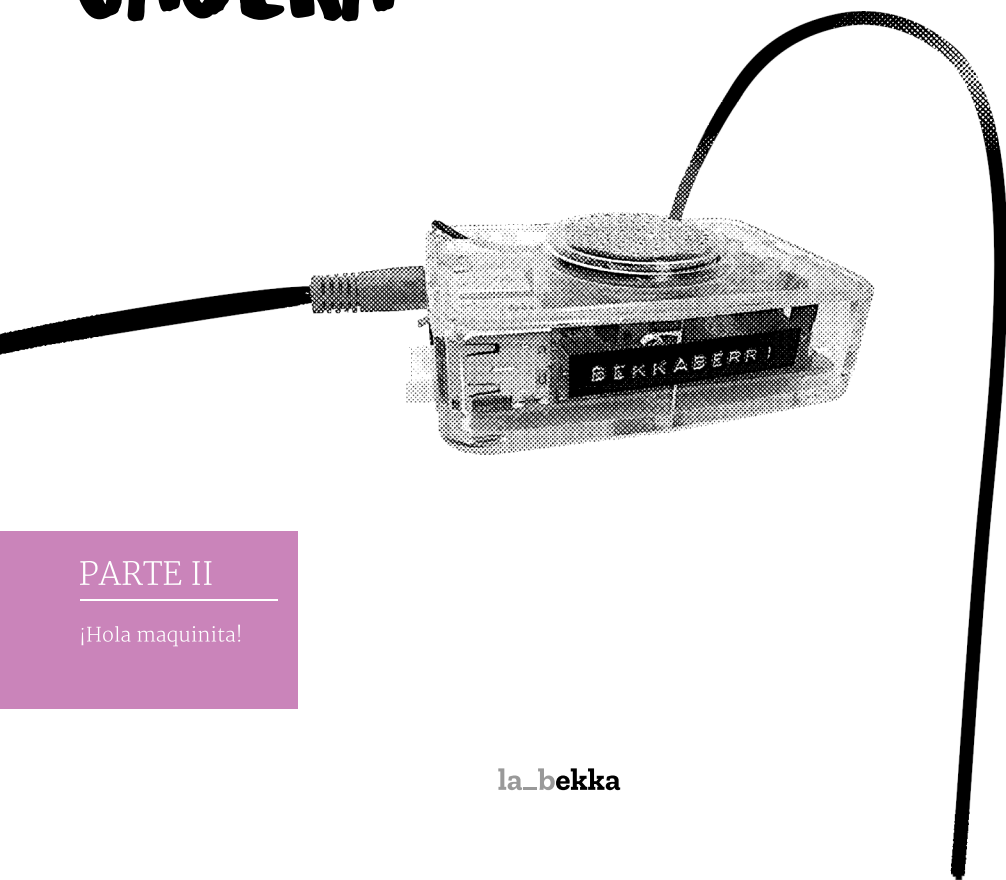


# CÓMO MONTAR UNA SERVIDORA FEMINISTA CON UNA CONEXIÓN CASERA



PARTE II

¡Hola maquina!

la\_bekka

**la\_bekka <espacio hackfeminista>**

<https://labekka.red>

[hacklabfeminista@riseup.net](mailto:hacklabfeminista@riseup.net)

GPG 0xD47C2A1A064BoD97

**Proceso de aprendizaje:**

Irene, m4rtu, Inés Binder, Andrea, Marta S.

**Redacción y contenidos:**

Inés Binder y m4rtu.

**Diseño y diagramación:**

Inés Binder y m4rtu.

Impreso en Canarias.



© la\_bekka, septiembre 2019

Este fanzine se libera bajo la licencia Producción Feminista de Pares. Esto significa que, reconociendo la autoría de la obra, se permite compartirla (copiarla, distribuirla, o comunicarla públicamente) y hacer obras derivadas y explotarlas comercialmente solamente si eres parte de una cooperativa, organización o colectiva sin fines de lucro, u organización de trabajadoras autogestionadas, que defiendan y se organicen bajo principios feministas. Además, debes compartir cualquier obra derivada de este fanzine con la misma licencia.

# PARTE II:

# ¡HOLA MAQUINITA!

---

## Contenidos

5. Introducción a GNU/Linux
  6. Conectemos la Raspberry Pi
  7. Instalemos un sistema operativo en la Raspberry Pi
  8. Accedamos remotamente a la Raspberry Pi usando SSH
- 

## 5. Introducción a GNU/Linux

GNU/Linux, también conocido informalmente como Linux, es un sistema operativo libre. Su desarrollo es uno de los ejemplos más prominentes de software libre: todo su código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera, bajo los términos de la licencia GPL (Licencia Pública General de GNU) y otra serie de licencias libres.

Existe una gran cantidad de distribuciones GNU/Linux, quizás les suenen Ubuntu, Debian, Mint, etc. Nosotras somos usuarias de Debian así que para esta guía elegimos usar Raspbian, una distribución para Raspberry Pi basada en Debian. Así que, como dijimos, las

instrucciones de esta guía están orientadas principalmente a GNU/Linux, lo que consideramos también como una apuesta política por el software libre y la soberanía tecnológica.

Aunque para seguir esta guía no hay que ser expertas sí es necesario tener algunos conocimientos básicos de GNU/Linux. Si no los tienen, o son escasos, no se preocupen. Esta puede ser una buena oportunidad para aprender. ¡Nosotras aprendimos mucho!

¡El proceso es más importante que el producto! ¡Disfrútenlo!

Pero repetimos, tengan paciencia, consulten otras fuentes y si dudan en algún paso ¡pregunten a las amigas, a la comunidad o a nosotras!

Veamos algunos poquitos comandos que sí van a necesitar:

**pwd**: muestra el directorio o ruta donde estén situadas en la terminal.

**ls -l**: lista todo aquello que esté en el directorio donde estén y especifica los permisos de cada elemento.

**ls -l /ruta/directorio**: lista el directorio que indiquen y muestra los permisos.

**ls -la**: lista todo al igual que **ls -l** pero incluyendo los archivos ocultos, que empiezan por un punto, por ejemplo: **.htaccess**.

**cd /ruta/directorio**: cambia al directorio que le indiquen.

**nano /ruta/archivo**: Nano es un editor de textos, así que con este comando pueden crear y editar un archivo de texto desde la terminal.

**ip a:** devuelve los datos de las interfaces de red, incluida la IP.

**ifconfig:** al igual que el comando anterior devuelve los datos de las interfaces de red, incluida la IP.

**hostname -I:** devuelve sólo la IP.

**chmod XXX /ruta:** cambia los permisos de un directorio o archivo (de escritura, lectura o ejecución). Las cifras XXX serán valores relacionados con los permisos que quieres cambiar. Aprenderemos más sobre permisos en el apartado 18.

**chown user:group /ruta:** cambia la usuaria y el grupo al que pertenece un directorio o archivo.

**sudo "comando":** permite ejecutar un comando como superusuaria, siempre que la usuaria que lo ejecute esté en el grupo sudoers.

**apt-get update:** actualiza la información de los repositorios.

**apt-get upgrade:** actualiza nuestro sistema operativo.

**apt-get install paquete:** instala el paquete que se le indica.

**service nombre\_del\_servicio status:** te muestra el estado de un servicio

**service nombre\_del\_servicio start** inicia un servicio

**service nombre\_del\_servicio stop** detiene un servicio

**cp /ruta/origen/ /ruta/destino/** copia un directorio o archivo de un lugar a otro.

**mkdir nuevo-directorio:** crea una directorio.

**rm /ruta/archivo:** borra un archivo.

**rm -r /ruta/directorio:** borra un directorio.

**mv /ruta/origen /ruta/destino:** se usa para cambiar los nombres de directorios y archivos.

**df -h:** nos dice el espacio libre en disco, en este caso en la tarjeta SD que tengamos en la Raspberry Pi.

**grep:** es un comando que te ayuda a buscar. Se puede usar para buscar dentro de lo que otro comando te devuelve poniendo **| grep** al final, y a continuación lo que quieran buscar.

**Regla de fuego en la terminal: ¡nunca ejecutamos comandos que no sepamos qué es lo que van a hacer en el sistema!**

Prueben a practicar con estos comandos en su terminal para estar cada vez más cómodas. Hay mucho contenido en Internet que puede ayudarlas con el uso de los comandos. Pueden buscar en su buscador favorito "Linux commands cheatsheet" y les saldrán miles de recursos (en Inglés).

Debemos saber que no todas las distribuciones GNU/Linux tienen los mismos comandos. Raspbian está basado en Debian y, en general, todos los comandos básicos para Debian funcionarán en Raspbian.

Ahora ya pueden hacer bromas ;) como:

```
sudo rm -r patriarcado  
sudo service machitrolls stop
```

Hay buena documentación sobre comandos en la web de Raspberry Pi: <https://www.raspberrypi.org/documentation/linux/> y <https://www.raspberrypi.org/documentation/raspbian/>, aunque lamentablemente sólo está en inglés.

Recomendamos tres recursos para aprender a usar la terminal:

- **Cliteratu**: <http://www.cliteratu.re/>. con ayuda de literatura nos introduce en este mundo.
- **Explain Shell**: <https://explainshell.com/>. explica qué significa cada cosa en un comando, en inglés.
- Una traducción al español de **The Linux Command Line** – Second Internet Edition en <https://licodeli.blogspot.com/>

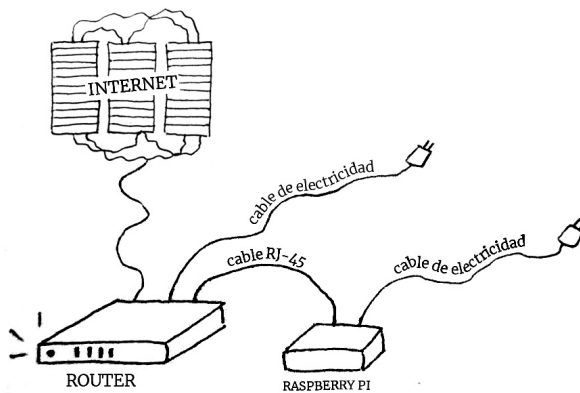
## 6. Conectemos la Raspberry Pi

Antes de hacer nada pueden comenzar familiarizándose un poco con su nueva máquina. ¿Identifican sus partes? ¡Conózcanla!, que ese pedazo de plástico y metales se convertirá en su servidorcita web autónoma.

Si están usando la misma Raspberry Pi que nosotras (la Raspberry Pi 3 Model B+, que en el momento de escribir esta guía era el último modelo), es importante saber sus prestaciones más básicas:

- Procesador Broadcom BCM2837Bo, Cortex-A53 (ARMv8) 64bit SoC @ 1.4GHz
- Memoria RAM 1GB LPDDR2 SDRAM.
- Interfaz de red inalámbrica 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN.
- Interfaz de red cableada Gigabit Ethernet over USB 2.0.
- Puerto Full-size HDMI.
- 4 puertos USB 2.0.
- Almacenamiento depende de la tarjeta Micro SD que le pongan.
- Conector de energía 5V/2.5A DC power input.

Ahora que ya están más familiarizadas con la Raspberry Pi, hay que elegir el lugar en el que van a querer ubicarla. Esta decisión es importante y hay que tener en cuenta varias consideraciones ya que va a estar allí mucho rato.



*Possible esquema de conexión.*



Algo imprescindible es que esté conectada al *router* con un cable de internet (RJ-45 se suelen llamar) y a la red eléctrica, así que no puede estar muy lejos del *router* y de un enchufe.

Algunos *routers* tienen puertos USB así que podrían aprovechar para obtener la energía desde ahí con un cable USB.

Pero también tenemos que hacer una evaluación de riesgos y tener en cuenta nuestra protección. Como el sitio web que alojaremos será accesible desde Internet tenemos que saber que con esa dirección IP se puede llegar al lugar físico en donde está la máquina. Y si ese lugar es nuestra casa implica un grado de riesgo. Además, tendremos que abrir algunos puertos del *router*. Si bien tomaremos medidas para no exponer a los equipos que estén conectados a esa misma red (nuestras computadoras, nuestros teléfonos, etc.) es importante que sepamos que siempre existe un riesgo.

En este sentido, es bueno valorar el lugar en términos de seguridad física. Es recomendable que tengamos acceso físico a la máquina pero así como lo tenemos nosotras también lo puede tener otra gente: ¿cuán accesible es este lugar para otras personas?, ¿cuán seguro es?, ¿cuán a la vista está la máquina?, ¿tiene una conexión de red y eléctrica estable?, ¿podría alguien desenchufarla o robarla?. A veces, incluso, puede ser sin intención. Alguien desconectó una de nuestras Raspberry Pi pensando que consumía mucha energía sin necesidad. Hablen con su comunidad o colectiva o pongan un cartelito para hacer saber que es importante cuidar la máquina. ¡Y no hace falta desenchufarla porque consume poquito!

Si tienen un lugar seguro pueden crear su pequeño altar transhackfeminista y colocar allí la servidora conectada al *router*. Nosotras, por ejemplo, le pusimos nombre. ¡Háganla suya!, los rituales son importantes para nosotras.



*Aquí nuestro pequeño altar transhackfeminista*

## **7. Instalemos un sistema operativo en la Raspberry Pi**

Lo primero que necesitamos para comunicarnos con la Raspberry Pi son algunos periféricos: tenemos que conectarle un monitor (o un televisor) a través de un cable HDMI, un teclado y un ratón/mouse a los puertos USB, y a Internet a través de un cable de red.

Una vez que tengamos instalado el sistema operativo en nuestra Raspberry Pi podremos acceder a ella de manera remota a través de otra computadora pero, como aún no tenemos nada, durante este proceso de instalación debemos hacerlo así. Luego ya podrán dejar la servidora en su "altar transhackfeminista" sin necesidad de pantallas y otros cables.

El sistema operativo que decidimos instalar es Raspbian, la distribución oficial de Debian para Raspberry Pi, es decir un sistema GNU/Linux (si para la servidora usarán una computadora normal pueden instalar Debian directamente). Se puede instalar Raspbian de varias maneras pero pensamos que usar el instalador NOOBS es lo más Internet y un lector de tarjetas SD, que las computadoras portátiles más nuevas tienen incorporado. Si no tienen, pueden usar un lector de SD o también creemos que se puede usar el celular/móvil.

Para proceder a la instalación el primer paso es descargar NOOBS desde la página de Raspberry Pi: <https://www.raspberrypi.org/downloads/noobs/>. Como nuestra Raspberry Pi estará conectada a Internet (con el cable de red al *router*) podemos elegir la *NOOBS Lite Network install only*, una versión más liviana que descarga paquetes de la red a medida que va instalando el sistema operativo.

Una vez descargado el archivo **.zip** lo descomprimos y pasamos todos los archivos a la tarjeta microSD (a través del conector del que hablamos). Si la tarjeta que están usando no está formateada pueden hacerlo antes de pasar los archivos. Se formatean igual que los pendrives.

Una vez que los archivos estén copiados en la tarjeta la sacamos del lector y del adaptador, la insertamos en la Raspberry Pi y la encendemos. La Raspberry Pi no tiene botón de encendido sino que se enciende apenas enchufa. Verán unas lucecitas.

En el monitor verán que el instalador NOOBS empieza a correr automáticamente. Lo primero que hará será preguntarnos qué sistema operativo queremos instalar: elegiremos Raspbian Lite que viene sin entorno de escritorio (la interfaz gráfica que no necesitaremos) y es más liviano.

Cuando termine el proceso de instalación veremos una pantalla negra que es equivalente a lo que vemos cuando abrimos la consola/terminal. Veremos algo así como **pi@raspberrypi:~\$**, que nos indica el nombre de la usuaria y de la máquina.

### ***7.1 Cambiemos la contraseña por defecto***

Raspbian viene con una usuaria por defecto que es **pi** y su contraseña es "raspberrypi". Como todas las instalaciones tienen la misma contraseña la primera medida de seguridad que tomaremos deberá ser cambiar la contraseña que trae por defecto.

Escribimos el siguiente comando en la consola y damos *enter*:

```
sudo passwd pi
```

Escribimos la nueva contraseña -no se preocupen si escriben y no aparece nada, se está escribiendo igual- y la confirmamos con *enter*. ¡Listo! Nuestro sistema operativo está instalado y con una contraseña única. Recuerden que para que una contraseña sea ro-

busta debería tener por lo menos 16 caracteres, letras mayúsculas y minúsculas, números y caracteres especiales; y, además, no debemos utilizarla para ningún otro servicio.

## **7.2 Actualicemos el sistema operativo**

El siguiente paso será actualizar los repositorios. Los repositorios son unas bases de datos donde están listados todos los programas, librerías, *drivers*, etc., que podamos necesitar. Existen repositorios oficiales –que son los recomendados porque sólo agregan código testeado y compatible con nuestro sistema operativo–.

Para actualizar los repositorios de nuestro sistema escribimos:

```
sudo apt-get update
```

Y luego actualizamos el sistema (será necesario que la Raspberry Pi esté conectada a Internet):

```
sudo apt-get upgrade
```

Ya tenemos nuestro sistema operativo instalado y actualizado. Si surgen problemas con la instalación o las explicaciones les quedan un poco cortas puedes buscar en Internet otros tutoriales en detalle, ¡hay muchos! :).

En este punto pueden aprovechar para familiarizarse con el entorno GNU/Linux para estar cada día más cómodas con la terminal. Pueden probar cambiar de directorios, crear archivos de texto, leerlos y editarlos, copiarlos de un directorio a otro, etc.

## 8. Accedamos remotamente a la Raspberry Pi usando SSH

Una conexión SSH nos permite conectarnos a otra computadora y trabajar en ella en remoto, como si estuviéramos delante de la propia máquina pero sin estarlo. Raspbian no viene con el servicio SSH habilitado por defecto desde 2016 así que debemos habilitarlo manualmente. Para hacerlo tenemos que ejecutar en la consola de la Raspberri Pi el siguiente comando:

```
sudo raspi-config
```

Este es un comando que nos abrirá una ventanita de diálogo para configurar rápido algunas cosas de nuestro sistema operativo (contraseñas, SSH, etc). Busquen la opción Interfacing Options, y entren pulsando *enter* en SSH. Nos preguntará si queremos habilitar el servicio: Would you like the SSH server to be enabled?, a lo que responderemos Yes. Luego le damos a Aceptar y para salir del menú vamos a Finish. Si necesitan más información sobre cómo habilitar SSH en Raspbian pueden visitar: <https://www.raspberrypi.org/documentation/remote-access/ssh/README.md>

Ahora ya nos podemos conectar a la Raspberry Pi por SSH desde otra computadora que esté conectada a la misma red interna; es decir, que esté conectada al mismo router.

¡Vamos a ello! Para conectarnos por SSH lo primero que necesitamos es saber la dirección IP de nuestra Raspberry Pi. Para averiguarla hay varias opciones. La más sencilla es ejecutar en la

terminal de la Raspberry Pi el comando **hostname -I**. Como respuesta tendrán la dirección IP que necesitan.

También pueden intentar trastear/cacharrear/jugar/experimentar con la configuración del *router* para ver qué dirección IP le ha asignado; o, escanear la red con el comando **nmap** desde otra computadora conectada a esa misma red; o, incluso, pueden usar alguna aplicación para Android que escanee la red. ¡Entonces ya van a ser expertas en redes!

Una vez que ya sepan la dirección IP de de la Raspberry Pi pueden ir a otra computadora abrir una terminal y ejecutar el siguiente comando:

```
ssh pi@<dirección ip>
```

Por ejemplo, se verá algo así:

```
ssh pi@192.168.1.145
```

Nota: pueden usar cualquier sistema operativo para conectarse con la servidora pero recuerden que las instrucciones que damos son para GNU/Linux o, en su defecto, para MacOS, que también deriva de Unix.

La primera vez que ejecuten este comando les dará una advertencia por defecto a la que contestaremos "yes". Luego nos pedirá la contraseña de **pi** (esa misma que le asignamos anteriormente cuando la cambiamos) y ya estaremos conectadas a nuestra Raspberry Pi. Igual que si estuviéramos delante de una terminal en su monitor. Verán que la terminal en vez de mostrar la usuaria y el

nombre de la máquina que estén usando cambiará a **pi@raspberry-\$. Si quieren saber más sobre las conexiones SSH desde GNU/Linux o MacOS pueden visitar: <https://www.raspberrypi.org/documentation/remote-access/ssh/unix.md>**

Si usan Windows pueden seguir esta guía: <https://www.raspberrypi.org/documentation/remote-access/ssh/windows.md>

### **8.1 Chau contraseñas, bienvenidas llaves SSH**

Ahora que ya saben conectarse por SSH a su Raspberry Pi con la usuaria **pi** y su contraseña, veamos cómo conectarnos de manera más segura utilizando llaves SSH. La diferencia está en que en vez de abrir la puerta de nuestra servidora con una contraseña, que la puede saber cualquiera, utilizaremos una llave que es única para cada usuaria. Aunque de manera muy simplificada podríamos decir que tenemos que subir nuestro candado (llave pública) a la servidora y así poder acceder con nuestra llave (llave privada). Cada llave es personal e intransferible.

Si no tienen su par de llaves SSH tienen que crearlas. En la computadora que estén usando para conectarse deben ejecutar el siguiente comando:

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

El sistema les solicitará algunos datos, que pueden dar o dejar vacíos dándole *enter*, y una contraseña. Esta contraseña será necesaria cada vez que se conecten usando este par de llaves. Por lo



general, el par de llaves se guardará en el directorio **.ssh** del directorio personal. Verán que la llave privada se llamará **id\_rsa** y la pública **id\_rsa.pub**. ¡Cuidado! Nunca compartan la llave privada.

El siguiente paso, por lo tanto, será subir la llave pública (el candado) a la Raspberry Pi. Esto significa que debemos copiarla al archivo **~/.ssh/authorized\_keys** en la Raspberry Pi. Lo podemos hacer escribiendo el siguiente comando en la consola de nuestra computadora (no de la Raspberri Pi):

```
ssh-copy-id -i pi@<ip de la raspberry>
```

Como todavía no está subida nuestra llave nos pedirá la contraseña de **pi** para poder establecer la conexión. Una vez que la ingresemos y demos enter nuestra llave pública se habrá copiado. Probemos entonces si podemos establecer una conexión SSH.

Volvamos a conectarnos, siempre desde la misma red local:

```
ssh pi@<dirección ip>
```

Ahora, en lugar de pedirnos la contraseña de **pi** nos pedirá la contraseña de nuestra llave SSH. Si todo salió bien deberíamos estar conectadas a nuestra Raspberry Pi. Como verán estos pasos son para ser ejecutados desde GNU/Linux pero seguro que hay tutoriales que les cuenten cómo generar llaves SSH desde Windows.

Lo último que haremos en este apartado será restringir el acceso remoto con contraseñas. De esa manera sólo podrán conectarse remotamente aquellas personas que hayan subido previamente su

llave SSH pública a la Raspberry Pi. Lo que tendremos que hacer es editar el archivo de configuración **sshd\_config**. Ejecutamos el siguiente comando en nuestra Raspberry Pi:

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

Y en la línea donde pone PasswordAuthentication poner **no**:

```
PasswordAuthentication no
```

Guardamos los cambios (**Ctrl+O**, *enter*, **Ctrl+X**) y, por último, reiniciamos el servicio SSH para que se haga efectivo el cambio en la configuración. Ejecutamos:

```
sudo service ssh restart
```

¡Felicitaciones amigas!

En este apartado logramos muchas cosas: instalamos un sistema operativo en la Raspberry Pi, tomamos las primeras medidas de seguridad, habilitamos el servicio SSH, creamos nuestras llaves SSH y nos conectamos de manera remota a nuestra máquina.

¡Ahora nos queda convertir esta maquineta en una servidora!

labekka.red