

# Servidor para Raspberry Pi desde cero

Cristian González García  
gonzalezcristian@uniovi.es

v 1.0.1 Febrero 2019

# Índice

- ◉ Qué es la Raspberry Pi
  - ◉ Microordenador vs microcontrolador
  - ◉ Posibles componentes del pack
  - ◉ RPI vs Arduino
  - ◉ RPI vs Servidor Clásico vs PC
- ◉ Máquina Virtual
- ◉ Sistema operativo para RPI
- ◉ Preconfiguraciones
- ◉ Raspberry Pi
- ◉ Montar servidor
  - ◉ Apache
  - ◉ PHP
  - ◉ MariaDB
  - ◉ Ejemplo
- ◉ Conexión remota
- ◉ Otro software

¿Qué es?

# Raspberry Pi

- Microordenador
- Bajo coste
- Desarrollado en el Reino Unido
- Objetivo de estimular la enseñanza de la informática
- Software de código abierto
  - SO oficial basado en Debian: Raspbian
  - Admite otros SO: Windows, Ubuntu, etc.
- Actualmente, 6 modelos diferentes
  - 1A, 1B, 1B+, 2B, 3B, 3B+
  - Cada nuevo modelo aumenta la potencia
- Otras placas similares: OrangePi



# Posibles componentes del pack



# RPi vs Arduino

## o RPi

- o Microordenador
- o Admite un sistema operativo
  - o Es igual que cualquier ordenador de sobremesa
    - o Instalar programas, tener aplicaciones, etc.
- o Perfecto para usar como un mini ordenador
  - o PC portátil/remoto, programas con lógica muy compleja (robots más potentes), etc.
- o Pines: digitales y algún digital PWM, no tiene analógicos
  - o Analógicos: necesita de dispositivos externos: Arduino, conversores, electrónica, etc.
- o Más computación que un Arduino

## o Arduino

- o Microcontrolador
- o No tiene sistema operativo, se sube lo que quieres que se ejecute
  - o Ejecuta un programa una y otra vez
  - o No tiene multitarea
- o Perfecto para tareas «simples» y repetitivas
  - o Abrir/cerrar puertas, mover robots, etc.
- o Pines: digitales, analógicos y digitales PWM
- o Muchos escudos (Shields) disponibles



# RPi vs Servidor clásico vs PC

- ◉ **RPi en casa**

- ◉ Sería como tener nuestro PC de casa, por cuestiones de capacidad de red y electricidad, salvo SAI, pero consumiendo menos electricidad
- ◉ Servidor de descarga, servicio propio, centro de ocio, etc.

- ◉ **RPi no es un servidor tipo Amazon o Azure** para ejecutar Hadoop o apps muy pesadas

- ◉ Sí es para ejecutar servicios o aplicaciones pequeñas o servicios propios
- ◉ Incluso algo complejo si tuviéramos un clúster bien montado

- ◉ **RPi ni es malo ni es maravilloso**, es lo que es

- ◉ Tiene su capacidad de computo pequeña, su poca RAM, etc.
- ◉ Bajo precio (≈40€), tamaño y consumo
- ◉ **Consumo** sin incluir teclado, ratón ni monitor, kW/h=0,22
  - ◉ RPi 3 ≈ 4W máximo
  - ◉ PC ≈ 204W ≈ 70€/año - 500W máximo (gráfica + sonido) ≈ 200€/año
  - ◉ <https://outervision.com/power-supply-calculator>

# Máquina Virtual

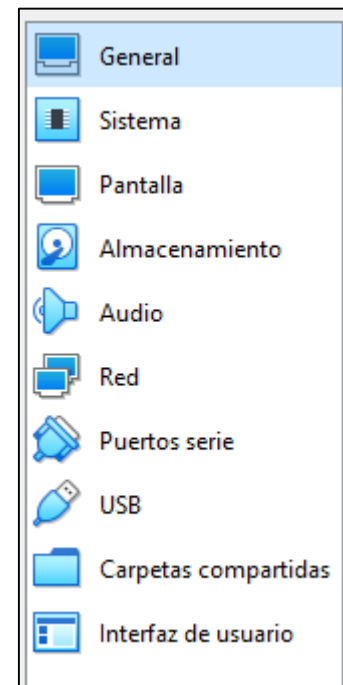


# Virtual Box

- <https://www.virtualbox.org/wiki/Downloads>
  - 5.2.8 (6.0.2)
  - Linux
    - Otro Linux de 32 bits (armv7l)
  - 1024 MB RAM
  - 12 GB HD
  - VDI vs VHD vs VMDK
    - Los 3 permiten la reserva dinámica de disco (crece pero no decrece según requisitos)
    - **VDI**: nativo y original de VirtualBox
    - VHD: nativo y original de Microsoft pero soportado por otros productos
    - VMDK: nativo y original de VMWare pero soportado por otros productos

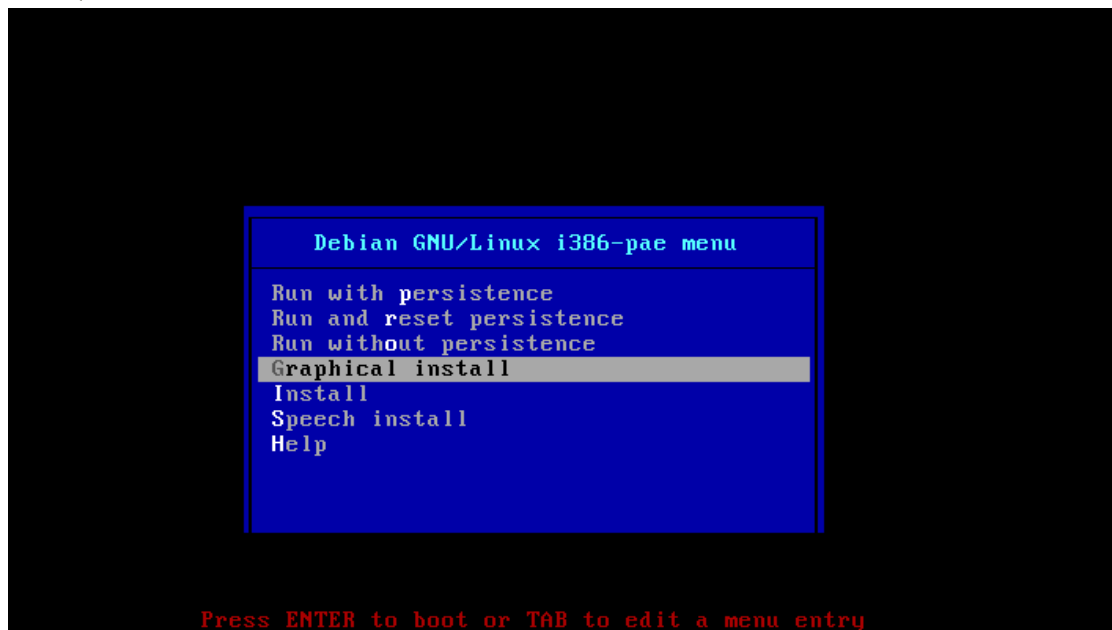
# Virtual Box

- Otras opciones (Configuración utilizada)
  - General Avanzado
    - Compartir cortapapeles y arrastrar elementos
    - Cifrado de disco
  - Sistema
    - Unidades conectadas
    - Procesadores (**2**)
  - Pantalla
    - Memoria de video (**128 MB**)
  - Red
    - NAT
  - Etc.



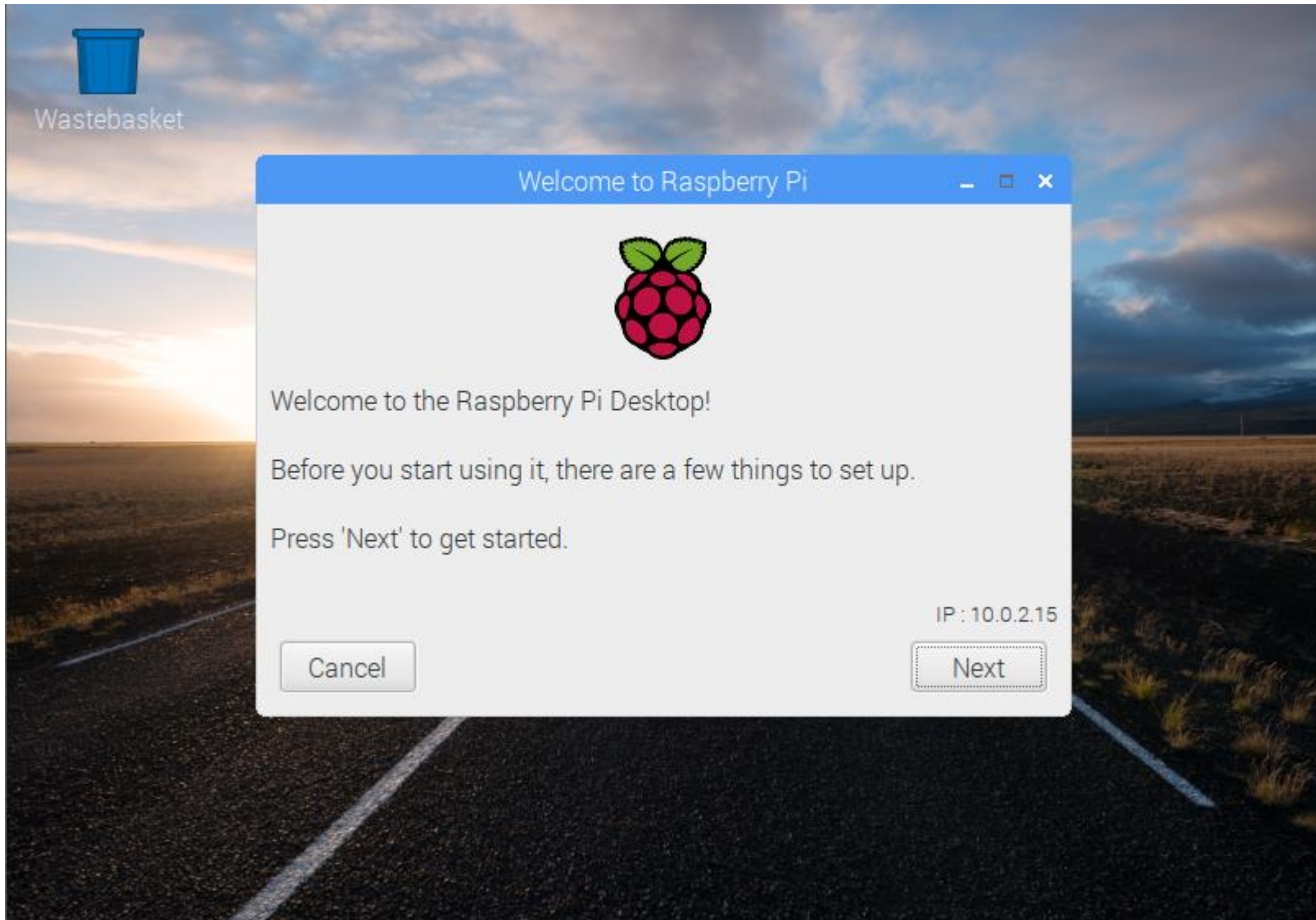
# Instalar la ISO

- Run
  - Ejecuta directamente Raspbian sin instalarlo
    - Con, sin, o reseteando la persistencia
- Instalación
  - Gráfica, comandos o hablada



# Menú de opciones

- Idioma
- Tipo de particionado
  - **Usar disco entero**
    - **Probablemente necesitemos de casi todo el espacio desde el principio, salvo tarjetas muy grandes, y tengamos una sola partición**
  - Usar disco entero y LVM
  - Usar disco entero y LVM encriptado
  - Manual
- Particiones
  - **Todo en una**
    - **Falta de espacio**
  - Separar /home
  - Separar /home, /var y /tmp
- Instalar GRUB en la partición del SO
  - Gestor de arranque del SO



# Opciones a configurar inicialmente

- Configurar
  - Región
  - Idioma
  - Zona horaria
  - Teclado
- Contraseña
  - Cambiar la contraseña por defecto
    - Raspbian
  - Seguridad ante todo
    - Alfanumérica + caracteres especiales
- Actualizar RPi
  - Seguridad, errores, y mejoras

# Sistema Operativo para RPi

# Descargar el SO

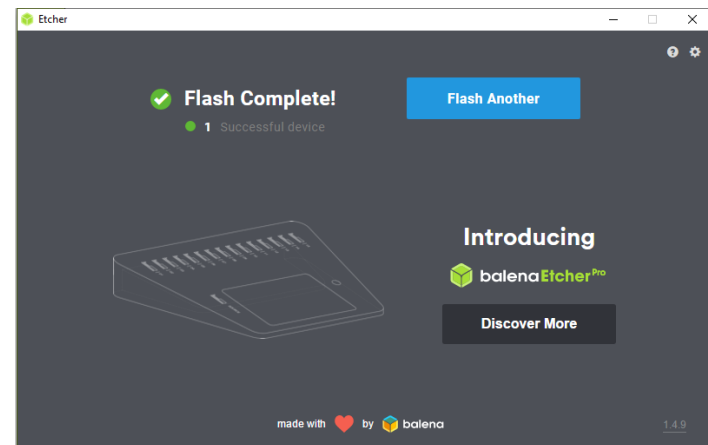
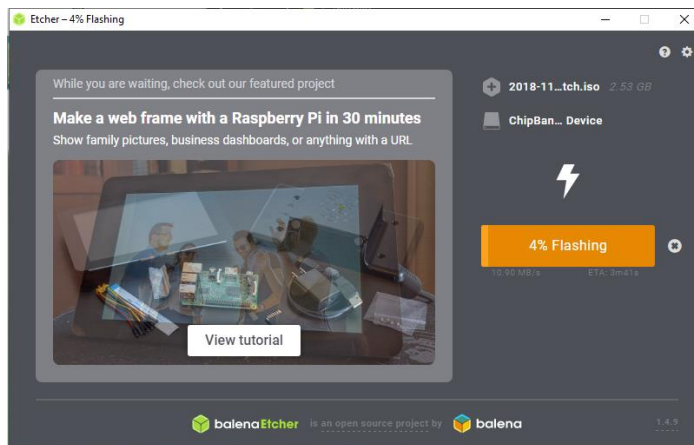
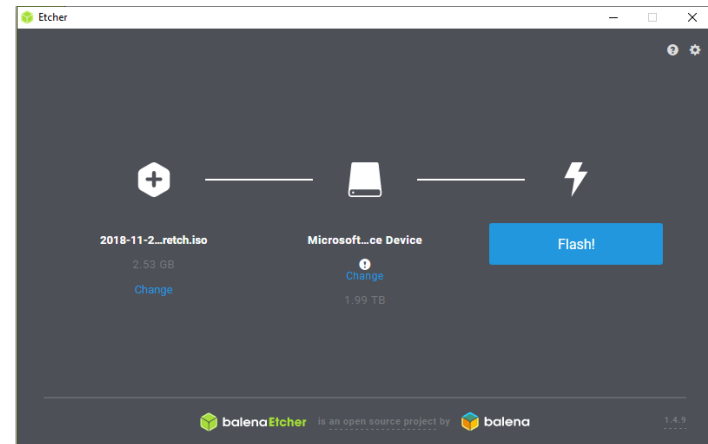
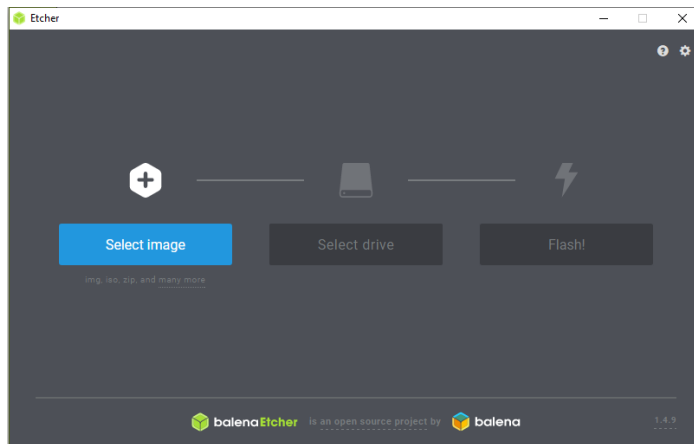
- ◉ <https://www.raspberrypi.org/downloads/>
- ◉ Varios sabores
  - ◉ **Noobs**
    - ◉ Instalación sencilla con opciones de descargar diferentes SSOO
    - ◉ Contiene Raspbian ya descargado
  - ◉ **Noobs lite**
    - ◉ Igual que Noobs, pero sin raspbian predescargado
  - ◉ **Raspbian** (basado en Debian Strechts)
    - ◉ Contiene ya aplicaciones de uso normal preinstaladas
      - ◉ Python, Java, Scratch, algún videojuego, ...
    - ◉ 3 tipos, con más o menos software preinstalado
      - ◉ Raspbian Stretch with desktop and recommended software
        - ◉ Ideal si quieres un miniordenador
      - ◉ **Raspbian Stretch with desktop**
        - ◉ Ideal para un servidor si vas a usar la parte gráfica
      - ◉ **Raspbian Stretch Lite**
        - ◉ Ideal si quieres un servidor mínimo y no necesitas nada de lo otro
        - ◉ Se puede instalar después la interfaz gráfica
  - ◉ **Rasberry Pi Desktop**
    - ◉ Para MVs o el PC de sobremesa
  - ◉ **Otros**
    - ◉ Windows, Ubuntu, etc.



# Instalar SO en Raspberry Pi

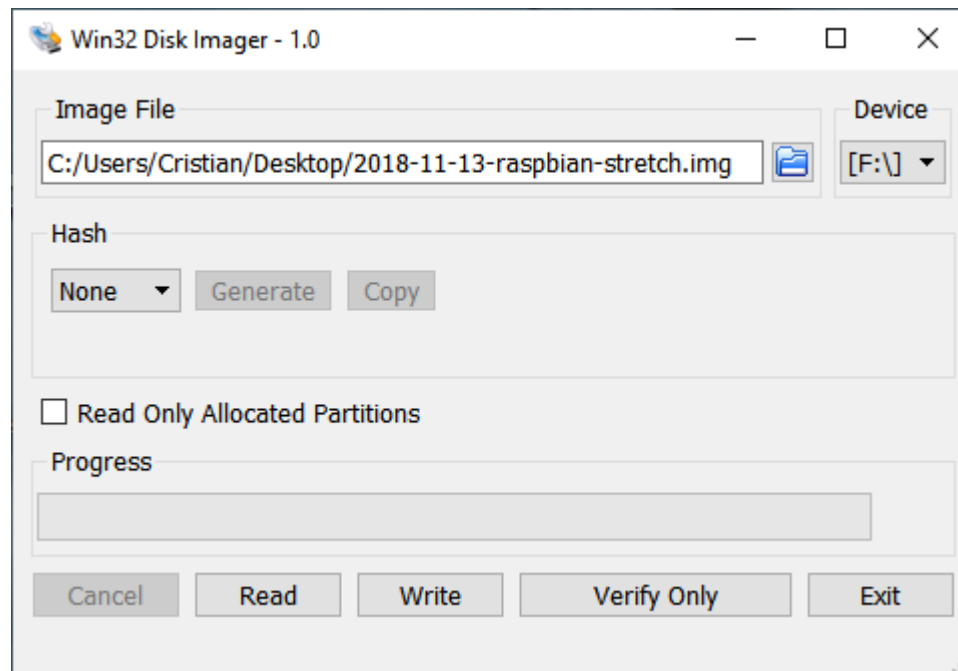
- <https://www.raspberrypi.org/documentation/installation/installing-images/README.md>
- Descargar Win32 Disk o Etcher
  - Etcher (a veces da problemas)
    - <https://sourceforge.net/projects/win32diskimager/> o <https://etcher.io/>
    - Herramienta multiplataforma para grabar en tarjetas SD
      - Soporta imágenes (ISO) (des)comprimidas
      - Windows, Mac, Linux
  - Win32 Disk
    - Solo Windows
    - Permite hacer backups de la tarjeta SD
- Seleccionar ISO -> Tarjeta SD -> Flash!
  - **Cuidado de no seleccionar un HD del SO**

# Instalar So en Raspberry Pi – Etcher



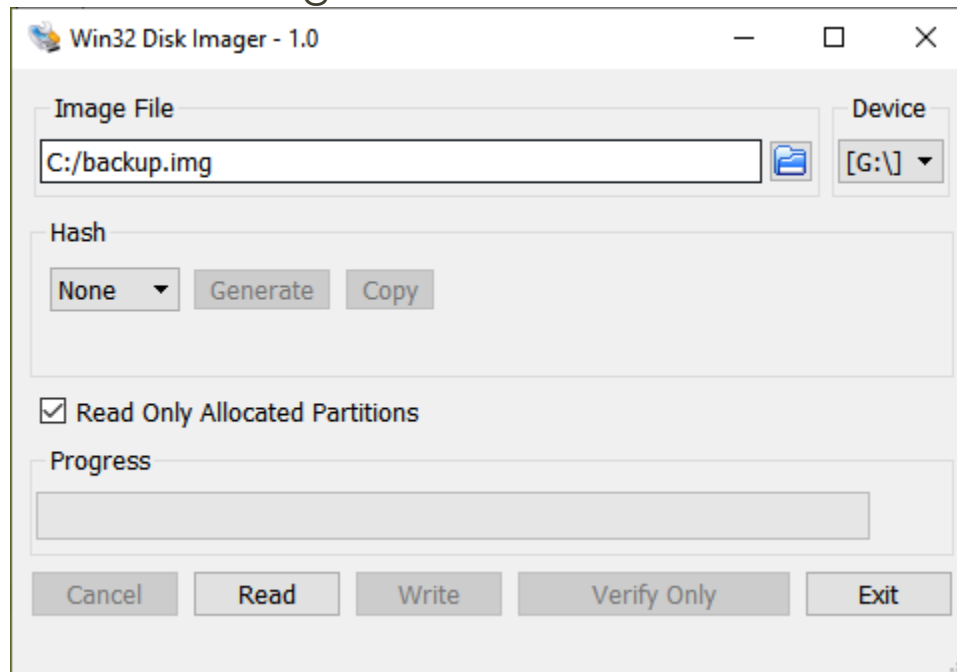
# Instalar So en Raspberry Pi – Win32 Disk I

- Grabar imagen: **Write**
- **Cuidado** si pregunta dp si se quiere formatear, puede ser una pregunta sobre un disco de Windows



# Instalar So en Raspberry Pi – Win32 Disk II

- Copia de seguridad: **Backup**
  - Seleccionar «Read Only Allocated Partitions»
    - Lee solo lo que hay escrito en disco, no lo vacío
      - La imagen así pesara lo que ocupe, 2GB, 4 GB, y no los 16 o 32 de la SD
  - Poner nombre a la imagen



# Preconfiguraciones

# Config.txt I

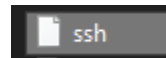
- ◉ **Acceder a la SD** dónde se ha instalado Raspbian
- ◉ Abrir el **fichero /boot/config.txt**
  - ◉ <https://www.raspberrypi.org/documentation/configuration/config-txt/>
- ◉ **Modificar**
  - ◉ Descomentar: **hdmi\_force\_hotplug=1**
    - ◉ Fuerza a la GPU a usar el HDMI
      - ◉ A veces no funciona el HDMI en una instalación limpia y hay que hacer esto
      - ◉ Hacerlo solo si no funciona el HDMI
  - ◉ Mirar las tablas de las variables:  
<https://www.raspberrypi.org/documentation/configuration/config-txt/video.md>
    - ◉ Tipo de salida de video
      - ◉ **hdmi\_group=2**
    - ◉ Resolución en base al tipo de salida (Se mira en función del `hdmi_group`)
      - ◉ **hdmi\_mode=16**



## Config.txt II

- ◉ **Activar SSH**

- ◉ Hacerlo si solo se va a poder acceder por SSH
- ◉ Crear un fichero sin extensión y vacío llamado «ssh»



- ◉ Solo si se tiene que acceder sí o sí
- ◉ Para saber que IP le ha dado a la RPi cuando se encendió hay que escanear los dispositivos de red
  - ◉ Fing: <https://www.redeszone.net/2015/07/07/comprueba-quien-esta-conectado-a-tu-router-desde-un-pc-con-fing/>
- ◉ Al entrar a la RPi dirá que se ha activado y habrá que confirmarlo

# Port Forwarding – Acceso desde el exterior

- Hay que **acceder al router** y redirigir un puerto externo del router a un puerto de la IP que usa la Raspberry Pi
- Accedemos a la consola de administración del router
  - Normalmente, <http://192.168.1.1/>
  - Movistar funciona algo diferente y tiene su propia web
  - Telecable: <http://192.168.0.1/>
- Buscar contraseña de acceso**
  - Por defecto: suelen ser campos vacíos, admin/admin, buscar en la guía
- Ir a la opción «**Port Forwarding**»
  - Poner puerto exterior por el que se entrará
  - Seleccionar la IP dinámica
  - El puerto de entrada de esta

External			Internal				
Start Port	End Port		IP Address	Start Port	End Port	Protocol	Enable
3333	to 3333		192.168.0.13	5900	to 5900	TCP ▼	<input checked="" type="checkbox"/>
0	to 0		0.0.0.0	0	to 0	TCP ▼	<input type="checkbox"/>



# IP

- ◉ **Dinámica**

- ◉ Por DHCP
- ◉ Cada vez que reiniciemos/perdamos la conexión la IP puede cambiar

- ◉ **Estática**

- ◉ Siempre tendrá la misma IP

- ◉ **Router**

- ◉ Da una IP utilizando DHCP
- ◉ Esta IP se puede pedir siempre desde el mismo dispositivo

- ◉ **Interfaces**

- ◉ eth0: ethernet
- ◉ wlan0: Wi-Fi

# Configuración de la IP en la RPI

- Modificamos el archivo **/etc/dhcpd.conf**

```
:~ $ sudo nano /etc/dhcpd.conf
```

- Copiamos las siguientes líneas que están comentadas

```
# Example static IP configuration:
#interface eth0
#static ip_address=192.168.0.10/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
#static routers=192.168.0.1
#static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1
```

- Modificamos solo la IP y descomentamos el resto

```
# Example static IP configuration:
interface eth0
static ip_address=192.168.0.13/24
static routers=192.168.0.1
static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1
```

- Reiniciamos la Rpi
- Ahora siempre tendremos la misma «IP interna»

Raspberry Pi

# Usuario y nombre de la RPi

- Nombre de usuario @ nombre de la RPi

```
pi@raspberrypi:~ $
```

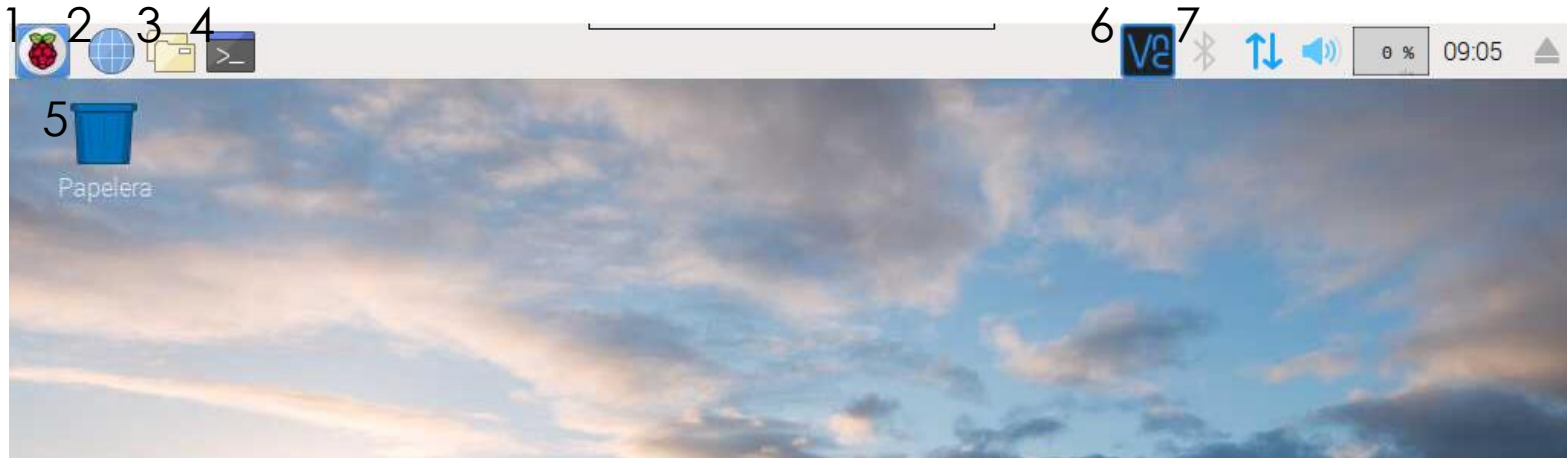
- Después de cambiar el nombre a la RPi
  - Importante**, pues si hay varias RPi en la misma red con mismos nombres, o no tienen, darán problemas de conexión

```
pi@RaspberryCris:~ $
```

- Por motivos de **seguridad**, siempre hay que **cambiar el nombre de usuario**
  - En la Raspberry PI el usuario por defecto es «pi»
  - Se puede cambiar, pero es muy problemático
    - Viene muy «hardcodeado»
    - Hay bastantes cosas que dejan de funcionar

# Escritorio

1. Menú inicio
2. Navegador web (Chromium)
3. Gestor de archivos
4. Terminal
5. Papelera
6. VNC (solo si está activado)
7. Bluetooth | Red | Volumen | Monitor de uso | Reloj | Extraer



# Configuración I

- Abrir la terminal
- Ejecutar como administrador la configuración
  - Modifica el archivo **/boot/config.txt** y algunos de Linux

```
~ $ sudo raspi-config
```



```
Raspberry Pi 2 Model B Rev 1.1

Raspberry Pi Software Configuration Tool (raspi-config)

1 Change User Password      Change password for the current u
2 Network Options           Configure network settings
3 Boot Options              Configure options for start-up
4 Localisation Options      Set up language and regional sett
5 Interfacing Options       Configure connections to peripher
6 Overclock                 Configure overclocking for your P
7 Advanced Options          Configure advanced settings
8 Update                    Update this tool to the latest ve
9 About raspi-config         Information about this configurat

<Select>                   <Finish>
```

# Configuración II

- Cambiar contraseña
- Opciones de red
  - **Nombre de esa Rpi**
    - Muy importante, pues si no tiene o se llama igual que otra, tendremos problemas en las descargas  
`sudo: unable to resolve host raspberrypi`
  - **Conectarse a una Wi-Fi**
  - Predictable Network Interface Names
    - Servicio que asigna un nombre a una red
    - Nombrado de interfaces ethX (eth0, eth1, ...) para evitar problemas en reglas (seguridad, configuraciones, etc.)
- **Boot options**
  - **Inicio en escritorio o consola**
  - **Esperar a iniciar a que tenga red**
  - Mostrar la pantalla «\$plash»
- **Localisation**
  - Idioma, región, teclado y canales de la Wi-Fi
    - Posiblemente ya configurado previamente
- **Interfaces de conexión** (Activar/Desactivar)
  - Cámara, **SSH**, **VNC**, SPI, I2C, Serial, 1-Wire, Remote GPIO
- Overclock
  - Puede reducir la vida útil de la RPi

# Configuración III

- **Configuración avanzada**

- **Expandir sistema de ficheros**

- Al iniciar la RPi el sistema de ficheros no ocupa toda la SD, solo 2GB. Si queremos utilizarla toda, hay que usar esta opción
    - Si cambiamos el sistema a una tarjeta SD más grande, hay que hacer esto también

- OverScan: para TV viejas cuando aparecen bordes sin utilizar)

- Memory Split: permite cambiar la memoria asignada a la GPU

- Audio: cambia entre Audio por HDMI o Jack 3.5mm

- Resolución: resolución HDMI/DVI para cuando se inicie la RPi sin conector a TV y que puede afectar cuando se utiliza VNC y

- **Pixel doubling** (Soporte de pantallas con alta resolución (Retina, HDMI))

- Cada pixel ocupará exactamente el cuádruple, el doble en cada dirección (1p -> 2\*2p)

- Usar solo en esas pantallas, sino, no se verá todo el contenido en pantalla

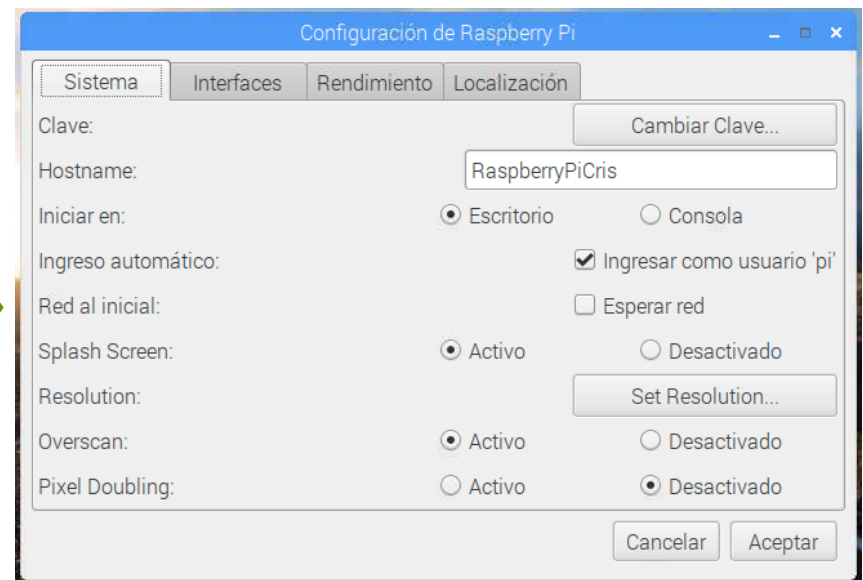
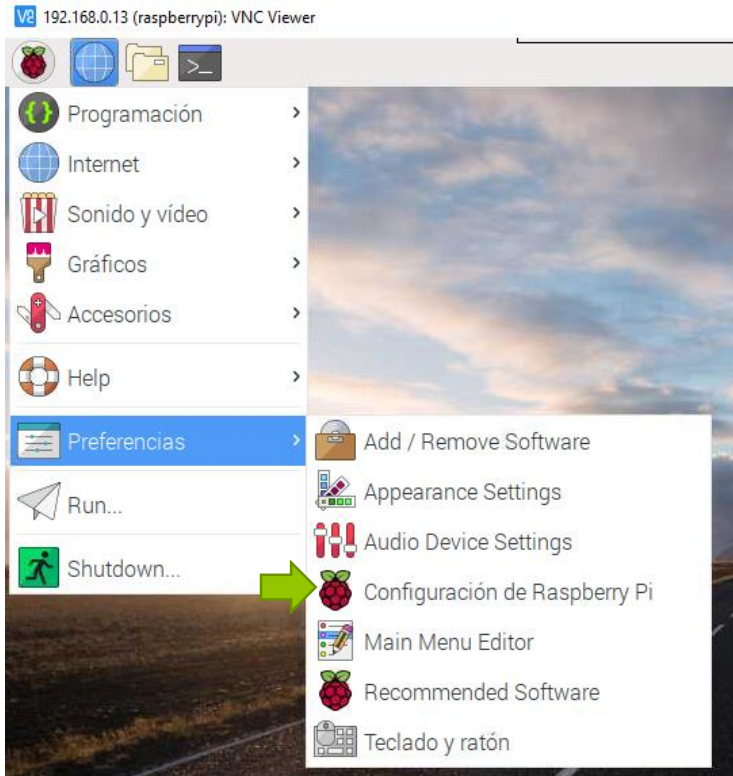
- GL Driver: driver experimental para los gráficos del escritorio

- **Update**

- Actualiza esta herramienta



# Configuración IV – Interfaz gráfica



# Apt, apt-get y caché

- apt
  - Comando inicial de Debian
  - Muestra más información
  - El recomendado
- apt-get
  - Más amistoso que apt
  - Una capa por encima de apt
  - División de apt en diferentes comandos más específicos
    - apt-get, apt-cache
- Caché del repositorio
  - Guarda paquetes para su posible reinstalación
  - Si la limpiamos, tendríamos que bajar de nuevo estos paquetes

# Comandos útiles I – Actualizar

- **Actualizar lista de paquetes** con las versiones actuales de ellos

```
:~ $ sudo apt-get update
```

- **Actualizar los paquetes instalados**, solo sus versiones

```
:~ $ sudo apt-get upgrade
```

- **Actualizar los paquetes instalados**

- Puede instalar/eliminar paquetes (in)necesarios (dependencias)

```
:~ $ sudo apt-get dist-upgrade
```

- **Actualiza el firmware/kernel** de la RPi a la última versión

- Recomendado solo para su uso en Raspbian

- **Solo usar en casos muy concretos** (el 1% de las veces): firmware, bugs, ...

- apt-get suele ser suficiente: seguridad, bugs, mejoras, etc.

- Puede estropearos el SO, paquetes instalados, incompatibilidades, etc.

- **Hacer copia de seguridad de la tarjeta SD**

- Es necesario reiniciar después

```
:/ $ sudo rpi-update
```

- Si no encuentra el comando...

```
:/ $ sudo apt-get install rpi-update
```

## Comandos útiles II – Limpiar

- Limpiar caché del repositorio de paquetes ()

- Limpia el repositorio local de los archivos

```
:~ $ sudo apt-get clean
```

- Clean + limpia paquetes que ya no se pueden descargar, o que tienen nuevas versiones, luego ya no son inútiles

```
:~ $ sudo apt-get autoclean
```

- Elimina paquetes que fueron instalados por necesidad de dependencias y que ya no son necesarios

```
:~ $ sudo apt-get autoremove
```

- «Inconveniente» de uso

- Para reinstalar un paquete tendremos que volver a descargarlo

# Máquina virtual – Guest Additions

- rpi-update puede hacer que se deje de ver el ratón, entre otros problemas, y tengáis que reinstalar las «Guest Additions»
  - Máquina Virtual -> Dispositivos -> Insertar imagen de CD de las «Guest Additions»...
  - Ir al CD

```
:/ $ cd /media/cdrom0/
```

- Ejecutarlas

```
:/media/cdrom0 $ sudo sh ./VBoxLinuxAdditions.run
```

Montar servidor

# HTTP Apache – ¿Qué es?



- ◉ Servidor de código abierto y multiplataforma
  - ◉ Servidor web, balanceador de carga y servidor de paso (proxy)
- ◉ Empezó en 1995
- ◉ Modular: gran cantidad de diferentes módulos
  - ◉ Seguridad, reglas, protocolos, lenguajes, etc.
- ◉ Popular
  - ◉ Mucha documentación y guías
  - ◉ De los más utilizados globalmente
- ◉ Lenguajes
  - ◉ PHP, Python, Perl, Ruby, .Net
- ◉ Mantenido por la fundación Apache Software Foundation
- ◉ Otros servidores web: nGinx, IIS, Cherokee, TomCat, etc.

# Apache – Instalación



- Actualizar paquetes si no lo estuvieran ya

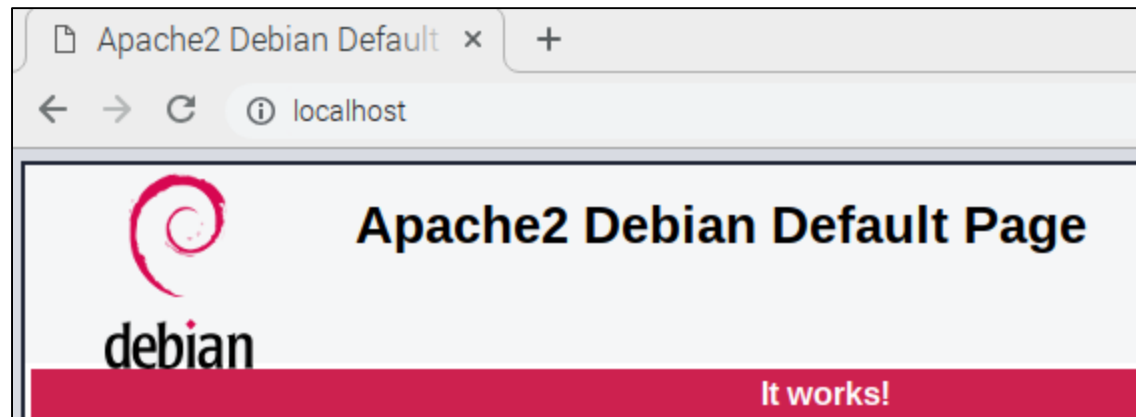
```
:/ $ sudo apt update
```

```
:/ $ sudo apt upgrade
```

- Instalar Apache HTTPD versión 2

```
:/ $ sudo apt install apache2
```


- Comprobar que todo funciona en local
  - Abrir Chromium
  - Ir a: localhost





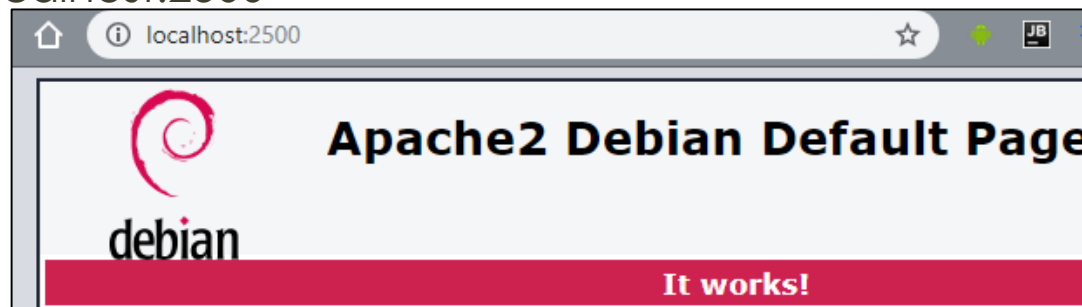
# Apache – Acceso por MV



- Acceder desde el PC anfitrión
  - Máquina Virtual -> Preferencias -> Preferencias de Red
  - Red -> NAT -> Avanzadas -> Reenvío de puertos
  - Agregar una nueva regla de reenvío 

Nombre	Protocolo	IP anfitrión	Puerto anfitrión	IP invitado	Puerto invitado
Rule 1	TCP		2500		80

- Aceptar
- Ir a un navegador web del anfitrión
- Ir a: localhost:2500



# Apache – Directorio web



- Se encuentra en /var/www/html

```
:/ $ cd /var/www/
```

- Por defecto, pertenece a root

```
:/var/www $ ls -l
```



```
drwxr-xr-x 2 root root 4096 ene 23 11:23 html
```

- Creamos un grupo de trabajo

```
:/var/www $ sudo groupadd server
```

- Añadimos al usuario «pi» al grupo

```
:/var/www $ sudo adduser pi server
```

- Cambiamos el propietario y grupo de la carpeta «html» para facilitar todo y poder editarlo con el usuario «pi»

```
:/var/www $ sudo chown -R pi:server html
```

```
:/var/www $ ls -l
```



```
drwxr-xr-x 2 pi server 4096 ene 23 11:23 html
```

# Apache – Prueba



- Entramos en la carpeta «html»

```
:/var/www $ cd html
```

- Creamos un fichero HTML nuevo en la carpeta «html»

```
:/var/www/html $ pico apacheTest.html
```

- Ejemplo

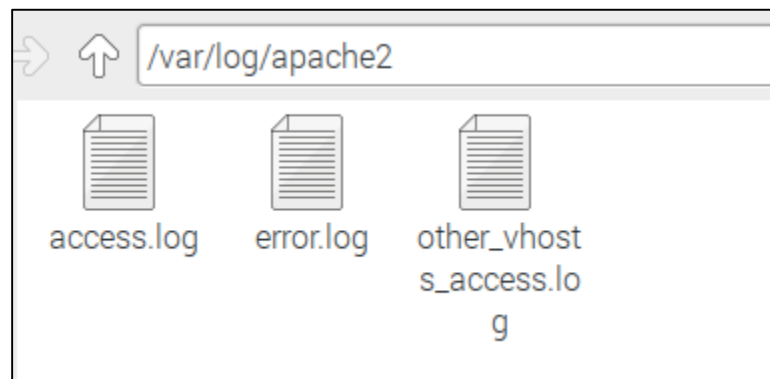
- <https://gist.github.com/gonzalezgarciacristian/0c260479fb49bc276ddef65452cda515>



# Apache – Logs



- Directorio de logs
  - Acceso
    - Día, hora y fecha de acceso, navegador, página de acceso, IP, tipo de petición (GET, POST, ...)
  - Errores
    - Errores de Apache y de PHP
- Virtual hosts
  - Permite alojar múltiples dominios en un solo IP



# Apache – Configuración I



- **Configuraciones en httpd.conf**

- <https://httpd.apache.org/docs/2.4/mod/core.html#servertokens>

```
:/ $ sudo nano /etc/apache2/apache2.conf
```

- Para aplicar los cambios, hay **que reiniciar Apache**

```
:/ $ /etc/init.d/apache2 restart
```

- **Directorios con permisos correctos**

- Si los usuarios pueden subir archivos, configurarlo adecuadamente
  - Alias «/uploadImages/» redirige a esa carpeta

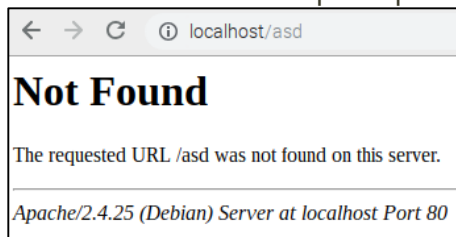
```
Alias /uploadImages/ "C:/Servidor/webs/uploadImages/"  
<Directory "C:/Servidor/webs/uploadImages/">  
    Order allow,deny  
    Allow from all  
</Directory>
```

# Apache – Configuración II

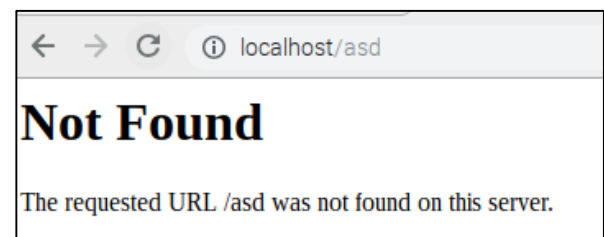


## ○ Ocultar versión y sistema

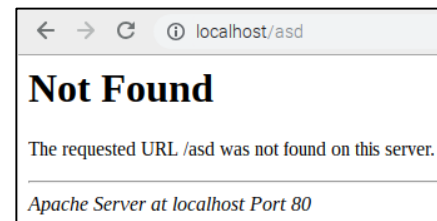
- /etc/apache2/conf-available/security.conf
- Ocultamos información relevante (servidor, versión, SO, ...)
- Puede ayudar a que encuentren vulnerabilidades en nuestro sistema
  - ServerSignature: añade el pie con la información del ServerTokens
  - ServerTokens: mostrar toda o parte de la información [Full]
  - Prod | Full | Major | Minor | OS



ServerSignature Off  
ServerTokens Prod



ServerSignature On  
ServerTokens Prod

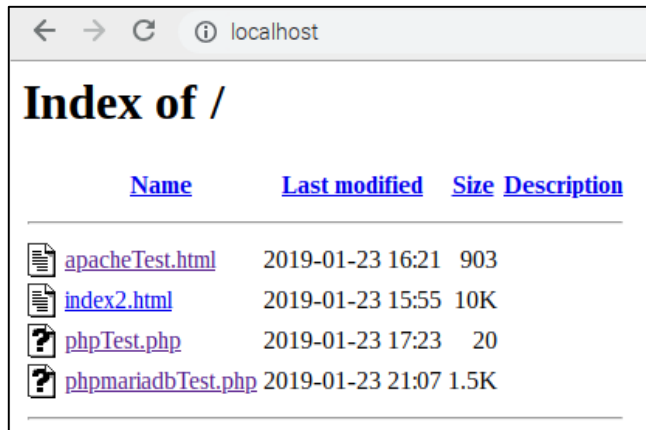






# Apache – Configuración III




- **No mostrar el árbol del directorio**

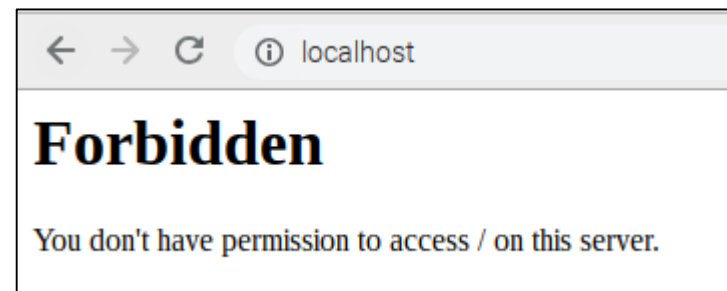
- Por defecto, si no hay index.html en ese directorio, muestra el árbol



	<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
	<a href="#">apacheTest.html</a>	2019-01-23 16:21	903	
	<a href="#">index2.html</a>	2019-01-23 15:55	10K	
	<a href="#">phpTest.php</a>	2019-01-23 17:23	20	
	<a href="#">phpmysqliTest.php</a>	2019-01-23 21:07	1.5K	



```
<Directory /var/www/html>  
Options -Indexes  
</Directory>
```



# Apache – Configuración IV



- **Mantener actualizado** Apache

```
:/ $ apache2 -v
```



```
Server version: Apache/2.4.25 (Debian)  
Server built: 2018-11-03T18:46:19
```

- **Activar/Deshabilitar módulos** (in) necesarios por seguridad y rendimiento

- Activar

```
:/ $ sudo a2enmod rewrite
```

- Desactivar

```
:/ $ sudo a2dismod rewrite
```

- Reiniciar Apache para que los cambios surtan efecto

```
:/ $ systemctl restart apache2
```

- Listado de los módulos: <https://httpd.apache.org/docs/2.4/es/mod/>

- Listar módulos activados 

```
:/ $ apache2ctl -M
```

- Static: compilado para Apache, siendo más rápido
  - Shared: dinámico
  - Los nombres no coinciden: hay que quitar «\_module» y añadir «mod\_»
  - Muchos se instalan usando apt-get



# Apache – Configuración V



- ◉ **Denegar/Permitir acceso a directorios**

- ◉ Directiva <directory /carpeta>
  - ◉ <https://httpd.apache.org/docs/2.4/howto/access.html>
- ◉ Futuros -> Directivas Require
  - ◉ [https://httpd.apache.org/docs/2.4/mod/mod\\_authz\\_core.html#require](https://httpd.apache.org/docs/2.4/mod/mod_authz_core.html#require)

- ◉ **Mod\_Security**

- ◉ Firewall, monitor de tráfico web, análisis, filtrado de peticiones, logs, bloqueo de IPs, previene ataques de fuerza bruta, etc.

- ◉ **Mod\_Evasive**

- ◉ Previene ataques DDoS y de fuerza bruta analizando todas las peticiones

- ◉ **Limitar tamaño de las peticiones**

- ◉ Evitar que puedan subir archivos muy grandes a un directorio
  - ◉ Por defecto en 0 (sin límite); 500k = 512000
- ◉ LimitRequestBody

```
<Directory /tmp/images>  
    LimitRequestBody 512000  
</Directory>
```



# PHP –

## ¿Qué es?

- PHP: Hypertext Preprocessor
- Lenguaje de propósito general (GPL)
- Código abierto
- Diseñado especialmente para propósitos web
- Apareció en 1995
- Actualmente por su 7 versión (la 6 no existió, fue la 5.3)
- Multiplataforma
- Permite su incorporación en el documento HTML
  - ¡Cuidado con la calidad del código!
- Lo desarrolla el PHP Group
  - La versión inicial fue de Rasmus Lerdorf
- Muchas extensiones
  - Apache, PDF, imágenes, BBDD SQL y NoSQL,



# PHP – Instalación I

- En apt-get está disponible PHP7.0 , pero... **expiró** el 3/12/2018
  - En APT no se encuentra

```
:/ $ sudo apt-get install php7.0
```

- Instalaremos PHP7.3 (Última versión del 6/12/2018)
- Usaremos APT
  - Bajamos los certificados de Mozilla para usar HTTPS
  - Indicamos que usaremos HTTPS en APT para descargar paquetes

```
:/ $ sudo apt install ca-certificates apt-transport-https
```

- Descargamos la clave de acceso al repositorio y la añadimos
  - Usaremos el repositorio <https://packages.sury.org/php/>
    - Colaborador en Debian desde el 2000

```
~ $ wget -q https://packages.sury.org/php/apt.gpg -O- | sudo apt-key add -
```

Letra O mayúscula



# PHP – Instalación II

- Imprimimos por pantalla y grabamos la lista de paquetes
  - Especificando la versión: «stretch»
  - tee: nos permite hacerlo todo en una línea usando tuberías

```
pi@RaspberryCris:/ $ echo "deb https://packages.sury.org/php/ stretch main" | sudo tee /etc/ap
pt/sources.list.d/php.list
```

- Actualizar paquetes si no lo estuvieran ya

```
:/ $ sudo apt update
```

```
:/ $ sudo apt upgrade
```

- Instalamos PHP7.3

```
:/ $ sudo apt install php7.3
```

- Si requiriéramos módulos instalaríamos php7.3<nombreMódulo>

```
pi@RaspberryCris:/ $ sudo apt install php7.3-
php7.3-bcmath          php7.3-imap            php7.3-pspell-dbgsym
php7.3-bcmath-dbgsym   php7.3-imap-dbgsym     php7.3-readline
php7.3-bz2             php7.3-interbase       php7.3-readline-dbgsym
php7.3-bz2-dbgsym      php7.3-interbase-dbgsym php7.3-recode
php7.3-cgi             php7.3-intl            php7.3-recode-dbgsym
php7.3-cgi-dbgsym      php7.3-intl-dbgsym     php7.3-snmp
```



# PHP – Instalación III

- Comprobamos que se ha instalado

```
:/ $ php -v
```



```
PHP 7.3.1-1+0~20190113101756.25+stretch~1.gbp15aaa9 (cli) (built: Jan 13 2019 10:17:57) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.3.1, Copyright (c) 1998-2018 Zend Technologies
    with Zend OPcache v7.3.1-1+0~20190113101756.25+stretch~1.gbp15aaa9, Copyright (c) 1999-2018, by Zend Technologies
```

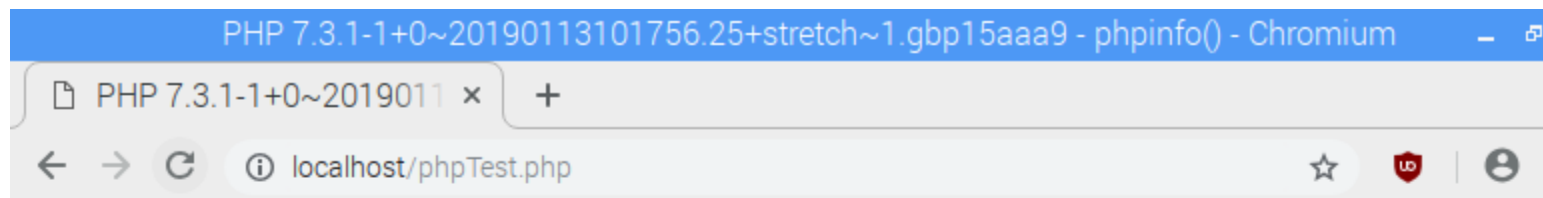
- Creamos una página para testear PHP

```
:/var/www/html $ pico phpTest.php
```



```
GNU nano 2.7.4 Fichero: phpTest.php
?php phpinfo(); ?>
```

# PHP – Prueba



## PHP Version 7.3.1-1+0~20190113101756.25+stretch~1.gbp15aaa9

System	Linux RaspberryCris 4.9.0-8-686 #1 SMP Debian 4.9.130-2 (2018-10-27) i686
Build Date	Jan 13 2019 10:17:57
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/apache2
Loaded Configuration File	/etc/php/7.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/apache2/conf.d
Additional .ini files parsed	/etc/php/7.3/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.3/apache2/conf.d/10-opc /etc/php/7.3/apache2/conf.d/10-pdo.ini, /etc/php/7.3/apache2/conf.d/20-calendar.in /etc/php/7.3/apache2/conf.d/20-ctype.ini, /etc/php/7.3/apache2/conf.d/20-exif.ini, /et fileinfo.ini, /etc/php/7.3/apache2/conf.d/20-ftp.ini, /etc/php/7.3/apache2/conf.d/20-g /etc/php/7.3/apache2/conf.d/20-iconv.ini, /etc/php/7.3/apache2/conf.d/20-json.ini, /e mysqli.ini, /etc/php/7.3/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.3/apache2/conf /etc/php/7.3/apache2/conf.d/20-posix.ini, /etc/php/7.3/apache2/conf.d/20-readline.i /etc/php/7.3/apache2/conf.d/20-shmop.ini, /etc/php/7.3/apache2/conf.d/20-sockets



# PHP – Configuración I

- Directorio de php.ini
  - Fichero de configuración de PHP

```
:~ $ cd /etc/php/7.3/apache2
```

- Hacemos copia de seguridad

```
:/etc/php/7.3/apache2 $ sudo cp php.ini php-copy.ini
```

- Lo editamos

```
:/etc/php/7.3/apache2 $ sudo nano php.ini
```

- Buscamos (CTRL+W) «timezone»
  - Definimos la zona horaria que utilizaremos

```
; Defines the default timezone used by the date functions  
; http://php.net/date.timezone  
date.timezone = Europe/Madrid
```



## PHP – Configuración II

- Buscamos (CTRL+W) «**session.save\_path**»
  - Nunca usar un directorio «/tmp» ya que puede ser accedido por otros usuarios, permitiendo así secuestros de sesión

```
; The path can be defined as:  
;  
session.save_path = "/var/lib/php/session/"
```

- Creamos la carpeta para la sesión

```
:~ $ sudo mkdir /var/lib/php/session/
```

- Cambiamos permisos para que solo pueda el propietario
  - Solo en esa carpeta, no en los padres

```
:~ $ sudo chmod 700 /var/lib/php/session/
```

- Cambiamos propietario y grupo a los correspondientes
  - Solo en esa carpeta, no en los padres

```
:~ $ sudo chown -R pi:server /var/lib/php/session/
```





# PHP – Configuración III

- Reiniciar configuración reiniciando Apache

```
:/ $ /etc/init.d/apache2 restart
```

- Usar solo los módulos necesarios por seguridad y rendimiento
  - Borrarlos o hacer instalación reducida, depende del módulo
  - Ver módulos activos

```
:/ $ php -m
```

- Cuanta menos información demos, mejor**
  - No usar la función `phpinfo()`;
  - `expose_php = Off` ([`php.ini` por defecto en `Off`])
    - En la petición dice que se usa PHP: X-Powered-By: PHP/7.3
  - `display_errors = Off` (en producción) = `On` (en desarrollo)
    - Muestra una traza del error entero en la web cuando ocurre
- Deshabilitar subida de ficheros** si no lo usamos (`file_uploads [On]`)
  - Limitarla a lo necesario si la usamos (1MB) (`upload_max_filesize`)



# PHP – Configuración IV

- ◉ **Ejecución remota de código**

- ◉ Permite traer archivos por HTTP o FTP
  - ◉ Si no se necesita: `allow_url_fopen = Off` [On] y `allow_url_include = Off` [Off]

- ◉ **Limitar capacidad de aceptación de datos de la petición POST**

- ◉ `post_max_size = 1M` [8M]

- ◉ **Limitar peticiones** (Limitar DDOS)

- ◉ `max_execution_time = 30` [30] // Tiempo en s de ejecución de un Script
- ◉ `max_input_time = 30` [60] // Tiempo máx. de parseo del Script
- ◉ `memory_limit = 30M` [120M] // Máx. memoria consumido por un Script

- ◉ **Deshabilitar funciones peligrosas** que no usemos

- ◉ `disable_functions`
  - ◉ `exec()`: ejecuta un programa externo
  - ◉ <https://stackoverflow.com/questions/3115559/exploitable-php-functions>



## PHP – Configuración V

- ◉ **Limitar acceso de PHP al sistema de ficheros**
  - ◉ `open_basedir = "directorio1:directorio2:..."`
    - ◉ Toda carpeta que no esté incluida será rechazada por PHP para ser abierta
- ◉ Especificar **carpeta de archivos temporales** que suba el usuario con los permisos adecuados, si es que pueden subir archivos
  - ◉ `upload_tmp_dir = ""`
- ◉ **Mantener PHP actualizado por seguridad y rendimiento**

## MariaDB – ¿Qué es?



- Base de datos relacional
- Fork de MySQL
- Lo mantienen los desarrolladores originales de MySQL hasta la compra Sun Microsystems por Oracle
  - Miedo a empeorar o no seguir mejorando MySQL para favorecer a Oracle
- Usa los mismos conectores, instrucciones y APIs que MySQL
- Tiene nuevos motores
- Lanzamiento inicial en enero de 2009
- Multiplataforma
- Código abierto
- Otras BBDD: MySQL, PostgreSQL, SQLite, ...

# MariaDB – Instalación I



- La página oficial da todas las instrucciones para varios SSOO
  - <https://downloads.mariadb.org/mariadb/repositories/#mirror=tedec>
- Para Raspberry Pi solo hace falta
  - Actualizamos la lista de paquetes al añadir un nuevo repositorio

```
:~ $ sudo apt-get update
```

- Instalamos mariadb-server

```
:/ $ sudo apt-get install mariadb-server
```

- Introducid una contraseña segura
  - Alfanumérica + caracteres especiales
- Comprobamos la versión

```
:/ $ mysql -V
```



```
mysql Ver 15.1 Distrib 10.1.37-MariaDB, for debian-linux-gnueabi (armv8l) using readline 5.2
```



# Cambiar contraseña del usuario root

- Si no pidió contraseña, esta será vacía
  - Muy peligroso
  - Puede ser por la versión o el SSO el que no la pida
- Cambiar contraseña del root
  - Entramos

```
:~ $ sudo mysql -u root -p
```

- Cambiamos a la tabs «mysql»

```
MariaDB [(none)]> USE mysql;
```

- Hacemos una consulta para cambiar la contraseña y ponemos una de verdad segura (**NOTA: respetar las mayúsculas**)

```
MariaDB [mysql]> UPDATE user SET password=PASSWORD('superpass') WHERE User='root'  
' AND Host = 'localhost';
```

- Recargamos privilegios

```
MariaDB [mysql]> FLUSH PRIVILEGES;
```

- Salimos

```
MariaDB [mysql]> exit
```



# MariaDB – Prueba de acceso

- Accedemos a MariaDB
  - Usar la contraseña escrita antes. Si no pidió, es vacía

```
:/ $ sudo mysql -u root -p
```

```
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 5
Server version: 10.1.37-MariaDB-0+deb9u1 Raspbian 9.0

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
3 rows in set (0.00 sec)

MariaDB [(none)]>
```

- Salir

```
MariaDB [(none)]> exit
```



# Conectar MariaDB/MySQL y PHP

- Necesitamos el paquete php-commons
  - Contiene varios módulos
    - Zip, JSON, curl, etc.

```
:/ $ sudo apt-get install php7.3-common
```

- Instalamos el conector de la BDD con PHP
  - Sirve para MariaDb y MySQL

```
:/ $ sudo apt-get install php7.3-mysql
```





# Preparación del ejemplo I

- Entrar a MariaDB

```
:/ $ sudo mysql -u root -p
```

- Crear base de datos «rpi»

```
MariaDB [(none)]> create database rpi;  
Query OK, 1 row affected (0.000 sec)
```

- Usar la nueva BDD «rpi»

```
MariaDB [(none)]> use rpi;
```

- Crear tabla «products» con motor Aria (=MyISAM) y UTF8

- Columnas

- id: bigint, unsigned, primary key, auto\_increment
- name: varchar 128, utf8
- quantity: int

```
MariaDB [rpi]> create table products (id bigint unsigned auto_increment primary key, name var  
char(128) charset utf8, quantity int) engine=aria charset utf8;  
Query OK, 0 rows affected (0.004 sec)
```



## Preparación del ejemplo II

- Introducir productos

```
MariaDB [rpi]> INSERT INTO products (name, quantity) VALUES ('Assassins Creed Origins', 1);
Query OK, 1 row affected (0.001 sec)

MariaDB [rpi]> INSERT INTO products (name, quantity) VALUES ('Civilization', 1);
Query OK, 1 row affected (0.001 sec)

MariaDB [rpi]> INSERT INTO products (name, quantity) VALUES ('16gb ram', 4);
Query OK, 1 row affected (0.001 sec)

MariaDB [rpi]> INSERT INTO products (name, quantity) VALUES ('2 HDD 2TB', 2);
Query OK, 1 row affected (0.001 sec)

MariaDB [rpi]> INSERT INTO products (name, quantity) VALUES ('Diablo', 1);
Query OK, 1 row affected (0.001 sec)

MariaDB [rpi]> INSERT INTO products (name, quantity) VALUES ('Nvidia GeForce 2080', 2);
Query OK, 1 row affected (0.001 sec)
```

- Comprobar

```
MariaDB [rpi]> SELECT * FROM products;
+----+-----+-----+
| id | name                | quantity |
+----+-----+-----+
| 1  | Assassins Creed Origins | 1        |
| 2  | Civilization          | 1        |
| 3  | 16gb ram              | 4        |
| 4  | 2 HDD 2TB             | 2        |
| 5  | Diablo                 | 1        |
| 6  | Nvidia GeForce 2080    | 2        |
+----+-----+-----+
6 rows in set (0.000 sec)
```



## Usuarios I

- Crear cuenta para el usuario «pi»

```
MariaDB [(none)]> CREATE USER 'pi'@'localhost' IDENTIFIED BY 'superpass';
Query OK, 0 rows affected (0.000 sec)
```

- Comprobar su creación con «SHOW GRANTS FOR ...»
  - Por defecto, las crea **sin privilegios** (USAGE): «GRANT USAGE ...»
  - Solo pueden usar la BDD

```
MariaDB [(none)]> SHOW GRANTS FOR 'pi'@'localhost';
+-----+
| Grants for pi@localhost |
+-----+
| GRANT USAGE ON *.* TO 'pi'@'localhost' IDENTIFIED BY PASSWORD '*72EA7A3B37C7CF2067FD7ACD7FF596E05B9A9242' |
+-----+
```

- Para borrar una cuenta...

```
MariaDB [(none)]> DROP USER 'pi'@'localhost';
```



## Usuarios II

- ◉ Darle privilegios (**¡Cuidado!**)
  - ◉ Nunca acceder con «root»
  - ◉ Dar solo los privilegios necesarios
    - ◉ Cuanto más divididos en cuentas estén, mejor seguridad tendrás
      - ◉ 1 para actualizar, otra para leer, otra para borrar, para tablas específicas o importantes, no dar acceso nunca a todo (tablas del sistema) o varias BDD, lugar de acceso, acceso solo por SSL, etc.
      - ◉ ALL, SELECT(columnas, ...), UPDAT(c1, c2, ...), DROP, ...
      - ◉ ON \*.\* (root, todo), ON rpi.\* (BDD «rpi» en cualquier tabla),
      - ◉ TO 'usuario'@'lugar': localhost, % (cualquiera: no recomendado),
    - ◉ Se puede limitar a consultas por hora, actualizaciones, conexiones por hora/máximas, etc.
    - ◉ Que solo pueda dar permisos el «root» o una cuenta que haga de administrador (... WITH GRANT OPTION)



```
MariaDB [(none)]> GRANT SELECT, INSERT ON rpi.* TO 'pi'@'localhost';
```

- ◉ Permisos root en la BDD «rpi» para cualquier tabla

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON rpi.* TO 'pi'@'localhost';
```



## Usuarios III

- Ver nuevos privilegios

```
MariaDB [(none)]> SHOW GRANTS FOR 'pi'@'localhost';
+-----+
| Grants for pi@localhost |
+-----+
| GRANT USAGE ON *.* TO 'pi'@'localhost' IDENTIFIED BY PASSWORD '*72EA7A3B37C7CF2067FD7ACD7FF596E05B9A9242' |
| GRANT ALL PRIVILEGES ON `rpi`.* TO 'pi'@'localhost' |
+-----+
```

- Revocar privilegios
  - Quitaría los privilegios dados anteriormente y quedaría con «USAGE»
  - Se podrían revocar solo una parte: DELETE, UPDATE, GRANT etc.

```
MariaDB [(none)]> REVOKE ALL ON rpi.* FROM 'pi'@'localhost';
```

- Recargamos permisos (buena práctica)

```
MariaDB [(none)]> FLUSH PRIVILEGES;
```

- Mas información es: <https://mariadb.com/kb/en/library/grant/>



# Backup I

- Lógico vs físico
  - El lógico
    - Ocupa más espacio, lo crea en base a las tablas (estructura lógica), más flexible (el hardware/BDD/versión destino no tiene por qué ser el mismo), necesita más tiempo para hacerlo y restaurarlo, no tiene los ficheros de configuración ni los logs
  - El físico
    - Replica la estructura de directorios
- Backup físico
  - <http://www.lenzg.net/mylvmbackup/>



## Backup II

- Crear Backup lógico

- `mysqldump --user=user -p «nombreBDD» > «nombreSalida»`
- Contiene la creación de la BDD y todos su datos

```
~ $ mysqldump --user=root -p rpi > rpiDB.sql
```

- Varias BDD

- `mysqldump --user=root -p --databases db1 db2 > databases.sql`

```
~ $ mysqldump --user=root -p --databases rpi mysql > databases.sql
```

- Restaurar Backup lógico

- La BDD de entrada debe existir

```
~ $ mysql --user=root -p rpi < rpiDB.sql
```

- Varias BBDD

```
~ $ mysqldump --user=root -p --databases rpi mysql < databases.sql
```



# Ejemplo Apache + PHP + MariaDB

- <https://gist.github.com/gonzalezgarciacristian/0d4ec75a25e51237e4e7f1c1810457cd>

Raspberry Pi desde cero x +

localhost/phpmariadbTest.php

## Prueba de Apache, PHP y MariaDB

### Lista de la compra

ID	Nombre	Cantidad
1	Assassins Creed Origins	1
2	Civilization	1
3	16gb ram	4
4	2 HDD 2TB	2
5	Diablo	1
6	Nvidia GeForce 2080	2



# Conexión remota

# VNC vs XRDP

## ○ XRDP

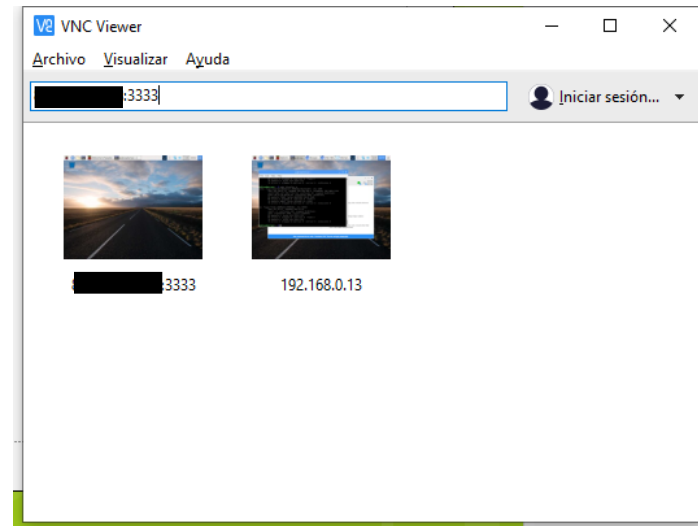
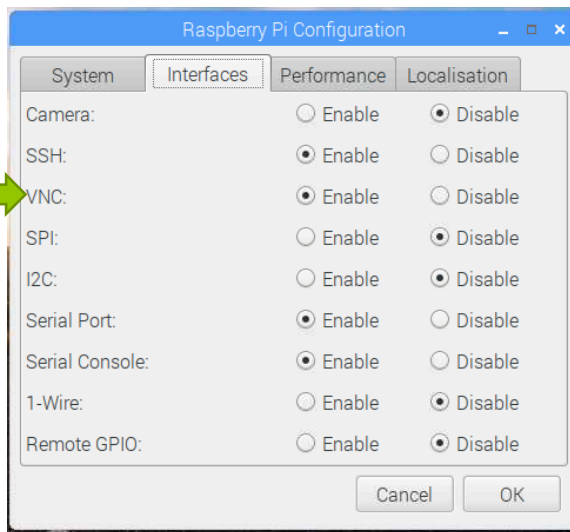
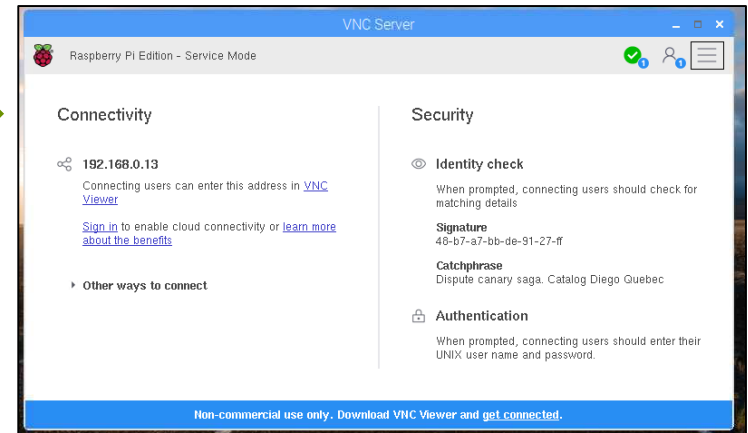
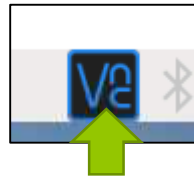
- Implementación gratuita del protocolo utilizado por el «Escritorio Remoto» de Windows (RDP)
- Se puede utilizar en remoto «in consecuencias» para el usuario local
  - Abre sesiones independientes desde cada sitio que se conecta
  - Dos ordenadores diferentes tienen diferente sesión abierta
- Más rápido que VNC
- En la última versión de Raspbian dejó de funcionar, de momento
  - Paquete de apt xrdp

## ○ VNC

- «Secuestra» el ratón para el usuario en local
  - Comparte la pantalla con el usuario local
  - Siempre abre la misma sesión
- Algo más lento que XRDP

# VNC

- Utiliza el Puerto 5900 por defecto
- Activarlo en las opciones de la Rpi
- Descargar el cliente (VNC Viewer)
  - <https://www.realvnc.com/en/connect/download/viewer/>



Otro software

## Otro Software

- MySQL
  - <https://dev.mysql.com/doc/mysql-apt-repo-quick-guide/en/>
- phpMyAdmin
  - <https://help.ubuntu.com/lts/serverguide/phpmyadmin.html.en>
- Administración BBDD
  - Windows: <https://www.heidisql.com/>
  - GNU/Linux: <https://alternativeto.net/software/dbeaver/about/>
- Ruby on Rails (seguir la guía de Ubuntu)
  - <https://gorails.com/setup/ubuntu/18.10>
- Python con Flask
  - <https://projects.raspberrypi.org/en/projects/python-web-server-with-flask>
- Foro de ayuda
  - <https://raspberrypi.stackexchange.com/>

# Servidor para Raspberry Pi desde cero

Cristian González García  
gonzalezcristian@uniovi.es

v 1.0.1 Febrero 2019

---