

**Instituto Metr pole Digital - Universidade Federal do Rio Grande do Norte (IMD/UFRN)**  
**Rio Grande do Norte - RN - Brazil**

**Airton C. de O. Neto, Bruno M. de L. Cabral, Daniel de O. Silva, Dionisio D. A. de Carvalho,**  
**Erick de O. Silva, Franklin M. da C. Lima, Luis E. R. Ferreira**

netoairton@yahoo.com.br, brunomatheuslc@hotmail.com, daniel99955@hotmail.com,  
dionisio@naracosta.com.br, erickoliveira.eos@gmail.com,  
franklinmatheus1@gmail.com, eduardo\_ri\_que@hotmail.com

**PiRouter: Tutorial da implementa  o do roteador Wi-Fi com Raspberry Pi**

**Natal/RN, 2017**

- **Configurando o Raspberry Pi como roteador Wi-Fi**

- 1 Instalar Raspbian.
- 2 `sudo raspi-config`
  - Alterar a senha
  - Ativar ssh
  - Expandir filesystem
- 3 `sudo reboot`
- 4 `sudo apt-get update`
- 5 `sudo apt-get install dnsmasq hostapd`
- 6 `sudo nano /etc/dhcpd.conf`
  - Adicionar na última linha: `denyinterfaces wlan0`
- 7 `sudo nano /etc/network/interfaces`
  - utilize o arquivo `interfaces` na pasta `config` deste github
- 8 `sudo reboot`
- 9 `sudo nano /etc/hostapd/hostapd.conf`
  - utilize o conteúdo do arquivo `hostapd_sem_freeradius.conf` na pasta `config` deste github
- 10 testar com: `sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf`
- 11 `sudo nano /etc/default/hostapd` - alterar a linha `#DAEMON_CONF=""` para:
  - `DAEMON_CONF="/etc/hostapd/hostapd.conf"`
- 12 `sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig`
- 13 `sudo nano /etc/dnsmasq.conf`
  - utilize o arquivo `dnsmasq.conf` na pasta `config` deste github
- 14 `sudo nano /etc/sysctl.conf` – alterar a linha `#net.ipv4.ip_forward=1` removendo o `#`
- 15 `sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE`
- 16 `sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT`
- 17 `sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT`
- 18 `sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"`
- 19 `sudo nano /etc/rc.local` – alterar a linha imediatamente acima do `"exit 0"` para:
  - `iptables-restore < /etc/iptables.ipv4.nat`
- 20 `sudo reboot`

**Pronto!** Após o reinício, seu Raspberry funcionará como roteador Wi-Fi.

## FUNCIONALIDADES EXTRAS

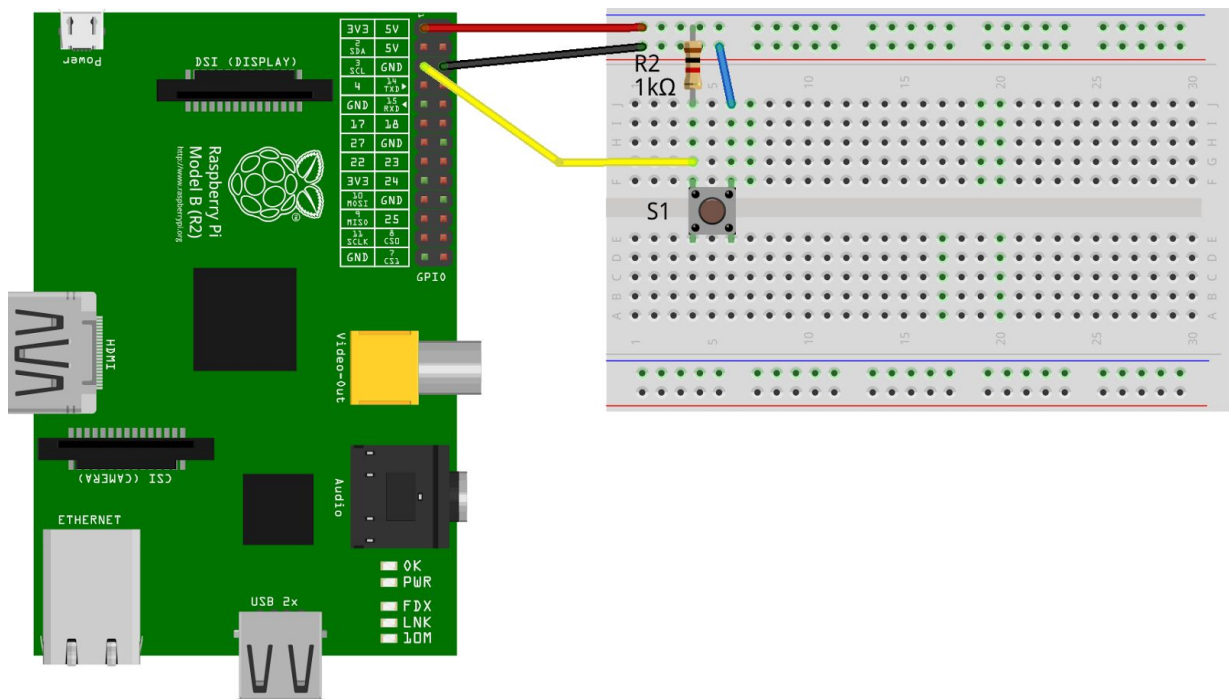
- **Instalando autenticação RADIUS**

- 1 `sudo apt-get install freeradius`
- 2 `sudo reboot`
- 3 `sudo nano /etc/freeradius/3.0/clients.conf`
  - procurar a linha que contém: `secret = testing123` e alterar para: `secret = depressao123`
- 4 `sudo cp /etc/freeradius/3.0/users /etc/freeradius/3.0/users.bak`
- 5 `sudo vi /etc/freeradius/3.0/users`
  - digitar → `:1,$d` (isto vai apagar todas as linhas do arquivo)
  - preencher com o cadastro dos usuários que vão entrar na rede da seguinte forma (exemplo):
    - `mario Cleartext-Password := "2016038001"`
    - `jose Cleartext-Password := "istoehumassenha"`
  - veja o arquivo exemplo (users) na pasta config deste github
- 6 `sudo systemctl daemon-reload`
- 7 `sudo systemctl enable freeradius.service`
- 8 `sudo systemctl start freeradius.service`
- 9 Para testar o servidor RADIUS basta usar:
  - `radtest nome_usuario senha_usuario localhost 1812 depressao123`
- 10 `sudo nano /etc/hostapd/hostapd.conf`
  - utilize o arquivo hostapd.conf da pasta config deste github
- 11 `sudo reboot`

**Pronto!** Após o reinício seu Raspberry estará funcionando como um roteador Wi-Fi com autenticação Radius interna.

## ● Instalando Botão Poweroff/Reset

- 1 `sudo nano /home/pi/pishutdown.py`
  - utilizar o código contido no arquivo `pishutdown.py` dentro da pasta `sources` deste github
- 2 `sudo crontab -e`
  - escolher o editor (caso peça);
  - adicionar no final do arquivo:
    - `@reboot sudo python /home/pi/pishutdown.py`
- 3 desligue e instale o botão de acordo com o circuito.

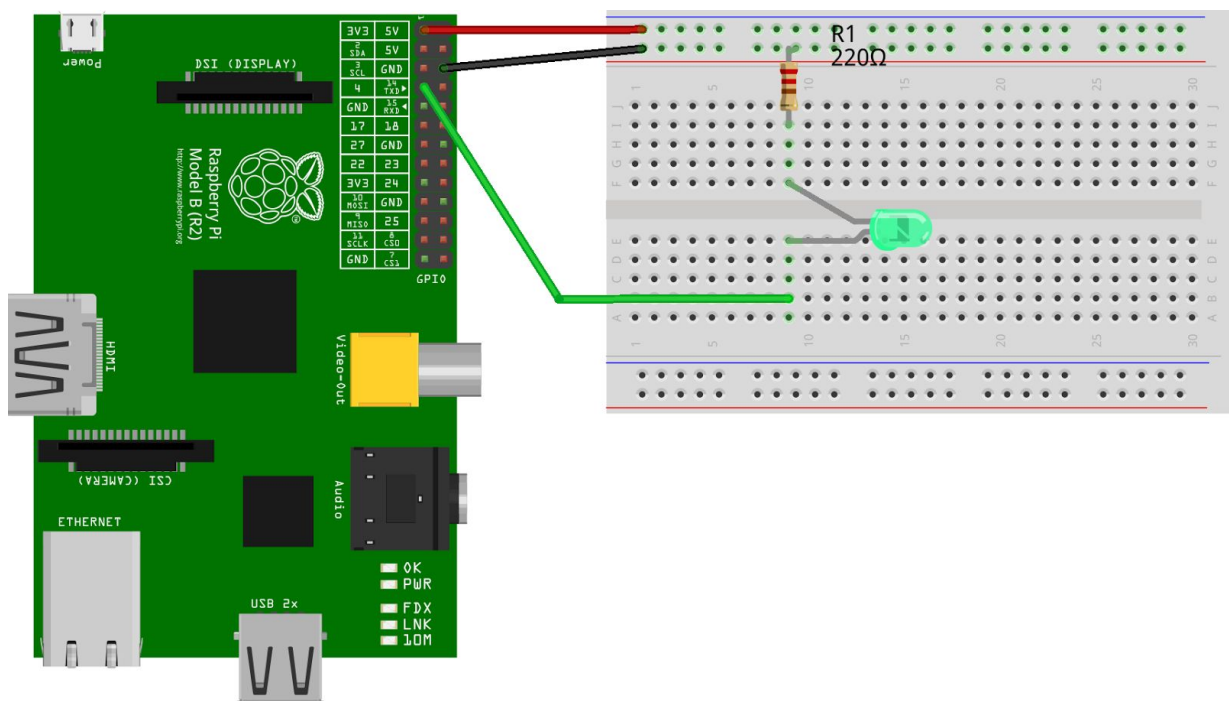


fritzing

**Pronto!** Após o reinício do seu Raspberry ele responderá às ações do botão (apertar e soltar em menos de 3 segundos fará ele reiniciar, por mais de 3 segundos fará ele desligar).

## • Instalando Led Indicador de Serviço

- 1 `sudo nano /home/pi/testservice.py`
  - utilize o código contido no arquivo `testservice.py` dentro da pasta `sources` deste github
- 2 `sudo crontab -e`
  - escolher o editor (caso peça);
  - adicionar no final do arquivo:
    - `@reboot sudo python /home/pi/testservice.py`
- 3 desligar e instalar o led de acordo com o circuito.



fritzing

**Pronto!** Após o reinício seu Raspberry indicará que os serviços `freeradius` e `hostapd` estão em funcionamento através do led.

- **Instalando auto atualização da lista de usuários através do SIGAA**

- 1 Solicitar acesso à API do SIGAA fazendo o cadastro
- 2 `sudo nano /home/pi/updateusers.py`
  - utilize o código do arquivo `updateusers.py` dentro da pasta `sources` deste github
- 3 `sudo apt-get install python-requests;`
- 4 `sudo crontab -e`
  - escolher o editor (caso peça);
  - adicionar no final do arquivo:
    - `@daily @reboot sudo python /home/pi/updateusers.py`
- 5 `sudo reboot`

**Pronto!** Após o reinício do seu Raspberry ele atualizará a lista de usuários para autenticação do FreeRadius após todas as reinicializações e diariamente.