

## Aprenda a instalar o PostgreSQL no Raspberry Pi

Date : 12 de Outubro de 2017

Apesar de ter um tamanho reduzido e um poder computacional limitado, o RaspberryPi (RPI) pode ser usado para vários projetos que não sejam muito exigentes. No Pplware já ensinamos a transformar o RPI num servidor Web, num servidor de VPNs, servidor de base de dados, etc.

Ao nível das bases de dados, depois de ensinarmos a instalar o SGBD MariaDB, hoje vamos aprender a instalar o fantástico [PostgreSQL](#) no RPi com o sistema PiPplware. Se ainda não tem um RPI, veja [aqui](#) a nossa promoção.



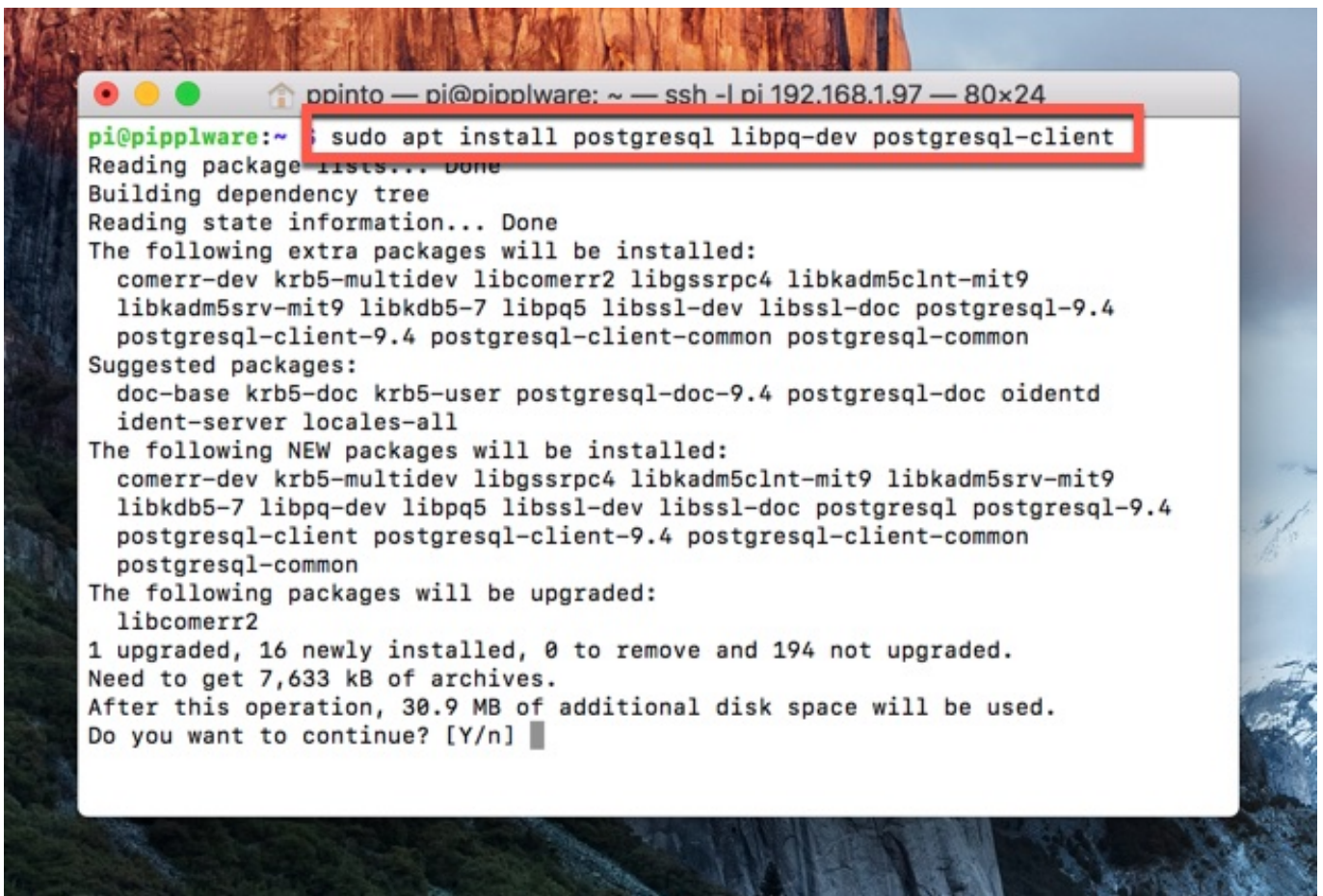
O [PostgreSQL](#) é atualmente o sistema open-source mais avançado e poderoso de gestão de bases de dados relacionais. Com suporte para a maioria dos sistemas operativos, incluindo GNU/Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS, Solaris, Tru64) e Windows OS, o PostgreSQL é extremamente eficiente e tem suporte para um [vasto leque](#) de solicitações: desde sistemas pessoais até complexos sistemas em cluster/distribuídos.

O PostgreSQL pode ser uma excelente opção para quem pretenda ter um poderoso sistema de gestão de base de dados a custo zero. O PostgreSQL é muito robusto e há muita documentação disponível.

## Como instalar?

Para proceder à instalação do PostgreSQL no RPi, basta abrir o terminal e inserir o seguinte comando

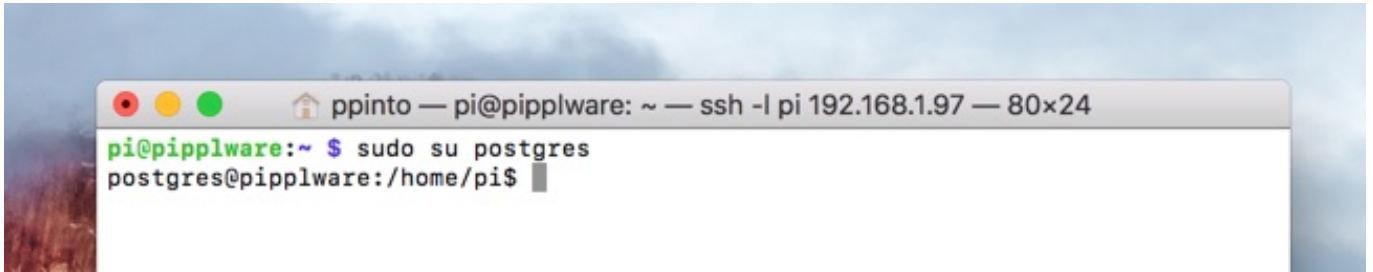
```
sudo apt install postgresql libpq-dev postgresql-client postgresql-client-common -y
```



```
pi@pipplware:~$ sudo apt install postgresql libpq-dev postgresql-client
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  comerr-dev krb5-multidev libcomerr2 libgssrpc4 libkadm5clnt-mit9
  libkadm5srv-mit9 libkdb5-7 libpq5 libssl-dev libssl-doc postgresql-9.4
  postgresql-client-9.4 postgresql-client-common postgresql-common
Suggested packages:
  doc-base krb5-doc krb5-user postgresql-doc-9.4 postgresql-doc oidentd
  ident-server locales-all
The following NEW packages will be installed:
  comerr-dev krb5-multidev libgssrpc4 libkadm5clnt-mit9 libkadm5srv-mit9
  libkdb5-7 libpq-dev libpq5 libssl-dev libssl-doc postgresql postgresql-9.4
  postgresql-client postgresql-client-9.4 postgresql-client-common
  postgresql-common
The following packages will be upgraded:
  libcomerr2
1 upgraded, 16 newly installed, 0 to remove and 194 not upgraded.
Need to get 7,633 kB of archives.
After this operation, 30.9 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Depois de instalado, mudem para o utilizador **Postgres** usando o comando

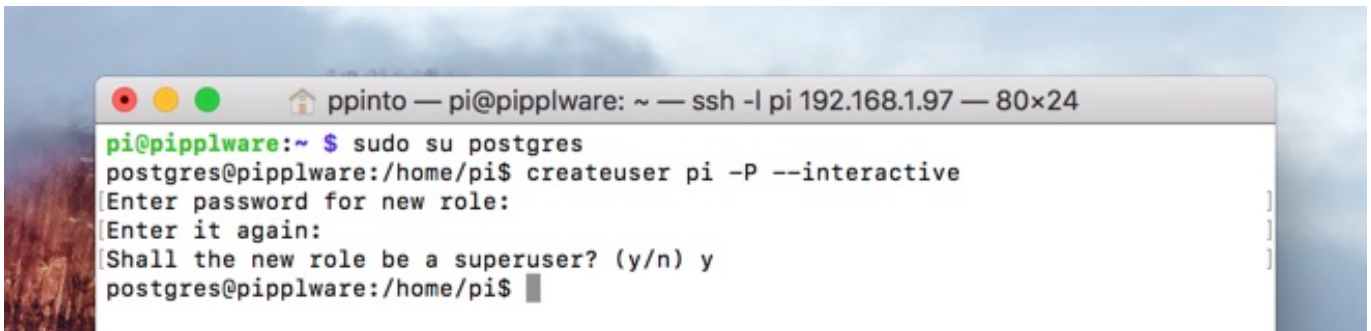
```
sudo su postgres
```



```
ppinto — pi@pipplware: ~ — ssh -l pi 192.168.1.97 — 80x24
pi@pipplware:~ $ sudo su postgres
postgres@pipplware:/home/pi$
```

O próximo passo é criar um utilizador. Se usarmos um utilizador já criado no sistema basta definir as permissões. Para tal podemos usar o comando **createuser** que nos permite definir as permissões de modo interativo.

```
createuser pi -P --interactive
```

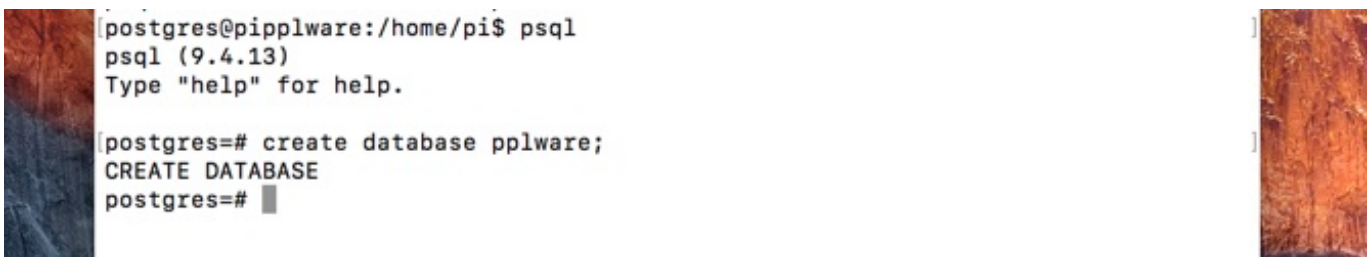


```
ppinto — pi@pipplware: ~ — ssh -l pi 192.168.1.97 — 80x24
pi@pipplware:~ $ sudo su postgres
postgres@pipplware:/home/pi$ createuser pi -P --interactive
[Enter password for new role:
[Enter it again:
[Shall the new role be a superuser? (y/n) y
postgres@pipplware:/home/pi$
```

## Como criar uma base de dados?

Para criar uma base de dados no Postgres basta usar os seguintes comandos:

```
$ psql > create database Pplware;
```



```
postgres@pipplware:/home/pi$ psql
psql (9.4.13)
Type "help" for help.

postgres=# create database pplware;
CREATE DATABASE
postgres=#
```

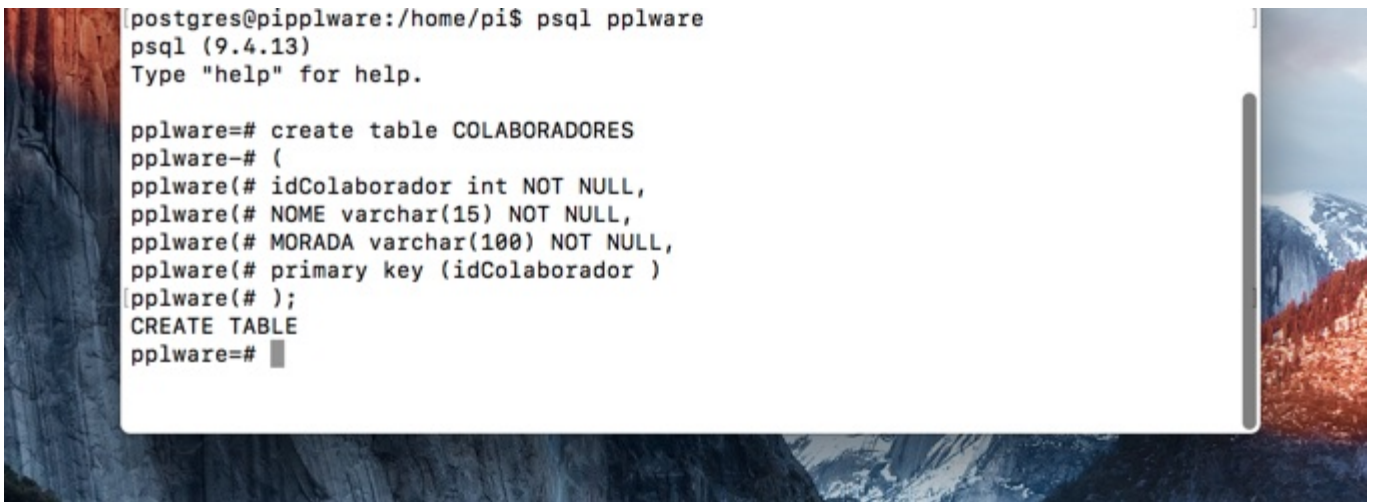
Depois de criada a base de dados, carreguem em CTRL+D para sair. Para entrar na base de dados basta que usem o seguinte comando:

psql Pplware

Vamos agora criar uma tabela de **COLABORADORES** usando a seguinte estrutura:

**|idColaborador|NOME|Morada**

```
create table COLABORADORES ( idColaborador int NOT NULL, NOME varchar(15) NOT NULL, MORADA varchar(100) NOT NULL, primary key (idColaborador ) );
```

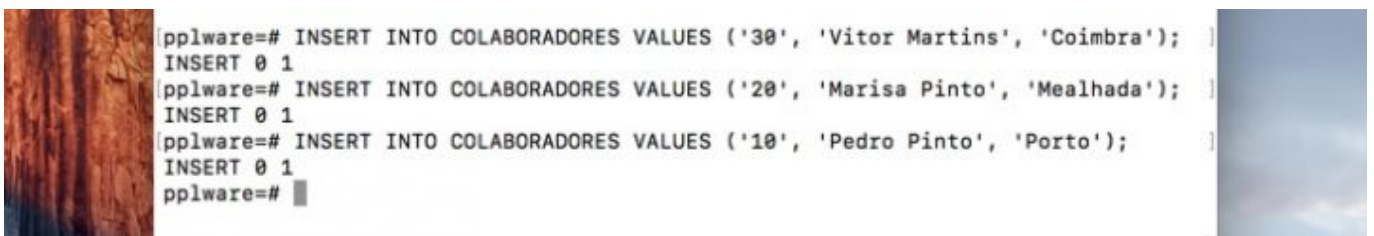


```
[postgres@pipplware:/home/pi$ psql pplware
psql (9.4.13)
Type "help" for help.

pplware=# create table COLABORADORES
pplware=# (
pplware(# idColaborador int NOT NULL,
pplware(# NOME varchar(15) NOT NULL,
pplware(# MORADA varchar(100) NOT NULL,
pplware(# primary key (idColaborador )
pplware(# );
CREATE TABLE
pplware=# █
```

Vamos agora criar três registos de utilizadores na tabela criada anteriormente:

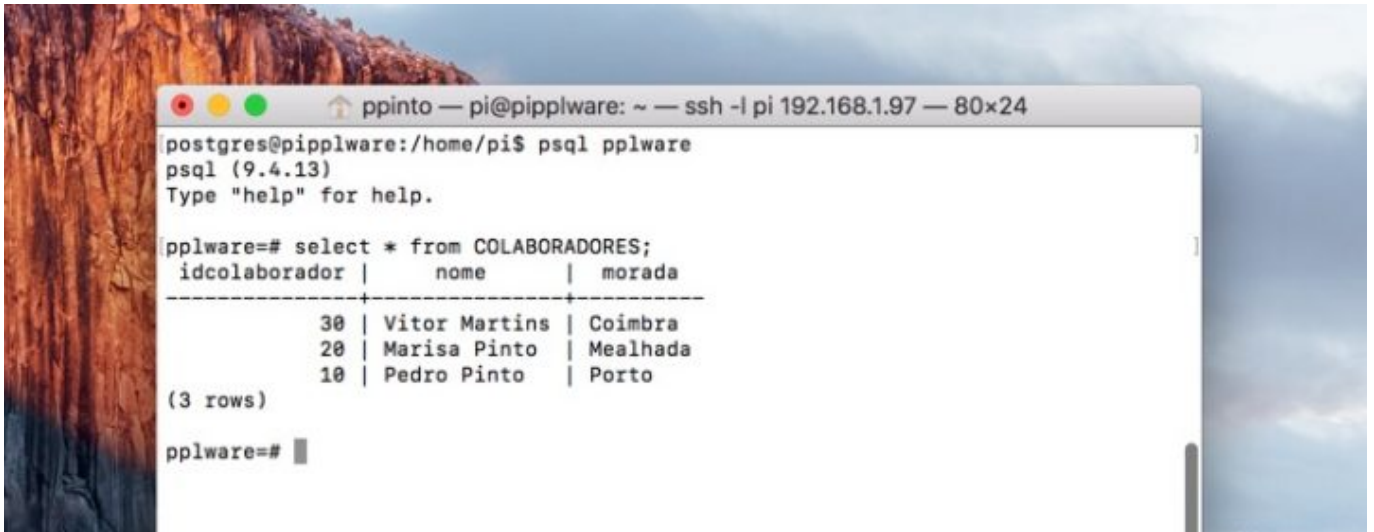
```
INSERT INTO COLABORADORES VALUES (10, 'Pedro Pinto', 'Porto'); INSERT INTO COLABORADORES VALUES (20, 'Marisa Pinto', 'Mealhada'); INSERT INTO COLABORADORES VALUES (30, 'Vitor Martins', 'Coimbra');
```



```
pplware=# INSERT INTO COLABORADORES VALUES ('30', 'Vitor Martins', 'Coimbra');
INSERT 0 1
pplware=# INSERT INTO COLABORADORES VALUES ('20', 'Marisa Pinto', 'Mealhada');
INSERT 0 1
pplware=# INSERT INTO COLABORADORES VALUES ('10', 'Pedro Pinto', 'Porto');
INSERT 0 1
pplware=# █
```

E está feito! Se agora pretenderem ver os dados inseridos na tabela basta usar o comando **SELECT**.

```
select * from COLABORADORES;
```



```
ppinto — pi@pipplware: ~ — ssh -l pi 192.168.1.97 — 80x24
postgres@pipplware:/home/pi$ psql pipplware
psql (9.4.13)
Type "help" for help.

pipplware=# select * from COLABORADORES;
 idcolaborador |   nome   | morada
-----+-----+-----
              30 | Vitor Martins | Coimbra
              20 | Marisa Pinto  | Mealhada
              10 | Pedro Pinto   | Porto
(3 rows)

pipplware=#
```

Comparativamente ao MySQL/MariaDB, o PostgreSQL traz alguma complexidade inicial mas é tudo uma questão de tempo. Para **Mark Shuttleworth**, fundador do Ubuntu, o PostgreSQL é o verdadeiro SGDB! Experimentem e caso dominem o PostgreSQL partilhem com os restantes leitores alguns comandos. Espreitem também [este](#) tutorial sobre PostgreSQL.