

## **1) Introdução e Visão geral do PostgreSQL**

- 1.1) Resumo da História do PostgreSQL - 3
- 1.2) Características Avançadas - 4
- 1.3) Limites do PostgreSQL - 4
- 1.4) Principais Atribuições de um DBA - 5
- 1.5) Suporte ao PostgreSQL no Brasil e Quem Usa - 6
- 1.6) Modelo Relacional e Objeto-Relacional - 7
- 1.7) Instalação do PostgreSQL 8.3 no Windows - 10

### **1.1) Resumo da História do PostgreSQL**

O PostgreSQL atual é derivado do POSTGRES, escrito pela universidade de Berkeley na Califórnia (EUA). O POSTGRES foi inicialmente patrocinado pela Agência de Projetos de Pesquisa Avançados de Defesa (DARPA), pelo Escritório de Pesquisa sobre Armas (ARO), pela Fundação Nacional de Ciências (NSF) e pelo ESF, Inc.

PostgreSQL é o trabalho coletivo de centenas de desenvolvedores, trabalhando sob vinte-um anos de desenvolvimento que começou na Universidade da Califórnia, Berkeley. Com seu suporte de longa data a funcionalidades de nível corporativo de bancos de dados transacionais e escalabilidade, o PostgreSQL está sendo utilizado por muitas das empresas e agências de governo mais exigentes. O PostgreSQL é distribuído sob a licença BSD, que permite uso e distribuição sem ônus para aplicações comerciais e não-comerciais.

POSTGRES - 1986

POSTGRES 1 - 06/1989

POSTGRES 2 - 06/1990

POSTGRES 3 - 06/1991

O Informix (adquirido pela IBM) foi originado do código do POSTGRES

Em 1994 Andrew Yu and Jolly Chen adiciona um interpretador de SQL substituindo o PostQUEL

Ele muda de nome para Postgres95 e teve seu código completamente reescrito em ANSI C

Postgres95 - 01/05/1995

PostgreSQL 1 – 05/09/1995

Em 1996 muda novamente de nome, agora para PostgreSQL

PostgreSQL 6 - 29/01/1997 (do 1.09 passou para o 6)

PostgreSQL 7 - 08/05/2000

PostgreSQL 8 - 19/01/2005 (primeira versão for Windows)

PostgreSQL 8.1 - 08/11/2005

PostgreSQL 8.2 - 05/12/2006

PostgreSQL 8.3 - 04/02/2008 (versão atual)

Uma nova versão sai aproximadamente a cada ano.

#### **Fontes:**

<http://www.postgresql.org/docs/8.3/interactive/release.html> e

<http://www.postgresql.org/docs/8.3/interactive/history.html>

## 1.2) Características Avançadas do PostgreSQL

- Exporta em XML
- Busca Textual, com TSearch2
- Novos tipos de dados: ENUM e matrizes de tipos compostos
- Suporte a SNMP
- Transações
- Replicação
- Banco de dados objeto-relacional
- Suporte a transações (padrão ACID)
- Lock por registro (row level locking)
- Integridade referencial
- Número ilimitado de linhas e índices em tabelas
- Extensão para GIS (base de dados geo-referenciados)
- Acesso via drivers ODBC e JDBC
- Interface gráfica de gerenciamento
- Uso otimizado de recursos do sistema operacional
- Suporte aos padrões ANSI SQL 92 e 99
- Joins: Implementa todos os tipos de join definidos pelo padrão SQL99: inner join, left, right, full outer join, natural join.
- Triggers, views e stored procedures
- Suporte ao armazenamento de BLOBs (binary large objects)
- Sub-queries e queries definidas na cláusula FROM
- Backup online
- Sofisticado mecanismo de tuning
- Suporte a conexões de banco de dados seguras (criptografia)
- Modelo de segurança para o acesso aos objetos de banco de dados por usuários e grupos de usuários

Fonte: site do Dextra Treinamentos

Lista completa: <http://www.postgresql.org/about/press/features83> e <http://www.postgresql.org/docs/8.3/static/release-8-3.html>

### 1.3) Limites do PostgreSQL

Tamanho de um Banco de Dados - ilimitado

Tamanho de uma tabela - 64 TB

Quantidade de registros por tabela - ilimitados

Quantidade de campos por tabela - 250 a 1600 (depende do tipo)

Quantidade de índices por tabela - ilimitados

Tamanho máximo de um campo - 1GB para a versão 7.1 e posteriores

Máximo de colunas numa tabela - 1600

### Licença

O PostgreSQL usa a [licença BSD](#), que apenas requer que o código fonte sob licença mantenha o seu direito de cópia e a informação da licença. Essa [licença certificada pela OSI](#) é amplamente vista como flexível e amigável a empresas, já que ela não restringe o uso do PostgreSQL com aplicações comerciais e proprietárias. Juntamente com suporte de várias empresas e propriedade pública do código, a licença BSD torna o PostgreSQL muito popular entre empresas que querem embutir o banco de dados nos seus próprios produtos sem temor de taxas, enclausuramento a uma empresa, ou mudanças nos termos da licença.

### 1.4) Principais Atribuições de um DBA

DBA - DataBase Administrator – Administrador de Banco de Dados

É o profissional responsável pela administração e gerenciamento do banco de dados. Isto normalmente envolve:

- A instalação do próprio (PostgreSQL, SQL Server, Oracle, Saybase, MySQL, etc) em um servidor específico para as bases de dados da organização.
- Cálculo e dimensionamento de clusters e blocagem para criação/instalação das bases de dados.
- A criação de usuários e o controle sobre o que cada um pode ou não fazer na base de dados (grants) e toda a segurança relacionada a acessos às bases de dados (pg\_hba.conf).
- A criação de objetos de dados como bancos, esquemas, tabelas, índices, constraints, views, funções, triggers, sequences, etc, seguindo as especificações (scripts de criação) de um analista de sistemas, engenheiro, arquiteto de software ou técnico responsável pelas atividades de modelagem de um projeto de software.
- Analisar os scripts de criação de objetos, propondo determinadas alterações eventualmente necessárias ao aumento de performance e ganho do banco de dados.
- Análise periódica das bases de dados considerando aspectos como estimativas e dados de crescimento das bases, performance, estatísticas de utilização, transações, etc.
- Elaboração (eventual) e garantia (normal) de nomenclaturas e padrões que devem ser seguidos pelos modeladores de software para objetos de dados (nomes de tabelas, atributos, índices, etc).

- Dimensionamento de hardware e software para instalação de bases de dados da organização ou de um determinado projeto de software.
- Planejamento, execução e guarda de cópias de segurança das bases de dados, bem como, eventualmente, a execução de atividades necessárias à restauração.

Eventualmente os DBA's envolvem-se com o desenvolvimento de Stored Procedures, Packages/Procedures nos bancos de dados. Mas isto não é normal.

Fonte: Espia em <http://forum.wmonline.com.br/index.php?showtopic=25911>

## **1.5) Suporte ao PostgreSQL no Brasil e quem usa**

### **Empresas que Patrocinam o PostgreSQL:**

- InterpriseDB
- Fijitsu
- RedHat
- Skype
- Sun

Lista completa - <http://www.postgresql.org/about/sponsors>

### **Oferecem Suporte ao PostgreSQL (consultores individuais e empresas):**

Lista completa: [http://www.postgresql.org/support/professional\\_support](http://www.postgresql.org/support/professional_support)

### **Mais empresas e consultores que oferecem suporte ao PostgreSQL:**

[http://www.opensourceexperts.com/Index/index\\_html/PostgreSQL/index.html](http://www.opensourceexperts.com/Index/index_html/PostgreSQL/index.html)

### **Oferecem Suporte ao PostgreSQL (no Brasil):**

- Carlos Smanioto (Bauru)
- dbExperts (SP)
- Dextra (Campinas)
- Flexsolutions Consultores (Goiânia)
- Locadata (Belo Horizonte)
- Mosman Consultoria e Desenvolvimento (São Pedro - SP)
- Studio Server Hospedagem de Sites (Santos)

Detalhes e Lista completa: [http://www.postgresql.org/support/professional\\_support\\_southamerica](http://www.postgresql.org/support/professional_support_southamerica)

### **Algumas Empresas que usam o PostgreSQL (internacionais):**

- Apple
- BASF
- Cisco
- OMS (Organização Mundial de Saúde)

## Algumas Empresas que usam o PostgreSQL (nacionais):

- FAB (Força Aérea Brasileira)
- Prefeitura Municipal de Sobral
- Vivo
- DNOCS

Obs.: Comenta-se que o governo do estado do Ceará vai adotar o PostgreSQL como SGBD oficial do Estado, ficando todas as suas secretarias obrigadas a converter suas bases para PostgreSQL. O governo do Rio de Janeiro já faz isso há alguns anos.

## Equipe de Desenvolvimento do PostgreSQL

These are the fine people that make PostgreSQL what it is today!

### Core Team

### Major Contributors

#### Contributors

Gevik Babakhani (pgdev at xs4all.nl)  
Christopher Browne (cbbrowne at ca.afiliat.info)  
Sean Chittenden (sean at chittenden.org)

Korry Douglas (korryd at enterprisedb.com)

Stephen Frost (sfrost at snowman.net)  
Michael Glaesemann (grzm at myrealbox.com)  
John Hansen (john at geeknet.com.au)  
Kris Jurka (jurka at ejurka.com)  
Manfred Koizar (mkoi-pg at aon.at)  
Marko Kreen (marko at l-t.ee)  
Heikki Linnakangas (hlinnaka at iki.fi)  
Abhijit Menon-Sen (ams at oryx.com)

Matthew T. O'Connor (matthew at zeut.net)

Larry Rosenman (ler at lerctr.org)  
Pavel Stehule (stehule at kix.fsv.cvut.cz)  
ITAGAKI Takahiro (itagaki.takahiro at lab.ntt.co.jp)  
Joachim Wieland (joe at mcknight.de)  
Andreas Zeugswetter (ZeugswetterA at wien.spardat.at)

Dennis Björklund (d at zigo.dhs.org)

D'Arcy Cain (darcy at druid.net)

Fabien Coelho (coelho at cri.ensmp.fr)

**Francisco Figueiredo** (fxjrlists at yahoo.com.br)

Michael Fuhr (mike at fuhr.org)

Thomas Hallgren (thomas at tada.se)

Jonah Harris (jonah.harris at gmail.com)

Mark Kirkwood (markir at paradise.net.nz)

Sergey E. Kopolov (math at sai.msu.ru)

Hannu Krosing (hannu at skype.net)

Robert Lor (Robert.Lor at Sun.com)

Fernando Nasser (fnasser at redhat.com)

**Euler Taveira de Oliveira** (eulerto at yahoo.com.br)

Greg Stark (stark at enterprisedb.com)

Sven Suursoho (sven at spam.pri.ee)

Philip Warner (pjm at rhyme.com.au)

Mark Wong (markw at osdl.org)

Qingqing Zhou (zhouqq at cs.toronto.edu)

### Hackers Emeritus

The following hackers were previously part of the core team. Although they no longer work on the project, they are included here in recognition of their valuable contributions over the years.

#### Contributor

#### Contribution

Thomas G. Lockhart (lockhart at alumni.caltech.edu)

[Jet Propulsion Laboratory](#)

Pasadena, California, USA

Vadim B. Mikheev (vadim4o at yahoo.com)

Worked on documentation, data types (particularly date/time and geometric), and SQL standards compliance.

Undertook large projects, like vacuum, subselects, triggers, Write Ahead Log (WAL) and multi-version concurrency control

San Francisco, California, USA (MVCC).

### Past Contributors

Julian Assange (proff at suburbia.net)	David Bennett (dave at bensoft.com)
Alexey Borzov (borz_off at cs.msu.su)	Emily Boyd (emily at emilyboyd.com)
Justin Clift (justin at postgresql.org)	Massimo Dal Zotto (dz at cs.unitn.it)
Dr_George_D Detlefsen (drgeorge at ilt.com)	Chris Dunlop (chris at onthe.net.au)
J. Douglas Dunlop (dunlop at eol.ists.ca)	Oliver Elphick (olly at lfix.co.uk)
Brian E. Gallew (geek+ at cmu.edu)	Cees de Groot (C.deGroot at inter.NL.net)
Christopher Kings-Lynne (chris.kingslynne at gmail.com)	Anoop Kumar (anoopk at pervasive-postgres.com)
Randy Kunkee (kunkee at Starbase.NeoSoft.COM)	Kurt J. Lidl (lidl at va.pubnix.com)
Barry Lind (barry at xythos.com)	Dan McGuirk (mcguirk at indirect.com)
<b>Roberto Mello</b> (roberto.mello at gmail.com)	Ernst Molitor (ernst.molitor at uni-bonn.de)
Claudio Natoli (claudio.natoli at memetrics.com)	Lamar Owen (lamar.owen at wgcr.org)
Ross J. Reedstrom (reedstrm at rice.edu)	Thomas van Reimersdahl (reimersd at dali.techinfo.rwth-aachen.de)
Sergej Sergeev (sergej at commandprompt.com)	Michael Siebenborn (michael.siebenborn at ae3.Hypo.DE)
Erich Stamberger (eberger at gewi.kfunigraz.ac.at)	Adam Sussman (myddryn at vidya.com)
Reini Urban (urban at x-ray.at)	Paul "Shag" Walmsley (ccshag at cclabs.missouri.edu)
Rick Weldon (rick at wisetech.com)	Jason Wright (jason at shiloh.vnet.net)
Karel Zak (kzak at redhat.com)	

All developers are listed in alphabetical order. Please report omissions or corrections

Lista completa em: <http://www.postgresql.org/community/contributors/>

## Guide to PostgreSQL version numbers

Josh Berkus

Thursday, September 20, 2007

With the [recent release of a PostgreSQL update](#), I was reminded again by a bewildered user that a lot of people don't seem to get the essentially simple structure of the PostgreSQL version numbers. There's a [nice page about versioning](#), but I doubt most users have read it.

To add a little to what it says there, consider the PostgreSQL version:

### PostgreSQL 8 . 1 . 10

The **8** is a "major version number", but could also be known as the "marketing version number". Every few years, based on making some long-sought landmark, we bump the first digit. For **7** (in 1999) it was not crashing; for **8** it was the Windows port, combined with cumulative improvements. When we'll do **9**, nobody knows; probably when we start to beat proprietary databases on *all* workloads.

The **1** in the second digit place is *also* a major release. It indicates the annual release of PostgreSQL. You combine the first two digits ... in this case "**8.1**" to make your major version, or major release, or branch, or sometimes just "version" of PostgreSQL. Users upgrading between major versions can expect an involved upgrade process and testing, which is why we support back branches so long. This using the first two digits confuses many commercial software developers, who assume that only the first digit is significant. If we used only the first digit, though, we'd be close to releasing PostgreSQL version **19**.

That **10** in the last place is the "minor version", which is a synonym for "patch release number". **8.1.10** thus includes 10 cumulative patches to major version **8.1**. Whatever your major version, it's very important that you update to the latest minor version of it. The PostgreSQL project issues only bug fixes in patches, and never feature enhancements, so if you are not on the latest minor version, then you're exposing yourself to known potential security and data loss threats.

Updating to the latest minor release does not require an "initdb" or "dump/reload", and most releases is possible within 2 minutes of downtime (as long as it takes you to copy over the binaries and restart). Occasionally, however, a security fix requires some API changes which will be documented in the release notes. If you're jumping a half dozen minor releases, make sure you scroll back through the release notes for warnings.

Now, go and update your PostgreSQL servers!

## **1.7) Instalação do PostgreSQL 8.3 no Windows**

### **Download**

Fazer o download de - <http://www.postgresql.org/ftp/binary/v8.3.0/win32/>

Caso esteja baixando outra versão deverá acessar:

<http://www.postgresql.org/ftp> e depois clicar em win32.

No caso da 8.3 selecionar e baixar o arquivo [postgresql-8.3.0-1.zip](#), que tem 20MB. É importante verificar o tamanho do arquivo baixado para garantir que o download está correto.

### **Remover Usuário Anterior**

Caso tenha uma instalação anterior do postgresql deverá, antes de instalar a nova, remover o super usuário da versão anterior:

- Painel de controle – Ferramentas Administrativas – Administração de Computadores – Usuários e Grupos locais – Usuários (remover o postgres)

### **Descompactar e Instalar**

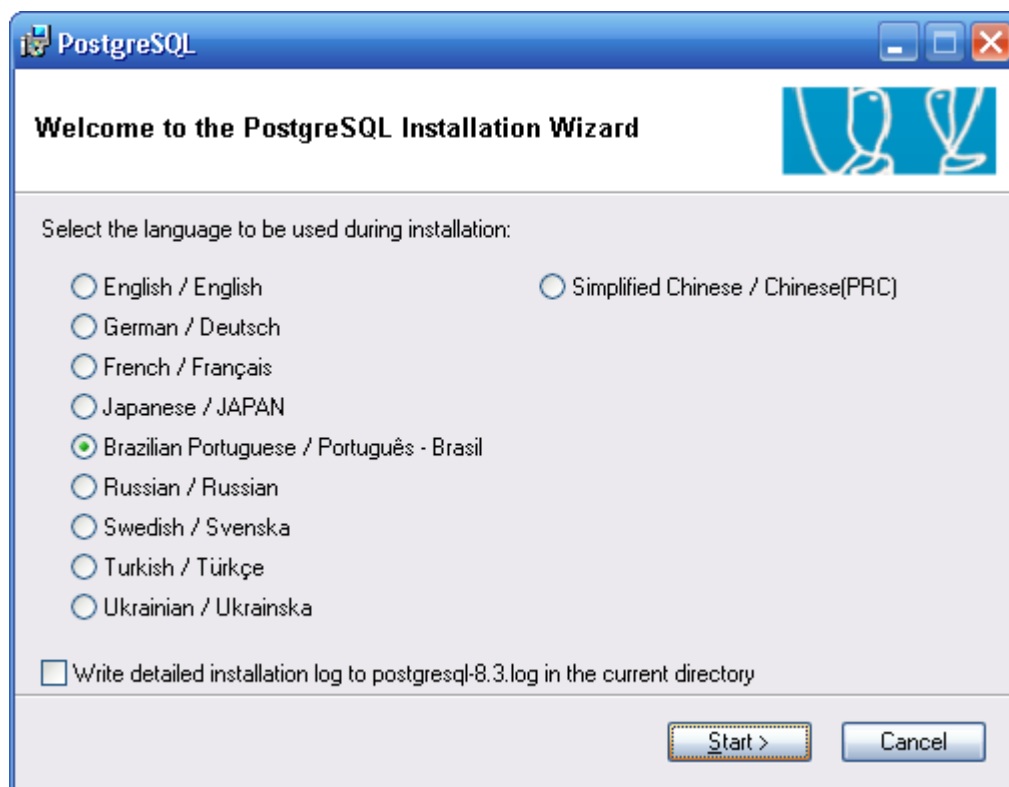
Descompacte o arquivo e verá 5 arquivos: postgresql-8.3.msi, postgresql-8.3-int.msi, README.txt, SETUP.bat e UPGRADE.bat.

O arquivo UPGRADE.bat é para efetuar uma atualização de versão existente.

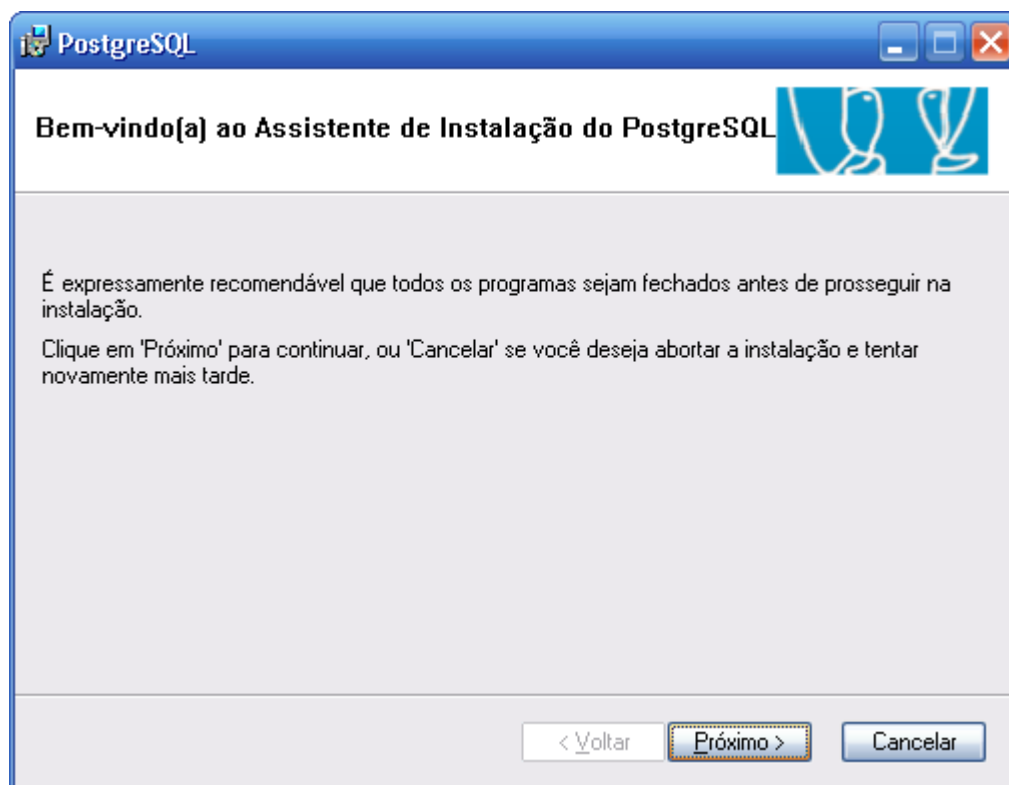
O arquivo postgresql-8.3.msi é que deve ser executado para a instalação.



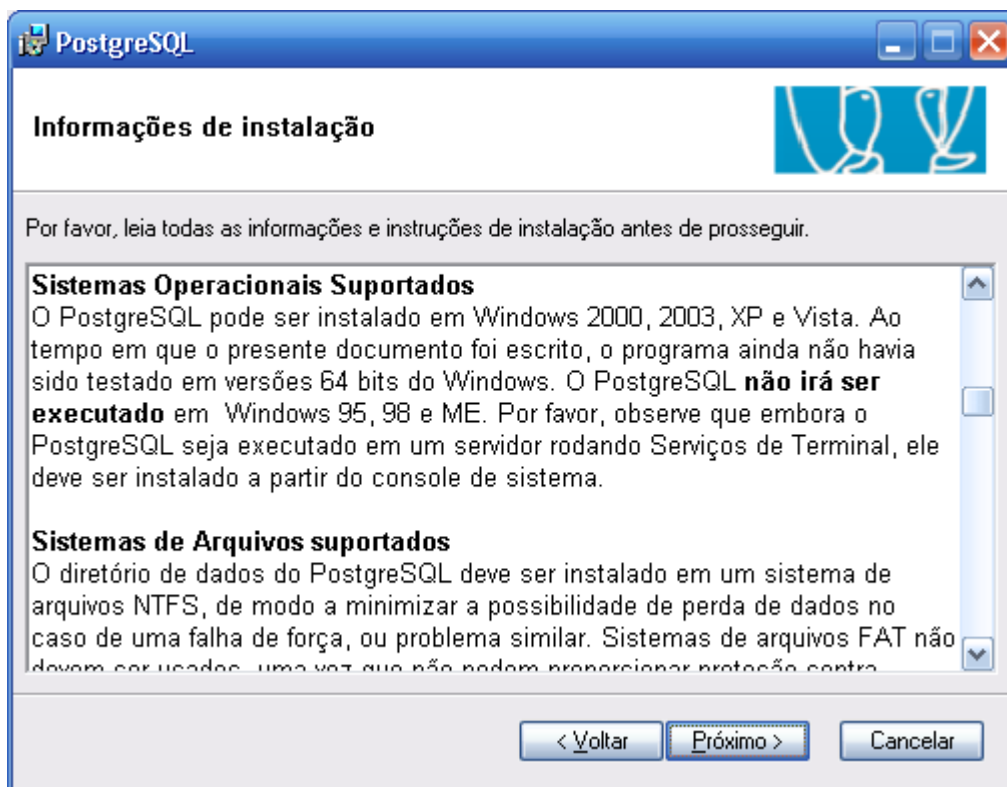
Execute o arquivo postgresql-8.3.msi:



Clique em Start.

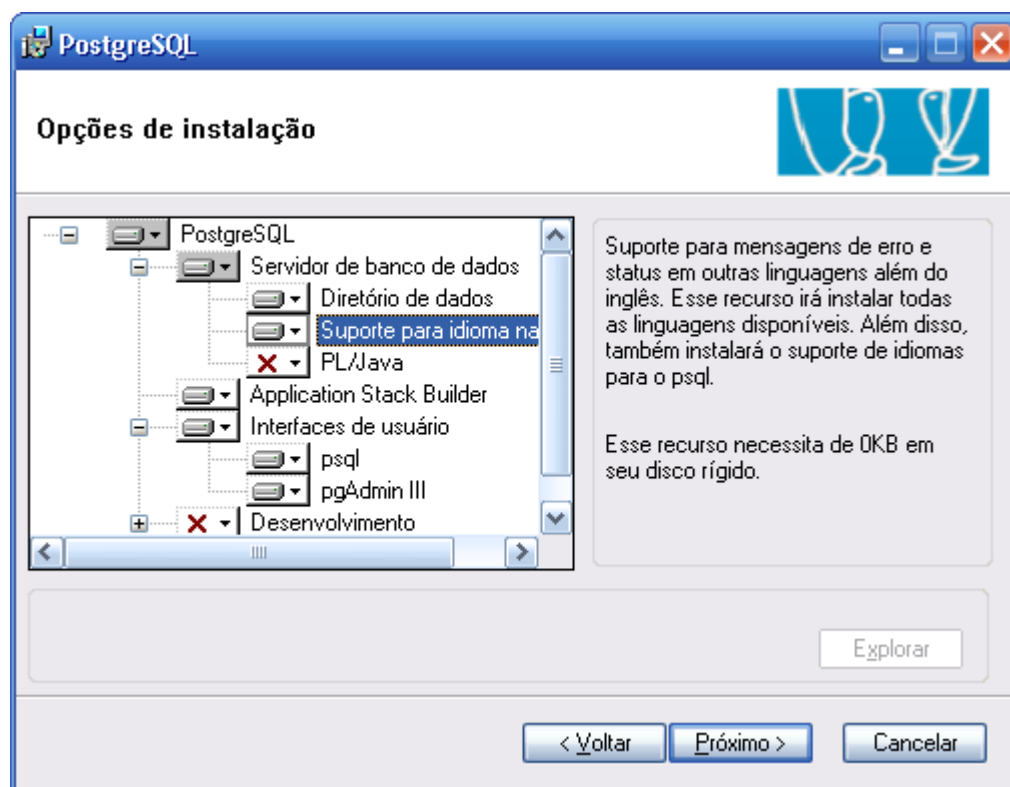


Desta tela em diante a instalação estará em português do Brasil.  
Feche todos os programas abertos e clique em próximo.



Neste tela temos informações importantes, como os sistemas operacionais suportados pela versão for Windows do PostgreSQL, que são o 2000, 2003, XP e Vista, como também o sistema de arquivos que é somente o NTFS.

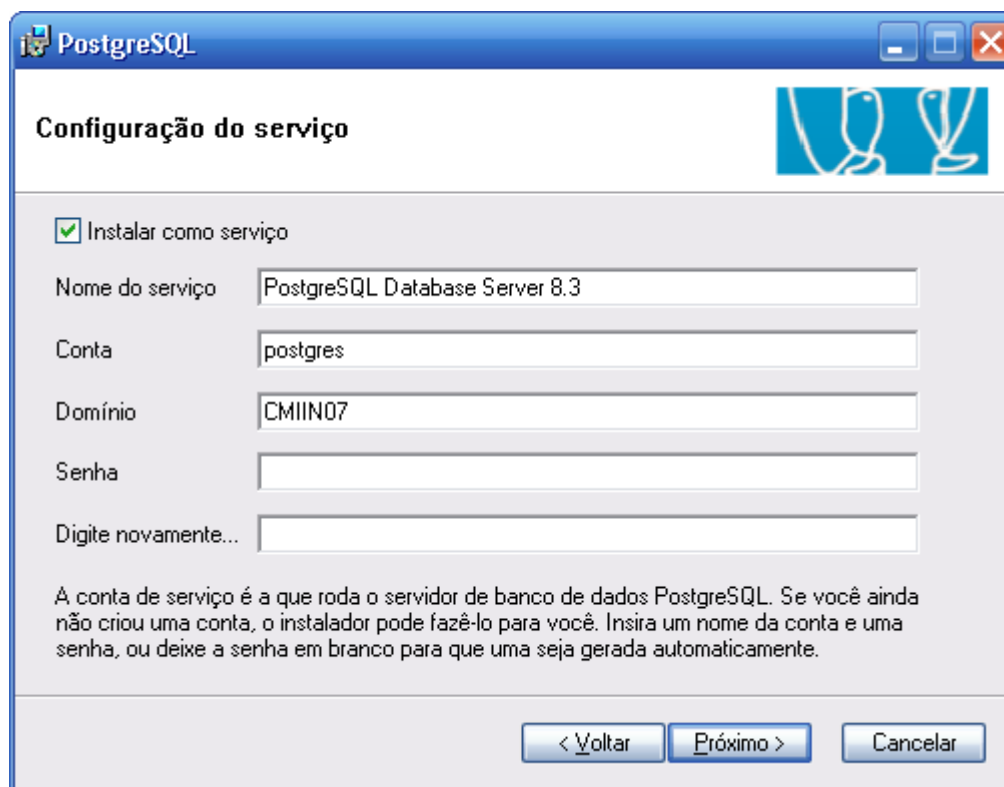
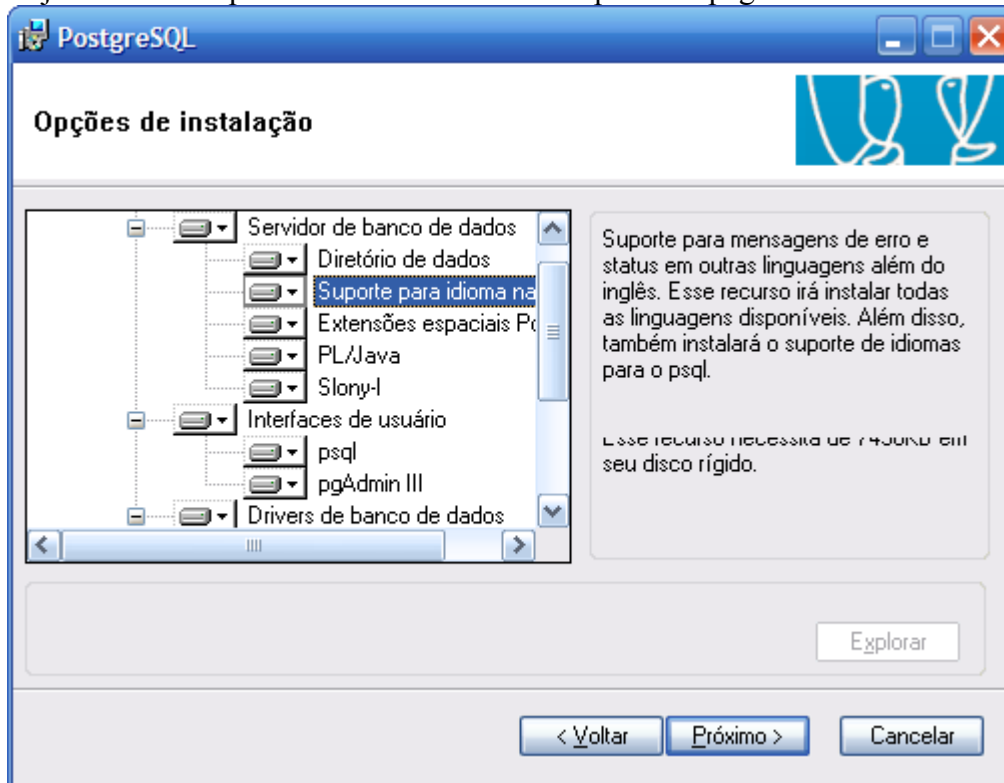
Clique em Próximo.



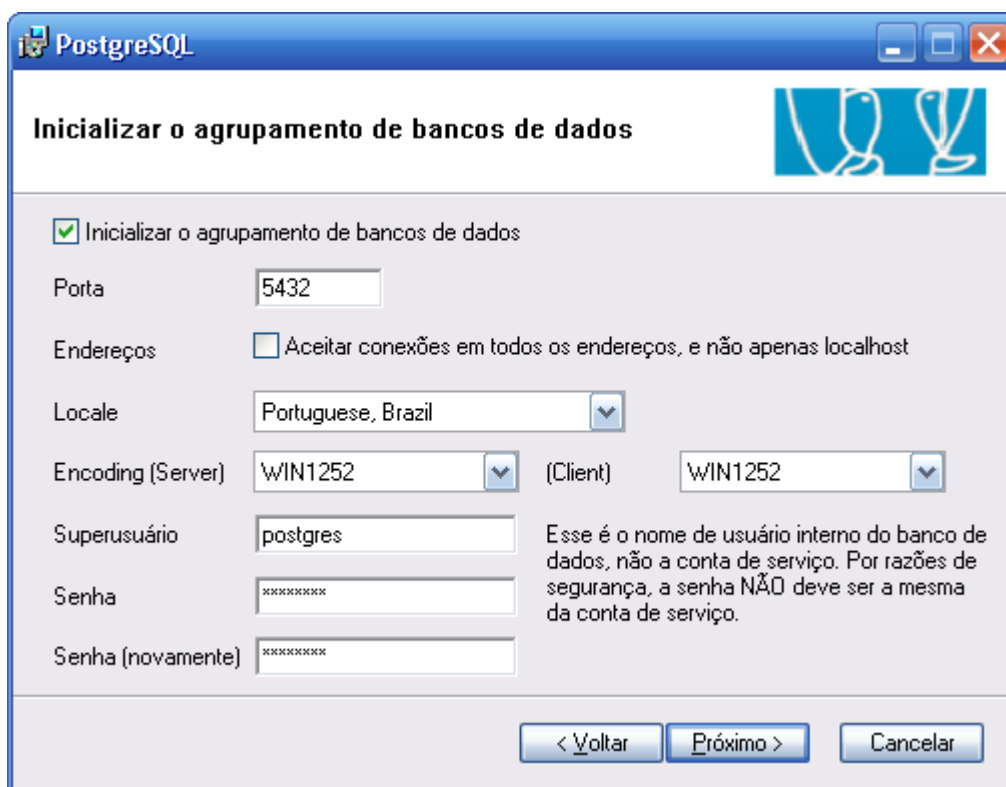
Nesta tela devemos ativar o Suporte para idiomas nativos, se quisermos ver mensagens de erro e o psql em português.

Clique em Próximo.

Veja a tela correspondente da versão 8.2.7 na próxima página



Nesta tela deixe a senha em branco, clique em Próximo e confirme. Veja que é opcional instalar como serviço, mas altamente recomendado.



PostgreSQL

### Inicializar o agrupamento de bancos de dados

☒ Inicializar o agrupamento de bancos de dados

Porta: 5432

Endereços: ☐ Aceitar conexões em todos os endereços, e não apenas localhost

Locale: Portuguese, Brazil

Encoding (Server): WIN1252 (Client): WIN1252

Superusuário: postgres

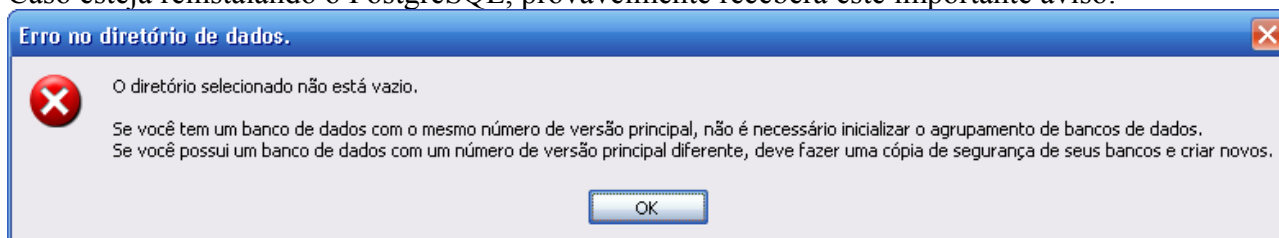
Senha:

Senha (novamente):


Esse é o nome de usuário interno do banco de dados, não a conta de serviço. Por razões de segurança, a senha NÃO deve ser a mesma da conta de serviço.

< Voltar Próximo > Cancelar

Caso esteja reinstalando o PostgreSQL, provavelmente receberá este importante aviso:



Erro no diretório de dados.

 O diretório selecionado não está vazio.

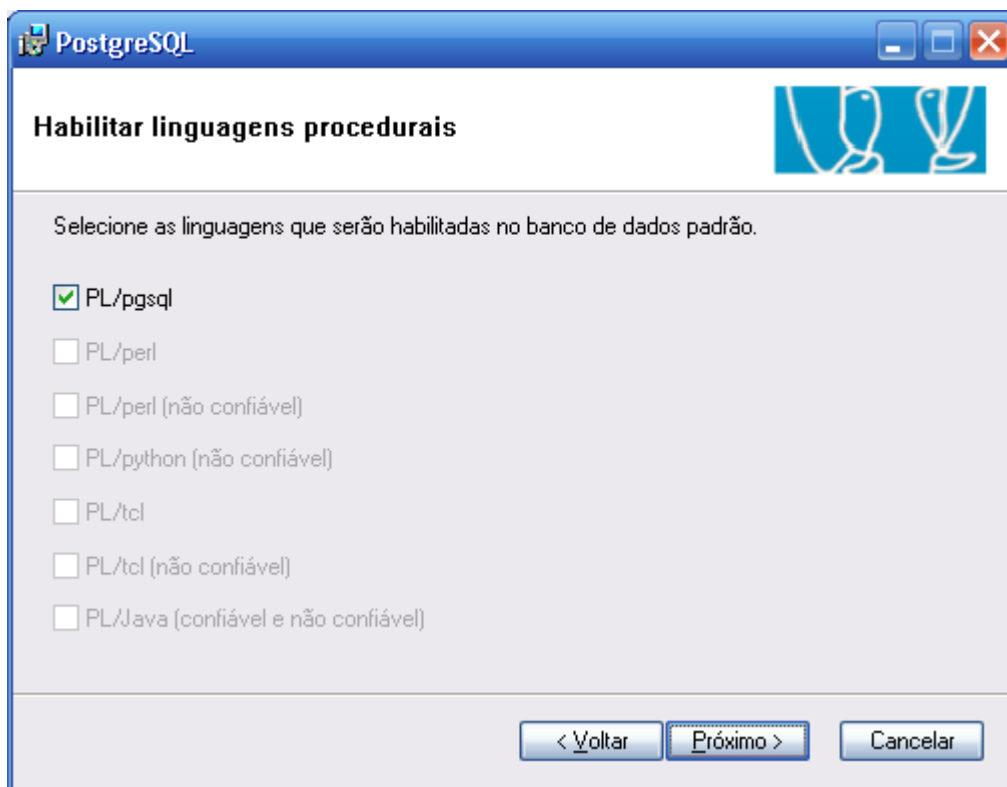
Se você tem um banco de dados com o mesmo número de versão principal, não é necessário inicializar o agrupamento de bancos de dados.  
Se você possui um banco de dados com um número de versão principal diferente, deve fazer uma cópia de segurança de seus bancos e criar novos.

OK

Na tela abaixo deixemos o **Encoding** do servidor e do cliente como Win1252, que são o padrão do Windows. Entre também com a senha do super usuário e clique em Próximo.

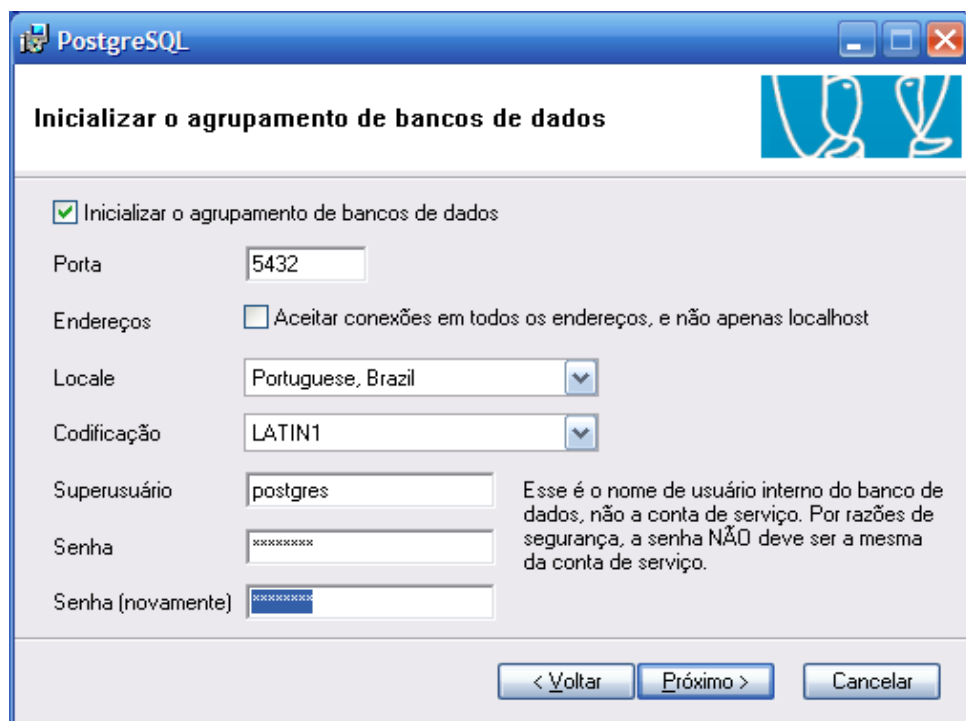
O **Locale** deve ser de acordo com o sistema operacional.

Nesta tela existe a opção de já configurar para aceitar conexões em todos os endereços, mas somente utilize com grande segurança do que está fazendo, pois o ideal em termos de segurança é permitir apenas acesso local.



Nesta tela desmarque o item PL/pgsql e lembre de ativar o suporte a plpgsql apenas nos bancos em que for usar essa linguagem. Assim terá os bancos mais limpos.

A tela correspondente da versão 8.2.7

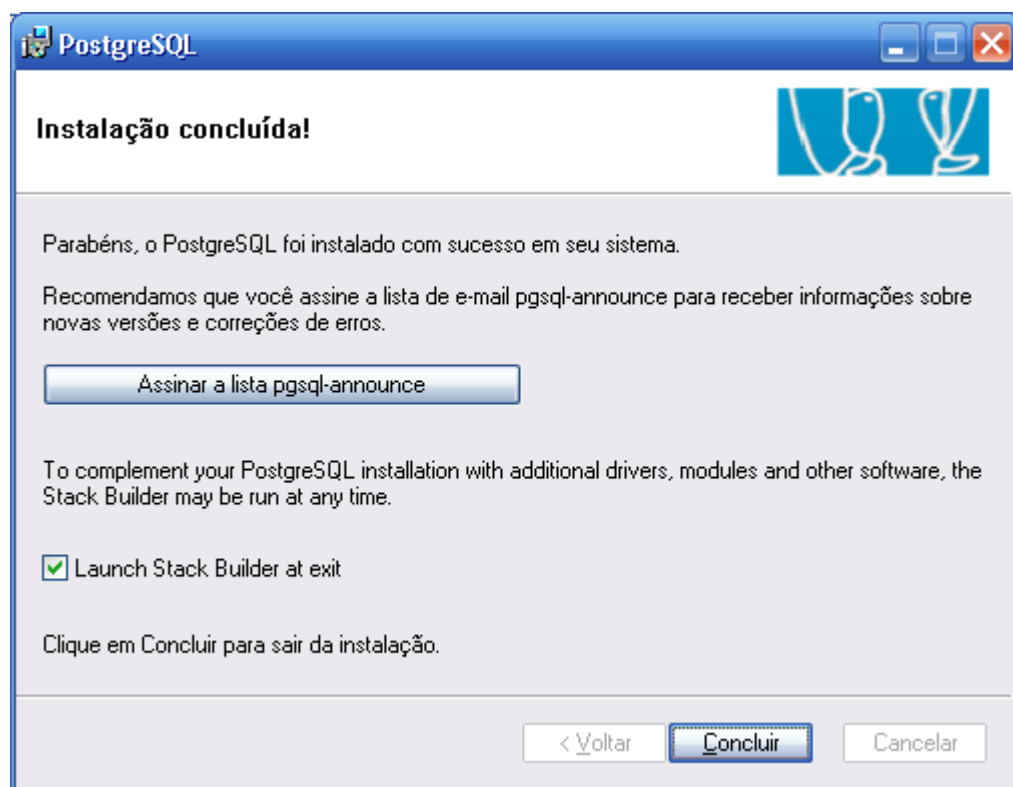


Clique em Próximo.



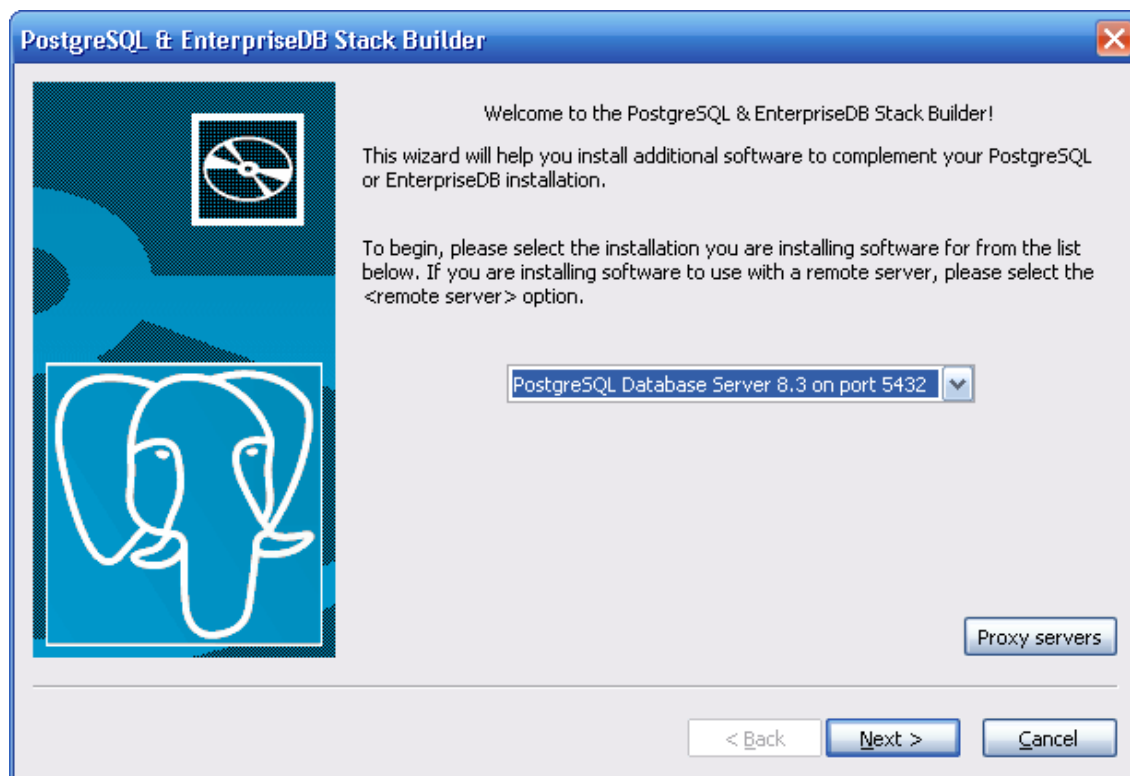
Esta tela é a das extensões chamadas cotribs. Marque apenas se precisar e souber de fato o que está fazendo. Deixe como está e clique em Próximo e confirme novamente na próxima tela..

Caso tenha um bom firewall instalado verá duas solicitações de uso das portas 5432 e da 2427. Aceite o acesso.



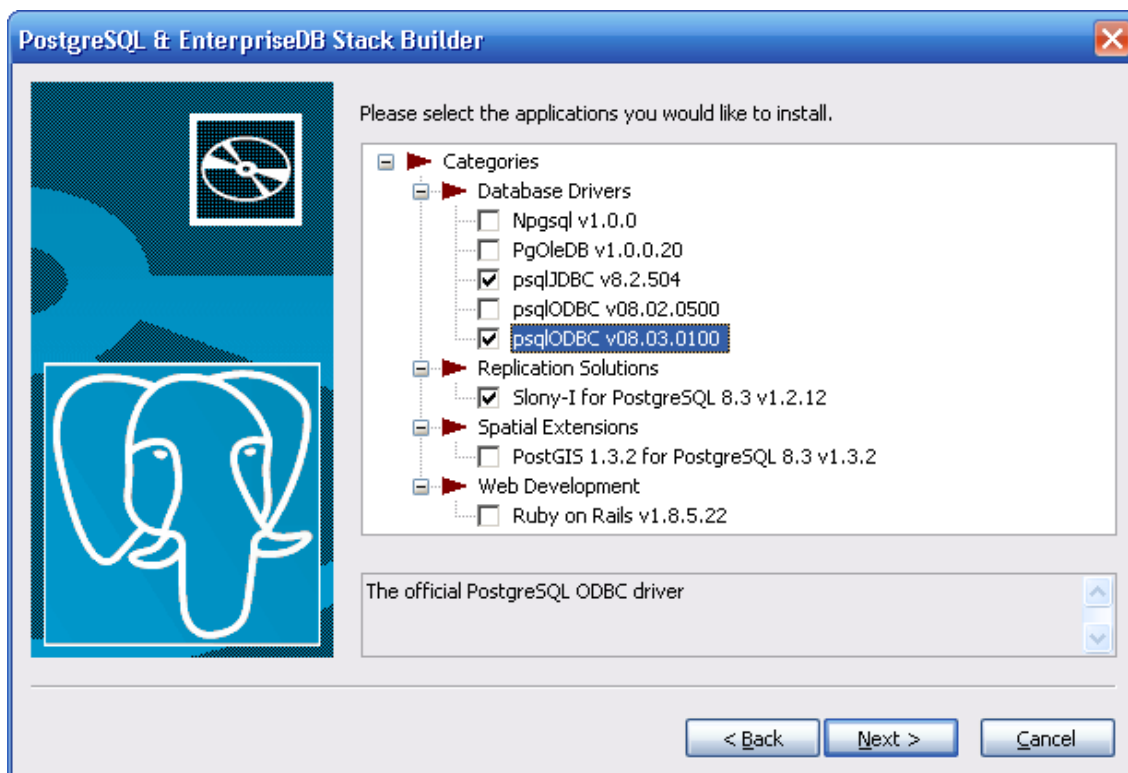
Caso você esteja realmente empenhado em aprender sobre o PostgreSQL aproveite esta oportunidade para assinar a lista pgsq-announce clicando no botão. Esta lista envia em torno de um e-mail por dia sobre as novidades do postgresql, novas ferramentas e informações importantes de segurança, bugs, etc.

Também nesta tela temos a opção de instalar drivers: JDBC para o postgresql, ODBC e outros, como outros softwares adicionais. Basta deixar marcado o checkbox e clicar em Concluir.

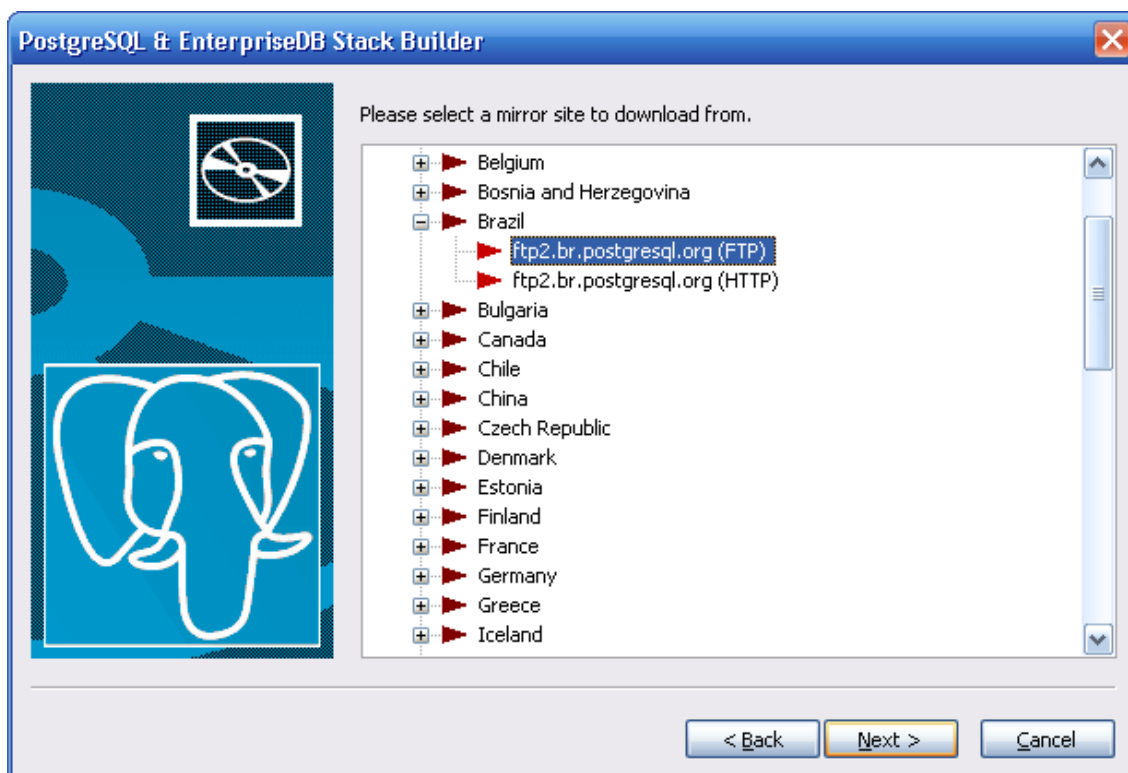


Selecione o servidor e caso esteja numa rede com proxy entre com as informações e clique em Next.

Novamente deverá dar permissão no firewall para que acesse o site para os downloads.

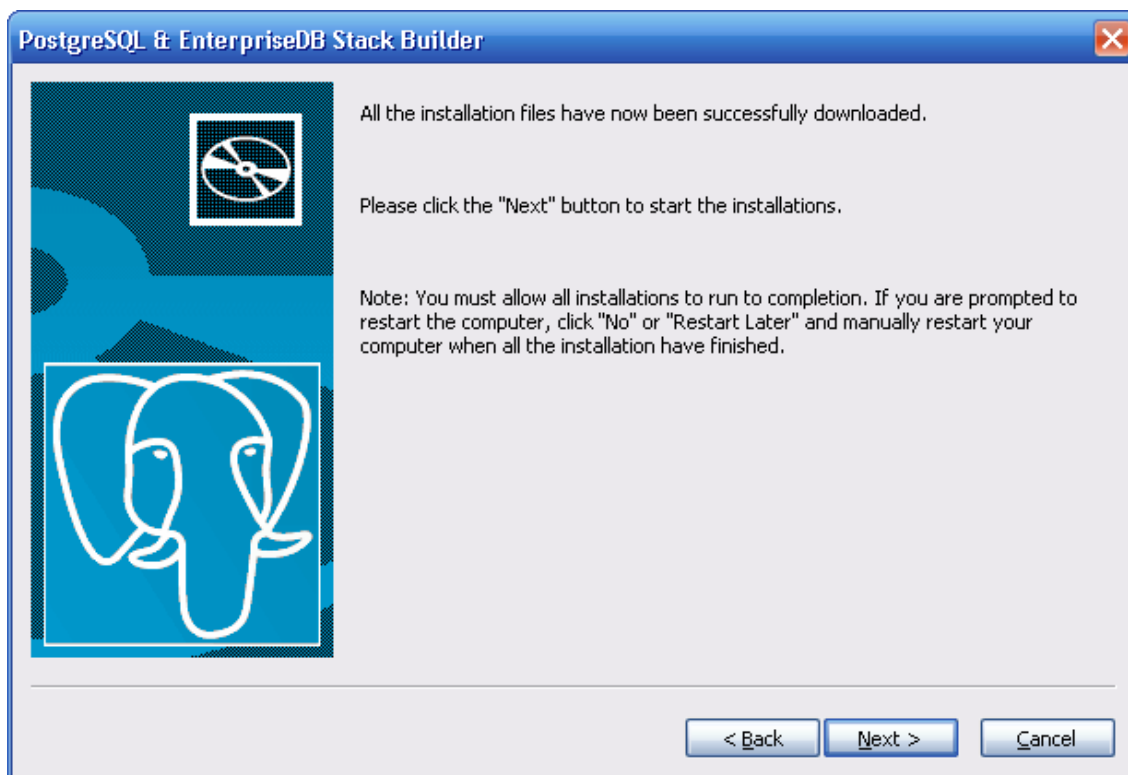


Marquei dois drivers importantes e o software para replicação Slony.  
Deve seleccionar os que deseja e clicar em Next.



Apenas selecione o espelho e clique em Next e na próxima tela confirme em Next novamente e aguarde o download.

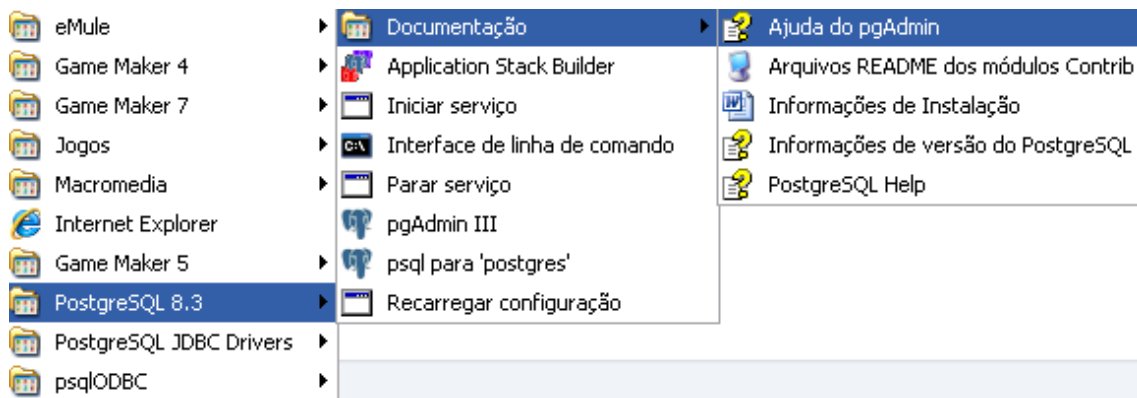




Veja a recomendação de caso seja solicitado reboot no micro, cancele e faça o reboot manualmente depois, quando todos os processos da instalação se concluírem.

Agora ele irá instalar os programas selecionados e finalizar a instalação.

Veja que agora temos alguns grupos de programas junto ao do PostgreSQL:



Temos aí o postgresql, os drivers, o pgadmin e o psql, além da documentação oficial em inglês sobre o postgresql e sobre o pgadmin. Além de alguns links para parar o serviço e iniciar e outros.

## **"Instalação" e Configurações do PostgreSQL for Windows do tipo Non-installer**

# Download do pacote para Win32 tipo binaries-no-installer:

[postgresql-8.3.0-2-binaries-no-installer.zip](http://www.postgresql.org/ftp/binary/latest/8.3.0-2/binaries-no-installer.zip)

# Descompactar (ex.: raiz do c:\)

# Acessar o prompt de comando e entrar no diretório criado (pgsql)

# Criar o cluster

cd pgsql\bin

initdb -D pgdata # ou outro diretório, como c:\data

Com o comando acima será criado o cluster de bancos do PostgreSQL na subpasta (pgdata) no diretório bin

# Iniciar o processo

bin\pg\_ctl -D pgdata start

# Criar usuário

bin\createuser postgres

# Acessar psql

bin\psql -U postgres

# Parar o processo

bin\pg\_ctl -U postgres -D pgdata stop

Para tornar mais ágeis e não precisar lembrar destes comandos podemos criar arquivos .bat que executem estes comandos acima.

Sugiro que criemos:

- Para a criação do cluster e do usuario não há necessidade, já que será uma única vez
- um para startar o processo
- um para acessar o psql
- um para parar o processo

Para facilitar ainda mais os .bats devem ficar no diretório bin.

# pgstart.bat

pg\_ctl -D pgdata start

Obs.: após este comando tecle Enter

# pgpsql.bat

psql -U postgres

Obs.: este comando acessa o banco default do postgres, que é o postgres.

# pgstop.bat

pg\_ctl -D pgdata stop

<http://www.microsoft.com/downloads/details.aspx?FamilyID=200b2fd9-ae1a-4a14-984d-389c36f85647&DisplayLang=en>