Manual para instalação do ambiente “Sistema para gerenciamento de rota de vendas”

Documento complementar a monografia para instalação do ambiente

INDAIATUBA

2018

CELSO ALVES DE SOUZA

O ambiente de desenvolvimento utilizado foi configurado no sistema operacional Debian GNU/Linux 9 (*stretch*), mas da forma como foi montado o ambiente, é aplicável para a maioria dos sistemas Unix/Linux.

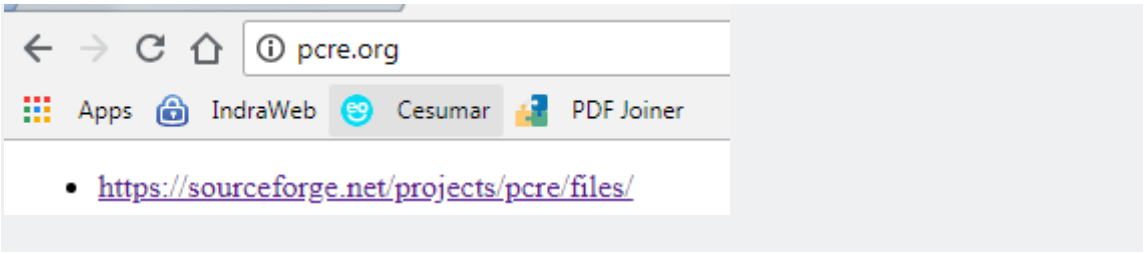
1. Instalação do servidor web APACHE versão 2.4.33

Instalação das bibliotecas necessárias para o servidor Apache:

sudo apt-get install libapr1-dev libaprutil1-dev

```
pi@cewarepi:~/apache/httpd-2.4.33 $ sudo apt-get install libapr1-dev libaprutil1-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Instalação das bibliotecas necessárias para o servidor Apache:



sudo unzip pcre-8.42.zip

./configure --prefix=/usr/local/pcre

```
pi@cewarepi:~/pcre/pcre-8.42 $ ./configure --prefix=/usr/local/pcre
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
```

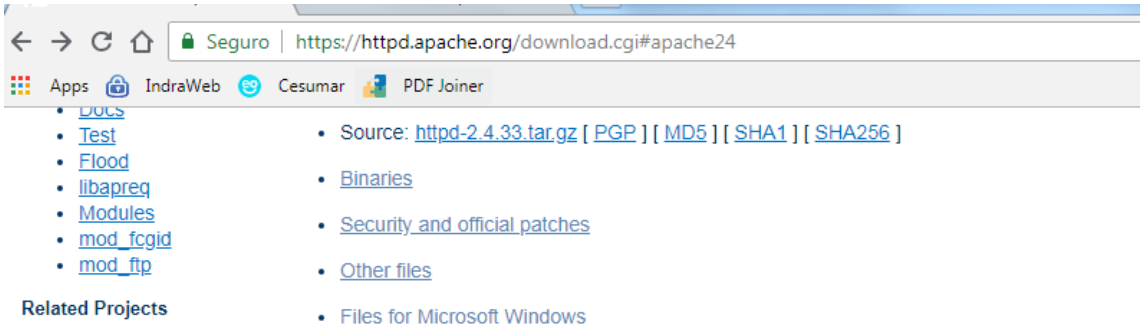
make

```
pi@cewarepi:~/pcre/pcre-8.42 $ make
rm -f pcre_chartables.c
ln -s ./pcre_chartables.c.dist pcre_chartables.c
make  all-am
make[1]: Entering directory '/home/pi/pcre/pcre-8.42'
  CC       libpcre_la-pcre_byte_order.lo
  CC       libpcre_la-pcre_compile.lo
```

sudo make install

```
pi@cewarepi:~/pcre/pcre-8.42 $ sudo make install
make  install-am
make[1]: Entering directory '/home/pi/pcre/pcre-8.42'
make[2]: Entering directory '/home/pi/pcre/pcre-8.42'
 /bin/mkdir -p '/usr/local/pcre/lib'
 /bin/bash ./libtool   --mode=install /usr/bin/install -c   libpcre.la libpcreposix.la libpcrecpp.la '/usr/local/pcre/lib'
libtool: install: /usr/bin/install -c .libs/libpcre.so.1.2.10 /usr/local/pcre/lib/libpcre.so.1.2.10
```

Baixar os binários para instalação no site do Apache:



Instalação do Apache:

tar –xvzf httpd-2.4.33.tar.gz

sudo ./configure --prefix=/usr/local/apache2 --with-pcre=/usr/local/pcre --enable-module=so --enable-so

```
pi@cewarepi:~/apache/httpd-2.4.33 $ sudo ./configure --prefix=/usr/local/apache2 --with-pcre=/usr/local/pcre
checking for chosen layout... Apache
checking for working mkdir -p... yes
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking build system type... armv7l-unknown-linux-gnueabihf
checking host system type... armv7l-unknown-linux-gnueabihf
checking target system type... armv7l-unknown-linux-gnueabihf
configure:
configure: Configuring Apache Portable Runtime library...
configure:
checking for APR... yes
  setting CC to "arm-linux-gnueabihf-gcc"
  setting CPP to "arm-linux-gnueabihf-gcc -E"
  setting CFLAGS to "  -pthread"
  setting CPPFLAGS to "  -DLINUX -D_REENTRANT -D_GNU_SOURCE -D_LARGEFILE64_SOURCE"
  setting LDFLAGS to " "
configure:
configure: Configuring Apache Portable Runtime Utility library...
configure:
```

make

```
pi@cewarepi:~/apache/httpd-2.4.33 $ make
Making all in srclib
make[1]: Entering directory '/home/pi/apache/httpd-2.4.33/srclib'
make[1]: Leaving directory '/home/pi/apache/httpd-2.4.33/srclib'
Making all in os
make[1]: Entering directory '/home/pi/apache/httpd-2.4.33/os'
Making all in unix
make[2]: Entering directory '/home/pi/apache/httpd-2.4.33/os/unix'
make[3]: Entering directory '/home/pi/apache/httpd-2.4.33/os/unix'
```

sudo make install

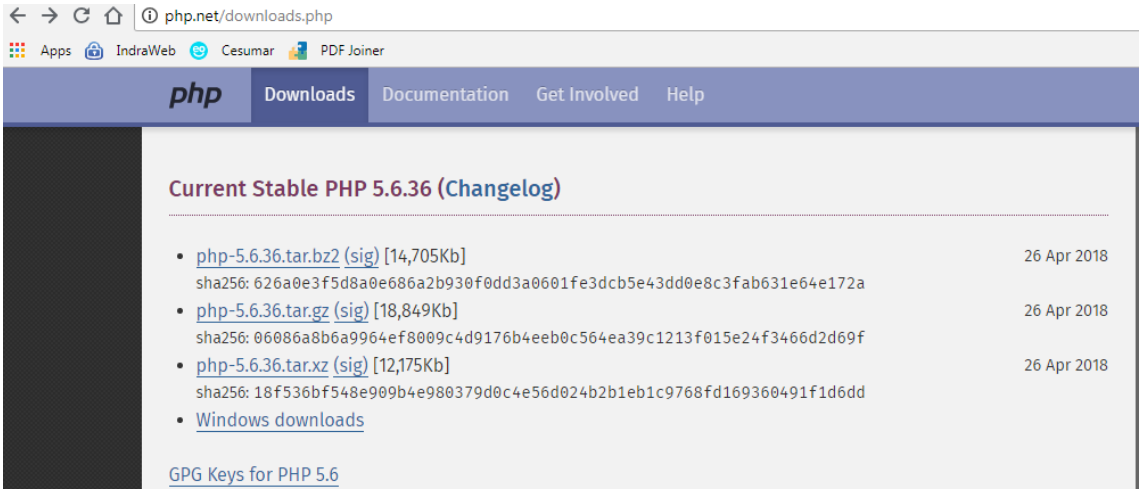
```
pi@cewarepi:~/apache/httpd-2.4.33 $ sudo make install
Making install in srclib
make[1]: Entering directory '/home/pi/apache/httpd-2.4.33/srclib'
make[2]: Entering directory '/home/pi/apache/httpd-2.4.33/srclib'
make[2]: Leaving directory '/home/pi/apache/httpd-2.4.33/srclib'
make[1]: Leaving directory '/home/pi/apache/httpd-2.4.33/srclib'
Making install in os
make[1]: Entering directory '/home/pi/apache/httpd-2.4.33/os'
Making install in unix
make[2]: Entering directory '/home/pi/apache/httpd-2.4.33/os/unix'
make[3]: Entering directory '/home/pi/apache/httpd-2.4.33/os/unix'
make[3]: Leaving directory '/home/pi/apache/httpd-2.4.33/os/unix'
make[2]: Leaving directory '/home/pi/apache/httpd-2.4.33/os/unix'
make[2]: Entering directory '/home/pi/apache/httpd-2.4.33/os'
make[2]: Leaving directory '/home/pi/apache/httpd-2.4.33/os'
make[1]: Leaving directory '/home/pi/apache/httpd-2.4.33/os'
Making install in server
make[1]: Entering directory '/home/pi/apache/httpd-2.4.33/server'
Making install in mpm
make[2]: Entering directory '/home/pi/apache/httpd-2.4.33/server/mpm'
Making install in event
make[3]: Entering directory '/home/pi/apache/httpd-2.4.33/server/mpm/event'
make[4]: Entering directory '/home/pi/apache/httpd-2.4.33/server/mpm/event'
mkdir /usr/local/apache2
mkdir /usr/local/apache2/modules
```

Para iniciar o servidor Apache, como root executar:

/usr/local/apache2/bin/apachectl start

2. Instalação do interpretador PHP

Baixar os binário do PHP no site:



tar –xvzf php-5.6.36.tar.gz

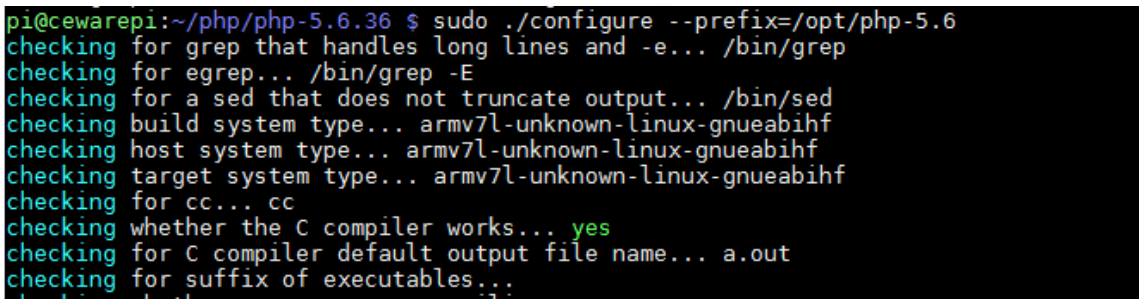
Instalando bibliotecas necessárias ao PHP:

sudo apt-get install libxml2-dev

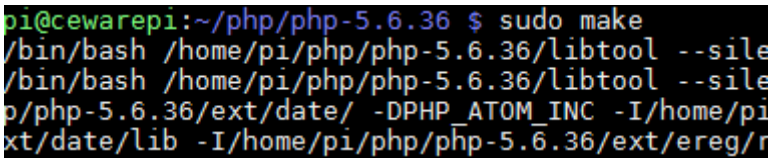
```
pi@cewarepi:~/php/php-5.6.36 $ sudo apt-get install libxml2-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  icu-devtools libicu-dev
Suggested packages:
  icu-doc
The following NEW packages will be installed:
  icu-devtools libicu-dev libxml2-dev
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 16.6 MB of archives.
After this operation, 73.7 MB of additional disk space will be used.
```

Instalando o PHP:

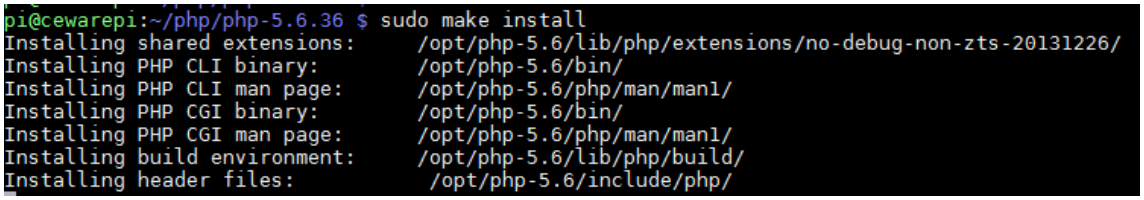
sudo ./configure --prefix=/opt/php-5.6 --with-apxs2=/usr/local/apache2/bin/apxs --with-pgsql=/usr/local/pgsql --with-openssl=/usr/bin/openssl



sudo make



sudo make install



Adicionar as linhas abaixo no /usr/local/apache2/conf/httpd.conf para que o Apache se integre ao PHP:

LoadModule php5_module modules/libphp5.so

DirectoryIndex **index.php** index.html index.html.var

AddType application/x-httpd-php .php

Reiniciar o Apache:

/usr/local/apache2/bin/apachectl stop

/usr/local/apache2/bin/apachectl start

Editar o arquivo /opt/php-5.6/lib/php.ini e adicionar as linhas abaixo:

register_globals = on

safe_mode = off

upload_max_filesize = 10M

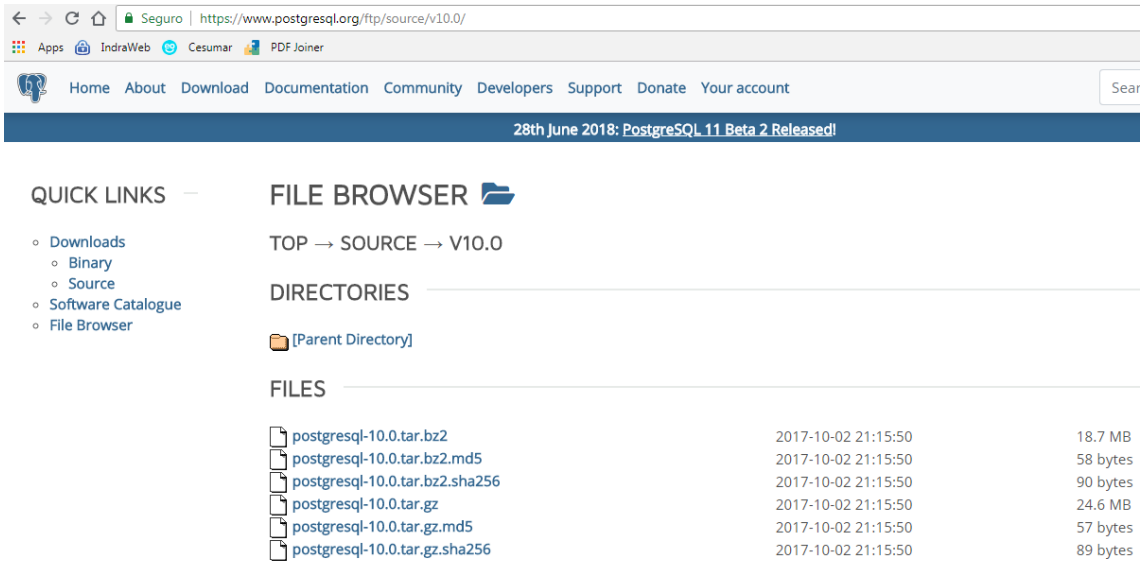
session.save_path = /home/ceware/sessions

magic_quotes_gpc = off

3. Instalação do banco de dados POSTGRESQL

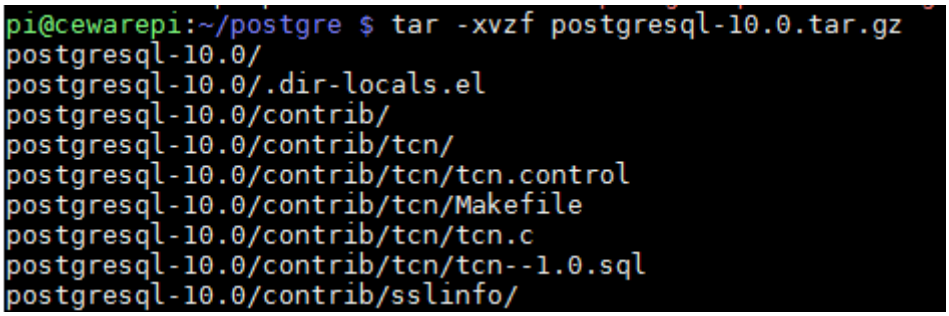
Baixar os binário no site:

wget -c https://ftp.postgresql.org/pub/source/v10.0/postgresql-10.0.tar.gz



Descompactando:

tar -xvzf postgresql-10.0.tar.gz



Instalando bibliotecas necessárias ao PostgreSQL

sudo apt-get install libreadline-dev

sudo apt-get install libreadline6

sudo apt-get install zlib1g-dev

sudo apt-get install libxslt1-dev

Instalando o PostgreSQL

./configure --prefix=/usr/local/pgsql --with-libxml --with-libxslt

sudo make

sudo make install

Adicionado usuário do Sistema Operacional:

useradd -m postgres

Iniciando o banco de dados:

su posgtes

bash

mkdir /home/postgres/data

su root

ln -s /home/postgres/data /usr/local/pgsql/data

su postgres

chmod 0700 /home/postgres/data

./initdb -U postgres -E utf8 -W -D /usr/local/pgsql/data

Edite o arquivo /home/postgres/data/postgresql.conf e retire o # da linha:

#listen_addresses = 'localhost'

Edite também o arquivo /home/postgres/data/pg_hba.conf e insira ao final dele a linha:

host all all 127.0.0.1 255.255.255.255 md5

su root

chown -R postgres:postgres /usr/local/pgsql

su postgres

bash

./pg_ctl -D /usr/local/pgsql/data -l /usr/local/pgsql/data/logfile start

Exemplo de um arquivo Shell para inciar/parar (como root):

```
/usr/local/pgsql/startPostgre.sh start

vi /usr/local/pgsql/startPostgre.sh

#/bin/bash

# Controla start / stop do Postgresql

PG_CTL_BIN="/usr/local/pgsql/bin/pgctl"

PG_DATA="/usr/local/pgsql/data"

case "$1" in

start) echo -n "Iniciando servico do PostgreSQL";

/bin/su - postgres -c "/usr/local/pgsql/bin/pg_ctl start -D
/usr/local/pgsql/data" > logfile 2>&1

;;

stop) echo -n "Parando servico PostgreSQL";

/bin/su - postgres -c "/usr/local/pgsql/bin/pg_ctl stop -D
/usr/local/pgsql/data" > logfile 2>&1

;;

restart) echo -n "Reiniciando servico PostgreSQL";

/bin/su - postgres -c "/usr/local/pgsql/bin/pg_ctl restart -D
/usr/local/pgsql/data" > logfile 2>&1

;;

esac

exit 0
```

Tornando o script executável:

chmod +x /usr/local/pgsql/startPostgre.sh

#Conectar pelo psql

psql -h 127.0.0.1 -p 5432 -U postgres -W

4. Instalação do PGADMIN para acesso ao banco de dados POSTGRESQL

Bibliotecas necessárias para o PGADMIN

apt install virtualenv python-pip libpq-dev python-dev

cd /home/your_user/

virtualenv pgadmin4

cd pgadmin4

source bin/activate

wget -c https://ftp.postgresql.org/pub/pgadmin/pgadmin4/v2.1/pip/pgadmin4-2.1-py2.py3-none-any.whl

O Aplicativo PGADMIN na versão Linux é executado através da linguagem Python:

pip install /home/ceware/TCC/Instaladores/PostgreSQL/pgadmin4/pgadmin4-2.1-py2.py3-none-any.whl

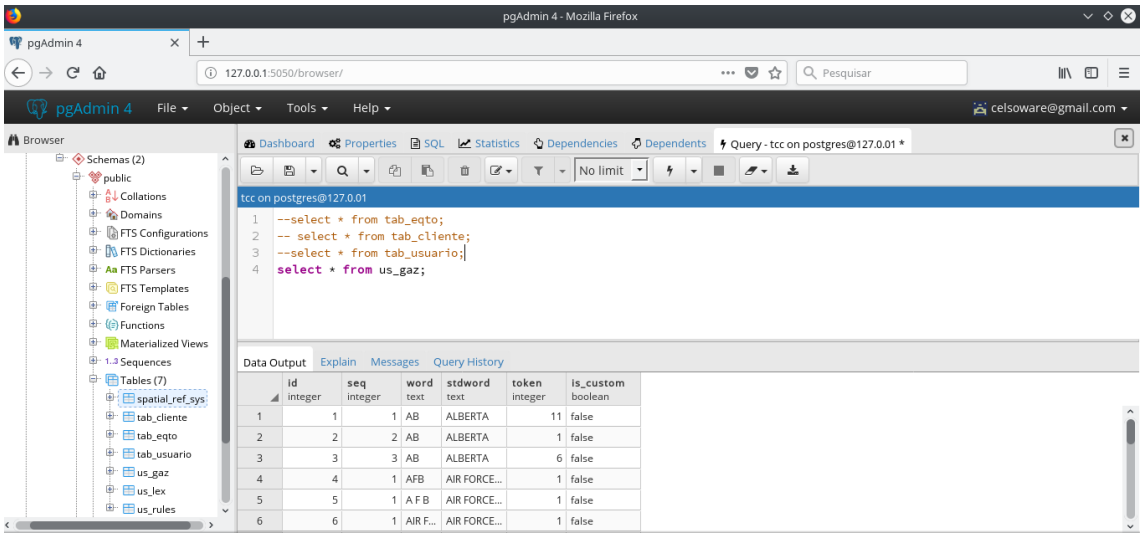
cd lib/python2.7/site-packages/pgadmin4

touch config_local.py

cd ../../../

python /home/ceware/pgadmin4/lib/python2.7/site-packages/pgadmin4/pgAdmin4.pyc

Em seguida Acessar pelo navegador a url http://127.0.0.1:5050



Script de criação do banco de dados:

```
CREATE DATABASE tcc
    WITH
    OWNER = postgres
    ENCODING = 'UTF8'
    LC_COLLATE = 'pt_BR.UTF-8'
    LC_CTYPE = 'pt_BR.UTF-8'
    TABLESPACE = pg_default
    CONNECTION LIMIT = -1;
```

Exemplo script para automatizar o início do PGADMIN:

```
vi /home/ceware/pgadmin4/pgadmin4.sh
#!/usr/bin/env bash
source /home/ceware/pgadmin4/bin/activate &&
python /home/ceware/pgadmin4/lib/python2.7/site-
packages/pgadmin4/pgAdmin4.py &
```

Tornado o script executável:

chmod +x /home/ceware/pgadmin4/pgadmin4.sh

Deve ser iniciado como root:

/home/ceware/pgadmin4/pgadmin4.sh

Em seguida Acessar http://127.0.0.1:5050/browser/

5. Instalação da extensão Postgis para trabalhar com dados geográficos no banco de dados POSTGRESQL

Baixando e instalando as bibliotecas necessárias para dados geográficos:

wget -c http://download.osgeo.org/geos/geos-3.7.0rc1.tar.bz2

tar -jxvf geos-3.7.0rc1.tar.bz2

cd geos-3.7.0

./configure

make

make install

wget -c http://download.osgeo.org/proj/proj-5.1.0.tar.gz

tar -xvzf proj-5.1.0.tar.gz

cd proj-5.1.0

./configure

make

make install

apt-get install gdal-bin libgdal-dev libgdal20 libsfcgal-dev libsfcgal1

Instalando a extensão Postgis:

cd postgis-2.4.4

./configure --with-pgconfig=/usr/local/pgsql/bin/pg_config --with-sfcgal=/usr/bin/sfcgal-config

make

make install

cd extensions

cd postgis

make clean

make

```
make install
cd ..
cd postgis_topology
make clean
make
make install
cd ..
cd postgis_sfcgal
make clean
make
make install
cd ..
cd address_standardizer
make clean
make
make install
make installcheck
cd ..
cd postgis_tiger_geocoder
make clean
make
make install
make installcheck
```

Executar o select abaixo para ver se a extensão está habilitada:

```
SELECT name, default_version,installed_version

FROM pg_available_extensions WHERE name LIKE 'postgis%' or name LIKE 'address%';

#Conectar no banco que for usar (não rodar esses commandos no banco postgres)

CREATE EXTENSION postgis;

CREATE EXTENSION postgis_sfcgal;

CREATE EXTENSION address_standardizer;

CREATE EXTENSION address_standardizer_data_us;

CREATE EXTENSION postgis_topology;
```

Dúvidas consultar:

https://postgis.net/docs/manual-2.4/postgis-br.html