maior, assim como as estruturas de armazenamento mais complexas. Por outro lado, o custo de manutenção de tais sistemas tem-se elevado, principalmente devido à dependência existente entre programas e dados. Com a finalidade de armazenamento de dados apresentando-se como uma ótima alternativa, encontra-se no mercado o *PostgreSQL*, um Sistema Gerenciador de Banco de Dados Relacional (*SGBDR*). O *PostgreSQL* foi projetado para ser um *SGBDR* simples de administrar, robusto e capaz de atender a uma gama de transações simultâneas. Não requerer custos de licenciamento e possuir seu código aberto – chamado *Open Source* – denotam dois amplos fatores de distinção deste *SGBDR* em relação aos demais.

Mesmo o *SGBDR PostgreSQL* possuindo uma instalação e administração simplificada, para o uso em servidores *Linux* é necessário ter um nível de conhecimento plausível do sistema operacional para um melhor aproveitamento das funcionalidades do gerenciador de banco de dados.

O termo *SGBDR* não possui uma definição estabelecida, todavia diversos autores entram em consenso quando se tratam das seguintes funcionalidades:

- a) permitir a independência de dados;

- b) poder oferecer estruturas de dados compatíveis com as características dos diversos tipos de usuários;
- c) permitir e manter o relacionamento entre os dados;
- d) controlar a redundância entre os dados;
- e) permitir o acesso concorrente;
- f) oferecer facilidades que permitam estabelecer o controle de acesso;
- g) garantir a integridade das informações armazenadas;
- h) garantir a segurança dos dados contra ações externas ao ambiente do sistema;
- i) ter bom desempenho.

A Tecmicro Sistemas, empresa desenvolvedora de softwares para pequenas e médias empresas, com clientes em mais de 160 cidades brasileiras, visando atender o crescimento das necessidades de transição de dados de seus clientes, está realizando migrações de seus sistemas de banco de dados *Microsoft Access* para *SGBDR PostgreSQL*. Para que não seja necessário um intenso treinamento dos colaboradores da empresa, a ferramenta desenvolvida neste trabalho incorpora em um ambiente *Web*, as principais funcionalidades das quais não são encontradas em outras ferramentas disponíveis no mercado.

Este trabalho tem como objetivo, a criação de uma ferramenta gráfica para promover a manipulação de base de dados, configurações do *PostgreSQL* e agendamentos de *backup*. Diferindo das ferramentas presentes no mercado atual, que permitem ao seu usuário executar comandos do tipo *Data Manipulation Language (DML)* e *Data Definition Language (DDL)* no *SGBDR*, esta ferramenta está com seu foco voltado para execução de comandos ao nível de sistema operacional. A ferramenta batizada com o nome de *TECgres*, utiliza-se das linguagens de programação: *Hyper Text Markup Language (HTML)*, *Pré Hypertext Processor (PHP)*, *JavaScript* e *Linux Scripts* além de comandos de linha em *Linux*. A mistura dessas linguagens foi realizada com a intenção de proporcionar uma maior versatilidade à ferramenta.

Desenvolvida para o uso em navegadores de Internet, promove o acesso de qualquer computador que se localiza na mesma rede do servidor de banco de dados ou mesmo conectado a este, via protocolo de Internet (*TCP/IP*). A exemplo, redes virtuais privadas – *VPN* – sem a obrigação de qualquer tipo de instalação no computador que estiver acessando a ferramenta.

1.1 OBJETIVOS

O objetivo deste trabalho foi desenvolver um aplicativo gráfico que possa ser acessado via navegador, para realizar as configurações de funcionalidades do *SGBDR PostgreSQL 7.3*, intervindo em configurações e arquivos tanto do próprio *SGBDR* quanto do sistema operacional *Linux*.

Os objetivos específicos do trabalho são:

- a) criar um ambiente para configuração de liberação de acesso de estações de trabalho contidas nos arquivos de configuração do banco de dados;
- b) criar um ambiente para configuração dos parâmetros de conexão, uso de memória, acesso padrão de transações, parâmetros de otimização, otimização de *query's*, exposição de mensagens, registros do *SGBDR*, coleta de estatísticas, acesso às coletas de estatísticas, bloqueios de registro e regionalidade do banco de dados;
- c) desenvolver uma interface para manipulação de base de dados – cópia, criação, duplicação, envio via Internet e exclusão;
- d) implantação de estrutura física e interface para o agendamento de *backup's* físicos, lógicos e envio dos *backup's* para outra estação na rede;
- e) desenvolver um ambiente para atualização e visualização das estatísticas do *SGBDR*;
- f) criar um ambiente para ações de iniciar, reiniciar e parar os serviços que estão relacionados com a ferramenta – *PostgreSQL*, *Apache*, *Samba* e sistema operacional *Linux*.

1.2 ESTRUTURA DO TRABALHO

No primeiro capítulo encontra-se a parte introdutória.

O segundo capítulo contém a revisão bibliográfica com os conceitos necessários para o entendimento deste trabalho.

No terceiro capítulo tem-se a metodologia de desenvolvimento, ferramentas empregadas, a especificação e operacionalidade da ferramenta *TECgres*.

As conclusões e considerações finais estão contidas no quarto capítulo.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo faz-se uma revisão bibliográfica dos principais conceitos utilizados neste trabalho, tais como banco de dados relacional, banco *PostgreSQL*, linguagem *JavaScript*, ambiente *PHP*, linguagem *HTML*, *Linux Scripting*, *Dreamweaver*, Servidor *Apache*.

2.1 BANCO DE DADOS RELACIONAL

De acordo com Date (2000), *SGBDR* é um software que trata de todo o acesso ao banco de dados. Conceitualmente, o que ocorre é o seguinte:

- a) um usuário faz um pedido de acesso usando uma determinada sublinguagem de dados (em geral *SQL*);
- b) o *SGBDR* intercepta o pedido e analisa;
- c) o *SGBDR* inspeciona, por sua vez, o esquema externo (ou as versões objeto desse esquema) para esse usuário, o mapeamento externo/conceitual correspondente, o esquema conceitual, o mapeamento conceitual/interno e a definição da estrutura de armazenamento;

Criado no início dos anos 80 pela *IBM – Internation Business Machines Corporation*, o padrão *Structured Query Language (SQL)*, é a linguagem estruturada padrão para acessar dados em um *SGBDR*. Em meados de 80, o *ANSI – Instituto Americano de Padrões* – iniciou o desenvolvimento para a padronização de uma linguagem para os bancos de dados relacionais, publicando sua primeira padronização em 1986. Em 92, o *ANSI* e a *ISO – Organização Internacional de Padrões* – lançaram o *SQL ANSI-92*, que seria um conjunto maior de padronização.

A expansão desta linguagem deve-se ao fato de sair das estruturas de computadores de grande porte, chamados de *mainframe* e ter chegado nos computadores pessoais.

Os comandos *SQL* são divididos em duas principais categorias:

- a) *DDL* – Linguagem de definição de dados: utilizada para definição de dados e de objetos de um banco de dados, alterando as informações no dicionário de dados;
- b) *DML* – Linguagem de manipulação de dados: utilizada para manipulação de dados, recuperando, inserindo, excluindo e alterando dados de um banco de dados.

2.2 BANCO *POSTGRES*SQL

Segundo Pereira Neto (2003), o *PostgreSQL* é um *SGBDR* – Sistema Gerenciador de Banco de Dados Relacional (ou mais comumente indicado, do inglês *RDBMS* – *Relation Database Management System*) que está baseado nos padrões *SQL ANSI-92*, 96 e 99, de alta performance, de fácil administração e utilização em projetos (por especialistas *Database Administrator (DBA)* e Projetistas de Sistemas).

Com sua criação sendo iniciada em 1986 na Universidade Norte-Americana da Califórnia, a Universidade de *Berkeley*, o *SGBDR PostgreSQL* vem sendo aprimorado com muitos benefícios, além dos padrões *SQL ANSI*, possui uma gama de funções que estão implementadas em *SGBDRs* como *Oracle*, *SQL Server* e *DB/2*. Hoje em sua versão 8, o *PostgreSQL* conta com suporte nativo para plataforma *Windows* – que até a versão anterior necessitava de aplicativo emulando *Linux* –, pontos de salvamento, recuperação a partir de *logs* de transações, *tablespaces* e uma melhor otimização do uso de memória e disco.

2.2.1 Características do *PostgreSQL*

Algumas características do *PostgreSQL*:

- a) possibilita a manipulação de rotinas e regras complexas. Exemplos destas avançadas funcionalidades são o “*SQL query*” declarativo, controle de concorrência multi-versão, suporte a multi-usuário, otimização de *query*, herança e listas (*arrays*);
- b) altamente expansível, o *PostgreSQL* suporta operações, funções, métodos de acesso e tipos de dados;
- c) suporte *SQL* compreensível - *PostgreSQL* suporta a especificação *SQL99* e inclui características avançadas desde *SQL92*;
- d) integridade referencial - é usada para validar os dados de uma base de dados;
- e) *API* – *Application Program Interface* – flexível – a flexibilidade da *API* do *PostgreSQL* determina fácil suporte ao desenvolvimento do *PostgreSQL ORDBMS*. Tais interfaces incluem: *Object Pascal (Delphi)*, *Python*, *Pearl*, *PHP*, *ODBC*, *Java/JDBC*, *Ruby*, *TCL*, *C/C++* e *Pike*;
- f) linguagem de procedimentos (*Procedure*) - suporta linguagem procedural interna, incluindo uma linguagem nativa chamada *PL/pgSQL*. Outra vantagem do

PostgreSQL é a capacidade de usar *Pearl*, *Python* ou *TCL* como linguagem de procedimentos;

- g) *MVCC - Multi-Version Concurrency Control*, é a tecnologia que *PostgreSQL* usa para não precisar de trava (*locking*). Outros Sistemas Gerenciadores de Bancos de Dados (*SGBD*) como *MySQL* ou *Access*, podem bloqueiar os dados fazendo o usuário esperar uma resposta, isto é, o “Leitor” (*Reader*) do *PostgreSQL* fica bloqueado para atualizar os registros. *PostgreSQL* mantém-se a par de todas as transações executadas pelo usuário na base de dados, por isso está apto a manusear os registros sem causar o atraso;
- h) cliente/servidor - *PostgreSQL* utiliza um processo por usuário, a arquitetura cliente/servidor tem um processo principal que dispara outros processos adicionais para cada conexão;
- i) *Write Ahead Log (WAL)* - se houver um erro na base de dados, será criado um registro de uma transação para ser restaurada, ocasionando bons benefícios para o evento do erro. Se qualquer mudança for feita na base de dados, pode ser recuperado usando os arquivos de *log*. Uma vez restaurado, o usuário continua trabalhando do ponto antes do erro ter ocorrido.

2.2.2 Dicionário de dados

As informações internas de um banco de dados são armazenadas em seu dicionário de dados. Neste dicionário são armazenadas informações dos objetos, dispositivos de armazenamento, parâmetros de memória, entre outras informações.

Segundo Ramalho (1999), um dicionário de dados é criado junto com um banco de dados. Para refletir sempre com precisão o *status* desse banco de dados, o dicionário de dados é atualizado automaticamente pelo banco de dados quando ocorrem ações específicas (tais como uma alteração na estrutura do banco de dados). Ele é crítico para a operação do banco de dados, o qual depende desse dicionário para registrar, verificar e conduzir o trabalho constantemente.

2.2.3 Administração de banco de dados

Os *SGBDR's* por possuírem uma estrutura mais ampla que seus antecessores, necessitam de um técnico responsável com conhecimento para sua administração, a pessoa que realiza este trabalho é conhecida como administrador de banco de dados (*DBA*).

Segundo Date (1990), dentre as responsabilidades do *DBA* incluem-se:

- a) decidir o conteúdo de informações do banco de dados;
- b) decidir a estrutura de armazenamento e a estratégia de acesso;
- c) servir de elo de ligação com usuários;
- d) definir os controles de segurança e integridade;
- e) definir a estratégia de reserva e recuperação;
- f) monitorar o desempenho e atender as necessidades de modificações;
- g) rotinas de carga;
- h) rotinas de despejo na memória e recuperação;
- i) rotinas de reorganização;
- j) rotinas de estatísticas;
- k) rotinas analíticas.

É recomendado que toda empresa que se utiliza de um *SGBDR*, possua um *DBA* na empresa, habilitado para executar os serviços acima citados, porém na prática, empresas de médio e pequeno porte terceirizam estes serviços visando um menor custo e um melhor preparo desses especialistas.

2.3 TECNOLOGIAS

Para o desenvolvimento deste trabalho, além do conhecimento em *PostgreSQL*, foram utilizadas algumas tecnologias que a seguir são descritas.

2.3.1 *JavaScript*

Segundo Silva (2003), o *JavaScript* foi desenvolvido pela *Netscape* e, originalmente, se chamava *LiveScript*. Mais tarde resolveram trocar o nome para mostrar sua proximidade com o *Java*, ainda que sejam linguagens totalmente distintas. Por ser uma linguagem interpretada, códigos *JavaScript* são escritos em forma de texto e na linguagem (em inglês) compreensível por nós humanos, acrescenta Silva.

Da validação de dados à exibição de efeitos dinâmicos, o *JavaScript* permite enriquecer as tarefas de uma página de Internet.

2.3.2 *PHP*

Segundo Kabir (2002), o *PHP* é uma linguagem interpretada tradicionalmente utilizada para criação de *scripts* do lado servidor. Uma página *HTML* é incorporada a código de *PHP*, que é analisado por um módulo no servidor ou por um script *CGI* para produzir a saída desejada, a qual é então enviada ao cliente *Web*; o cliente *Web* nunca vê nenhum código *PHP*.

A cada dia que passa, o *PHP* se torna a solução de empresas, entre outras, pode-se citar algumas das razões segundo Kabir (2002):

- a) desenvolvimento *Web* rápido;
- b) portátil nas principais plataformas de sistemas operacionais;
- c) portátil nas principais plataformas de servidores *Web*;
- d) suporte interno para os banco de dados mais importantes;
- e) alto desempenho;
- f) fácil aprendizado para novos desenvolvedores;
- g) suporte para padrões da Internet;
- h) amigável para as empresas;
- i) comunidade de fonte aberta.

2.3.3 *HTML*

Linguagem padrão para criação de *Web*, a *HyperText Markup Language – HTML* – segundo Venetianer (1996), é uma linguagem de programação muito simples, utilizada para criar documentos *hipertexto*, que pode ser portada de uma plataforma computacional para outra. Isto significa que você pode escrever códigos-fonte *HTML* sem se preocupar em qual computador e por qual sistema operacional este documento será visualizado. Um documento texto é constituído de trechos de texto, muito parecidos com aqueles que você produz num editor de textos. Na verdade, se você quiser produzir páginas codificadas em *HTML*, basta utilizar um editor de textos.

A *HTML* (Linguagem de Anotação de *Hipertexto*) recebeu este nome por seu código possibilitar a ligação do trecho de um texto com outro texto. Nos tempos atuais é utilizada em

um contexto mais amplo, pois é a base para a criação de páginas *Web* que vão desde simples páginas pessoais até portais corporativos de redes externas.

2.3.4 *Linux Scripts*

De acordo com Ball (1999), os programas de *shell* estão relacionados de perto com os *batch files* do mundo *DOS*. Entretanto, ao contrário dos *batch files*, os programas de *shell* possuem funcionalidade muito mais avançada: eles são aparentados com as linguagens de programação convencionais. Usando pelo agendamento de comandos do sistema operacional *Linux*, os *Linux scripts* realizam uma seqüência de comandos na hora em que foram determinados, realizando assim operações em segundo plano sem a necessidade de intervenção de humanos.

2.3.5 *Dreamweaver*

De acordo com Lowery (2002), o *Dreamweaver* da *Macromedia* é um editor de aplicações *Web* que permite criar, de forma simples, vários estilos de *homepages*. Utilizado para o desenvolvimento do *TECgres*, a versão *MX 2004* oferece uma série de recursos visuais para a maximização dos resultados visuais e minimização do tempo de desenvolvimento.

2.3.6 *Servidor Apache*

Segundo Kabir (2002), o *Apache* trata-se de um servidor *Web*, que surgiu de um outro projeto criado pela *National Center for Super Computing Applications – NCSA* – que colocou seu servidor *Web* no mercado e tornou-o número um entre os servidores no início de 1995. Com a saída do principal desenvolvedor do projeto, o mesmo começou a perder fôlego. A primeira versão do *Apache* (0.6.2) foi lançada em abril de 1995 e surgiu da troca de correções efetuadas pelos próprios usuários do servidor da *NCSA*, nascendo assim o *Apache Group*.

De acordo com Kabir (2002), alguns dos motivos que torna o *Apache* o servidor *Web* mais utilizado são:

- a) o *Apache* é um servidor *Web* altamente confiável com um projeto modular;
- b) tecnologia gratuita e aberta;
- c) o *Apache* funciona muito bem com *Perl*, *PHP* e outras linguagens de *scripts*;
- d) funciona em *Linux* e outros sistemas operacionais, como *Unix* e *Windows*.

Além de ser multi-plataforma, o *Apache* possui outras características, tais como:

- a) suporte para o protocolo *http* 1.1, mais recente;
- b) configuração baseada em arquivos, simples, mas poderosa;
- c) suporte para *Common Gateway Interface (CGI)*;
- d) suporte para *FastCGI*;
- e) suporte para *hosts* virtuais;
- f) suporte para autenticação de *http*;
- g) *Perl* integrado;
- h) suporte para criação de scripts *PHP*;
- i) suporte para *servlets Java*;
- j) servidor de *proxy* integrado;
- k) *status* do servidor e *logs* personalizáveis;
- l) suporte para *Server-Side Includes (SSI)*;
- m) suporte para *Secured Socket Layer (SSL)*.

3 DESENVOLVIMENTO DA FERRAMENTA

Neste capítulo apresentam-se as etapas existentes para a execução deste trabalho, tais como os requisitos, a especificação, a implementação e a operacionalidade da ferramenta.

3.1 REQUISITOS FUNCIONAIS

Na definição dos requisitos se contemplou a necessidade de criar uma ferramenta gráfica para administração de base de dados, configuração de *backup's* e configuração de parâmetros do *SGBDR PostgreSQL*. No levantamento de informações efetuado na empresa Tecmicro, obteve-se os seguintes requisitos funcionais:

- a) verificar arquivos de *flag* de recriação de banco e estrutura de dados;
- b) recriar o banco de dados e objetos com parâmetros que atendam as necessidades dos sistemas da Tecmicro;
- c) criar base de dados no *SGBDR*;
- d) restaurar base de dados através de *backup's* anteriores ou arquivos externos;
- e) realizar *backup* de base de dados;
- f) duplicar base de dados, a partir de uma origem;
- g) enviar base de dados para servidor de *File Transfer Protocol (FTP)* da Tecmicro Sistemas;
- h) excluir base de dados do *SGBDR*;
- i) gravar informações para *backup* remoto;
- j) gravar comandos para serem utilizados no agendamento;
- k) ativar agendamentos para os usuários do sistema operacional do servidor;
- l) atualizar estatísticas do banco de dados;
- m) listar as estatísticas do banco de dados demonstrando os tamanhos dos objetos no banco de dados;
- n) listar as estatísticas do banco de dados demonstrando os acessos dos objetos no banco de dados;
- o) alterar configurações de rede do *SGBDR PostgreSQL*;
- p) alterar parâmetros de configuração do *SGBDR PostgreSQL*;
- q) realizar ações nos serviços relacionados com a ferramenta.

As características que agregam valor à ferramenta são:

- a) *interface* simples e clara para a realização das ações;

- b) reutilizar código sempre que possível, para uma manutenção simplificada.

3.2 REQUISITOS NÃO FUNCIONAIS

Os requisitos não funcionais são os requisitos necessários para a execução da ferramenta, mas que não estão contempladas na mesma. Estes requisitos estão nas configurações da estação de *backup* remoto, *Linux*, *Apache* e módulo *PHP* e funcionamento da rede, são eles:

- a) o sistema operacional *Linux Red Hat 9* deverá estar instalado e com os pacotes necessários para o uso do *PostgreSQL* e *Apache* com *PHP* utilizando o módulo para *PostgreSQL*;
- b) o usuário do sistema operacional do *Linux* que executará o *Apache* deve ser o mesmo padrão do *SGBDR*;
- c) para a cópia dos *backup*'s para uma estação *Windows*, o aplicativo *Samba* deve estar configurado e o diretório destino deverá estar com o mapeamento configurado;
- d) a estrutura lógica e física da rede deve estar funcionando corretamente;
- e) para o envio da base de dados para o servidor *FTP* da Tecmicro, o acesso por este protocolo deve estar liberado para o servidor;
- f) deverá ser criada uma pasta na estação de *backup* que conterà os arquivos e pastas necessárias para a configuração do sistema e armazenamento de arquivos de *backup*;
- g) a pasta que irá conter os arquivos no servidor *Apache* deverá estar protegida por senha;
- h) para restaurar arquivos com mais de 512 *KB* de estações através de envio via ferramenta, as configurações do *Apache* e do *PHP* deverão ser alteradas;
- i) o usuário padrão do *PostgreSQL* deverá ter direitos de administrador.

3.3 ESPECIFICAÇÃO

Após a obtenção das informações das funções que o TECgres deve realizar para o auxílio de técnicos e consultores, foi confeccionado o diagrama de caso de uso, em seguida o modelo conceitual necessário para o armazenamento das informações e com isso, criado o modelo de entidade e relacionamento. Todas essas três etapas foram realizadas com a utilização da ferramenta *PowerDesigner 9* da *Sybase*.

3.3.1 Caso de uso

Do inglês *use-case*, para Furlan (1998), têm como propósito descrever os requisitos funcionais do sistema de maneira consensual entre os usuários e desenvolvedores de sistemas, visando fornecer uma descrição consistente e clara sobre as responsabilidades que devem ser cumpridas pelo sistema.

A figura 1 apresenta o caso de uso da ferramenta TECgres.

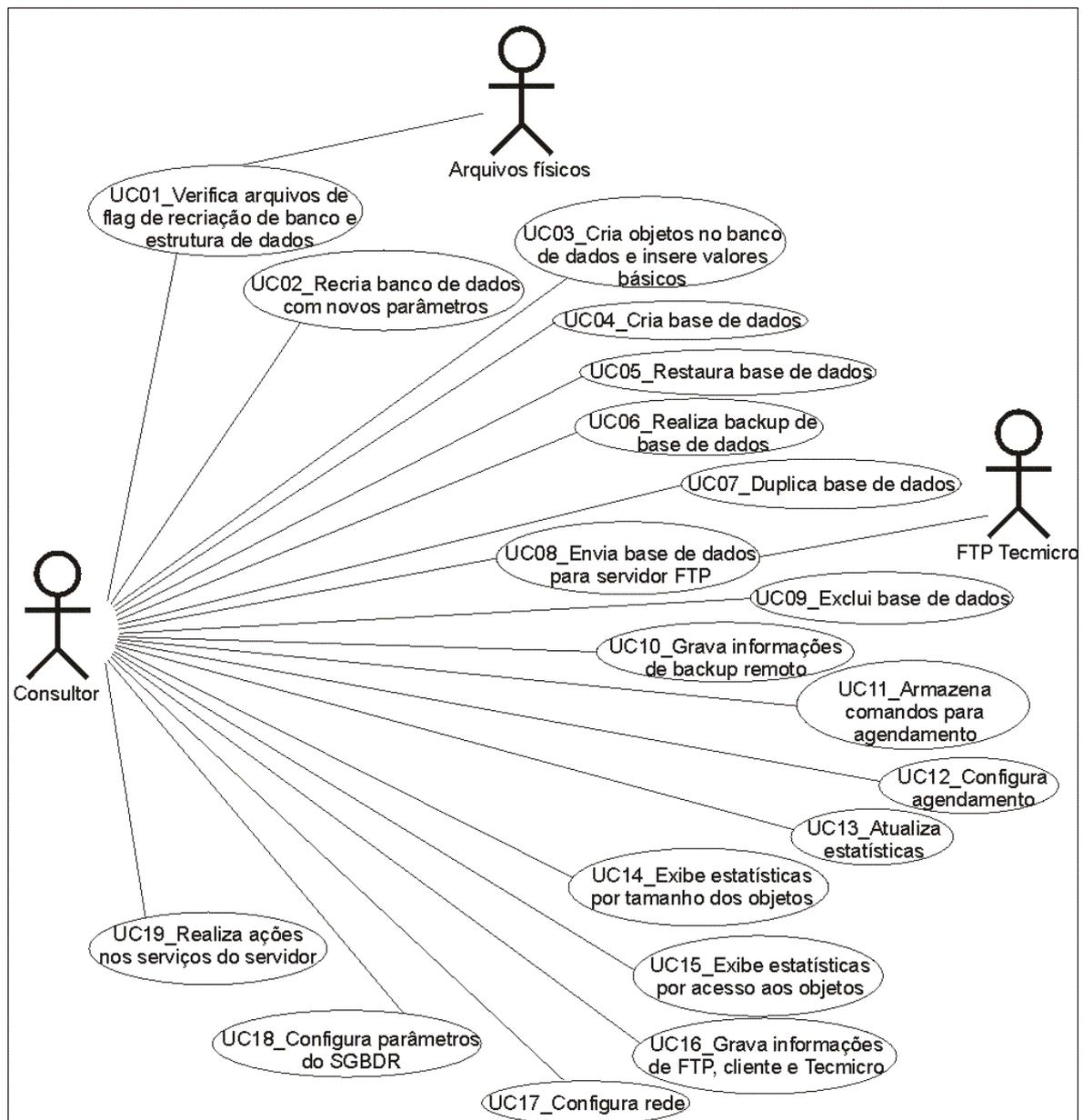


Figura 1: Caso de uso

3.3.2 Modelo Conceitual

Para o armazenamento das informações e como próximo passo, foi criado o modelo conceitual, definindo os objetos de dados e suas agregações necessárias para o sistema. Este primeiro nível – ou passo – trata-se do mais abstrato no que se refere a modelagem de dados.

A ferramenta Tecgres utiliza-se de uma estrutura simples dados, porém necessária para armazenar todas as informações de configuração, este modelo pode ser visualizado na figura 2. Toda a definição dos atributos e entidades estão listadas no item 3.4, dicionário de dados.

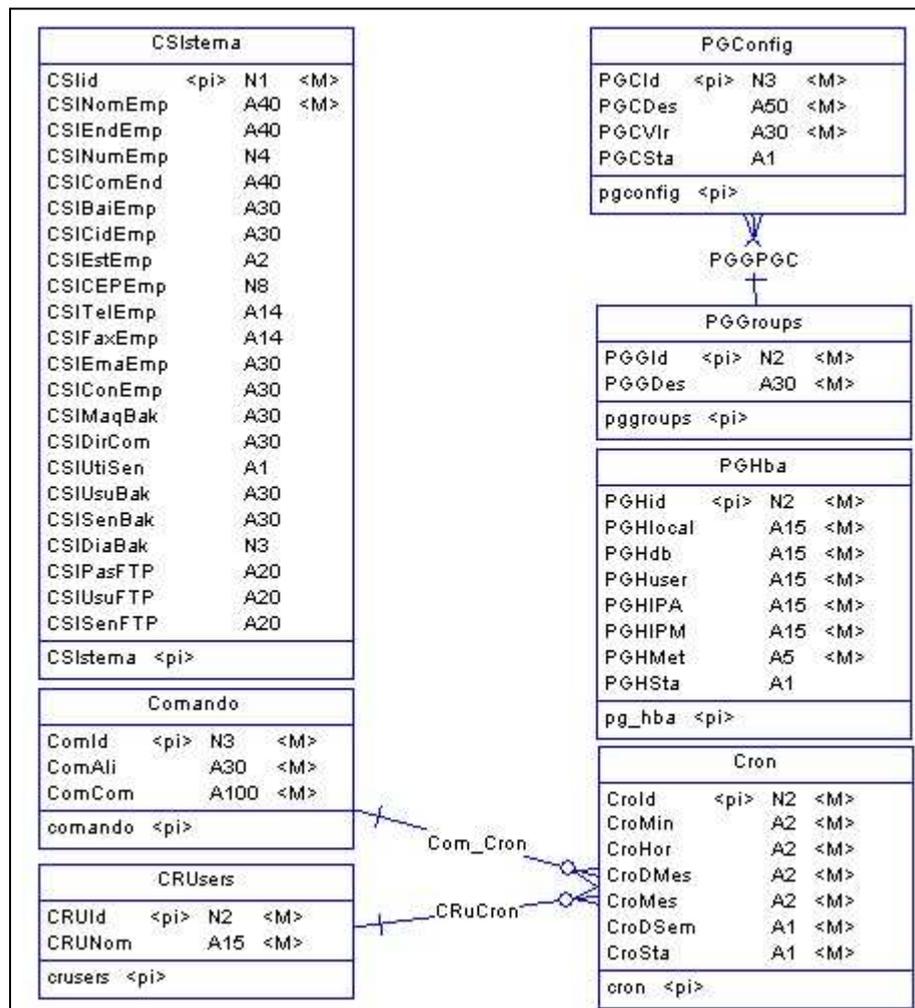


Figura 2: Modelo conceitual

3.3.3 MER

A partir da definição do modelo conceitual, foi gerado no *PowerDesigner 9*, o modelo de entidade relacionamento, definindo assim os objetos no *SGBDR* para o funcionamento da

ferramenta. Neste modelo é possível visualizar a estrutura física das entidades e seus relacionamentos, na figura 3 é possível visualizar este modelo.

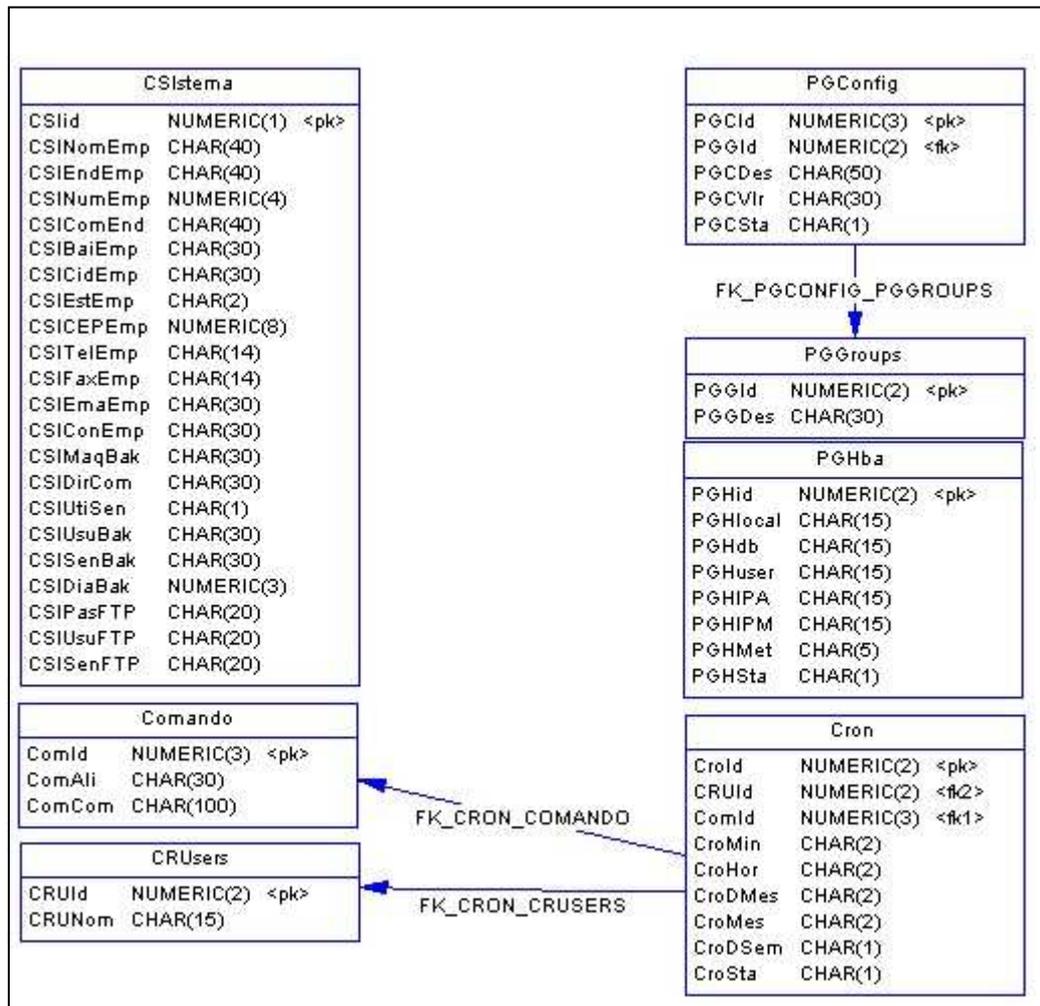


Figura 3: Modelo de entidade relacionamento

3.4 DICIONÁRIO DE DADOS

O dicionário de dados contém as definições dos objetos do modelo de entidade relacionamento da ferramenta TECgres, com suas entidades e atributos.

Nos quadros a seguir, é possível visualizar o dicionário de dados da ferramenta.

Entidade CSistema

Nome	Definição	Tipo de dado	Atributo(s)
CSId	Identificador da empresa	N1	PK
CSINomEmp	Nome da empresa	C40	

CSIEndEmp	Endereço da empresa	C40	
CSINumEmp	Número do endereço da empresa	N4	
CSIComEmp	Complemento do endereço da empresa	C40	
CSIBaiEmp	Bairro da empresa	C30	
CSICidEmp	Cidade da empresa	C30	
CSIEstEmp	Estado da empresa	C2	
CSICEPEmp	CEP da empresa	N8	
CSITelEmp	Telefone da empresa	C14	
CSIFaxEmp	Fax da empresa	C14	
CSIEmaEmp	E-mail da empresa	C30	
CSIConEmp	Contato na empresa	C30	
CSIMaqBak	Máquina de <i>backup</i> remoto	C30	
CSIDirCom	Diretório de <i>backup</i> remoto	C30	
CSIUtiSen	Se diretório com senha	C1	
CSIUsuBak	Usuário do diretório na máquina remota	C30	
CSISenBak	Senha do usuário na máquina remota	C30	
CSIDiaBak	Dias de permanência do <i>backup</i>	N3	
CSIPasFTP	Pasta no servidor <i>FTP</i> da Tecmicro	C20	
CSIUsuFTP	Usuário do servidor <i>FTP</i> da Tecmicro	C20	
CSISenFTP	Senha do usuário do <i>FTP</i> da Tecmicro	C20	

Quadro 1: Dados das empresas

Na entidade CSistema além de dados para simples informações do cliente e da Tecmicro, armazena informações que são utilizadas para a realização de cópia do *backup* para uma estação remota, dias que os arquivos de *backup* ficarão armazenados e as informações de *login* para o uso no envio de base de dados para a Tecmicro.

Entidade PGGroups

Nome	Definição	Tipo de dado	Atributo(s)
PGGId	Identificador do grupo	N2	PK
PGGDes	Descrição do grupo	C30	

Quadro 2: Grupos da configuração do SGBDR PostgreSQL

Os parâmetros de configuração do PostgreSQL são organizadas em grupos para uma melhor visualização, a ferramenta Tecgres também organiza esses grupos para uma melhor visualização do usuário, estes grupos são armazenados na entidade PGGroups.

Entidade PGConfig

Nome	Definição	Tipo de dado	Atributo(s)
PGCIId	Identificador da configuração	N3	PK
PGGIId	Identificador do grupo	N2	FK
PGCDes	Descrição da configuração	C50	
PGCVlr	Valor da configuração	C30	
PGCSta	<i>Status</i> da configuração	C1	

Quadro 3: Configurações do SGBDR PostgreSQL

A entidade PGGconfig armazena as configurações do PostgreSQL para gerar o arquivo final que o SGBDR utilizará. Além dos configurações e seus valores, esta entidade também armazena o grupo de cada configuração e se ela estará ativa ou não.

Entidade PGHba

Nome	Definição	Tipo de dado	Atributo(s)
PGHId	Identificador da configuração	N2	PK
PGHLocal	Tipo de <i>host</i>	C15	
PGHDb	Base de dados de acesso	C15	
PGHUser	Usuário de acesso	C15	
PGHIPA	IP de acesso	C15	
PGHIPM	Máscara do IP de acesso	C15	
PGHMet	Método de conexão	C5	
PGHSta	<i>Status</i> da configuração	C1	

Quadro 4: Configuração de acesso remoto do SGBDR PostgreSQL

Antes de ser criado o arquivo que o SGBDR usa para as configurações de acesso da rede, as informações são gravadas na entidade PGHba. Esta entidade contém todas as informações necessárias para a configuração do acesso da rede ao SGBDR.

Entidade Comando

Nome	Definição	Tipo de dado	Atributo(s)
ComId	Identificador do comando	N3	PK
ComAli	Apelido do comando	C30	
ComCom	Comando	C100	

Quadro 5: Comandos para o agendamento

A entidade Comando tem como finalidade facilitar o agendamento as ações no servidor e na implementação da ferramenta no servidor, os *scripts* de *backup* já estão armazenados nesta entidade e prontos para o uso no agendamento, facilitando a configuração. Nela fica apenas armazenado um apelido para o comando ou *script* e seu caminho completo.

Entidade CRUsers

Nome	Definição	Tipo de dado	Atributo(s)
CRUId	Identificador do usuário	N2	PK
CRUNom	Nome do usuário para o agendamento	C15	

Quadro 6: Usuários para o agendamento

Como o agendamento pode ser efetuado para vários usuários do sistema operacional, a entidade CRUsers armazena os usuários que serão usados no agendamento, como padrão da ferramenta, os usuários *postgres* e *root*.

Entidade Cron

Nome	Definição	Tipo de dado	Atributo(s)
CroId	Identificador do agendamento	N2	PK
CRUId	Identificador do usuário	N2	FK
ComId	Identificador do comando	N3	FK
CroMin	Minuto do agendamento	C2	
CroHor	Hora do agendamento	C2	
CroDMes	Dia do mês do agendamento	C2	
CroMes	Mês do agendamento	C2	
CroDSem	Dia da semana do agendamento	C5	
PGHSta	<i>Status</i> do agendamento	C1	

Quadro 7: Agendamento no sistema operacional

A entidade Cron reúne as informações do agendamento para a criação dos arquivos que serão usados para os respectivos usuários. Nela contém as informações de quando será executado, o comando que será executado e se este agendamento estará ativo ou não.

Além de facilitar a manipulação, as informações estão armazenadas em banco de dados para garantir o *backup* de qualquer configuração que seja realizado no SGBDR. Elas estarão em todos os *backup's* realizados da base de dados utilizados pelos sistemas da Tecmicro.

3.5 IMPLEMENTAÇÃO

Como citado anteriormente, foi realizada uma combinação de linguagens para atender as necessidades de criação desta ferramenta. Com exceção de *Linux Scripts* e *SQL* para a criação dos objetos no *SGBDR*, todos os outros códigos necessários foram criados através do *Dreamweaver MX 2004*.

Para um melhor aproveitamento do *Dreamweaver MX 2004*, foi configurado nele os diretórios locais e remotos onde estariam sendo salvos os arquivos e para uma melhor agilidade, ao serem salvos os arquivos, os mesmos são atualizados automaticamente no servidor. Para esta ação, foi configurada a rede *Windows* no lado do desenvolvimento e a rede *Samba* no lado do servidor.

Na figura 4 apresenta-se o navegador de objetos, mostrando os arquivos e diretórios remotos.



Figura 4: Navegador dos objetos no *Dreamweaver MX 2004*

Para a criação visual da ferramenta em *HTML*, foi utilizado largamente o editor gráfico do *Macromedia Dreamweaver MX 2004*, possibilitando assim uma *interface* mais agradável para o usuário e um desenvolvimento mais rápido para o desenvolvedor.

Para a edição de alguns códigos em *HTML* e todos em *PHP*, *JavaScript* e comandos de linha para o *Linux*, utilizou-se o editor de código do *Dreamweaver*. Além de uma cor diferenciada para os tipos de comandos e linguagens existentes, faz a endentação e correção de alguns comandos. A utilização dessas quatro linguagens pode ser apreciada na figura 5.

```

12     } else {
13         alert("O processo encerrou com sucesso");
14     }
15     window.close();
16 }
17 </script>
18 </head>
19 <body onLoad="javascript:termino(<?php echo $status; ?>);">
20 <?php
21     $dir = date("Hidmy");
22     system("sudo sh /etc/init.d/postgresql stop");
23     echo "<br>";
24     system("sudo mv /var/lib/pgsql/data /var/lib/pgsql/data$dir");
25     system("sudo cp /etc/sysconfig/il8n /etc/sysconfig/il8nl");

```

Figura 5: Editor de comando do *Dreamweaver*

Para a criação dos *Scripts Shell* foi utilizada o editor de texto padrão do *Linux Red Hat 9*, *vim*. Na figura 6 podemos identificar através das cores, variáveis, comandos *Linux* e *SQL*.

```

MAQUINA= `expr substr "$MAQUINA" 2 31`
DIRETORIO= `psql tecmicro -t -c "select trim(csidircom) from dbtec.csis
DIRETORIO= `expr substr "$DIRETORIO" 2 31`
BKPREMOTO= `psql tecmicro -t -c "select csutisen from dbtec.csistema w
BKPREMOTO= `expr substr "$BKPREMOTO" 2 2`
if [ -d /mnt/backup/logico ];
then
    sudo smbmount /mnt/backup
fi
if [ "$BKPREMOTO" = "3" ]
then
    USUARIO= `psql tecmicro -t -c "select csiusubak from dbtec.csis
    USUARIO= `expr substr "$USUARIO" 2 31`
    SENHA= `psql tecmicro -t -c "select csisenbak from dbtec.csiste
    SENHA= `expr substr "$SENHA" 2 31`
    sudo smbmount //$MAQUINA/$DIRETORIO /mnt/backup -o username="$
else
    sudo smbmount //$MAQUINA/$DIRETORIO /mnt/backup -o guest
fi
sudo cp --reply=no /backup/logico/* /mnt/backup/logico/
sudo smbmount /mnt/backup
~
~
~

```

Figura 6: Editor de arquivo texto no *Linux*

O SGBDR PostgreSQL possibilita a execução de vários comandos diretamente no *shell* do sistema operacional. No desenvolvimento da ferramenta TECgres este conceito foi largamente utilizado.

Nas figuras a seguir (7, 8 e 9) é possível visualizar vários comandos sendo executados diretamente no sistema operacional. Estes comandos podem ser visualizados a partir do

comando *system* do PHP. Os comentários dos principais comandos estão nas respectivas figuras.

```

9 <?php
10 // recebe base de origem
11 $origem = $_HTTP_GET_VARS["origem"];
12 // recebe base de destino
13 $destino = $_HTTP_GET_VARS["destino"];
14 // monta a partir da data e hora do sistema o nome do arquivo
15 $arquivo = "/backup/tmp".date("Ddm_Hi").".gz";
16 // gera um backup compactado da base de origem
17 system("pg_dump $origem | gzip > $arquivo");
18 // restaura para a base de destino o arquivo criado anteriormente
19 system("gunzip -c $arquivo | psql $destino");
20 // remove o arquivo criado para a migração dos dados
21 system("rm -f $arquivo");
22 ?>

```

Figura 7: Duplicação de base de dados

```

1 <?php
2 //recebe o nome da base para ser restaurada
3 $database = $_HTTP_GET_VARS["database"];
4 // recebe o arquivo a ser restaurado
5 $arquivoremoto = $_HTTP_GET_VARS["arquivoremoto"];
6 // recebe o tipo de arquivo
7 $tipofile = $_HTTP_GET_VARS["tipofile"];
8 // recebe se o arquivo deve ser convertido para Unix
9 $unix = $_HTTP_GET_VARS["unix"];
10 if ($tipofile == 'txt') {
11     if ($unix == 1) {
12         // converte o arquivo para unix através de comando do SO
13         system("sudo dos2unix $arquivoremoto");
14         // atribui o usuário postgres como dono do arquivo convertido
15         system("sudo chown postgres.postgres $arquivoremoto");
16     }
17     // restaura o arquivo para a base informada
18     system("psql $database < $arquivoremoto");
19 }
20 if ($tipofile == 'gz') {
21     // restaura o arquivo gz para a base descompactando-o
22     system("gunzip -c $arquivoremoto | psql $database");
23 }

```

Figura 8: Conversão de arquivo e restauração de base de dados

```

1 <?php
2 // remove o arquivo de backup da configuração atual
3 system("rm -f /var/lib/pgsql/data/postgresql.conf.bak");
4 // cria um backup do atual
5 system("mv /var/lib/pgsql/data/postgresql.conf /var/lib/pgsql/data/postgresql.conf.bak");
6 // cria e insere um alerta no novo arquivo de configuração
7 system('echo "#altere este arquivo somente usando a ferramenta da tecgres da tecmicro sist
8 // abre o arquivo de configuração em modo de inserção
9 $file = fopen('/var/lib/pgsql/data/postgresql.conf', 'a');
10 // conecta no SGBDR
11 include("../includes/php/conecta.php");
12 // lê a tabela de configurações do SGBDR
13 $sql = "select * from tecgres.pgconfig order by 1";
14 $result = pg_query($conexao, $sql);
15 while ($row = pg_fetch_array($result)) {
16     if ($row[pgcsta] == '#') {
17         $text = $row[pgcsta].trim($row[pgcdes])." = ".trim($row[pgcvlr])."\n";
18     } else {
19         $text = trim($row[pgcdes])." = ".trim($row[pgcvlr])."\n";
20     }
21     // escreve no arquivo a variavel text decorrente da concatenação anterior
22     fwrite($file, $text);
23 }
24 // fecha o arquivo
25 fclose($file);
26 // reinicia o SGBDR
27 system("sudo sh /etc/init.d/postgresql restart");
28 >>

```

Figura 9: Recriação do arquivo de configuração do SGBDR PostgreSQL

3.6 PRINCIPAIS FUNCIONALIDADES

Este item está destinado à apresentação das principais funcionalidades da ferramenta TECgres, iniciando com seu menu de navegação e seguindo por algumas funcionalidades.

3.6.1 Navegação

Para o uso da ferramenta TECgres, sua navegabilidade divide-se em dois níveis de menu, um primário e outro secundário, conforme pode ser visto na figura 10.

Primário	Secundário
Databases	Criar Restaurar Backup Duplicar Enviar FTP Excluir
Backup	Dados Comandos Agendamento
Estatísticas	Tamanho Acesso
Configurações	Dados Rede Banco de Dados Ações
Sair	

Figura 10: Menu de navegação

3.6.2 Manipulação de Base de Dados

Estando como um dos primeiros objetos a ser criado em um *SGBDR*, a base de dados – ou *database* – tem como finalidade armazenar todos os objetos, tais como tabela, visões e índices. As principais funcionalidades para este objeto do *SGBDR* que a ferramenta *TECgres* contempla são a restauração, duplicação e envio via *FTP* de base de dados.

A restauração, conforme pode ser visto na figura 11, pode decorrer de um *backup* anterior ou de um arquivo externo. Para a restauração a partir de um backup anterior, o sistema lista os arquivos contidos no diretório em que são armazenados os *backup's* agendados e os solicitados em tempo real. Para a restauração através de um arquivo, existe a possibilidade de restaurar uma base de dados a partir de arquivos texto nos formatos *Unix* e *MS-DOS* – convertendo para o formato *Unix* –, ambos podendo ser compactados, no *Gzip* e *Zip* para o formato *Unix* e *Zip* para *MS-DOS*.

Restaurar Database:

Esta ferramenta irá restaurar base de dados a partir de arquivos texto. Informe em qual database deseja restaurar:

Para restaurar a base de um arquivo externo, selecione a base de dados, o arquivo que deseja restaurar e o tipo de arquivo.

Database	tecmicro	
Arquivo	fi\angelina.txt	Procurar...
Tipo de arquivo	Texto	<input checked="" type="checkbox"/> Converter para Unix

Restaurar

Para restaurar a base de um backup feito através do sistema, selecione a base e o arquivo que deseja restaurar.

Database	tecmicro	
Arquivo	TEC07May1742.gz	

- TEC07May1742.gz
- TEC07May1744.gz
- TEC08Jun2042.gz
- TEC19Jun2023.gz

Figura 11: Restauração de base de dados

Para a realização de testes de atualização do sistema de informação de uma empresa, a importância de uma base de testes é primordial. Partindo da base de dados oficial do *SGBDR*, é possível através da ferramenta a duplicação dessa base para uma outra recém criada, onde o usuário seleciona uma base de dados de origem e uma de destino e a ferramenta se encarrega dessa duplicação. Esta ação pode ser vista na figura 12.

Duplicar Database:

Origem	Destino
tecmicro	pttecmicro

Duplicar

Figura 12: Duplicação de base de dados

Finalizando as principais funcionalidades da ferramenta para a administração de base de dados, o recebimento de base de dados de clientes na Tecmicro Sistema é uma rotina quase que diária. Para as bases em *Microsoft Access* este processo é simples, pois se trata apenas de compactar um arquivo e enviar via *e-mail* ou servidor *FTP*, mas para um *SGBDR* a ação é bem diferente. A ferramenta não possibilita apenas o envio de base de dados de *backup's* anteriores realizados via agendamento ou em tempo real, mas também no momento exato do

envio, realizando um *backup* e enviando-o para o servidor *FTP* da empresa Tecmicro Sistemas. A figura 13 ilustra esta funcionalidade.



Figura 13: Envio de base de dados via *FTP*

3.6.3 Configuração de *backup*

Em uma rotina de *backup*, o salvamento dos arquivos necessários para recuperar uma base de dados em outra máquina torna-se fundamental. Com a possibilidade de cópia de arquivos entre os sistemas operacionais *Linux* e *Windows* através de uma rede *Samba*, a ferramenta *TECgres* agiliza esta operação.

Informando os dados necessários – conforme figura 14 – para a ação de transferência de arquivos entre servidor e estação, a ferramenta realiza esta cópia no horário em que esta ação foi agendada. Outra funcionalidade trata-se dos dias em que os arquivos de *backup* deverão permanecer salvos, esta informação é utilizada por outro *script* da ferramenta para a exclusão de arquivos mais antigos do que o informado.

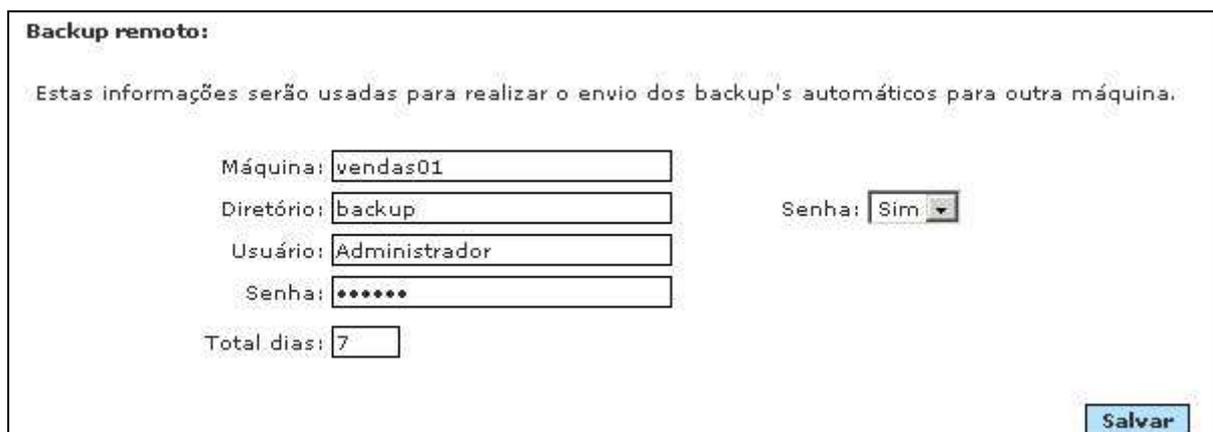


Figura 14: Informação para *backup*

Conforme pode ser visto na figura 15, além dos comandos padrões da ferramenta, há a possibilidade de cadastrar comandos externos para serem usados no agendamento via TECgres, que contém uma *interface* para cadastrar um comando com um apelido – *alias*.

Estes comandos serão disponibilizados para o uso no agendamento da máquina:

Alias	Comando	Excluir
Backup Logico	sh /backup/scripts/logico.sh	<input checked="" type="checkbox"/>
Backup Logico Tecmicro	sh /backup/scripts/logico.sh tecmicro	<input checked="" type="checkbox"/>
Backup Fisico	sh /backup/scripts/fisico.sh	<input checked="" type="checkbox"/>
Copia Logico pra Rede	sh /backup/script/cplogico.sh	<input checked="" type="checkbox"/>
Copia Fisico para Rede	sh /backup/script/cpfisico.sh	<input checked="" type="checkbox"/>
Remove Backups Antigos	sh /backup/scripts/rmantigo.sh	<input checked="" type="checkbox"/>
Reiniciar Servidor	sudo reboot	<input checked="" type="checkbox"/>

Incluir comando:

Alias	Comando
Desligar servidor	sudo shutdown -h now

Figura 15: Comandos a serem usados no agendamento

Após inserir os comandos na base de dados da ferramenta, é possível utilizá-lo no agendamento do sistema operacional através de uma *interface* mais simples que a edição de arquivo texto no *Linux*. Nesta *interface* com uma ação muito simples podemos cancelar – sem eliminar – um agendamento, caracterizado com um sinal de atenção na frente da linha. Podemos visualizar esta interface na figura 16.

Ativo	Hora	Minuto	Dia	Mês	Dia da Semana	Usuário	Comando	Ação
<input checked="" type="checkbox"/>	12	00	31	Todos	Todos	postgres	Backup Físico	
	12	45	*	Todos	Todos	postgres	Copia Logico pra Rede	

Para inserir novas configurações de acesso no PostgreSQL, informe os valores abaixo. Estes valores só estarão após confirmar a configuração no botão mais abaixo.

Hora	Minuto	Dia	Mês	Dia da semana	Usuário	Comando
14	00	*	Todos	Sábado	postgres	Desligar servidor

- Backup Logico
- Backup Logico Tecmicro
- Backup Físico
- Copia Logico pra Rede
- Copia Físico para Rede
- Remove Backups Antigos
- Reiniciar Servidor
- Desligar servidor

Figura 16: Agendamento de tarefas no servidor

3.6.4 Estatísticas

As estatísticas auxiliam na velocidade de acesso à informação em um *SGBDR*, a ferramenta *TECgres* possibilita a atualização e verificação dessas para a realização de um ajuste. As figuras 17 e 18 permitem visualizar as formas que podem ser vistas as estatísticas do *PostgreSQL*, visualizando os objetos por tamanho e por acesso, respectivamente.

Ordenar por: Descrescente Total de linhas: 15

Objeto	MB's	Linhas	Arquivo
pg_depend_depender_index	0	2856	16622
pg_depend_reference_index	0	2856	16623
pg_depend	0	2856	16598
pg_proc	0	1492	1255
pg_proc_oid_index	0	1492	16639
pg_proc_prname_args_nsp_index	0	1492	16640
pg_description	0	1397	16416
pg_description_o_c_o_index	0	1397	16624
pg_attribute_relid_attnam_index	0	929	16608
pg_attribute_relid_attnum_index	0	929	16609
pg_attribute	0	929	1249
pg_operator_oid_index	0	643	16637
pg_operator_oprname_l_r_n_index	0	643	16638
pg_operator	0	643	16392
pg_amop_opc_opr_index	0	180	16603

Figura 17: Visualização de estatísticas por tamanho, retornando o arquivo para realização de ajuste

Ordenar por: Total de linhas:

Schema	Objeto	Leitura Sequencial	Registros Lidos	Bloco em Discos	Blocos em Cache
tecgres	pgconfig	1	102	3	0
tecgres	comando	4	32	1	3
tecgres	crusers	2	4	1	1
tecgres	csistema	2	4	1	1
tecgres	cron	1	2	1	0

Figura 18: Estatísticas por acesso

3.6.5 Liberação de acesso

A configuração de acesso local e de estações de trabalho é feita através de arquivo texto, na figura 19 é possível visualizar a *interface* criada para esta tarefa, com os campos determinados. Após a confirmação, a ferramenta recria o arquivo do *PostgreSQL* com os parâmetros informados.

Ativo	Local	Database	Usuário	IP	Máscara	Método	Ação
<input checked="" type="checkbox"/>	host	all	all	192.168.0.90	255.255.255.0	sem senha	<input type="button" value="▼"/>
<input checked="" type="checkbox"/>	host	all	all	192.168.0.0	255.255.255.0	com senha	<input type="button" value="▼"/>

Para inserir novas configurações de acesso no PostgreSQL, informe os valores abaixo. Estes valores só estarão após confirmar a configuração no botão mais abaixo.

Local	Database	Usuário	IP	Máscara	Método
<input type="text" value="host"/>	<input type="text" value="tecmicro"/> ▼	<input type="text" value="todos"/> ▼	<input type="text" value="192.168.0.91"/>	<input type="text" value="255.255.255.0"/>	<input type="text" value="Sem senha"/> ▼

Figura 19: Liberação de acesso.

3.6.6 Configuração de parâmetros do SGBDR

As configurações de memória, estatística e *log* entre outros, são também realizadas em arquivo texto, para a simplificação desta tarefa, a ferramenta *TECGres* possui uma *interface* para esta configuração, listando as possíveis configurações, liberando ao usuário além da configuração dos valores para cada parâmetro, a possibilidade de apenas desativar determinada configuração. Na figura 20 é possível visualizar alguns parâmetros ativos – indicado pela figura verde na frente – e inativos – figura amarela.

Connection Parameters			
<input checked="" type="checkbox"/>	tcpip_socket	true	Ok Desativar
<input type="checkbox"/>	ssl	false	Ok Ativar
<input checked="" type="checkbox"/>	max_connections	5	Ok Desativar
<input type="checkbox"/>	superuser_reserved_connections	2	Ok Ativar
<input type="checkbox"/>	port	5432	Ok Ativar
<input type="checkbox"/>	hostname_lookup	false	Ok Ativar
<input type="checkbox"/>	show_source_port	false	Ok Ativar
<input type="checkbox"/>	unix_socket_directory	"	Ok Ativar
<input type="checkbox"/>	unix_socket_group	"	Ok Ativar
<input type="checkbox"/>	unix_socket_permissions	0777	Ok Ativar
<input type="checkbox"/>	virtual_host	"	Ok Ativar
<input type="checkbox"/>	krb_server_keyfile	"	Ok Ativar
Shared Memory Size			
<input checked="" type="checkbox"/>	shared_buffers	1024	Ok Desativar
<input type="checkbox"/>	max_fsm_relations	1000	Ok Ativar
<input type="checkbox"/>	max_fsm_pages	10000	Ok Ativar

Figura 20: Configuração de parâmetros do PostgreSQL

3.6.7 Ações no servidor

Como última das principais funções da ferramenta, na figura 21 podemos visualizar os serviços do servidor que estão relacionados com os sistemas ou ferramentas da Tecmicro Sistema. Foi observada a dificuldade de clientes e desenvolvedores em realizar as funções de iniciar, parar ou reiniciar um serviço ou o servidor *Linux*. Com a ferramenta TECgres esta tarefa é realizada em apenas um clique.

PostgreSQL - Banco de Dados	Iniciar	Reiniciar	Baixar
Apache - Servidor Web			Baixar
Samba - Servidor de Arquivos	Iniciar	Reiniciar	Baixar
Servidor		Reiniciar	Desligar

Figura 21: Serviços do servidor

4 CONSIDERAÇÕES FINAIS

4.1 CONCLUSÕES

Durante o projeto e desenvolvimento deste trabalho ficou evidenciado a possibilidade de criação de uma ferramenta com uma grande versatilidade de linguagens, possibilitando criar uma interface *Web* intervindo em arquivos físicos e serviços de um servidor remoto. Tudo isso face à execução de comandos de linha no *Linux* através do servidor *Apache*, executando comando em *PHP*.

Todos os objetivos iniciais da ferramenta foram alcançados e validada perante consultores e técnicos da empresa Tecmicro Sistemas. A ferramenta se mostrou útil para manipulação de base de dados e configuração de *backup* e do *SGBDR PostgreSQL*. A declaração de aprovação da empresa está contida anexo a este trabalho.

4.2 EXTENSÕES

Como sugestão para trabalhos futuros, duas linhas poderão ser seguidas: uma pré-ferramenta e a outra um aprimoramento da mesma. Para a primeira situação, o atendimento dos requisitos não funcionais, criando um instalador simplificado, alterando configuração do servidor *Apache*, módulo *PHP* e sistema operacional *Linux*. Como aprimoramento, extender suas funcionalidades, tanto no *PostgreSQL*, com a criação de um monitoramento do *SGBDR*, quanto outros serviços como servidor de arquivos *Samba*.

REFERÊNCIAS BIBLIOGRÁFICAS

BALL, Bill. **Usando Linux**. Rio de Janeiro: Campus, 1991.

DATE, C. J. **Introdução a sistemas de banco de dados**. Rio de Janeiro: Campus, 1990.

DATE, C. J. **Introdução a sistemas de banco de dados**. Rio de Janeiro: Campus, 2000.

FURLAN, José David. **Modelagem de objetos através da UML**. São Paulo: Makron Books, 1998.

KABIR, Mohammed J. **Apache Server 2, a Bíblia**. Rio de Janeiro: Campus, 2002.

LEITE, Leonardo Lellis Pereira. **Introdução aos sistemas de gerência de banco de dados**. São Paulo: E. Blucher, 1980.

LOWERY, Joseph. **Dreamweaver MX: a Bíblia**. Rio de Janeiro: Campus, 2002.

PEREIRA NETO, Álvaro. **PostgreSQL: técnicas avançadas: versões open source: soluções para desenvolvedores e administradores de banco de dados**. São Paulo: Érica, 2003.

RAMALHO, José Antônio. **Oracle 8i**. São Paulo: Berkeley Brasil, 1999.

SILVA, Osmar. **JavaScript avançado: animação, interatividade e desenvolvimento de aplicativos**. São Paulo: Érica, 2003.

VENETIANER, Tomas. **HTML: desmistificando a linguagem da Internet**. São Paulo: Makron Books, 1997.

ANEXO A – DECLARAÇÃO DA TECMICRO SISTEMAS

Eu, Leandro Merico, diretor-presidente da empresa desenvolvedora de softwares Tecmicro Sistemas, declaro que a ferramenta TECgres desenvolvida por Reuly Bússolo Mendes para conclusão do curso de Sistemas de Informação, demonstra confiabilidade e utilidade para a empresa, sendo realizado diversos testes e disponibilizada para o uso de consultores.

Leandro Merico – Diretor Presidente
Tecmicro Sistemas