

**Configurando as páginas e o
buffer (cache) no Firebird**

Novidades do InterBase 2007

**Instalação do PostgreSQL 8.0
para Win32 (tutorial)**

Ganhe R\$ 10,00 de
desconto na
inscrição para o SQL
Connect 2007. Use o
cupom DBFREE10.

Editorial

Esta edição está sendo lançada especialmente para divulgar o primeiro evento promovido pela **DB FreeMagazine**. O **SQL Connect 2007** será realizado em Piracicaba, interior de São Paulo (160km da capital), no teatro da UNIMEP, no dia **31 de Março de 2007** (sábado). O evento trará palestras relacionadas não somente aos principais SGBDs, mas também a todas as tecnologias relacionadas ao tema, incluindo replicação, desenvolvimento de software, dicas de SQL, etc.

Os leitores desta edição terão direito a um desconto de R\$ 10,00 na inscrição para o evento. Basta informar o código do cupom **DBFREE10**. E não esqueça de que o quanto antes você fizer sua inscrição, menos pagará, por isso não deixe para a última hora.

A próxima edição da revista deverá sair em entre Abril e Maio, trazendo os slides das palestras que foram apresentadas no evento.

Contamos com a participação de todos vocês neste que será o primeiro de uma série de eventos dedicados aos Bancos de Dados.

Quanto ao conteúdo desta edição, temos um artigo sobre tamanho de páginas e buffer no Firebird, assunto que sempre desperta dúvidas entre os usuários deste SGBD.

Um pequeno artigo apresenta algumas das novidades do InterBase 2007, recém lançado pela CodeGear, que estará presente no evento SQL Connect falando sobre as novas versões do Delphi (2007 e Delphi for PHP).

Para os usuários de PostgreSQL, estamos publicando um tutorial de instalação para Win32 que deve facilitar a vida especialmente dos novos usuários que desejam usar o banco neste sistema operacional.

Boa leitura a todos!

Carlos H. Cantu
Março/2007

DB FREE MAGAZINE

Informações

DB FreeMagazine nº 010 - Ano III
Março/2007
Contato geral:
webmaster@dbfreemagazine.com.br

Equipe editorial

Carlos H. Cantu
(cantu@dbfreemagazine.com.br)
Luiz Paulo de Oliveira Santos
(lpaulo@dbfreemagazine.com.br)

Contribuíram nessa edição

- Carlos H. Cantu
- Mendo Leonel
- Cláudio Duarte de Freitas

É proibida a reprodução de qualquer parte do conteúdo dessa publicação sem autorização prévia por escrito.

Dica para melhor visualização

Utilize a resolução **1024x768** pixels e configure o *Acrobat Reader* para **Zoom** de **100%**. Feche todas as abas laterais e esconda as barras de ferramentas, liberando o máximo de área útil na tela, ou simplesmente rode a revista em modo *fullscreen*. Usuários de **Linux**: Recomendamos utilizar o **Acrobat Reader** para Linux a fim de garantir 100% de compatibilidade com o formato.

ANUNCIE NA DB FreeMagazine

Valorize seu produto ou serviço!

anuncios@dbfreemagazine.com.br

Configurando as páginas e o buffer (cache) no Firebird

por Carlos H. Cantu

Um arquivo de banco de dados do Firebird (geralmente com extensão *.fdb*) é composto por diversas páginas, de diferentes tipos, exemplo: página de dados, de blobs, de ponteiros, de índices, etc (ver **figura 1**). As páginas têm um tamanho fixo (definido em *bytes*), determinado no momento da criação de um novo banco de dados.

É comum usar 4096 bytes (4K) como valor “padrão” para o tamanho da página, sendo que a faixa de valores aceita pelo Firebird é de 1.024 *bytes* (1K) até 16K. Não existe regra ou fórmula mágica para definir o melhor tamanho para a página de um banco. A “lenda” diz que o ideal é definir a página com o mesmo tamanho de um *cluster* do HD, fazendo com que a leitura de um cluster equivale à leitura de uma página inteira do banco, no entanto, **isso não deve ser encarado como verdade absoluta**. O ideal é fazer testes com diferentes tamanhos de páginas e verificar qual ofereceu a melhor performance. No momento, minha opinião é de que páginas de **8K** sejam ideais.

A única forma de alterar o tamanho das páginas de um banco de dados já existente é através de um *backup* e *restore*. Para isso, no momento de restaurar o *backup*, usamos o parâmetro `-p` seguido pelo tamanho da página (em *bytes*, lembrando que 1K = 1024 *bytes*) que desejamos usar no banco que será restaurado, por exemplo:

```
gbak -c -p 4096 -user SYSDBA
-pas masterkey backup.gbk banco.fdb
```

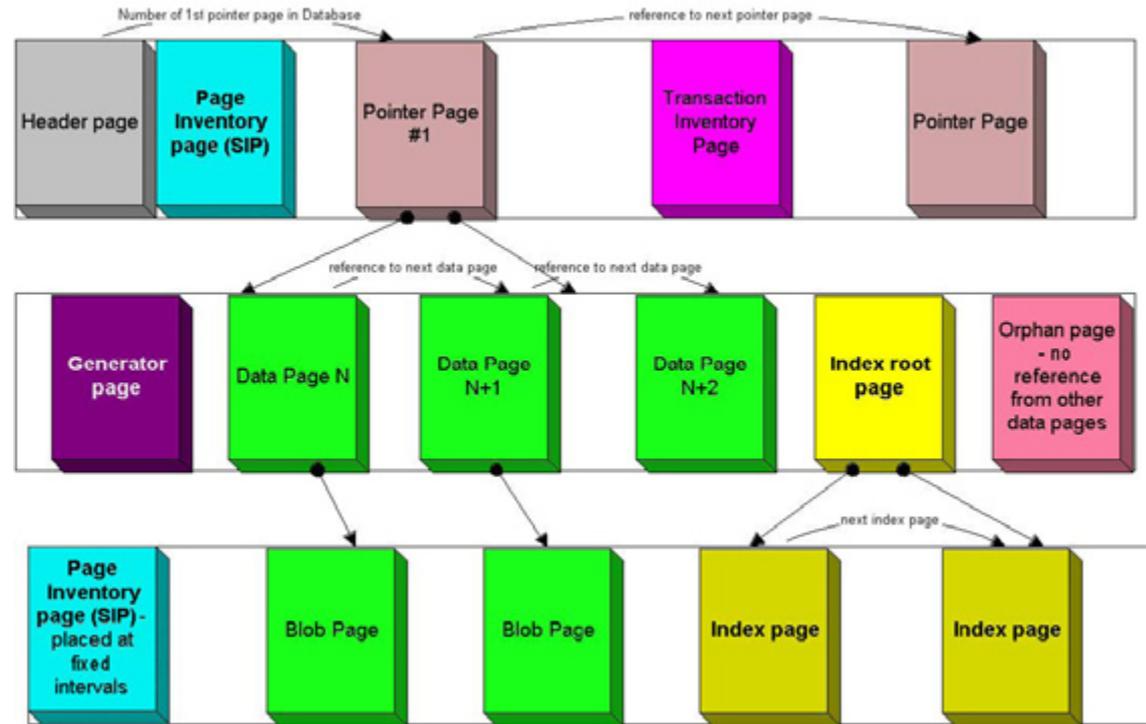


Figura 1. Estrutura de um banco de dados, listando os tipos de páginas

No exemplo anterior, o banco será criado com páginas de 4K (4.096 *bytes*).

O tamanho da página de um banco de dados tem influência direta no tamanho do *cache* de memória alocada pelo Firebird para aquele banco. Isto porque o tamanho do *buffer/cache* é especificado em número de páginas, portanto, quanto maior o tamanho da página, maior será a quantidade de memória alocada para o *buffer*. Para quem não está familiarizado com o conceito de *cache*, ele define uma quantidade de memória RAM utilizada para armazenar dados do BD, fazendo com que o Firebird gaste menos tempo acessando o HD, pois se a informação desejada estiver presente no *cache*, ela será lida da memória e não do disco.

O tamanho padrão do *buffer* é definido no arquivo *firebird.conf*, através do parâmetro

DefaultDbCachePages. Quando um novo banco de dados é criado, o valor do *buffer* definido no arquivo de banco de dados é zero, fazendo com que o Firebird utilize o valor de *DefaultDbCachePages* como padrão para o banco em questão. Podemos definir diferentes tamanhos de *buffer* para diferentes bancos de dados, bastando para isso usar o utilitário *gfix*, por exemplo:

```
gfix -user SYSDBA -pas masterkey
-buffers 8192 banco.fdb
```

O comando acima fará com que o *buffer* alocado para o banco de dados *banco.fdb* seja de 8.192 páginas. Isso quer dizer que se a página definida para *banco.fdb* for de 4096 *bytes*, o total de memória alocada pelo Firebird para *buffer* deste banco será de $8192 * 4096 = 32.768K$ ou 32MB.

Devemos lembrar também que no Firebird *ClassicServer*, cada conexão com o banco aloca um *buffer* próprio, portanto, no exemplo acima, se tivermos 10 conexões ativas ao mesmo tempo com o mesmo banco, o Firebird terá alocado $10 * 32MB = 320MB$ de RAM somente para os *buffers* dessas conexões. O *SuperServer* compartilha o mesmo *buffer* entre diferentes conexões de um mesmo banco, portanto o problema de consumo de memória é bem menor com o *SuperServer*.

O tamanho do *buffer* ainda pode ser especificado via API, através das seguintes funções:

- **isc_dpb_num_buffers:** interessante para ser usada com o *ClassicServer*, pois permite que cada conexão com o banco especifique um tamanho de *buffer* diferente. Se for usada com o *SuperServer*, terá efeito apenas se o banco de dados ainda não estiver aberto, e perderá seu efeito assim que o Firebird fechar a última conexão ativa com o banco.
- **isc_dpb_set_page_buffers:** tem a mesma função que o *gfix -buffers*.

A definição do tamanho do *buffer* deve ser feita com cautela, mais uma vez testando na prática diferentes valores, para se obter a melhor configuração. **Um buffer muito grande pode fazer com que o sistema operacional faça swap de memória, prejudicando a performance ao invés de melhorá-la!**

Dica

Nas versões do FB menor que 2.0, *buffers* definidos com mais de 10.000 páginas geravam queda de performance.

O problema foi corrigido no Firebird 2.

Implicações indiretas

Além do que já foi comentado, a mudança no tamanho da página do banco de dados possui implicações indiretas, muitas vezes ignoradas pelos desenvolvedores. Abaixo cito algumas delas:

- Aumentar o tamanho da página pode diminuir a profundidade dos níveis da árvore *b-tree* dos índices, agilizando as buscas.
- Bancos com páginas menores que 4K resultam em performance ruim. Nas versões mais antigas do Firebird, o tamanho padrão das páginas para bancos recém criados era de 1K. Fique atento e se for o caso, aumente o tamanho da página para esses bancos.
- Páginas maiores que 8K podem comprometer a performance das buscas, pois elas são feitas sequencialmente nas páginas. O Firebird 2.0 trouxe algumas mudanças nessa área e é menos afetado por esse “problema”.
- A partir do Firebird 2, o tamanho da página do banco de dados afeta diretamente o tamanho máximo que uma chave de um índice pode ter. O limite da chave é cerca de $\frac{1}{4}$ do tamanho da página, portanto páginas de 4K permitem chaves com no máximo 1024 bytes de extensão.

Conclusão

Fiquem atentos com a definição do tamanho das páginas e do *buffer* dos bancos de dados, pois ambos os parâmetros podem fazer uma grande diferença na performance das aplicações, tanto para mais, como para menos.

Faça testes com diferentes tamanhos de páginas e *buffers* em bases já populadas, a fim de determinar qual a configuração ideal para cada banco de dados.

Autor:

[Carlos Henrique Cantu](#)

Mini-curriculo

Bacharel em Ciência da Computação e pós-graduado em análise de Sistemas. Trabalha com desenvolvimento de sistemas há 16 anos. É consultor especializado em bancos de dados Firebird. Mantenedor do site FireBase, autor dos livros Firebird Essencial e Firebird 2, coordenador do DUG-BR, tendo ministrado palestras sobre Firebird para milhares de pessoas em diversas cidades do Brasil. Foi palestrante da Conferência Internacional de Firebird 2005 e 2006, realizadas em Praga (República Tcheca). Também é editor da DB FreeMagazine.

Avalie esse artigo

Vox on Demand

JobVox

Sistemas de automação de voz

Automação de discagem e emissão de recados	URA - Unidade Remota de Atendimento
Gravação de ligações telefônicas	

Site: <http://www.jobvox.com.br>
E-Mail: SAC@JOBVOX.COM.BR

Piracicaba
São Paulo

Novidades do InterBase 2007

por Mendo Leonel

O lançamento da versão 7.0 do InterBase trouxe inúmeras atualizações, deixando o SGBD mais robusto e com recursos interessantes, como: *multi-instance*, suporte SMP, novas funções (CASE, COALESCE e NULLIF), performance monitor, *savepoints* e muito mais. O produto, que continua em constante evolução, está agora nas mãos da CodeGear, subdivisão da Borland voltada para os desenvolvedores.

A versão mais recente do InterBase é a 2007, lançada recentemente, e neste artigo comentarei algumas das suas novidades.

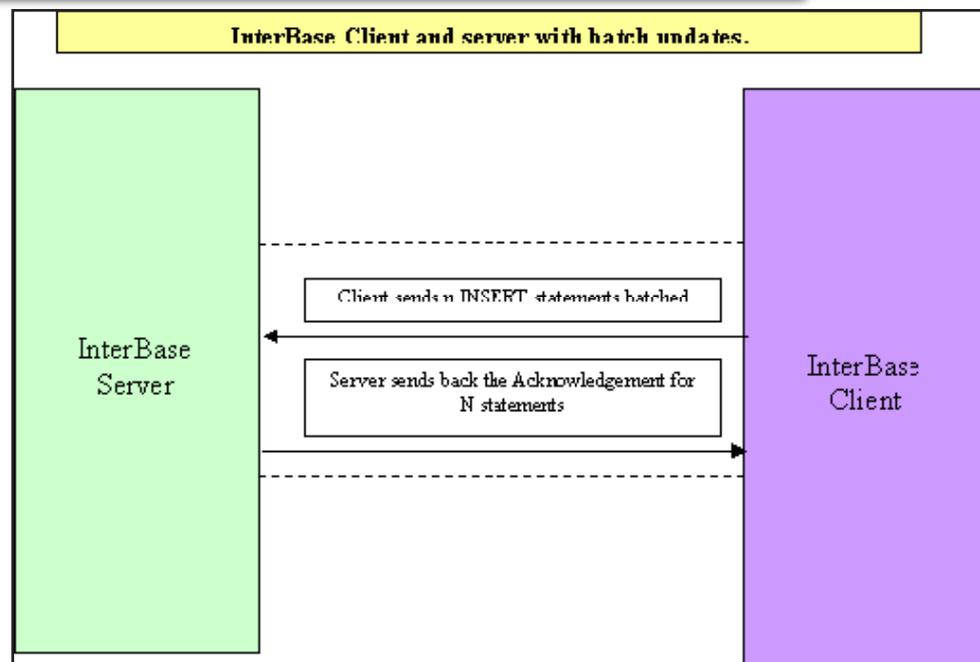
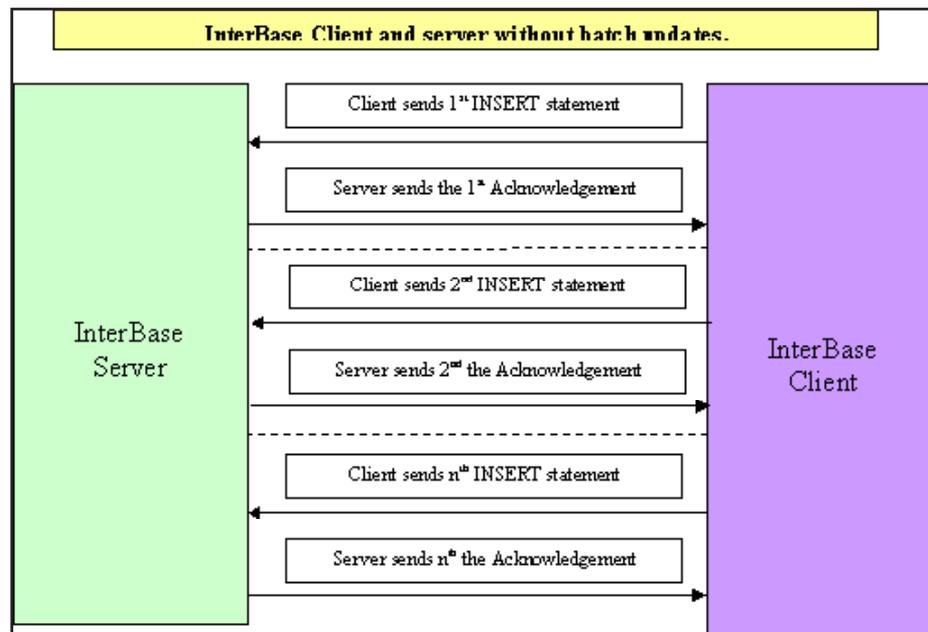
Backup Incremental

Também chamado de *on-line dump*, era uma das grandes solicitações dos usuários. Esse recurso permite que sejam feitos backups diferenciais em bancos em produção, sem a necessidade de tirá-lo do ar. Sendo assim, o backup incremental armazena somente as páginas do banco de dados que sofreram alterações desde a última vez que foi feito um backup.

Importante saber que backup incremental só pode ser feito em arquivos com a ODS (On Disk Structure) versão 12.

Journaling

É uma forma de recuperação de dados parcial ou total, devido a alguma falha de hardware ou software. Essa característica está disponível somente na versão **server edition**. Para ganhar mais desempenho, é recomendável que os arquivos do banco de dados e do *journal* sejam colocados em discos diferentes.



Batch Updates

Batch updates nos fornece uma maneira de enviar um conjunto de comandos de uma única vez para o servidor, reduzindo assim o tráfego de rede e conseqüentemente melhorando o desempenho da aplicação.

Com exceção dos comandos *Select* e *Create Database*, todos os outros comandos são suportados pelo *batch update* e em todos os casos é retornado um *array* com o número de linhas afetadas pelo comando.

A figura da página anterior representa graficamente uma comunicação com um banco de dados, sem a utilização da API *batch update* e outra com a utilização da API.

Conclusão

O InterBase é disponibilizado em dois “sabores”: **Desktop Edition** e **Server Edition**. O primeiro é indicado para o ambiente de desenvolvimento e testes, e o outro para ambientes profissionais. Nem todos os recursos estão disponíveis na versão *desktop*. Um documento comparando os recursos disponíveis nas duas versões pode ser visto em http://www.codegear.com/portals/0/documents/products/interbase/ib_feature_matrix.pdf

O artigo mostrou algumas características da nova versão do Interbase 2007, todas elas buscando fornecer um banco mais escalável, confiável, robusto e, com tudo isso, um banco simples de instalar e com baixo custo de manutenção,

Em uma próxima oportunidade, vamos nos aprofundar em cada um dos assuntos citados, mostrando as melhores formas de implementação e dicas de como e quando utilizar cada recurso.

Versões de avaliação do InterBase 2007 podem ser baixadas gratuitamente no endereço <http://www.codegear.com/Default.aspx?tabid=142>

Autor:
Mendo Leonel
Mini-curriculo
Mendo Leonel é Bacharel em Ciência da Computação e Pós-Graduado em Engenharia de Software pelo Instituto Brasileiro de Tecnologia Avançada. Com mais de 10 anos de experiência no mercado de TI, atualmente atua em consultorias especializadas em gestão de projetos e gestão de requisitos. Consultor especialista em Interbase, atuando no suporte ao produto em todo o Brasil.

Avalie esse artigo

Mantenha-se protegido contra corrupção de bancos de dados InterBase/Firebird!

IBFirstAID - recupera BDs corrompidos
IBBackupSurgeon - recupera backups problemáticos
IBAnalyst - Analisa problemas de performance
(versões de avaliação disponíveis para download)

www.firebase.com.br/fb/parceria_ibaid.html

Preços reduzidos para os brasileiros, através de uma parceria exclusiva com a FireBase!

Não deixe de conferir!

Instalação do PostgreSQL 8.0 para Win32

Cláudio Duarte de Freitas

O PostgreSQL é um sistema gerenciador de banco de dados objeto relacional (SGBDOR), desenvolvido como projeto de software livre.

Em suas versões mais recentes, possui suporte nativo para a plataforma Windows, mas apenas para as versões Windows 2000, XP e 2003. Para proceder com a instalação no Windows NT são necessários ajustes no processo de instalação, lembrando que é necessário usar o sistema de arquivos NTFS, pois além de fornecer alta performance para armazenamento, é mais seguro que a “velha” FAT.

Para baixar a versão mais recente do banco, acesse o endereço: <http://www.postgresql.org/download/>

Descompacte o arquivo, e, execute o arquivo postgresql-8.0.msi.

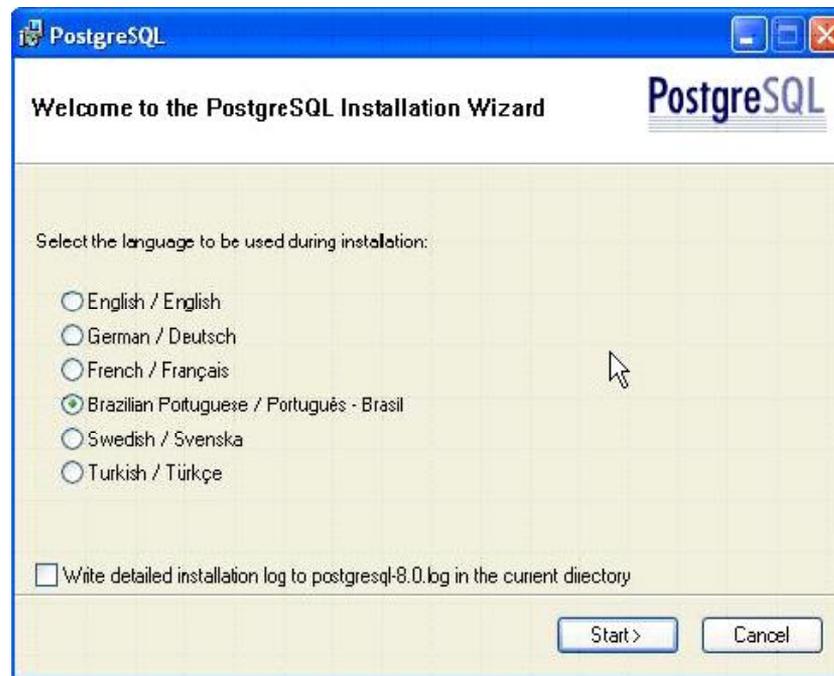
MSI – MicroSoft Installer

A instalação baseada em MSI (*Microsoft Installer*) foi chamada inicialmente de Darwin, sendo um avançado *engine* de instalação, gerenciamento e remoção de software disponível para as últimas versões do *Microsoft Windows*.

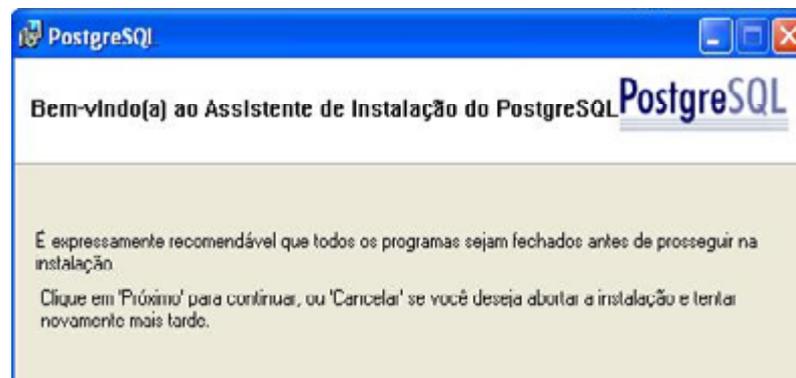
É uma ferramenta muito útil, e depende do *msiexec.exe*, sendo o arquivo .MSI apenas um pacote, com a funcionalidade similar à de um .RPM ou .deb do Linux.

Procedendo com a instalação:

Após chamar o arquivo .MSI, será exibido o formulário a seguir. Selecione o idioma e clique no botão “**Start >**”.



Em seguida, será exibida a tela de Boas Vindas. Para ir adiante basta clicar no botão “**Próximo >**”:

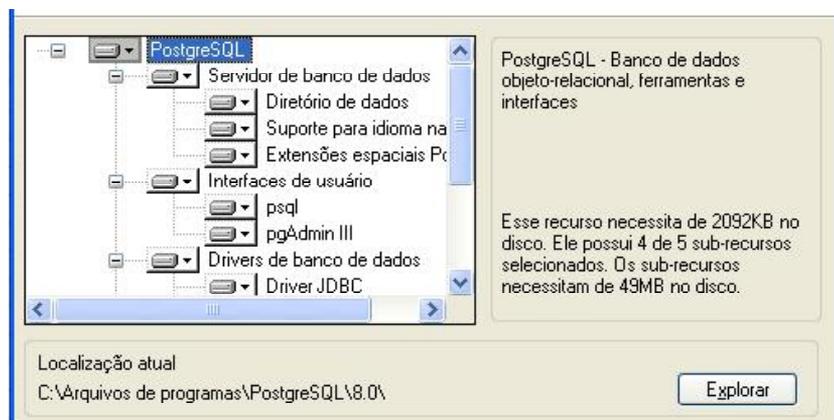


Após isso, uma tela com as informações de instalação do PostgreSQL 8 será exibida. Clique no botão **Próximo** para continuar.



O próximo formulário apresenta opções de instalação. Selecione as opções, *drivers* e *interfaces* que desejar instalar. Você poderá customizar o diretório de instalação selecionando **PostgreSQL** e clicando no botão **Explorar** para indicar um diretório de sua escolha. O mesmo poderá ser feito para qualquer opção individual, ou seja, selecione a opção e clique no botão **Explorar**.

Clique no botão **Próximo** para continuar.



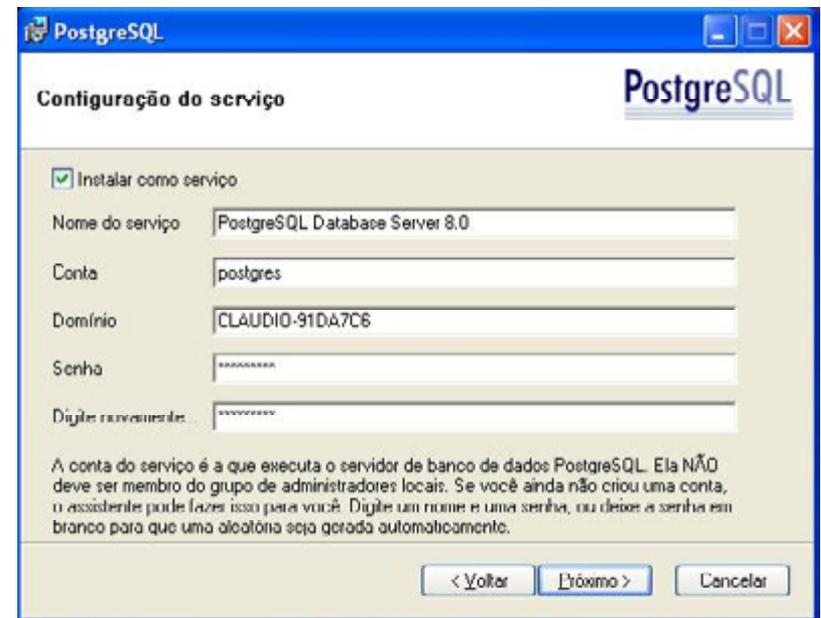
A próxima tela é importante! **Fique atento!**

Se você deseja que o PostgreSQL seja instalado como um serviço no Windows, habilite a opção "*Instalar como serviço*". Aqui vem uma das partes mais críticas da instalação, que é relacionada à configuração do usuário que vai rodar o serviço. Esse usuário não é um usuário do SGBD, mas do próprio *Microsoft Windows*, e não pode ser o *administrador* bem como deve obrigatoriamente ter uma senha definida.

Rodar como serviço

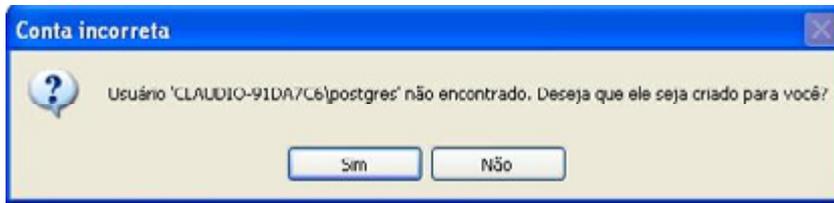
Nas versões do Windows superiores ao NT, uma aplicação pode rodar como *serviço*, ou seja, ela não aparecerá no Taskbar ou mesmo na IconTray, e será iniciada mesmo quando o *login* no windows não foi realizado. Com a aplicação rodando como serviço, pode-se determinar prioridade no uso bem como os direitos que o usuário que dará início à execução do serviço terá.

Dica: Crie você mesmo um usuário e atribua a senha para ele, pois se deixar isso a cargo do instalador do PostgreSQL, ele criará uma senha complexa e de difícil memorização (porém, mais segura). Você pode criar seu *user* e atribuir sua senha, lembrando de utilizar sempre senhas seguras e confiáveis.

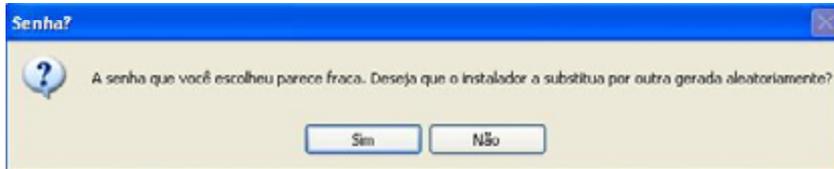


A tela abaixo mostra que se o usuário informado não existir no Windows, o instalador propõe criá-lo automaticamente.

Clique no botão **Sim** para que o mesmo possa ser criado.



O instalador verifica a segurança de sua senha, conforme demonstrado na tela abaixo. Ele alertará caso a senha informada seja considerada fraca; nesse momento pode-se autorizar o instalador a gerar uma senha aleatória.



A seguir a confirmação de que foi concedida a permissão de rodar o PostgreSQL como serviço. Clique em **OK** para continuar.



A próxima tela trará diversas informações: porta utilizada pelo servidor, codificação, senha do *super usuário* do banco, lembrando que não poderá ser a mesma senha do usuário do Windows (informada na tela de *Configuração do Serviço*), pois será um usuário interno do SGBD (o usuário poderá ter o mesmo nome).

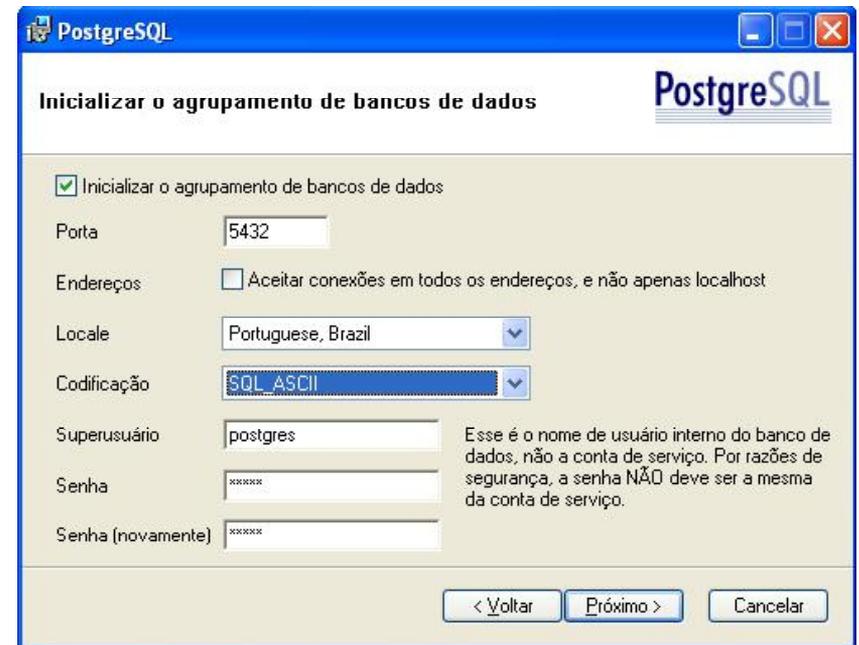
Para permitir que usuários remotos se conectem ao servidor, ative a opção **Aceitar conexões em todos os endereços**, e não apenas localhost.

localhost

É um “nome” geralmente atribuído ao IP 127.0.0.1, ou seja, um nome para se acessar a própria máquina. Cabe lembrar que em ambientes Cliente/Servidor, o *localhost* é o servidor e não a estação de trabalho do usuário.

Mesmo que a opção acima tenha sido marcada, você deverá habilitar acesso por *host*, editando o arquivo **pg_hba.conf**, para que os usuários remotos possam se conectar ao servidor.

A próxima tela somente será exibida se você escolheu instalar o PostgreSQL como um serviço. Clique em **Próximo** para continuar.



Na próxima tela, habilitam-se as linguagens procedurais. Os arquivos PL/pgsql sempre são instalados como padrão para todos os bancos de dados. A tela só é exibida quando se optou pela instalação do PostgreSQL como serviço, e tenha-se escolhido ter um agrupamento de dados inicializado.

Clique no botão **Próximo** para continuar.

SQL> SELECT * FROM HOSPEDAGEM WHERE QUALIDADE="INSUPERAVEL";

WWW.BAVS.COM.BR

MySQL Firebird

Cada vez mais empresas desenvolvem sistemas multi-usuários que necessitam que seus dados sejam disponibilizados na Internet, em tempo real, para atender às necessidades de seus clientes. Oferecemos as melhores soluções em hospedagem de bancos de dados MySQL e Firebird (1.0 e 1.5), com acesso direto e sem restrições de conexões.

PLANOS DE HOSPEDAGEM		
PRO I	PRO III	SEMI D. II
<p>Instalação Gratuita</p> <ul style="list-style-type: none"> 100 Mb espaço em disco Firebird 1.5 MySQL 3.23 PHP 4 Perl 5 CGI Domínio SSL Gratuito Configuração Semáfora Manutenção: R\$ 29,90 	<p>Instalação Gratuita</p> <ul style="list-style-type: none"> 200 Mb espaço em disco Firebird 1.5 MySQL 4 PHP 4 / Perl 5 / CGI JSP (Tomcat) Servlet ASP .NET (Mono/Clr) Configuração Semáfora Manutenção: R\$ 59,90 	<p>Instalação Gratuita</p> <ul style="list-style-type: none"> 1 Gb espaço em disco Firebird 1.5 MySQL 4 PHP 4 / Perl 5 / CGI JSP (Tomcat) Servlet ASP .NET (Mono/Clr) Configuração Semáfora Manutenção: R\$ 155,90

acesse: WWW.BAVS.COM.BR + PLANOS + INFORMAÇÕES

clientes.com.satisfacao
 Email: info@bavs.com.br
 Atendimento Eletrônico 24hrs: (11) 2421-0221
 Vendas On-Line (ambiente seguro): http://www.bavs.com.br

Database Workbench
cross database development tool

Principais recursos do Database Workbench:

- Uma única IDE para todos os SGBDs suportados
- Várias ferramentas de produtividade
- Ferramentas para browse de metadada e query
- SQL Insight com suporte a JOINS
- Ferramentas para transferência de Metadata
- Ferramentas de teste e medição de performance
- Ferramentas de gerenciamento e relatórios
- E muito mais...

Bancos de dados suportados:

- Borland® InterBase® 4 - 7
- Microsoft® SQL Server™ 7 - 2005
- Oracle® 8i - 10g
- MySQL® 4 - 5
- Firebird™ 1 - 2

Pague menos registrando através da FireBase (www.firebase.com.br)

Upscene Productions Database Tools for Developers
 Upscene Productions - http://www.upscene.com - info@upscene.com

Precisando de cursos ou treinamentos de Firebird?

A FireBase oferece cursos/treinamentos de Firebird ministrados dentro da sua empresa. Os cursos são ministrados por **Carlos H. Cantu**, um dos maiores evangelistas do Firebird no Brasil, autor do livro Firebird Essencial. Mais informações pelo email cursos@firebase.com.br

Firebird Essencial

Primeiro livro brasileiro que trata especificamente dos recursos do SGBD Firebird (versões 1.0 e 1.5). O autor reuniu no livro todo o material produzido por ele para as revistas ClubeDelphi e SQLMagazine. Os artigos foram revisados, atualizados e muitos deles complementados, de forma a proporcionar ao leitor uma fonte de informação rica, atualizada e confiável. Um capítulo inédito sobre a criação de UDFs foi escrito exclusivamente para o livro.

Você aprenderá a instalar o SGBD, criar procedures, catálogos em CDROM, criar backups, gerenciar usuários, utilizar campos BLOB de forma adequada, identificar os tipos de dados disponíveis no Firebird, e muito mais!

Verifique o sumário do livro em www.firebase.com.br/fb/livro/fbessencial

Compre autografado em www.firebase.com.br

Firebird - Dicas de Segurança

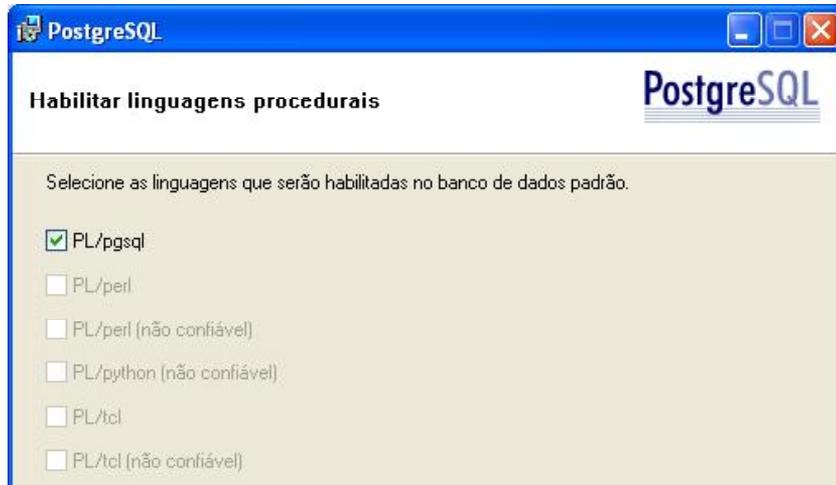
Luiz Paulo de Oliveira Santos

Primeiro livro sobre segurança para o Firebird. Indicado para desenvolvedores experientes que podem utilizá-lo com o check-list para a segurança de seus bancos, bem como para usuários iniciantes que ainda não implementaram nenhuma política de segurança.

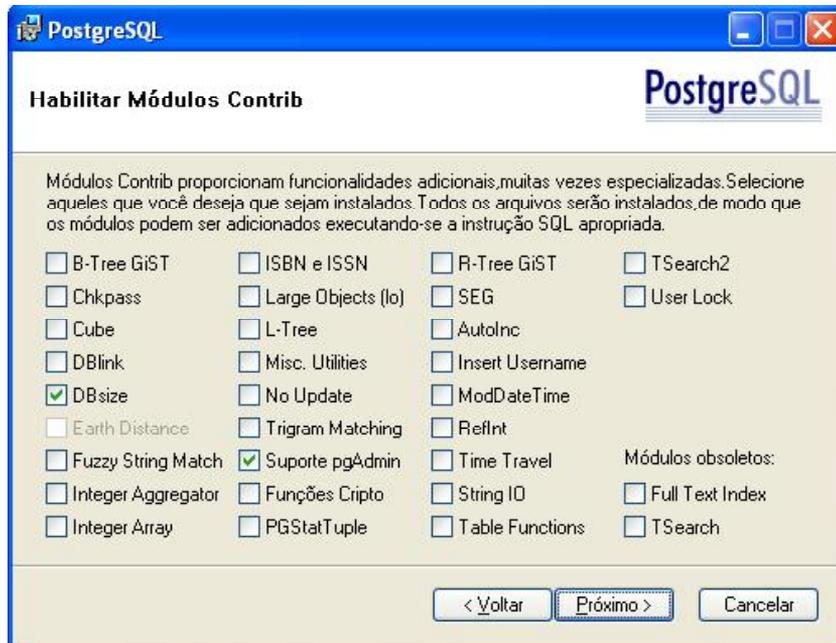
Em 159 páginas você encontrará dicas de como implementar Firewall para Firebird, dicas de como implementar tunel seguro e criptografado para Firebird, BACK UP, sniffers e o Firebird, uso do GSEC dicas sobre os sistemas de arquivos e muito mais.

O autor é editor da revista DB Freemagazine, trabalha na área de informática desde 1984, utiliza Interbase desde 1996, foi palestrante nas três edições do Firebird Developers Day e também na Conferência Internacional de Firebird na República Tcheca em 2005.

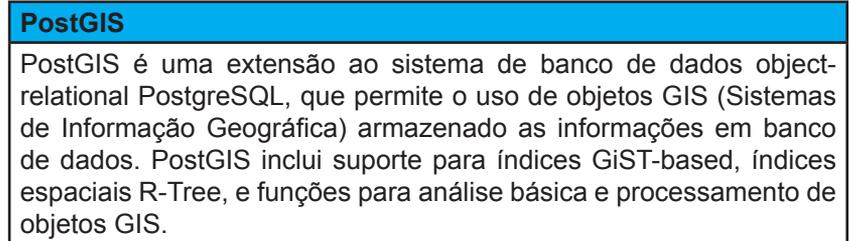
Compre on-line em www.firebase.com.br



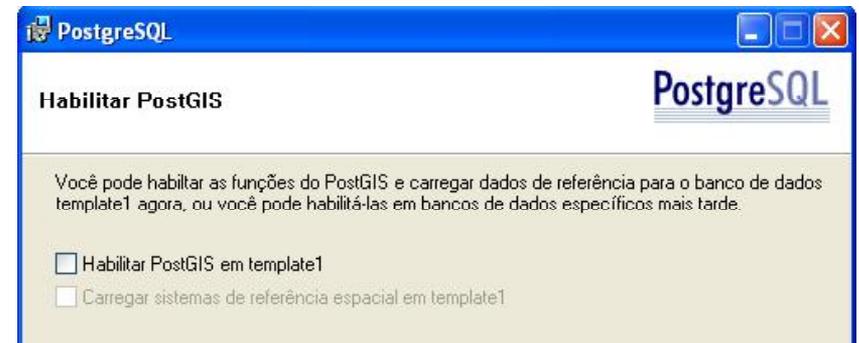
Selecione os módulos que deseja ativar no *template1*. Todos os módulos são instalados, e esse passo apenas ativará os serviços selecionados. No exemplo abaixo, está ativado apenas o DBsize e o Suporte ao *pgAdmin* (que já são ativados como padrão).



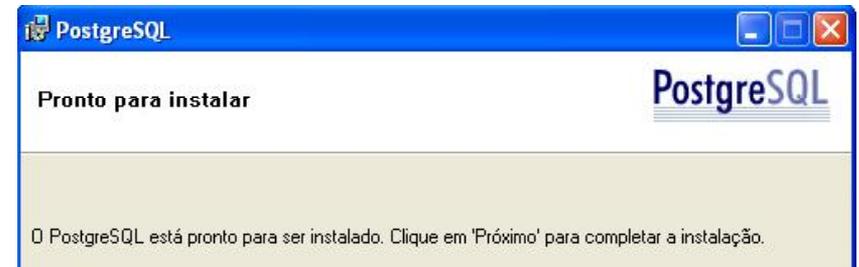
Durante o processo de instalação, é solicitado habilitar as funções do PostGIS. No exemplo, optei por não ativá-lo.



Clique no botão **Próximo** para continuar.

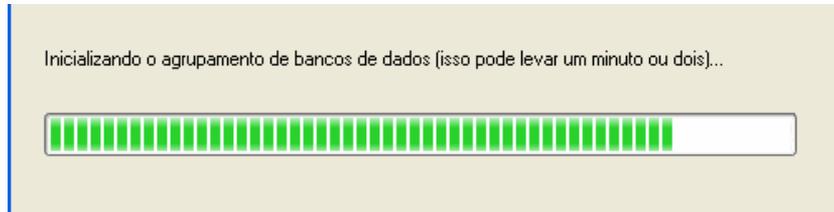


Clique no botão **Próximo** para continuar.



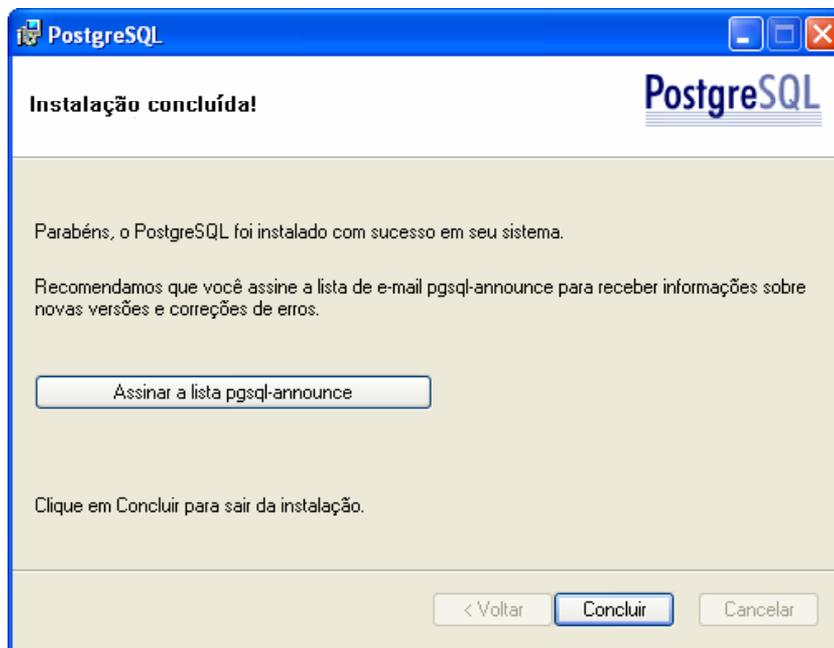
Poderá surgir uma tela preta (*prompt* do DOS) durante o processo de instalação. Não a feche! pois ela será encerrada automaticamente depois de aproximadamente 2 minutos. Este evento ocorre em equipamentos com uma versão do Windows XP sem o Service Pack 2.

Será exibida a barra de progresso de instalação, conforme figura abaixo:



Após o término da instalação, você poderá se inscrever no *pgsql-announce*.

Clique em **Concluir** para terminar a instalação.



A instalação encerra-se aqui!

Criando e configurando as variáveis do Sistema Operacional

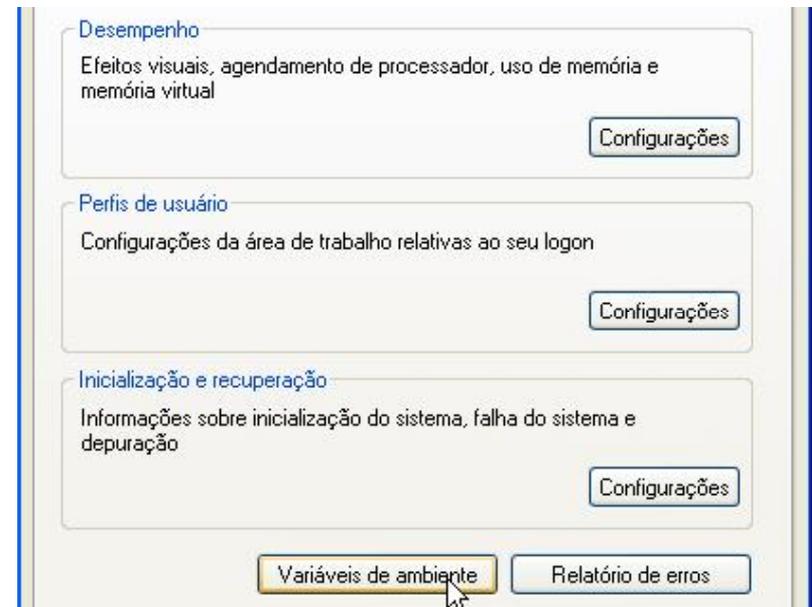
Antes de iniciar o uso do PostgreSQL, devemos primeiramente criar as variáveis de sistema. Mostrarei a seguir, como criar as variáveis para sistemas

operacionais NT, 2000 e XP.

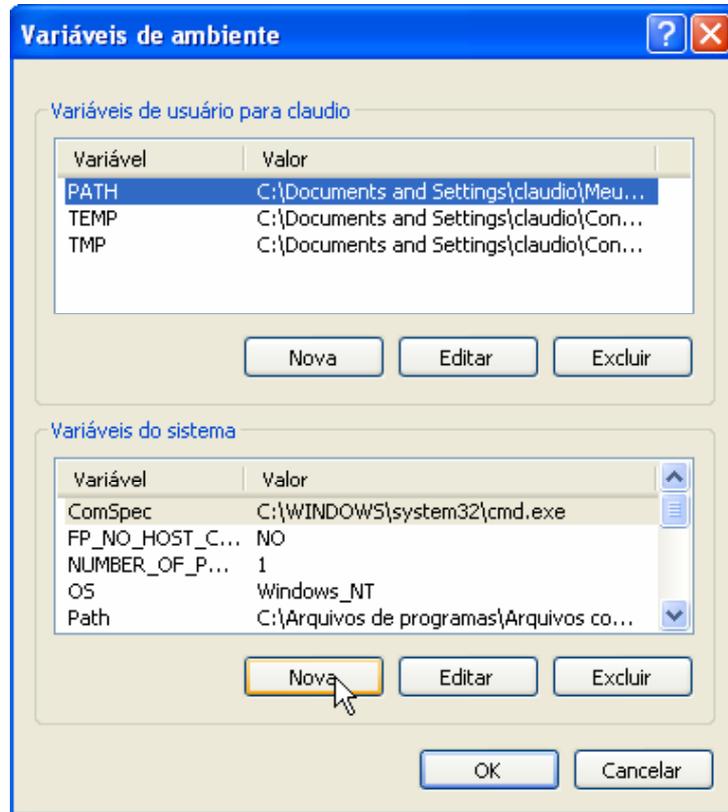
Clique com o botão do mouse no ícone **Meu computador** em sua área de trabalho e escolha a opção **Propriedades** (conforme a figura a seguir):



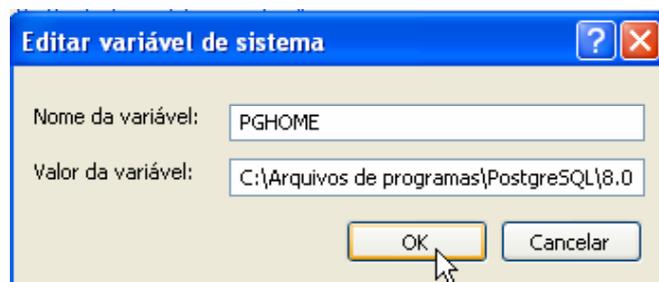
Escolha a aba **Avançado** e clique no botão **Variáveis de ambiente**.



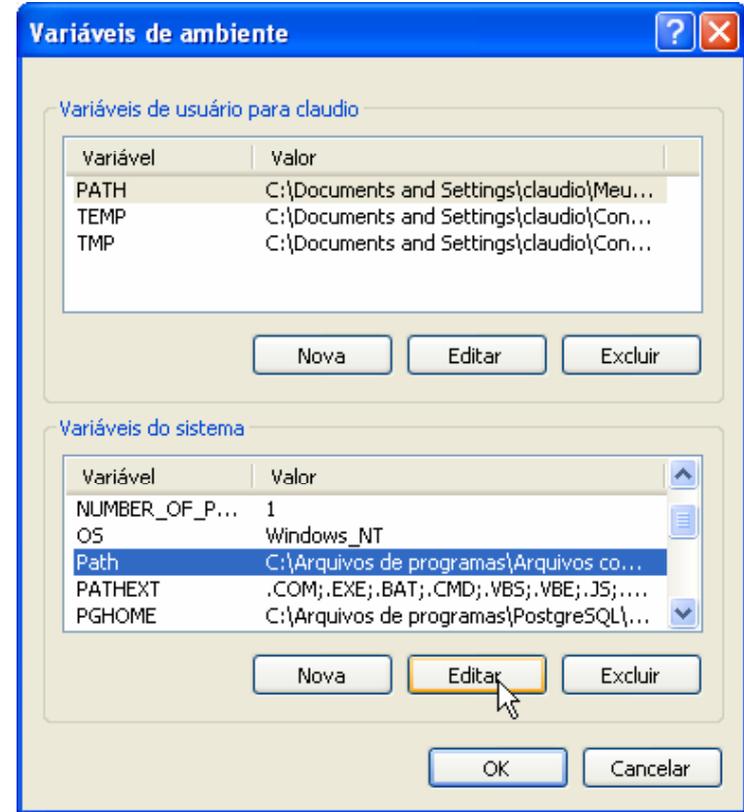
Clique no botão **Nova** em **Variáveis do sistema**.



No campo **Nome da variável**, atribua à variável o nome que preferir, e no campo **Valor da variável** preencha com o caminho do diretório do PostgreSQL onde se encontra a pasta bin (não inclua *bin* no caminho), em seguida clique em OK.



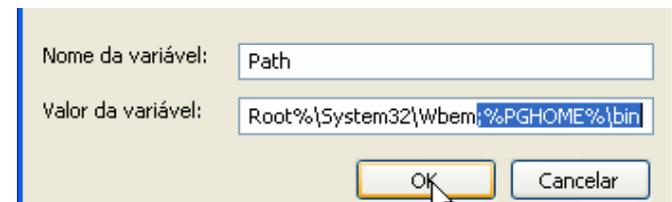
Selecione a variável **Path** em **Variáveis do sistema** e clique no botão **Editar**.



No final da linha no campo **Valor da variável** acrescente o seguinte valor:

;%PGHOME%bin

(note que existe um **ponto e vírgula** antes de %PGHOME%\bin) e clique no botão **OK** e reinicialize o computador para continuar.



Criando um Banco de Dados

Para facilitar o uso do PostgreSQL, sugiro os seguintes ícones na área de trabalho:

- Prompt do DOS;
- Inicializador de serviço do PostgreSQL;
- Finalizador de serviço do PostgreSQL;
- pgAdmin III.

Para criar um banco de dados, deve-se primeiramente iniciar o serviço do PostgreSQL clicando no ícone na área de trabalho **Iniciar serviço**, ou acessando **Iniciar\Programas\PostgreSQL 8.0\Iniciar serviço**.

Irá aparecer rapidamente o prompt do DOS, e logo depois fechará automaticamente.

Após iniciar o serviço, iremos acessar o **prompt de comando** (DOS) clicando no ícone na área de trabalho ou acessando **Iniciar\Programas\Acessórios\Prompt de comando**.

Podemos criar um banco de dados (no exemplo com o nome **primeiro_bd**) de duas formas diferentes:

- Se você estiver usando o mesmo usuário do Windows informado na instalação, poderá digitar apenas:

```
createdb primeiro_bd
```

Será solicitada a senha, após digitá-la a resposta exibida deverá ser a seguinte:

CREATE DATABASE

- Porém, se estiver usando outro usuário do sistema operacional poderá também criar o banco de dados digitando o seguinte comando:

```
createdb -U postgres primeiro_bd
```

A senha será solicitada, e deverá ter a mesma resposta conforme acima.

Veja o que ocorre caso você digite o comando **createdb primeiro_bd**, usando outro usuário que não seja o do

PostgreSQL.

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

C:\>createdb primeiro_bd
Password:
createdb: could not connect to database template1: FATAL: password authentication failed for user "claudio"

C:\>_
```

No comando **createdb -U postgres primeiro_bd**, você poderá estar logado com qualquer usuário, pois **-U postgres** é que identifica o usuário criador do banco de dados.



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

C:\>createdb -U postgres primeiro_bd
Password:
CREATE DATABASE

C:\>
```

Se você digitar o comando **createdb -U postgres**, o nome do banco de dados será o do usuário que está logado.

Importante: A seguir será citada a exclusão de um banco, após excluir um banco não há como restaurá-lo, exceto restaurando um *back up*. Cuidado com os próximos procedimentos.

Para deletar o banco de dados você poderá fazê-lo digitando:

```
dropdb -U postgres primeiro_bd
```

Solicitará novamente a senha e a resposta será:

DROP DATABASE

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

C:\>dropdb -U postgres primeiro_bd
Password:
DROP DATABASE
```

Lembrando que essa ação remove definitivamente todos os arquivos que estiverem associados ao banco de dados, não podendo ser desfeita.

Continuando...

Iremos criar novamente o banco de dados *primeiro_bd* para podermos acessá-lo.

Vamos executar um programa chamado *psql*, para podermos editar, entrar e executar comandos de SQL.

psql -U postgres primeiro_bd

Será solicitada novamente a senha, veja na figura abaixo:

```

C:\>psql -U postgres primeiro_bd
Password:
Welcome to psql 8.0.3, the PostgreSQL interactive terminal.
Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit
Warning: Console code page (850) differs from Windows code page (1252)
        8-bit characters may not work correctly. See psql reference
        page "Notes for Windows users" for details.
primeiro_bd=# _
  
```

Podemos observar na figura acima que existem comandos internos que são mostrados na tela inicial para realizarmos ações, como por exemplo:

- Para obter ajuda sobre comandos de SQL devemos digitar depois de **primeiro_bd=#** o comando **\h**.
- Para obter ajuda de comandos do psql devemos digitar **\?**.
- Para sair do psql devemos digitar **\q**.

Vamos usar uma instrução SQL:

```
select date_current, user;
```

Veja na figura abaixo a resposta:

```

C:\>psql -U postgres primeiro_bd
primeiro_bd=# select current_date, user;
 date | current_user
-----+-----
 2005-07-18 | postgres
<1 row>
primeiro_bd=# _
  
```

Criação de tabelas

Para criar uma tabela devemos digitar o nome da coluna e logo depois seu tipo, por exemplo:

```

CREATE TABLE  FUNCIONARIO (
ID_FUNCIONARIO      int,
SEXO                 char(1),
NOME                 varchar(50),  -- nome do funcionário
ENDERECO            varchar(60),
FONE                 varchar(9),
ADMISSAO            date
);
  
```

Note que você poderá teclar *Enter* para ir para outra linha, pois o *psql* somente irá terminar o comando após encontrar o ponto e vírgula.

Dicas: Note que o ultimo campo criado não possui a vírgula no final. Usam-se dois hífen (sinais de menos) para comentários, ou seja, tudo que vier depois deles será ignorado pelo psql.

Veja abaixo a tela de *prompt*.

```

primeiro_bd=# CREATE TABLE FUNCIONARIO(
primeiro_bd<# ID_FUNCIONARIO      int,
primeiro_bd<# SEXO                 char(1),
primeiro_bd<# NOME                 varchar(50), -- Nome do funcionário
primeiro_bd<# ENDERECO            varchar(60),
primeiro_bd<# FONE                 varchar(9),
primeiro_bd<# ADMISSAO            date
primeiro_bd<# );
CREATE TABLE
primeiro_bd=# _
  
```

Inserindo dados em tabelas

Com a instrução INSERT podemos inserir dados na tabela, conforme abaixo:

```
INSERT INTO FUNCIONARIO VALUES (100, 'M', 'JOSE DA SILVA', 'RUA DAS ROSAS, 505', '3210-5520', '22/10/2001');
```

A sintaxe usada acima, realiza a inserção dos dados em ordem conforme suas colunas. Os dados que não são numéricos deverão estar entre apóstrofes (') e os tipos numéricos não precisam.

Podemos inserir configurações de datas tais como 2001-22-10 (separados por hífens) ou 10/22/2001 (separados por barras).

```
INSERT INTO FUNCIONARIO (ID_FUNCIONARIO, SEXO, NOME, ENDERECO, FONE, ADMISSAO) VALUES (100, 'M', 'JOSE DA SILVA', 'RUA DAS ROSAS, 505', '3210-5520', '10/22/2001');
```

```
primeiro_bd=# INSERT INTO FUNCIONARIO (ID_FUNCIONARIO, SEXO, NOME,
primeiro_bd=# ENDERECO, FONE, ADMISSAO)
primeiro_bd=# VALUES (100, 'M', 'JOSE DA SILVA', 'RUA DAS ROSAS, 505',
primeiro_bd=# '3210-5520', '10/22/2001');
INSERT 1/2000 1
primeiro_bd=# _
```

A sintaxe acima permite relacionar as colunas com as entradas de dados.

Você poderá também entrar com dados não necessariamente em ordem, como no exemplo abaixo:

```
INSERT INTO FUNCIONARIO (SEXO, NOME, FONE, ADMISSAO) VALUES ('M', 'JOSE DA SILVA', '3210-5520', '10/22/2001');
```

Realizando consultas nas tabelas

A instrução SELECT do SQL é responsável por efetuar consultas e retornar os dados obtidos da(s) tabela(s).

Exemplo: Para extrair todos os dados da tabela FUNCIONARIO, deve-se digitar:

```
SELECT * FROM FUNCIONARIO;
```

Na sintaxe acima, o * significa que você esta solicitando ver todos os campos da tabela.

```
primeiro_bd=# SELECT * FROM FUNCIONARIO;
 id_funcionario | sexo | nome                | endereco                | fone      | admissao
-----
 100 | M   | JOSE DA SILVA      | RUA DAS ROSAS, 505     | 3210-5520 | 2001-10-22
 105 | F   | MARIA HELENA      | RUA DA PAZ, 200       | 3200-0254 | 2002-09-05
 107 | F   | RONILDA MORAIS    | RUA JATOBÁ, 1520     | 3211-1205 | 2005-10-15
(3 rows)
```

Podemos empregar o "coringa", associando o % ao LIKE, conforme abaixo:

```
SELECT * FROM FUNCIONARIO
WHERE SEXO = 'F'
AND NOME LIKE 'RONIL%';
```

```
primeiro_bd=# SELECT * FROM FUNCIONARIO
primeiro_bd=# WHERE SEXO = 'F'
primeiro_bd=# AND NOME LIKE 'RONIL%';
 id_funcionario | sexo | nome                | endereco                | fone      | admissao
-----
 107 | F   | RONILDA MORAIS    | RUA JATOBÁ, 1520     | 3211-1205 | 2005-10-15
(1 row)
```

Segue outro exemplo de sintaxe para consulta:

Novamente empregando o coringa, para extrair apenas os nomes e telefones que iniciam com 321.

```
ca Prompt de comando - psql -U postgres primeiro_bd
primeiro_bd=# SELECT NOME, FONE FROM FUNCIONARIO
primeiro_bd=# WHERE FONE LIKE '321%';
      nome      | fone
-----
 JOSE DA SILVA | 3210-5520
 RONILDA MORAIS | 3211-1205
(2 rows)
primeiro_bd=# _
```

Para atualizar, modificar dados na(s) tabela(s), usamos a instrução UPDATE. Abaixo está a sintaxe empregada para mudarmos o telefone da pessoa RONILDA MORAIS de 3211-1205 para 3211-1250.

```
UPDATE FUNCIONARIO
SET FONE = '3211-1250'
WHERE NOME = 'RONILDA MORAIS';
```

```
primeiro_bd=# UPDATE FUNCIONARIO
primeiro_bd=# SET FONE = '3211-1250'
primeiro_bd=# WHERE NOME = 'RONILDA MORAIS';
UPDATE 1
primeiro_bd=#
```

Conforme tela abaixo, podemos confirmar que a mudança ocorreu com sucesso.

```
primeiro_bd=# SELECT NOME, FONE FROM FUNCIONARIO
primeiro_bd=# WHERE NOME LIKE 'RONIL%';
   nome      | fone
-----+-----
RONILDA MORAIS | 3211-1250
(1 row)
```

Excluindo Dados

O comando DELETE é o responsável por excluir registros das tabelas;

```
DELETE FROM FUNCIONARIO WHERE NOME LIKE 'JOSE D%';
```

A sintaxe anterior exclui os registros que contêm dados de pessoas cujo nome começa com JOSE D, conforme podemos verificar na consulta abaixo:

```
primeiro_bd=# SELECT * FROM FUNCIONARIO;
 id_funcionario | sexo | nome           | endereco           | fone      | data_nascimento
-----+-----+-----+-----+-----+-----
105 | F | MARIA HELENA | RUA DA PAZ, 200 | 3200-0254 | 2002-09-05
107 | F | RONILDA MORAIS | RUA JATOBÁ, 1520 | 3211-1250 | 2005-10-15
(2 rows)
```

Alerta: A sintaxe abaixo exclui todos os registros da tabela. O sistema não pede confirmação para a ação.

```
DELETE FROM nome_da_tabela;
```

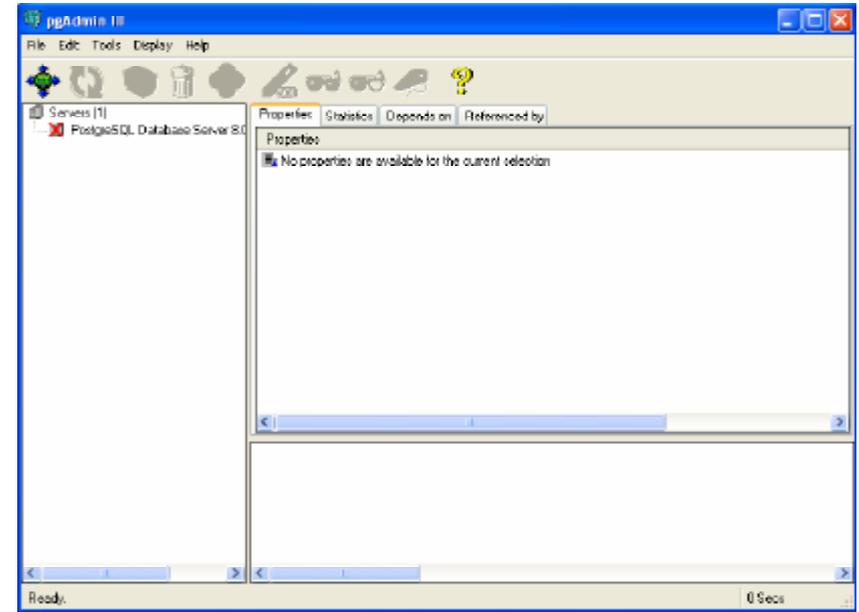
Ferramentas de administração

Existem diversas ferramentas de administração para PostgreSQL disponíveis no mercado, porém para esse *tutorial* iremos trabalhar com o **PgAdmin III** que acompanha o PostgreSQL.

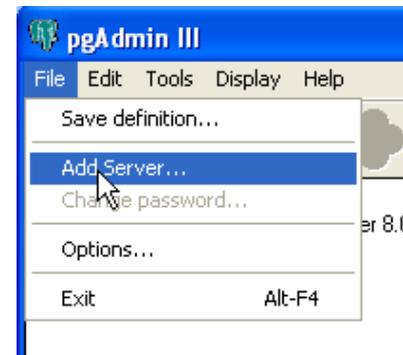
O **PgAdmin III** é um gerenciador e administrador de banco de dados gráfico, ou seja, todas as instruções SQL mostradas anteriormente poderão ser executadas de forma mais fácil com uma interface amigável para a maioria dos usuários (principalmente para os iniciantes). Isso não exige o administrador do banco de dados de saber utilizar as linhas de comandos, pois nem sempre o gerenciador gráfico estará instalado nas máquinas e, mesmo para consultas nos gerenciadores gráficos, os comandos de SQL deverão ser utilizados.

Para trabalhar com os gerenciadores gráficos deveremos também iniciar o serviço como mostrado inicialmente no tutorial.

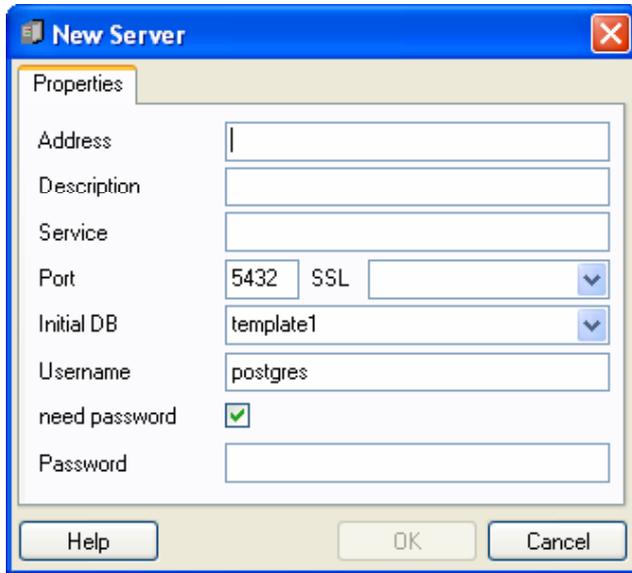
Para iniciar o PgAdmin III clique em **Iniciar\Programas\PostgreSQL 8.0\PgAdmin III**. Abaixo figura da tela inicial.



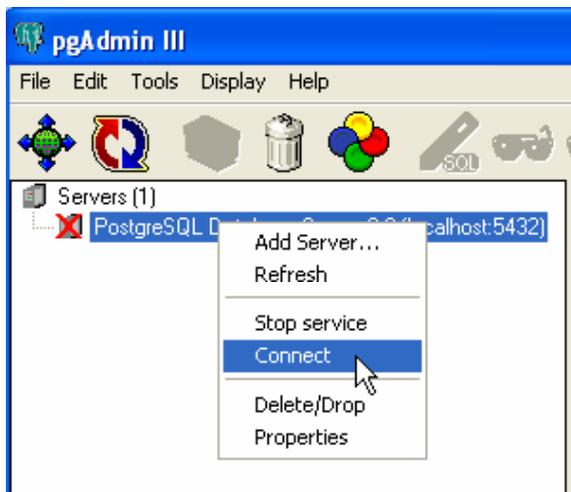
Caso haja a necessidade de adicionar um novo servidor, selecione o menu **File\Add Server**, conforme figura abaixo.



Aparecerá a tela para que possa inserir os dados do servidor.



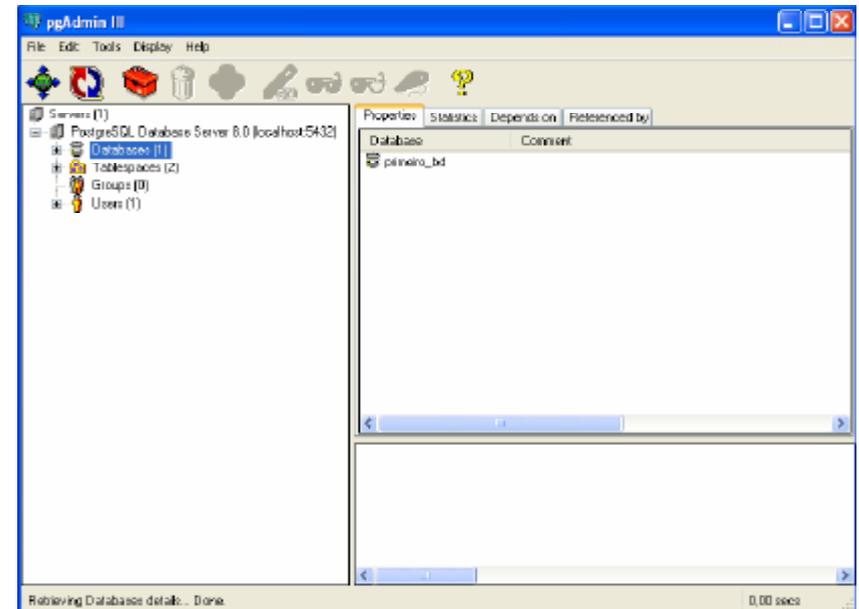
Como trabalharemos com o servidor local instalado na máquina, selecione o item do lado esquerdo e clique com o botão direito do mouse, conforme tela abaixo.



Informe a senha do super usuário, e clique em OK.



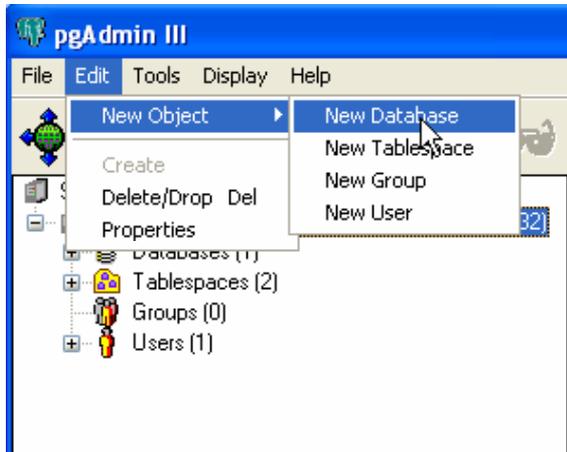
Após a autenticação, o seguinte formulário será exibido, e o acesso ao servidor estará disponível.



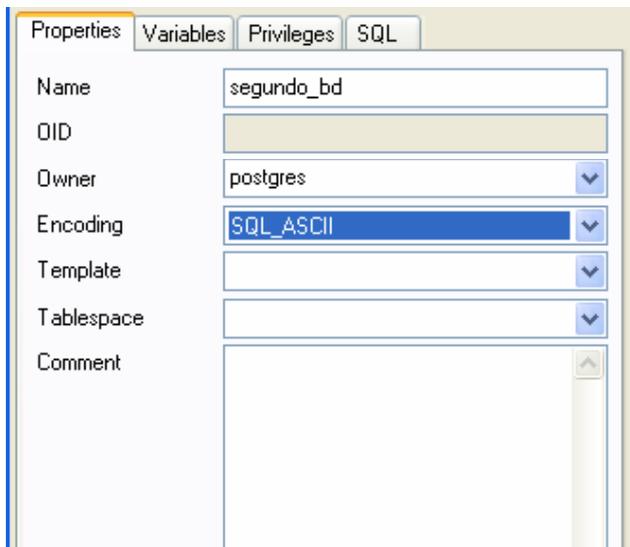
A seguir, realizaremos graficamente (ou se preferir visualmente) todos os passos mostrados anteriormente em linhas de comando, desde a criação até consultas.

Criando um Banco de Dados

Clique no menu **Edit\New Object\New Database**, conforme figura abaixo.

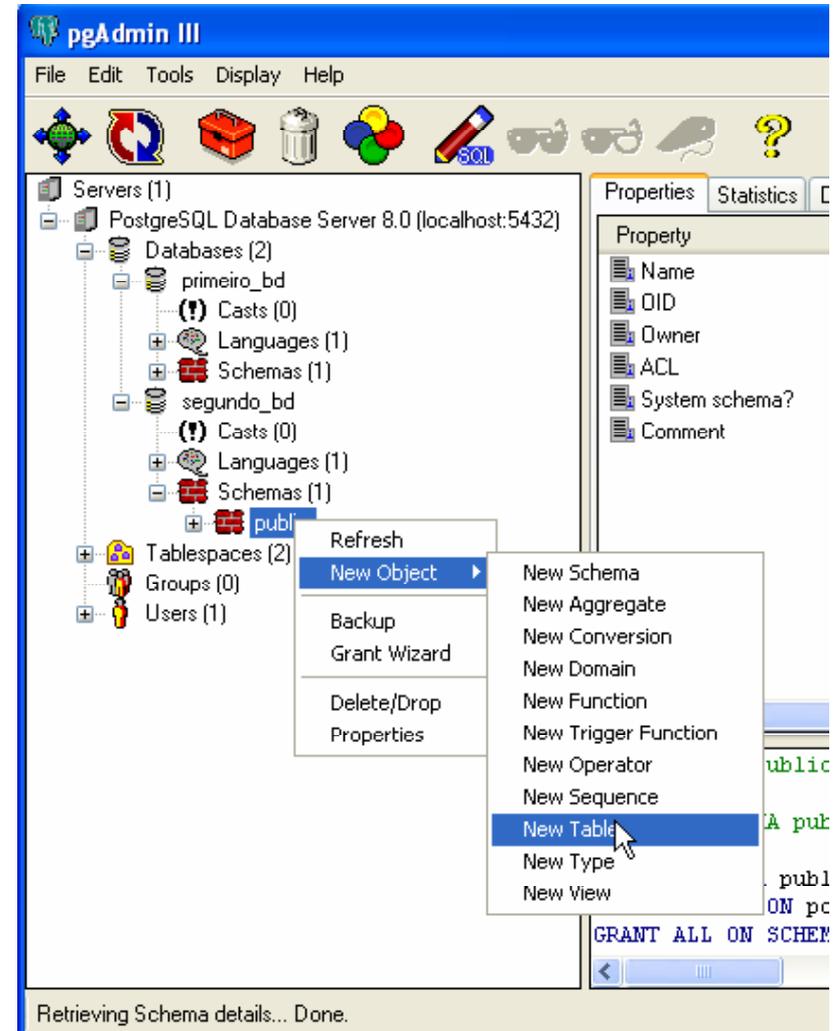


Na tela que será aberta, devemos entrar com o nome do novo banco de dados. No exemplo, preenchi somente os campos necessários, porém você poderá preencher mais campos se desejar, inclusive nas demais abas (Variables, Privileges, SQL).

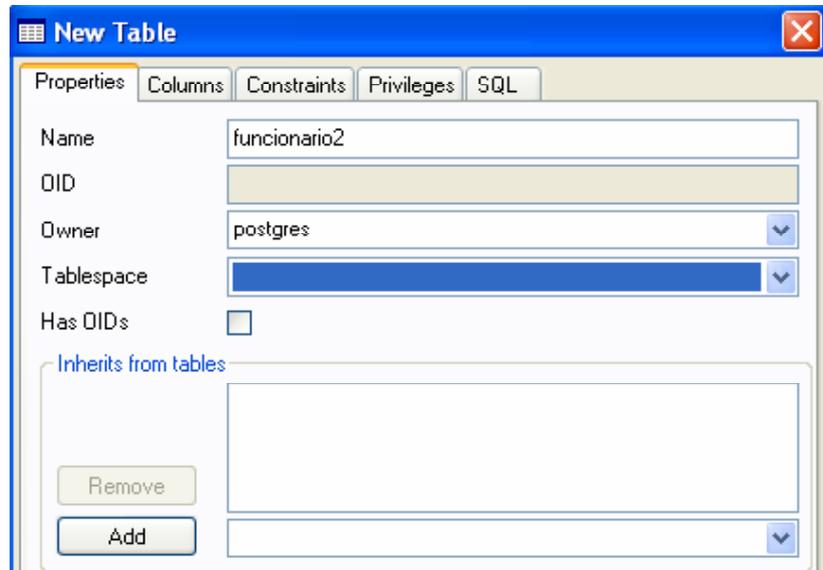


Criação de tabelas

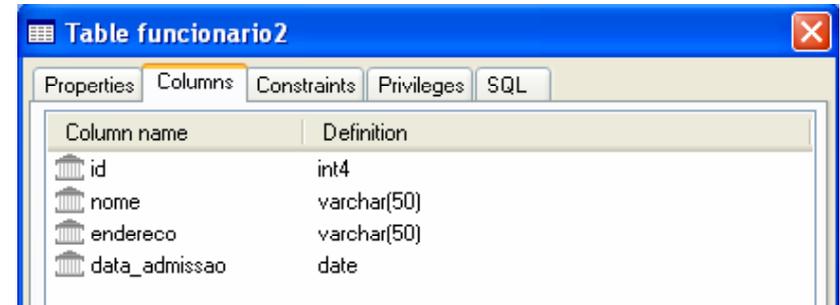
Para criar uma nova tabela, acesse no *TreeView* o item **Schemas\Public**, clique com o botão direito e escolha **New Object\New Table** (conforme figura abaixo).



A tela mostrada abaixo aparecerá, permitindo que se possa digitar os dados da tabela.

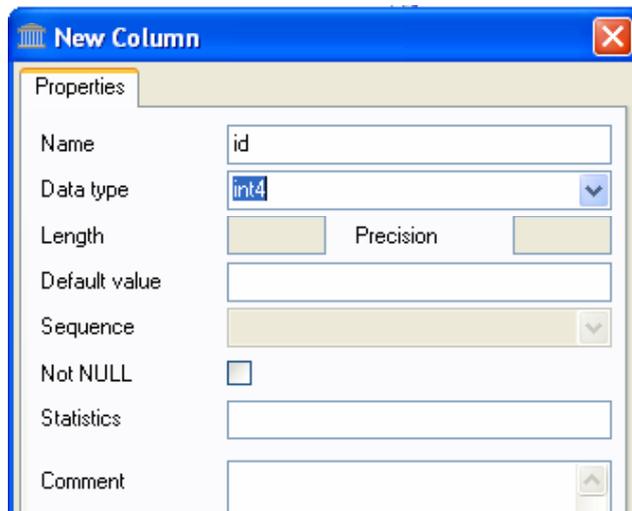
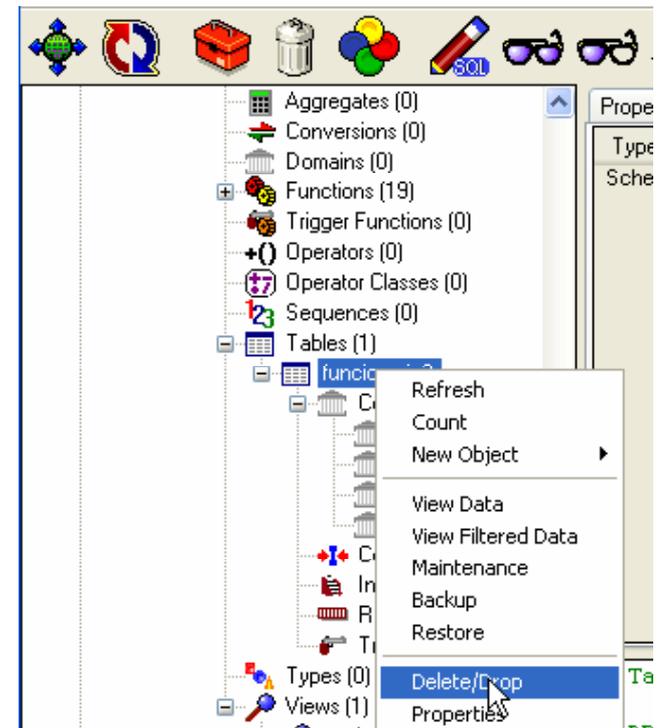


Para adicionar os campos, clique na aba **Columns** e no botão **Add**, usando a tela que será aberta para preenchimento de cada campo e clique no botão OK



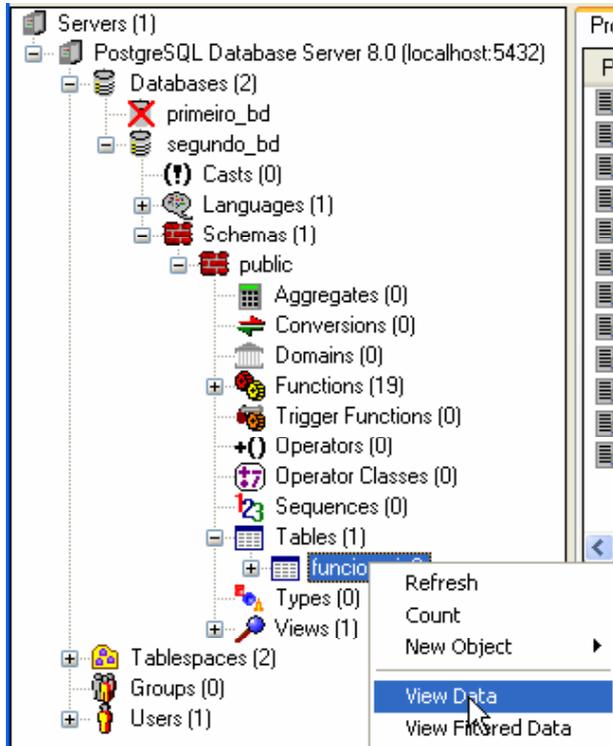
Exclusão de tabelas e campos

Para excluir tanto tabelas e campos, devemos apenas clicar com o botão direito do mouse sobre o item que quisermos (em alguma tabela abaixo de **Tables** ou em um dos campos abaixo de **Columns**) e selecionar **Delete/Drop**, conforme mostrado na figura abaixo:



Veja abaixo a lista dos campos criados.

Podemos visualizar o cadastro referente à tabela *funcionario2*, clicando com o botão direito sobre a tabela citada acima, e em **View Data**, conforme figura abaixo:



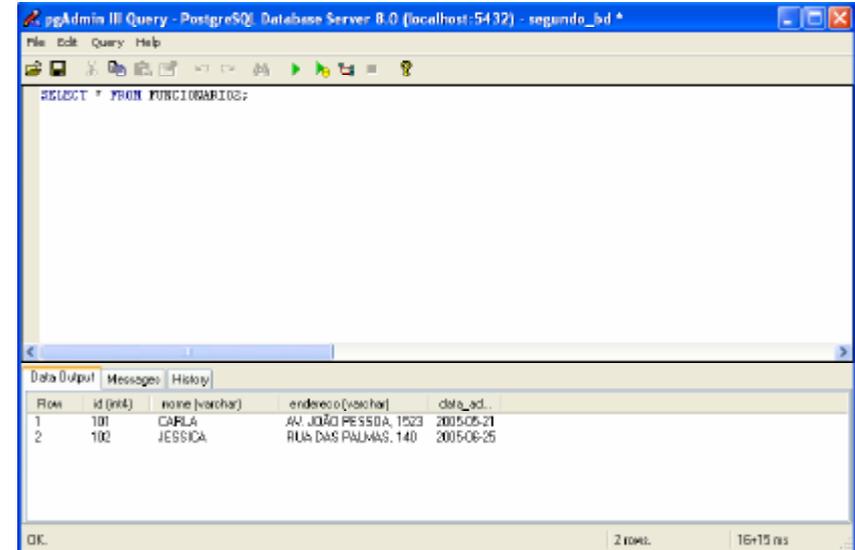
Será apresentada uma tela com os dados, conforme figura abaixo:

	id int4	nome varchar	endereco varchar	data_admissao date
1	101	CARLA	AV. JOÃO PESSOA, 1523	2005-05-21
2	102	JESSICA	RUA DAS PALMAS, 140	2005-06-25

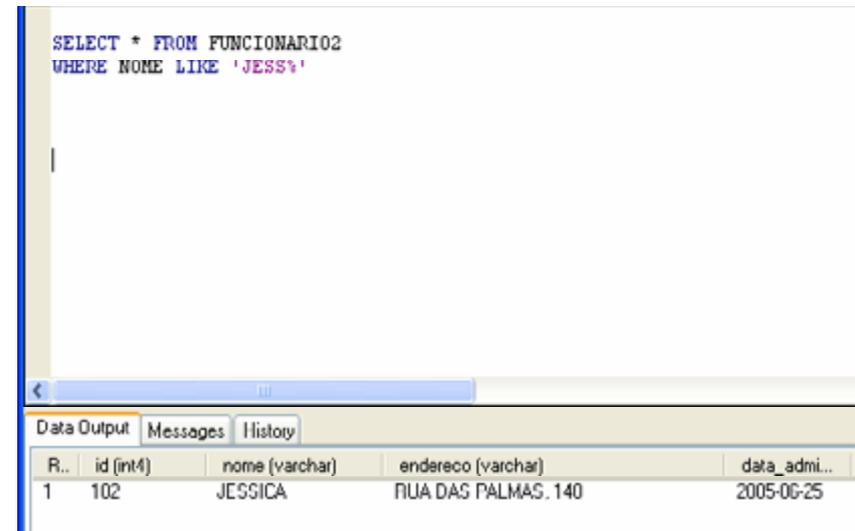
Realizando consulta em tabelas

As sintaxes são as mesmas demonstradas anteriormente, apenas as ex-

cutaremos em um modo visual.



O comando abaixo listará todos os campos da tabela funcionários, onde o nome comece com JESS.



Mudando os dados das tabelas com o comando UPDATE:

```
UPDATE FUNCIONARIO2
SET NOME = 'JÉSSICA DA SILVA'
WHERE ID = 102
```

A seguir, o resultado:

R..	id (int4)	nome (varchar)	endereco (varchar)	data_admi...
1	101	CARLA	AV. JOÃO PESSOA, 1523	2005-05-21
2	102	JÉSSICA DA SILVA	RUA DAS PALMAS, 140	2005-06-25

OK. 2 rows. 16+15 ms

Existem outras ferramentas de administração, inclusive *freewares* e que podem encontradas na *internet*, abaixo segue endereços de mais duas ferramentas:

EMS PostgreSQL Manager Lite Edition for Windows – Faça o download do arquivo EMS PostgreSQL Manager Lite for Windows (full installation package) de 10,24Mb - <http://sqlmanager.net/en/products/postgresql/manager/download>.

PG Explorer - Faça o download da Versão 2.4 Setup (Including Help) de 3Mb - <http://www.pgexplorer.com/>

Conclusão

O tutorial demonstra que o PostgreSQL é um SGBD com funcionalidades que atendem a necessidade de praticamente todos os criadores de sistemas.

Autor:

[Cláudio Duarte de Freitas](#)

Mini-curriculo

Aluno do 5º período do curso de Sistemas de Informação da Unitri-Centro Universitário do Triângulo.

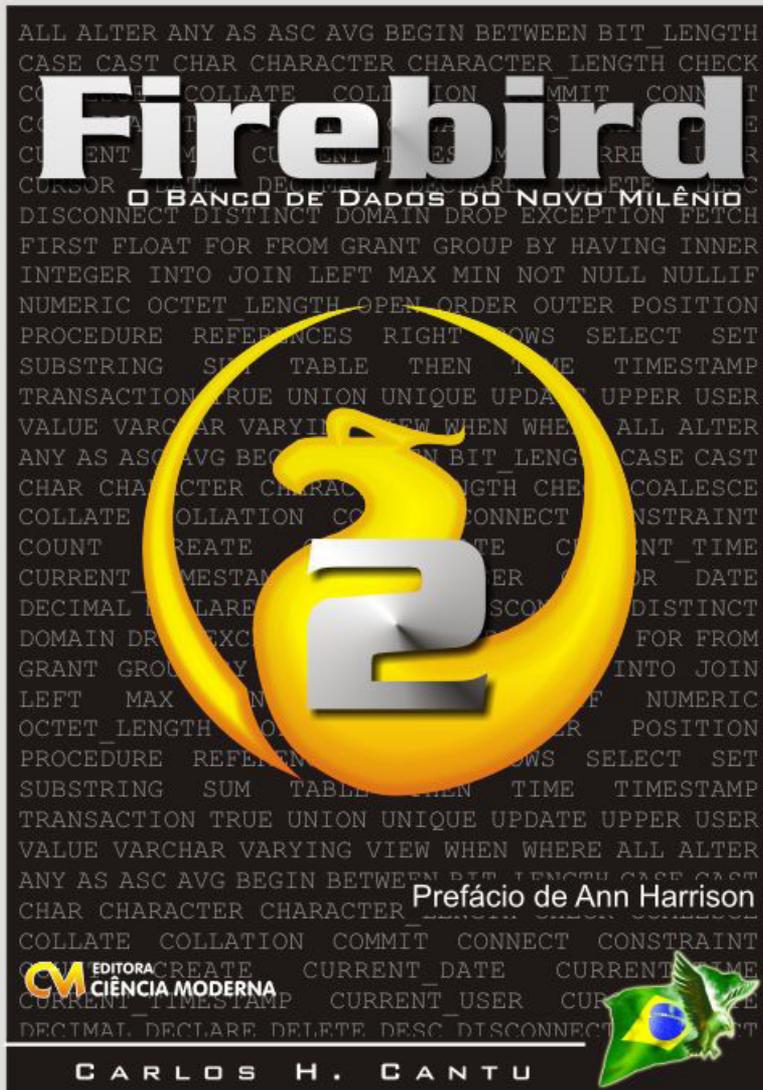
Avalie esse artigo

Proteja seu Software

www.proteq.com.br

SafeNet
The Foundation of Information Security

(11) 4208-7700



Após o grande sucesso de seu primeiro livro, **Firebird Essencial**, Carlos H. Cantu apresenta agora sua mais recente obra: **Firebird 2**.

O livro aborda as inúmeras novidades inseridas na versão 2.0 do Firebird. Cada recurso é discutido em capítulos específicos, sempre acompanhados de exemplos práticos que facilitam o entendimento do leitor. Os exemplos do livro são baseados em um banco de dados de *vendas de filmes*, tornando mais fácil a compreensão dos diversos temas abordados, em situações reais de utilização.

Apesar da maioria dos assuntos abordados serem **inéditos** nas obras do autor, alguns temas apresentados no livro Firebird Essencial são revisitados, englobando agora as alterações de comportamento inseridas na versão 2 do Firebird, além de informações adicionais, inéditas ou aprofundadas.

Todos os exemplos do livro são apresentados utilizando SQL e PSQL (linguagem procedural nativa do Firebird), com a exceção do capítulo sobre UDFs, que faz uso da linguagem Delphi/Pascal para criação das funções. **Este livro é indicado para todos os desenvolvedores, independente da linguagem de programação utilizada por eles**, pois trata essencialmente do Firebird.

Instalação do Firebird 2 no Linux e no Windows, Backups incrementais, Variáveis contextuais, Garbage Collection, Collations Case/Accent insensitive, Índices e Protocolos de Comunicação, são apenas alguns dos assuntos discutidos nesta obra, que pretende lhe poupar horas de investigação, trazendo as informações concentradas de forma clara e objetiva, uma das "marcas registradas" do autor.

Compre on-line no site da FireBase, e leve sua edição autografada!

www.FireBase.com.br

Calendário de Eventos

Data	Evento	Site
31/Março	SQL Connect 2007 Piracicaba - SP	http://www.sqlconnect.com.br
12-14 Abril/2007	8º Forum Internacional de Software Livre Porto Alegre - RS	http://fisl.softwarelivre.org
07-11 Maio/2007	InfoOeste 2007	http://www2.unoeste.br/~chaves/webfif/2006/lados_in_the_cam/
30/Jun-06/Jul 2007	VI Encontro Nacional de Inteligência Artificial	http://www.sbc.de9.ime.eb.br/br/eventos/enia.htm
14/Julho/2007	4º Firebird Developers Day Piracicaba - SP	http://www.FirebirdDevelopersDay.com.br

Se você sabe de algum evento focado em Banco de Dados ou em Software Livre que não esteja listado aqui, envie-nos um email com os dados do evento para que possamos incluí-lo na próxima edição e no calendário do site.

Vídeo-Aula



Curso de ClientDataSet com DBExpress e Firebird

9 horas de Vídeo-Aula por apenas
+ 2 Apostilas Impressa **RS 149,00 ***

* Ganhe 20% de desconto
adquirindo as apostilas em formato eletrônico

Cansado de corrupção de
tabelas e índices no Paradox?

Utilize o Firebird, um banco confiável,
robusto e gratuito!

Cansado do BDE, + de 10mb de instalação,
gerando muito tráfego em rede?

Utilize a DBExpress, menos de 150 kb de
instalação e super leve!

Este kit lhe dará todo caminho necessário para você começar a utilizar o banco Firebird
com a engine de acesso DBExpress em conjunto com o ClientDataSet

Mais informações acesse: www.edudelphipage.com.br por Eduardo Rocha

Aprendendo Firebird?

Softwares e produtos com desconto, DVDs, CDs, Livros!

Visite agora a loja on-line da FireBase!