



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS



Bancos de Dados Geográficos

#05 – Introdução ao PostgreSQL

Dr. Gilberto Ribeiro de Queiroz <gribeiro@dpi.inpe.br>

Dr. Eymar Lopes <eymar@dpi.inpe.br>

PostgreSQL

- É um sistema gerenciador de bancos de dados objeto-relacional open-source (licença ao estilo BSD):
 - <http://www.postgresql.org>
- O código fonte do seu núcleo encontra-s escrito na Linguagem de Programação C.
- Disponível para diversas plataformas: Linux, Mac OS X e Microsoft Windows.
- Versão que usaremos durante o curso: 9.3.0

Histórico do PostgreSQL

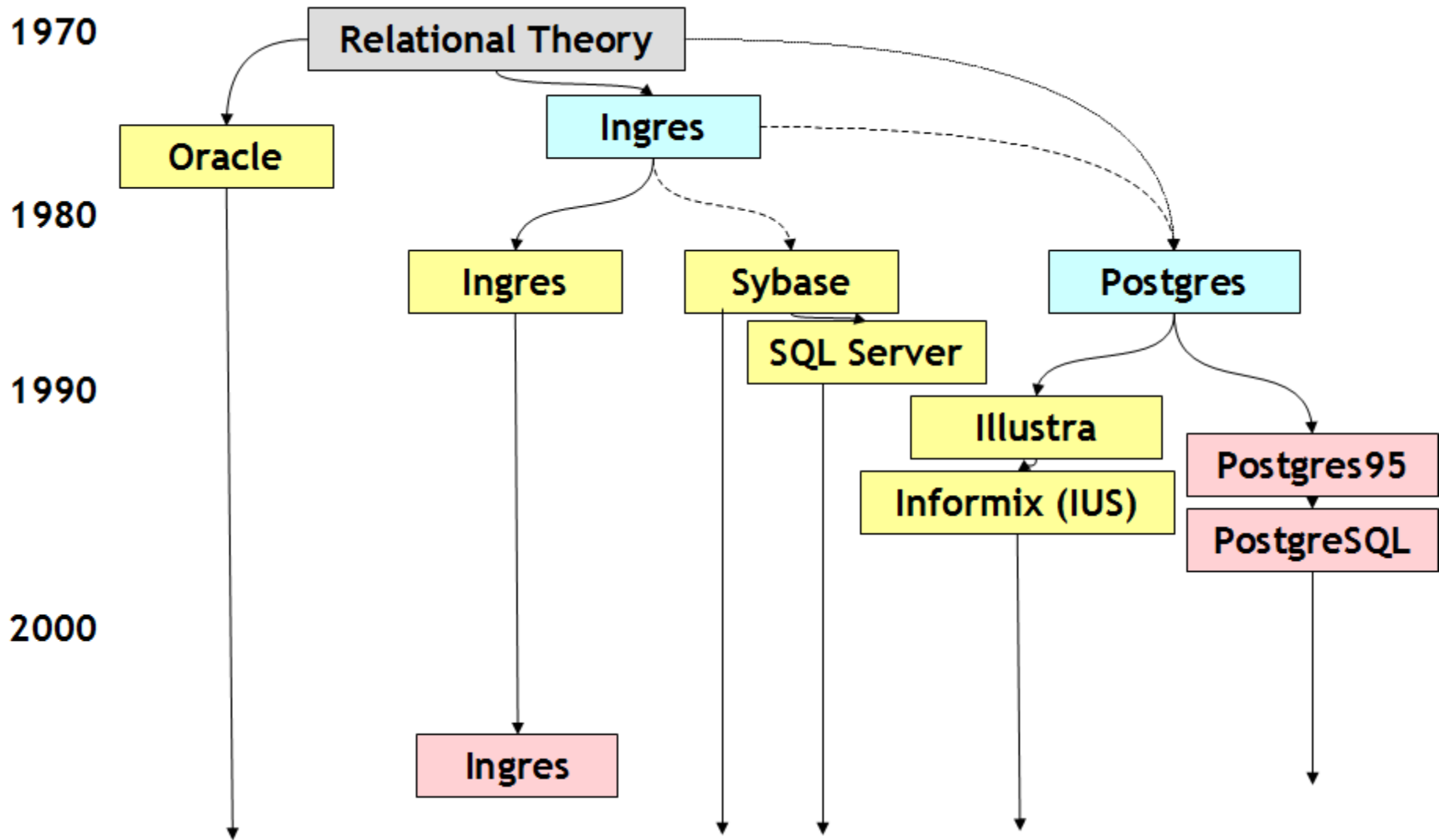


Fonte: Wikipedia (2013)

- Michael Stonebraker*:
 - Pesquisador na Universidade da California e MIT
- Liderou um time de alunos de graduação que criaram implementações reais de SGBDs:
 - Ingres (1974-1984)
 - Postgres (1985-1993)
 - Cohera (1997-2001)
 - Aurora
 - C-Store (Vertica)
 - VoltDB
 - SciDB

*ACM Turing Award (2014)

Histórico do PostgreSQL



Fonte: Ramsey (2007)

PostgreSQL: características

- Funções (UDFs): PL/pgSQL, PL/Perl, PL/Python, PL/Tcl, C, C++, ...
- Triggers (gatilhos)
- Índices: B⁺-tree, Hash, R-Tree, GiST
- Transação e Concorrência
- Chaves estrangeiras (foreign keys)
- Tipos de dados e funções definidos pelo usuário
- hstore (key => value)
- PostGIS
- Full Text Search (FTS)
- Terminal iterativo: psql
- Interface gráfica para administração: pgAdmin
- Replicação (Slony, pgpool-II, Bucardo)

Instalando o Servidor PostgreSQL

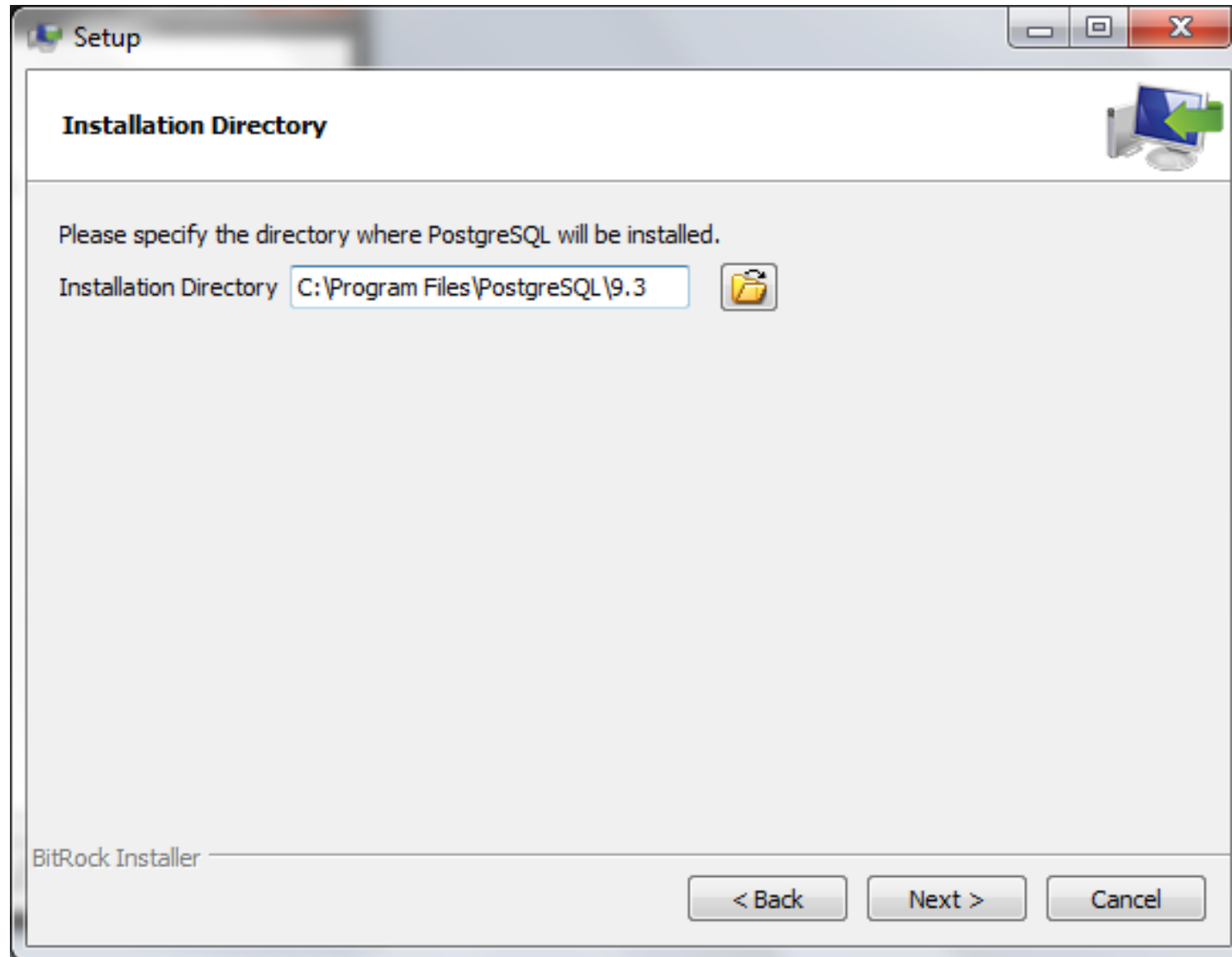
Microsoft Windows 7

PostgreSQL 9.3.0

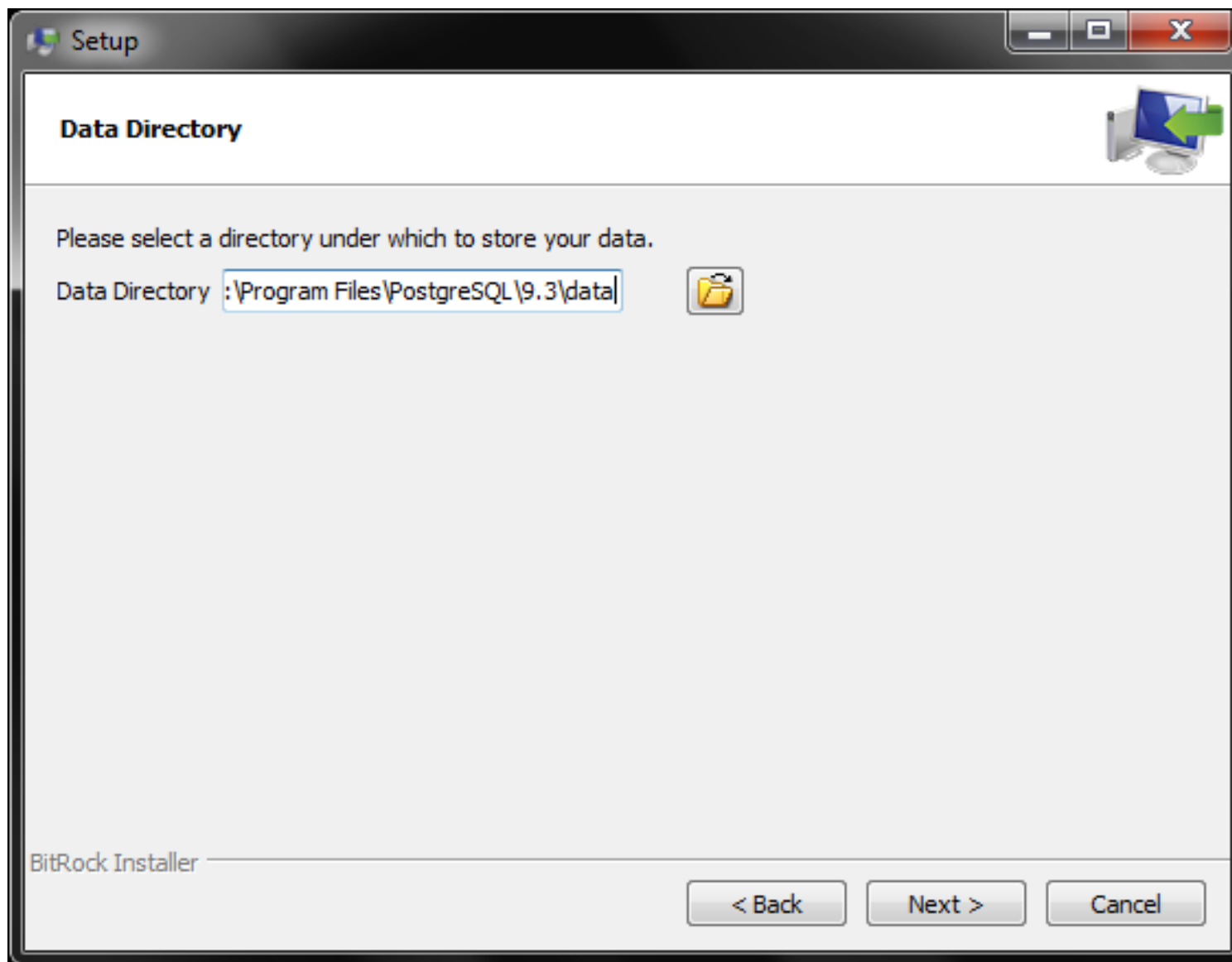
Passo 01: janela de boas vindas do instalador



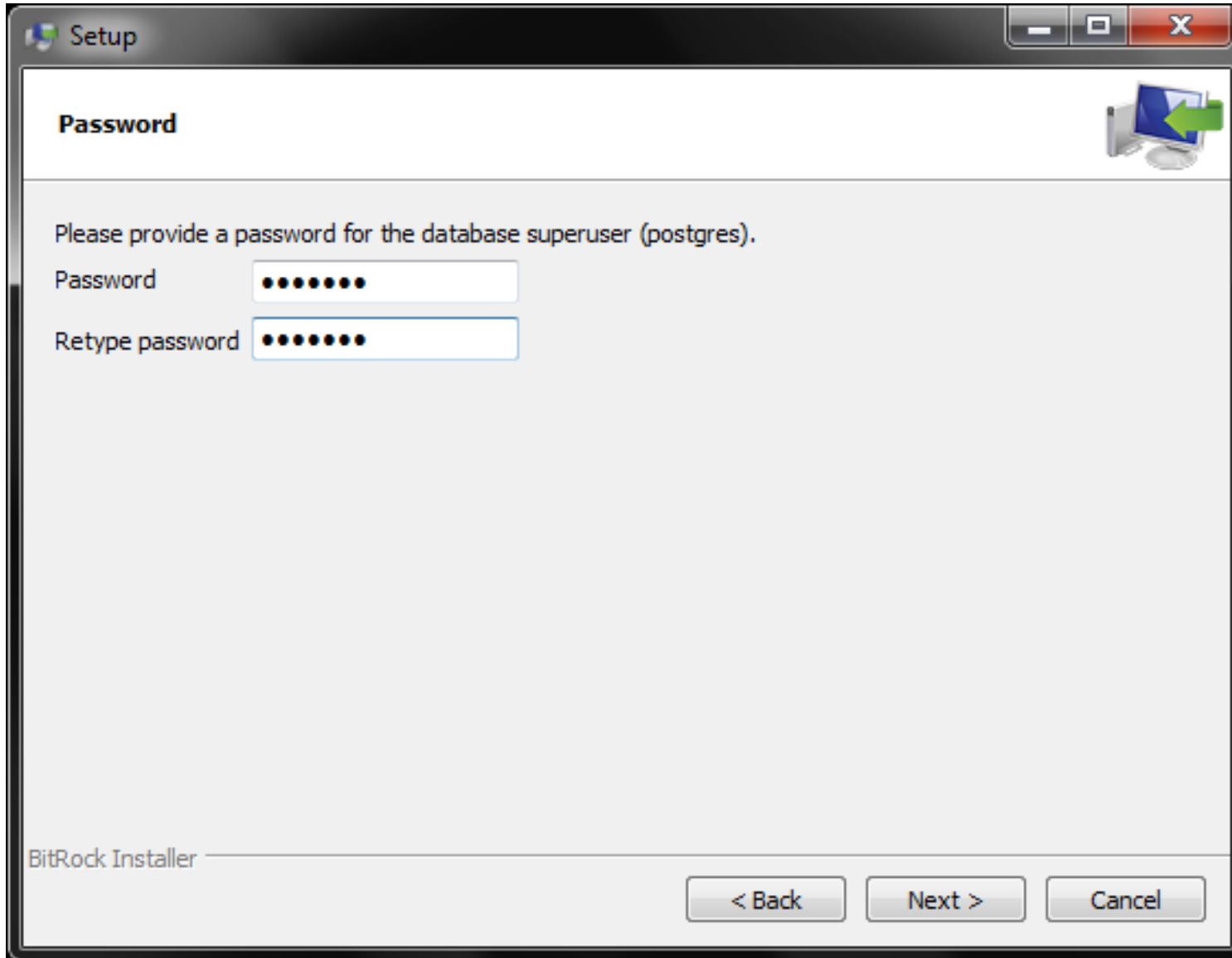
Passo 02: escolhendo a localização da instalação



Passo 03: local dos arquivos de dados



Passo 04: senha do usuário postgres (super-usuário)



Setup

Password

Please provide a password for the database superuser (postgres).

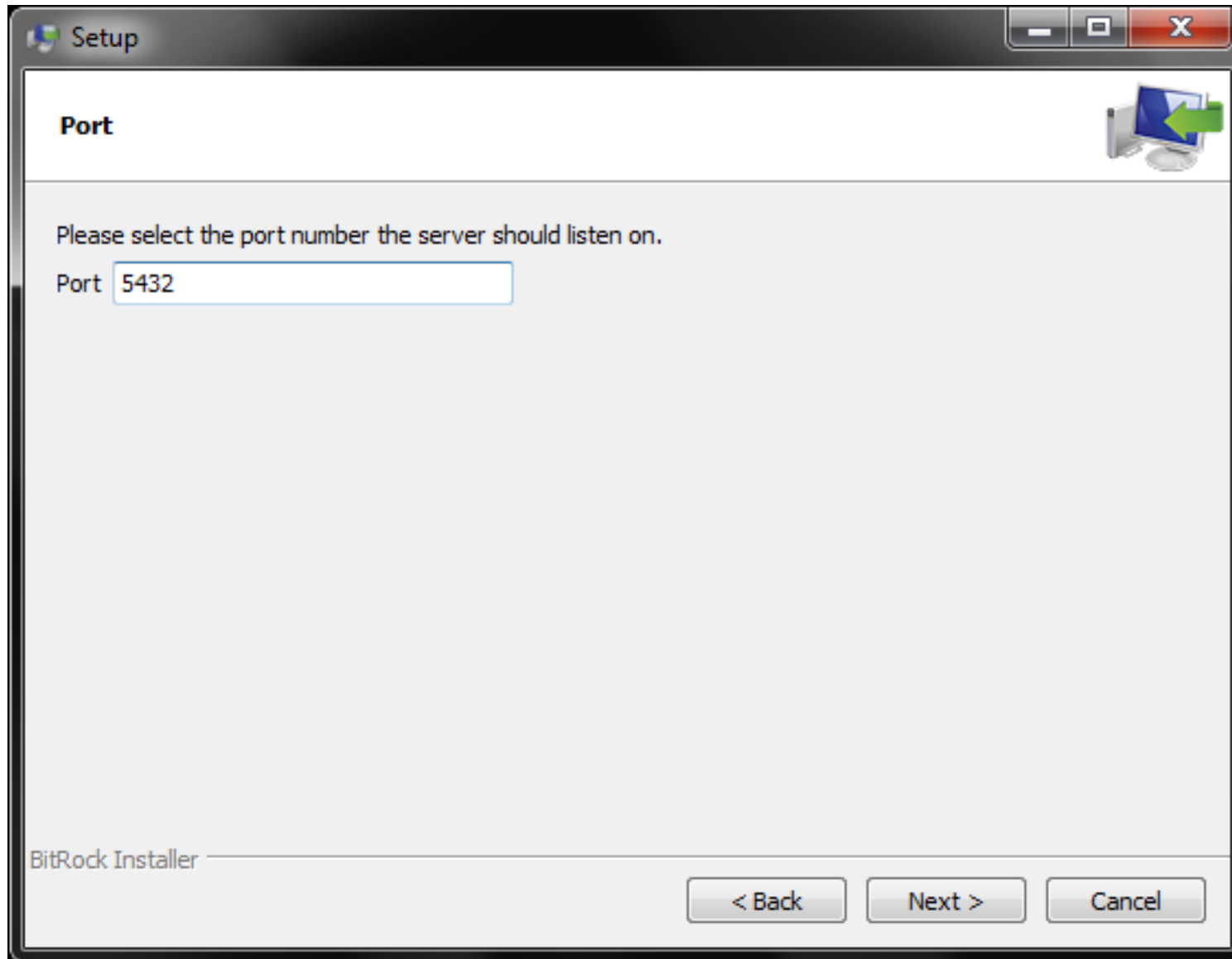
Password

Retype password

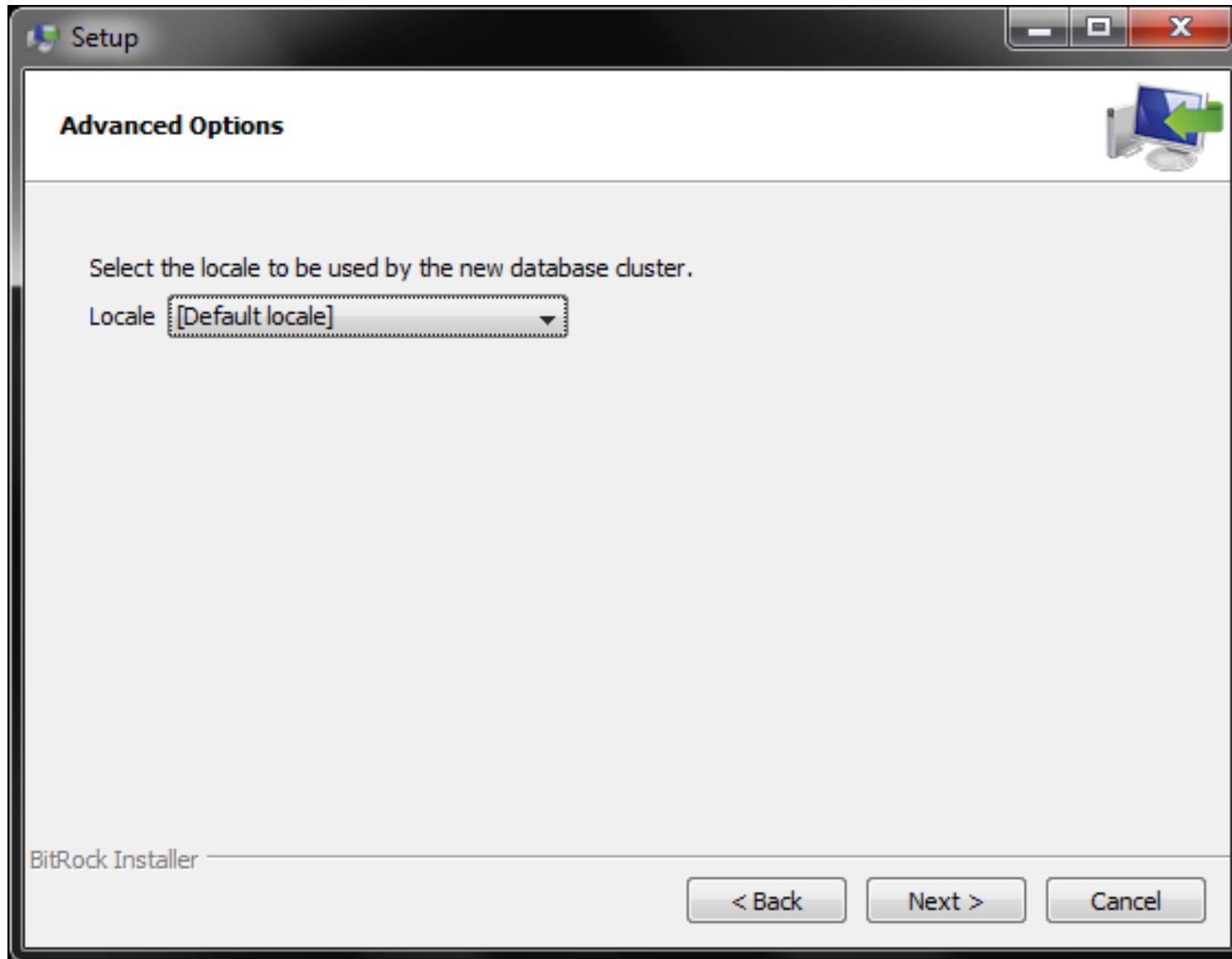
BitRock Installer

< Back Next > Cancel

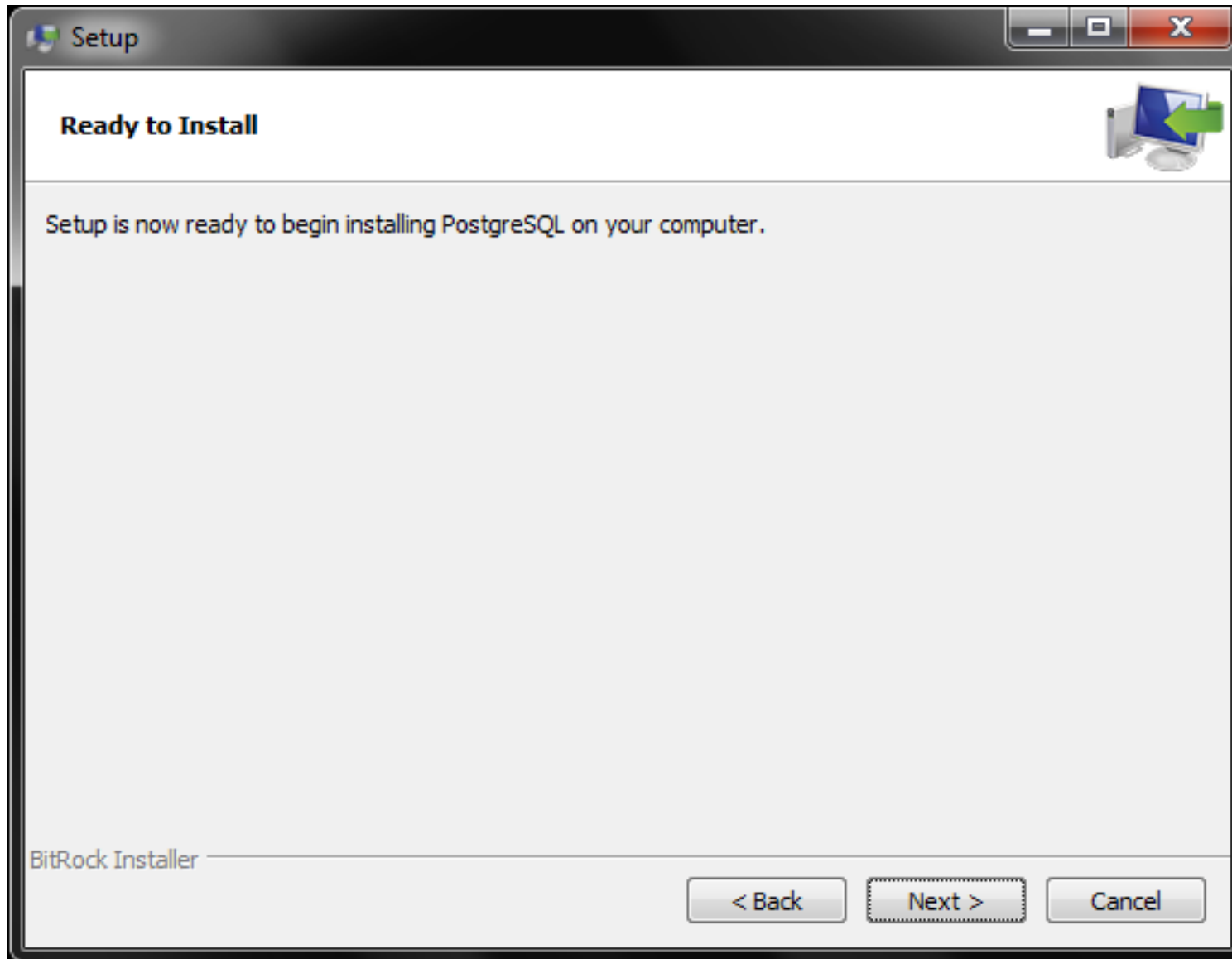
Passo 05: porta de conexão ao servidor



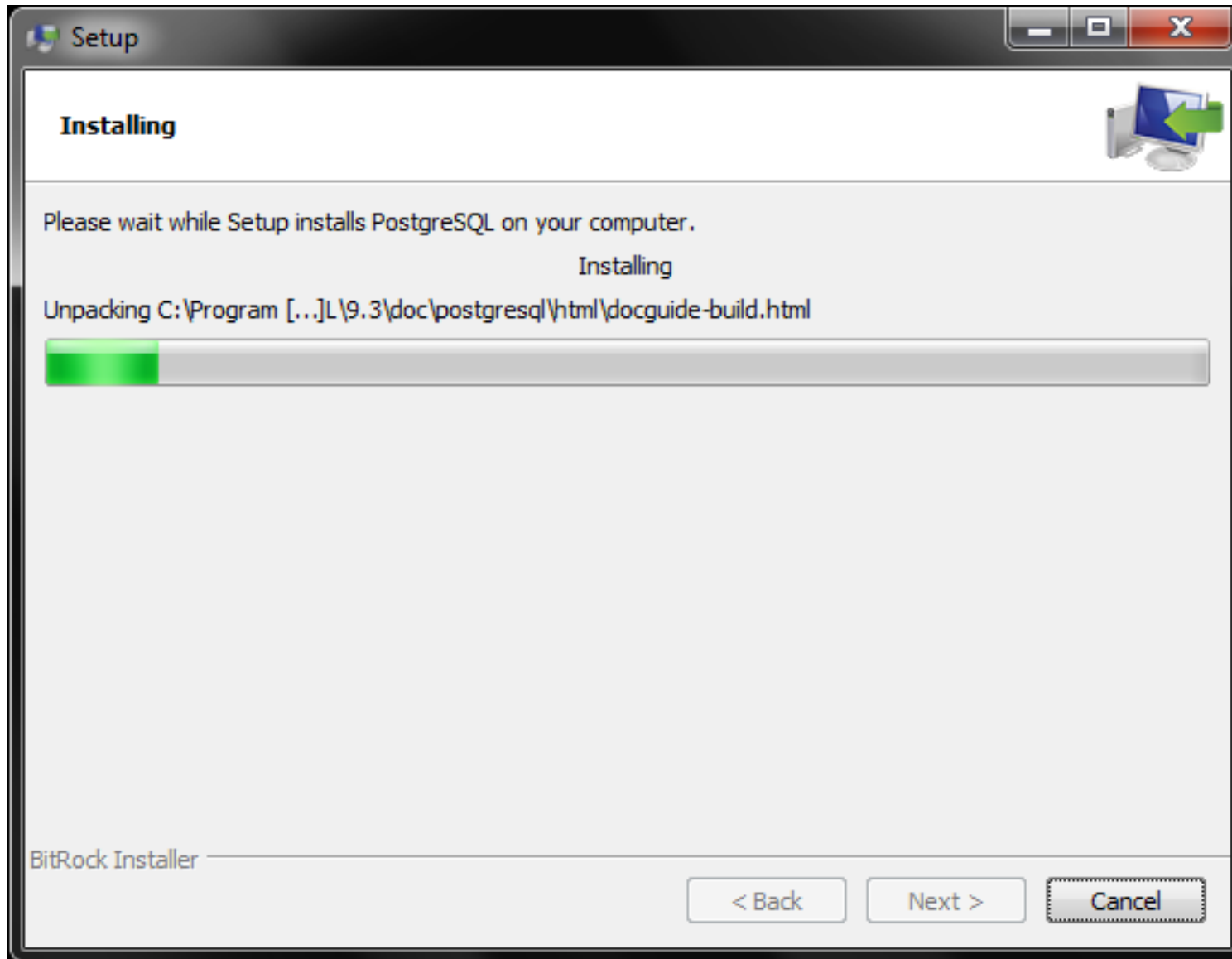
Passo 06: configurações regionais do servidor



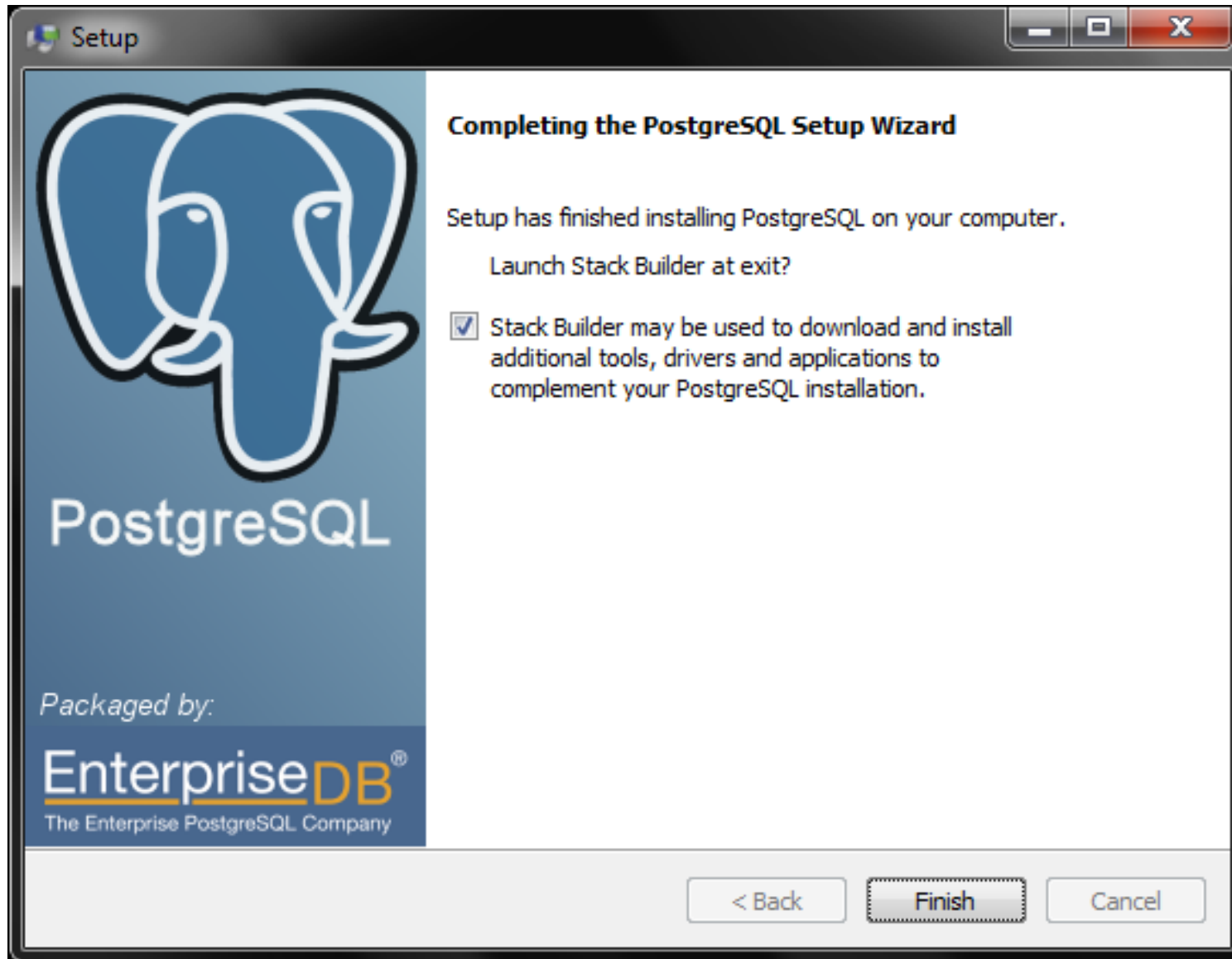
Passo 07: confirmando a configuração



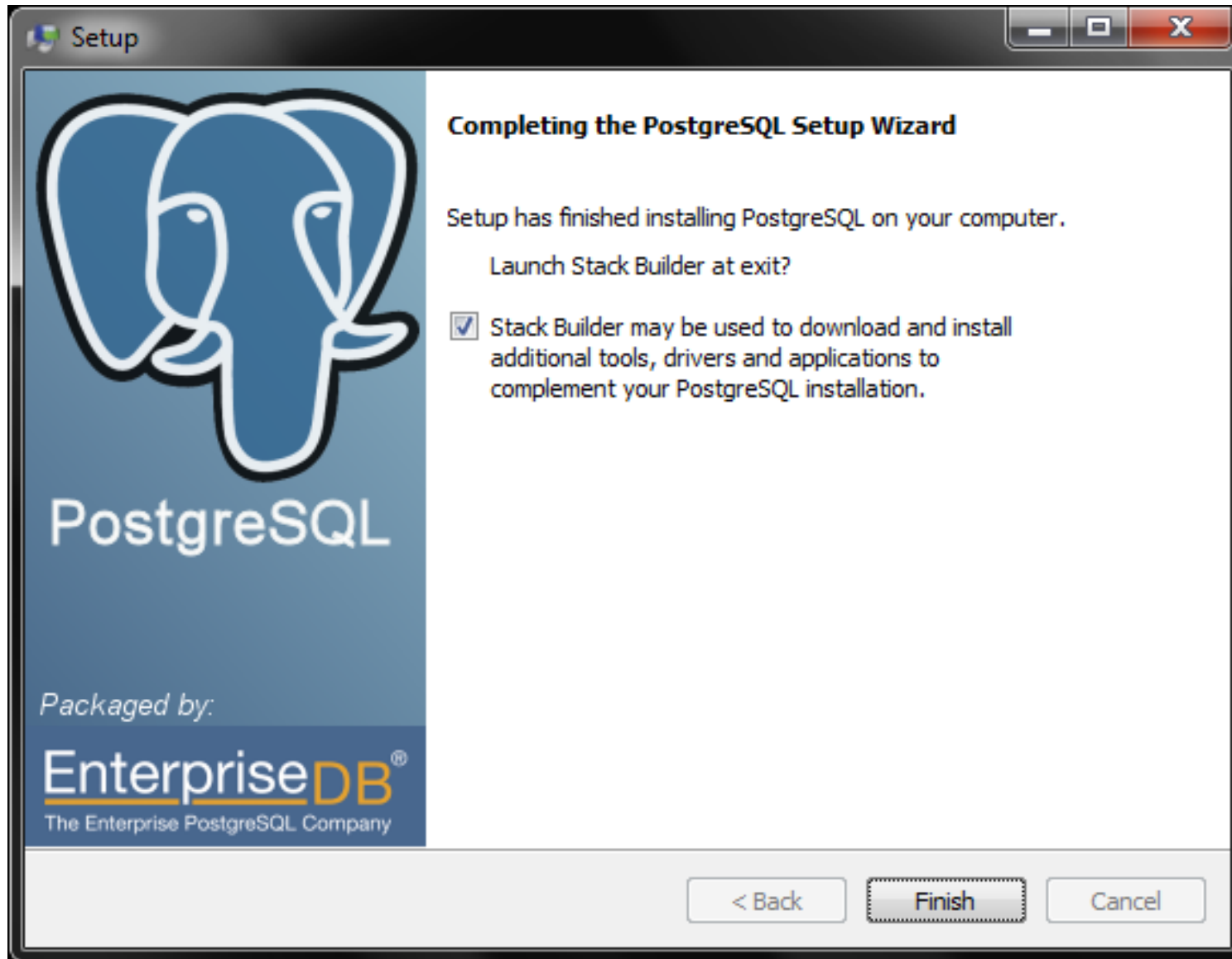
Passo 08: instalação em progresso



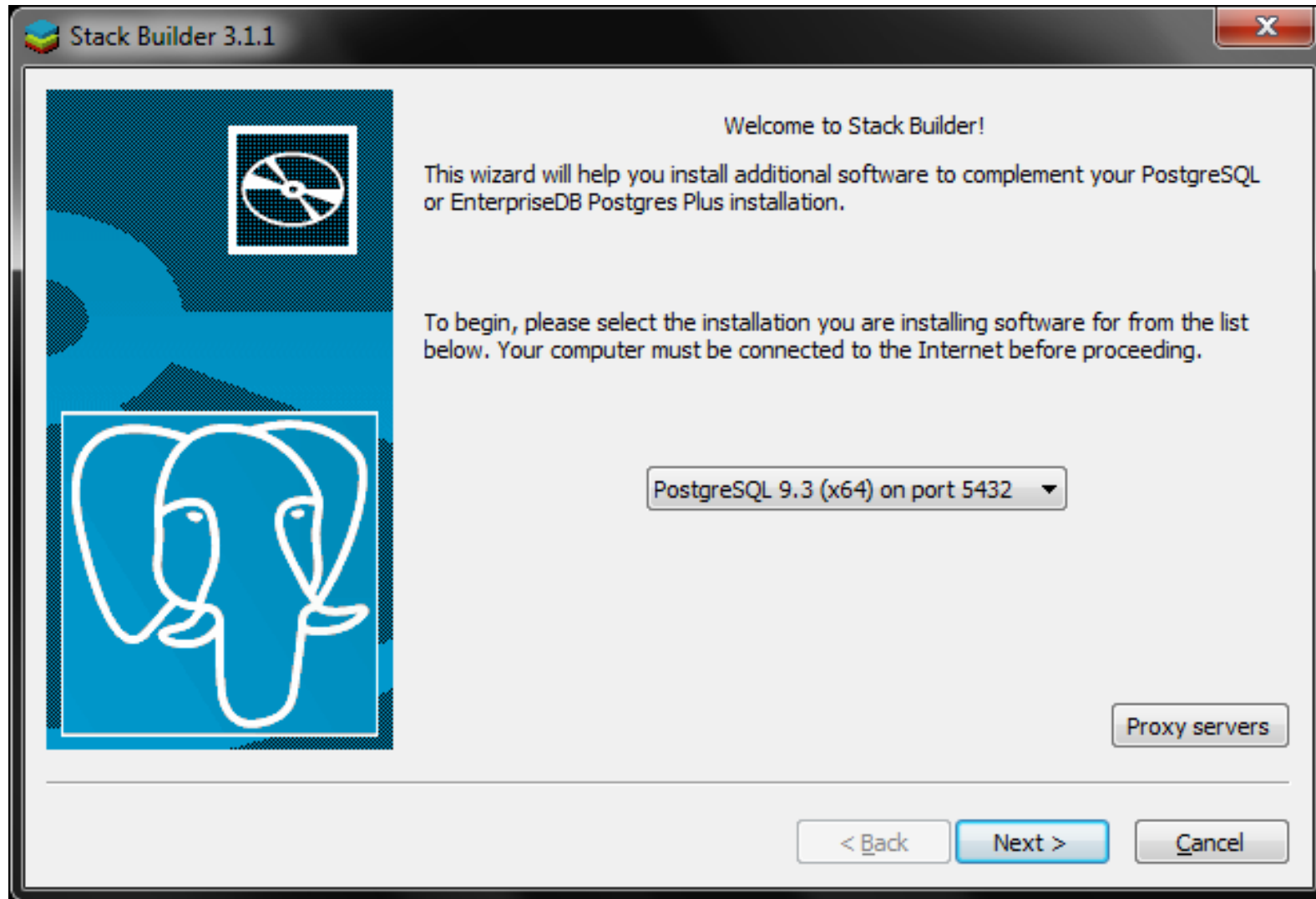
Passo 09: instalação do servidor finalizada



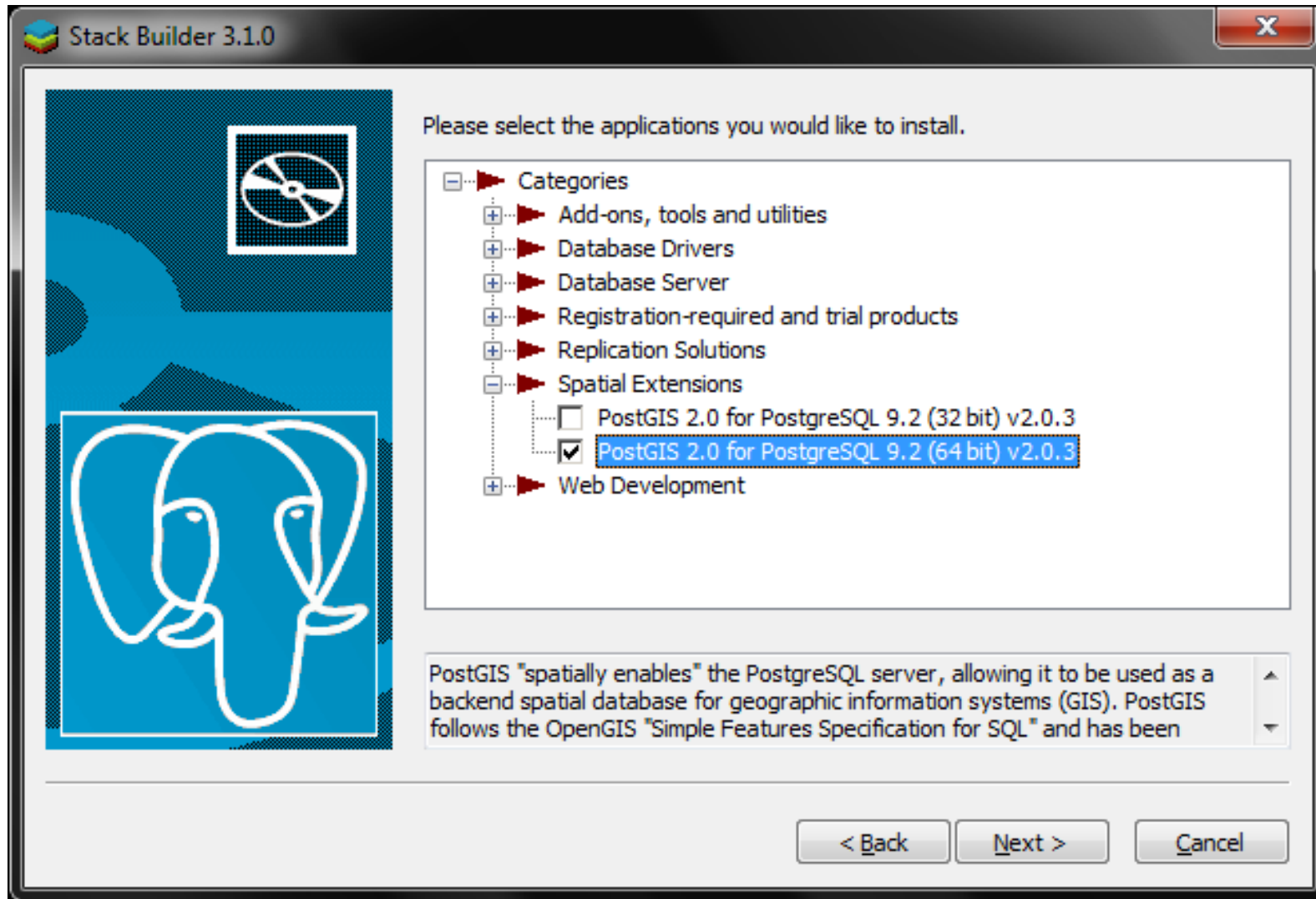
Passo 10: lançar o Stack Builder



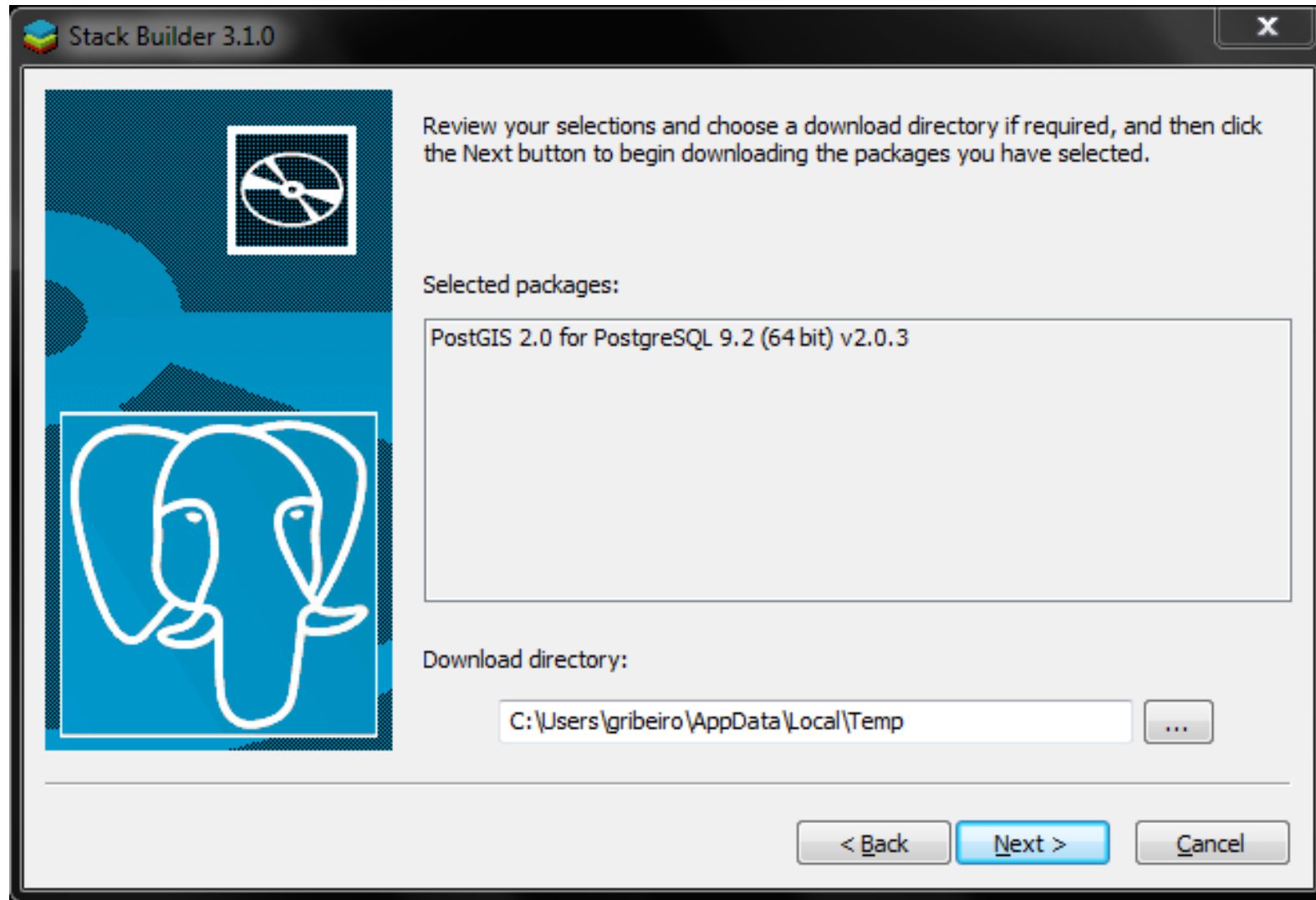
Passo 11: informando o servidor para o qual desejamos adicionar extensões/plugins/módulos



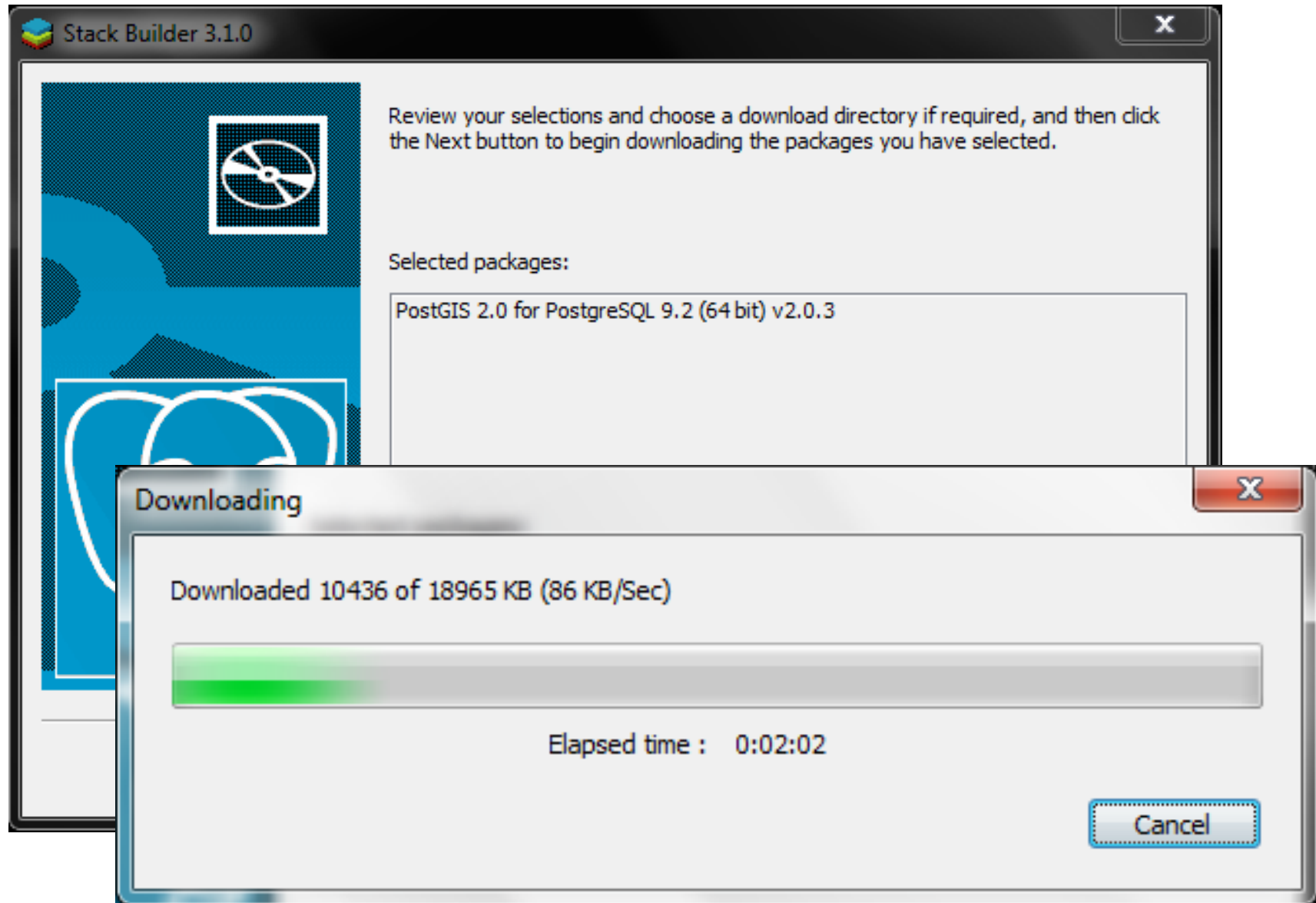
Passo 12: selecionando a extensão PostGIS



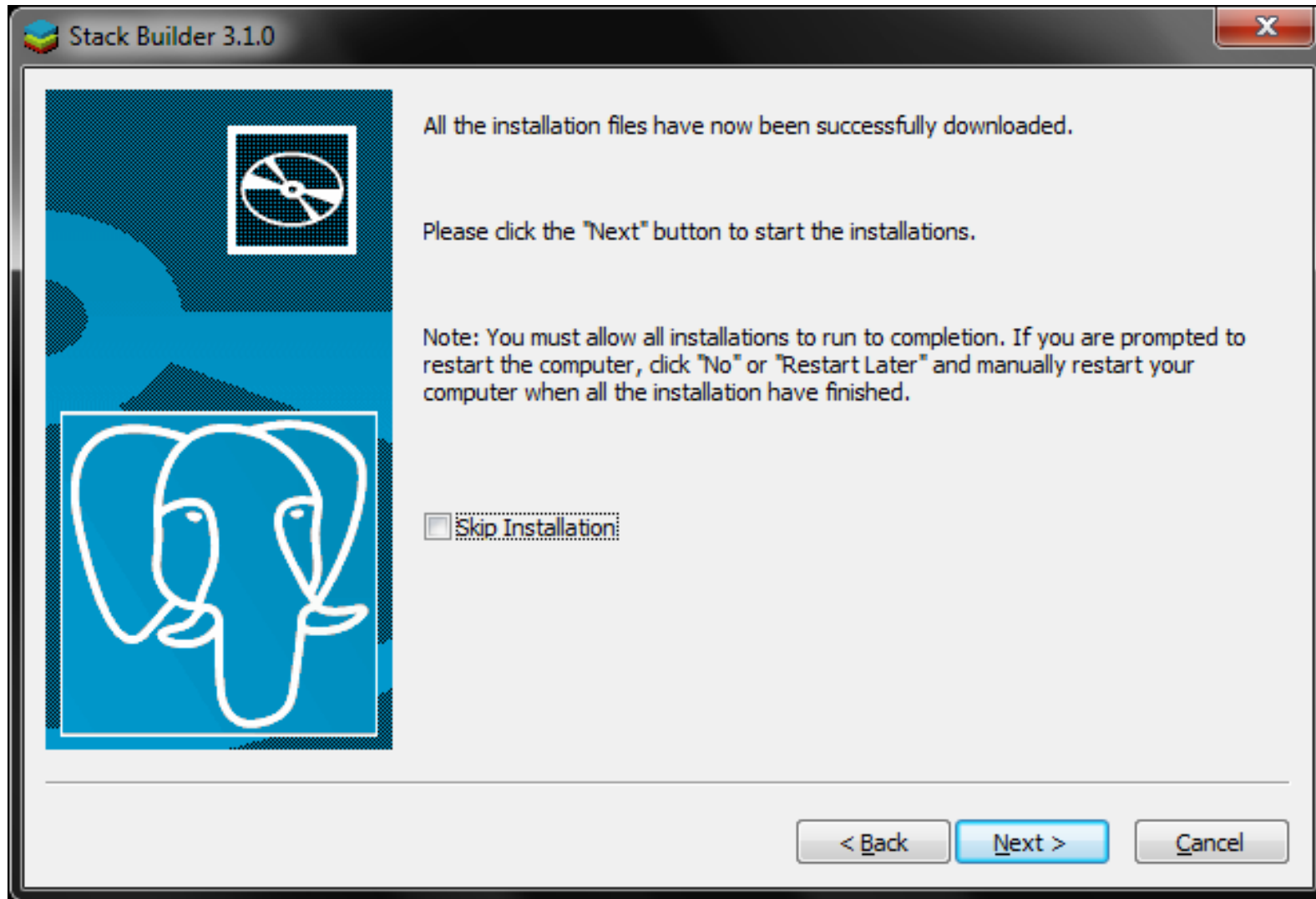
Passo 13: iniciando a instalação do PostGIS



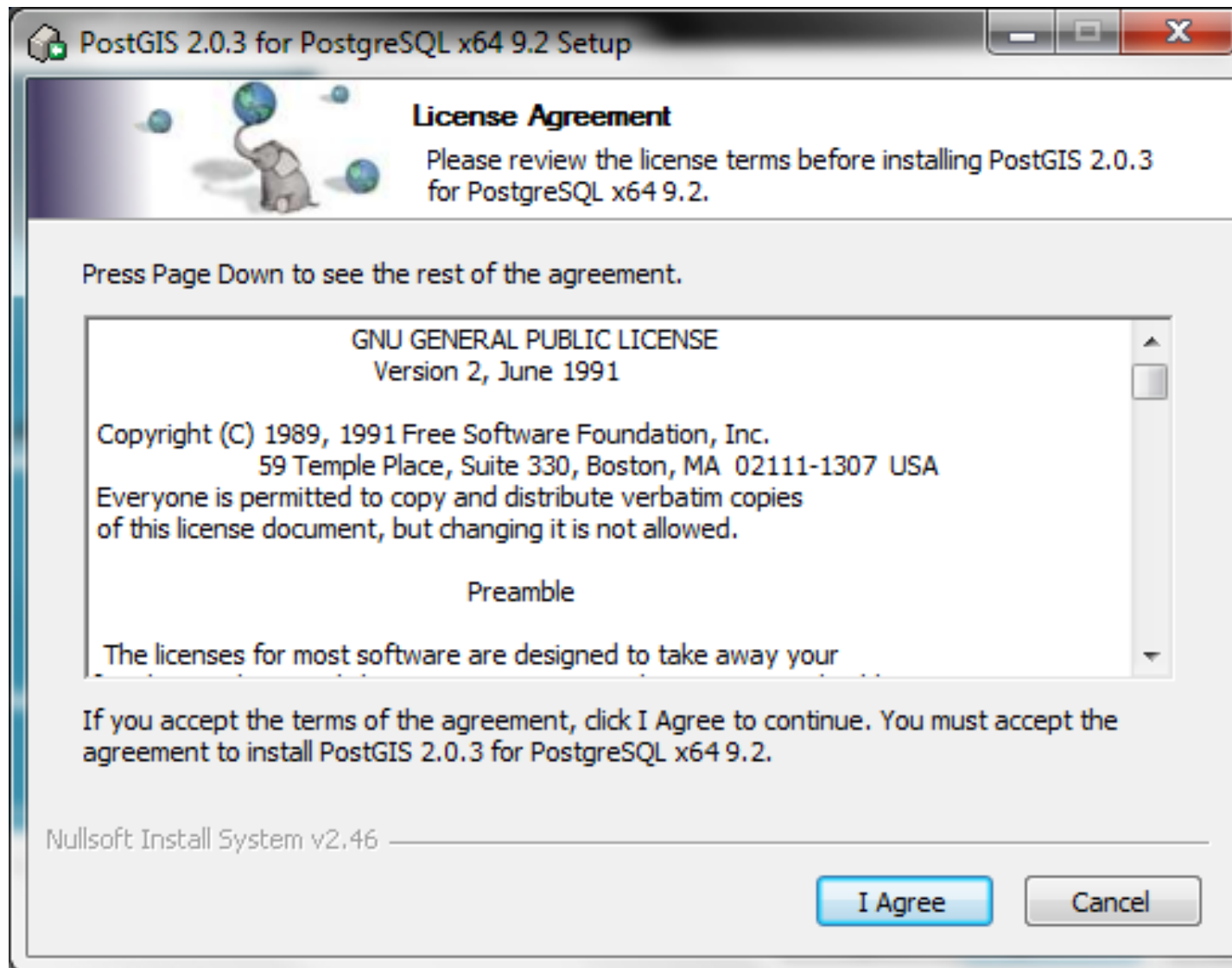
Passo 14: download do PostGIS em progresso



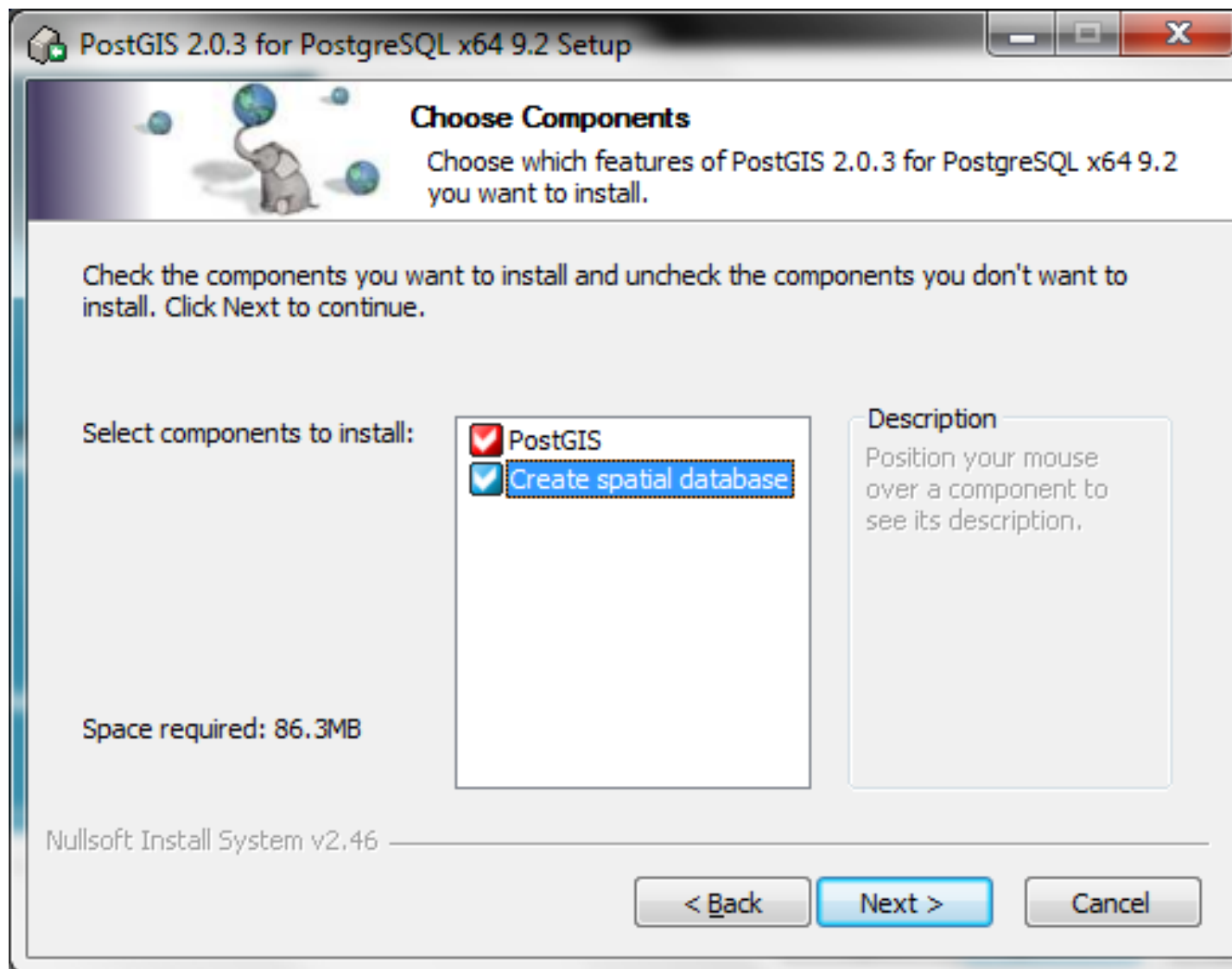
Passo 15: instalar a extensão PostGIS



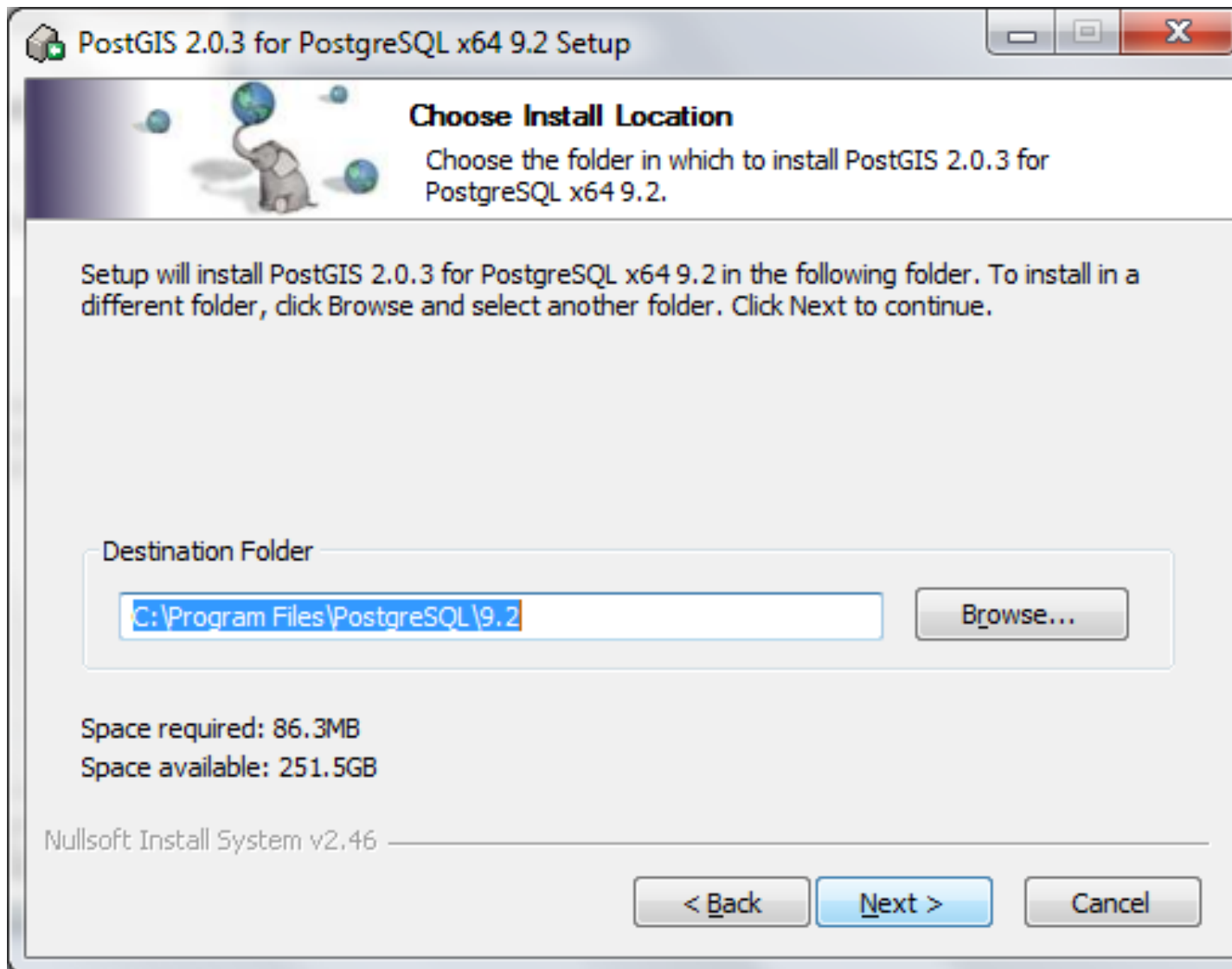
Passo 16: aceitar a licença (GPL-2)



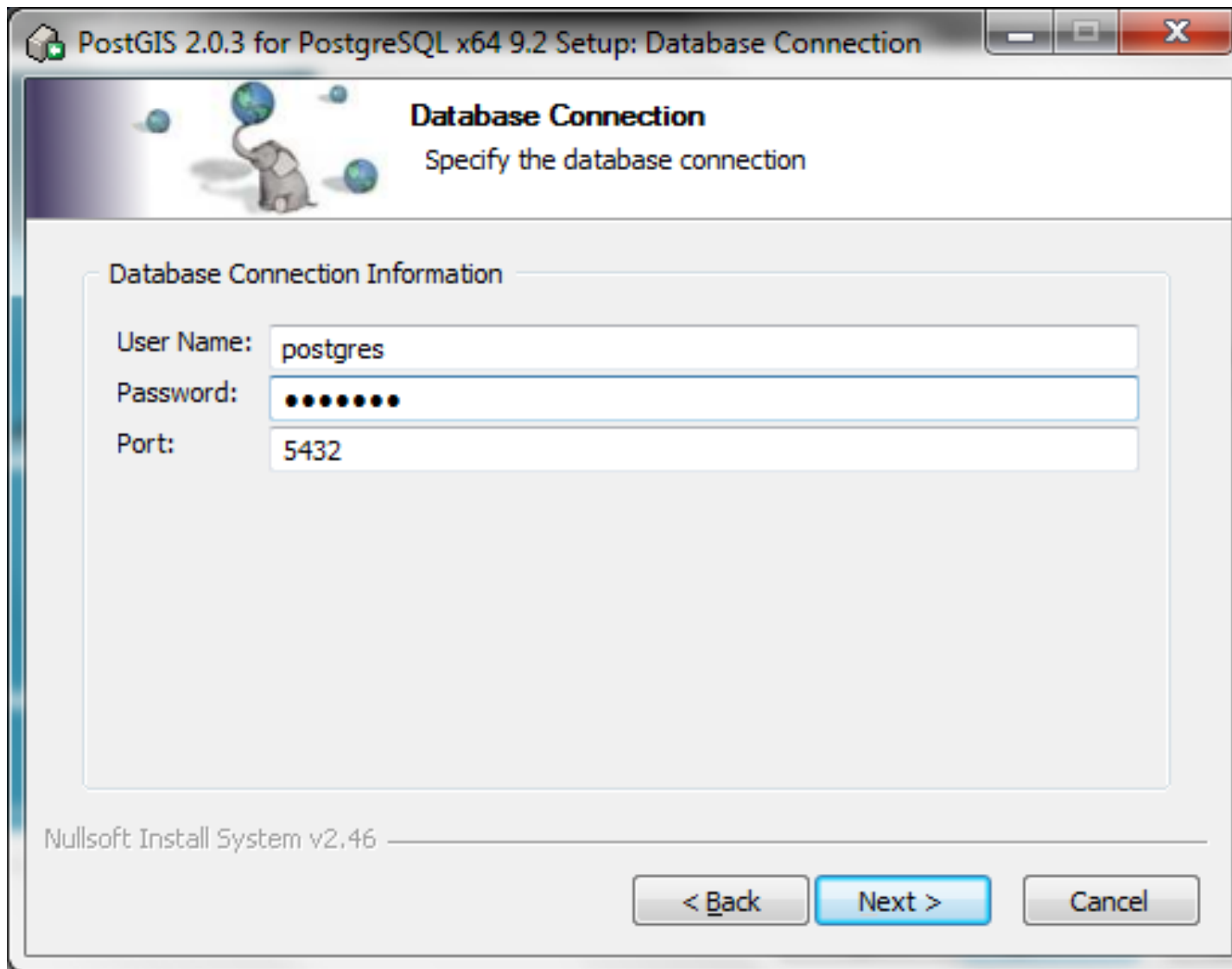
Passo 17: bancos espaciais para molde



Passo 18: selecionar o local da instalação



Passo 19: informar usuário e senha do servidor PostgreSQL para instalação da extensão PostGIS



The image shows a Windows-style dialog box titled "PostGIS 2.0.3 for PostgreSQL x64 9.2 Setup: Database Connection". The dialog has a header area with a globe icon and the text "Database Connection" and "Specify the database connection". Below this is a section titled "Database Connection Information" containing three input fields: "User Name:" with the text "postgres", "Password:" with a masked password of ten dots, and "Port:" with the text "5432". At the bottom of the dialog, there is a footer that reads "Nullsoft Install System v2.46" and three buttons: "< Back", "Next >" (highlighted in blue), and "Cancel".

PostGIS 2.0.3 for PostgreSQL x64 9.2 Setup: Database Connection

Database Connection
Specify the database connection

Database Connection Information

User Name: postgres

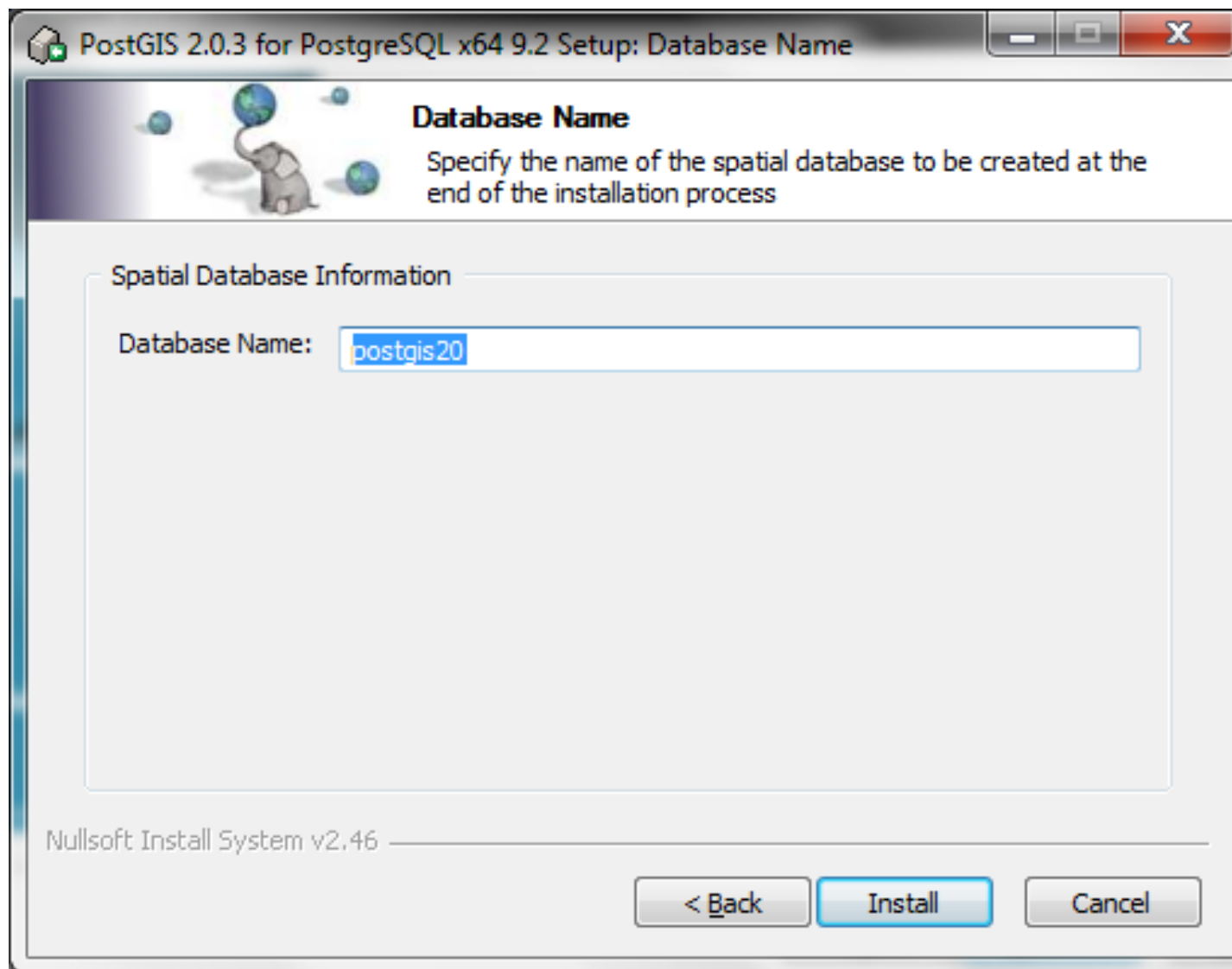
Password: ●●●●●●●●●●

Port: 5432

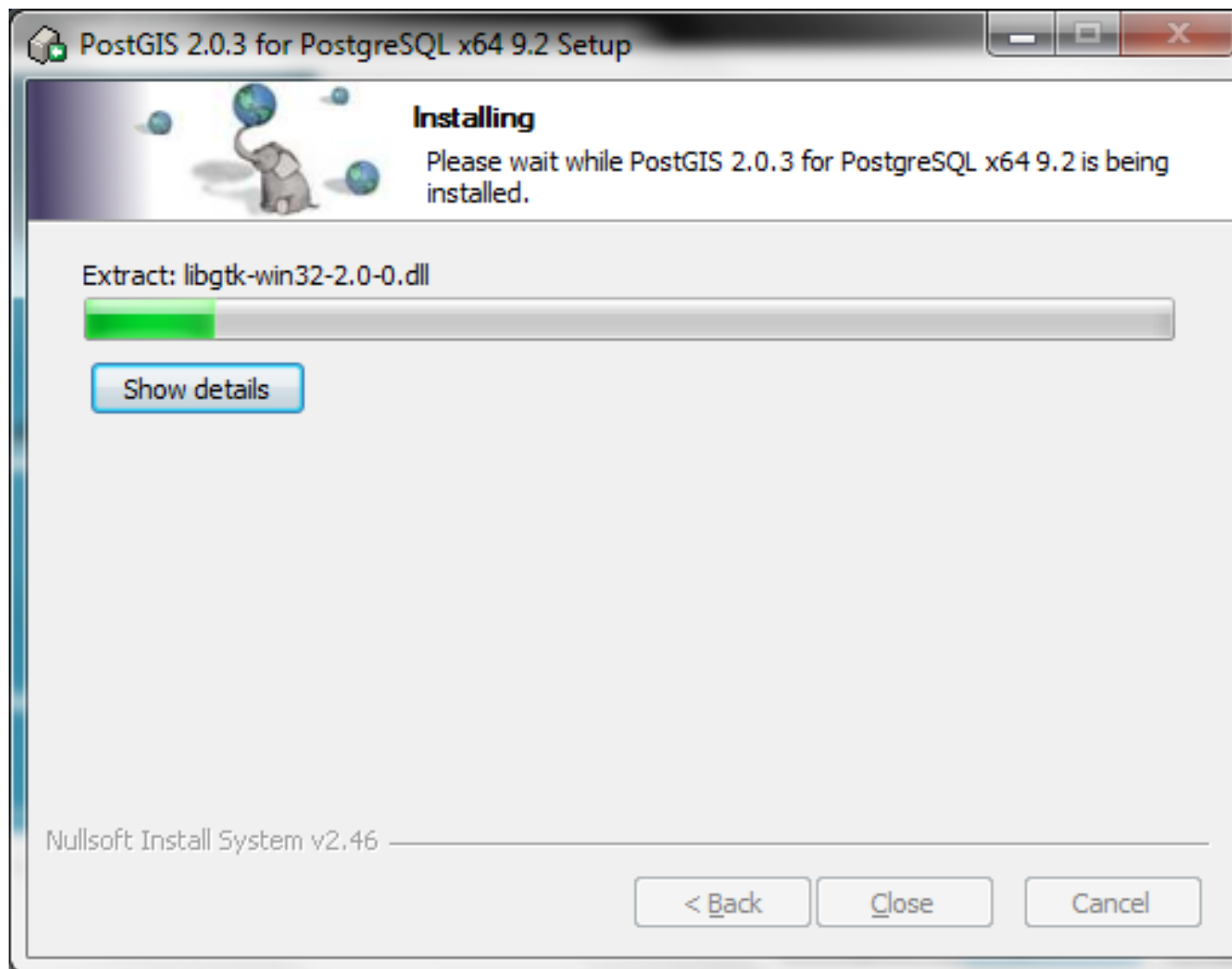
Nullsoft Install System v2.46

< Back Next > Cancel

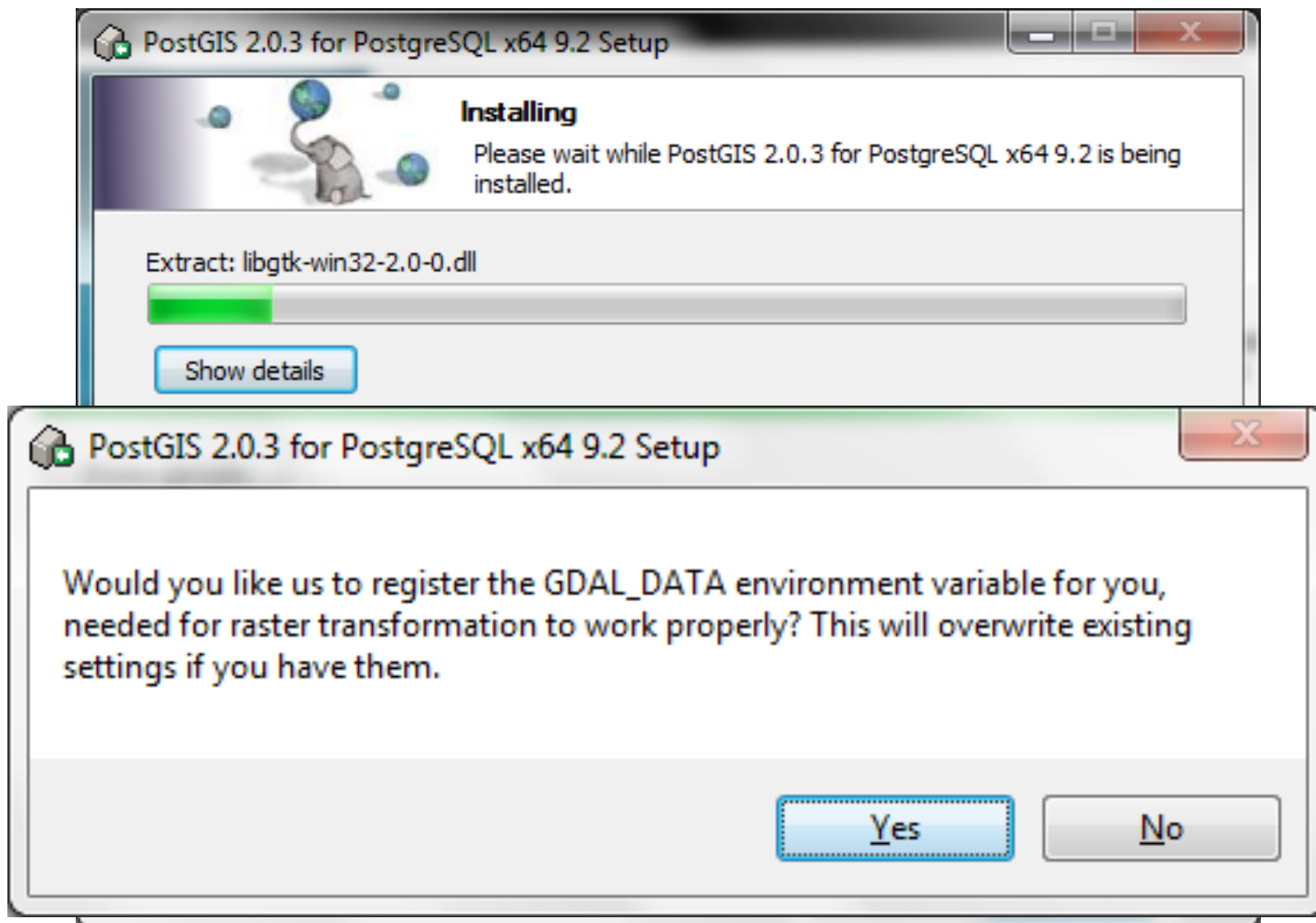
Passo 20: Nome do banco de dados com a extensão espacial habilitada



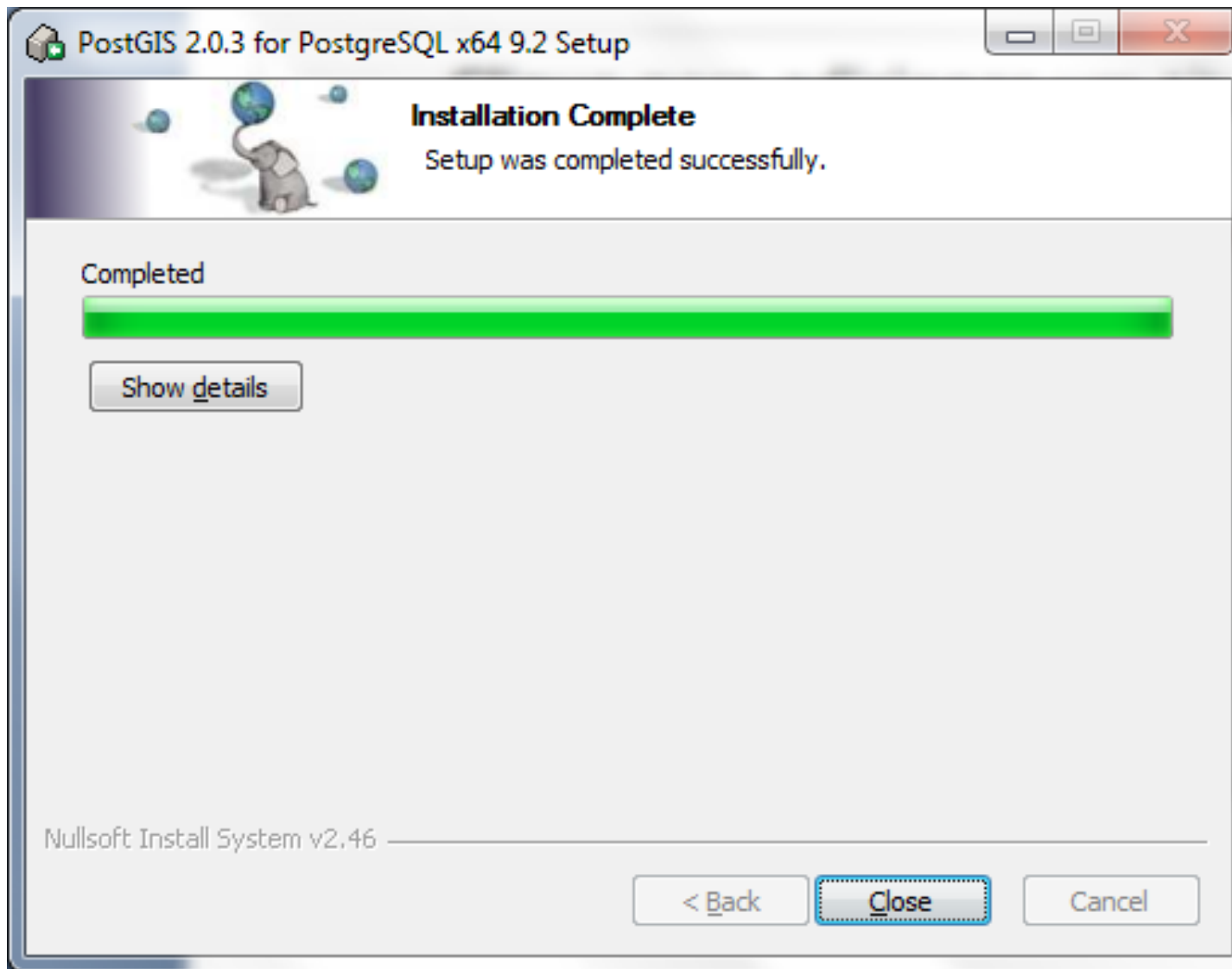
Passo 21: progresso da instalação do PostGIS



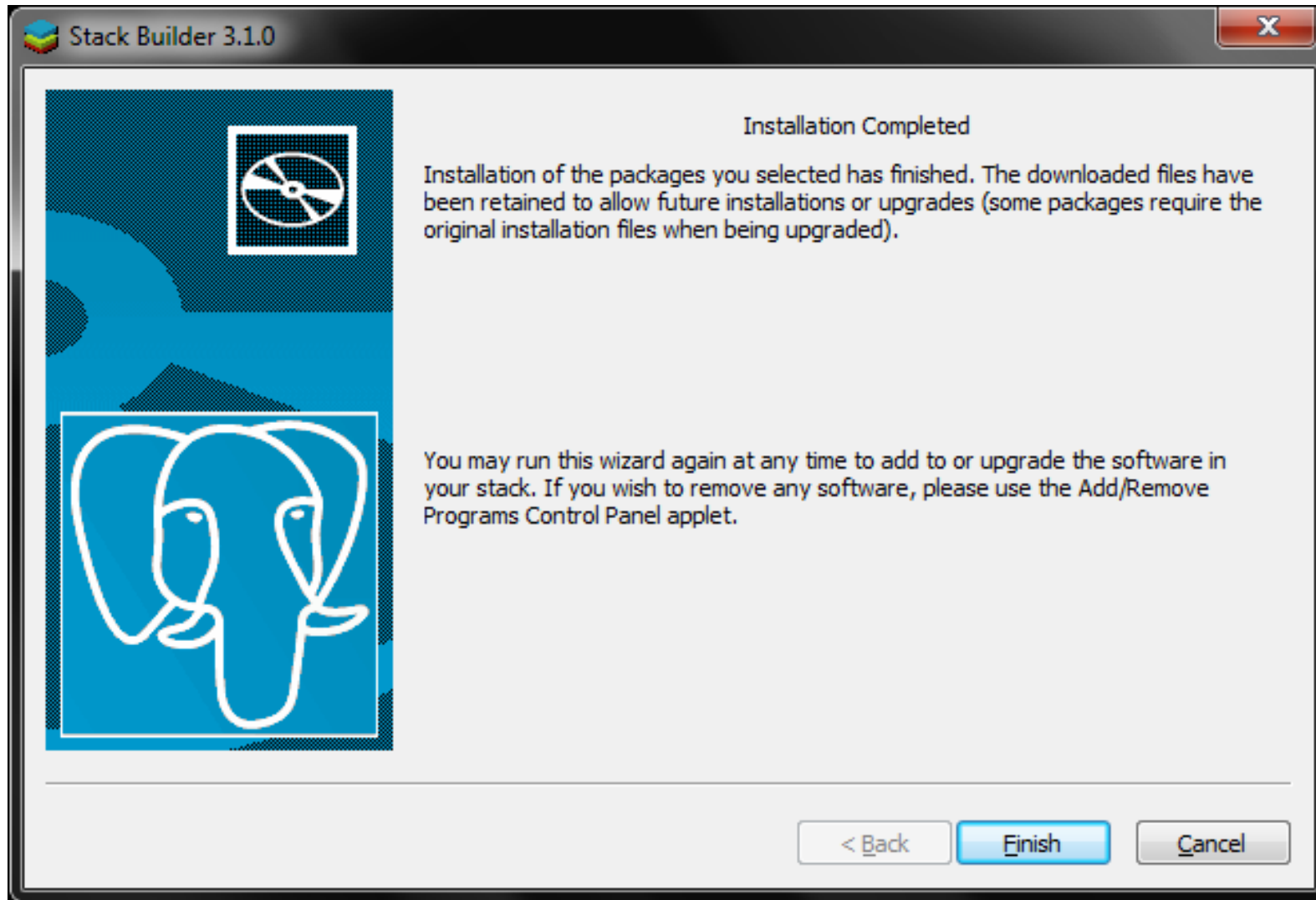
Passo 22: variável de ambiente GDAL_DATA



Passo 23: instalação da extensão PostGIS finalizada



Passo 24: instalação das extensões/plugins/ módulos finalizada



Instalando o Servidor PostgreSQL

Linux

Ubuntu 12.04 LTS

Instalando com auxílio do gerenciador de pacotes

- Abra uma janela do shell (linha de comando).
- Use o comando apt-get para instalar os pacotes:

```
wget --quiet -O - http://apt.postgresql.org/pub/repos/  
  apt/ACCC4CF8.asc | sudo apt-key add -  
sudo apt-get update  
sudo apt-get install postgresql-9.2 pgadmin3
```
- Dica:
 - <http://trac.osgeo.org/postgis/wiki/UsersWikiPostGIS20Ubuntu1204>

Instalando a partir dos fontes

```
CPPFLAGS="-I/usr/local/include" LDFLAGS="-lstdc++  
-L/usr/local/lib" ./configure --with-libxml  
--with-libxslt --with-openssl --with-openssl  
--with-python --prefix=/usr/local/pgsql  
--with-includes=/usr/local --with-libraries=/usr/local
```

```
make
```

```
sudo make install
```

```
sudo adduser postgres
```

```
sudo mkdir /usr/local/pgsql/data
```

```
sudo chown postgres /usr/local/pgsql/data
```

Instalando a partir dos fontes

```
sudo su - postgres
```

```
/usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data
```

```
exit
```

```
sudo cp contrib/start-scripts/linux /etc/init.d/  
postgresql-9.2.3
```

```
editar arquivo e corrigir pasta de instalação do  
servidor postgresql
```

```
sudo chmod ug+x /etc/init.d/postgresql-9.2.3
```

Instalando a partir dos fontes

- adicionar um link na pasta sbin para: /usr/lib/insserv
`sudo ln -s /usr/lib/insserv/insserv /sbin/
insserv`

```
sudo chkconfig --add postgresql-9.2.3
```

```
sudo /etc/init.d/postgresql-9.2.3 start
```

Instalando a partir dos fontes

- Para rodar o servidor sem instalá-lo como serviço:

```
/usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/  
data start
```

- Você terá que instalar o pgAdmin se quiser usar um ambiente gráfico durante o curso.

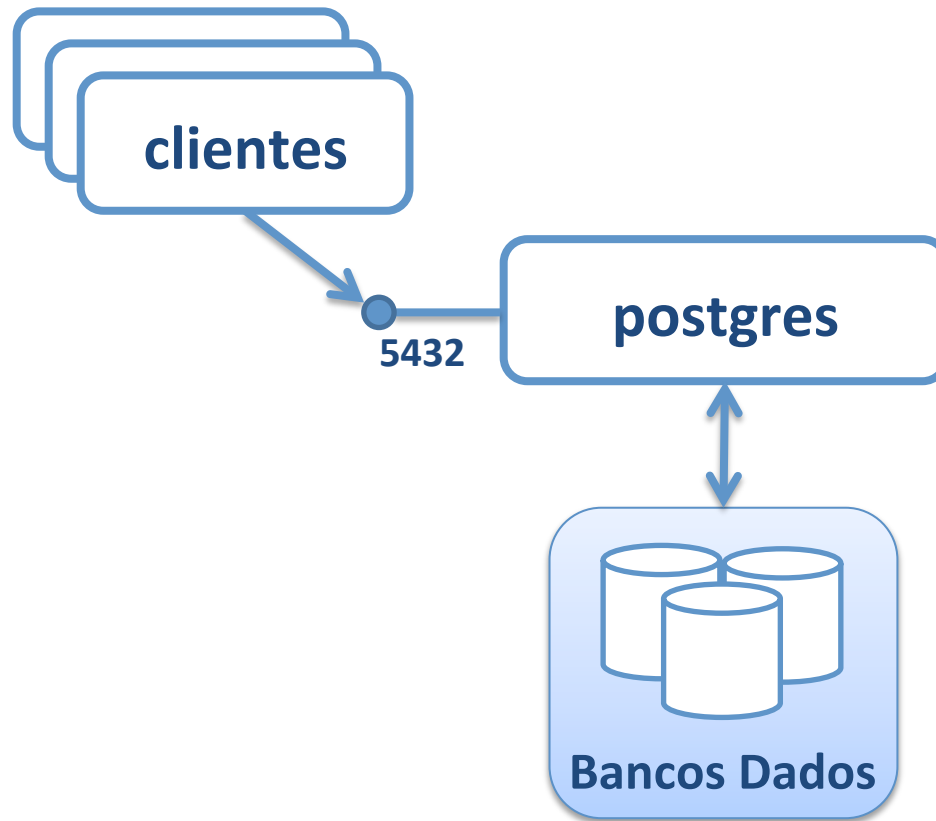
Instalando o Servidor PostgreSQL

Mac OS X (Yosemite)

Postgres.app

- A forma mais simples de se ter uma versão do PostgreSQL rodando no Mac OS X é através da distribuição Postgres.app:
<http://postgresapp.com>
- Faça o download da versão 9.4.1 e arraste o arquivo Postgres para a pasta Applications (Aplicativos).
- Por default a pasta de dados estará localizada em:
`/Users/gribeiro/Library/Application Support/Postgres/var-9.4`
- A pasta onde as ferramentas utilitárias estarão instaladas:
`/Applications/Postgres.app/Contents/Versions/9.4/bin`

Após a instalação temos o serviço rodando e aguardando por conexões

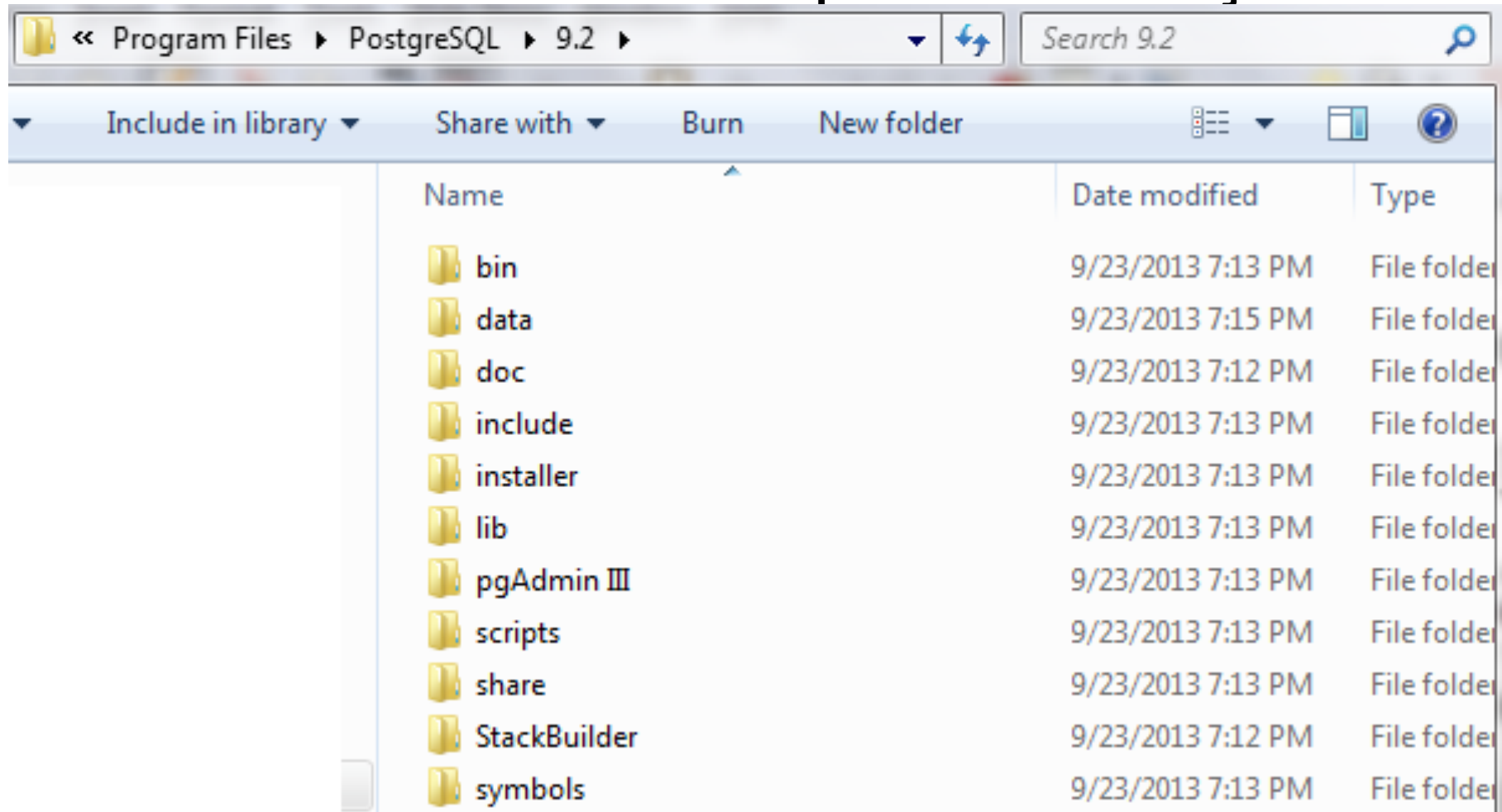


Estrutura da Instalação

Microsoft Windows 7

PostgreSQL 9.2.0

Diretórios criados após a instalação

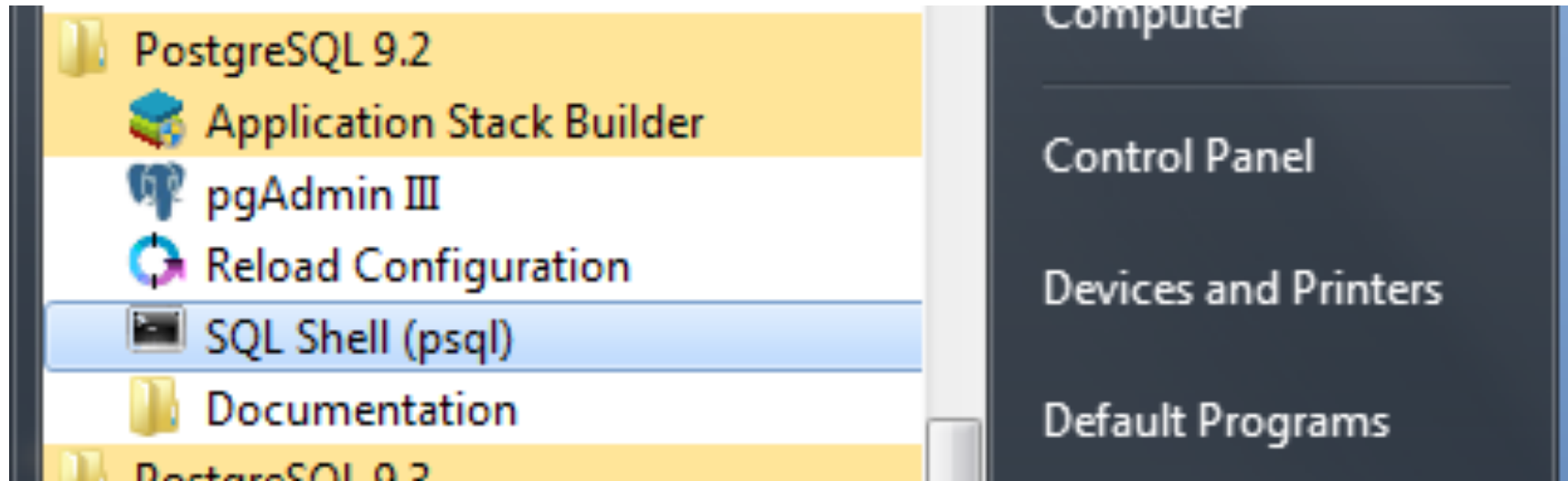


Name	Date modified	Type
bin	9/23/2013 7:13 PM	File folder
data	9/23/2013 7:15 PM	File folder
doc	9/23/2013 7:12 PM	File folder
include	9/23/2013 7:13 PM	File folder
installer	9/23/2013 7:13 PM	File folder
lib	9/23/2013 7:13 PM	File folder
pgAdmin III	9/23/2013 7:13 PM	File folder
scripts	9/23/2013 7:13 PM	File folder
share	9/23/2013 7:13 PM	File folder
StackBuilder	9/23/2013 7:12 PM	File folder
symbols	9/23/2013 7:13 PM	File folder

Nota: a estrutura de diretórios criada no Linux na instalação padrão é muito semelhante a esta

Conectando ao servidor PostgreSQL
através do terminal interativo psql

psql no Windows



```
psql (9.2.3)  
Type "help" for help.
```

```
postgres=#
```

psql no Windows

```
C:\Users\gribeiro>cd "\Program Files\PostgreSQL\9.2\bin"
```

```
C:\Program Files\PostgreSQL\9.2\bin>chcp 1252
```

```
C:\Program Files\PostgreSQL\9.2\bin>psql -U postgres
```

```
psql (9.2.3)  
Type "help" for help.
```

```
postgres=#
```

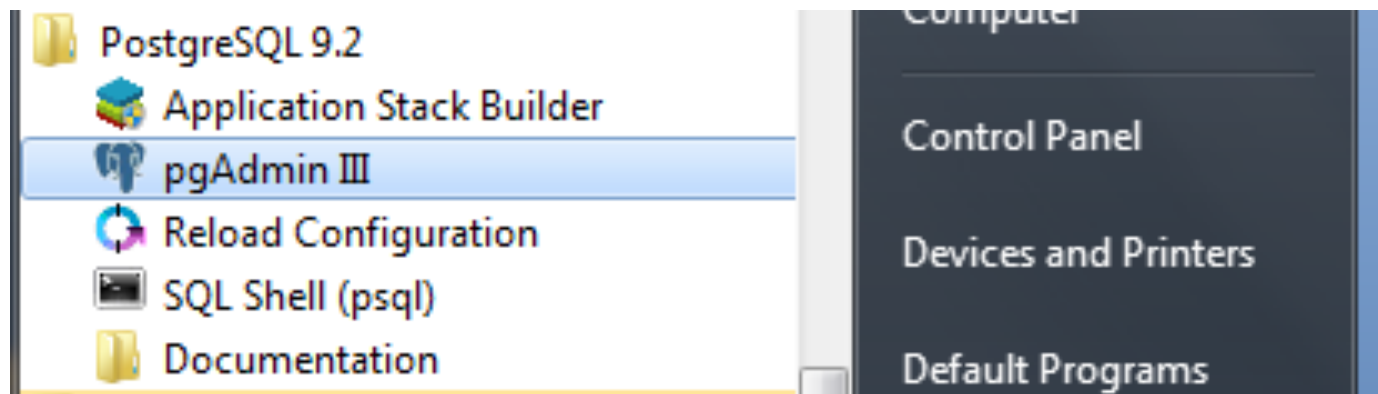
psql no Linux

```
/usr/local/pgsql/bin/psql -U postgres
```

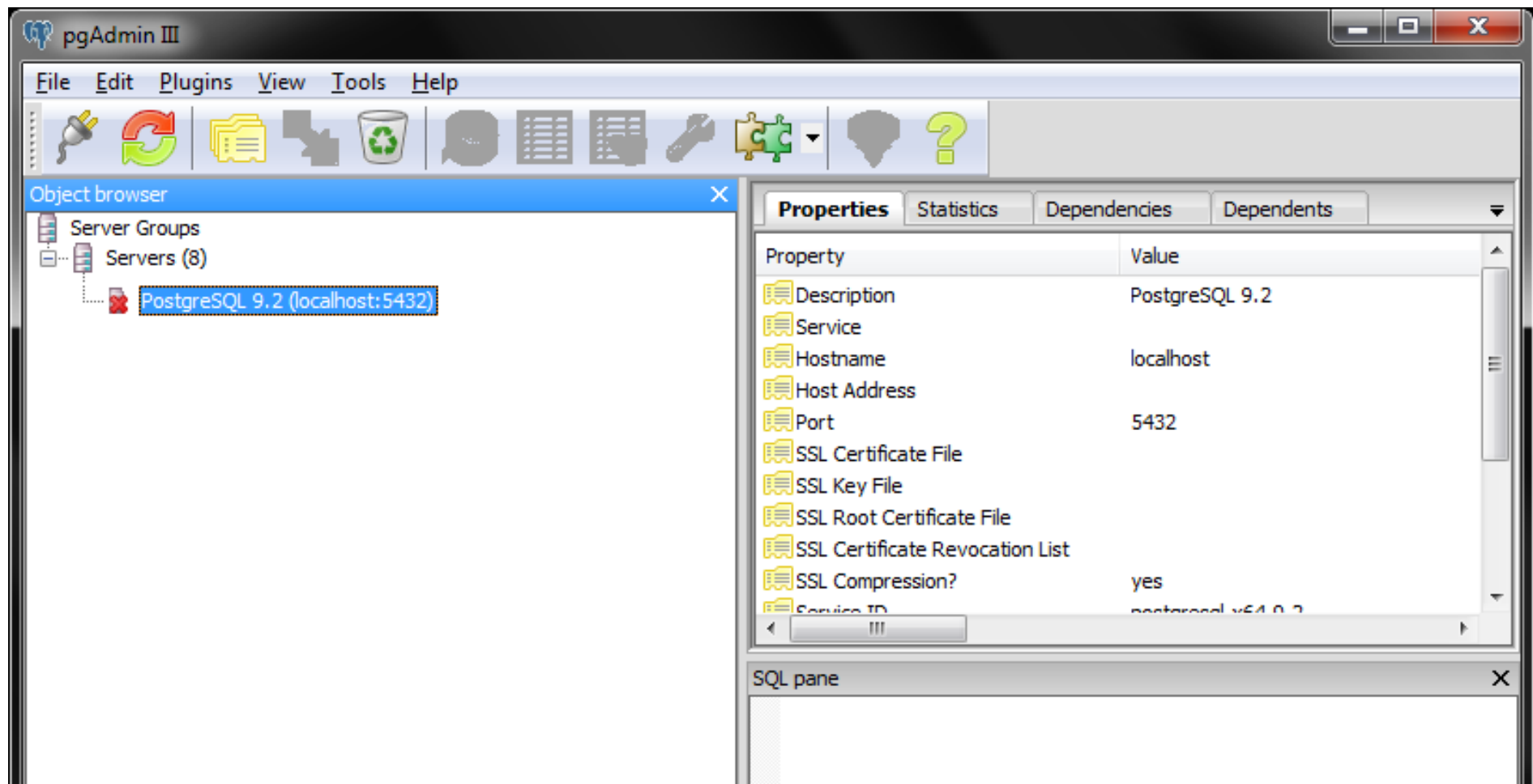
```
psql (9.2.3)  
Type "help" for help.
```

```
postgres=#
```

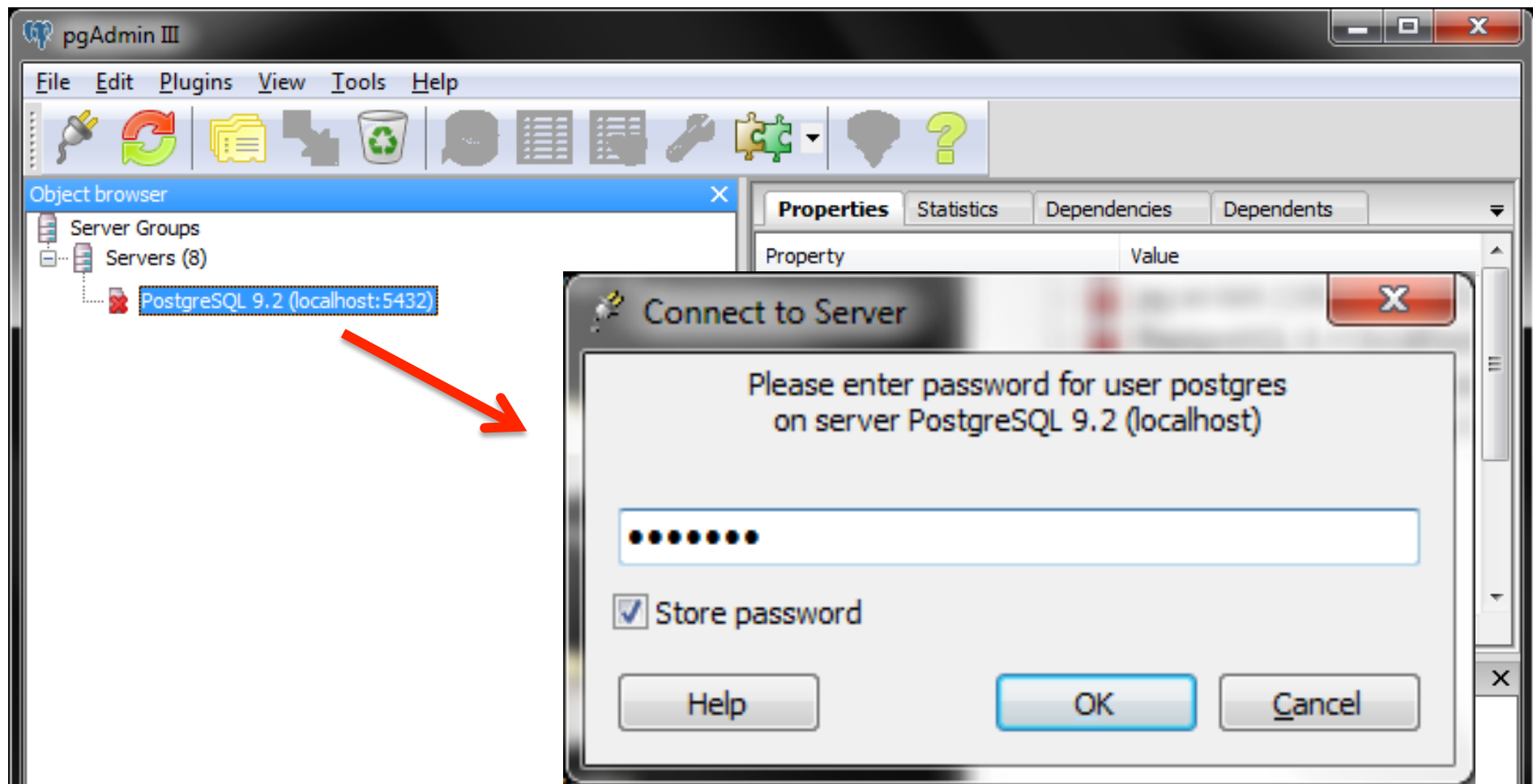
Conectando ao servidor PostgreSQL através do pgAdmin



Janela principal do pgAdmin



Janela principal do pgAdmin



Criando um banco de dados para o curso:
bdgcurso

O que é o template de um banco de dados?

Criando um banco de dados

- Durante a instalação do PostgreSQL é executado um programa chamado *initdb* que realiza a criação de dois bancos:
 - primeiro banco: postgres
 - segundo banco: template1
- Para criar um novo banco de dados precisamos:
 - Estar conectado a algum banco. Em geral: postgres.
 - Usar um banco de molde para realizar um clone. Em geral usamos: template1.
 - No caso da criação de bancos com suporte espacial, em geral, usamos como molde: template_postgis_20.
- Comando:
CREATE DATABASE bdgcurso **TEMPLATE** template1;

Criando Tabelas

CREATE TABLE

DDL: Criação Tabelas

```
CREATE TABLE <table_name>  
( <column_name> <column_type>  
    [<attribute_constraint>]  
    [, <column_name> <column_type>  
        [<attribute_constraint>] ]  
  
    [<table_constraint> [, <table_constraint> ]]  
);
```

Criar a tabela curso

```
CREATE TABLE curso
(
    curso_id          SERIAL,
    titulo            VARCHAR(30) NOT NULL,
    carga_horaria    INTEGER      NOT NULL,
    PRIMARY KEY(course_id)
);
```

Criar a tabela aluno

```
CREATE TABLE aluno
(
    matricula    SERIAL,
    nome         VARCHAR(40) NOT NULL,
    curso_id     INTEGER      NOT NULL,
    PRIMARY KEY(matricula),
    FOREIGN KEY (curso_id)
                REFERENCES curso(curso_id)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);
```

Inserindo Dados

INSERT INTO

Inserção Dados

```
INSERT INTO nome-tabela (campo1, campo2, ...)  
  VALUES (v1, v2, ...);
```

- Exemplos:

```
INSERT INTO curso (titulo, carga_horaria)  
  VALUES ('informatica', 4);
```

```
INSERT INTO curso (titulo, carga_horaria)  
  VALUES ('matematica', 6);
```

```
INSERT INTO curso (titulo, carga_horaria)  
  VALUES ('quimica', 2);
```

```
INSERT INTO curso (titulo, carga_horaria)  
  VALUES ('biologia', 3);
```


Inserção Dados

- Exemplos:

```
INSERT INTO aluno (nome, curso_id)
VALUES ('gilberto', 1);
```

```
INSERT INTO aluno (nome, curso_id)
VALUES ('edson', 2);
```

```
INSERT INTO aluno (nome, curso_id)
VALUES ('eduardo', 3);
```

```
INSERT INTO aluno (nome, curso_id)
VALUES ('cassia', 4);
```

Consultas

Expressões

Restrições

Junção

Funções de Agregação

DML: recuperando datos

```
SELECT [DISTINCT] ( * |  
(<column_name> | <function> (( [DISTINCT]  
  <column_name> | * )))  
  {,( <column_name> |<function> (( [DISTINCT]  
    <column_name> | * )))}
```

FROM (<table_name> {<alias>} | <jointed_table>)
 {,(<table_name> {<alias>} | <jointed_table>) }

[**WHERE** <condition>]
[**GROUP BY** <column_name> {, <column_name>}
 [**HAVING** <group_selection_condition>]]
[**ORDER BY** <column_name> [(**ASC** | **DESC**)]
 {, <column_name> [(**ASC** | **DESC**)] }]

Exemplos

- Recuperar os dados do aluno gilberto:

```
SELECT * FROM aluno WHERE nome = 'gilberto'
```

- Qual o nome da disciplina do aluno gilberto?

```
SELECT * FROM aluno, curso WHERE nome = 'gilberto' AND  
aluno.curso_id = curso.curso_id;
```

ou

```
SELECT * FROM aluno INNER JOIN curso ON  
aluno.curso_id = curso.curso_id;
```

- Quantos cursos são oferecidos?

```
SELECT COUNT(*) FROM curso;
```

Operadores de Agregação

- Podem ser aplicados para todos os registros de uma coluna ou para grupos de registros usando a cláusula GROUP BY:
 - AVG(): média dos valores da coluna
 - SUM(): soma dos valores da coluna
 - COUNT(): número de valores na coluna
 - MAX(): maior valor na coluna
 - MIN(): menor valor na coluna

Atualizando Dados

UPDATE

DML: UPDATE

```
UPDATE curso SET titulo = 'estadística'  
WHERE titulo = 'matemática';
```

Criando Views

CREATE VIEW

Criando uma View: junção aluno e curso

```
CREATE OR REPLACE VIEW aluno_curso AS
```

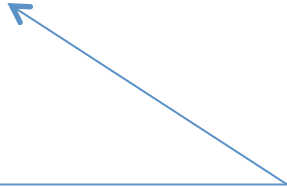
```
SELECT matricula, nome, titulo FROM aluno, curso WHERE  
aluno.curso_id = curso.curso_id;
```

```
SELECT * FROM aluno_curso
```

DML: DELETE

```
DELETE FROM curso WHERE titulo = 'estatística'
```

```
DELETE FROM aluno;
```



Atenção: restrição de integridade referencial

Algumas dicas sobre inserção

Múltiplas linhas

Criando uma tabela a partir de uma consulta

Alterando a Estrutura de uma Tabela

ALTER TABLE

DDL: Alteração Tabela

```
ALTER TABLE aluno ADD COLUMN endereco VARCHAR(200)
```

```
ALTER TABLE aluno ALTER COLUMN endereco VARCHAR(400)
```

```
ALTER TABLE aluno ALTER COLUMN endereco TYPE  
VARCHAR(400)
```

```
ALTER TABLE aluno RENAME COLUMN endereco TO  
endereco2
```

```
ALTER TABLE aluno DROP COLUMN endereco2
```

```
ALTER TABLE aluno RENAME TO aluno2
```

Removendo uma Tabela

`DROP TABLE`

DDL: Removendo Tabelas

DROP TABLE aluno;

O que acontecerá com a view?



DROP TABLE aluno **CASCADE**;

E neste caso?



Exercícios

Esquemas/Namespaces

CREATE SCHEMA

Organizando os objetos do banco

```
CREATE SCHEMA bdgeo;
```

```
CREATE TABLE bdgeo.curso  
(  
    curso_id    SERIAL,  
    titulo      VARCHAR(30) NOT NULL,  
    duracao     INTEGER      NOT NULL,  
    PRIMARY KEY(curso_id)  
);
```

Removendo Dados

DELETE

Tablespaces

Locais alternativos para armazenamento dos dados do banco

Tablespaces

- Por default o servidor mantém os arquivos de dados na pasta indicada por PG_DATA ou flag de inicialização do servidor. Ex:
C:\Program Files\PostgreSQL\9.2\data

- Para criar uma nova tablespace podemos:

```
CREATE TABLESPACE disco_c  
OWNER postgres  
LOCATION 'c:/pg-data';
```

- Para remover uma tablespace:

```
DROP TABLESPACE disco_c;
```

Consultas sobre informações do BD

Espaço usado em disco

```
SELECT pg_database_size('bdgcurso');
```

```
SELECT pg_size_pretty(pg_database_size('bdgcurso'));
```

```
SELECT pg_size_pretty(pg_relation_size('drenagem'));
```

```
SELECT pg_size_pretty(pg_table_size('drenagem'));
```

```
SELECT  
    pg_size_pretty(pg_total_relation_size('drenagem'));
```

Tabelas de Metadado

Catálogo do Sistema

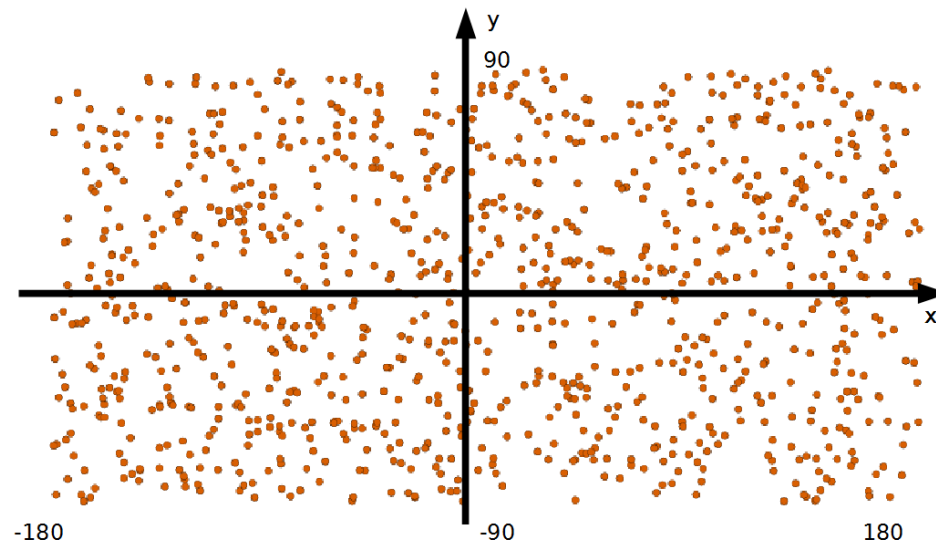
Listando tamanho de tabelas e índices

```
SELECT nspname || '.' || relname AS "relation",
        pg_size_pretty(pg_relation_size(C.oid)) AS
        "size",
        relkind AS "tipo"
FROM pg_class C
LEFT JOIN pg_namespace N ON (N.oid =
C.relnamespace)
WHERE nspname NOT IN ('pg_catalog',
        'information_schema', 'pg_toast', 'topology')
AND relkind IN ('r', 'i')
ORDER BY pg_relation_size(C.oid) DESC
LIMIT 20;
```

Stored Procedures

Abrir o script: [plpgsql-pts-table-pg.sql](#)

```
SELECT criar_tabela_pontos_pg('pt1m', 1000000);
```



Como um SGBD-R consegue ser eficiente na recuperação de dados?

Métodos de Indexação

Árvores-B⁺

GiST

Índices Funcionais

Como a consulta abaixo é processada?

```
SELECT * FROM pt1m  
WHERE x > 0.0 AND x < 1.0 AND y > 0.0 AND Y < 1.0
```

Como a consulta abaixo é processada?

```
SELECT * FROM pt1m
WHERE x > 0.0 AND x < 1.0 AND y > 0.0 AND Y < 1.0
```

```
EXPLAIN ANALYZE SELECT * FROM pt1m
WHERE x > 0.0 AND x < 1.0 AND y > 0.0 AND Y < 1.0
```

```
"Seq Scan on pt1m (cost=0.00..39231.00 rows=15
width=122) (actual time=8.129..365.055 rows=14
loops=1)"
```

```
" Filter: ((x > 0::double precision) AND (x < 1::double
precision) AND (y > 0::double precision) AND (y <
1::double precision))"
```

```
" Rows Removed by Filter: 999986"
```

```
"Total runtime: 365.119 ms"
```

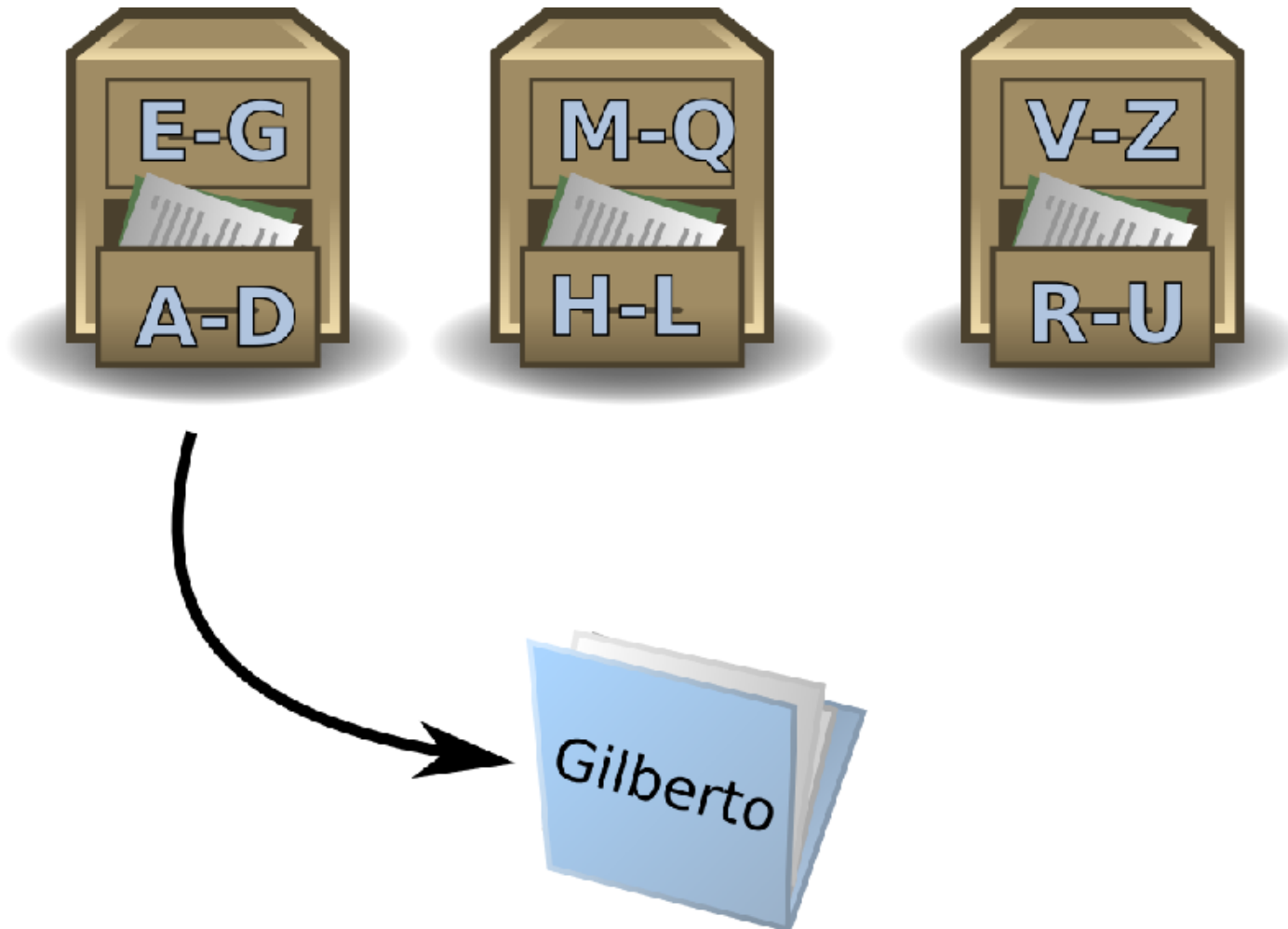
Acesso Sequencial Completo

- Recuperar o telefone de todos os funcionários.

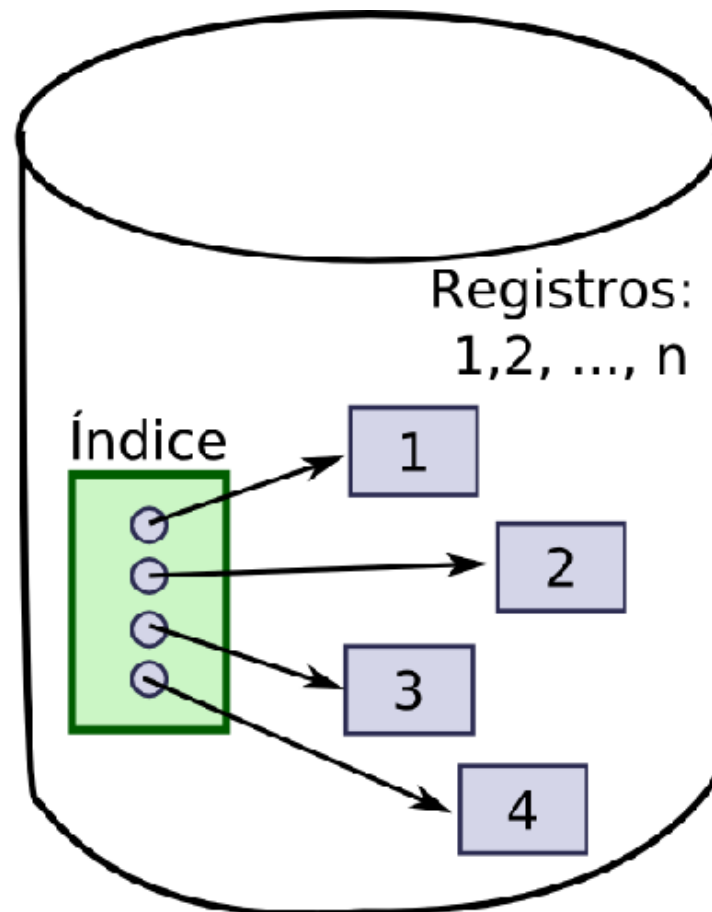


Seleção: Acesso Aleatório

- Recuperar as informações sobre o funcionário 'Gilberto'.

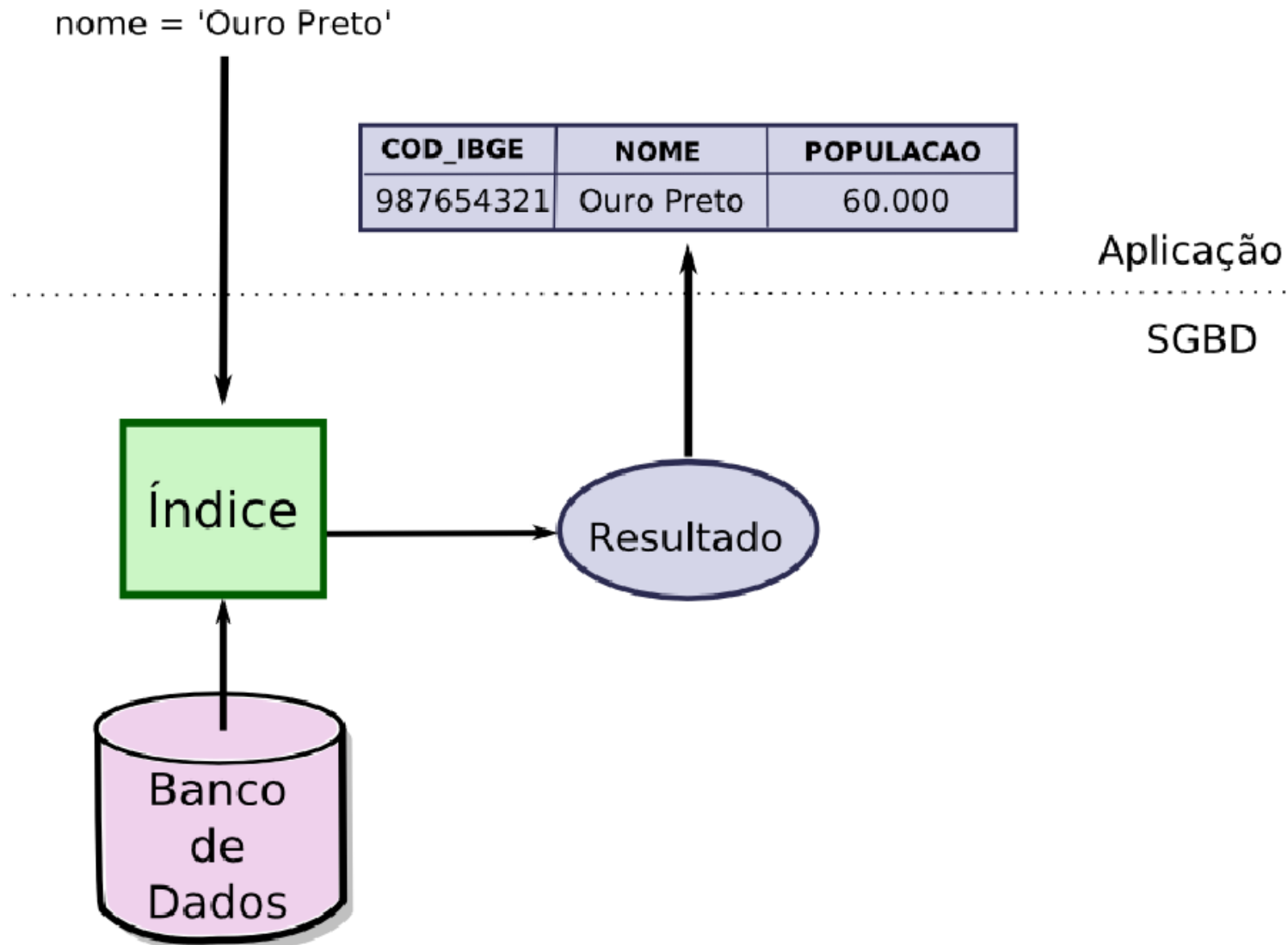


Índices: Visão Geral



Registros de um arquivo e o índice associado a este arquivo

Processamento Consultas Usando Índices



Vamos criar um índice e ver como fica nossa consulta?

```
CREATE INDEX pt1m_idx ON pt1m(x, y);
```

```
EXPLAIN ANALYZE SELECT * FROM pt1m  
WHERE x > 0.0 AND x < 1.0 AND y > 0.0 AND Y < 1.0
```

Vamos criar um índice e ver como fica nossa consulta?

```
CREATE INDEX pt1m_idx ON pt1m(x, y);
```

```
EXPLAIN ANALYZE SELECT * FROM pt1m  
WHERE x > 0.0 AND x < 1.0 AND y > 0.0 AND Y < 1.0
```

```
"Index Scan using pt1m_idx on pt1m (cost=0.00..146.51  
rows=15 width=122) (actual time=0.327..1.986 rows=14  
loops=1)"
```

```
" Index Cond: ((x > 0::double precision) AND (x <  
1::double precision) AND (y > 0::double precision) AND  
(y < 1::double precision))"
```

```
"Total runtime: 2.050 ms"
```

EXPLAIN

- Os valores do EXPLAIN correspondem:
 - Estimativa de custo inicial (ex: etapa de ordenação)
 - Estimativa total do custo
 - Estimativa do número total de linhas na saída
 - Estimativa média do tamanho (em bytes) de uma linha da saída
- Medida de custo:
 - Medida em relação ao custo de recuperar uma unidade do disco

Sintaxe: CREATE INDEX

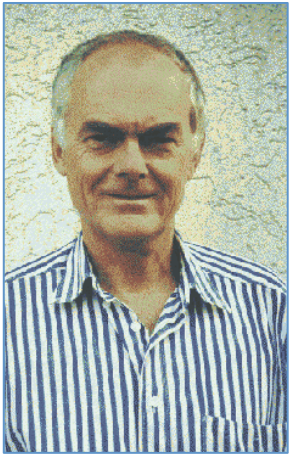
```
CREATE [UNIQUE] INDEX [CONCURRENTLY] name  
    ON table [USING method]  
({column|(expression)} [opclass] [ASC|DESC]  
[NULLS {FIRST|LAST}] [, ...])  
[WITH (storage_parameter = value [, ...])]  
[TABLESPACE tablespace]  
[WHERE predicate]
```

Organizando os Dados

- Acesso a Disco x Acesso em Memória Principal:
 - O acesso à memória secundária (hard-disk) é muito mais custoso do que à memória principal (RAM).
- A tabela pode conter muitos registros, inviabilizando mantermos todas as suas entradas na memória principal.
- No banco de dados (repositório propriamente dito) podemos ter diversas tabelas com milhares de linhas cada.
- Quais as estruturas utilizadas para auxiliar o processamento de consultas?
 - Índices Convencionais x Índices Espaciais

Métodos de Acesso Unidimensionais

Árvores-B (B-tree)



Rudolf Bayer
(1939)

Fields: Computer Science

Nationality: Germany

Institution: Technical
University Munich

Best known for:

- ✓ B-tree
- ✓ Red-black tree



Edward M.
McCreight

Fields: Computer Science

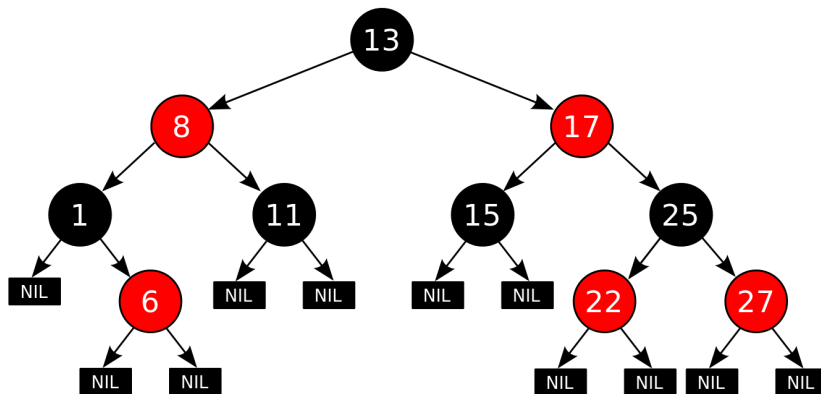
Nationality: American

Institutions:

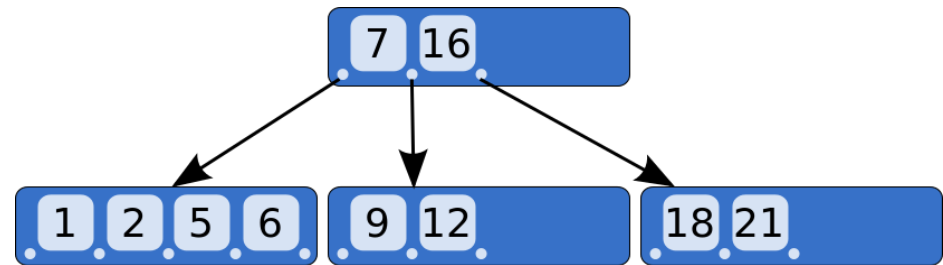
- Boeing
- Xerox PARC
- Adobe Systems

Best known for:

- ✓ B-tree



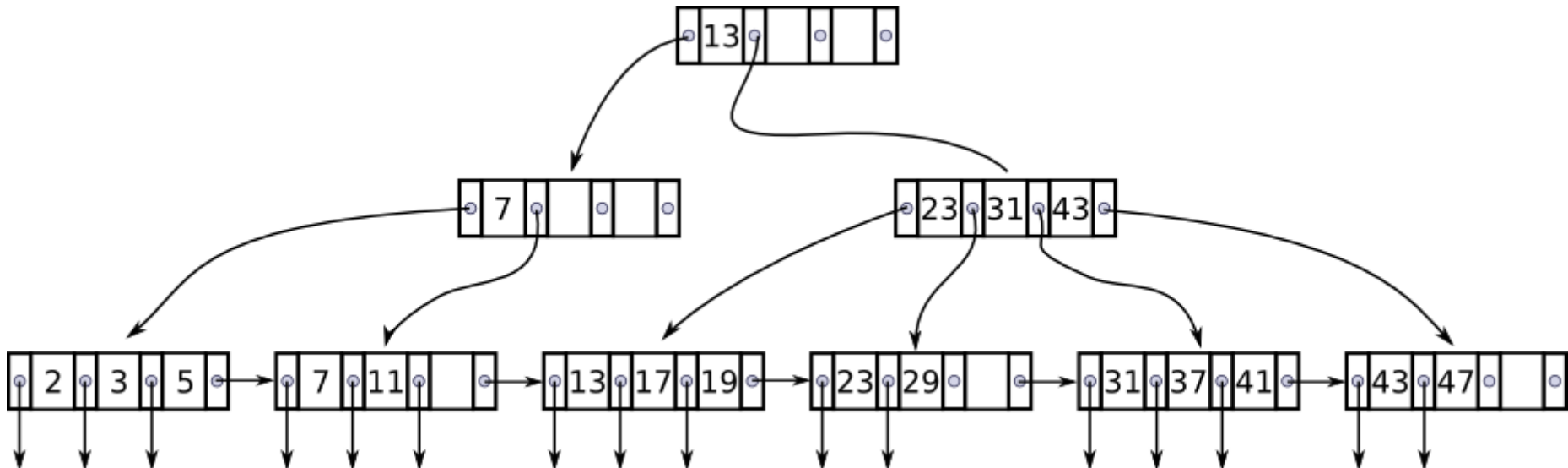
A red-black tree
Source: [Wikipedia](#)



A B-tree of Order 2 (or 5)
Source: [Wikipedia](#)

PostgreSQL → Índice Convencional → Árvores-B⁺

- Cada nodo pode conter k descendentes e $k - 1$ chaves:
 - Com $\lfloor d / 2 \rfloor \leq k \leq d$, onde d é a ordem da árvore
- A raiz tem pelo menos 2 filhos.
- Todas as folhas da árvore estão no mesmo nível
- As chaves em cada nodo estão em ordem crescente;



Árvores-B⁺: Aplicações Práticas

- É usada para manter restrições como:
 - chave primária (primary key)
 - valor único (unique key)
- Possibilita a criação de diversos índices de pesquisa, diferentes da sequência física do arquivo/tabela
- Permite responder algumas consultas sem ter que acessar o arquivo:
 - Count/Min/Max;
 - um determinado número de matrícula existe?

Árvores-B: Eficiência nas Recuperações

- O pior caso ocorre quando:

- a árvore tem o menor fator de ramificação em cada nó: $q = \lceil \frac{d}{2} \rceil$
- e a busca tem que atingir um nó folha

- Em uma árvore B de altura h existem:

1 chave na raiz +

$2(q-1)$ chaves no nível 1 +

$2q(q-1)$ chaves no nível 2 +

$2q^2(q-1)$ chaves no nível 3 +

...

$2q^{h-1}(q-1)$ chaves no nível h =

$$= 1 + 2(q-1) \left(\sum_0^{h-1} q^i \right) = 2q^h - 1$$

Número de chaves:

$$n \geq 2q^h - 1$$

ou seja:

$$h \leq \log_q \left[\frac{(n+1)}{2} \right]$$

*Na verdade, Árvore-B é uma família geral de estruturas de dados, existindo diversas variantes.

Árvores-B: Eficiência nas Recuperações

- Exemplo:
 - $d = 200$ e $n = 2.000.000$
- Então: $q = 100$ e $h \leq \log_{100}((2.000.000 + 1) / 2) = 3$
- Portanto, encontrar uma chave nessa árvore exige, no pior caso, $h + 1 = 4$ buscas.

Número de Nós Acessados – Pior Caso

Páginas ½ cheias	Número de Registros					
	1,00E+3	1,00E+4	1,00E+5	1,00E+6	1,00E+7	1,00E+8
5	5	7	8	10	11	12
25	3	4	5	5	6	7
50	3	3	4	5	5	6
100	3	3	4	4	5	5
150	2	3	3	4	4	5
200	2	3	3	4	4	4
250	2	3	3	4	4	4

Árvores-B: Eficiência Inserção

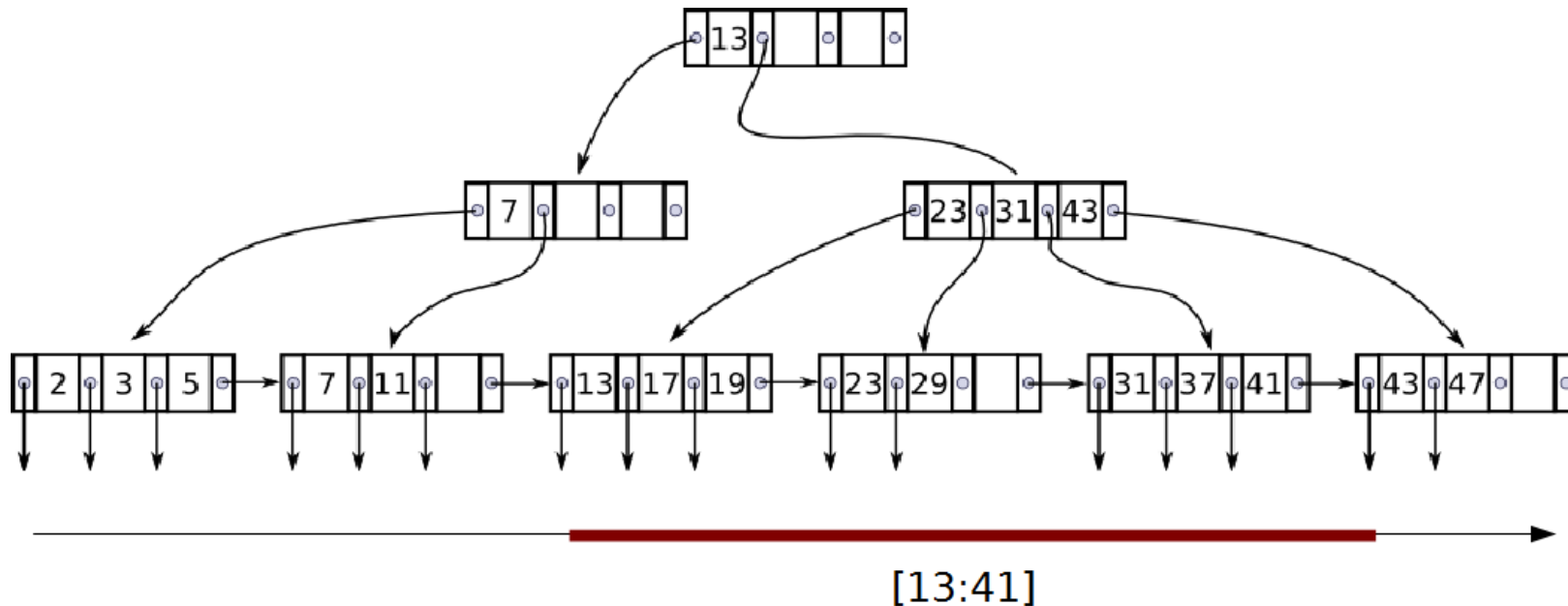
- Como podemos ver, no pior caso, a inserção de uma chave pode causar a propagação de divisões até chegar à raiz da árvore.
- Se a raiz for dividida, haverá um aumento de uma unidade na altura da árvore.
- Este processo é responsável por manter o balanceamento da árvore-B
- Ele ocorre em um caminho único da folha para raiz, de forma a não causar uma reorganização total da árvore, isto é, sem causar nenhum overhead.

Árvores-B⁺: Consultas de Intervalo (Unidim.)

- Também são úteis em consultas que solicitam um intervalo (unidimensional) de valores:

```
SELECT *  
  FROM tabela  
 WHERE coluna > 31
```

```
SELECT *  
  FROM tabela  
 WHERE coluna >= 13 AND coluna <= 41
```



Árvores-B⁺: PostgreSQL

- Considerações:
 - Uma vez criado, não é necessário mais intervenções pois o índice é atualizado toda vez que a tabela é modificada.
 - índices podem ser construídos e removidos a qualquer momento.
- Quais os comandos e operações que podem se beneficiar da existência de um índice?
 - SELECT, INSERT, UPDATE, DELETE, ORDER BY, LIMIT, Junção de tabelas
- A criação de um índice em uma tabela muito grande pode levar um tempo considerável.
- O query planner considerará o uso de um índice tipo Árvore-B quando a coluna(s) indexada(s) envolver(em) comparações usando os seguintes operadores:
 - <, <=, =, >=, >
 - BETWEEN, IN
 - LIKE (quando o carácter curinga está no fim)

Estatísticas do Banco

- Coletando estatísticas sobre o banco de dados:
 - Objetivo: Auxiliar a escolha do plano de execução mais eficiente p/ uma consulta
- Sintaxe:
VACUUM ANALYZE [table_name] [column_name]

ANALYZE [**VERBOSE**] [table [(column [, ...])]]
- Observações:
 - Sem parâmetros, este comando examina todas as tabelas do banco de dados
 - Ele armazena as estatísticas sobre o conteúdo das tabelas do banco de dados na tabela do catálogo do sistema chamada pg_statistic

Índices sobre Expressões ou Índices Funcionais

- Muitas vezes realizamos consultas como:
`SELECT * FROM test1 WHERE lower(col1) = 'value';`
- Esta consulta pode usar um índice?

Índices sobre Expressões ou Índices Funcionais

- Muitas vezes realizamos consultas como:

```
SELECT * FROM test1 WHERE lower(col1) = 'value';
```

- Esta consulta pode usar um índice?

- **Sim!**

```
CREATE INDEX test1_lower_col1_idx ON test1  
(lower(col1));
```

Extensibilidade

PostgreSQL

PostgreSQL: User Defined Types (UDT)

```
CREATE TYPE geo_point AS  
(  
    x    REAL,  
    y    REAL,  
    srid INTEGER  
);
```

PostgreSQL: User Defined Types (UDT)

```
CREATE TABLE sedes_municipais  
(  
  id INTEGER PRIMARY KEY,  
  location GEO_POINT  
);
```



```
INSERT INTO sedes_municipais  
VALUES (1, '(1, 2, 4326)::GEO_POINT);
```

PostgreSQL: User Defined Functions (UDF)

- Possibilita criar ou estender a álgebra de um determinado tipo de dado.

```
CREATE OR REPLACE FUNCTION less_than(first GEO_POINT, second GEO_POINT)
RETURNS REAL
AS $$
BEGIN
    IF(first.x < second.x)
    THEN
        RETURN TRUE;
    END IF;

    IF(first.x > second.x)
    THEN
        RETURN FALSE;
    END IF;

    ...
    RETURN FALSE;
END;
$$
LANGUAGE plpgsql;
```

PostgreSQL: User Defined Functions (UDF)

- Possibilita criar ou estender a álgebra de um determinado tipo de dado.

```
CREATE OR REPLACE FUNCTION distance(first GEO_POINT, second GEO_POINT)
RETURNS REAL
AS $$
DECLARE
    dx REAL;
    dy REAL;
BEGIN
    dx = (first.x - second.x) * (first.x - second.x);

    dy = (first.y - second.y) * (first.y - second.y);

    RETURN sqrt(dx + dy);
END;
$$
LANGUAGE plpgsql;
```

PostgreSQL: User Defined Functions (UDF)

- UDFs passam a fazer parte da linguagem de consulta do SGBD:

```
SELECT less_than('(1, 2, 4326)::GEO_POINT, '(10, 20, 4326)::GEO_POINT);
```

```
SELECT less_than('(1, 2, 4326)::GEO_POINT, '(-1, 2, 4326)::GEO_POINT);
```

```
SELECT distance('(1, 2, 4326)::GEO_POINT, '(10, 20, 4326)::GEO_POINT);
```


PostgreSQL: Sobrecarga de Operadores

```
CREATE OPERATOR <  
(  
  leftarg = GEO_POINT,  
  rightarg = GEO_POINT,  
  procedure = less_than,  
  commutator = >,  
  negator = >=  
);
```



```
SELECT '(1, 2, 4326)::GEO_POINT < '(10, 2, 4326)::GEO_POINT;
```

PostgreSQL: UDTs mais Complexos

- Em geral, utilizamos a linguagem C para criar tipos mais complexos no PostgreSQL.
- O núcleo da extensão PostGIS é um módulo em linguagem C.

Referências

Referências

- Wikipedia. [**Fotografia - Michael Stonebraker**](#). Acesso em: Setembro de 2013.
- [**Documentação do PostgreSQL**](#). Acesso em: Setembro de 2013.
- COMER, D. ***The Ubiquitous B-tree***. ACM Computing Surveys, v. 11, n. 2, Junho 1979.