

# RaspiSecureSystem

Aquest és un projecte per a l'especialitat d'IT de la carrera de *Grau en Enginyeria Informàtica* de la FIB, que consisteix en crear un petit sistema de seguretat utilitzant una Raspberry Pi (a partir d'ara, Raspi), una càmera IP, un Arduino i varis sensors.

Aquest sistema ens permetrà veure què passa a la nostra casa o habitació des d'una pàgina web, i gràcies a un sensor de moviment podrem fer que la càmera faci una foto i ens l'envii per correu o ens avisi amb una notificació al mòbil.

De la mateixa manera farem servir un sensor de temperatura que ens podrà servir com a “detector d'incendis”, i podrem fer també que si passa d'una certa temperatura ens avisi d'alguna manera. Afegirem també l'opció de poder encendre o apagar alguns llums remotament (via web), per si arribem tard a casa poder “fer veure” que hi ha algú.

Tant l'accés a la càmera com controlar els llums també es podrà fer des d'una aplicació mòbil.

Els materials que s'han fet servir han estat:

- Raspberry Pi (model 1 B)
- Càmera IP (Dlink DCS-932L)
- Arduino (Mega)
- Targeta de memòria SD de 32GB classe 10
- Sensor de temperatura
- Sensor de moviment

A continuació es detallen els passos que s'han seguit per tal de poder configurar-ho tot.

## Raspberry Pi

La Raspi és el nucli del nostre sistema, pel que és la que més treball requereix de configuració. Els passos que s'han seguit són:

1. [Instal·lar i preparar el SO](#)
2. [Assignar una IP estàtica \(3 opcions\)](#)
  1. [dhcpcd.conf](#)
  2. [interfaces](#)
  3. [interfaces + wpa\\_supplicant.conf](#)
3. [Canviar el port SSH \(Opcional\)](#)
4. [Generar una clau SSH a un dispositiu “client” \(Opcional\)](#)
5. [Instal·lar i configurar no-ip](#)
6. [Obrir els ports del router](#)
7. [Instal·lar i configurar un servidor FTP \(Opcional\)](#)
8. [Instal·lar un servidor Web \(2 opcions\)](#)
  1. [Apache](#)
  2. [nginx](#)
9. [Instal·lar MySQL y phpMyAdmin](#)
10. [Configurar notifikacions amb l'API de Pushover](#)
11. [Configurar notifikacions per Telegram \(Opcional\)](#)
12. [Enviar notifikacions quan algú fa login](#)

## 1. Instal·lar i preparar el SO

El SO escollit per aquest projecte ha estat **Raspbian** versió “*Pixel*”, una distribució Linux basada en Debian-Jessie i adaptada pel chip ARM de la Raspi. Per instal·lar el SO ens hem servit de l'eina que ens proporciona la comunitat de Raspberry, anomenada *NOOBS*, el que ens permet instal·lar el sistema d'una forma senzilla.

El primer que hem de fer és baixar [NOOBS](#) de la pàgina de Raspberry Pi i descomprimir-lo. A continuació hem de donar format a la targeta; per això podem fer servir l'eina [SDFormatter](#), oficial de SD Association.

Una vegada tenim la targeta formatejada i NOOBS descomprimit, copiem tot el contingut de la carpeta NOOBS dins de la targeta.

**COMPTE:** NO copiar la carpeta com a únic arxiu, sino tots els arxius que trobem dins de la mateixa.

Per fer la instal·lació ens ajudarem d'una pantalla amb entrada HDMI, un teclat i un ratolí. Connectem tots els perifèrics a la Raspi i la connectem a la corrent. Després d'uns moments ens apareixerà una pantalla on ens demana quin sistema volem instal·lar. Seleccionem *Raspbian* i cliquem “Install”. Aquest procés pot tardar entre 20 i 30 minuts.

Quan la instal·lació hagi acabat, reiniciem el sistema i ja podrem arrencar Raspbian normalment.

Des del menú d'aplicacions anirem a *Preferences -> Mouse and Keyboard Settings*, i posarem el teclat en Espanyol (Català).

A continuació anirem a *Preferences -> Raspberry Pi Configuration*, modificarem el **password**, i marcarem l'opció de **Boot** que diu *To CLI*. Això farà que quan arrenqui la Raspi no carregui l'entorn gràfic, ja que normalment no el farem servir i així tenim més recursos disponibles.

En cas que vulguem accedir a l'entorn gràfic (si tenim la Raspi conecteda a una tele o pantalla, per exemple), ho podrem fer amb la comanda *startx*

**startx**

De moment encara mantenim la pantalla, teclat i ratolí, ja que ens queda fer algunes configuracions.

## 2. Assignar IP estàtica

Una part important i que dóna sentit a una Raspi és el fet de poder accedir a ella des de qualsevol lloc a través d'un terminal. Això ho podem fer sempre i quan sapiguem la seva adreça IP però, com sabem, aquesta pot canviar si el router es reinicia o si reiniciem la Raspi. Per evitar-ho li assignarem una IP estàtica i així ens assegurarem que sempre té la mateixa.

Per fer això tenim varies opcions:

**2.1 dhcpd.conf (l'opció més senzilla):** Abans de res ens connectem a la nostra WiFi, d'aquesta manera ja tindrem registrat el SSID i Password, i a continuació obrim un terminal amb la combinació de tecles **Ctrl+Alt+t**. Podem fer-ho també des del menú d'aplicacions.

A continuació hem de modificar l'arxiu *dhcpd.conf*

```
sudo nano /etc/dhcpd.conf
```

Afegint al final el següent codi si volem fer la connexió per cable:

```
interface eth0

static ip_address=192.168.1.XX/24
static routers=192.168.1.1
static domain_name_servers=192.168.1.1
```

O aquest si la volem fer per WiFi:

```
interface wlan0

static ip_address=192.168.1.XX/24
static routers=192.168.1.1
static domain_name_servers=192.168.1.1
```

Sortim amb *Ctrl+x*, acceptem els canvis amb *y*, i premem *enter*.

**NOTA:** Substituïm el valor XX per l'adreça que vulguem, tenint en compte de posar un valor que estigui fora del rang DHCP. Normalment es comencen a donar adreces a partir del número 33 (tot i que pot variar), però difícilment comença per adreces baixes. Això ens permet assignar sense cap problema adreces a partir de la 2 o la 3. Igualment, hem de posar l'adreça del router i DNS que correspongui amb el nostre router, i normalment sol ser 192.168.1.1 en els routers domèstics, però assegureu-vos abans per si de cas.

**2.2 interfaces:** Una altra opció és modificar directament l'arxiu `interfaces`

```
sudo nano /etc/network/interfaces
```

I modifiquem les dades de `wlan0` per les següents si farem servir el WiFi:

```
auto wlan0
allow-hotplug wlan0
iface wlan0 inet static
address 192.168.1.XX
netmask 255.255.255.0
gateway 192.168.1.1
wpa-passphrase wifi-password
wpa-ssid my-ssid
```

Canviant *wifi-password* pel password de la nostra xarxa, *my-ssid* pel nom de la nostra xarxa, i modificant el valor `XX` per l'adreça que volem.

O les dades de `eth0` si ens connectarem per cable:

```
auto eth0
iface eth0 inet static
address 192.168.1.XX
netmask 255.255.255.0
gateway 192.168.1.1
```

Sortim amb *Ctrl+x*, acceptem els canvis amb *y*, i premem *enter*.

**2.3 interfaces + wpa\_supplicant.conf:** Ens queda encara una alternativa, que és fer servir l'arxiu *wpa\_supplicant.conf*

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Aquest arxiu és l'encarregat de guardar els noms i contrassenyes de les xarxes WiFi, i per tant haurem d'afegir al final la nostra xarxa si no ens hem connectat amb anterioritat a la xarxa WiFi. Si ja tenim una xarxa amb el nom i password correctes, no cal fer res:

```
network={
ssid="WiFi_name"
psk="Wifi_password"
}
```

Sortim amb *Ctrl+x*, acceptem els canvis amb *y*, i premem *enter*.

Ara anem a modificar l'arxiu `interfaces`

```
sudo nano /etc/network/interfaces
```

I canviem la part de `wlan0` pel següent si farem servir la connexió WiFi:

```
allow-hotplug wlan0
iface wlan0 inet static
address 192.168.1.XX
netmask 255.255.255.0
gateway 192.168.1.1
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

O la part de `eth0` si ens connectarem per cable:

```
auto eth0
iface eth0 inet static
address 192.168.1.XX
netmask 255.255.255.0
gateway 192.168.1.1
```

Com veiem, els canvis d'aquest arxiu son molt semblants a l'opció 2, però aquí no posem explícitament el nom i password de la nostra xarxa, sino que ens servim de l'arxiu `wpa_supplicant.conf`. Pensar a canviar el valor `XX` de l'adreça IP.

---

Una vegada modificada la configuració de la IP reiniciarem la Raspi:

```
sudo reboot now
```

I quan hagi tornat a arrencar, comprovarem que ens ha assignat l'adreça que li hem dit amb `ifconfig`:

```
ifconfig
```

En cas que no ho hagi fet, revisem els arxius que hem modificat per si ens hem errat en alguna cosa.

En aquest moment ja ens podem desfer de la pantalla, teclat i ratolí, ja que totes les comunicacions les farem per `ssh`. La manera d'accedir a la Raspi és, des d'un terminal de Linux/Mac teclejar el següent:

```
ssh pi@192.168.1.XX
```

Ens demanarà la contrassenya que hem modificat al punt 1 (*Mentre s'escriu la contrassenya no es veurà res per pantalla.*) i ja estarem connectats a la Raspi.

**Nota:** Si fem servir Windows, hi podrem accedir mitjançant l'aplicació PuTTY.

### 3. Canviar el port SSH (Opcional)

Addicionalment podríem voler emprar un port SSH diferent de l'standart (per defecte és el 22). Això ens pot ser útil si volem accedir a la Raspi des de fora de casa i el port 22 ja està utilitzat, o per intentar tenir una mica més de seguretat evitant valors per defecte.

Per fer-ho modifiquem l'arxiu *sshd\_config*

```
sudo nano /etc/ssh/sshd_config
```

Busquem la línia on indica el port i el canviem pel que ens interessi, ja sigui modificant la línia en qüestió o comentant aquesta i afegint una nova

```
#port 22  
port 2234
```

Per comoditat, lo més pràctic és afegir 2 valors més al 22 inicial, així ens serà més fàcil d'associar el port a SSH, per exemple 2234, però podem posar el que vulguem (sempre i quan no estigui ja utilitzat)

Finalment reiniciem el servidor ssh

```
sudo service ssh restart
```

Una vegada fet el canvi, la manera d'accedir a la Raspi per terminal serà

```
ssh -p 2234 pi@192.168.1.XX
```

Sent XX l'adreça que li haurem donat a la Raspi en el punt 2.



#### 4. Generar una clau SSH a un dispositiu “client” (Opcional)

Una opció interessant de connectar-nos a la nostra Raspi per SSH és mitjançant claus. Això ens permet una connexió més ràpida ja que no haurem d'introduir la nostra contrassenya cada vegada i, per extensió, més segura, i és necessària per realitzar certes accions (com còpies de seguretat en dispositius remots, com és el nostre cas, i que s'explicarà a l'apartat d'scripts).

Per aconseguir aquesta connexió, primer necessitem generar les claus tant públiques com privades des del dispositiu client, que en aquest cas serà el nostre portàtil (ja que la Raspi faria de “servidor”). Per això, des d'un terminal executem la següent comanda

```
ssh-keygen -t rsa
```

Una vegada executada ens farà algunes preguntes, però ens bastarà amb premer enter fins que ens mostri un missatge semblant al següent

```
ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
4a:dd:0a:c6:35:4e:3f:ed:27:38:8c:74:44:4d:93:67 user@a
The key's randomart image is:
+--[ RSA 2048 ]-----+
|           .oo.      |
|          . o.E      |
|         + . o       |
|        . = = .      |
|       = S = .       |
|      o + = +        |
|     . o + o .       |
|          . o        |
|                     |
+-----+
```

Això genera les claus públiques i privades del nostre dispositiu, i les guarda a `/home/user/.ssh/idrsa.pub` i `/home/user/.ssh/idrsa` respectivament.

Una vegada tenim les claus, executarem la següent comanda

```
ssh-copy-id pi@192.168.1.XX
```

Sent XX l'adreça que li haurem donat a la Raspi prèviament.

Segurament ens sortirà un missatge semblant al següent, ons ens demanarà acceptar la connexió i introduir la contrassenya

```
The authenticity of host '12.34.56.78 (12.34.56.78)' can't be established.  
RSA key fingerprint is b1:2d:33:67:ce:35:4d:5f:f3:a8:cd:c0:c4:48:86:12.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '12.34.56.78' (RSA) to the list of known hosts.  
user@12.34.56.78's password:  
Now try logging into the machine, with "ssh 'user@12.34.56.78'", and check in:
```

```
~/.ssh/authorized_keys
```

to make sure we haven't added extra keys that you weren't expecting.

A partir d'ara, les connexions que es facin des del nostre dispositiu cap a la Raspi no requeriran de contrassenya.

*NOTA:* Si el que volem és guardar la clau per fer connexions des de fora de la xarxa, farem

```
ssh-copy-id pi@elmeudomini.ddns.net
```

Però com crear dominis i accedir des de fora de la xarxa local s'explica al següent punt.

**Esborrar claus** Si el que volem fer és eliminar la clau que acabem de guardar a la nostra Raspi (per revocar l'accés automàtic, per exemple), el que farem serà obrir el següent fitxer

```
sudo nano ~/.ssh/known_hosts
```

Cercar la línia que correspon al nostre dispositiu (generalment serà la darrera línia del fitxer si no s'han afegit més claus) i l'esborrem.

A partir d'ara, ens tornarà a demanar contrassenya a cada connexió.

## 5. Instal·lar i configurar no-ip

De la mateixa manera que les direccions IP locals poden canviar, també ho fan les direccions IP públiques, pel que potser que ara en tiguem una i demà una altra. Això és un problema si intentem accedir a la Raspi des de fora de la xarxa local, ja que no sabem quina direcció tenim assignada. Per evitar aquest problema ens ajudarem del servei no-ip.

No-ip és un servidor DNS que el que fa és traduir una direcció web (<http://www.google.com>) a la seva IP pública (<http://74.125.224.72>, per exemple). Així, el que farem ara serà accedir a [noip.com](http://noip.com), crear un compte (gratuit) i registrar un domini, per exemple *elmeudomini.ddns.net*.

Quan ja tenim el nostre domini registrat, toca instal·lar el client a la Raspi.

Hi accedim per ssh amb:

```
ssh pi@192.168.1.XX
```

Abans ens assegurarem de tenir el sistema actualitzat, pel que farem un update i un upgrade. Això pot tardar fins a 20 o 30 minuts, depenent del que s'hagi d'actualitzar.

```
sudo apt-get update && sudo apt-get upgrade
```

És recomanable fer aquest procés cada cert temps, ja que hi pot haver actualitzacions de seguretat del sistema o d'alguna altra aplicació.

A continuació descarreguem el client de no-ip

```
wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz
```

El descomprimim

```
tar -zxvf noip-duc-linux.tar.gz
```

Accedim a la carpeta que s'ha creat

```
cd noip-2.1.9-1
```

I l'instal·lem

```
sudo make  
sudo make install
```

En aquest punt ens demanarà el nom d'usuari i la contrassenya del nostre compte de no-ip, i degut a que només tindrem un domini registrat, agafarà aquest per defecte. El temps de refresc el podem deixar per defecte, i a la següent pregunta, respondrem que NO (n).

Ara creem un nou fitxer que li direm noip2

```
sudo nano /etc/init.d/noip2
```

I afegim la següent comanda

```
sudo /usr/local/bin/noip2
```

Guardem el fitxer amb Ctrl+x -> y -> enter, i li donem permissos d'execució

```
sudo chmod +x /etc/init.d/noip2
```

Actualitzem el fitxer d'inici perquè arrenqui cada vegada que engeguem la Raspi

```
sudo update-rc.d noip2 defaults
```

I posem el servei en marxa

```
sudo /usr/local/bin/noip2
```

Ens falta un darrer punt molt important, i que justifica la importància del punt 2. Per poder accedir des de fora de la xarxa, necessitem saber l'adreça pública (problema que hem solucionat amb no-ip), però també necessitem saber a quina adreça privada volem anar. Això se soluciona fent “port-forwarding” al router i indicant que tot el que vingui des de fora que vulgui anar al port 22 (o el que haguem configurat si hem fet el punt 3), vagi a la nostra Raspi. Aquí veiem la importància de tenir una IP estàtica.

Per aconseguir això necessitem accedir al router teclejant la seva adreça a qualsevol navegador, que normalment sol ser 192.168.1.1, i posem el nom d'usuari i contrassenya. Els nous routers de fibra òptica de Movistar només demanen una contrassenya que es troba abaix del router, i els més antics normalment tenen “admin” com a usuari, i “admin” o “1234” com a contrassenya (sense cometes). En cas que no vagi bé, s'haurien de cercar les credencials d'accés a Internet.

Una vegada dins cerquem alguna opció que es digui “Ports” (depèn molt de cada fabricant), i creem una nova regla que indiqui el següent:

```
Port: 22 (o el que correspongui)
Tipus de protocol: TCP/UDP
Adreça destinació: 192.168.1.XX
```

Aquesta és la informació rellevant, sent XX l'adreça que hem assignat en el punt 2. Potser ens demana també un nom, que li podem posar SSH, per exemple.

Amb això ja tenim configurat no-ip i el port 22 del router obert, pel que hauriem de poder accedir a la nostra raspi des de qualsevol lloc fora de la nostra xarxa local. No ho podem provar sempre que estiguem connectats a la pròpia xarxa, però ho podem provar des del mòbil (amb una aplicació com JuiceSSH per Android) o demanant a algú que estigui en una altra xarxa que ens ho miri.

Des del terminal (o PuTTY si estem utilitzant Windows) teclegem

```
ssh pi@elmeudomini.ddns.net
```

O si hem canviat el port ssh

```
ssh -p 2234 pi@elmeudomini.ddns.net
```

Si ens demana contrassenya, vol dir que tot ha sortit bé, i ja podem accedir a la nostra Raspi des de qualsevol banda (inclús des del mòbil).

## 6. Obrir els ports del router

En el punt anterior hem vist com obrir un port del nostre router, i aquí aprofitarem per obrir tots els que ens faci falta pels propers serveis. Així, una vegada dins del router, i com que ja sabem quins tipus de serveis farem servir, aprofitarem per obrir-los tots de cop i ens estalviem haver de fer-ho després un per un.

En el nostre cas, a banda del port 22, obrirem els següents apuntant també a l'adreça de la nostra Raspi:

```
Ports: 80 (servidor web per defecte), 21 (servidor FTP per defecte)
Protocols: TCP/UDP
IP destí: 192.168.1.XX
```

A més a més, ja que també farem servir una càmera IP, podem obrir el port que li correspondrà:

```
Port: 8081 (per exemple)
Protocols: TCP/UDP
IP destí: 192.168.1.XY
```

Canviarem XY per l'adreça que posteriorment li donarem a la nostra càmera, i així ja no haurem de tornar a configurar el router una vegada instal·lada.

## 7. Instal·lar i configurar un servidor FTP (Opcional)

Un servidor ftp ens pot servir per agafar dades de la Raspi d'una forma senzilla. Per això, tot i que les captures i vídeo de la càmera es faran directament sobre la seva IP, inicialment vam fer servir el servidor ftp per enviar les fotos, pel que creiem convenient explicar el seu procés de configuració.

Farem servir el servidor vsftpd, pel que procedim a la seva instal·lació

```
sudo apt-get install vsftpd
```

I un cop instal·lat, obrim l'arxiu de configuració per realitzar alguns canvis

```
sudo nano /etc/vsftpd.conf
```

Modificarem les línies que es mostren a continuació:

```
anonymous_enable=NO
....
local_enable=YES
...
write_enable=YES
...
```

Amb això permetrem poder interactuar amb els arxius, i no permetem l'accés anònim.

Tot i així, si no fem cap més modificació, tots els usuaris podrien accedir a tots els arxius. Si volem restringir només l'accés a les carpetes de cada usuari, modifiquem la següent línia:

```
chroot_local_user=YES
```

**Connexió per SSL (Opcional)** Si volem afegir un certificat SSL a la nostra connexió, realitzarem els següents passos.

Primer creem el certificat amb la següent comanda:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout \
/etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem
```

Això ens crea varis arxius, la ruta dels quals haurem d'afegir al fitxer de configuració del servidor. Per tant, el tornem a obrir

```
sudo nano /etc/vsftpd.conf
```

I al final de l'arxiu cerquem la línia que habilita l'accés per ssl i la descomentem.  
Afegim també, si cal, les rutes dels certificats

```
rsa_cert_file=/etc/ssl/private/vsftpd.pem  
rsa_private_key_file=/etc/ssl/private/vsftpd.pem  
ssl_enable=YES
```

I afegim les següents línies al final del fitxer

```
allow_anon_ssl=NO  
force_local_data_ssl=YES  
force_local_logins_ssl=YES
```

```
ssl_tlsv1=YES  
ssl_sslv2=NO  
ssl_sslv3=NO
```

```
require_ssl_reuse=NO  
ssl_ciphers=HIGH
```

Un cop hem guardat l'arxiu (Ctrl+x -> y -> enter), ens queda reiniciar el servidor:

```
sudo service vsftpd restart
```

**Canviar el port ftp (Opcional)** Igual que hem fet amb el port ssh, podem tenir la necessitat de modificar el port ftp. Per això, obrim el fitxer de configuració

```
sudo nano /etc/vsftpd.conf
```

I realitzaem els següents canvis a les línies corresponents

```
#connect_from_port_20=YES  
ftp_data_port=2121  
listen_port=2121
```

Aquí, com amb ssh, podem posar el port que vulguem sempre i quan estigui disponible, però la millor opció és mantenir el 21 inicial (per fer referència a ftp), i afegim dos dígit més.

I per acabar, reiniciem el servidor

```
sudo service vsftpd restart
```



## 8. Instal·lar un servidor WEB

De servidors web n'hi ha varis, i potser un dels més coneguts sigui Apache. Tot i que la Raspi no té problemes per treballar amb Apache, depen del seu propòsit potser amb un de més lleuger com nginx o lighttpd ja en tenim prou.

Aquí es detalla la instal·lació d'Apache i nginx juntament amb php, i en el nostre cas degut a que utilitzem llibreries JSON per interactuar amb l'aplicació mòbil, ens hem quedat amb Apache.

**Apache** La instal·lació d'Apache és relativament senzilla, però requereix d'alguns preparatius.

Primer crearem un nou grup www-data

```
sudo addgroup www-data
```

I afegim els usuaris al grup

```
sudo usermod -a -G www-data www-data
```

Ara, per instal·lar el servidor, ho farem amb

```
sudo apt-get install apache2 apache2-utils php5 libapache2-mod-php5
```

Iniciem el servidor

```
sudo /etc/init.d/apache2 restart
```

I provem que funciona accedint a la direcció IP de la Raspi si estem a la mateixa xarxa

```
http://IP_ADDRESS
```

O al nom del domini que hem creat a no-ip si estem a una xarxa externa

```
http://elmeudomini.ddns.net
```

Per provar php, creem el següent fitxer

```
sudo nano /var/www/html/info.php
```

Amb el següent contingut

```
<?php
    phpinfo();
?>
```

I provem que funciona accedint a

`http://ID_ADDRESS/info.php`

**Nginx** Per instal·lar nginx, només hem de executar la comanda

```
sudo apt-get install nginx php5 libapache2-mod-php5
```

Una vegada instal·lat, creem la carpeta `/var/www/html` en cas que no existeixi

```
sudo mkdir /var/www/html
```

I modifiquem l'arxiu de configuració de nginx

```
sudo nano /etc/nginx/sites-available/default
```

Deixant-lo de la següent manera

```
server {
    listen 80;
    server_name $domain_name;
    root /var/www/html;
    index index.html index.htm;
    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;
}
```

Reiniciem el servidor

```
sudo service nginx restart
```

I comprovem que funciona escrivint a qualsevol navegador l'adreça IP de la Raspi si estem connectats a la mateixa xarxa

`http://IP_ADDRESS`

o el nom del nostre domini si estem a una xarxa diferent.

```
http://elmeudomini.ddns.net
```

Per instal·lar php executem

```
sudo apt-get install php5 libapache2-mod-php5
```

Obrim l'arxiu de configuració

```
sudo nano /etc/nginx/sites-available/default
```

I el deixem de la següent manera

```
server {
    listen 80;
    server_name $domain_name;
    root /var/www/html;
    index index.html index.htm;
    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;

    location ~\.php$ {
        fastcgi_pass unix:/var/run/php5-fpm.sock;
        fastcgi_split_path_info ^(.+\.php)(/.*)$;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param HTTPS off;
        try_files $uri =404;
        include fastcgi_params;
    }
}
```

Ara només ens falta reiniciar el servidor

```
sudo service nginx restart
```

Crear un fitxer php

```
sudo nano /var/www/test.php
```

Amb el següent codi de prova

```
<?php
phpinfo();
?>
```

I provar que el php funciona correctament accedint a

`http://IP_ADDRESS/test.php`

o des de fora de la xarxa

`http://elmeudomini.ddns.net/test.php`

## 9. Instal·lar MySQL y phpMyAdmin

Instal·lem els arxius de MySQL

```
sudo apt-get install mysql-server mysql-client php5-mysql
```

Durant l'instal·lació ens demanarà certa informació, que anirem emplenant.

Una vegada acabada la instal·lació, reiniciem el servei amb

```
sudo service mysql start
```

I provem si funciona

```
mysql -u root -p
```

Introduïm el password que hem escollit durant la instal·lació, i si carrega correctament la instal·lació ha sortit bé.

Premem Ctrl+c per sortir.

Per instal·lar phpMyAdmin fem

```
sudo apt-get install libapache2-mod-auth-mysql php5-mysql phpmyadmin
```

Durant aquesta instal·lació li direm quin servidor fem anar, quan ens demani si volem fer servir MySQL li direm que sí, i posarem una contrassenya per phpMyAdmin.

A continuació obrirem el següent fitxer

```
sudo nano /etc/php5/apache2/php.ini
```

I afegirem la següent línia al principi del fitxer

```
extension=mysql.so
```

Després executem la següent comanda

```
sudo ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf.d/phpmyadmin.conf
```

I reiniciem el servidor

```
sudo /etc/init.d/apache2 reload
```

Provem a veure si podem accedir a phpMyAdmin amb

```
http://IP_ADDRESS/phpmyadmin
```

**Nota:** Tots els fitxers html, php, etc. que farem servir es troben a la carpeta “web” del repositori.

## 10. Configurar notificaciones Push amb l'API de Pushover

Ja que les notificaciones push natives en la nostra pròpia aplicació requereixen de certs passos una mica farragosos (registrar l'app a Google, tenir compte de Google Developer, etc.), hem optat per servir-nos d'una aplicació que ens proporciona aquesta funcionalitat: Pushover.

El que necessitem primer de tot és crear un compte a la web de Pushover (la qual cosa ens generarà una clau única d'usuari) i posteriorment crearem una “aplicació” (que no és més que generar un Token per fer servir juntament amb l'API de Pushover). Posteriorment baixarem l'app de Pushover de Google Play o Apple Store.

Una vegada tenim la nostra clau d'usuari, el token de la nostra aplicació i l'aplicació mòbil instal·lada, anem a la Raspi i creem un nou arxiu que li direm, per exemple, notificaciones.sh.

Una recomanació és crear una carpeta al nostre home on guardar els nostres scripts, així ho tenim tot més ordenat.

```
mkdir scripts
cd scripts
sudo nano notificaciones.sh
```

I dins d'aquest fitxer copiem el següent codi:

```
#!/bin/bash

MSG="message=Escriure aquí el missatge desitjat"

curl -s \
    --form-string "token=APPTOKEN" \
    --form-string "user=USERTOKEN" \
    --form-string "$MSG" \
    https://api.pushover.net/1/messages.json
exit 0
```

Aquí haurem de substituir APPTOKEN i USERTOKEN pel token de la nostra app i la clau d'usuari, respectivament.

I finalment li hem de donar permisos d'execució

```
sudo chmod +x notificaciones.sh
```

Aquest seria l'script base per enviar notificaciones des de la Raspi al nostre mòbil, i ho podem provar simplement executant l'script

```
./notificacions.sh
```

A partir d'aquí, ens podem crear tants scripts com vulguem per enviar notificacions depenent del que ens interessi. Inclús podem cridar aquest script des d'un altre script, si volem.

## 11. Configurar notificaciones per Telegram (Opcional)

Telegram és una aplicació de missatgeria molt versàtil i que gràcies a que és de programari lliure ens permet treballar amb ella més enllà d'enviar missatges als nostres amics.

Per instal·lar Telegram, primer necessitem les següents llibreries

```
sudo apt-get install libreadline-dev libconfig-dev libssl-dev lua5.2 \
```

```
liblua5.2-dev libevent-dev libjansson-dev libpython-dev make
```

I després baixar el fork de Telegram per Linux

```
git clone --recursive https://github.com/vysheng/tg.git && cd tg
```

I fem

```
./configure
```

```
make
```

Una vegada instal·lat l'executem amb

```
bin/telegram-cli -k tg-server.pub -W
```

*NOTA:* Potser en aquest punt ens dona un error com el següent

```
pi@raspberrypi:~/tg $ bin/telegram-cli -k tg-server.pub -W
Telegram-cli version 1.4.1, Copyright (C) 2013-2015 Vitaly Valtman
Telegram-cli comes with ABSOLUTELY NO WARRANTY; for details type 'show_license'.
This is free software, and you are welcome to redistribute it
under certain conditions; type 'show_license' for details.
Telegram-cli uses libtcl version 2.1.0
Telegram-cli includes software developed by the OpenSSL Project
for use in the OpenSSL Toolkit. (http://www.openssl.org/)
I: config dir=[/home/pi/.telegram-cli]
> telegram-cli: tgl/mtproto-utils.c:101: BN2ull: Assertion '0' failed.
SIGNAL received
```

Si és així, haurem de modificar el fitxer *tgl/mtproto-utils.c* i comentar les línies 101, 105 i tornar a compilar.

Quan el tenim correctament instal·lat, i la primera vegada que l'executem, ens demanarà el nostre número de telèfon, al qual haurem d'incloure el codi de país



(+34 en el cas d'Espanya), i introduir el codi que rebrem via Telegram al nostre mòbil.

I ja el tenim instal·lat. Ara, per enviar missatges ho farem amb la següent comanda

```
msg Name_Lastname message
```

*NOTA:* Name és el nom del nostre contacte, i Lastname el cognom, i s'han de separar amb la barra baixa

**Scripts per enviar missatges** Podem crear varis scripts per enviar missatges més còmodament sense haver de tenir el client en marxa continuament. Per això, crearem un script amb el nom que vulguem

```
sudo nano /home/pi/scripts/tg.sh
```

i hi afegirem el següent codi

```
#!/bin/bash
to=$1
msg=$2
tgpath=/path/to/telegram/tg
cd ${tgpath}
cmd="bin/telegram-cli -W -k server.pub -e \"msg $to $msg\""
eval $cmd
```

*NOTA:* Tenir en compte que la variable *tgpath* ha de contenir la ruta on hem ubicat la carpeta tg que ens hem baixat el repositori.

I li donem els permisos necessaris

```
sudo chmod 0655 /home/pi/scripts/tg.sh
```

Per executar-lo, fem

```
tg.sh Name_Lastname Message
```

Amb això, podríem utilitzar Telegram per enviar missatges al nostre mòbil en comptes de fer servir Pushover.

## 12. Enviar notificacions quan es fa login

Una funcionalitat interessant és la d'enviar una notificació quan algun usuari accedeix a la Raspi. Per això, hem aprofitat l'script de Pushover del punt 10

```
sudo nano loginNotification.sh
```

I hem afegit el següent codi

```
#!/bin/bash

#Using PAM
MSG="message=Login from user $PAM_USER"

if [ "$PAM_TYPE" != "close_session" ]; then
curl -s \
    --form-string "token=APPTOKEN" \
    --form-string "user=USERTOKEN" \
    --form-string "$MSG" \
    https://api.pushover.net/1/messages.json
fi
exit 0
```

El que fa aquest codi és executar el curl només quan es fa login gràcies a la condició, ja que en principi no ens interessa saber quan es desconnecta. En cas que també ho vulguem saber, n'hi ha prou amb treure el condicional.

I li hem de donar també els permisos d'execució

```
sudo chmod +x loginNotification.sh
```

Per fer que aquest script funcioni, ens queda modificar un arxiu, i és **important fer-ho amb cura ja que un error ens pot bloquejar l'accés a la Raspi**.

Obrim el següent fitxer

```
sudo nano /etc/pam.d/sshd
```

Busquem la part de codi següent, i afegim les línies corresponents al nostre script, tenint cura de no modificar la resta de codi

```
.....
# SELinux needs to be the first session rule.  This ensures that any
# lingering context has been cleared.  Without this it is possible that a
```

```
# module could execute code in the wrong domain.
session [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so close

# El meu script
session required pam_exec.so seteuidd /home/pi/scripts/loginNotification.sh
# Fi del meu script

# Set the loginuid process attribute.
session      required      pam_loginuid.so

# Create a new session keyring.
session      optional      pam_keyinit.so force revoke
.....
```

Guardem el fitxer amb Ctrl+x -> y - enter.

Ara, per provar que tot ha sortit bé, obrirem un nou terminal amb Ctrl+alt+t, i provarem a fer login amb el nostre usuari

```
ssh pi@192.168.1.XX
```

Si després d'introduir la contrassenya aconseguim entrar, és que tot ha sortit com esperàvem. En cas que una vegada introduïda la contrassenya ens tregui fora de la sessió, hem de revisar el que hem fet i buscar algun error. Per això és important fer aquesta prova des d'un altre terminal, ja que si hem fet alguna cosa malament, i sortim de la sessió actual no podríem tornar a accedir a la Raspi.

## Càmera IP

En quant a la càmera IP, normalment tenen de per sí una interfície web amb la que podem configurar ja algunes opcions, però en el nostre cas ens interessa dependre lo mínim possible d'aquest sistema i interactuar directament amb ella a través de la seva IP.

Això ens dóna la llibertat de poder crear la nostra pròpia web o aplicació mòbil i accedir a la imatge de la càmera o fer captures gràcies als sensors que muntarem a l'Arduino. Tot i així, per configurar la càmera farem servir la pròpia interfície web, ja que al ser un entorn gràfic és més còmode.

Segons les instruccions del fabricant, les càmeres IP tenen una adreça establerta a la qual s'hi pot accedir des d'un navegador si la connectem a un ordinador per cable ethernet, pel que la configuració inicial la farem des d'un ordinador. Una vegada sabem quina adreça té, i després d'haver introduït el nom d'usuari i contrassenya per defecte, farem les següents configuracions:

1. Assignar un port diferent del port 80 que ve per defecte, que serà el que hem obert al router abans, destinat a la càmera.
2. Canviar la contrassenya d'accés
3. Configurar la connexió FTP per comunicar-se amb la Raspi (Opcional)
4. Anotar l'adreça MAC
5. Configurar la connexió WiFi afegint el SSID i contrassenya de la nostra xarxa

**Nota:** Cada càmera té la seva interfície, pel que on està cada opció pot variar.

Pel que fa referència a la IP, com podem imaginar també li hem d'assignar una IP estàtica, però en aquest cas hem optat per fer-ho des del router utilitzant l'adreça mac de la càmera, ja que no podem editar cap fitxer (o no tan fàcilment) com sí podem fer amb la Raspi.

Accedim al router i anem a l'opció de control MAC (novament, cada router és diferent). Afegim la MAC de la càmera i li assignem la mateixa IP que hem posat al obrir el port corresponent en el punt 5.

Després d'això, ja podem desconnectar la càmera de l'ordinador, apagar-la i tornar-la a encendre. Si tot ha sortit com esperavem, no hi hauria d'haver problema en accedir a la càmera des de qualsevol navegador web, simplement posant l'adreça IP seguida de : i el port, si estem dins de la xarxa local

`http://192.168.1.XY:8081`

O fent servir el nostre domini de no-ip, i que gràcies a la Raspi sempre estarà actualitzat

`http://elmeudomini.ddns.net:8081`

Amb això accedim al menú de la pròpia càmera i podrem realitzar les configuracions pertinents en cas de ser necessari, però com hem dit abans, no serà aquesta la forma en que agafarem les fotos, sino que ens servirem de les peticions per http que ens ofereix (i que podem consultar als [foros de Dlikn](#)). Si es fa servir una càmera diferent, s'haurà de consultar el manual d'usuari i les comandes necessaries per configurar la càmera i addecir a ella.

Ja podem col·locar la càmera on vulguem, sempre que estigui dins de la cobertura WiFi.

## **Arduino**

### **Aplicació mòbil**

De l'aplicació mòbil no en parlarem, ja que podem trobar el codi a la carpeta “App” del repositori, que es podrà baixar, consultar i provar.