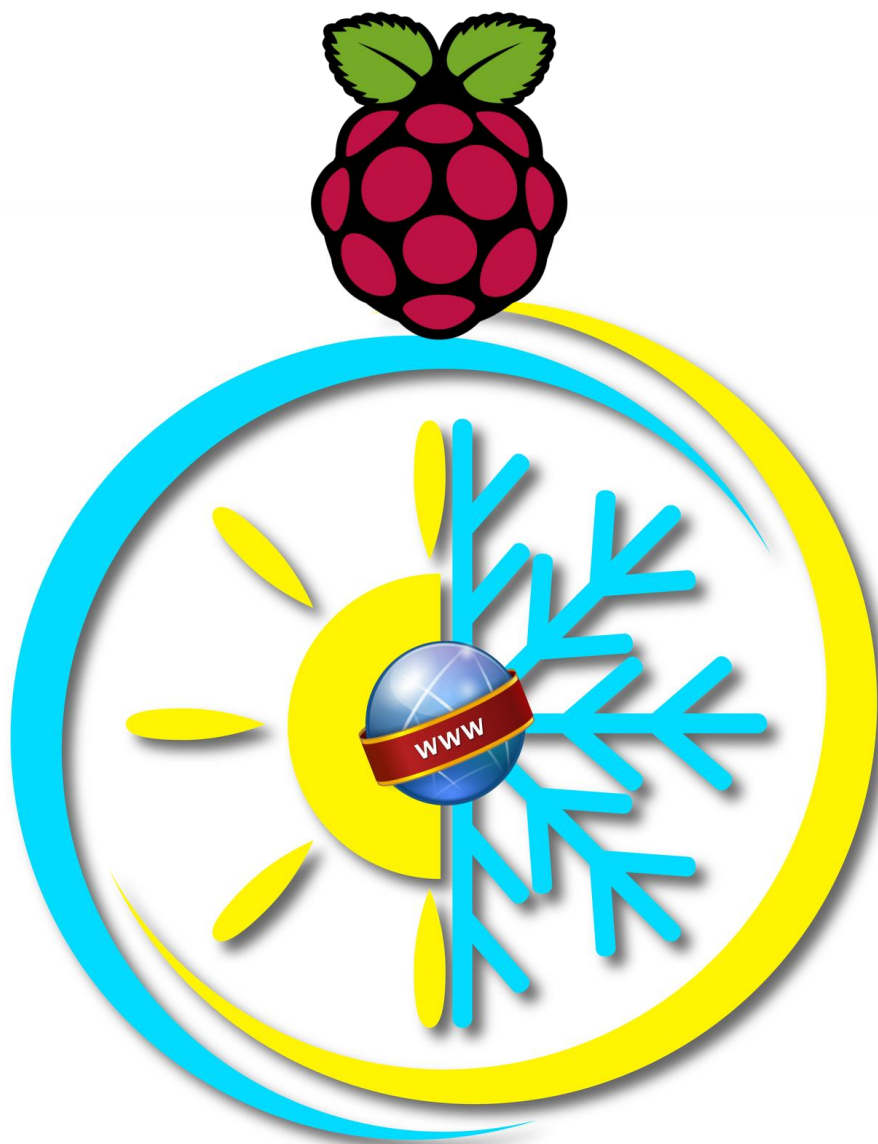


# Sistema de Climatização Inteligente Controlado Remotamente



**Trabalho Elaborado por:**

Ângelo Neves Bastos Nº 28910

António Silva Nº 26800

Marco Silva Nº 28455

Sebastien Caussin Nº29981

**ÍNDICE**

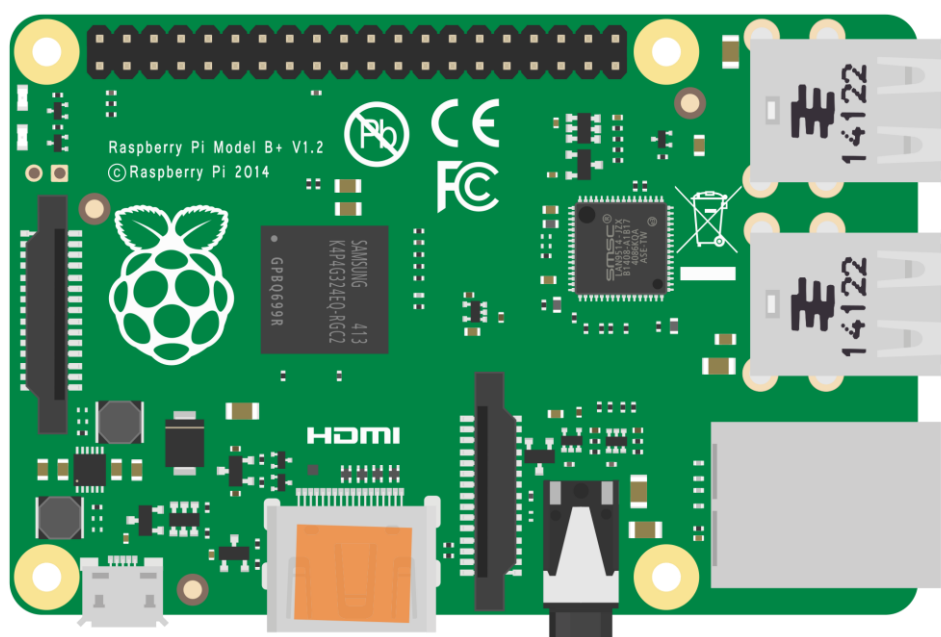
RESUMO .....	3
INTRODUÇÃO .....	4
COMPONENTES DO PROJETO .....	4
HARDWARE .....	4
SOFTWARE.....	5
ETAPAS .....	6
1ª FASE .....	6
2ª FASE .....	7
3ª FASE .....	11
4ª FASE .....	16
5ª FASE .....	19
6ª FASE .....	21
7ª FASE .....	25
8ª FASE .....	26
9ª FASE .....	31
CONCLUSÃO .....	34
SOFTWARE QUE FOI USADO NA CONSTRUÇÃO DA APLICAÇÃO.....	34
INTERFACE GRÁFICA DA APLICAÇÃO.....	35
REFERÊNCIAS.....	36

## RESUMO

O sistema tem como principal objetivo poder controlar um sistema de climatização numa casa à distancia. Ou seja, através de um smartphone ou computador com acesso à Internet, acedendo a um endereço IP, é possível utilizar a aplicação.

A aplicação está implementada num Raspberry Pi, e este age como servidor, de forma a fornecer à aplicação conectividade com o exterior. Dentro da aplicação, que foi desenvolvida utilizando HTML, PHP, AJAX, JavaScript, C e Bootstrap, está o código da aplicação e as configurações corretas do Raspberry Pi, bem como a conectividade dos sensores e respetivos LEDs ao GPIO do Raspberry Pi.

Resumindo então, um Raspberry Pi torna-se num servidor Web e disponibiliza a aplicação, com a possibilidade de acesso remoto. Permite controlar os LEDs correspondentes ao aquecimento e ao arrefecimento, assim como obter leituras de temperatura do ambiente em que se encontra, sendo possível controlar a temperatura do ar condicionado manualmente e/ou automaticamente.



## INTRODUÇÃO

Para a concretização do projeto, inicialmente tivemos de preparar o Raspberry Pi. Nomeadamente o sistema operativo, as devidas configurações de conectividade e acesso remoto, assim como os serviços Web que o tornam num servidor. Tivemos ainda de configurar as respetivas ligações dos LEDs, e do sensor de temperatura. Por fim, procedemos à criação e implementação da aplicação Web no Raspberry Pi, bem como a importação de bibliotecas, e a programação da aplicação e dos respetivos scripts de código.

De seguida, teremos os componentes do projeto e todas as etapas detalhadas, até à etapa final do projeto.

## COMPONENTES DO PROJETO

### HARDWARE

- Raspberry Pi 2 e transformador/power bank USB;
- Inicialmente, um monitor com ligação HDMI, cabo HDMI, teclado e rato USB;
- Cartão MicroSD 32GB;
- Leitor Multi-Card USB;
- Modem 4G/Wi-Fi;
- Pen Wi-Fi;
- 2 LEDs e 2 pares de cabos de ligação;
- Sensor de temperatura e humidade DHT11, e 3 cabos de ligação;

## SOFTWARE

- NOOBS 2.0 (Linux Debian);
- Biblioteca DHT11 (Porrey 2015)
- SDFormatter 4
- WinSCP 5.9.2 Portable
- PuTTY
- IPscan24
- Win32DiskImager
- VNC Viewer
- Notepad++
- IntelliJ
- Xara Web Designer
- Dreamweaver CS6

## ETAPAS

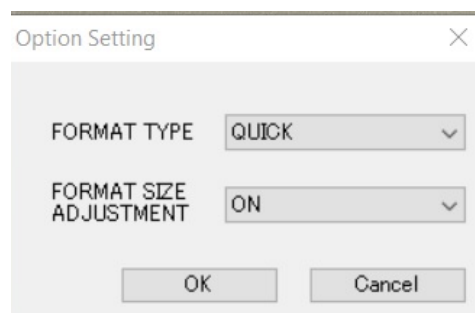
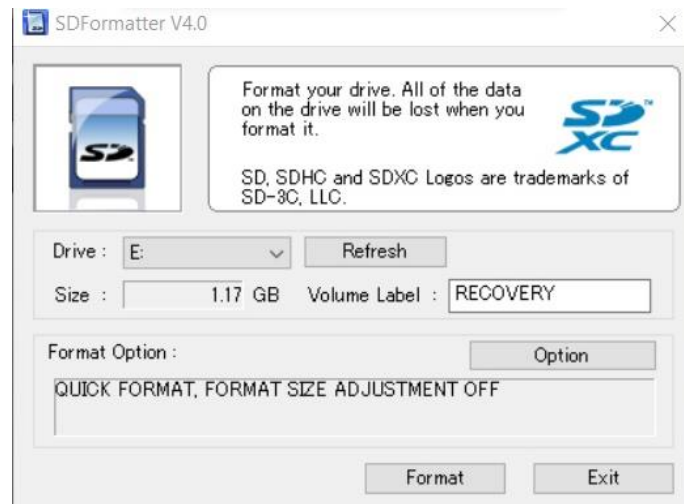
### 1ª FASE

#### Preparação do cartão SD do Raspberry PI.

#### 1º PASSO

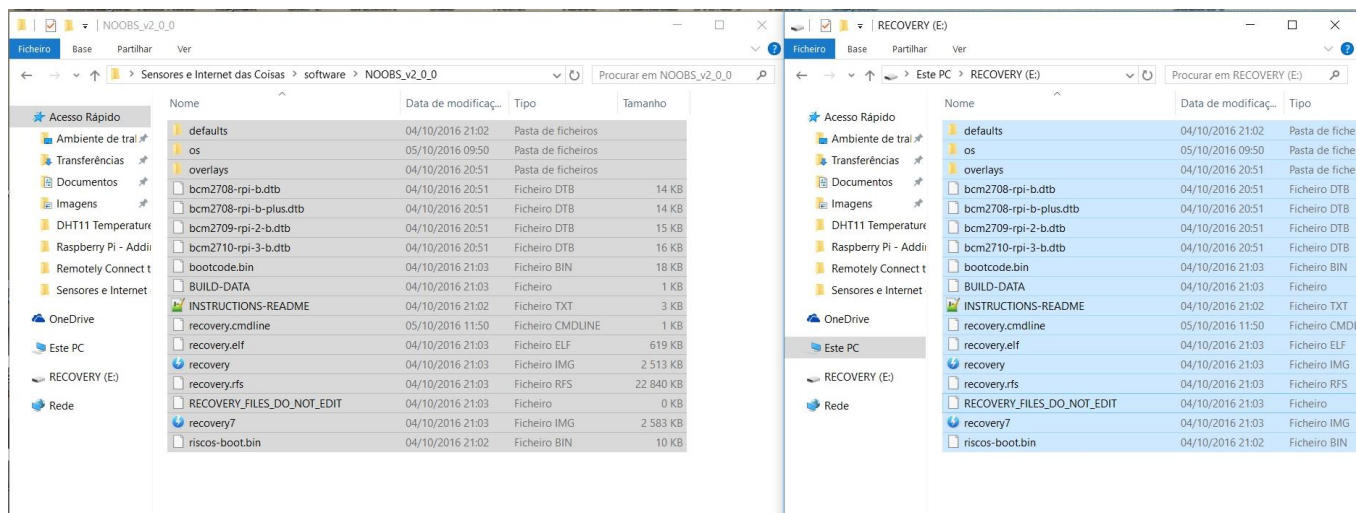
Inserir o cartão SD de 32GB no leitor de cartões, e ligar o leitor ao computador com sistema operativo da família Windows.

De seguida formatar o cartão de memória como demonstram as seguintes imagens.



## 2º PASSO

Copiar o total do conteúdo da pasta NOOBS v2.0, para a raiz do cartão de memória.



## 2ª FASE

**Arranque do NOOBS 2.0, seguido da instalação do SO Linux Debian, aprovisionamento e preparação do acesso remoto do Raspberry Pi, e ligação à rede WIFI.**

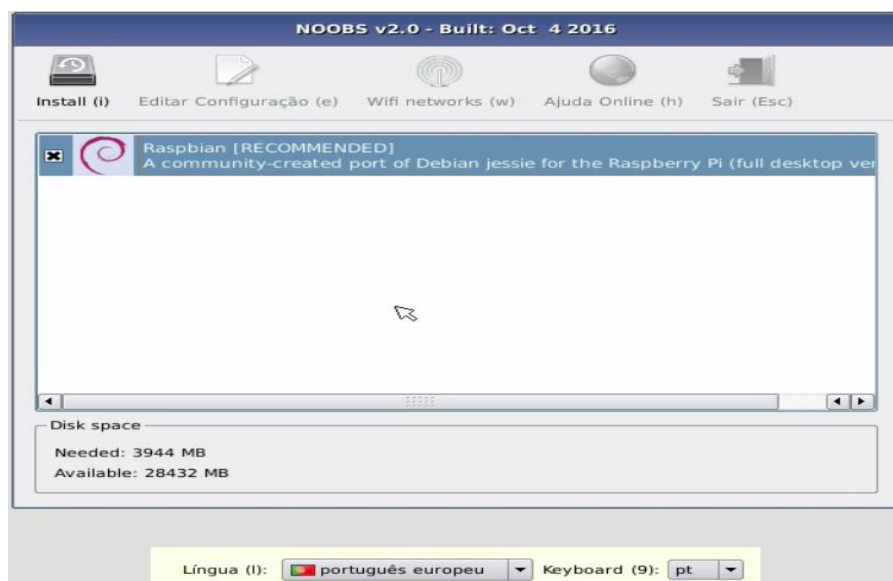
## 1º PASSO

Retirar o cartão SD do leitor de cartões, e inserir o cartão no Raspberry Pi.

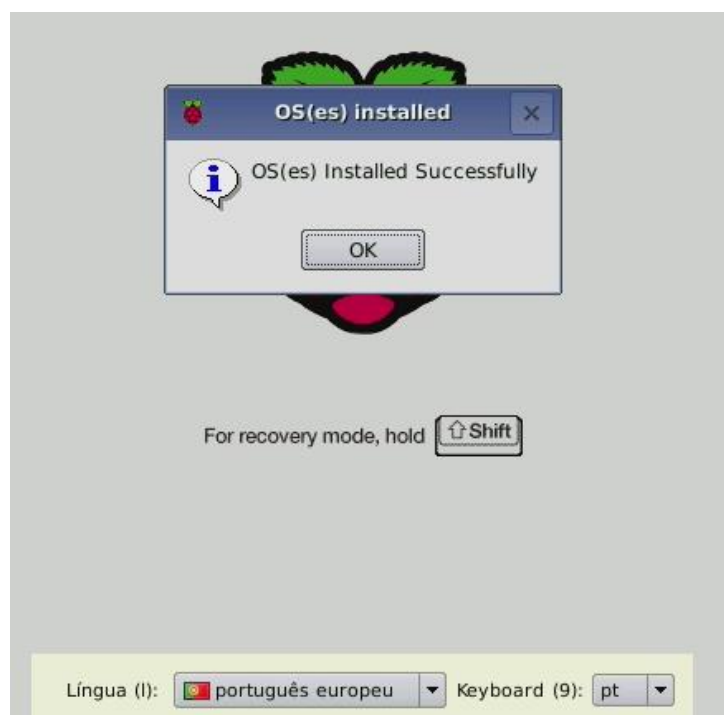
Ligar o Raspberry Pi ao monitor por cabo HDMI, e ligar ao Raspberry Pi um teclado USB e rato USB, e uma pen Wi-Fi compatível com o sistema Linux Debian.

Por fim, ligar ao raspberry pi o carregador, e ligar o mesmo à corrente elétrica. Alternativamente, ligar ao Raspberry Pi um power bank USB.

Ao arrancar, aparecerá um menu no qual se poderá selecionar o sistema operativo, e escolher o idioma do mesmo. Devemos clicar em “install” para dar início à instalação.



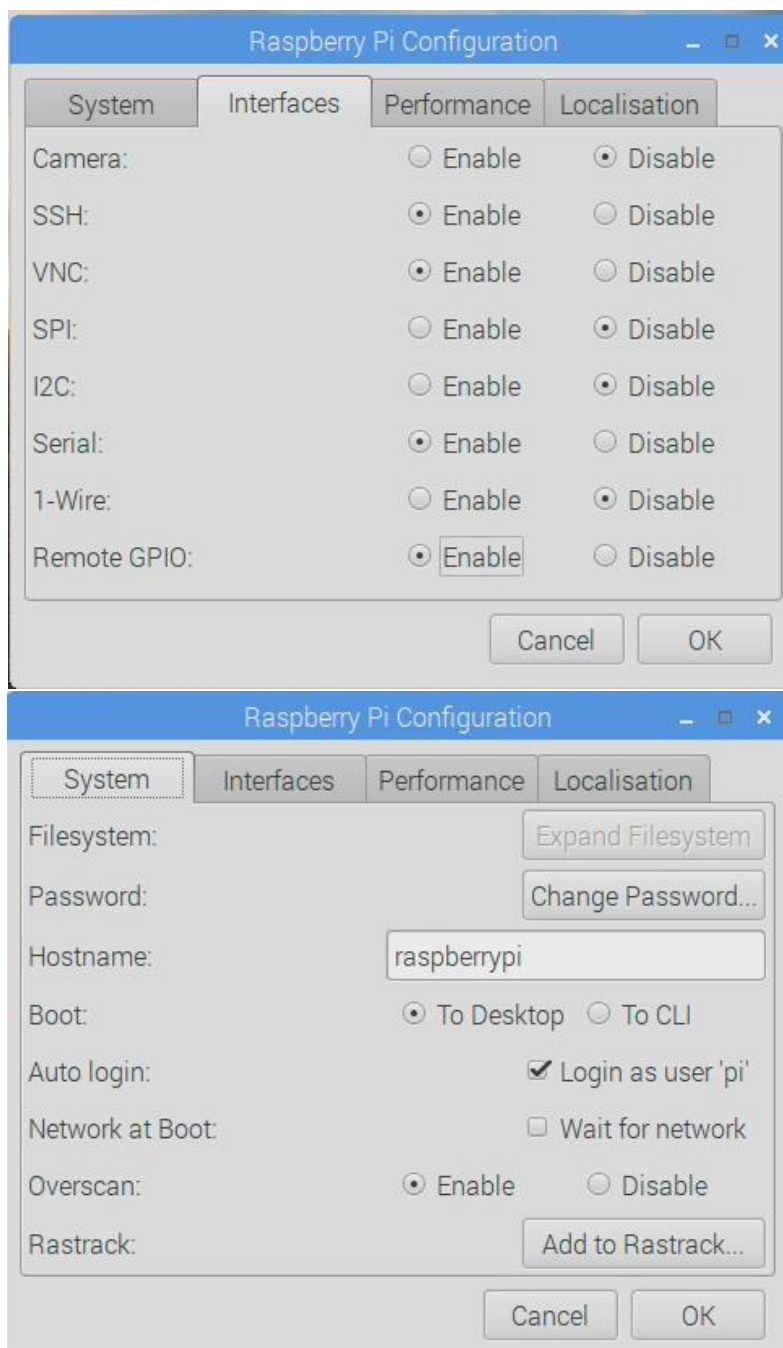
Depois de alguns minutos de espera, o sistema operativo indicará que a instalação foi bem sucedida.





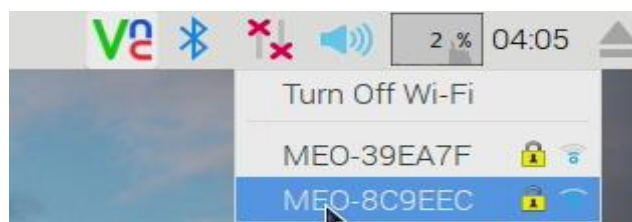
**2º PASSO**

Após o sucesso da instalação do sistema operativo, é necessário ir ao menu da configuração do sistema do Raspberry Pi, de forma a ativar o serviço SSH e o VNC nas interfaces. Para verificar que as definições encontram-se como desejadas, devemos ir a “System”.



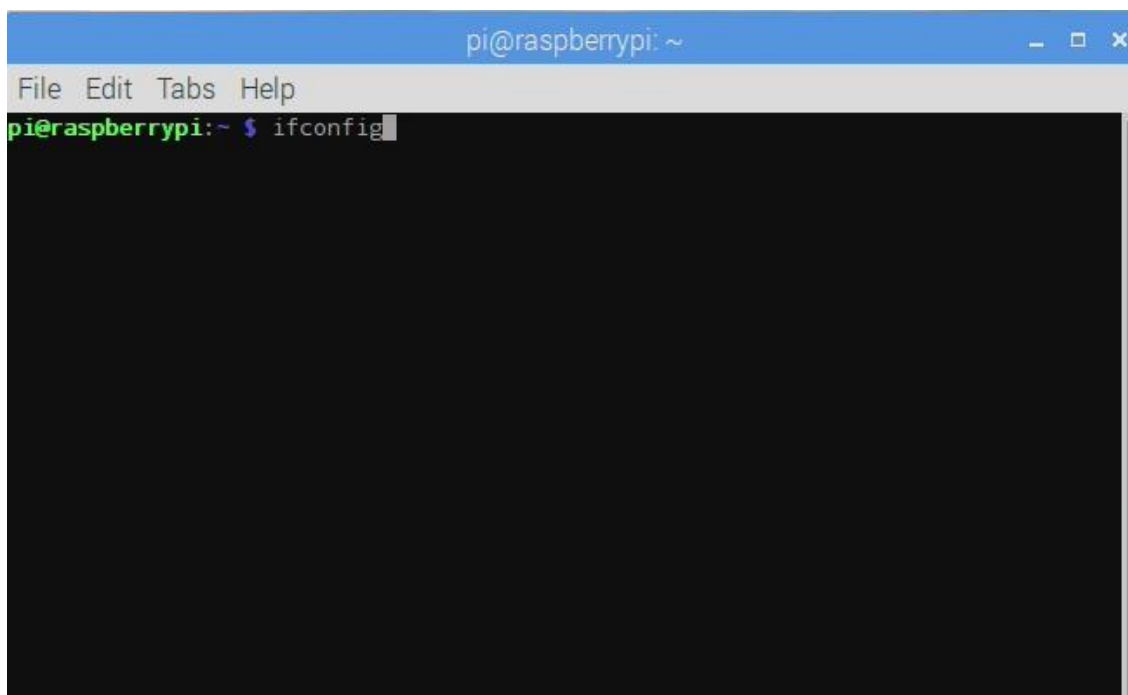
De seguida, seleccionamos o ícone Wi-Fi, e depois a rede à qual nos pretendemos ligar. Após a autenticação na rede, visualizamos o IP que foi atribuído automaticamente ao Raspberry Pi.

## 3º PASSO



## 4º PASSO

Para verificarmos o IP do Raspberry Pi, devemos abrir o terminal.



Seguidamente, digitamos o comando “ifconfig”.

```
pi@raspberrypi:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:5a:85:e7
          inet6 addr: fe80::b5c:aa78:70f4:f091/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:353 errors:0 dropped:0 overruns:0 frame:0
          TX packets:353 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:28888 (28.2 KiB)  TX bytes:28888 (28.2 KiB)

wlan0     Link encap:Ethernet  HWaddr 74:da:38:5e:17:8f
          inet addr:192.168.0.110  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::575b:accf:dd9f:48d3/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:82 errors:0 dropped:0 overruns:0 frame:0
          TX packets:90 errors:0 dropped:1 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:16062 (15.6 KiB)  TX bytes:14315 (13.9 KiB)

pi@raspberrypi:~$
```

Agora poderemos observar que o IP desta máquina é: 192.168.0.110.

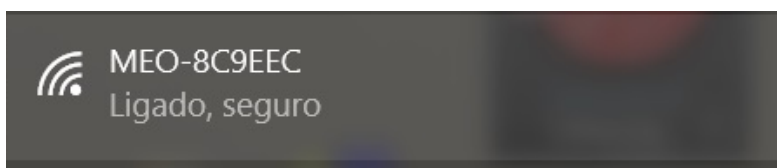
Este IP é o endereço que pelo qual iremos aceder remotamente ao Raspberry Pi.

## 3ª FASE

**Preparação de um computador com o software de acesso remoto (SSH -> PuTTY; FTP -> WinSCP; Remote Desktop GUI -> VNC-Viewer)**

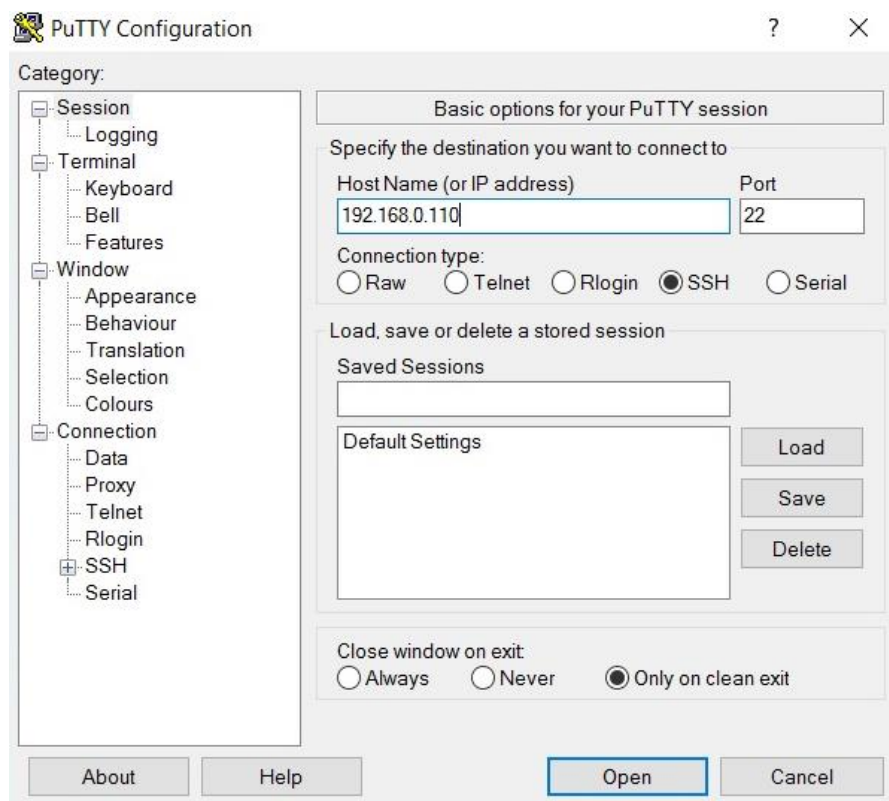
## 1º PASSO

Ligar o computador à mesma rede Wi-Fi em que o Raspberry Pi se encontra.



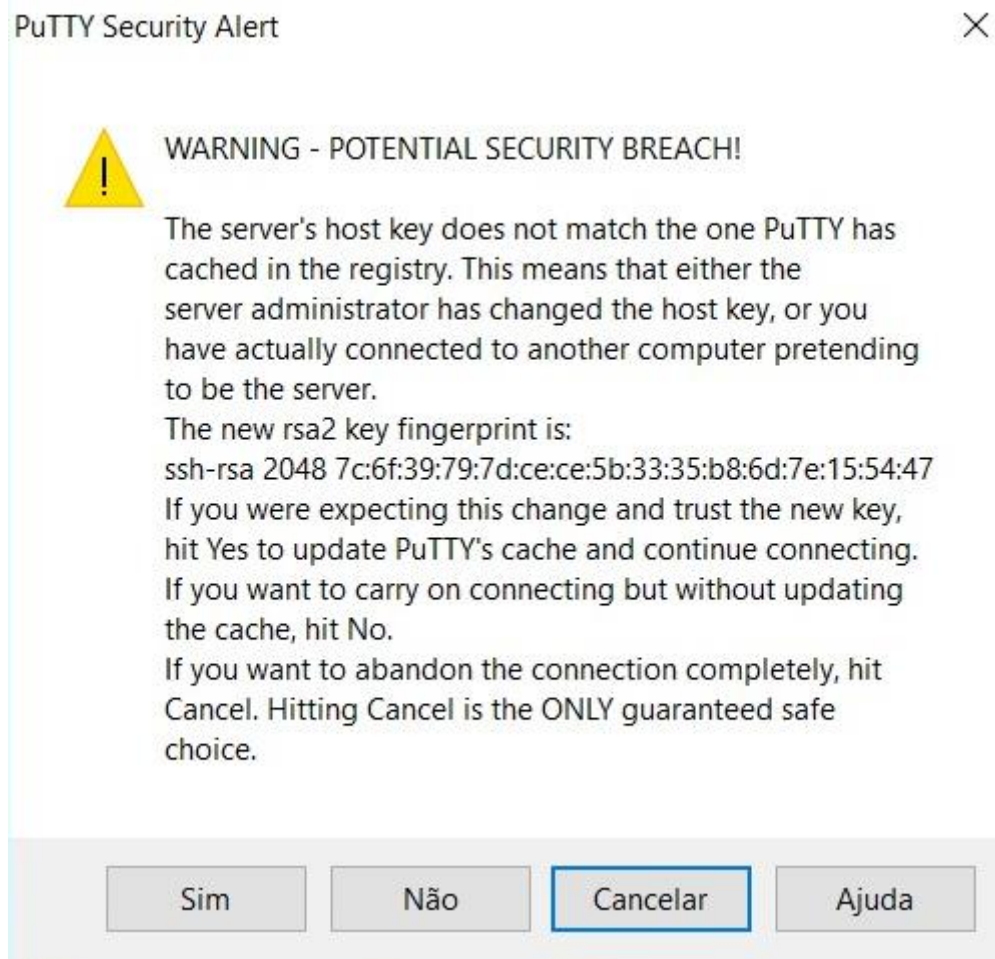
**2º PASSO**

Iniciar o **PuTTY**, inserir o IP do Raspberry Pi, e clicar em **OPEN**.



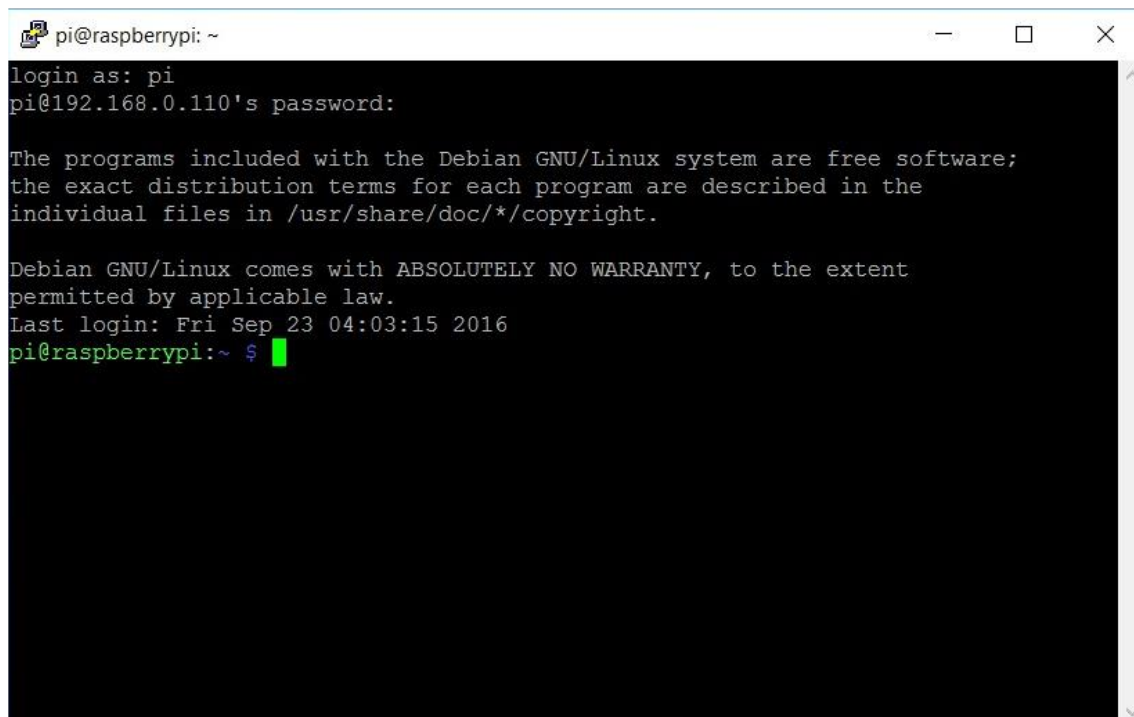
## 3º PASSO

Clicar em sim para aceitar a nova chave de segurança.



## 4º PASSO

Introdução do **username** e **password**, que neste caso irá ser “pi” e “raspberrypi”, e pressionar na tecla *ENTER* do teclado.



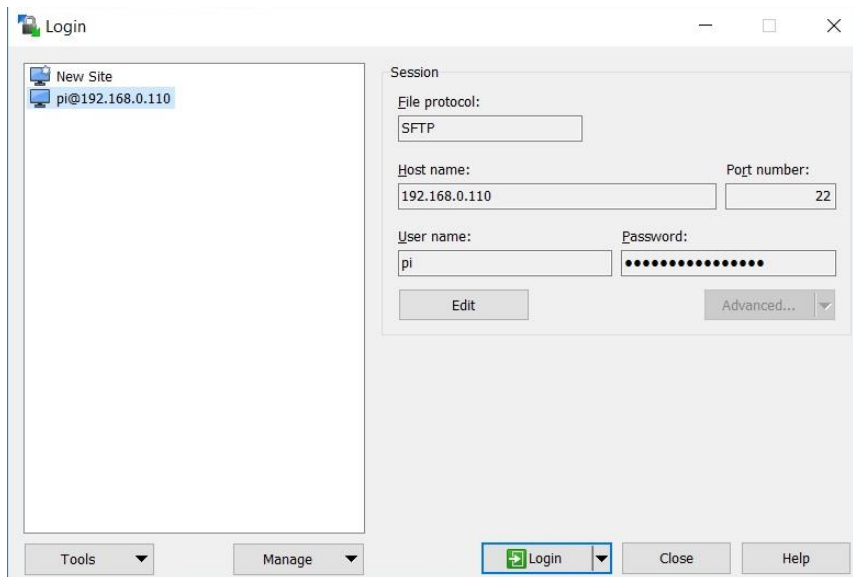
```
pi@raspberrypi: ~
login as: pi
pi@192.168.0.110's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

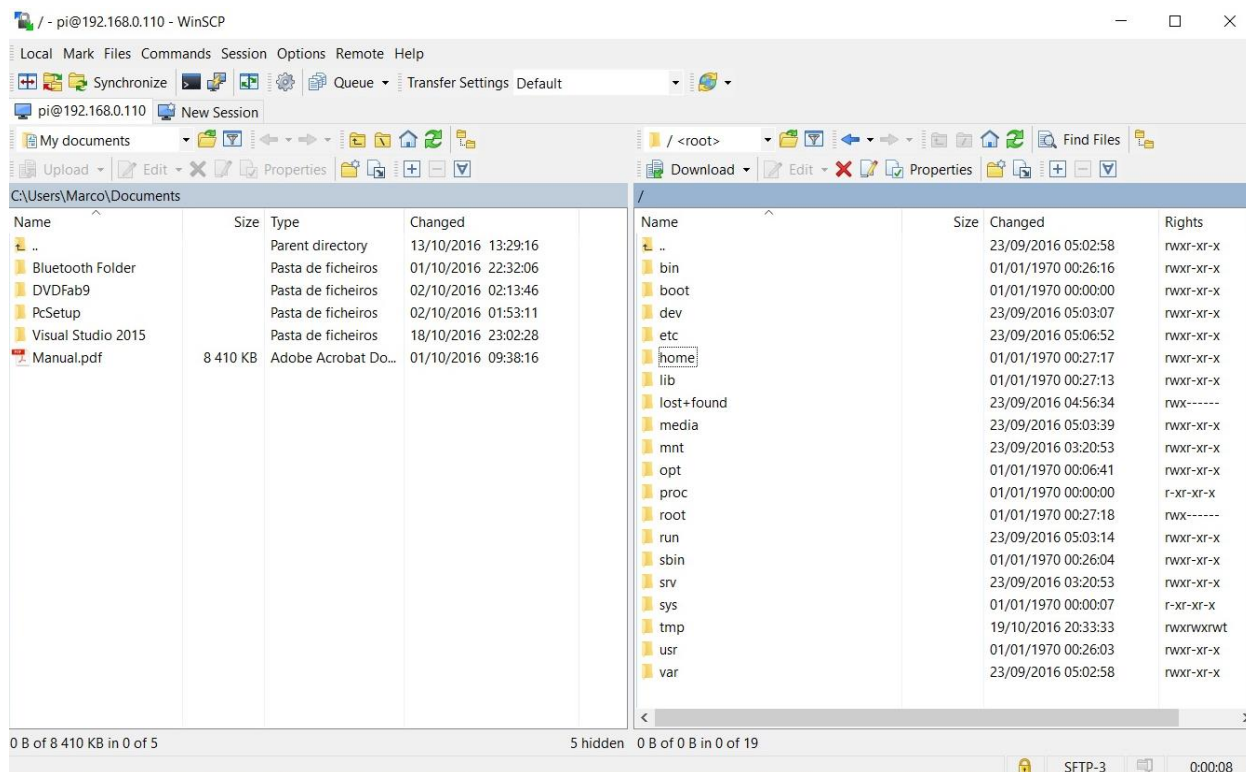
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Sep 23 04:03:15 2016
pi@raspberrypi:~ $
```

**5º PASSO**

No computador, iniciar o WinSCP e inserir o **Host name**, **Username** e **Password**, que neste caso serão: 192.168.0.110, pi , raspberry.



Depois do login efetuado com sucesso, irá ser aberta a janela da imagem abaixo, na qual conseguiremos transferir ficheiros do computador para o Raspberry Pi, e vice-versa.





**4ª FASE****Instalação dos serviços Web no Raspberry Pi (Apache e PHP)****1º PASSO**

Devemos atualizar o repositório de pacotes primeiramente:

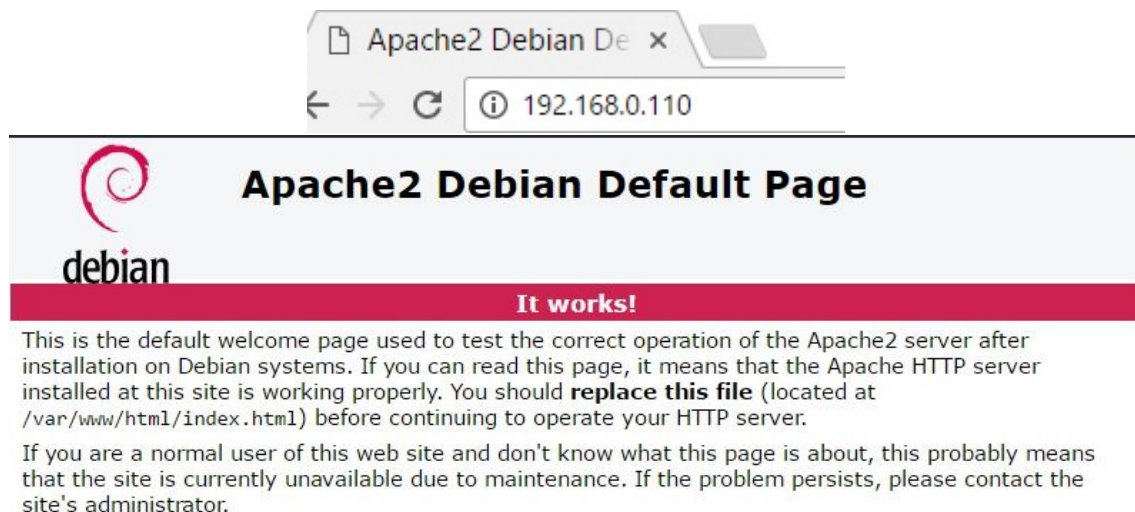
```
pi@raspberrypi:~ $ sudo apt-get update
```

**2º PASSO**

De seguida instalamos o **Apache**:

```
pi@raspberrypi:~ $ sudo apt-get install apache2 -y
```

Depois da instalação do Apache ter sido concluída com sucesso, abrimos um browser, e na barra de endereço colocamos o IP do Raspberry Pi. Neste caso, esse IP será 192.168.0.110, e a página de teste do Apache aparecerá no browser, como demonstra a imagem abaixo.





**3º PASSO**

Primeiro deslocamo-nos para a pasta “html”, que se encontra dentro da pasta “www”, e esta ultima encontra-se na pasta “var”.

De seguida adicionamos ao **Apache** o módulo **PHP**. Para isso instalamos o modulo com o seguinte comando da imagem seguinte.

```
pi@raspberrypi:/var/www/html $ sudo apt-get install php5 libapache2-mod-php5 -y
```

Seguidamente apagamos o ficheiro *index.html*.

Posteriormente criamos um novo ficheiro *index.php*, de forma a testar a funcionalidade nova.

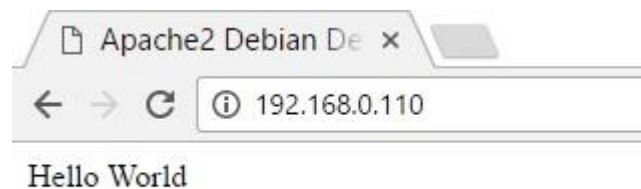
```
pi@raspberrypi:/var/www/html $ sudo nano index.php
```



```
GNU nano 2.2.6 File: index.php

<?php echo "Hello World"; ?>
```

Após isto, testamos o resultado no browser.



**4º PASSO**

Atribuímos permissões de acesso e escrita máximas para todos os utilizadores, para as pastas **html**, que está dentro da pasta **www**, e esta que está contida na pasta **var**.

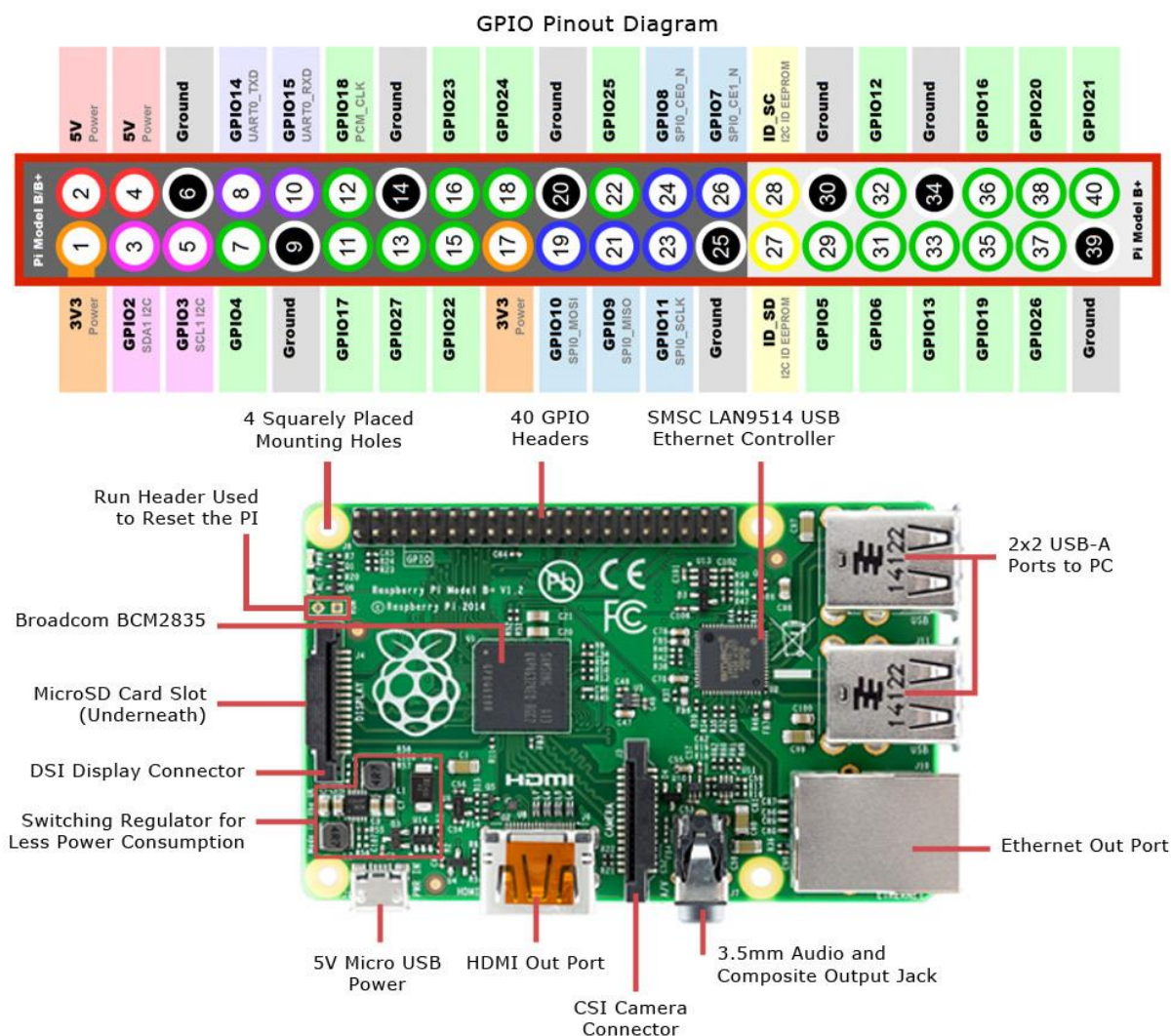
```
pi@raspberrypi: /var/www/html
pi@raspberrypi:/ $ sudo chmod 777 var
pi@raspberrypi:/ $ ls
bin  dev  home  lost+found  mnt  proc  run  srv  tmp  var
boot  etc  lib  media      opt  root  sbin  sys  usr
pi@raspberrypi:/ $ cd var
pi@raspberrypi:/var $ sudo chmod 777 www
pi@raspberrypi:/var $ ls
backups  cache  lib  local  lock  log  mail  opt  run  spool  swap  tmp  www
pi@raspberrypi:/var $ cd www
pi@raspberrypi:/var/www $ sudo chmod 777 html
pi@raspberrypi:/var/www $ ls
html
pi@raspberrypi:/var/www $ cd html
pi@raspberrypi:/var/www/html $
```

## 5ª FASE

Interpretação do diagrama GPIO do Raspberry Pi, e efetuar as respetivas ligações do sensor de temperatura e humidade (DHT11), e ligações dos LEDs (LED verde = arrefecimento; LED vermelho = aquecimento)

## 1º PASSO

Procedemos à interpretação do diagrama GPIO do Raspberry Pi, e efetuamos as respetivas ligações dos LEDs (LED verde = arrefecimento; LED vermelho = aquecimento), e ligações do sensor de temperatura e humidade (DHT11).



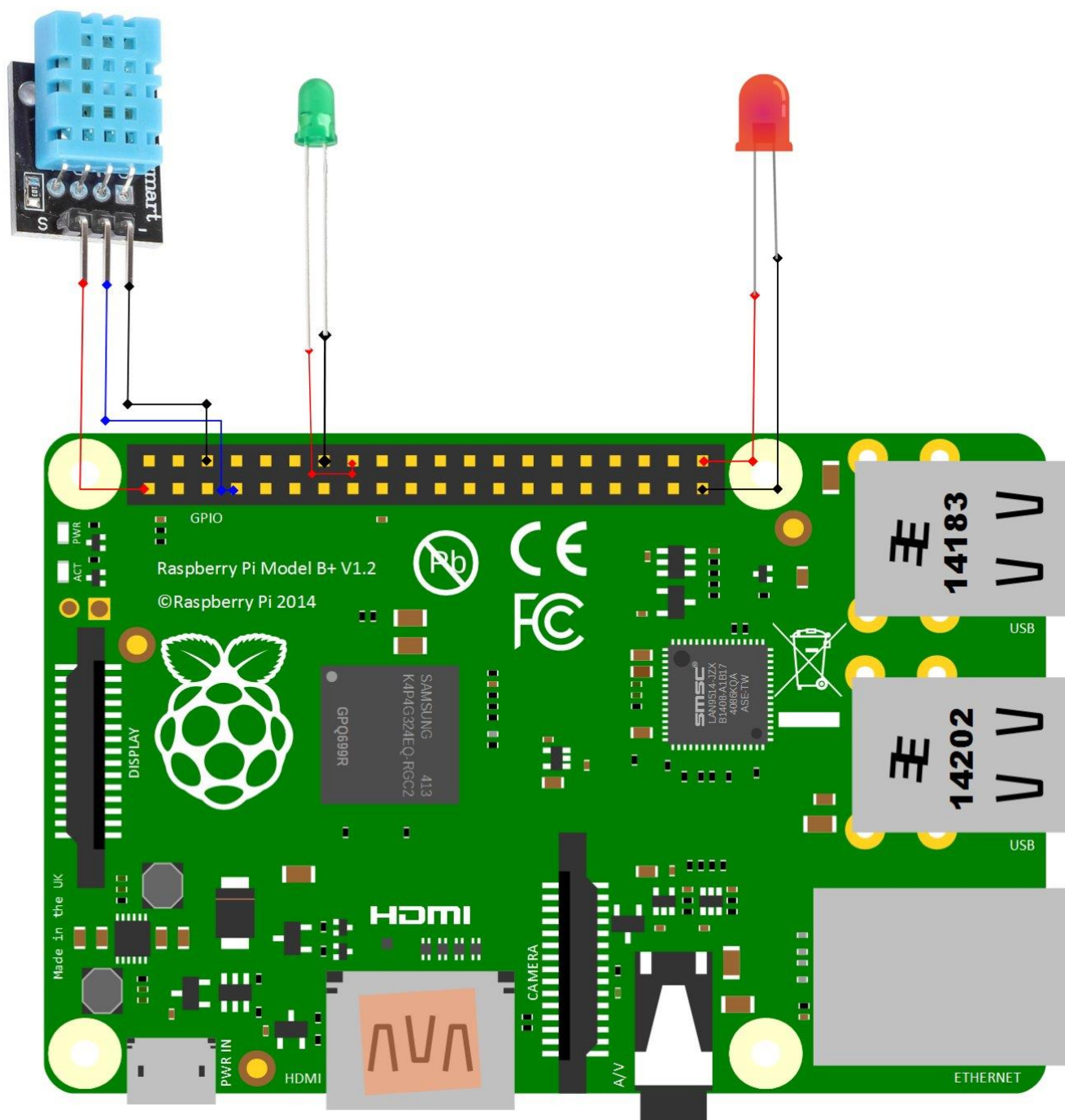
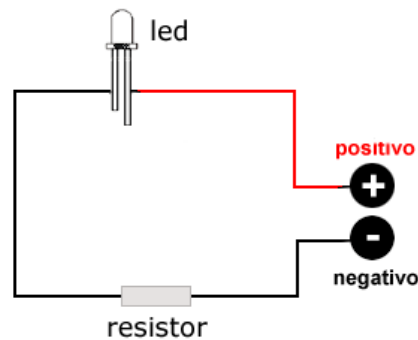


Diagrama de Polaridade de um LED



## 6ª FASE

Importação de bibliotecas do DHT11, e criação do script de leitura de temperatura, e a compilação e execução do mesmo script, e a criação do script de leitura da temperatura em PHP.

### 1º PASSO

Para a importação de bibliotecas do DHT11:

```
pi@raspberrypi:~ $ git clone git://git.drogon.net/wiringPi
```

### 2º PASSO

Executamos o build do pacote:

```
pi@raspberrypi:~ $ cd wiringPi
pi@raspberrypi:~/wiringPi $ ./build
```

### 3º PASSO

Editamos o script **Dht11\_Speed.c** original, adaptando-o consoante a necessidade da nossa aplicação.



#### 4º PASSO

Compilamos para um ficheiro executável o ficheiro que contem o algoritmo de leitura da temperatura:




```
pi@raspberrypi:~/wiringPi $ gcc -oteste Dht11_Speed.c -L/usr/local/lib -lwiringPi -std=c99 -lm
```

Verificação do ficheiro compilado com o nome "teste":

```
pi@raspberrypi:~/wiringPi $ ls
build      debian-template  examples  newVersion  README.TXT
COPYING.LESSER  devLib          gpio      People      VERSION
debian     Dht11_Speed.c   INSTALL   pins        wiringPi
pi@raspberrypi:~/wiringPi $ gcc -oteste Dht11_Speed.c -L/usr/local/lib -lwiringPi -std=c99 -lm
pi@raspberrypi:~/wiringPi $ ls
build      debian      devLib      examples  INSTALL   People  README.TXT  VERSION
COPYING.LESSER  debian-template  Dht11_Speed.c  gpio      newVersion  pins    teste      wiringPi
pi@raspberrypi:~/wiringPi $
```

#### 5º PASSO

Movemos o ficheiro **teste** para dentro da pasta **html**.

/var/www/html			
Name	Size	Changed	Rights
		19/10/2016 20:56:15	rw-rw-rw-r
 index.php	1 KB	19/10/2016 21:12:57	rw-r--r--
 teste	8 KB	19/10/2016 21:37:08	rw-r--r--

#### 6º PASSO

Demos permissões de acesso máximas ao ficheiro **teste**, e realizamos um teste ao ficheiro.

```
pi@raspberrypi:/var/www/html $ chmod 777 teste
pi@raspberrypi:/var/www/html $ sudo ./teste
20pi@raspberrypi:/var/www/html $
```

## 7º PASSO

Demos permissões de acesso totais à pasta **www** para todos os utilizadores, e removemos a encriptação por password.

Adicionamos a ultima linha: `www-data ALL =(ALL) NOPASSWD: ALL`

```
pi@raspberrypi:/var/www/html $ sudo nano /etc/sudoers
GNU nano 2.2.6 File: /etc/sudoers

# Allow members of group sudo to execute any command
%sudo  ALL=(ALL:ALL) ALL





# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d

pi ALL=(ALL) NOPASSWD: ALL
www-data ALL =(ALL) NOPASSWD: ALL
```

**8º PASSO**

Importamos o ficheiro “leituratemperatura.php”, que contem o script de leitura da temperatura, e executamos o teste no browser.

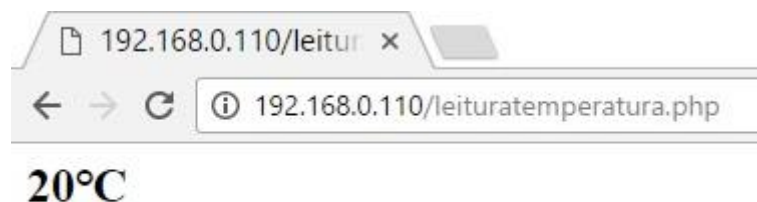
/var/www/html			
Name	Size	Changed	Rights
		19/10/2016 20:56:15	rw-rw-rw-
 index.php	1 KB	19/10/2016 21:12:57	rw-r--r--
 leituratemperatura.php	1 KB	19/10/2016 21:45:56	rw-r--r--
 teste	8 KB	19/10/2016 21:37:08	rw-rw-rw-

## SCRIPT

```
<?php
$last_line = exec("sudo ./teste", $output, $temp);

echo "<h2>" . $last_line . "&#8451;</h2>";
?>
```

## TESTE





## 7ª FASE

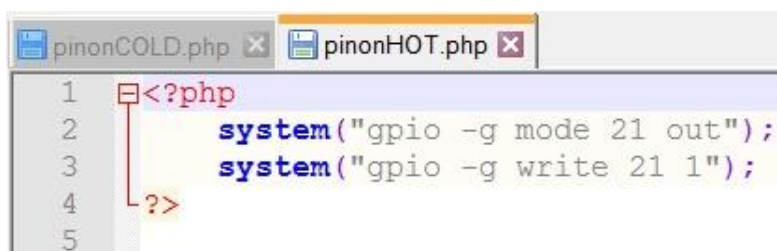
Criação de scripts ON e OFF para cada LED

### 1º PASSO

Criação dos scripts ON e OFF para cada LED:

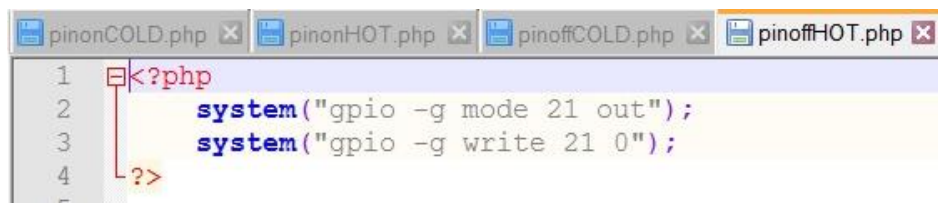
**LED correspondente ao calor**

ON



```
1 <?php
2     system("gpio -g mode 21 out");
3     system("gpio -g write 21 1");
4 ?>
```

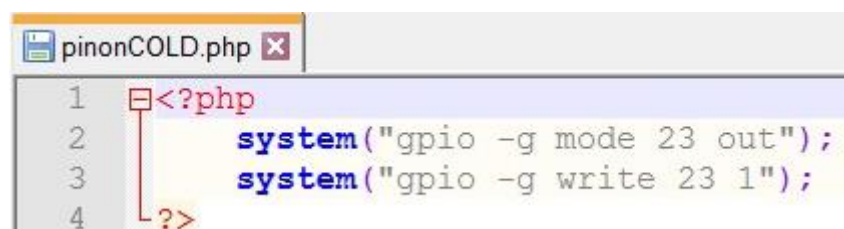
OFF



```
1 <?php
2     system("gpio -g mode 21 out");
3     system("gpio -g write 21 0");
4 ?>
```

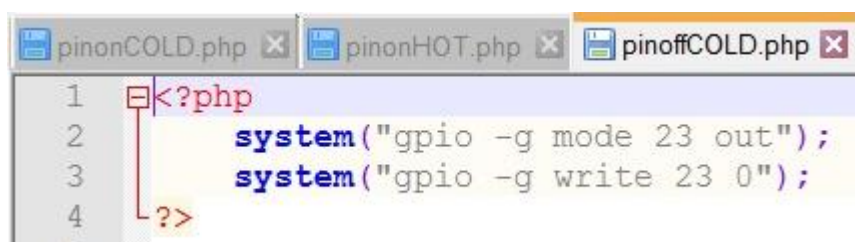
**LED correspondente ao frio**

ON



```
1 <?php
2     system("gpio -g mode 23 out");
3     system("gpio -g write 23 1");
4 ?>
```

OFF



```
1 <?php
2     system("gpio -g mode 23 out");
3     system("gpio -g write 23 0");
4 ?>
```

## 8ª FASE

Instalação de um serviço de monitorização do Raspberry Pi (WebMin), e instalação do serviço de Acesso Gráfico Remoto VNC Server

### 1º PASSO

Comando para download do serviço **webmin** no Raspberry Pi:

```
pi@raspberrypi:~ $ sudo wget http://prdownloads.sourceforge.net/webadmin/webmin-1.580.tar.gz
```

### 2º PASSO

Descompactar o pacote webmin:

```
pi@raspberrypi:~ $ sudo tar -zxvf webmin-1.580.tar.gz
```

### 3º PASSO

Criação de um diretório para o serviço webmin:

```
pi@raspberrypi:~ $ sudo mkdir /var/www/webmin
```

### 4º PASSO

Trocamos para a pasta webmin-1.580:

```
pi@raspberrypi:~ $ cd webmin-1.580/  
pi@raspberrypi:~/webmin-1.580 $
```

### 5º PASSO

Executamos a instalação do serviço **webmin**:

```
pi@raspberrypi:~/webmin-1.580 $ sudo sh setup.sh /var/www/webmin
```

No final da instalação deverá aparecer um resultado semelhante à figura em baixo:

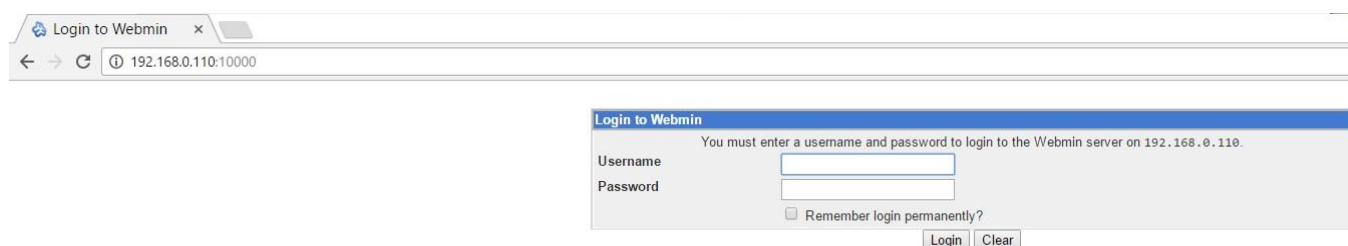
```
*****
Webmin has been installed and started successfully. Use your web
browser to go to

  http://raspberrypi:10000/

and login with the name and password you entered previously.
pi@raspberrypi:~/webmin-1.580 $
```

## 6º PASSO

Inserir no browser o IP do Raspberry Pi, com a porta 10000. Neste caso serão:  
**192.168.0.110:10000**

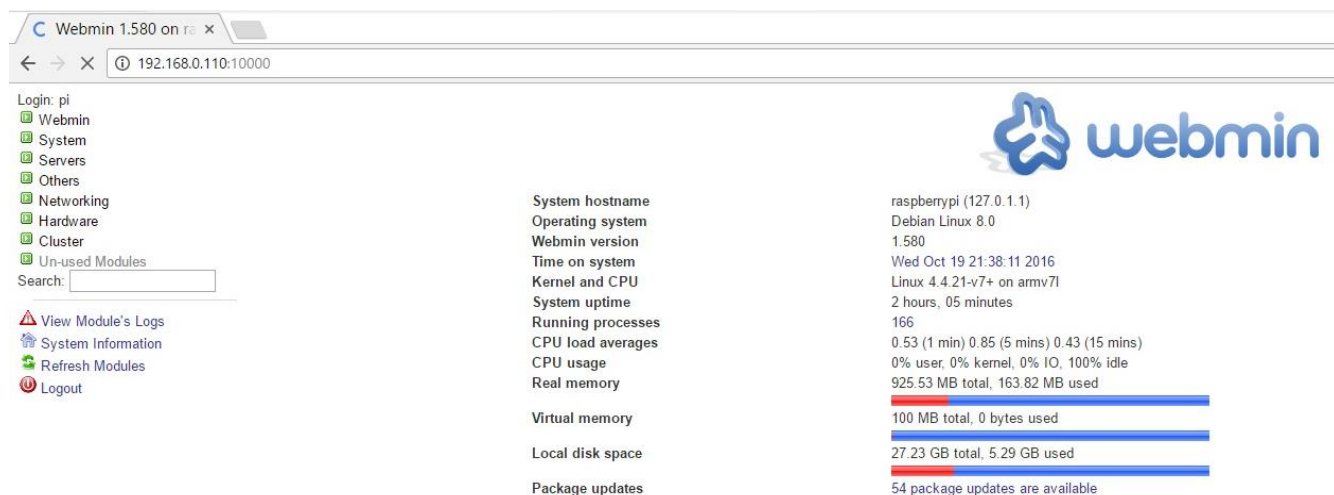


Inserir o username e password que correspondem à conta Linux:

**Username:** pi

**Password:** raspberry

Depois do login ter sido efetuado com sucesso, deverá aparecer uma janela como a imagem a seguir.



## 7º PASSO

Efetuar atualizações aos repositórios de pacotes:

```
pi@raspberrypi:/ $ sudo apt-get update
```

## 8º PASSO

Efetuar upgrade dos pacotes do sistema operativo:

```
pi@raspberrypi:/ $ sudo apt-get upgrade
```

## 9º PASSO

Instalação do serviço VNC-Server:

```
pi@raspberrypi:/ $ sudo apt-get install tightvncserver
```

## 10º PASSO

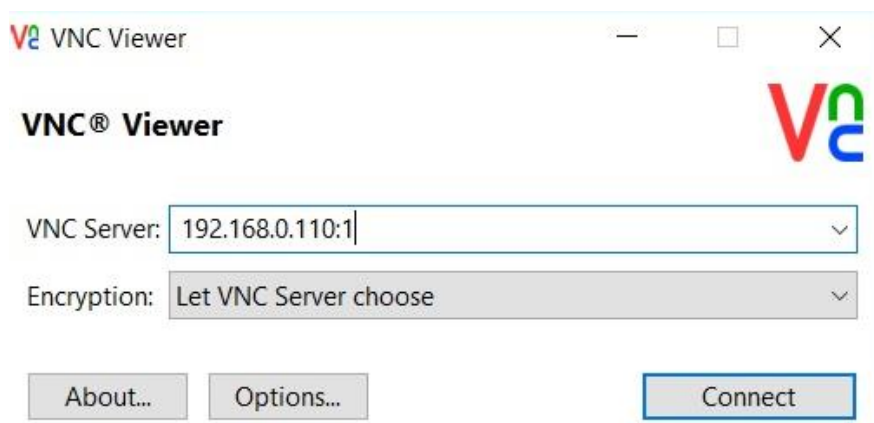
Iniciação do serviço VNC com a resolução pretendida:

```
pi@raspberrypi:/ $ sudo vncserver :1 -geometry 1920x1080 -depth 24
```

**11º PASSO**

Arranque do serviço VNC Server, no computador a partir do qual se pretende aceder remotamente ao Raspberry Pi.

Inserimos o **IP** do Raspberry Pi, seguido de **:1**, que foi a porta atribuída anteriormente.

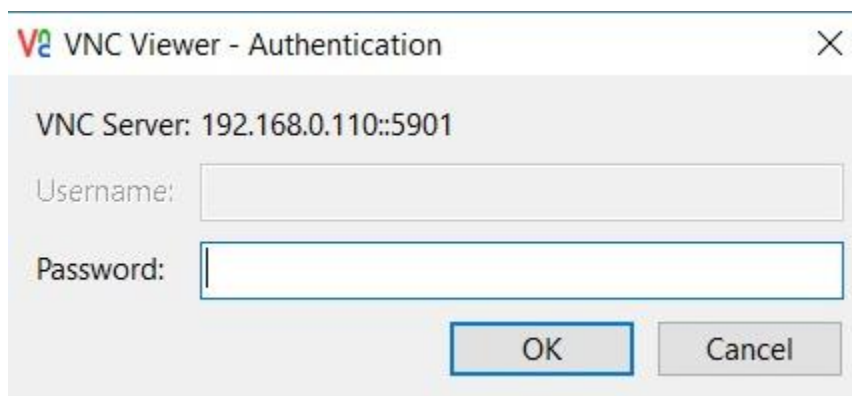
**12º PASSO**

Pressionamos em **Continue** para aceitar a ligação:

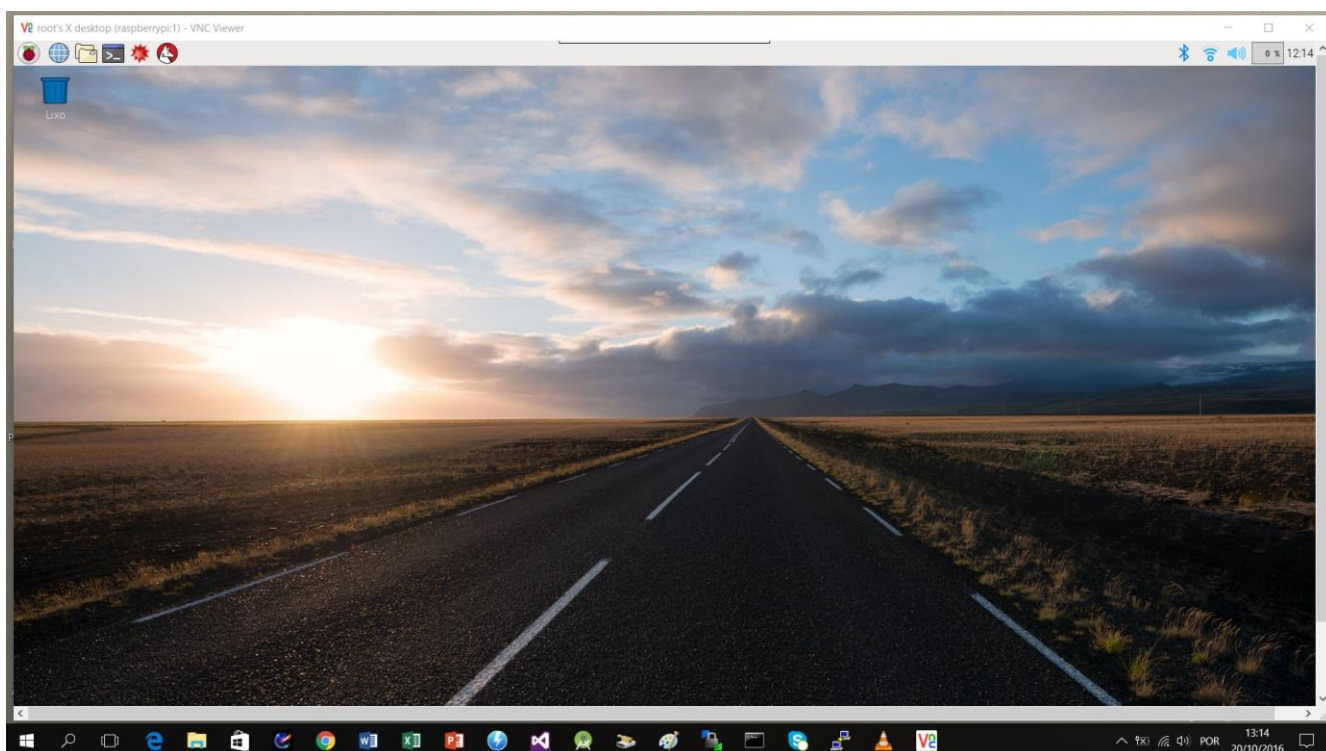


## 13º PASSO

Inserimos a password: **raspberry**



E por fim conseguimos entrar no ambiente gráfico do Raspberry Pi, através do computador remoto.



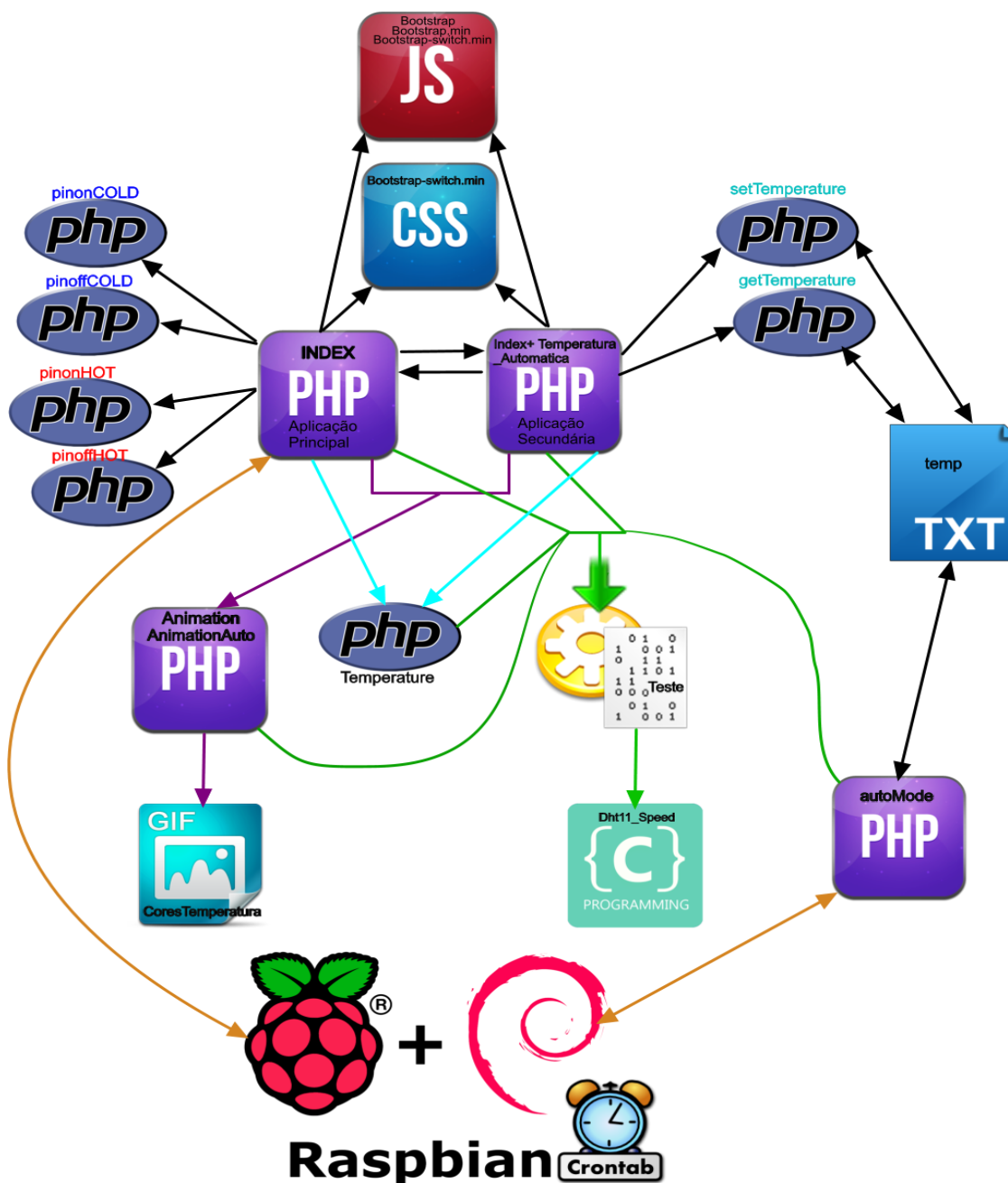


### 9ª FASE

Criação da aplicação Web (Index.php) dentro do servidor Apache do Raspberry Pi, que conterá a ligação de todos os scripts independentes. Estes já criados, de forma a receber a temperatura e poder ligar ou desligar o respetivo LED, consoante as leituras de temperatura que o utilizador interpretar adicionalmente com a aplicação secundária (index Temperatura\_Automatica).




















### 1º PASSO

Diagrama de estrutura da aplicação:



## 2º PASSO

Estrutura dos ficheiros da aplicação:

	CoresTemperatura	03/12/2016 11:50	Pasta de ficheiros	
	css	03/12/2016 11:50	Pasta de ficheiros	
	js	03/12/2016 11:50	Pasta de ficheiros	
	animation	16/11/2016 22:19	PHP Script	1 KB
	animationAuto	16/11/2016 22:30	PHP Script	1 KB
	autoMode	02/12/2016 23:43	PHP Script	1 KB
	Dht11_Speed.c	24/10/2016 21:27	Ficheiro C	4 KB
	getTemperature	02/12/2016 23:11	PHP Script	1 KB
	index	03/12/2016 11:48	PHP Script	4 KB
	index+Temperatura_Automatica	03/12/2016 11:48	PHP Script	4 KB
	pinoffCOLD	24/10/2016 21:49	PHP Script	1 KB
	pinoffHOT	24/10/2016 21:49	PHP Script	1 KB
	pinonCOLD	24/10/2016 21:49	PHP Script	1 KB
	pinonHOT	24/10/2016 21:49	PHP Script	1 KB
	setTemperature	02/12/2016 23:08	PHP Script	1 KB
	temp	03/12/2016 11:14	Ficheiro TXT	1 KB
	Temperature Control.iml	25/10/2016 20:15	Ficheiro IML	1 KB
	temperature	11/11/2016 22:24	PHP Script	1 KB
	temperatureAuto	11/11/2016 22:24	PHP Script	1 KB
	teste	29/10/2016 13:22	Ficheiro	8 KB



**3º PASSO**

Inicialização do **cronjob** para definir a prioridade do ficheiro **autoMode.php**, que contem o algoritmo de inteligência do modo automático da aplicação.

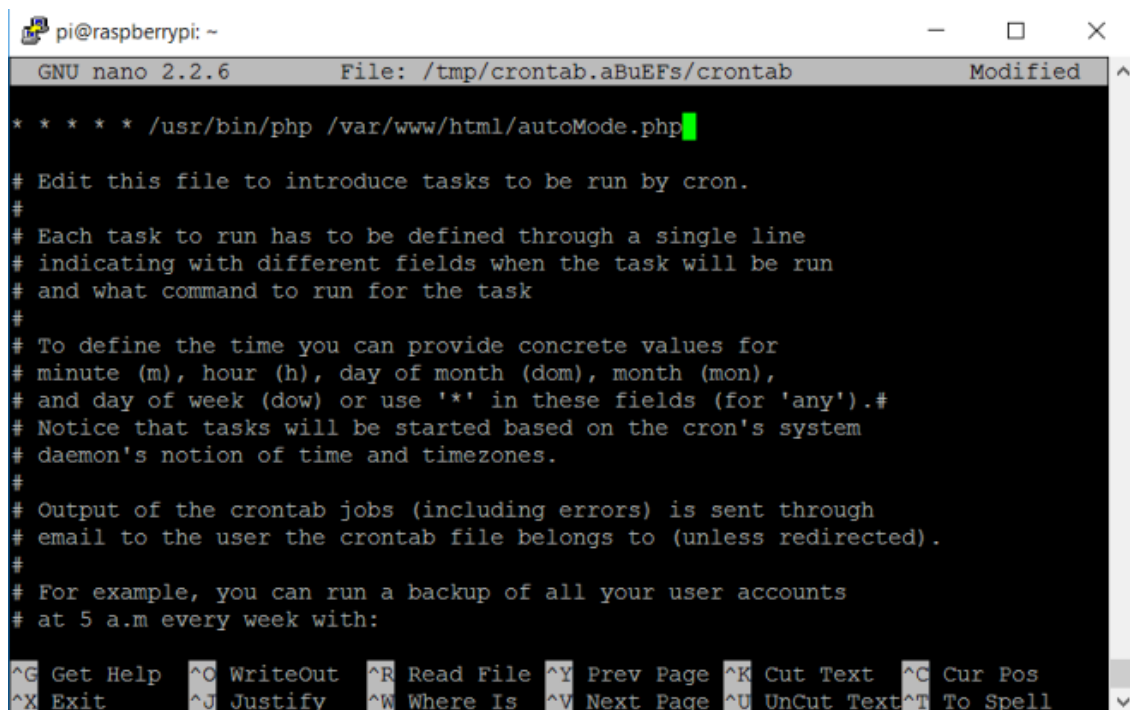
```
pi@raspberrypi:~ $ /usr/bin/php -v
PHP 5.6.27-0+deb8u1 (cli) (built: Oct 24 2016 18:22:27)
Copyright (c) 1997-2016 The PHP Group
Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies
    with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2016, by Zend Technologies
pi@raspberrypi:~ $ crontab -e
no crontab for pi - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/ed
 2. /bin/nano      <---- easiest
 3. /usr/bin/vim.tiny

Choose 1-3 [2]: 2
crontab: installing new crontab
```

**4º PASSO**

Adicionamos a linha: `* * * * * /usr/bin/php /var/www/html/autoMode.php`



```
pi@raspberrypi: ~
GNU nano 2.2.6      File: /tmp/crontab.aBuEFs/crontab      Modified
* * * * * /usr/bin/php /var/www/html/autoMode.php

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
```

**5º PASSO**

Testes de funcionamento do ficheiro **autoMode.php**:

```
1015Aligar friopi@raspberrypi:/var/www $ php html/autoMode.php
161015Aligar quente@raspberrypi:/var/www $ php html/autoMode.php
161017deligar tudopi@raspberrypi:/var/www $
```

**CONCLUSÃO****SOFTWARE QUE FOI USADO NA CONSTRUÇÃO DA APLICAÇÃO**

Inicialmente, depois do sistema operativo NOOBS 2.0 estar instalado, a primeira versão foi construída com o Notepad++ e Xara Web Design 7. Consoante o crescimento da aplicação, usamos DreamWeaver CS6 para ajudar nas ligações e interações do programa. Na fase final usamos o IntelliJ para ilustrar a aplicação. À medida que a aplicação foi sendo desenvolvida, a mesma foi sempre monitorizada e testada diretamente no Raspberry Pi, com a ajuda de software como o PuTTY, WinSCP, IPScan24, VNC Viewer e Webmin.

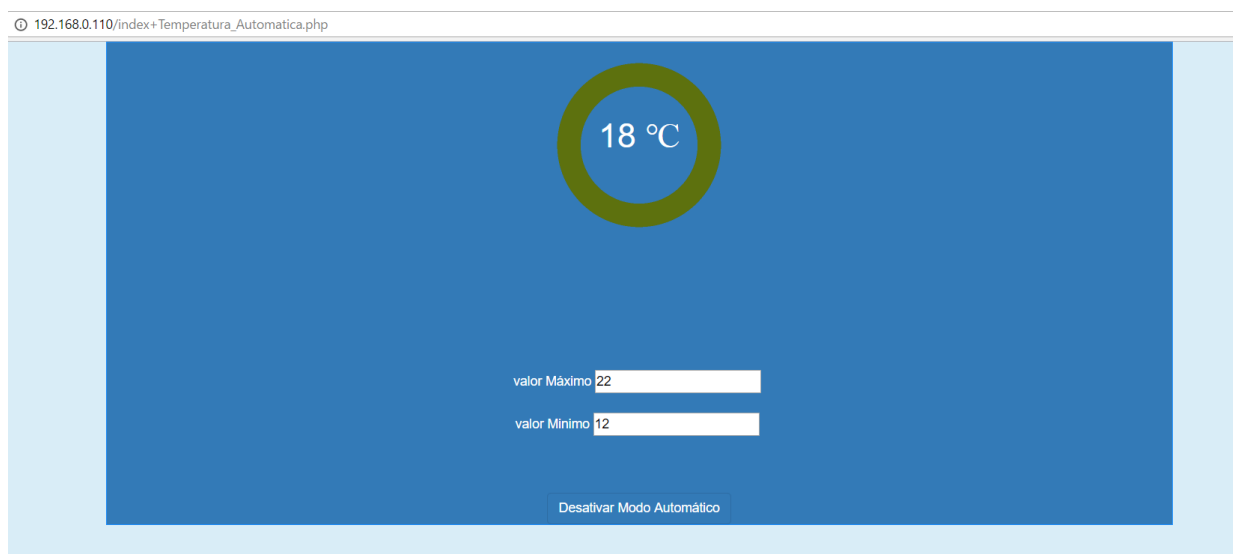


## INTERFACE GRÁFICA DA APLICAÇÃO

### Aplicação Principal



### Aplicação Secundária



## REFERÊNCIAS

<https://www.youtube.com/watch?v=toWBmUsWD6M>

<https://www.youtube.com/watch?v=DPvxsHoD7kc>

<http://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-the-raspberry-pi/>

<http://www.uugear.com/portfolio/read-dht1122-temperature-humidity-sensor-from-raspberry-pi/>

<https://www.youtube.com/watch?v=Q-uFss9tTAc>

<https://www.youtube.com/watch?v=X54qZ7riwI8>

<https://www.youtube.com/watch?v=RyNii3UcHPw>

<https://www.youtube.com/watch?v=a7xkBYlQw1s>

<https://www.youtube.com/watch?v=eP412AudA3g>

<https://www.youtube.com/watch?v=EAMLwbShFFQ>