Vacas Martínez, José Antonio

# Crea proyectos con Raspberry Pi



Licencia CC by SA

## by @javacasm

### José Antonio Vacas Martínez

### http://elCacharreo.com

## Julio 2020

# Tema 4 - Uso de Raspberry Pi

En este tema vamos a ver los usos normales de Raspberry Pi.

Dado que el uso de los típicos programas de ofimática o similares es idéntico al que se hace en otros ordenadores nos vamos a hablar de ellos.

Nos centraremos en los usos más típicos donde se trabaja con la consola/terminal. Es el típido uso que se hace en los sistemas Linux. La mayoría de los veremos se puede hacer en los sistemas con Linux de las distribuciones [Debian](https://www.debian.org/index.es.html) y [Ubuntu](https://www.ubuntu.com/) en los que está basado Raspbian.

## Problemas habituales

Siempre podemos encontrarnos con problemas. Veamos los más frecuentes

### Alimentación

Necesitamos un mínimo de 2A (3A para V4), si la alimentación está por debajo se pueden producir cuelgues inesperados e incluso que no arranque.

Cuando la Raspberry detecta que no tiene suficiente alimentación visualiza un rayo amarillo en la pantalla



Problemas de alimentación. Fuente RasberryParaTorpes

### Velocidad de la tarjeta

Se recomienda velocidad 10, una velocidad menor da problemas como bloqueos

### Espacio en disco

Al menos 8Gb por sistema operativo, mejor 16Gb o más

### No se ve nada en el monitor

¿Lo arrancaste con el monitor conectado? Es necesario arrancar con el monitor conectado, si Raspberry no detecta ningún monitor en arranque desactiva la salida de vídeo.

## Mantenimiento

Una vez instalado el sistema, necesitamos de vez en cuando actualizarlo. Veamos como hacerlo.

### Actualización (update)

Desde un terminal/consola tecleamos lo siguiente

Para buscar cambios

sudo apt update

Para instalar estos cambios

sudo apt upgrade

Para actualizar el sistema

sudo apt dist-upgrade

Para actualizar a una nueva distribución

sudo apt full-upgrade

Para instalar un paquete determinado

sudo apt install paquete

Vemos como en todos los comandos utilizamos la palabra "sudo" esto es debido a que se necesitan privilegios de administrador para todo lo relacionado con la actualización del sistema.

[](https://youtu.be/BaVfTWFUHtU)

[Vídeo: Actualizar e instalar software desde terminal en Raspberry Pi](https://youtu.be/BaVfTWFUHtU)

#### Actualización de los distintos firmwares

Los diferentes componentes de la Raspberry necesitan de varios firmwares para funcionar, que también conviene tener actualizados. Podemos actualizarlos con:

sudo rpi-update

Hay que tener **cuidado con esta comando**: si actualizamos un firmware pero el sistema operativo no lo está y por tanto espera encontrar una versión anterior del firmware, podemos encontrarnos con que nuestra Raspberry se queda completamente bloqueada.

#### Actualización de una versión a la siguiente

Aunque es conveniente que cuando salga una nueva versión hagamos una instalación desde cero, a veces tenemos muchas aplicaciones instaladas o mucha información del usuario que nos llevaría mucho tiempo extraer y luego volver a copiar/installar. En ese caso nos podemos plantear hacer una actualización de una versión a otra.

Vamos a ver un ejemplo de cómo actualizar de una versión del SO a la siguiente. En este caso particular cómo pasar de "Stretch" a "Buster".

1. Modificamos la versión a la apuntan los repositorios de 'stretch' a 'buster'. Podemos hacerlo cambiando a mano el ficheros **/etc/apt/sources.list** y **/etc/apt/sources.list.d/raspi.list** o con un comando como el siguiente:

grep -rl stretch /etc/apt/ | sudo xargs sed -i 's/stretch/buster/g'

1. Actualizamos el sistema completamente:

sudo apt update  
sudo apt dist-upgrade  
sudo apt full-upgrade

1. Actualizamos los firmwares:

sudo rpi-update

1. Quitamos aplicaciones que ya no están en buster

* sudo apt purge timidity lxmusic gnome-disk-utility deluge-gtk evince wicd wicd-gtk clipit usermode gucharmap gnome-system-tools pavucontrol

1. Ahora actualizamos el aspecto con la nueva configuración visual abriendo ‘Appearance Settings’ y en la pestaña ‘Defaults’ pulsamos ‘Set Defaults’ lo que debería ponernos un tamaño de fuente y de iconos acorde a la resolución usada.

##### Recursos

[Fuente](https://www.raspberrypi.org/blog/buster-the-new-version-of-raspbian/)

#### Instalación de programas

Además de la línea de comandos, siempre podemos instalar desde la herramienta visual "Añadir programas" en el menú Preferencias.

[](https://youtu.be/3eeIHe-NCZs)

[Vídeo: Cómo actualizar e instalar software Raspberry Pi](https://youtu.be/3eeIHe-NCZs)

#### Instalación de paquetes a partir del código fuente

* Descargamos el código fuente (normalmente comprimido)
* Lo descomprimimos con
* unzip codigo\_fuente.zip
* ó
* tar xvf cofigo\_fuente.tgz
* (según el formato en el que esté comprimido)

Dentro del directorio del código ya descomprimido normalmente encontramos un fichero README o INSTALL que nos dará las instrucciónes, pero suelen ser muy parecidas a estas:

Preparan el código para que compile en nuestro sistema y además comprueban que tengamos las herramientas y librerías necesarias con:

cmake .

ó

configure

Compila el código y generamos un ejecutable

make

Lo instalamos en el sistema (por eso necesitamos usar sudo)

sudo make install

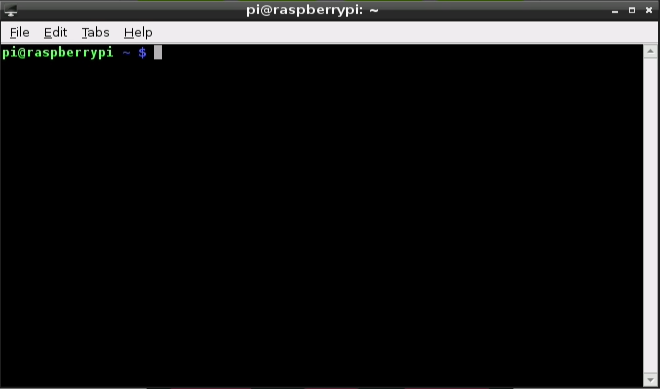
## Manejando tu Raspberry Pi

Como sabes Raspberry es una máquina Linux, con lo que podrás manejarla igual que se maneja cualquier otra equipo Linux.

### Consola (línea de comandos)

Podemos hacer casi todo desde el habitual entorno gráfico, pero también desde el terminal o la consola, también llamada líneas de comandos.

Si te acostumbras a usarla verás que ganas en productividad y además verás que puedes automatizar muchas tareas.



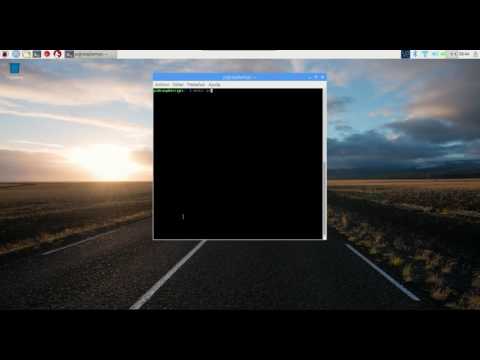
Consola

#### Comandos básicos:

Como ya hemos dicho, Raspbian es una versión de [Linux](https://es.wikipedia.org/wiki/GNU/Linux), que nos más que una versión moderna del sistema operativo [Unix](https://es.wikipedia.org/wiki/Unix). Por esto tenemos acceso a los comandos de esos sistemas operativos.

Veamos algunos de los comandos más utilizados:

* La tecla Tabulador nos permite completar el nombre del fichero/directorio
* **ls** : muestra los archivos y directorios ( **ls -l** para más detalles y **ls -a** para mostrar todos)
* **cd** : cambia de directorio (**cd ~** nos lleva a nuestro directorio home y **cd ..** sale del directorio actual)
* **chmod** : cambia los permisos de un fichero/directorio (**chmod ugo-w fichero** quita todos los permisos de escritura)
* **pwd** : nos dice el directorio actual
* **mv** : mueve directorios/ficheros a un nuevo destino
* **rm** : borra directorios/ficheros
* **mkdir** : crea un directorio
* **passwd** : cambia la contraseña del usuario actual
* **ps -ef** : muestra los procesos en ejecución
* **top** : administrador de tareas
* **clear** : borra todo el contenido del terminal
* **df** : muestra el % de disco ocupado
* **nano** : editor de texto básico
* **vi** : editor de texto avanzado pero complejo
* **du** : muestra lo que ocupa un directorio (**du -s** \* muestra lo que ocupa un directorio y todo lo que contiene)
* **sudo halt** apaga la raspberry
* **sudo shutdown -h now** apaga la raspberry
* **history** : muestra todos los comandos que se han ejecutado antes. Podemos ejecutar el comando de la posición n, con !n . Las teclas abajo/arriba del cursor nos permiten iterar por los comandos usados.
* **man comando**: Para obtener ayuda sobre comando
* Para hacer fichero script: añadimos los comandos, chmod u+x fichero y para ejecutarlo ./fichero

[](https://youtu.be/BF0Kjb4g454)

[Vídeo: Uso del terminal y comandos Linux en Raspberry Pi](https://youtu.be/BF0Kjb4g454)

#### Estructura de ficheros

Algunas características de sistema de fichero de linux

* Usa un formato de partición ext4 (también existen aunque en desuso el ext3 y el ext2), aunque permite usar FAT
* El árbol de directorios tiene un único directorio raíz del que cuelga todo. Todos los dispositivos (pen drives, discos externos, discos de red) se integran dentro de este árbol, **montando** su raiz en un directorio determinado (montamos y desmontamos con **mount** y **unmonut** )

El usuario sólo acceso a su directorio y el solo el administrador (**root**) puede acceder al resto de directorios

##### Algunos directorios

* / directorio raiz
* /etc configuración
* /home usuario
* /usr programas para usuarios
* /usr/share recursos de programas (imágenes, traducciones)
* /usr/share/doc documentación
* /bin ejecutables del sistema
* /lib librerías
* /boot Arranque del sistema
* /usr/bin ejecutables para usuarios
* /media o /mnt punto de montaje de dispositivos de almacenamiento externo

#### Usuarios

El usuario por defecto es "**pi**" con contraseña "**raspberry**" por defecto

#### Cuidado con sudo

Los usuarios normales pueden hacer muchas cosas, pero las tareas más importantes (y por tanto peligrosas si se hacen mal), como pueden ser la configuración o borrado de ficheros críticos no están permitidas.

Esas tareas sólo las puede hacer el usuario administrador, llamado **root**, que puede hacer cualquier cosa.

Como hay veces que un usuario necesita hacer alguna de estas tareas, por ejemplo editar un fichero de configuración, podemos solicitar permisos para hacer esa tarea anteponiento al comando la palabra "sudo". Algunas acciones nos pedirán que introduzcamos el password de nuestro usuario como medida de seguridad.

Por ejemplo si queremos editar un fichero de la carpeta de configuración etc llamado ftab, haremos

sudo geany /etc/fstab

geany es el editor de ficheros de texto, y al usar "sudo" estamos pidiendo permiso para hacer algo como root

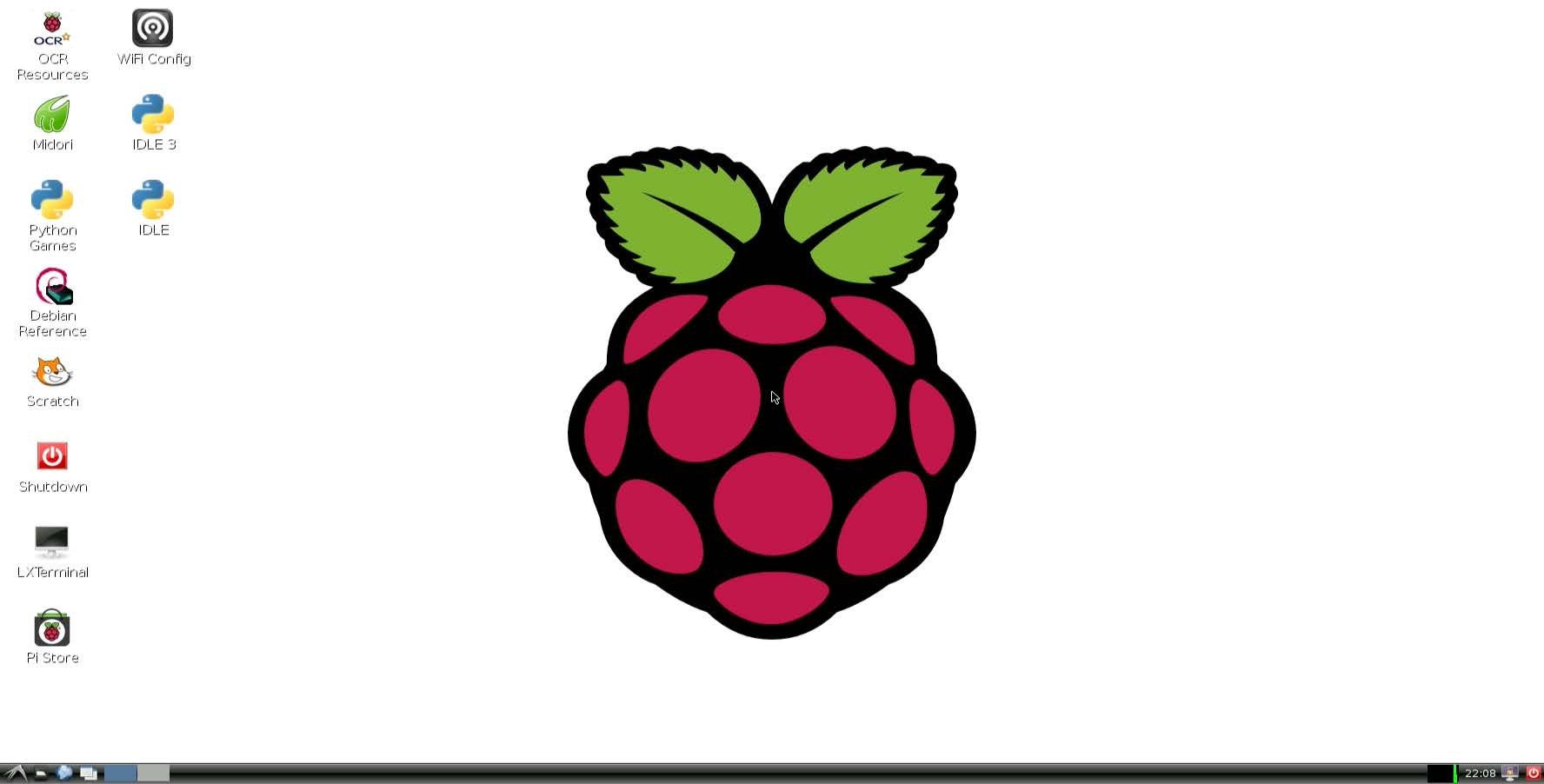
Esto nos sirve para ver que podemos abrir aplicaciones de escritorio desde la consola.

Nos da todo el poder del usuario administrador (**root**)

### Interface gráfico

Para arrancar el interface gráfico desde consola si no está arrancado usando

startx



startx

### Acceso remoto

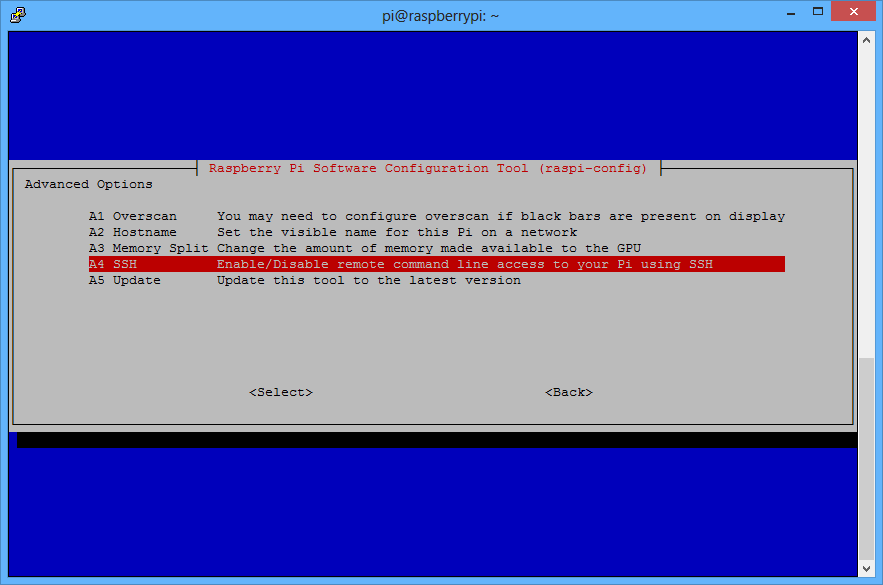
Algo muy frecuente es que queramos acceder a nuestra Raspberry Pi remotamente, es decir sin un teclado ni monitor conectado directamente. Evidentemente necesitamos tener un SO instalado y habilitar el acceso remoto. Veamos algunas de las formas de hacerlo.

### SSH (vía consola)

SSH es el protocolo de acceso remoto por consola (o terminal). Se suele utilizar cuando no necesitamos (o no tenemos) entorno gráfico. Es una forma de conexión segura (usa encriptación y verificación a ambos lados) utilizada para acceder a router y otros dispositivos remotos

Tenemos que activarlo en la configuración para poder acceder desde fuera.Entramos en la configuración avanzada

sudo raspi-config



ssh

Podemos hacerlo también por comandos con

sudo service ssh start  
sudo insserv ssh

Ahora podremos conectarnos remotamente con ssh, desde Linux con el comando **ssh** o con otras herramientas que veremos a continuación

ssh pi@192.189.0.123

Si lo vamos a activar se recomienda cambiar la contraseña para evitar que cualquiera pueda acceder a nuestro equipo.

[](https://youtu.be/-BH3spberkc)

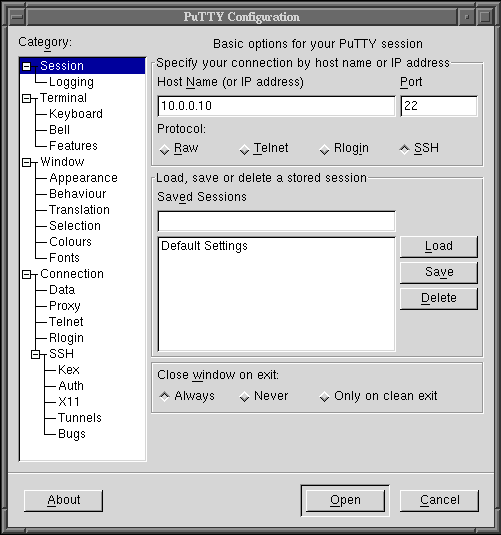
[Vídeo: Conexión vía SSH a Raspberry Pi](https://youtu.be/-BH3spberkc)

### Herramientas de conexión ssh

Son muchas la herramientas que exiten para poder conectarnos entre equipos usando SSH, vamos a ver un par de ellas.

#### Putty

[Putty](http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html) es una herramienta opensource disponible para muchos sistemas operativos pensada para conectar remotamente con equipos usando distintos protocolos como ssh, telnet, serie, etc



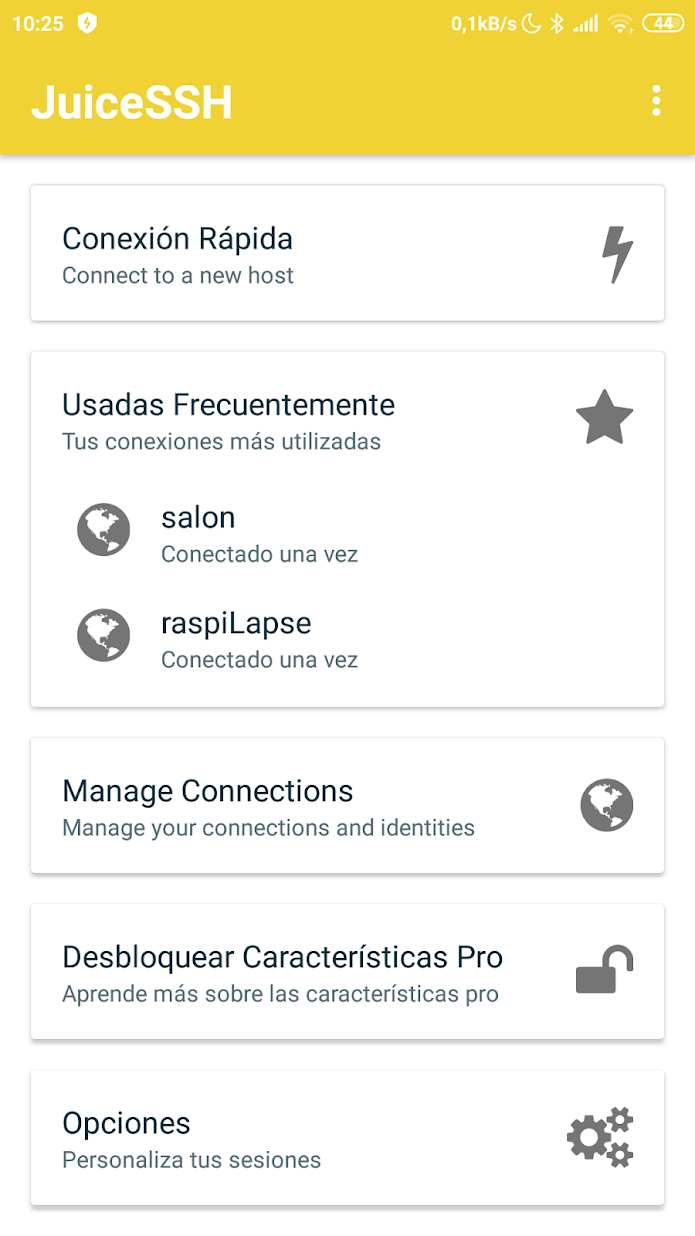
Putty

Permite guardar las credenciales de cada sistema, facilitándo la conexión.

#### JuiceSSH

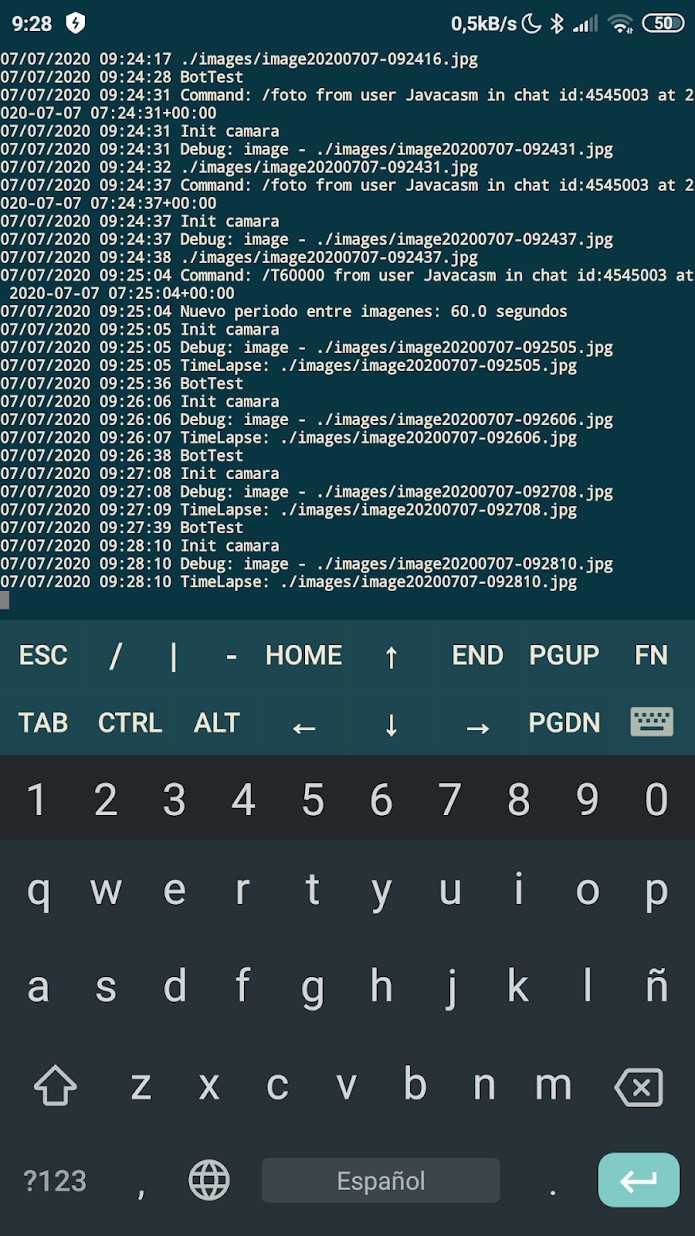
[JuiceSSH](https://play.google.com/store/apps/details?id=com.sonelli.juicessh) es una herramienta de conexión ssh para móviles y tabletas Android.

También nos permite gestionar las credenciales de acceso y los diferentes equipos a los que conectarnos:



juicessh1.png

El uso incluso en un móvil es más cómodo de lo que parece, puesto que permite el uso de teclas como TAB, ctrl, cursor, etc.



juicessh2.png

[](https://youtu.be/LLiZ52ss2DQ)

[Cliente SSH para móviles y tabletas Android: juicceSSH](https://youtu.be/LLiZ52ss2DQ)

### Automatizar conexiones

Si vamos a conectarnos frecuentemente a un mismo equipo puede resultar pesado tener que poner siempre usuario y contraseña.

El protocolo ssh permite automatizarlo sin más que añadir nuestra **clave pública ssh** a la lista de host autorizados del servidor al que nos vamos a conectar, haciendo lo siguiente:

1. Generar nuestra **clave pública ssh**, asociada al usuario y la máquina

* ssh-keygen -t rsa -b 4096 -C user@email.com

1. Ahora copiamos nuestra clave pública ssh a

* cat ~/.ssh/id\_rsa.pub | ssh username@server.address.com 'cat >> ~/.ssh/authorized\_keys'
* La siguiente vez que nos conectemos lo haremos sin necesidad de instroducir la clave.

Como podemos ver en el último comando que hemos usado, podemos usar ssh para enviar ficheros y su contenido entre equipos. En proyecto más adelante lo usaremos bastante.

### VNC

VNC es un protocolo que nos permite acceder remotamente al escritorio de otra máquina.

En las nuevas versiones de Raspbian podemos activar VNC desde la configuración (o desde raspi-config).

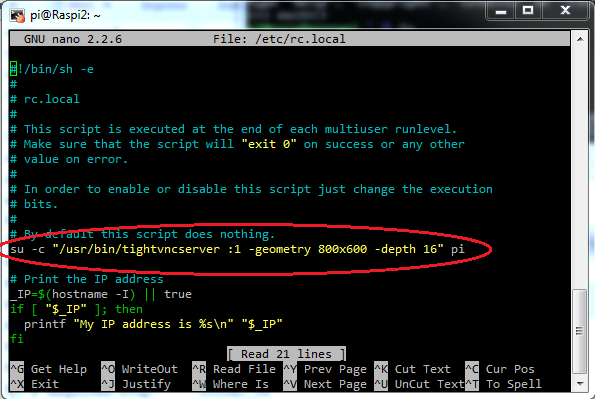
#### VNC en equipos antiguos

Si no está disponible en nuestra instalación podemos instalarlo de manera sencilla con:

sudo apt-get install tightvncserver

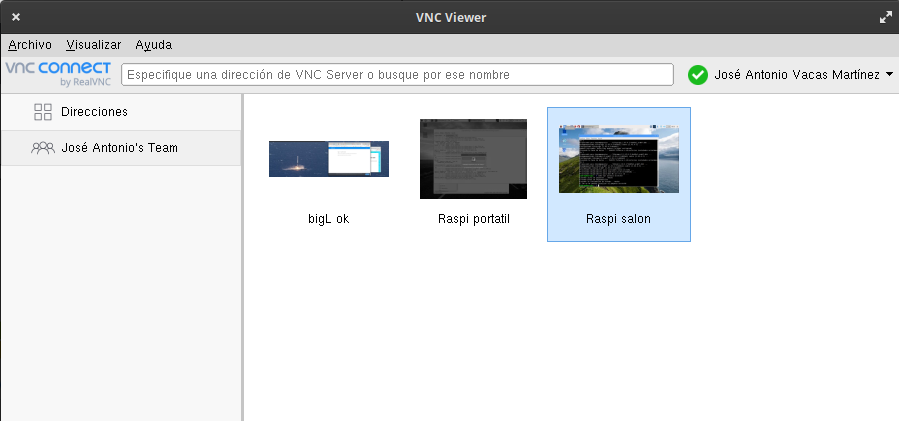
Este software requiere que un servicio se ejecute al arrancar si queremos acceder en cualquier momento. Podemos instalarlo añadiendo la siguiente línea al archivo **/etc/rc.local**

su -c "/usr/bin/tightvncserver :1 -geometry 800x600 -depth 16" pi



vnc

Ahora accederemos usando un cliente vnc, como por ejemplo [VNC Viewer](https://www.realvnc.com/es/connect/download/viewer/) de RealVNC. Si nos creamos una cuenta en RealVNC podremos acceder desde cualquier parte, incluso desde internet, sorteando nuestro router.



Acceso con cuenta de usuario VNC

Existen clientes de VNC para teléfonos móviles y tabletas, lo que nos da muchas posibilidades

La conexión por VNC nos permite acceder como si estuvieramos conectados directamente

### Conexión y acceso directo

Vamos a configurar nuestra raspberry y un portátil con Ubuntu para facilitar al máximo la conexión y así no tener que utilizar muchos componentes. De esta manera podremos trastear con un kit mínimo, evitando tener que usar un teclado, ratón y sobre todo un monitor.



Conexión directa entre Raspberry y Portatil

En concreto usaremos símplemente un cable de red (ethernet) y un cable micro-usb para alimentar la raspberry.

Con esta configuración no podemos consumir en total más de los 500mA que proporciona el USB.

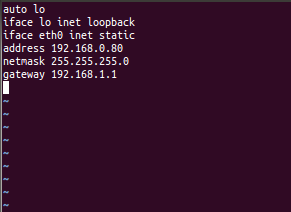
Tendremos que modificar ficheros de configuración en el PC y en la raspberry.

Asumiremos que tenemos conexión a internet via Wifi y utilizaremos el cable ethernet para dar conectividad a la raspberry. Crearemos una red entre el portátil y la raspberry creando una subred distinta y haremos que el portátil actúe como gateway de esa red enrutando los paquetes hacia la raspberry y dándole acceso a internet.

Comencemos editando la configuración del pc, para lo que ejecutaremos en el pc:

sudo vi /etc/network/interfaces

y dejamos el contenido del fichero (la red que se usa normalmente es las 192.168.1.x de ahí que el gateway sea 192.168.1.1 que es el real)

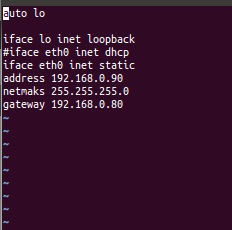


Configuración inicial de la red local

Ahora vamos a editar la configuración de la raspberry. La forma más sencilla es editando los ficheros de configuración desde el pc, para lo que insertamos la tarjeta sd de la raspberry (obviamente con esta apagada) en el pc y ejecutamos en este:

sudo vi /media/10b4c001-2137-4418-b29e-57b7d15a6cbc/etc/network/interfaces

Quedando el mismo:



Configuración final de la red local

Ahora, colocamos la tarjeta sd en la raspberry y volvemos a encenderla

Conectamos el cable ethernet entre los dos

En el PC hacemos comprobamos que la tarjeta eth0 está ok y con la ip correspondiente, haciendo

ifconfig /all

Veremos que aparece el interface eth0 con ip 192.168.0.80

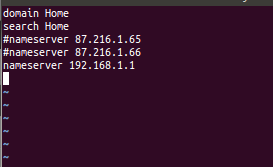
Ahora vamos a hacer que el portátil actúe como router. Para ello ejecutamos los siguientes comandos

sudo su -  
root@ubuntu-asus:~# echo 1 > /proc/sys/net/ipv4/ip\_forward  
root@ubuntu-asus:~# /sbin/iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE

Por último editamos el fichero de configuración de DNS con

sudo vi /etc/resolv.conf

y lo dejamos así



Configuración de servidor de nombres

Ahora solo falta probar que tenemos conectividad, haciendo un ping

ping 192.168.0.90

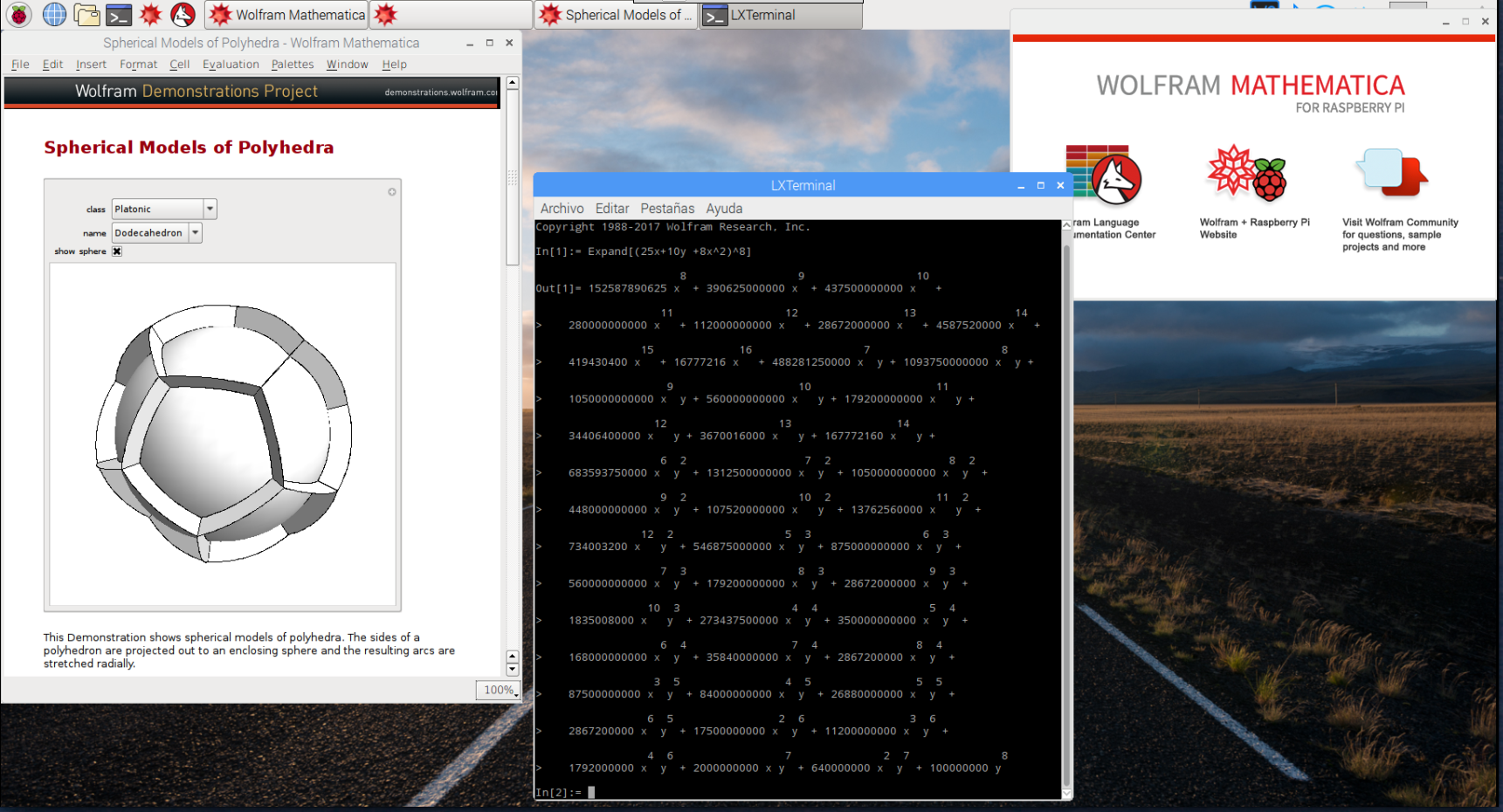
Si todo es correcto ya podremos acceder via ssh o VNC

## Usos

Veamos cómo podemos utilizar lo aprendido...

### Para hacer cálculos con Mathematica

Hay una versión gratuita (para uso no comercial) de Worlfram Mathematica instalada por defecto en Raspbian



Mathematica en Raspberry Pi

[](https://youtu.be/VVHoREZ8Rc4)

[Vídeo: Trabajando con Mathematica en Raspberry](https://youtu.be/VVHoREZ8Rc4)

### Vigilancia

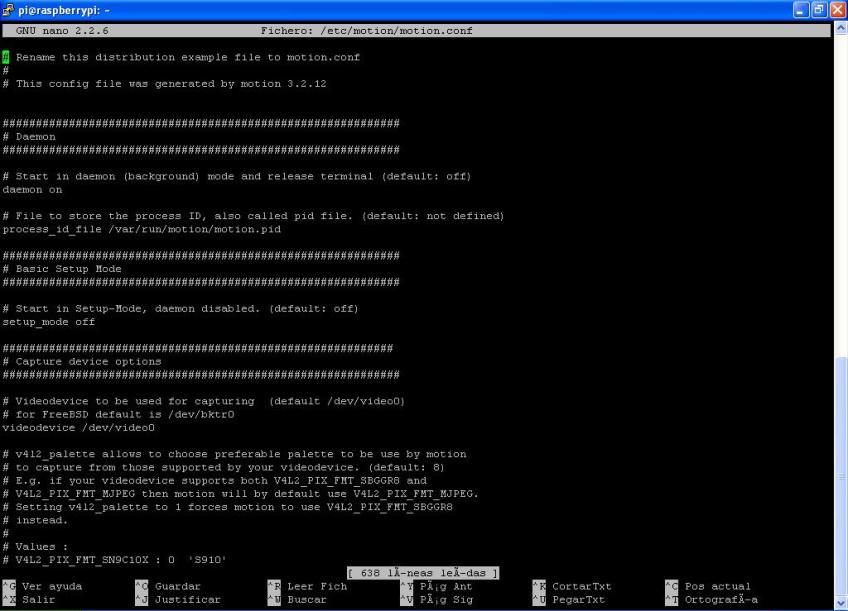
Podemos usar su cámara (la original o una USB)

Usaremos un software standard de Linux: **motion**

sudo apt-get install motion

Editamos la configuracion

sudo nano /etc/motion/motion.conf



usando motion

Lo arrancamos

montion -n

Podremos acceder a la imagen en vivo de la cámara con

http://rasperry\_ip:8081

# Instalación Arduino

Un uso muy frecuente de Raspberry es para trabajar en proyectos de electrónica o robótica donde también contamos con Arduino. Es por ello que resulta interesante tener instalado el entorno de programación de Arduino.

Desgraciadamente la versión que se instala desde sus repositorios es extremadamente antigua, con lo que es mejor instalarlo directamente desde la web de Arduino.

Descargamos desde la [página de descargas de arduino.cc](https://www.arduino.cc/en/Main/Software) la versión del [IDE de Arduino para Linux ARM para 32bits](https://www.arduino.cc/download_handler.php?f=/arduino-1.8.13-linuxarm.tar.xz)

Una vez descargado extraemos el contenido con

tar xvf Downloads/arduino-1.8.13-linuxarm.tar.xz

Ejecutamos **Install.sh** para crear el acceso directo de los menús y dar los permisos necesarios.

Cuando termine la instalación intentamos abrir desde el menú (en el vídeo nos da un error que vemos cómo arreglar)

Si queremos abrir el entorno desde consola, sólo tenemos que entrar en la correspondiente carpeta y ejecutar

./arduino

[](https://youtu.be/-PdmFyhnQV0)

[Vídeo: Instalación del IDE de Arduino en Raspberry Pi](https://youtu.be/-PdmFyhnQV0)

Más adelante en el curso veremos cómo utilizar Arduino junto con Raspberry en nuestros proyectos.

### Fritzing

Otra herramienta relacionada con los proyectos de robótica y electrónica es Fritzing que también podemos usar en nuestra Raspberry Pi. Para ello la instalamos con

En este caso la versión disponible para Raspberry en los repositorios, aunque no es la última, sí que es bastante reciente y usable.

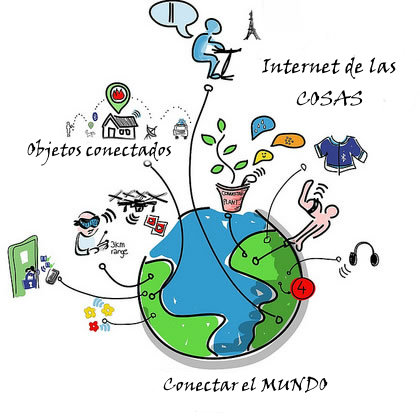
Podemos instalarla desde la herramienta de installación de software o desde consola:

sudo apt install fritzing

[](https://youtu.be/P_-ZmPEDHzs)

[Vídeo: Instalación de Fritzing en Raspberry Pi](https://youtu.be/P_-ZmPEDHzs)

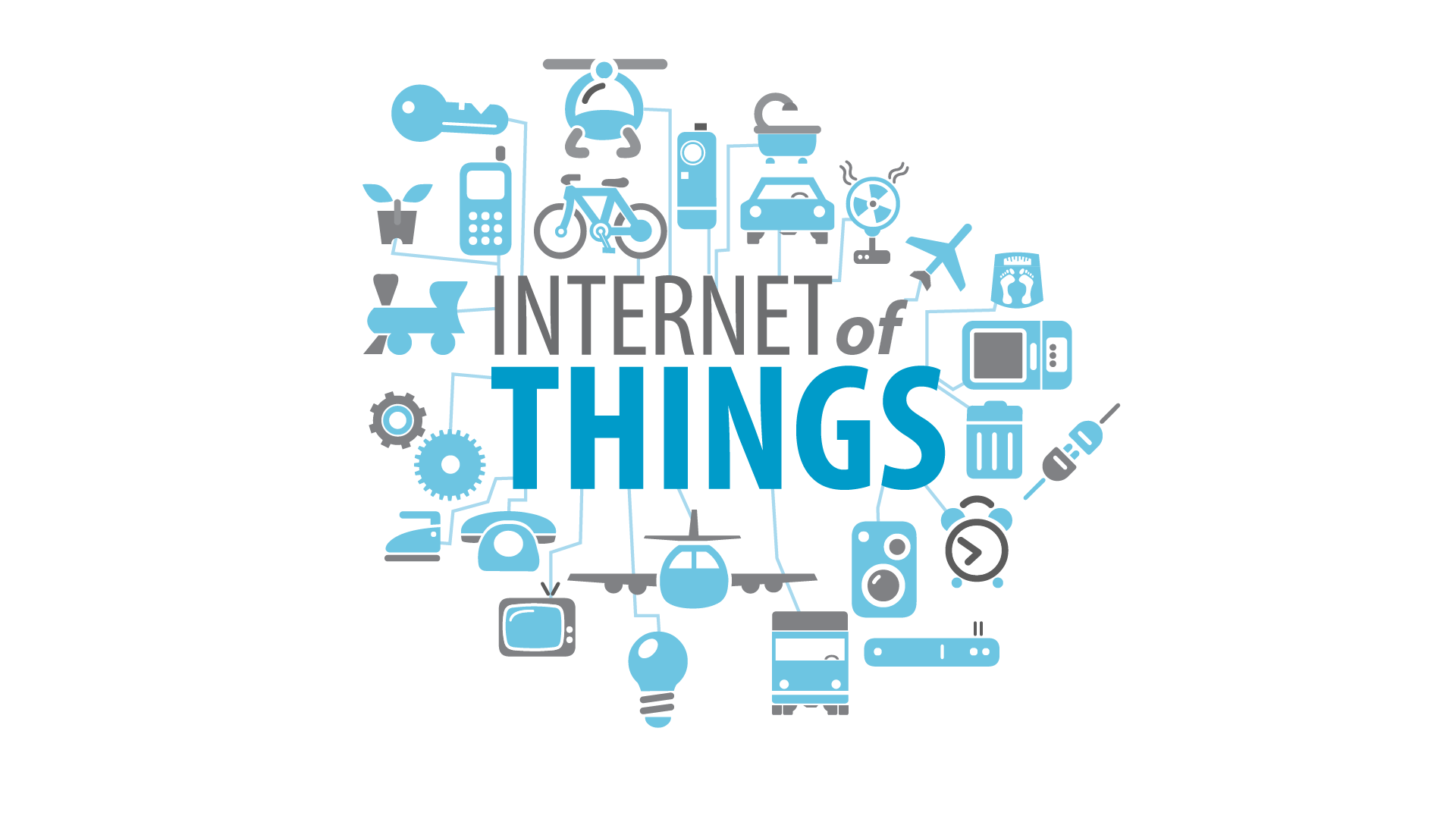
## [¿Qué es IOT?](https://es.wikipedia.org/wiki/Internet_de_las_cosas)



IOT

(De Drawed by Wilgengebroed on Flickr - Translated by Prades97, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=32745149)

¿Qué dispositivos IOT conoces?



Dispositivos IOT

## IOT con Raspberry

Dada la capacidad de procesamiento y la compatibilidad con todo tipo de sistemas, Rasspberry Pi es ideal para ser el centro de las comunicaciones de un sistema IOT. Puede actuar como: \* Broker de comunicaciones, a donde se conectan todos los otros dispositivos. \* Base de datos para guardar los datos. \* Visualizador de datos usando aplicaciones específicas. \* Hacer datamining sobre los datos guardados \* Publicar contenidos en forma de página web, canal de Telegram, etc

# IOT (Internet de las cosas)

¿Qué dispositivos te gustaría controlar remotamente?

* Calefacción.
* Aire acondicionado.
* Encender la cocina para calentar la comida antes de que lleguemos a casa.
* ¿Estás seguro que se han apagado todas las luces de la casa al salir?.
* Una alarma que nos avise de si hay algún problema cuando no estamos en casa.
* Poder ver el estado de un familiar remotamente.
* Controlar el ejercicio que hace durante un día para así motivarte a hacer más.
* Todas las aplicaciones que si tu curro de la domótica de una casa.

## Características de los dispositivos IOT

* Bajo consumo o mejor autosuficiente (un panel solar y una batería). Si tiene una pulsera de actividad sabrá que es necesario cargarla una vez al mes.
* Conectividad con otros dispositivos como puede ser un teléfono móvil ya sea por medio de Bluetooth y /o WiFi.
* Medir algo: actividad, temperatura, consumo.
* Almacenar los datos, que se guardan hasta que se vuelve a sincronizar.
* Estos datos se suelen subir a páginas web (thingspeak, blynk, adafruit, Cayenne, etc) Dónde se almacenan, y dónde posteriormente podemos construir nuestros graficos e informes.

## Proyectos

Veamos algunos de los proyectos en los que podríamos trabajar:

* Estación meteorológica meditemos las magnitudes Atmosférica, mostrando los datos, guardando los y publicándolos en internet parado su posterior análisis.
* Medida de consumo y fomento de la eficiencia energética: mediremos el consumo de nuestro propio equipo y veremos como a medida que vamos incluyendo más componentes el consumo se hace mayor. Podríamos podríamos adaptar este sistema aún uso real dentro de la casa y viendo el consumo de los electrodomésticos.
* Control remoto de componentes mediante el uso de un relé vamos a poder encender y apagar dispositivo remotamente con nuestro sistema bien por el cambio de alguna tela sistemas que medimos o bien por accionamiento externo.
* Medida de calidad del aire. Usaremos sensores de medida de partículas para determinar la calidad del aire al mismo tiempo también mediremos en el contenido en gases y los parámetros atmosféricos. Publicaremos todos estos datos en una página web para poder comparar la medida en diferentes lugares.

## Algunos ejemplos:

* Pulsera para medir la actividad física.
* Su batería dura unos ¡¡¡30 días!!!
* Almacena sus datos (en local) y cuando puede se sincroniza (guardando datos en la nube).
* Sólo proporciona información que luego vemos en otros dispositivos.
* Sistema de riego inteligente.
* Un Arduino que mide la humedad y decide si activar o no una bomba de agua es una prueba de concepto.
* Sistema de riego para una instalación de gran tamaño:
  + Cientos/miles de sensores de humedad.
  + Multitud de puntos de riego (válvulas/bombas).
  + Hacemos un datamining sobre los datos de humedad y se activan los puntos de riego.

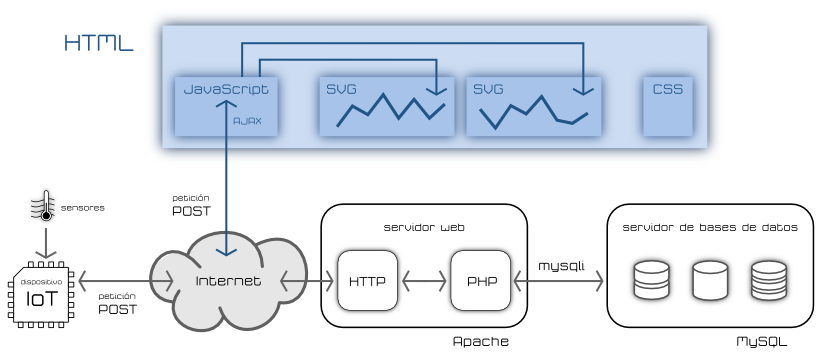
### Qué le pedimos al IOT

* Inteligencia:
  + [Niveles](http://www.domodesk.com/a-fondo-que-es-el-internet-de-las-cosas): identidad, ubicación, estado, contexto, criterio. Ejemplo: los cientos de sensores de humedad de una gran explotación. Un nivel más de Inteligencia sería incluir predicción (por ejemplo la atmosférica: si sabemos a que va llover, esperamos antes de regar).
* Arquitectura
* [Protocolos](http://www.domodesk.com/a-fondo-que-es-el-internet-de-las-cosas): Cable, Wifi, Zigbee, bluetooth, GSM (y todas sus Gs), ...
* M2M: Comunicación Machine To Machine.
  + Un drone recoge datos del nivel de suciedad de los paneles solares.
  + Envía los datos a la central.
  + Cuando el nivel es el adecuado se activa el robot de limpieza en determinada zona.

## ¿Cómo hacerse un IOT?

¿Qué necesito?

## [Arquitectura de un sistema completo IOT](https://polaridad.es/grafica-sensor-internet-de-las-cosas-iot/)



iot

## Publicación en servicios externos

Existen muchos servicios externos donde podemos publicar nuestros datos y representarlos gráficamente.

Estos servicios online, o en la nube como ahora se suele decir, exponen una serie de reglas para acceder a ellos.

Es lo que se conoce de forma genérica como un API (Application Programming Interface).

Existen multitud de estos servicios, algunos nos permite publicar contenidos en redes sociales (como Twitter y Facebook) otros nos facilitan el almacenar nuestros datos y mostrarlos como gráficas. Además estos servicios  dan una dimensión social a nuestros datos pudiendo compartirlos con otros usuarios.

### Plataformas de publicación de datos

La mayoría de estas plataformas nos permiten subir nuestros datos y obtener gráficas con ellos



Gráficos online de thingSpeak

Casi todas tienen una versión limitada (limitación en la frecuencia de envío de los datos o en el volumen de estos) gratuita y otra profesional de pago.

Además algunas nos permiten establecer disparadores (trigger) para vigilar que algún dato tome cierto valor, en cuyo caso envían un aviso a nuestro sistema.

* Adafruit IO https://io.adafruit.com/
* Blynk https://blynk.io/
* Cayenne https://mydevices.com/
* Grafana https://grafana.com/
* Connect2me https://www.c2m.net
* All Things Talk https://www.allthingstalk.com/
* Thingspeak https://thingspeak.com/
* Xively https://www.xively.com/

## Plataforma Xively

Uno de los más conocidos servicios que nos permiten gestionar y visualizar nuestros datos es el antes gratuito y conocido como Pachube, luego llamado Cosm y de momento [Xively](https://xively.com/).

Veamos cómo acceder a este servicio:

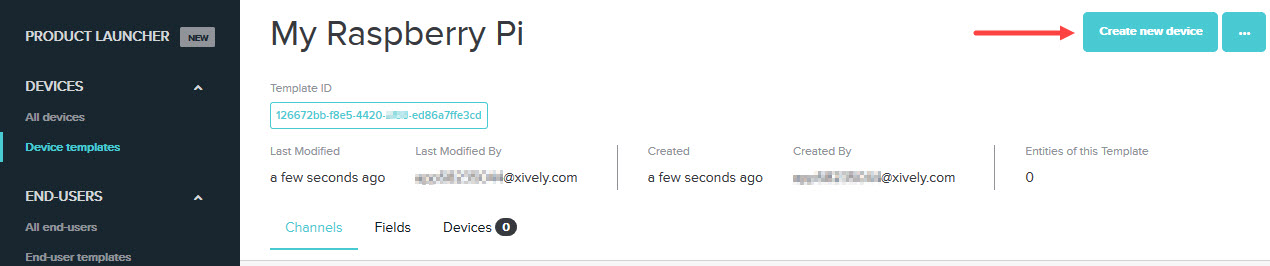
* Accedemos a la página de desarrolladores https://www.developerxively.com
* Buscamos los detalles para nuestro hardware, https://www.developerxively.com/docs/raspberry-pi. Vemos que en el listado no aparece ninguna de las placas Arduino, eso se debe a que Google, el actual dueño de Xively considera que ninguna de las placas Arduino puede garantizar comunicaciones seguras.

Aquí nos indica como configurar nuestra placa y descargar un ejemplo en el que cambiaremos nuestro API Key (código que obtenemos al registrarnos).

Instalamos la librería de xively con

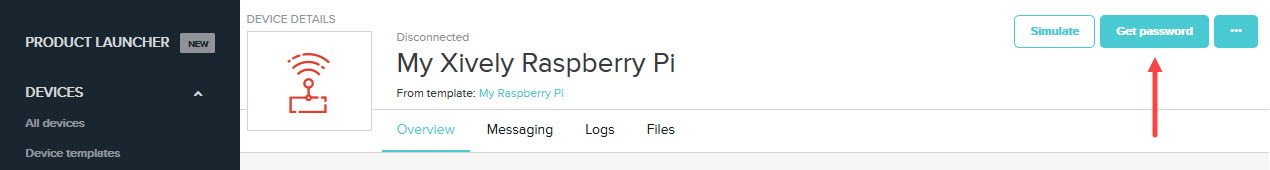
pip3 install --user --pre xiP

Creamos un dispositivo en la plataforma



Alta del dispositivo en Xively

Damos de alta nuestro dispositivo para obtener las credenciales en forma de fichero MQTTCredentials.txt



Descargamos las credenciales

Descargamos la aplicación de ejemplo con

wget https://s3.amazonaws.com/xipy-examples/xively\_pub.py

Probamos a publicar contenidos con (tenemos el código python y el fichero MQTTCredentials.txt en la misma carpeta)

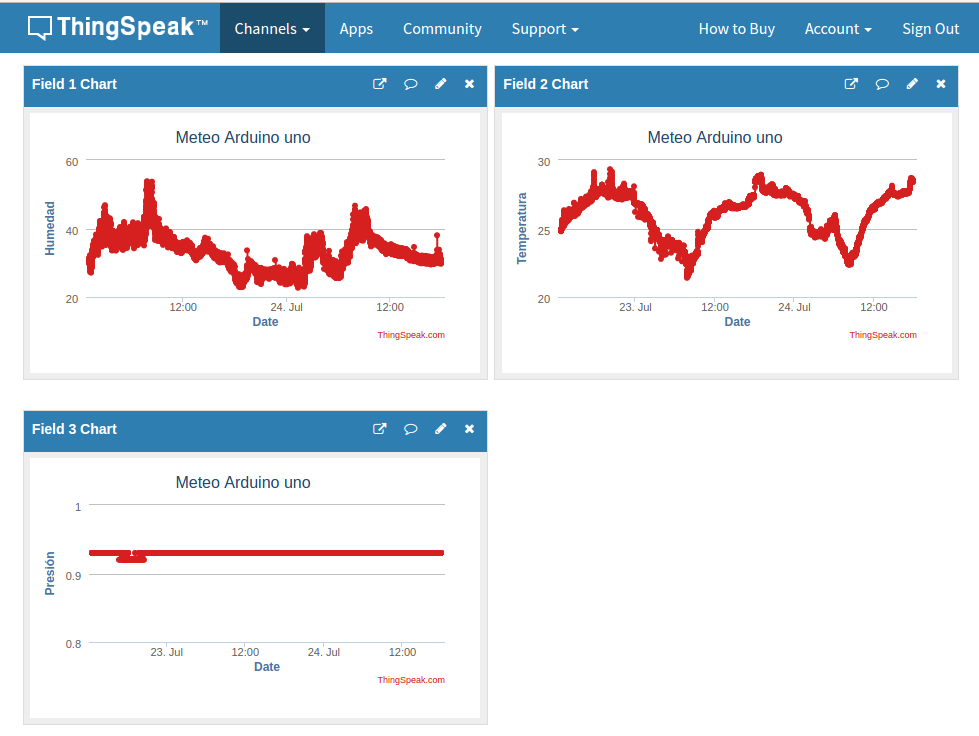
python xively\_pub.py --message 'Hola desde nuestra Raspberry Pi'

Si todo ha ido bien podremos ver nuestro mensaje en la pestaña de del navegador de uestro canal.

[Documentación](https://www.developerxively.com/docs/raspberry-pi#create-a-device-template)

# Publicación en ThingSpeak

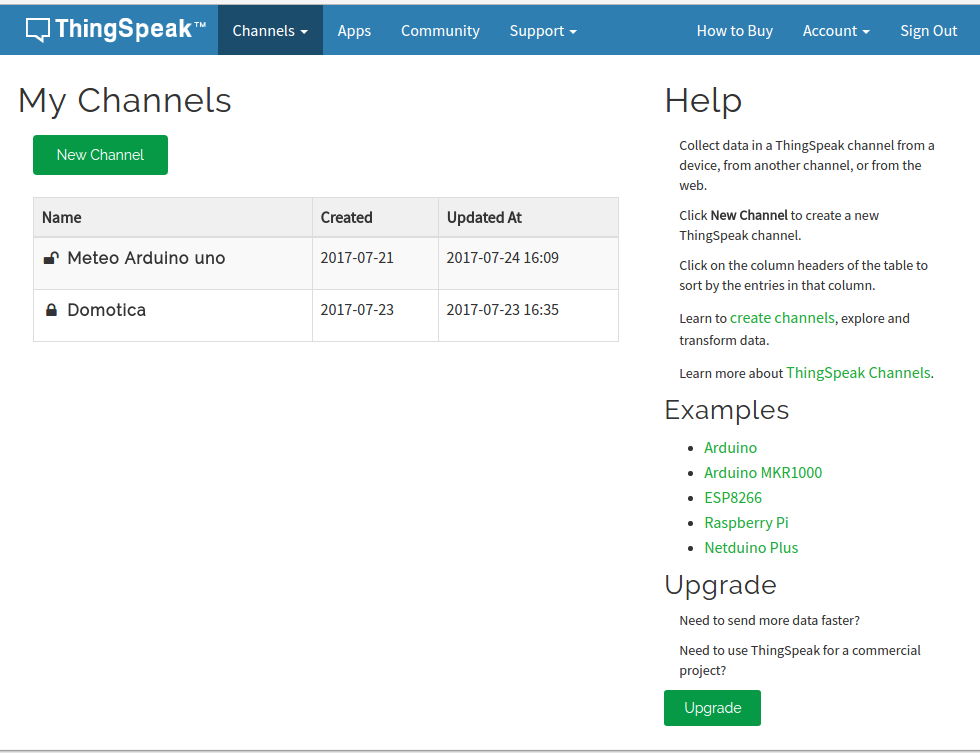
ThingSpeak es una servicio web que nos permite publicar datos de las medidas de nuestros dispositivos IOT (o de cualquier otro).



Es gratuito para cierto número de datos y nos permite de manera muy sencilla subir datos.

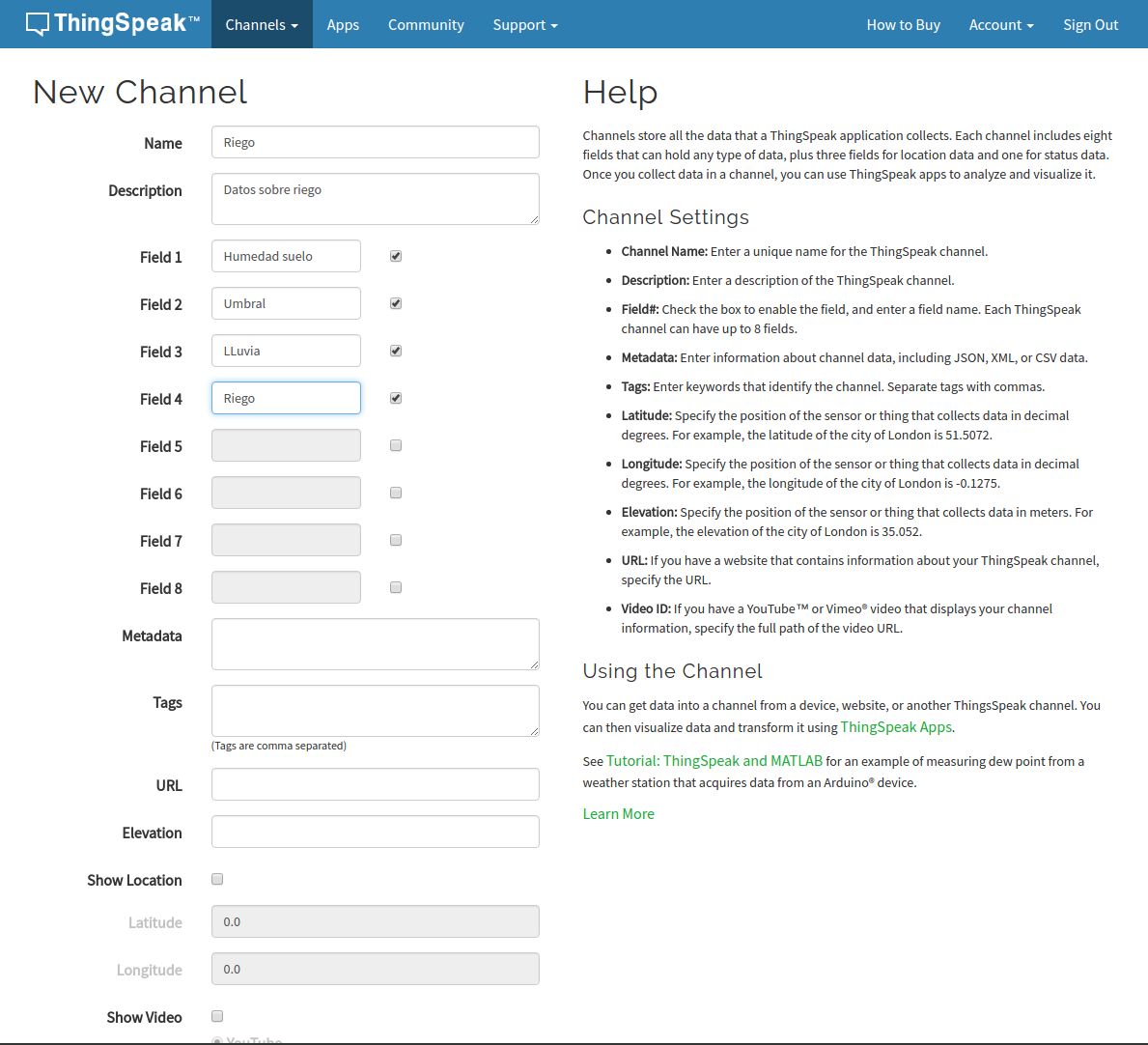
## Creación del canal (Channel)

* Nos hacemos una cuenta en ThingSpeak, recibiremos un email y lo verificamos.
* Entramos en Channels->My Channels y pulsamos en "New Channel".



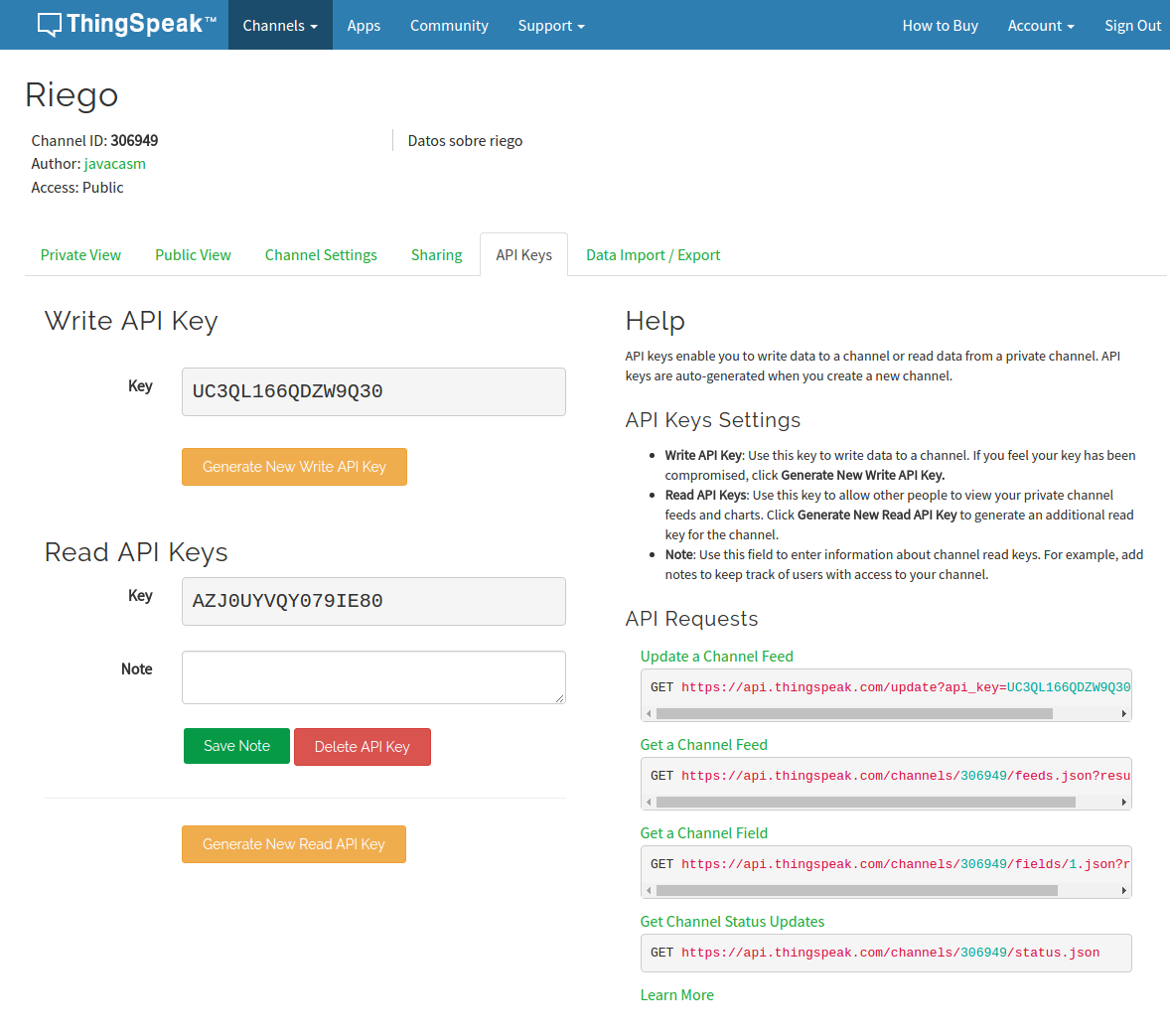
ThingSpeak New Channel

* Configuraremos el canal, indicando los datos que se van a enviar. Podemos añadir una descripción y datos como la web, canal de youtube, etc...



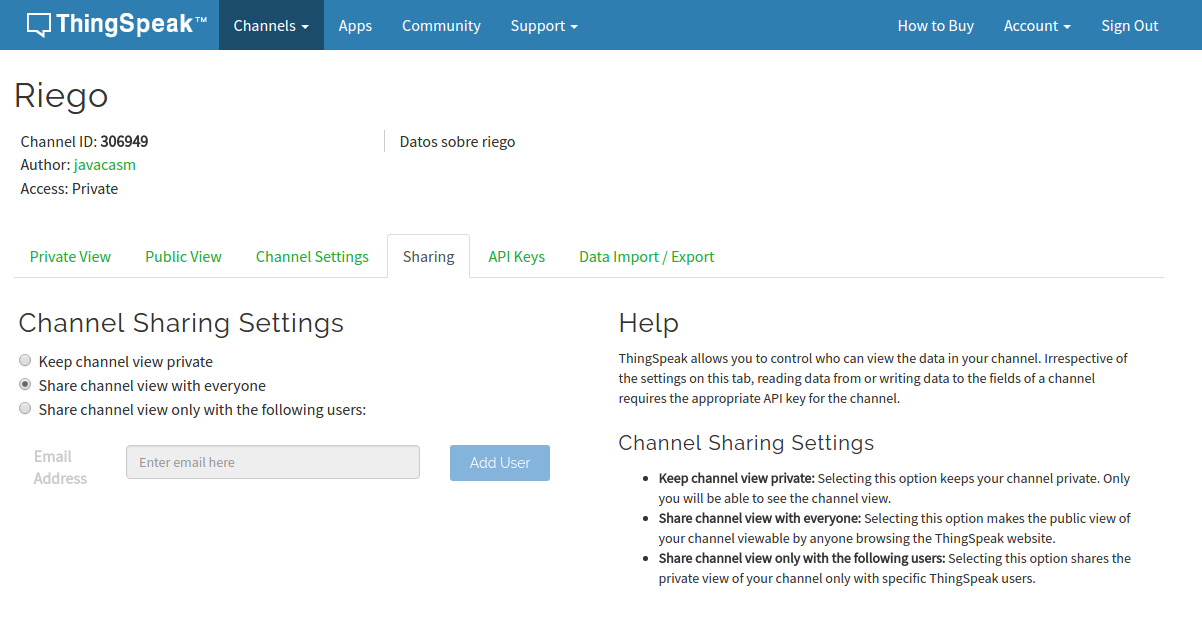
Configuración del canal

* Para poder enviar datos al canal necesitamos el API KEY que lo identifica que incluiremos en nuestro código.



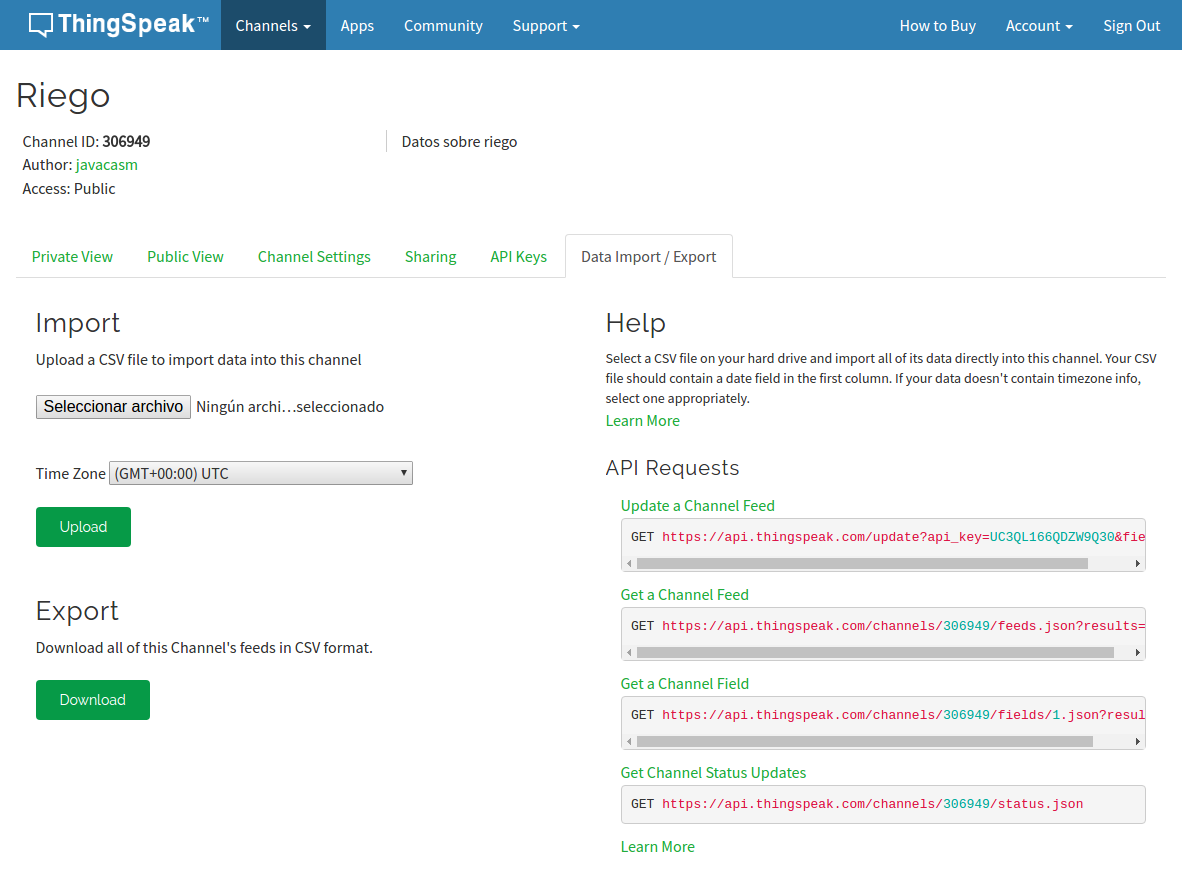
ThingSpeak API Keys

* Para que cualquiera pueda ver los datos, podemos hacer que el canal sea público, desde la pestaña Sharing.

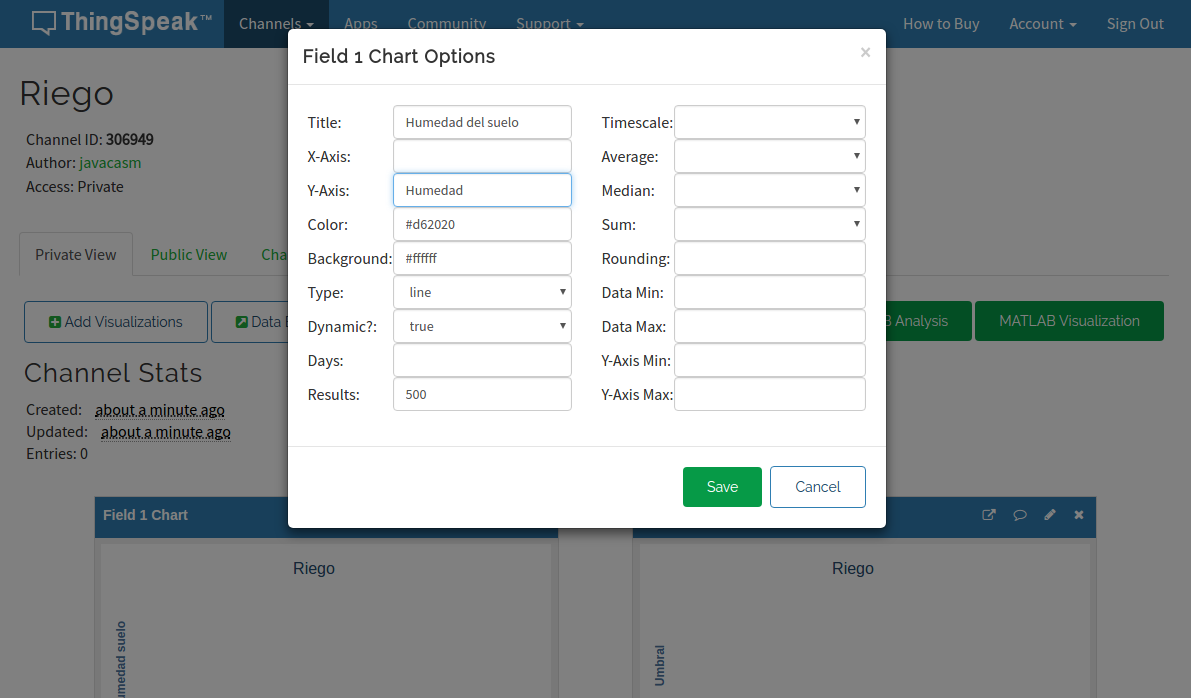


Sharing

* Una vez creado el canal podemos configurar los detalles de cada gráfico, para lo que pulsaremos sobre el icono "lápiz" de cada uno.
* En cualquier momento podemos importar/exportar los datos de un gráfico dado.



ThingSpeak Import-Export



ThingSpeak Configuración de un gráfico

## Código Raspberry

Éste es el [código de ejemplo para Raspberry Pi](https://iotdesignpro.com/projects/how-to-send-data-to-thingspeak-cloud-using-raspberry-pi) que nos va a enviar datos sobre el uso de CPU de nuestra Raspberry

import httplib  
import urllib  
import time  
key = "ABCD" # Put your API Key here  
def thermometer():  
 while True:  
 #Calculate CPU temperature of Raspberry Pi in Degrees C  
 temp = int(open('/sys/class/thermal/thermal\_zone0/temp').read()) / 1e3 # Get Raspberry Pi CPU temp  
 params = urllib.urlencode({'field1': temp, 'key':key })   
 headers = {"Content-typZZe": "application/x-www-form-urlencoded","Accept": "text/plain"}  
 conn = httplib.HTTPConnection("api.thingspeak.com:80")  
 try:  
 conn.request("POST", "/update", params, headers)  
 response = conn.getresponse()  
 print temp  
 print response.status, response.reason  
 data = response.read()  
 conn.close()  
 except:  
 print "connection failed"  
 break  
if \_\_name\_\_ == "\_\_main\_\_":  
 while True:  
 thermometer()

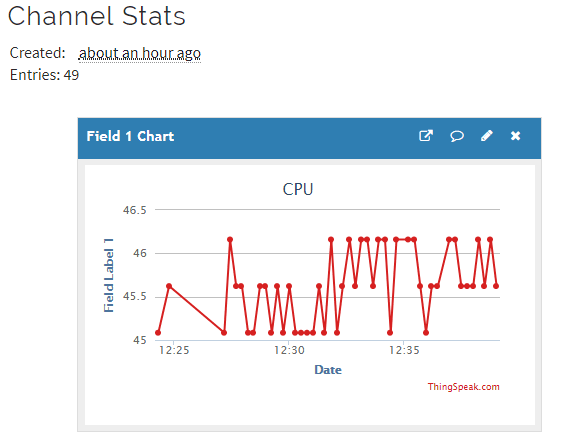
Para ejecutarlo necestiamos tener instalados los módulos **httplib** y **urllib**

sudo apt-get install httplib  
sudo apt-get install urllib

Ahora si lo ejecutamos

python cpu.py

podremos ver el gráfico en su correspondiente canal



Check ThingSpeak channel

#### Tutoriales para otras plataformas

La mayoría de las plataformas tienen tutoriales detallados, y muchas de ellas como por ejemplo Blynk incluso generan el código necesario para utilizarlas, con lo que nosotros sólo tenemos que personalizarlo con nuestro sensores y preferencias.

[Blynk](https://blynk.io/en/getting-started) [Instructable](https://www.instructables.com/id/Blynk-JavaScript-in-20-minutes-Raspberry-Pi-Edison/)

[Cayenne](https://www.instructables.com/id/Platform-IoT-Cayenne-Mydevices-ESP8266-12E-NodeMCU/) ([Librería mqtt](https://github.com/myDevicesIoT/Cayenne-MQTT-ESP))

[Grafana](https://www.spainlabs.com/foros/tema-SpainLabsIoT2018-Grafana-Dashboard-Open-Source)

Más adelante, cuando ya sepamos programar haremos otros ejemplos de publicación como por ejemplo vía Telegram.

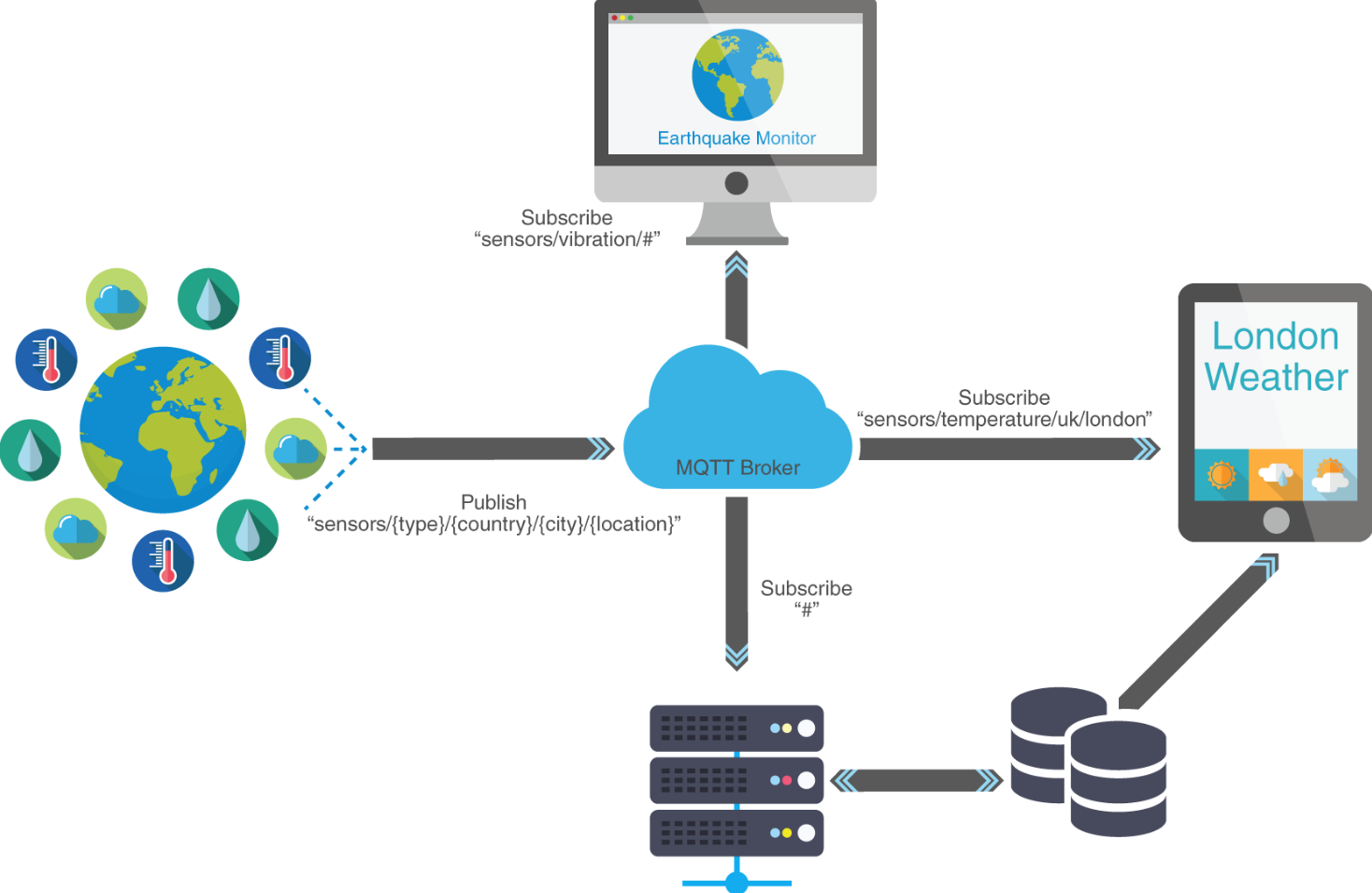
## MQTT

MQTT es un servicio de comunicaciones entre dispositivos sencillo y ligero. Está pensado para que pueda funcionar en equipos con poca capacidad de cálculo.

Los equipos envían y reciben mensajes que está formatos por un **Topic** que es como una etiqueta que tiene estructura arbórea y un mensaje o contenido.

Todos los participantes pueden **publicar** mensajes y/o también se pueden **suscribir** a determinados topics, de manera que reciban los mensajes con ese topic.

Utiliza una arquitectura como la que se ve en la imagen, donde el sistema central actúa como **Broker**, recibiendo los mensajes de todos los equipos y notificando a aquellos que se han suscrito a topics.



Arquitectura MQTT

Al ser un servicio sencillo actúa como transporte en sistemas más complejos.

Existen brokers accesibles a través de internet como por ejemplo

# https://diyodemag.com/projects/part\_2\_google\_assistant\_controlled\_devices

# MQTT

## Instalación en Raspberry

Instalamos servidor **mosquitto** que actúa como broker en la Raspberry:

sudo apt install mosquitto

Ejecutamos mosquitto

Si queremos que se arranque como servicio al iniciar la raspberry, hacemos

sudo systemctl enable mosquitto.service

Para depurar el funcionamiento de mosquitto y ver los logs cuando funciona como servicio podemos usar

https://community.home-assistant.io/t/how-to-debug-mosquitto-mqtt/107709/20 http://www.steves-internet-guide.com/mosquitto-logging/ https://github.com/thomasnordquist/MQTT-Explorer

Para publicar y recibir mensajes necesitaremos las herramientas cliente, que podemos instalar con

sudo apt install mosquitto-clients

Podemos suscribirnos a un tema/topic con el comando

mosquitto\_sub -h servidorMQTT -t Tema

Para publicar en un "Topic" un "Mensaje" (siempre son cadenas)

mosquitto\_pub -h servidorMQTT -t "Topic" -m "Mensaje"

## Ejemplos

Vamos a suscribirnos al topic "MeteoSalon/#", es decir a todos los mensajes que "cuelgen" del topic "MeteoSalon". La opción **-v** es para que muestre más detalles sobre los mensajes

mosquitto\_sub -h 192.168.1.200 -t "MeteoSalon/#" -v

y la aplicación quedará esperando hasta que se reciban mensajes con un topic compatible

Desde el mismo servidor podemos probar que funciona con la utilidad **mosquitto\_pub**

mosquitto\_pub -h 192.168.1.200 -t "MeteoSalon/led" -m "On"

En el servidor vemos la siguiente traza

1574598811: New connection from 192.168.1.200 on port 1883.  
1574598811: New client connected from 192.168.1.200 as mosqpub/7375-raspberryp (c1, k60).  
1574598811: Client mosqpub/7375-raspberryp disconnected.

y en la aplicación cliente

MeteoSalon/led On

### Ejemplo de arquitectura de topics

A medida que vamos añadiendo dispositivos y enviado más mensajes se puede complicar el árbol de topics

Para ellos es mejor usar una arquitectura. Por ejemplo esta [tomada del blog de ricardo veal](https://ricveal.com/blog/sonoff-mqtt/)

state\_topic: "stat/sonoff/1/POWER"  
command\_topic: "cmnd/sonoff/1/POWER"  
availability\_topic: "tele/sonoff/1/LWT"

Telemetría para que cuenten sus cosas ¿Por ejemplo los sensores? Command para peticiones ¿request? Stat para confirmaciones de estados

## Recursos

[Instalación de mosquito en la Raspberry)[https://randomnerdtutorials.com/how-to-install-mosquitto-broker-on-raspberry-pi/]

https://randomnerdtutorials.com/micropython-mqtt-esp32-esp8266

https://geekytheory.com/tutorial-raspberry-pi-gpio-y-mqtt-parte-1

## Conexión con Zigbee

zigbee2mqtt Pasarele zigbee mqtt integraci'on con Home Assistant https://www.zigbee2mqtt.io/integration/home\_assistant.html

## Recursos

[Domótica por Jorge Pascual](https://www.youtube.com/watch?v=IQLFgVg4TjM&list=PLFe_vhJmgS_51XlV_a5Atl1Re4fJA_b-d)

# Home automation

# Domótica con Home Assistant

https://learn.adafruit.com/set-up-home-assistant-with-a-raspberry-pi?view=all

[Instalacion de la imagen](https://home-assistant.io/hassio/installation/) usando Balena Etcher https://www.balena.io/etcher/

[Instalacion manual](https://www.home-assistant.io/docs/installation/raspberry-pi/)

[Conexión con Google Assistant](https://www.home-assistant.io/addons/google_assistant/)

[Getting started with Home Assistant](https://randomnerdtutorials.com/getting-started-with-home-assistant-on-raspberry-pi/#more-43192)

https://www.hackster.io/ahmedibrrahim/smart-home-automation-iot-using-raspberry-pi-and-python-47fb62

http://mbrobotics.es/blog/homekit-raspberry-pi-2-v2/

# Google Assistant

https://diyodemag.com/projects/part\_2\_google\_assistant\_controlled\_devices

https://diyodemag.com/projects/part\_1\_google\_assistant\_controlled\_devices

[Google Home para gobernalos a todos](https://www.youtube.com/watch?v=0i8ROl1KS3Y&list=PLFe_vhJmgS_51XlV_a5Atl1Re4fJA_b-d&index=24)

# Retropie

https://www.programoergosum.es/tutoriales/consola-arcade-basada-en-raspberry-pi-con-retropie

https://www.raspberrypi.org/blog/retro-console-with-retropie-raspberry-pi-1/

https://www.raspberrypi.org/blog/retro-console-with-retropie-raspberry-pi-2/

https://blog.bricogeek.com/noticias/raspberry-pi/video-tutorial-recalbox-batocera-para-raspberry-pi/

https://www.hwlibre.com/retropie/?utm\_source=dlvr.it&utm\_medium=twitter ## Raspberry como centro multimedia

Existen varias distribuciones ... xbmc elec...

siempre podemos instalar ... ## Raspberry como centro multimedia: Kodi / libreelec

La distribución LibreELEC está pensada para usar nuestra Raspberry como un centro multimedia, conectándola a un Televisor y con todo lo necesario para poder reproducir tanto vídeo, música o imágenes.

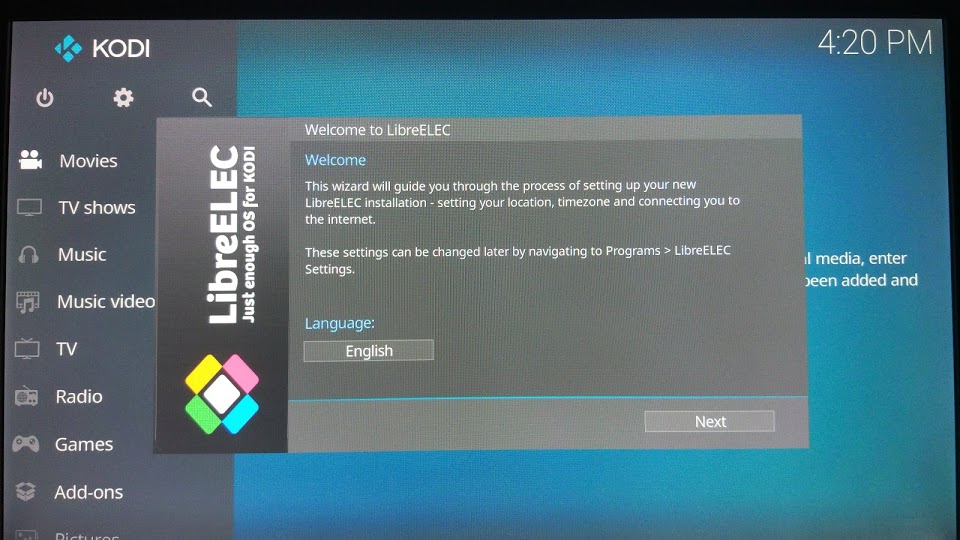
El centro de la distribución es Kodi, un reproductor multimedia muy avanzado y por su naturaleza modular podemos adaptar para todo tipo de contenidos. Además podemos personalizarlo instalando Add-ons (complementos) para reproducir y para obtener nuestros contenidos.



LibreELEC

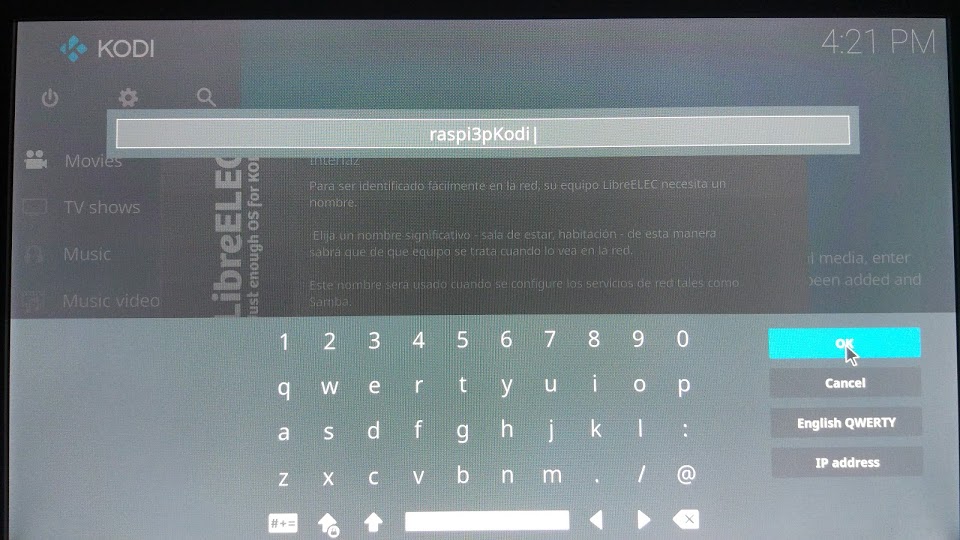
Una vez instalada la distribución LibreELEC la seleccionamos en el arranque y entramos directamente a Kodi

La primera vez tendremos que seleccionar el idioma



Selección de idioma en Kodi

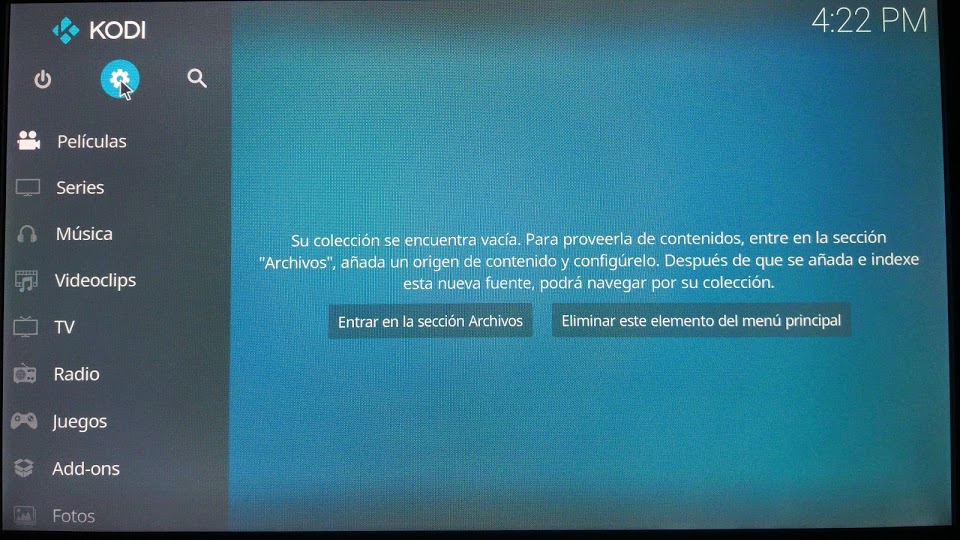
Vemos como el interface está perfectamente adaptado a pantallas grandes y no necesitamos un teclado físico, si no que podemos usar un simple puntero/ratón.



Teclado en pantalla de Kodi

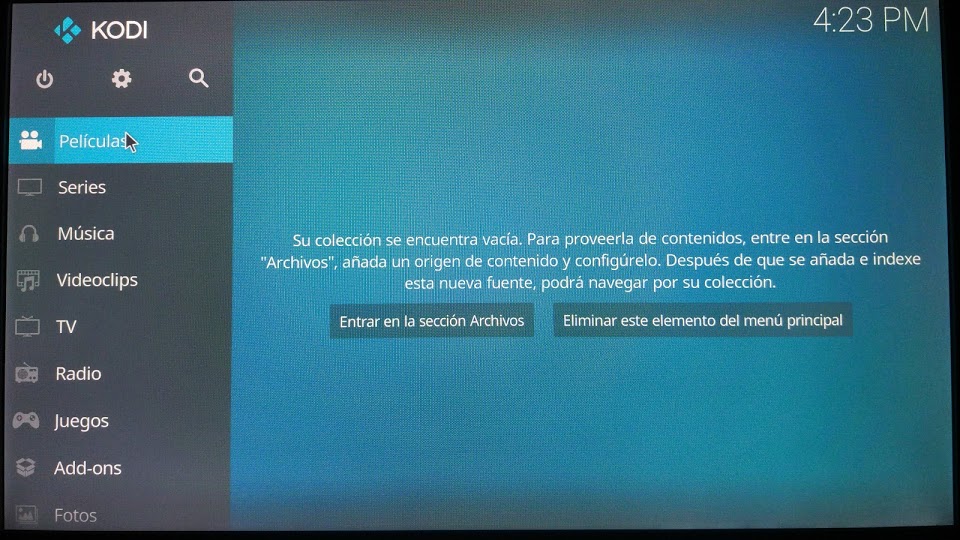
Damos un nombre al equipo y configuramos el acceso remoto por ssh (para acceder desde otro equipo) y por samba (para poder acceder a carpetas compartidas por equipo Windows o NAS), para lo que nos solicitará una contraseña.

Después de ésto se actualizará y estará listo para usarlo



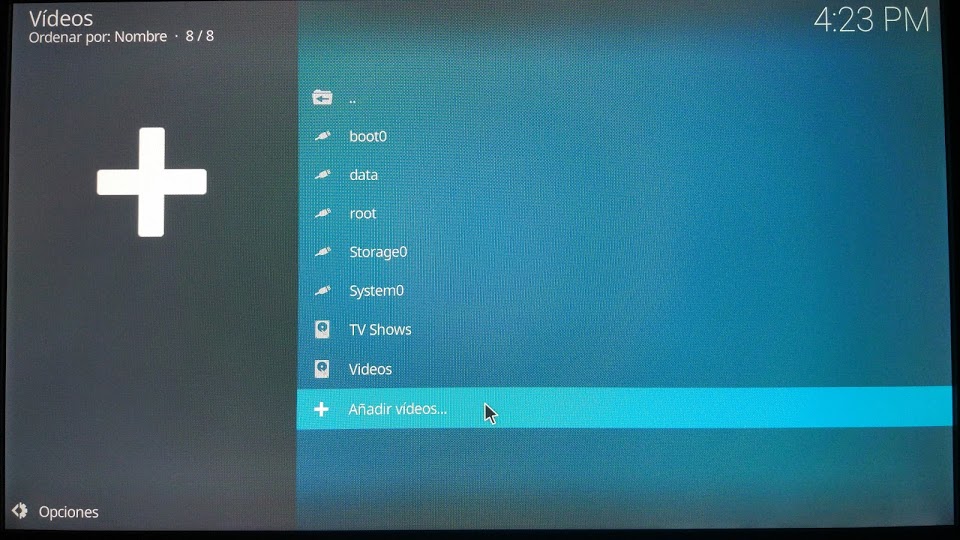
Pantalla de Kodi

Para añadir las carpetas locales o remotes donde se almacenan los contenidos, entramos en el tipo que queramos (Películas en este ejemplo) y pulsamos sobre "Entrar en la sección de Archivos"



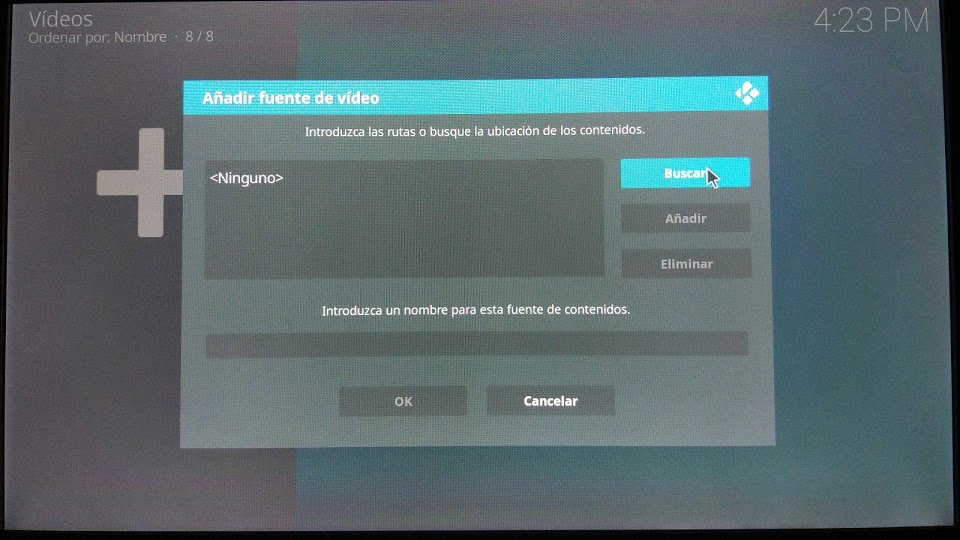
Añadiendo carpetas de contenidos

Pulsamos en "Añadir vídeos"



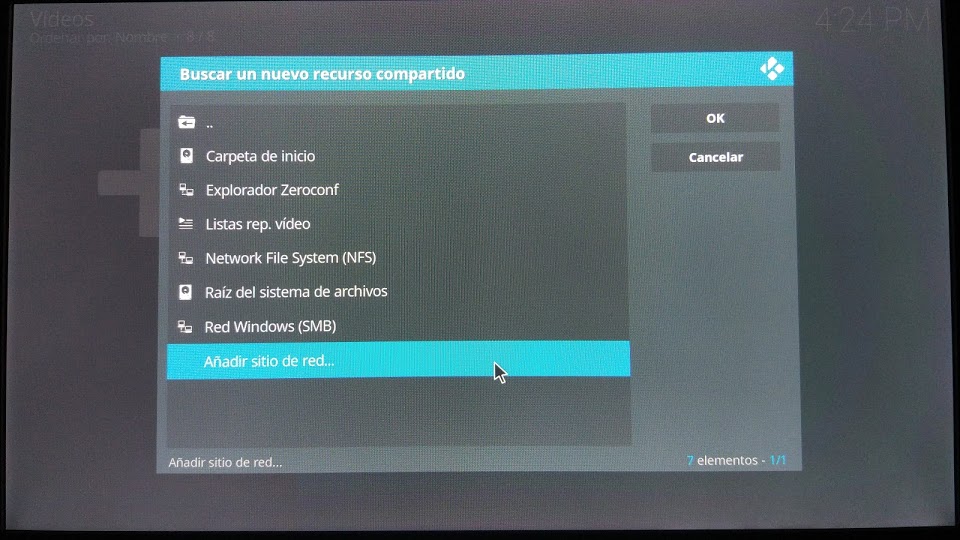
Añadiendo vídeos a Kodi

Pulsamos "Buscar"



Buscar contenidos en Kodi

Y luego los buscamos en las opciones de red que nos aparece o pulsamos sobre "Añadir sitio de red" para rellenar los datos de la ip, nombre de carpeta, usuario y password



Añadir sitio de red

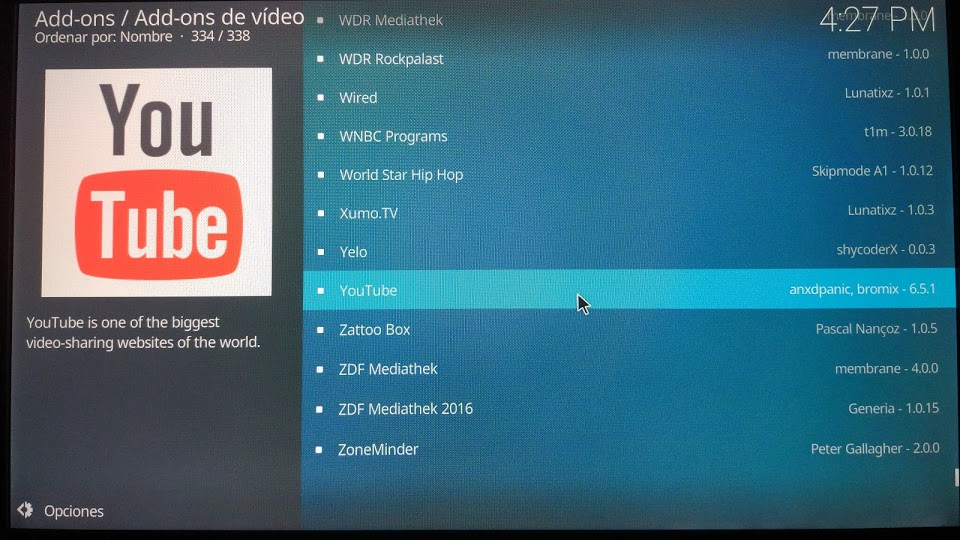
Aceptamos y ya debemos ver la carpeta para incluirla entre los contenidos multimedia.

También podemos añadir Add-ons entrando en la sección Add-ons y en Descargar. Vamos a ver cómo instalar el Add-ons de youtube



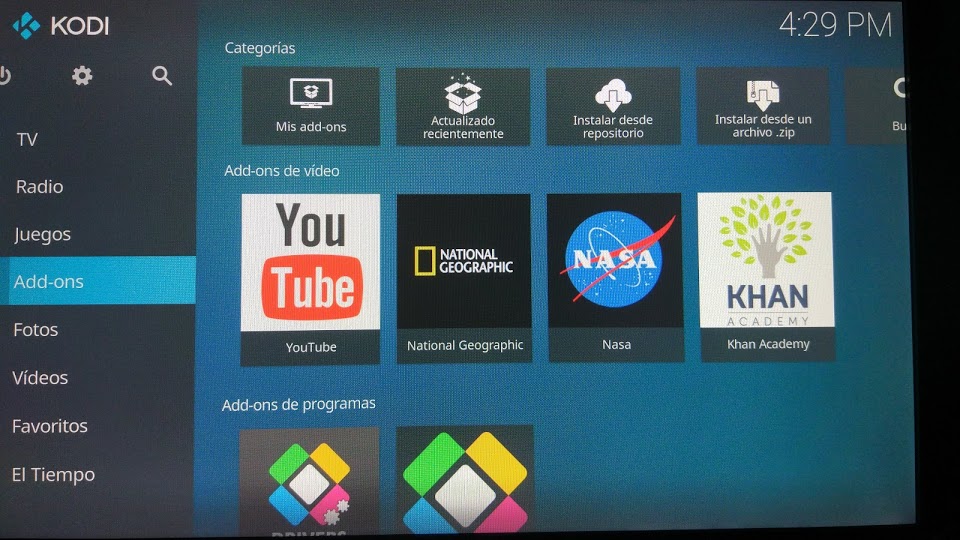
Descargar Add-ons

Seleccionamos el tipo (vídeo en nuestro caso) y buscamos el que queramos



Addons de youtube

Pulsamos instalar y nos dirá si necesitamos instalar algunos complementos de los que depende. Aceptamos y los tendremos disponibles como nueva fuentes de contenido multimedia.



Add-ons de vídeo

Ahora a disfrutar de nuestro reproductor multimedia

### Referencias

https://libreelec.tv/2019/01/libreelec-leia-v8-95-2-beta/ https://www.raspberrypi.org/magpi/best-raspberry-pi-media-players/

con la llegada de la Raspi 4 el panorama ha cambiado bastante: \* Permite arrancar desde USB (todavía en Beta pero va bastante bien), lo que permite usar un disco SSD barato de capacidad, velocidad y duración mucho mayores que las tarjetas SD. \* La cantidad de RAM hace totalmente posible el uso en el día a día. Yo creo que 4Gb son suficientes para un uso normal, pero si te da el presupuesto incluye 8Gb \* El usar tecaldos y ratones inalámbricos tiene la ventaja de que el consumo por USB es menor \* Arduino y microbit funcionan perfecto, pero hasta donde yo sé Lego no está soportado Necesitamos una tarjeta SD para configurar inicialmente las raspi4, pero ya no para que arranquen en el día a día Yo estoy haciendo pruebas con una versión de 128Gb de este disco https://es.aliexpress.com/item/33053472759.html?spm=a2g0s.9042311.0.0.274263c0sLwovz que me costó 25$ También uso la pantalla de un portátil roto con un adaptador HDMI

Integración herramientas Google

Clonado de tarjetas Onwcloud

## Usando SQLite

Actualizamos

sudo apt update  
sudo apt upgrade

Requisitos https://doc.owncloud.org/server/10.4/admin\_manual/installation/manual\_installation.html#prerequisites

SEguimos las instrucxciones de la propia owncloud https://doc.owncloud.org/server/10.4/admin\_manual/installation/manual\_installation.html#install-the-required-packages

Instalamos apache, php5, curl (Se están usando las ultimas versiones de php a día de hoy 7/5/2020)

sudo apt-get install apache2 php7.3 php7.3-json php7.3-xml php7.3-gd php7.3-sqlite3 curl libcurl4 php7.3-curl php7.3-common php7.3-zip php7.3-xml php7.3-intl php7.3-mbstring

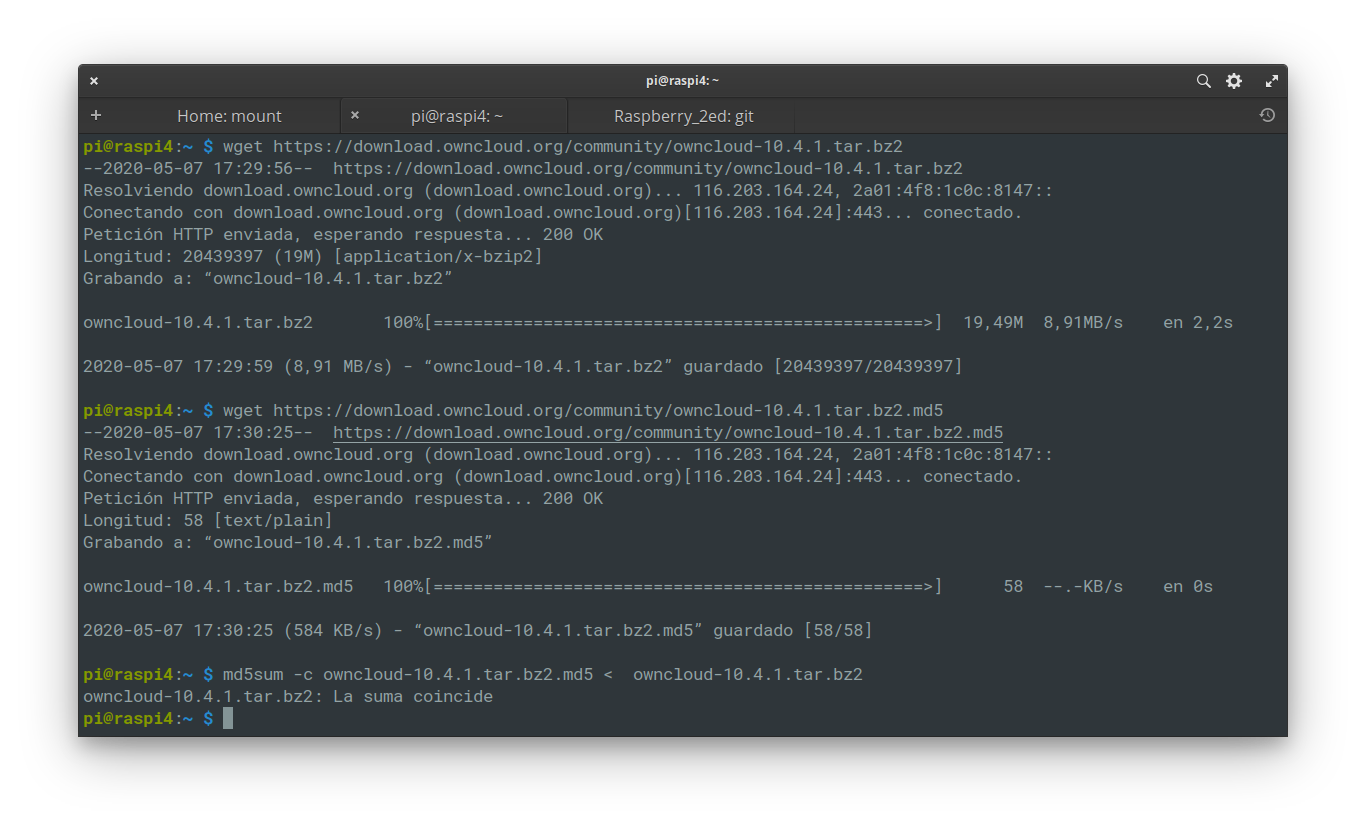
Descargamos la ultima version de owncloud wget https://download.owncloud.org/community/owncloud-10.4.1.tar.bz2

Descargamos el fichero md5

wget https://download.owncloud.org/community/owncloud-10.4.1.tar.bz2.md5

Comprobamos la integridad del fichero

md5sum -c owncloud-10.4.1.tar.bz2.md5 < owncloud-10.4.1.tar.bz2



Check\_md5\_owncloud

Descomprimimos

tar xvf owncloud-10.4.1.tar.bz2

copiamos el contenido en el directorio raiz del servidor apache

sudo cp -r owncloud /var/www/

Creamos el fichero de configuracion del sito en

/etc/apache2/sites-available/owncloud.conf

con

sudo nano /etc/apache2/sites-available/owncloud.conf

Poniendo el siguiente contenido

Alias /owncloud "/var/www/owncloud/"  
  
<Directory /var/www/owncloud/>  
 Options +FollowSymlinks  
 AllowOverride All  
  
 <IfModule mod\_dav.c>  
 Dav off  
 </IfModule>  
</Directory>

y ahora creamos un enlace a /etc/apache2/sites-enabled con

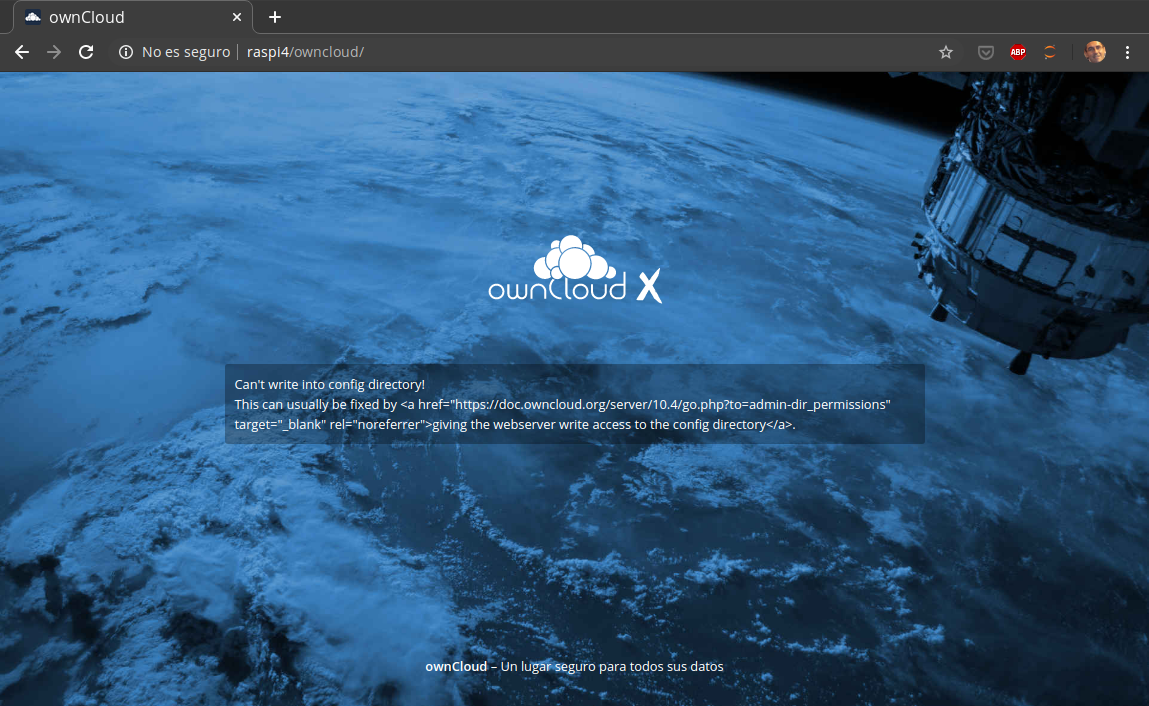
sudo ln -s /etc/apache2/sites-available/owncloud.conf /etc/apache2/sites-enabled/owncloud.conf

Ahora comprobamos que tenemos todos los módulos necesarios de apache activos y reiniciamos el servidor apache

sudo a2enmod headers  
sudo a2enmod env  
sudo a2enmod dir  
sudo a2enmod mime  
sudo a2enmod unique\_id   
sudo systemctl restart apache2

Probamos a acceder

http://raspi4/owncloud/



firstown Cloud

Hasta este punto lo tenemos OK

Vamos a ejecutar la configuración para ello le damos permisos para que el servidor pueda modificar los ficheros de configuración de a la carpeta (esto es un potencial problema de seguridad y por eso viene así por defecto)

Para ello hacemos

sudo chown -R www-data:www-data /var/www/owncloud/

https://geekytheory.com/tutorial-raspberry-pi-crea-una-nube-privada-con-pydio http://www.electroensaimada.com/owncloud.html#

## Usando MariaDB o MySQL

https://computerhoy.com/noticias/tecnologia/haz-raspberry-pi-sea-nube-personal-412645 https://blog.desdelinux.net/convierte-tu-raspberry-pi-en-una-nube-personal-con-owncloud/

BookServer con calibre

## Tu Biblioteca desde cualquier parte del mundo

* Calibre
* Servidor calibre
* Acceso de usuarios
* Abrir puertos
* Compartir archivos
* Backup