

# CONFIGURAÇÃO APLICAÇÃO

## Servidor de Aplicação POLO

1- Instalar o Nginx;

Obs1: Configurar arquivos **polo.conf** dentro dos diretórios **/etc/nginx/sites-available** e **/etc/nginx/sites-enabled**

Obs2: utilizar “**sudo nano /etc/nginx/NomeDoDiretorio/polo.conf**” para criar o arquivo polo.conf nos repositórios acima, e copiar os dados que se encontram no mesmo arquivo no servidor de produção, e remover trecho das configurações de SSL. Os dois diretórios terão o mesmo arquivo de configuração.

2- Adicionar repositório do PHP7.0 com “**sudo add-apt-repository ppa:ondrej/php**” e atualizar com o “**sudo apt-get update**”;

3- Instalar as dependências do PHP com “**sudo apt-get install php7.0 php7.0-dev php7.0-cli php7.0-fpm**”;

4- Instalar os seguintes pacotes do PHP:

- php-bcmath
- php-gd
- php-mbstring
- php-odbc
- php7.0-pdo-pgsql
- php-pgsql
- php-zip
- php-dev

5- Instalar o COMPOSER na versão 1;

**sudo curl -sS https://getcomposer.org/installer | sudo php -- --install-dir=/usr/local/bin --filename=composer --version=1.10.23**

**6-** Instalar o git;

**7-** Instalar o gcc, make, autoconf, pkg-config, uuid-dev, libtool, build-essential, libssl-dev, fonts-texgyre, gdebi, locales;

INSTALAR OS COMANDOS ABAIXO:

### **ETAPA 1**

a) `wget http://http.us.debian.org/debian/pool/main/libj/libjpeg-turbo/libjpeg62-turbo_2.1.5-2_amd64.deb`

b) `gdebi --n libjpeg62-turbo_2.1.5-2_amd64.deb`

c) `rm libjpeg62-turbo_2.1.5-2_amd64.deb`

### **ETAPA 2**

a) `wget https://github.com/wkhtmltopdf/wkhtmltopdf/releases/download/0.12.2.1/wkhtmltox-0.12.2.1_linux-jessie-amd64.deb`

b) `gdebi --n wkhtmltox-0.12.2.1_linux-jessie-amd64.deb`

c) `rm wkhtmltox-0.12.2.1_linux-jessie-amd64.deb`

### **ETAPA 3**

a) `cd /tmp`

b) `wget https://github.com/zeromq/libzmq/releases/download/v4.2.2/zeromq-4.2.2.tar.gz`

c) `tar -zxf zeromq-4.2.2.tar.gz`

d) `cd zeromq-4.2.2`

e) `sudo apt-get install libunwind-dev`

f) `./configure`

g) `make CXXFLAGS="-Wno-error=stringop-truncation"`

h) `sudo make install`

i) `sudo ldconfig`

#### **ETAPA 4**

a) `cd /tmp`

b) `git clone https://github.com/mkoppanen/php-zmq.git /tmp/php-zmq`

c) `cd /tmp/php-zmq`

d) `phpize`

e) `./configure`

f) make

g) sudo make install

h) echo "extension=zmq.so" >> /etc/php/7.0/fpm/conf.d/99-zmq.ini

i) echo "extension=zmq.so" >> /etc/php/7.0/cli/conf.d/99-zmq.ini

## **ETAPA 5**

a) cd /tmp

b) wget https://pecl.php.net/get/mongodb-1.2.11.tgz

c) tar -zxvf mongodb-1.2.11.tgz

d) cd mongodb-1.2.11/

e) phpize

f) ./configure

g) make all -j 5

h) make install

i) echo "extension=mongodb.so" >> /etc/php/7.0/fpm/conf.d/99-mongodb.ini

j) echo "extension=mongodb.so" >> /etc/php/7.0/cli/conf.d/99-mongodb.ini

## SERVIDOR BANCO DO POSTGRE

1- Instalar a ultima versão do postgre

- Liberar a porta 5432 para comunicação:

- 1- `sudo ufw allow 5432;`
- 2- `ufw status;`
- 3- `sudo ufw disable;`
- 4- `sudo ufw enable;`

- Configurar o hash de autenticação no banco para aceitar a criptografia md5:

1- `sudo nano /etc/postgresql/<versão>/main/pg_hba.conf`

2- `host all all 127.0.0.1/32 md5` ( IDENTIFIQUE ESSA LINHA DE CONFIGURAÇÃO DE IPV4 )

3- `sudo systemctl restart postgresql`

- Fazer Backup do banco postgre:

1- `pg_dump -h <hostname> -p <porta> -U <usuario> -d <nome_do_banco> -f /caminho/do/backup.sql`

## SERVIDOR BANCO DO MONGODB

- Instalar versão do mongo 4.2 ( <https://www.cherryservers.com/blog/install-mongodb-ubuntu-22-04> )

- Configurar um usuário de acesso ao banco ( <https://www.cherryservers.com/blog/install-mongodb-ubuntu-22-04> )

- Liberar porta 27017 para comunicação:

- 5- `sudo ufw allow 27017;`

- 6- ufw status;
- 7- sudo ufw disable;
- 8- sudo ufw enable;

- Fazer Backup do banco mongodb:

- 1- mongodump --host localhost --port 27017 --db meubanco --out /caminho/do/backup

## **CONFIGURAÇÃO NO PROJETO**

Dentro do diretório do polo, você irá executar o comando:

- 1- **sudo nano .env**

irá abrir o arquivo de configuração dos bancos. Lembre-se de mudar o ip dos bancos para o 127.0.0.1 e modificar o nome dos dbnames e dos dbusers caso estejam diferentes dos criados nos seus ambientes postgre e mongo. Além disso, você irá fazer a seguinte configuração: