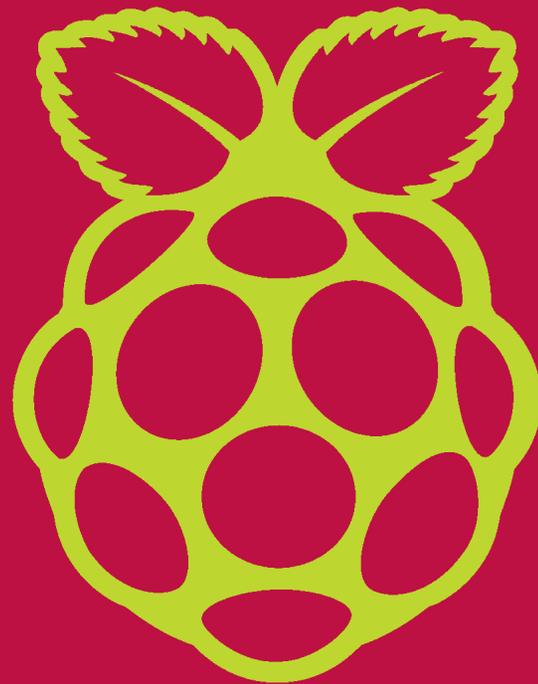


Cómo programar una caja de música con Raspberry Pi.



Introducción

Este tutorial nos permitirá crear nuestro primer generador aleatorio de sonidos. Aprenderemos el uso y configuración de la micro computadora Raspberry Pi usando el software de programación de sonido Pure Data.

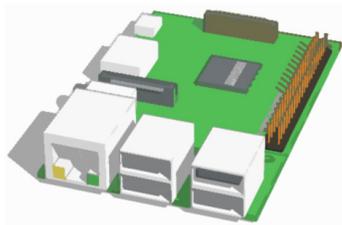
Raspberry Pi es un proyecto educacional que consiste en una micro computadora modular del tamaño de una tarjeta de crédito. Está diseñada para aprender a programar algunos lenguajes como Scratch y Python, pero de igual manera se conecta a un monitor vía HDMI, a un teclado y mouse con USB, a internet con un puerto Ethernet y a unas bocinas o audífonos con una salida de audio de 3.5mm. Prácticamente tenemos una computadora completamente funcional a un precio muy accesible, que además de fomentar la exploración de algunos lenguajes de programación nos permitirá usar internet, hojas de cálculo, editores de texto, ver videos y hasta correr algunos juegos.

Raspberry Pi no funcionará exactamente como estamos acostumbrados con sistemas operativos comerciales como son MAC OSX y Windows. Raspberry tendrá sus propios sistemas operativos basados en Linux, una de las distribuciones de sistema operativo open source más importantes. En este tutorial aprenderemos a instalar el sistema operativo Raspbian Wheezy.

Por otro lado, Pure Data es un software libre (open source) de programación de audio con un ambiente totalmente gráfico. No sólo ofrece una interfaz sumamente amigable para el usuario, sino que también nos da la posibilidad de involucrarnos con la programación de una forma más atractiva.

Existen diferentes versiones de Raspberry Pi, pero en este tutorial usaremos Raspberry Pi B+. Sin importar el modelo de tu Raspberry, este tutorial funcionará para todos los modelos siempre y cuando instales el sistema operativo Raspbian Wheezy, ya que la versión de Pure Data para Raspberry está soportada para esa distribución.

Materiales



Raspberry Pi
(modelos A, A+, 2B, 2B+)



Cable micro USB



Eliminador de 2
Amp(máximo), 5 v.



Tarjeta micro SD 4GB mínimo



Adaptador USB de micro SD
(En caso de no tener entrada
SD en tu computadora)



Teclado, mouse (USB)



Monitor con entrada HDMI



Cable Ethernet

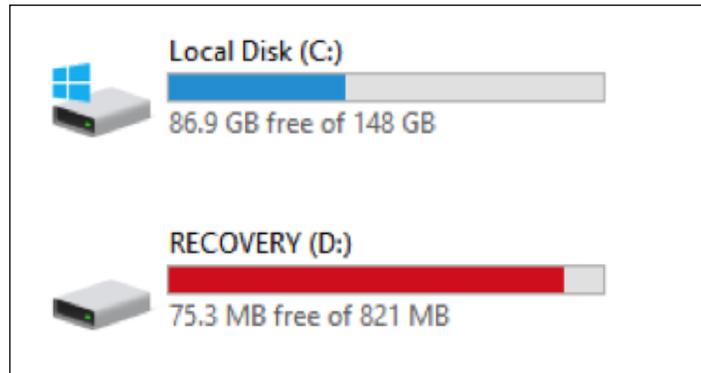
Windows

Una vez teniendo los materiales, descarga el sistema operativo Raspbian Wheezy en tu computadora personal. Lo puedes encontrar en la página oficial de Raspberry Pi: <https://www.raspberrypi.org/downloads/raspbian/>. Una vez descargado, es necesario seguir las instrucciones dependiendo del sistema operativo de tu computadora.

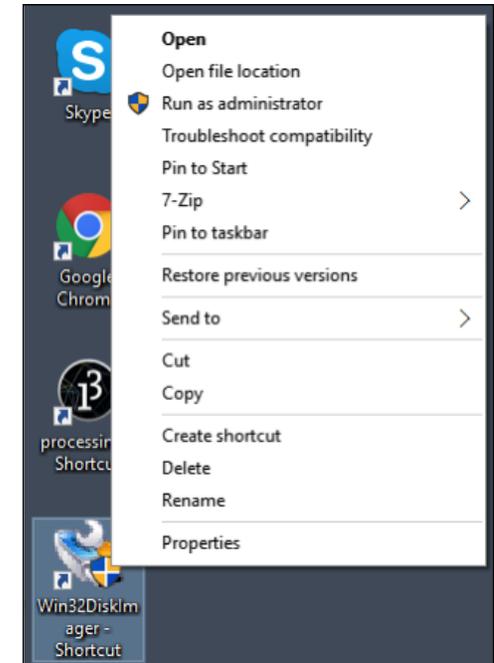
PARA WINDOWS

Descargar e instalar la utilería de Win32diskimager:
<http://sourceforge.net/projects/win32diskimager/>

Una vez instalada, conecta el micro SD a la computadora, ya sea mediante un adaptador o directamente a la computadora. Al conectarlo es importante recordar la letra asignada a la unidad extraíble. (Véase imagen)



Después es necesario correr el programa Win32diskimager en modo Administrador. Esto lo podremos hacer dando click derecho en el icono y seleccionando la opción.



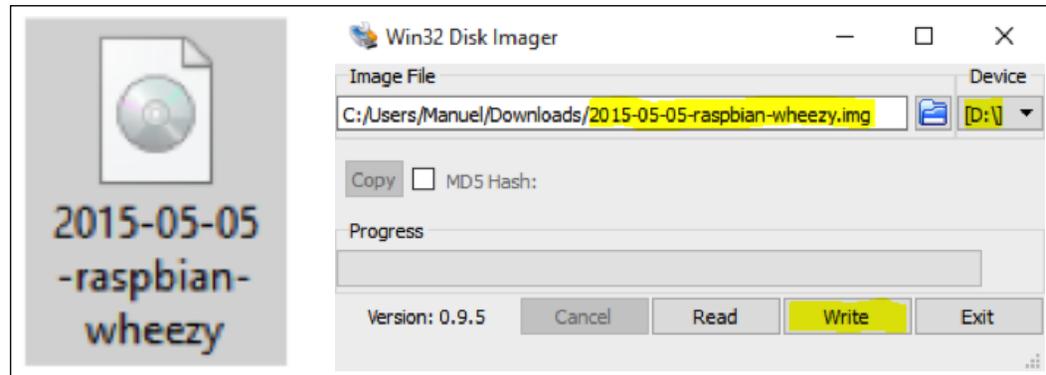
NOTA

*En el caso de mi computadora, a la unidad se le asignó la letra (D:)

Windows

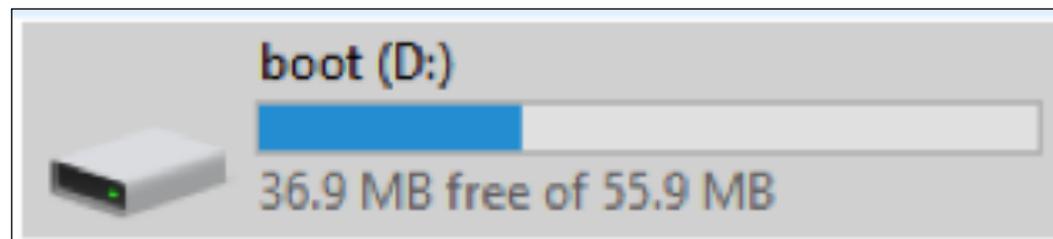
Para los usuarios de Windows será necesario descomprimir el .zip de Raspbian Wheezy con 7-ZIP, un programa gratuito para descomprimir archivos. En caso de no tenerlo instalado se puede bajar de la página oficial: <http://www.7-zip.org/>

Al descomprimirlo obtendremos un archivo con la terminación **.iso**. Con todo esto listo, seleccionaremos la imagen para escribirla en el SD como podemos ver a continuación.



Lo más importante es corroborar que el dispositivo corresponda a la letra asignada (abajo), de lo contrario podrías borrar toda la información de tu disco duro (en caso de seleccionar C:/, por ejemplo). Una vez confirmado da click en Escribir/Write. Este proceso podría durar de 5 a 10 minutos dependiendo de tu computadora.

Una vez terminado, tendremos nuestro SD disco de arranque listo para conectar a la Raspberry Pi.



Mac OS

Para MAC OS

Una de las formas más sencillas de instalar Raspbian, así como la mayoría de los otros sistemas disponibles está a nuestro alcance mediante la opción de NOOBS, la cual se puede obtener desde la página oficial de Raspberry:

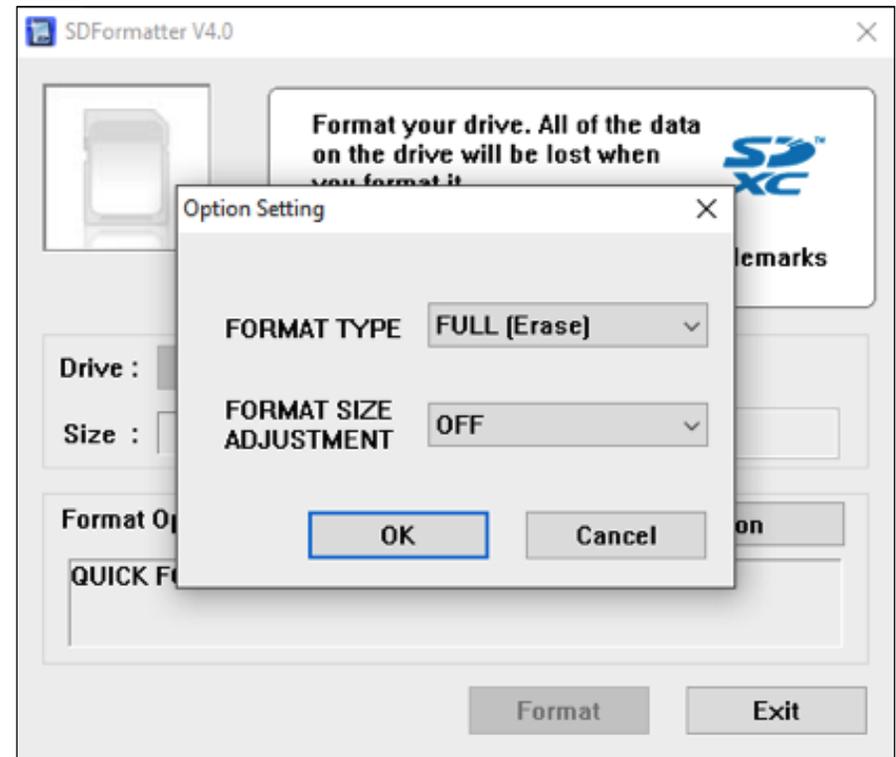
<https://www.raspberrypi.org/downloads/noobs/>

Esta es una solución multi-plataforma, sin importar cuál sea el sistema operativo con el que vayas a configurar tu SD. Si cuando compraste tu Raspberry Pi incluía una SD con NOOBS, entonces puedes saltar a la parte II.

Si no tienes la SD con NOOBS entonces descárgalo del link anterior y de igual forma será necesario descargar el software de SDFormater V4.0:

https://www.sdcard.org/downloads/formatter_4/

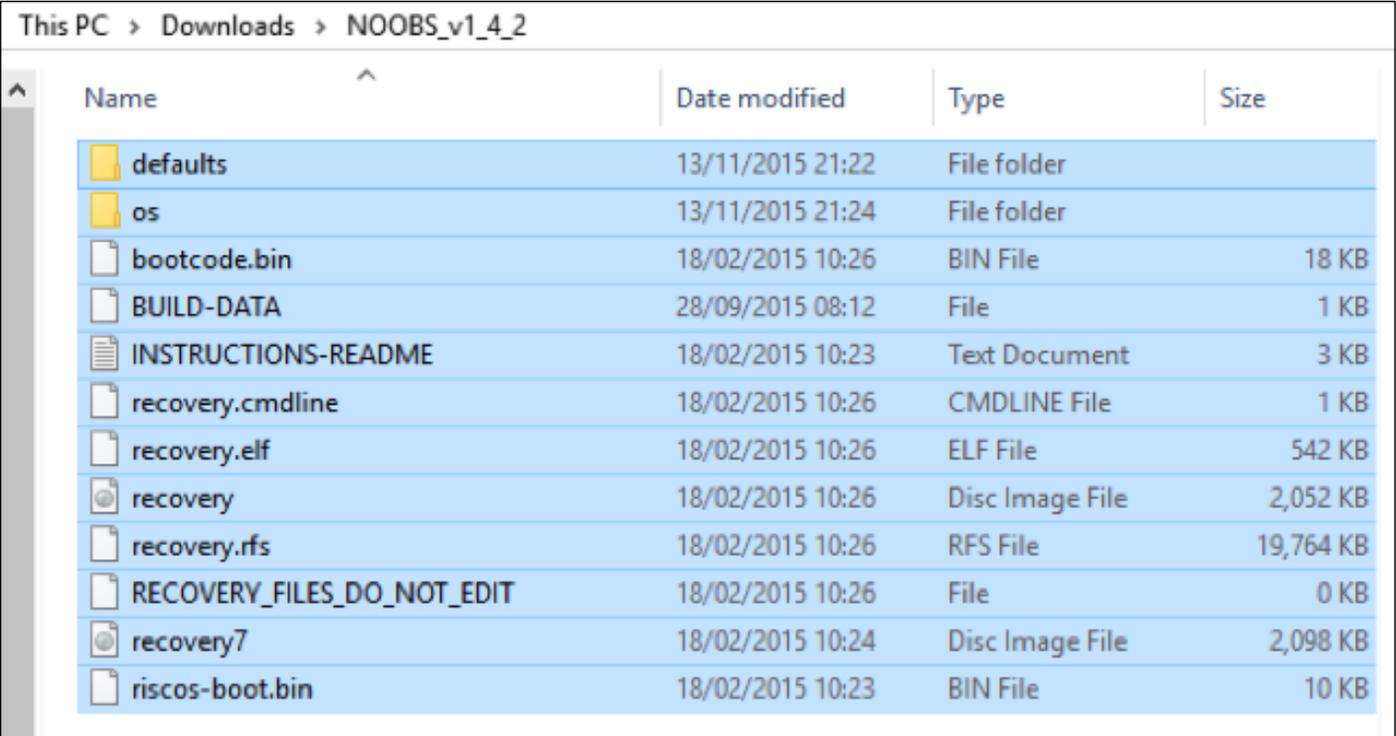
Descomprime ambos .zip e instala SDFormater 4.0. Ejecuta el SDFormater y escoge la opción de **borrado completo** como veremos a continuación.



Mac OS

Después elige la letra con la que el sistema asignará a tu SD* y finalmente formatea.

Al finalizar el formateado del SD abre la carpeta donde descomprimiste el NOOBS y arrastra los archivos hacia la SD formateada. Por default se llamará BOOT.



Name	Date modified	Type	Size
defaults	13/11/2015 21:22	File folder	
os	13/11/2015 21:24	File folder	
bootcode.bin	18/02/2015 10:26	BIN File	18 KB
BUILD-DATA	28/09/2015 08:12	File	1 KB
INSTRUCTIONS-README	18/02/2015 10:23	Text Document	3 KB
recovery.cmdline	18/02/2015 10:26	CMDLINE File	1 KB
recovery.elf	18/02/2015 10:26	ELF File	542 KB
recovery	18/02/2015 10:26	Disc Image File	2,052 KB
recovery.rfs	18/02/2015 10:26	RFS File	19,764 KB
RECOVERY_FILES_DO_NOT_EDIT	18/02/2015 10:26	File	0 KB
recovery7	18/02/2015 10:24	Disc Image File	2,098 KB
riscos-boot.bin	18/02/2015 10:23	BIN File	10 KB

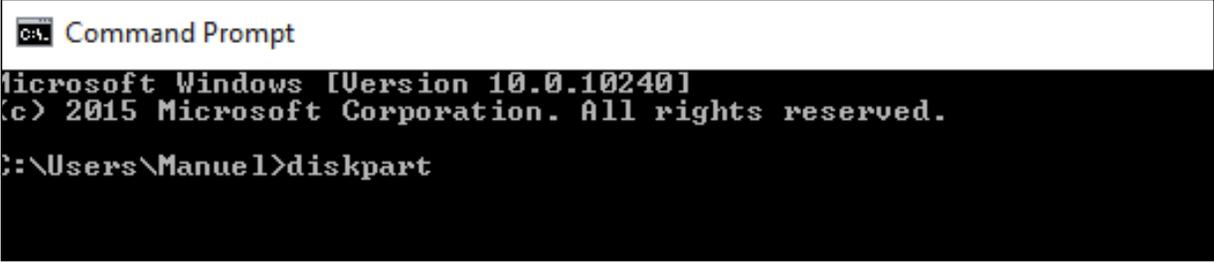
NOTA

*Es importante saber qué letra se asigna D:/ o E:/, etc. Ya que de lo contrario podríamos formatear alguna unidad extraíble que no deseáramos.

Instrucciones

Una vez copiados los archivos extrae el SD de manera segura y continua con la Parte II.

Una alternativa adicional es realizar el formateo mediante el símbolo del sistema en Windows. Para lograr esto escribiremos **cmd** en la barra de búsqueda del menú de Windows y se abrirá la terminal de símbolo del sistema.



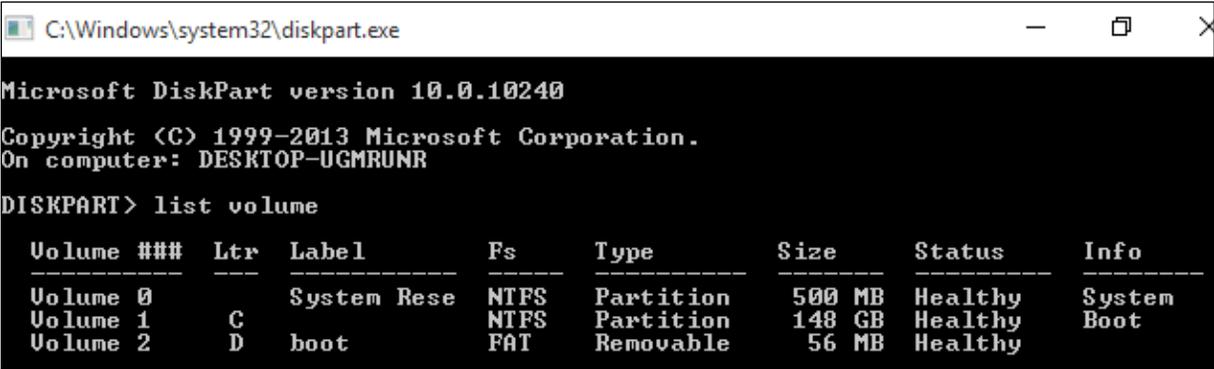
```
C:\> Command Prompt

Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Manuel>diskpart
```

Una vez que tengamos nuestra terminal de símbolo del sistema ejecutaremos el comando **diskpart**. Este comando ejecutará otro programa con la misma interfaz que el símbolo del sistema y requerirá el permiso del administrador para ejecutarse.

A continuación hay que ejecutar un comando dentro de la terminal de diskpart. Este comando enlistará los volúmenes de almacenamiento lógicos conectados interna y externamente en la computadora.



```
C:\Windows\system32\diskpart.exe

Microsoft DiskPart version 10.0.10240
Copyright (C) 1999-2013 Microsoft Corporation.
On computer: DESKTOP-UGMRUNR

DISKPART> list volume

  Volume ###  Ltr  Label          Fs          Type        Size     Status       Info
  -----  -  -  -  -  -  -  -  -
  Volume 0             System Rese  NTFS       Partition   500 MB    Healthy      System
  Volume 1             C          NTFS       Partition  148 GB    Healthy      Boot
  Volume 2             D          boot       FAT         Removable  56 MB     Healthy
```

Instrucciones

Aquí podemos ver que hay una partición del dispositivo SD que está asociada al Volume 2, entonces procede a enlistar los discos físicos con el comando **list disk** para asegurarnos que el volumen lógico efectivamente corresponde al disco físico. Una vez asegurado lo anterior, ya que el SD es de 16 Gb, continua a seleccionar el disco para realizar el formateo con el comando **select disk1**.

```
DISKPART> list disk

   Disk ###  Status              Size               Free               Dyn  Gpt
   -----  -
   Disk 0    Online              149 GB              0 B
   Disk 1    Online              14 GB              3072 KB

DISKPART> select disk 1

Disk 1 is now the selected disk.

DISKPART> list disk

   Disk ###  Status              Size               Free               Dyn  Gpt
   -----  -
   Disk 0    Online              149 GB              0 B
  * Disk 1    Online              14 GB              3072 KB
```

Instrucciones

Al volver a enlistar los discos físicos de la computadora verás el disco que acabas de seleccionar con un asterisco. Entonces puedes continuar para limpiar y formatear el disco SD.

```
DISKPART> clean
DiskPart succeeded in cleaning the disk.
DISKPART> create partition primary
DiskPart succeeded in creating the specified partition.
DISKPART> format fs=fat32 quick
    100 percent completed
DiskPart successfully formatted the volume.
DISKPART> assign
DiskPart successfully assigned the drive letter or mount point.
DISKPART>
```

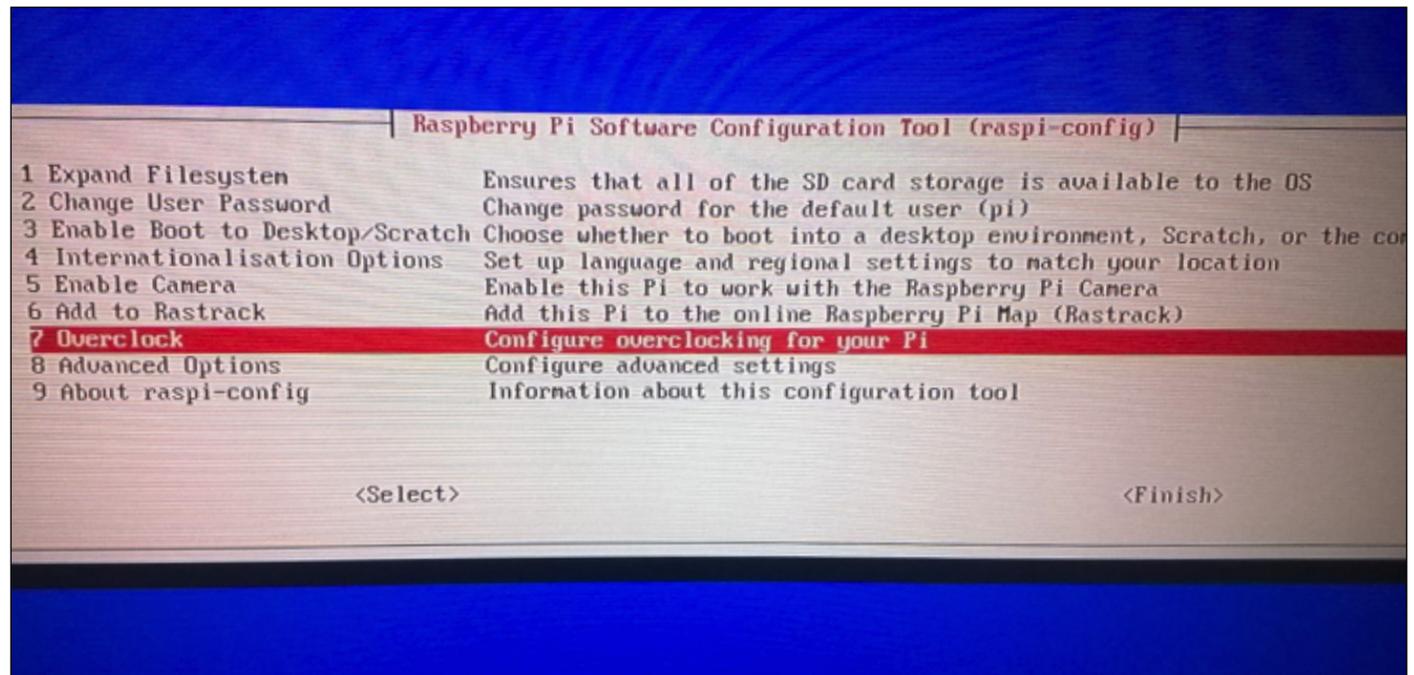
Con los comandos en el orden anterior podrás formatear completamente el SD con el símbolo del sistema. Puedes proceder con el copiado de los archivos o la escritura de la imagen de **raspbian**.

Raspbian Wheezy

En esta sección instalaremos y configuraremos todo lo necesario para poder usar el sistema operativo de Raspbian con su interfaz gráfica y con la terminal.

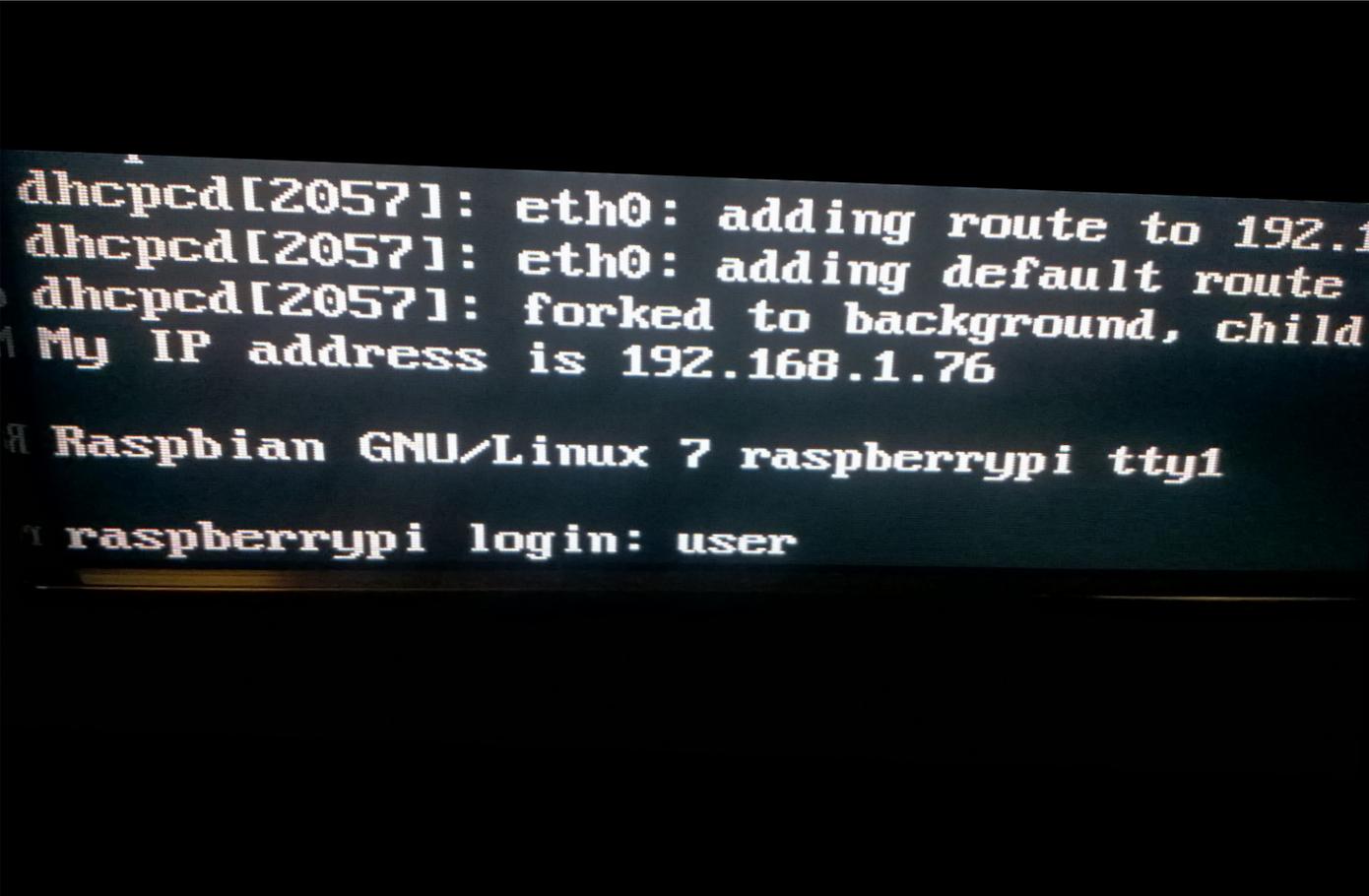
Si decidiste usar la primera opción de instalación entonces tu SD tiene cargada la imagen de Raspbian Wheezy solamente. Deberás conectar tu teclado, mouse, cable HDMI a tu monitor; cable Ethernet y cable micro USB conectado a tu cargador de 1 - 2 Amp.

Al conectar la alimentación del cable USB, la Raspberry encenderá y comenzará la instalación automáticamente. Mandará una pantalla que permite escoger algunas de las opciones avanzadas para la Raspberry.



Raspbian Wheezy

En esta pantalla lo único que hay que cambiar es la configuración del **overclocking** a **moderada** para poder así obtener mejor rendimiento en el software de Pure Data. También se podrá configurar la región en la que se encuentre. Al finalizar, los cambios deberán ser guardados y la instalación del sistema continuará.



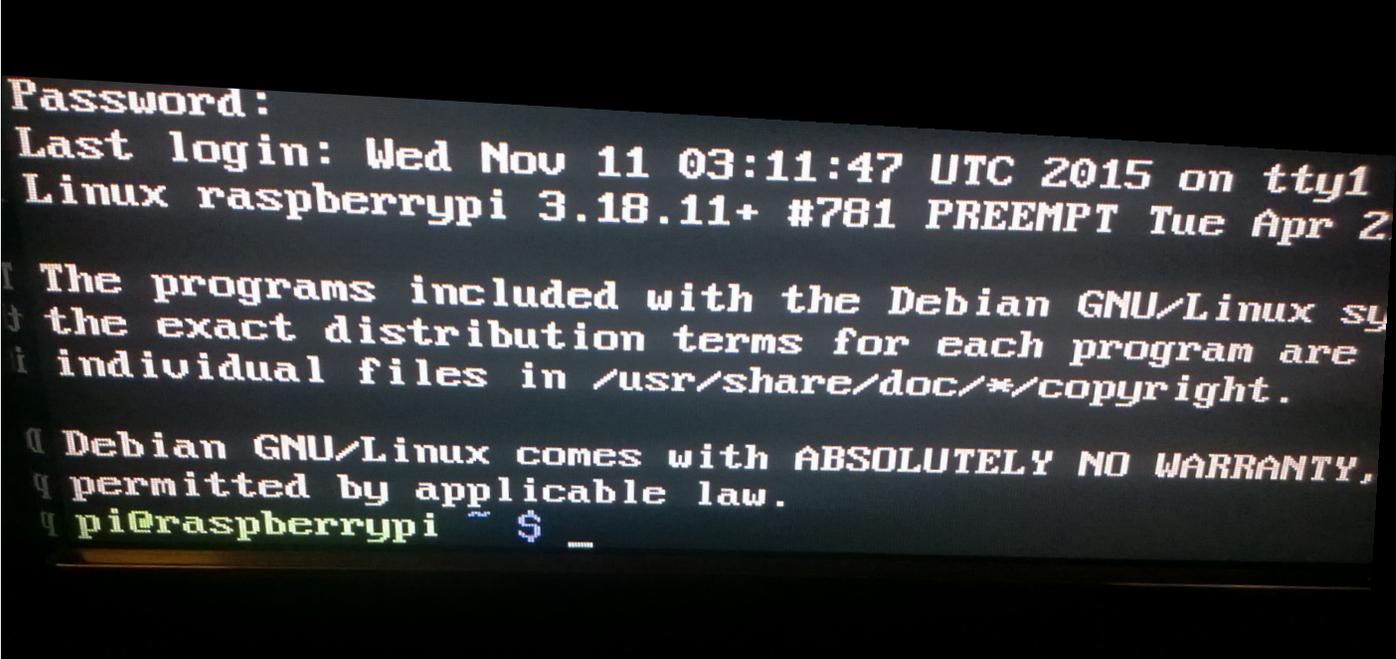
```
dhcpcd[2057]: eth0: adding route to 192.168.1.1
dhcpcd[2057]: eth0: adding default route
dhcpcd[2057]: forked to background, child pid: 2058
My IP address is 192.168.1.76

Raspbian GNU/Linux 7 raspberrypi tty1
raspberrypi login: user
```

Raspbian Wheezy

Al finalizar la instalación esta pantalla negra pide el usuario y la contraseña, las cuales están configuradas por default a **usuario**: pi; **contraseña**: raspberry

Una vez logrado el acceso tendrás una terminal como la siguiente:



```
Password:
Last login: Wed Nov 11 03:11:47 UTC 2015 on tty1
Linux raspberrypi 3.18.11+ #781 PREEMPT Tue Apr 2
The programs included with the Debian GNU/Linux system
are free software; the exact distribution terms for each program are
given in the individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY,
to the extent permitted by applicable law.
pi@raspberrypi ~ $
```

A partir de aquí podemos comenzar con la parte de la programación de Raspberry Pi.

Esta es una terminal Shell que usa un lenguaje simple llamado **Bash**. Dicho lenguaje es una compilación de órdenes redactadas en comandos, los cuales te permiten dar instrucciones a la computadora.

Raspbian Wheezy

Por ejemplo, estos son algunos comandos básicos en la terminal:

Pwd sirve para ubicar el directorio en el que estas actualmente.

```
pi@raspberrypi ~ $ pwd
/home/pi
```

Dir permite saber todos los directorios (carpetas) que hay dentro del directorio en el que estas actualmente.

```
pi@raspberrypi ~ $ dir
Desktop      Downloads  Pictures  python_games  Videos
Documents   Music      Public    Templates
```

El comando **cd** servirá para navegar entre los directorios. Por ejemplo, si estas en `/home/pi`, sabes que dentro del directorio `/pi` hay 9 directorios, de los cuales **Documents** es uno de ellos, por lo que al usar **cd Documents** nos llevará dentro del directorio de **Documents**. Para regresar al directorio `pi/` simplemente utiliza **cd**.

```
pi@raspberrypi ~ $ cd Documents
pi@raspberrypi ~/Documents $ cd ..
pi@raspberrypi ~ $ localhost
```

El comando **hostname** te da el nombre que tiene la computadora que estes usando actualmente. Esto será útil posteriormente. Puedes experimentar con estos comandos para navegar por las carpetas del sistema.

```
pi@raspberrypi ~ $ hostname
raspberrypi
```

Raspbian Wheezy

Apt-get (Advanced Package Tool) es probablemente uno de los comandos más importantes, con el modificador **update**. **Sudo** será necesario usarlo cuando se requieran permisos especiales para modificar archivos o configuraciones importantes en el sistema. La mayoría de los sistemas basados en Linux usan protocolos de seguridad muy efectivos. Al correr este comando se deberá entonces correr el siguiente comando: **sudo apt-get upgrade #**(tomará un tiempo). Estos comandos sirven para actualizar e instalar los paquetes necesarios para actualizar al sistema. También, este comando servirá para instalar o desinstalar programas.

```
pi@raspberrypi ~ $ sudo apt-get update
Hit http://archive.raspberrypi.org jessie InRelease
Hit http://mirrordirector.raspbian.org jessie InRelease
Hit http://archive.raspberrypi.org jessie/main Sources
Hit http://mirrordirector.raspbian.org jessie/main armhf Packages
Hit http://archive.raspberrypi.org jessie/ui Sources
Hit http://mirrordirector.raspbian.org jessie/contrib armhf Packages
Hit http://archive.raspberrypi.org jessie/main armhf Packages
Hit http://mirrordirector.raspbian.org jessie/non-free armhf Packages
Hit http://archive.raspberrypi.org jessie/ui armhf Packages
Hit http://mirrordirector.raspbian.org jessie/rpi armhf Packages
Ign http://archive.raspberrypi.org jessie/main Translation-en_GB
Ign http://archive.raspberrypi.org jessie/main Translation-en
Ign http://archive.raspberrypi.org jessie/ui Translation-en_GB
Ign http://archive.raspberrypi.org jessie/ui Translation-en
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en
Ign http://mirrordirector.raspbian.org jessie/main Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/main Translation-en
```

Raspbian Wheezy

Una vez listos, el sistema estará actualizado con todos los paquetes necesarios, entonces instala el primer programa con sólo agregar **install** y posteriormente el nombre del programa a instalar, siempre y cuando este nombre este dentro del repositorio de paquetes de Raspbian, pero por ahora no es necesario preocuparse por eso. Una vez más es necesario usar **sudo** para poder instalar un programa. Primero enlistará todos los paquetes necesarios para poder instalar el software y luego las dependencias o los programas necesarios para que funcione correctamente. Preguntará si estás de acuerdo, presiona “Y” para proceder.

```
pi@raspberrypi ~ $ sudo apt-get install puredata
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
 fonts-droid gem gem-doc gem-extra gem-plugin-gmerlin gem-plugin-lqt
 gem-plugin-magick gem-plugin-v4l2 ghostscript imagemagick-common
 liba52-0.7.4 libftgl2 libgavl1 libgmerlin-avdec1 libgs9 libgs9-common
 libijs-0.35 libjbig2dec0 liblqr-1-0 libmagick++-6.q16-5
 libmagickcore-6.q16-2 libmagickwand-6.q16-2 libmpeg2-4 libpaper-utils
 libpaper1 libpostproc52 libquicktime2 puredata-core puredata-dev
 puredata-doc puredata-extra puredata-gui puredata-utils tcl tcl8.6 tk tk8.6
Suggested packages:
 pd-zexy ttf-dejavu v4l2loopback-dkms v4l2loopback-modules ghostscript-x
 libmagickcore-6.q16-2-extra pd-csound pd-pdp pd-audio tcl-tclreadline
The following NEW packages will be installed:
 fonts-droid gem gem-doc gem-extra gem-plugin-gmerlin gem-plugin-lqt
 gem-plugin-magick gem-plugin-v4l2 ghostscript imagemagick-common
 liba52-0.7.4 libftgl2 libgavl1 libgmerlin-avdec1 libgs9 libgs9-common
 libijs-0.35 libjbig2dec0 liblqr-1-0 libmagick++-6.q16-5
 libmagickcore-6.q16-2 libmagickwand-6.q16-2 libmpeg2-4 libpaper-utils
 libpaper1 libpostproc52 libquicktime2 puredata puredata-core puredata-dev
 puredata-doc puredata-extra puredata-gui puredata-utils tcl tcl8.6 tk tk8.6
0 upgraded, 38 newly installed, 0 to remove and 1 not upgraded.
```

Ambiente gráfico

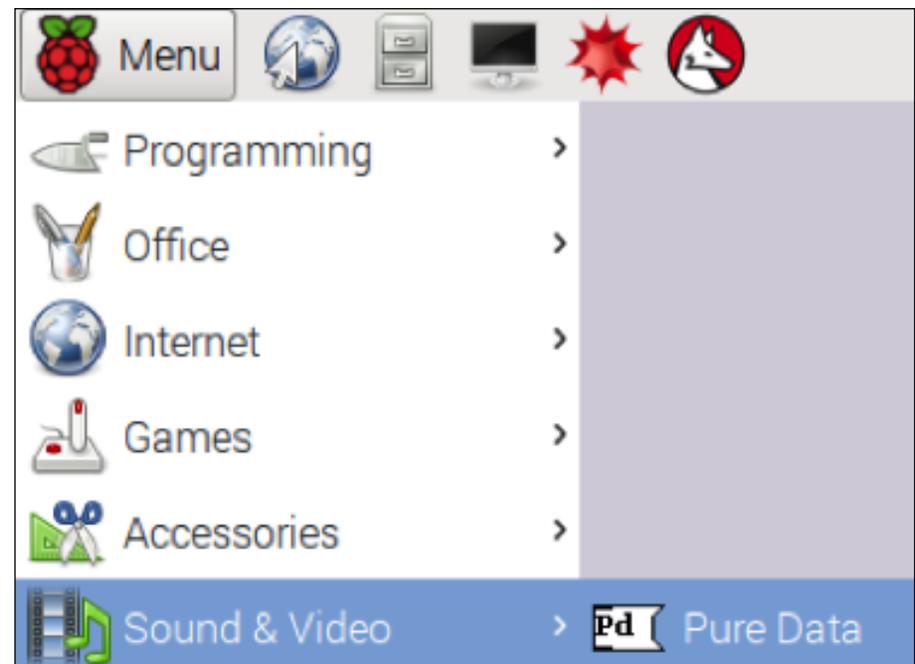
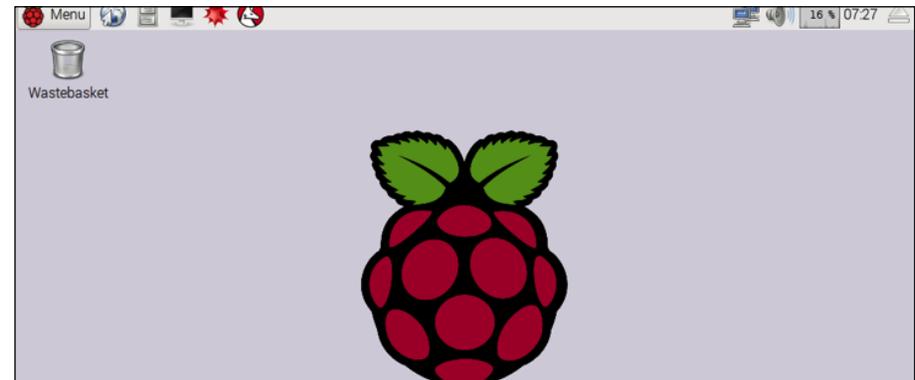
Es hora de pasar a la parte gráfica de este tutorial. Para esto escribe el comando **startx**.

Como puedes observar la Raspberry Pi posee un ambiente gráfico como otros sistemas operativos actuales, basado en ventanas.

Ya instalado Pure Data puedes pasar a la parte III de este tutorial, abriendo **menú** y yendo a la sección de **audio** y **video**.

También verás otros elementos como un **browser**, el **explorador de archivos**, la **terminal**, etc.

Puedes tomar un segundo para buscar entre las diferentes pestañas con los programas preinstalados en esta distribución de Raspbian.



NOOBS

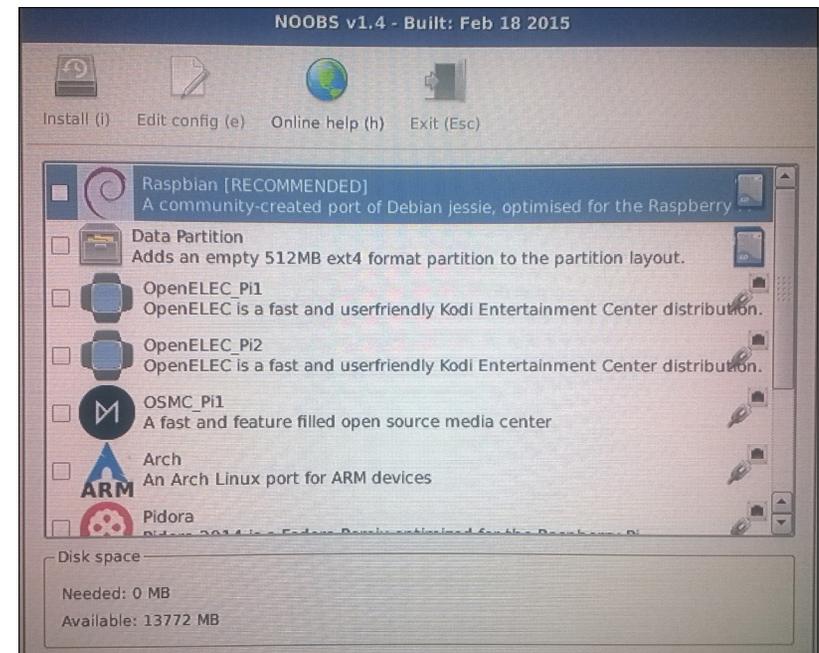
En el caso de haber escogido la instalación mediante NOOBS, el proceso se simplificará mucho ya que es la forma más sencilla de instalar, no sólo un sistema operativo, sino que da la opción de instalar la mayoría de las distribuciones que existen.

Después de conectar la Raspberry Pi, como se menciona en al principio de esta parte II, aparecerá esta pantalla donde nos preguntará que sistema operativo deseas usar.

El sistema que debes escoger es el de Raspbian ya que viene precargado en la SD que creaste en la parte I.

Elige la opción de **instalar** y espera. El proceso dura aproximadamente 20 minutos, e irá dando información sobre las utilidades del sistema operativo así como de la terminal.

Al finalizar aparecerá esta pantalla y después de unos minutos cargará el ambiente gráfico del sistema operativo. De lo contrario, aparecerá la terminal donde hay que escribir el comando: **startx**. Abre una terminal y sigue los pasos descritos en la Parte II con Raspbian Wheezy, ya que la terminal es un programa universal en estos sistemas, e instala Pure Data.



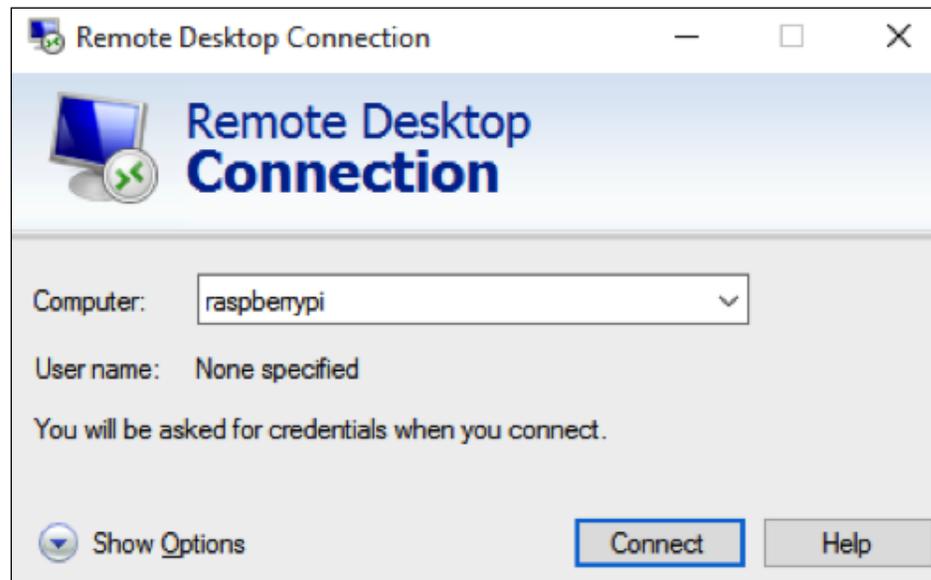
Remote Desktop Connection

Remote Desktop (opcional)

Una de las soluciones para poder acceder a Raspberry Pi sin tener mouse, teclado o monitor HDMI es el **protocolo RDP de escritorio remoto**, desarrollado por Microsoft. Lo más importante es tener una computadora y estar conectado a la misma red Wi-Fi que el Raspberry Pi (El Raspberry Pi debe estar conectado a la red vía Ethernet, mouse, teclado o pantalla HDMI la primera vez que se configura esta modalidad). Usa los comandos aprendidos para instalar el programa dentro de la Raspberry Pi, para esto será necesario abrir una terminal y escribir lo siguiente: **sudo apt-get install xrdp**.

Una vez instalado empezará a correr el servidor de **escritorio remoto** en nuestra Raspberry Pi.

Si tu computadora es Windows abre el **Remote desktop connection** (conexión de escritorio remoto), el cual está preinstalado en Windows 7, 8 y 10. En caso de tener una mac puedes instalar un cliente de RDP: <http://cord.sourceforge.net/>



Remote Desktop Connection

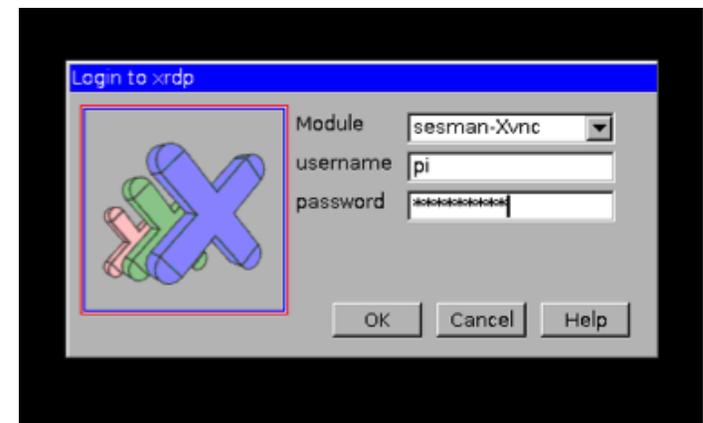
Escribe el nombre del Raspberry (el default es **raspberrypi**, de lo contrario puedes consultarlo con el comando **hostname** en la terminal del Raspberry) y conectate.

En caso de no lograr la conexión entra a Raspberry Pi y escribe el comando **ifconfig** en la terminal. En vez de escribir el nombre de **raspberrypi** escribe el número subrayado en rojo, el cual corresponde a la IP del raspberrypi.

```
pi@raspberrypiuno ~ $ ifconfig
eth0    Link encap:Ethernet  HWaddr b8:27:eb:7f:2f:1a
        inet addr:120.120.4.70  Bcast:120.120.7.255  Mask:255.255.248.0
        inet6 addr: fe80::ba27:ebff:fe7f:2f1a/64  Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:11149  errors:0  dropped:65  overruns:0  frame:0
        TX packets:1601  errors:0  dropped:0  overruns:0  carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:818608 (799.4 KiB)  TX bytes:691193 (674.9 KiB)
```

Se abrirá esta pantalla donde pondremos nuestras credenciales para el usuario de raspberry, deben de ser las que están en default, a menos que se hayan creado más usuarios con privilegios suficientes para permitir el escritorio remoto.

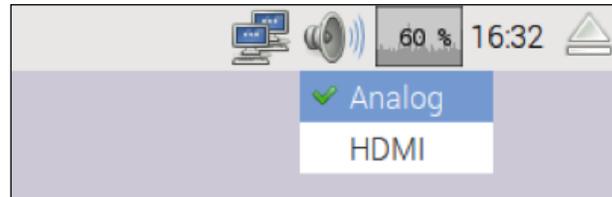
En este punto podrías controlar la raspberry remotamente desde nuestra pc o mac, usando el teclado, monitor, mouse o track pad de nuestra computadora mientras que el Raspberry podrá estar simplemente conectado al voltaje y a la red.



Herramientas de configuración

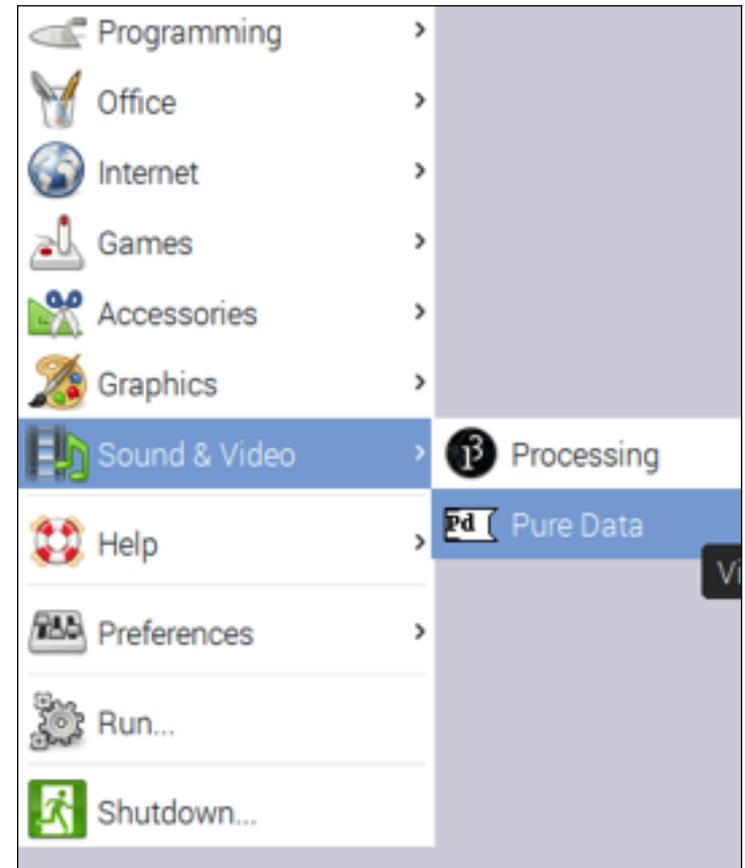
Lo primero que necesitas es configurar la salida de audio para que use la salida analógica de 3.5 mm. Para esto da click derecho en el icono de la bocina y selecciona **Analog**.

Ahora aprenderás a usar el software de Pure Data. Ve al menú, luego en la pestaña de audio y video y encontrarás el programa instalado.



Al abrirlo por primera vez verás la consola de Pure Data donde el sistema podrá darte información relevante sobre el parche que este corriendo actualmente así como los drivers de sonido y las conexiones de entrada y salida. Pure Data tiene diferentes versiones, la versión actual será la de Vanilla. Vanilla fue la primera versión de Pure Data creada por Puckette Miller la cual tiene cierto número de librerías y objetos que puedes usar.

Posteriormente con la contribución de muchos entusiastas de este proyecto se publicó la versión Pd-Extended. La cual contiene un número mayor de librerías y objetos. Como su nombre lo indica, extiende las posibilidades de procesamiento y conexión para Pure Data. En este caso solo será necesario usar las librerías básicas ya que este tutorial es introductorio. (ver Apéndice)



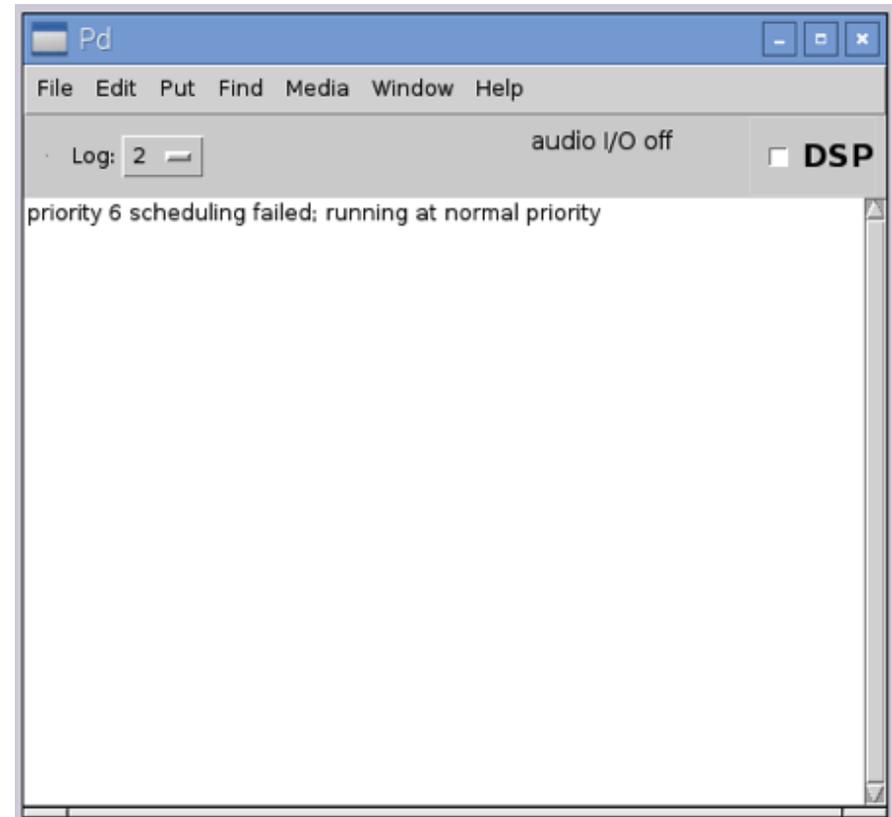
Herramientas de configuración

Una vez logrado esto crea una nueva ventana con **ctrl + n** o en la pestaña de **File/New**.

Se abre un lienzo blanco, totalmente vacío. Entonces puedes empezar a crear nuevos elementos en el parche.

Primero haz un mensaje con **ctrl + 2** o en la pestaña de **Put/Message**.

Escribe un mensaje, por ejemplo: "Hola Mundo".



Hola Mundo

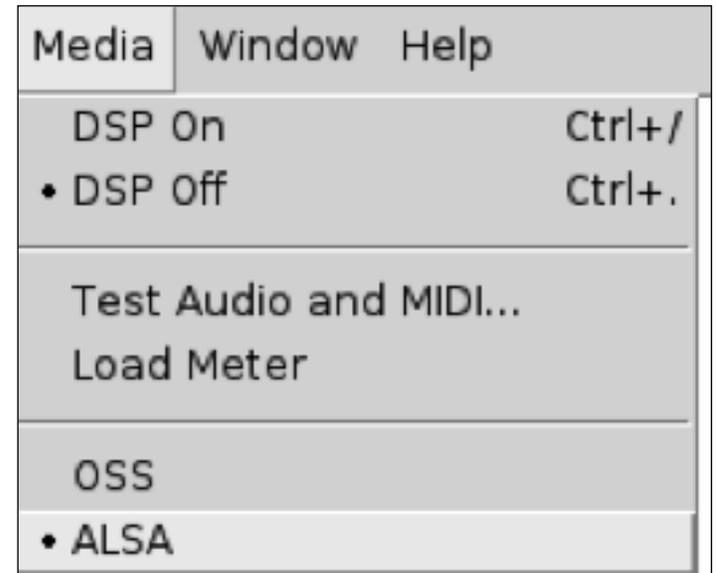
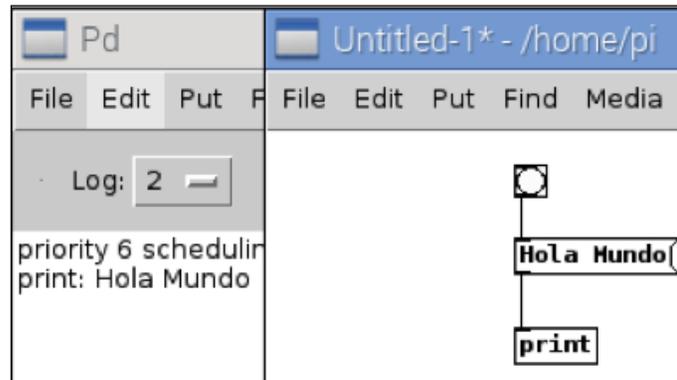
ALSA y ALSA MIDI

Los mensajes te permiten pasar instrucciones a los objetos. En este caso la instrucción es un texto que no tiene significado concreto más allá de una cadena de caracteres.

Para poder hacer que este mensaje haga algo haz un objeto con **ctrl + 1** o en la pestaña de **Put/Object**. Dentro de nuestro nuevo objeto escribiremos **"print"**. Este objeto nos permitirá imprimir información relevante en la consola de Pure Data.

Una vez logrado esto genera un bang con **ctrl + shift + b** o en la pestaña de **Put** lo encontraremos como **Bang**. Y como puedes suponer, debes conectar todos estos elementos con el mouse.

Pero antes de todo, debes salir del modo de edición con **ctrl + e** o en la pestaña de **Edit** deshabilitar el **Edit Mode**. Ahora da click al bang y observa la consola de Pure Data.



El objeto bang es indispensable para poder programar en Pure Data, ya que los bangs dominan el flujo de las instrucciones en los parches. Debido a que los objetos en la mayoría de los casos necesitan de un inicializador para empezar a funcionar y la forma de interactuar entre ellos es mediante bangs, los cuales pueden ser visibles o invisibles.

Ahora vas a crear un parche más complejo e interesante. Para asegurar que Pure Data suene, conecta unos audífonos o unas bocinas en la salida de 3.5mm. Pure Data se inicializa con los drivers de sonido que tienen disponibles, pero es recomendable usar los drivers de ALSA. Esto lo puedes cambiar en la pestaña de Media donde seleccionarás ALSA como dispositivo principal, así como ALSA-MIDI para cualquier conexión MIDI.

ALSA y ALSA MIDI

Una vez configurado esto crearemos un objeto nuevo **ctrl + 1**, y escribiremos **osc~**. Este objeto será diferente a los demás ya que es un objeto de sonido, el cual no necesita un bang para funcionar sino que constantemente está mandando una señal con los argumentos dados. Da doble click sobre el objeto y le dale un argumento de 440 precedido por un espacio.

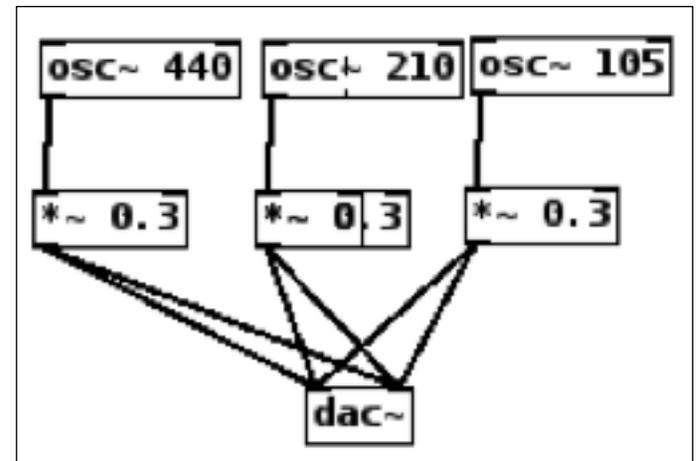
Uno de los elementos clave para cualquier aplicación de audio es la amplitud. Normalmente dentro de los softwares de edición de audio la representación de este valor será mediante un valor flotante con un máximo de 1 (0.0, 0.1, 0.2, ..., 1.0), ya que a partir del 1.0 empezarás a escuchar un sonido con clip digital*, el cual es sumamente molesto. Es por esto que debes decirle al objeto **osc~** que te dé un valor específico de amplitud. Esto lo lograrás mediante la multiplicación del objeto **osc~** por un valor flotante de 0.5.

Para esto genera un objeto ***~** con un argumento 0.5. Lo conectas a la salida del objeto **osc~**. Esto es en realidad una señal a 440 Hz creada mediante un oscilador y multiplicada por 0.5, lo cual es igual a decir que tendremos una amplitud de $\frac{1}{2}$ de la señal original.

Ahora, para hacerlo sonar debes crear un objeto **dac~**, el cual es un convertidor digital-analógico que simula la conversión que hará la computadora a continuación para lograr hacer sonar este oscilador. Deberás conectarlo como se ve en la figura. El último paso para hacer sonar el parche será prender el DSP en la ventana donde está la consola Pure Data. No olvides que el modo edición se puede habilitar y deshabilitar con **ctrl + e**. Probablemente te dará un buen susto ya que los sonidos puros, creados por los osciladores digitales suelen ser sumamente molestos. Prueba bajando el volumen a 0.3 para lograr escuchar más cómodamente.

Si esto fuera todo sería muy aburrido; hagámoslo más interesante. Selecciona el objeto **osc~** y ***~**. Duplicalo 2 veces con **ctrl + d** y cambia los valores de los osciladores a múltiplos de esa frecuencia. Cambia todas las amplitudes a 0.3.

No olvides conectar las salidas de los objetos ***~** a las dos entradas del **dac~** (canal izquierdo y canal derecho). Probablemente es muy molesto no poder apagarlo, entonces haz un control de volumen. Para esto necesitas dos nuevos objetos. El objeto **line~** y el objeto **vslider**.



NOTA

*Elimina las conexiones entre los ***~** y el **dac~**.

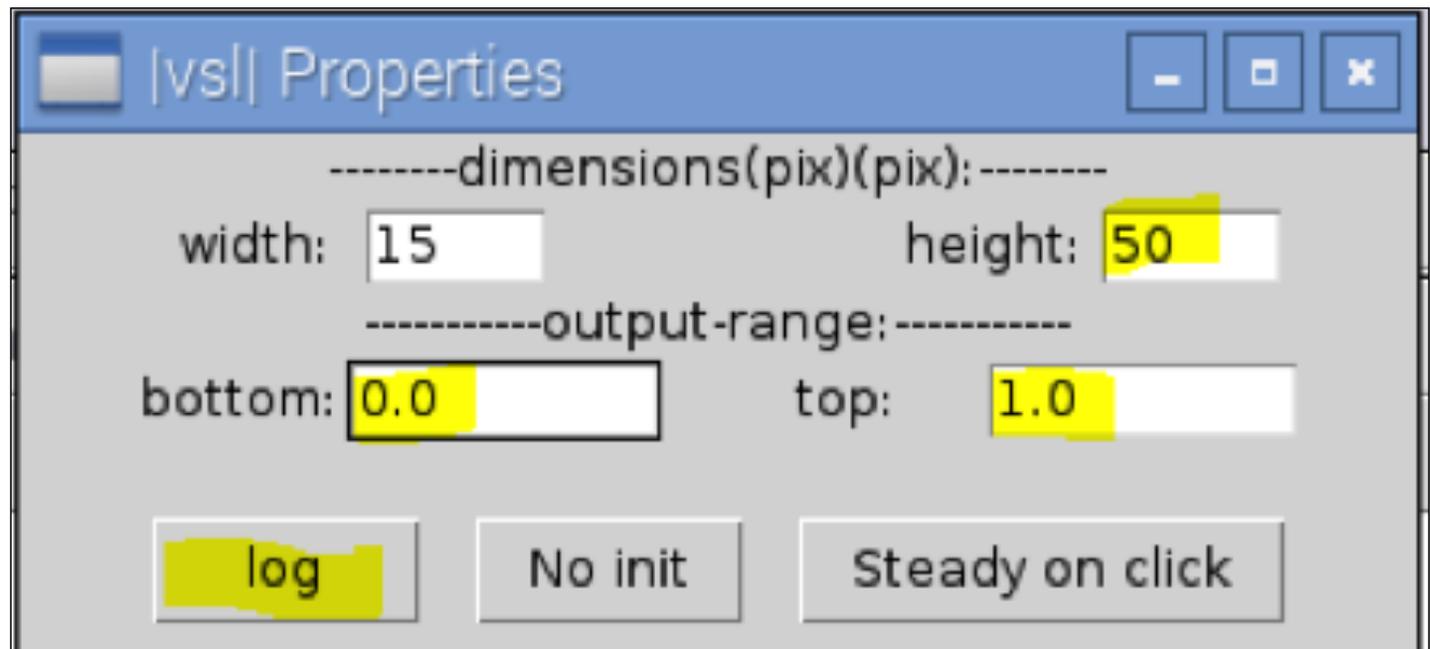
ALSA y ALSA MIDI

Todos los objetos tienen propiedades que pueden ser cambiadas dándole click derecho al objeto. Cambia algunas propiedades del **vslider**.

Cambia la altura y los valores mínimos y máximos del **slider**. El tipo deberá ser **log** para poder controlar más efectivamente la amplitud.

Crea un nuevo objeto con **ctrl + 3**; este objeto de número también será clave para poder programar con Pure Data. Servirá para poder visualizar y cambiar parámetros de número en tiempo real.

Conéctalo al **vslider** y el número al objeto **line~**.

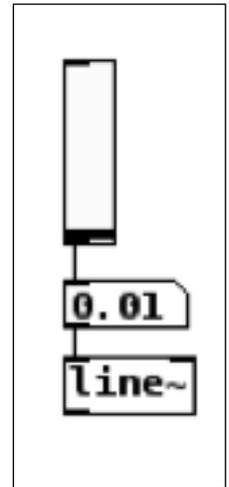


ALSA y ALSA MIDI

Los objetos siempre tienen *inlets* (entradas) y *outlets* (salidas). Para los objetos comunes normalmente existe un **hot inlet**, normalmente será el que este más a la izquierda, el cual detonara la función del objeto mientras que el **cold inlet**, el que este más a la derecha, nos servirá en algunos casos para cambiar argumentos del objeto. Esto será cierto en la mayoría de los objetos básicos, para los objetos más complejos puedes, consultar qué función tiene el **inlet** dándole click derecho y **help**.

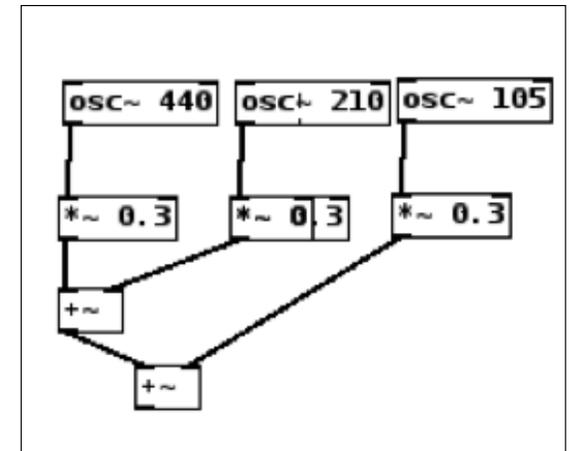
Has creado satisfactoriamente el control de volumen. Pero antes de probarlo debes hacer algo con los osciladores. Para poder lograr que dos señales se sumen es necesario representarlas mediante una gráfica o una función matemática. Ya sea que se logre esto sumando gráficamente o matemáticamente las señales, como vimos anteriormente, en Pure Data puedes hacer operaciones matemáticas de señales como $\ast\sim$. Entonces de igual forma podrás hacer un objeto de suma $\ast\sim$. Basta con poner una \sim para especificar que se trata de una señal de audio. Sabiendo esto entonces suma las 3 señales mediante objetos $\ast\sim$.

Es importante saber que debido a que la máxima amplitud siempre será 1.0, debes limitar las señales que estas sumando para que nunca excedan ese 1.0, es por eso que cada una de ellas se multiplican por 0.3.



El objeto $\ast\sim$ solo tiene dos *inlets* para sumar, por eso deberás usar dos objetos para poder sumar las 3 señales. Antes de conectar esto a la salida, lo conectarás al control de volumen. Y de ahí a la salida **dac~** mediante otro objeto $\ast\sim$.

Esta primera parte del parche es una síntesis aditiva de tres señales con un control de volumen. Notarás que el sonido es ligeramente más complejo que los primeros sonidos que lograste obtener y probablemente escucharás un *hiss*, pero esto se debe a que el hardware no procesa tan efectivamente el sonido. Es posible comprar módulos de sonido para la Raspberry para obtener mejor desempeño, pero por ahora no será necesario. Ahora crea otro parche encargado de controlar los números que entran a los osciladores para poder así crear diferentes sonidos.



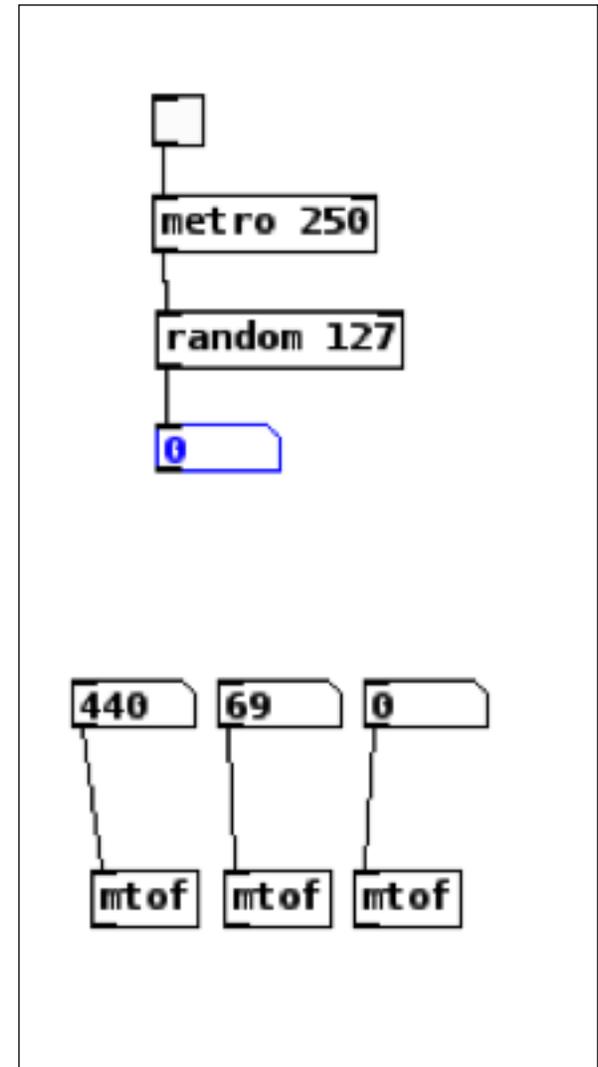
ALSA y ALSA MIDI

Empezarás creando nuevos objetos: harás un objeto **metro** con un **argumento 250**. El objeto **metro** es simplemente un metrónomo, el cual mandará **bangs** a una cierta velocidad por su argumento o su *cold inlet*. Este objeto necesita de un **switch/toggle** para ser inicializado. Después harás un objeto **random** con un **argumento 127**. El objeto random servirá para generar un número aleatorio entre 0 y su argumento principal. En este caso 127 será el máximo número que este objeto podrá crear. El objeto random necesitará **bangs** continuos para poder hacer diferentes números. A continuación harás un objeto **toggle** con **ctrl + t**. El objeto toggle funcionará como un switch dando un **on/off** o un **0/1** dependiendo de si está clickeado o no. Por último haremos un número con **ctrl + 3**.

Conectarás todos los objetos y podrás ver que hace el nuevo parche al darle click, a el toggle, para esto recuerda salir del modo edición (**ctrl + e**).

Se preguntaran el porqué del 127, no desesperen. Una de las formas más sencillas de representar sonidos digitalmente con una escala musical temperada como la conocemos hoy en día (do, re, mi, fa, sol, la, si) es mediante un código o protocolo conocido como MIDI. Entonces para cada nota tendrás un equivalente en número por ejemplo para la nota La en el índice acústico 4 (índice acústico = altura de la escala) equivale a la nota MIDI 69, por lo que sol equivaldrá a 70 (cada 1 equivale a medio tono más arriba, por lo que sumarle dos equivaldría a 1 tono completo). Existen un máximo de 127 notas MIDI las cuales representan casi la totalidad de las posibles alturas que puede haber en un instrumento musical.

Sabiendo esto podemos seguir en nuestro parche hagamos 3 números y 3 objetos "mtof". Este objeto realizará la conversión de nota MIDI a frecuencia ya que nuestros osciladores solo entienden frecuencias como argumentos y no notas MIDI.



ALSA y ALSA MIDI

Ahora realizaremos algunas operaciones con las notas MIDI demostrando la practicidad del uso de éstas en vez de frecuencias.

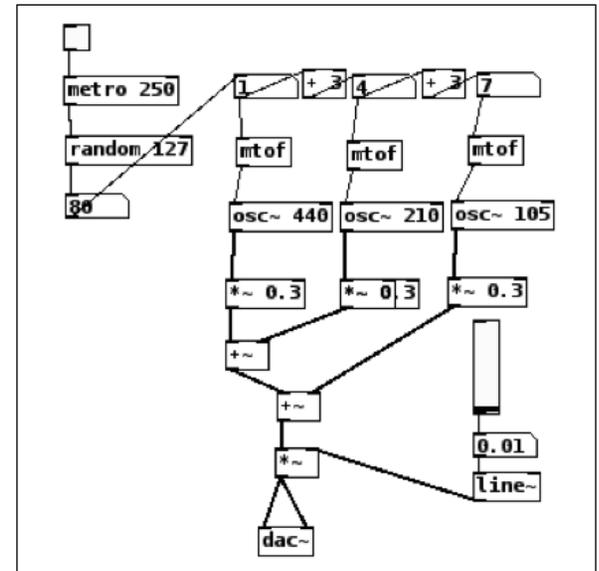
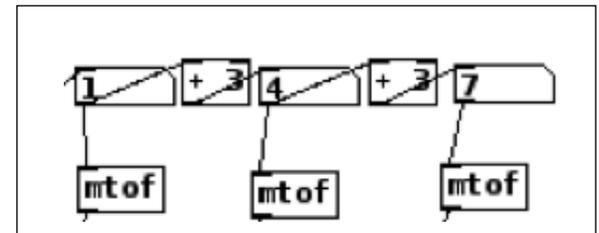
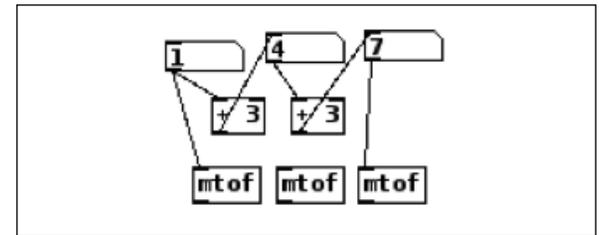
A los dos últimos números le sumaremos 3 con un objeto de +. Los cuales tomarán, como primer argumento, el número anterior.

La idea es crear sonidos con diferentes timbres y sonoridades. Les recomiendo reacomodar su parche de la siguiente forma. Finalmente debes conectar el generador de números aleatorios a la primera caja de número, y las salidas de nuestros convertidores de MIDI a frecuencia las conectaremos a las entradas de los osciladores.

Como podemos ver, programar en Pure Data se vuelve una tarea sumamente intuitiva y gráfica. Basta con entender algunos conceptos básicos de sonido y música para empezar a crear tus propios parches.

Hagamos este parche aún más interesante. Crea 4 mensajes (**ctrl + 2**) con los números **2, 3, 4, 6** y los conectaremos a los dos *cold inlets* de nuestros objetos de suma.

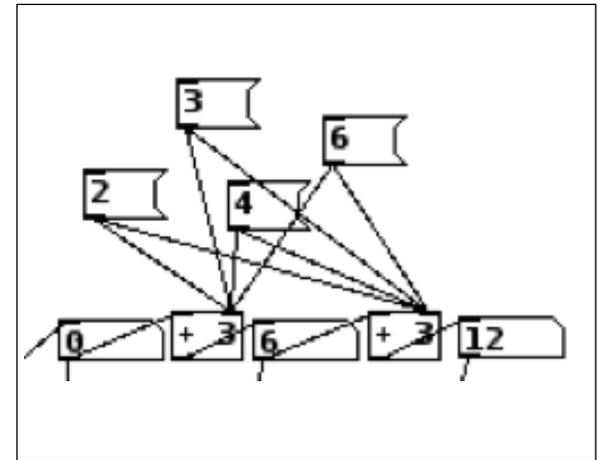
Esto provocará que al darle click o mandarle un bang al mensaje con los números, aumentará o disminuirá la separación entre las 3 señales de los osciladores, cambiando así el timbre del sonido.



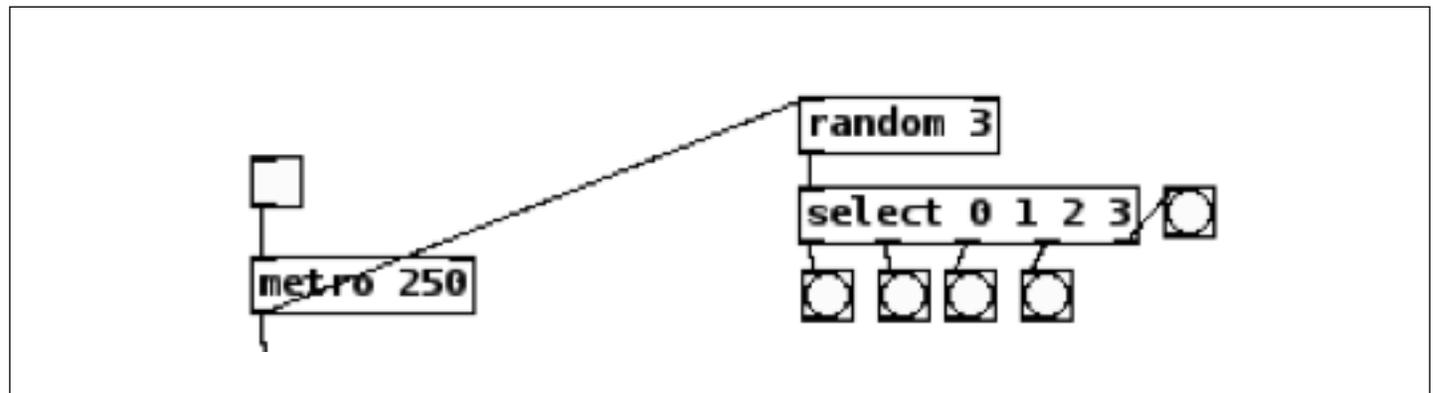
ALSA y ALSA MIDI

Ahora puedes comprobar cómo el sonido cambiará al darle click a los diferentes mensajes creados.

A continuación crearemos un objeto de **random** con un **argumento 3**. Después usaremos un nuevo objeto **select** con argumentos **0 1 2 3**. Se debe dejar un espacio entre cada argumento. Este objeto seleccionará y mandará un bang por cada vez que se iguale uno de los argumentos con los números en el primer *inlet*. Por cada uno de los outlets del objeto **select** conectaremos un bang (**ctrl + shift + b**) y conectaremos la salida del **random** a la entrada del **select** la entrada del **random** al objeto **metro** que ya habíamos creado anteriormente.

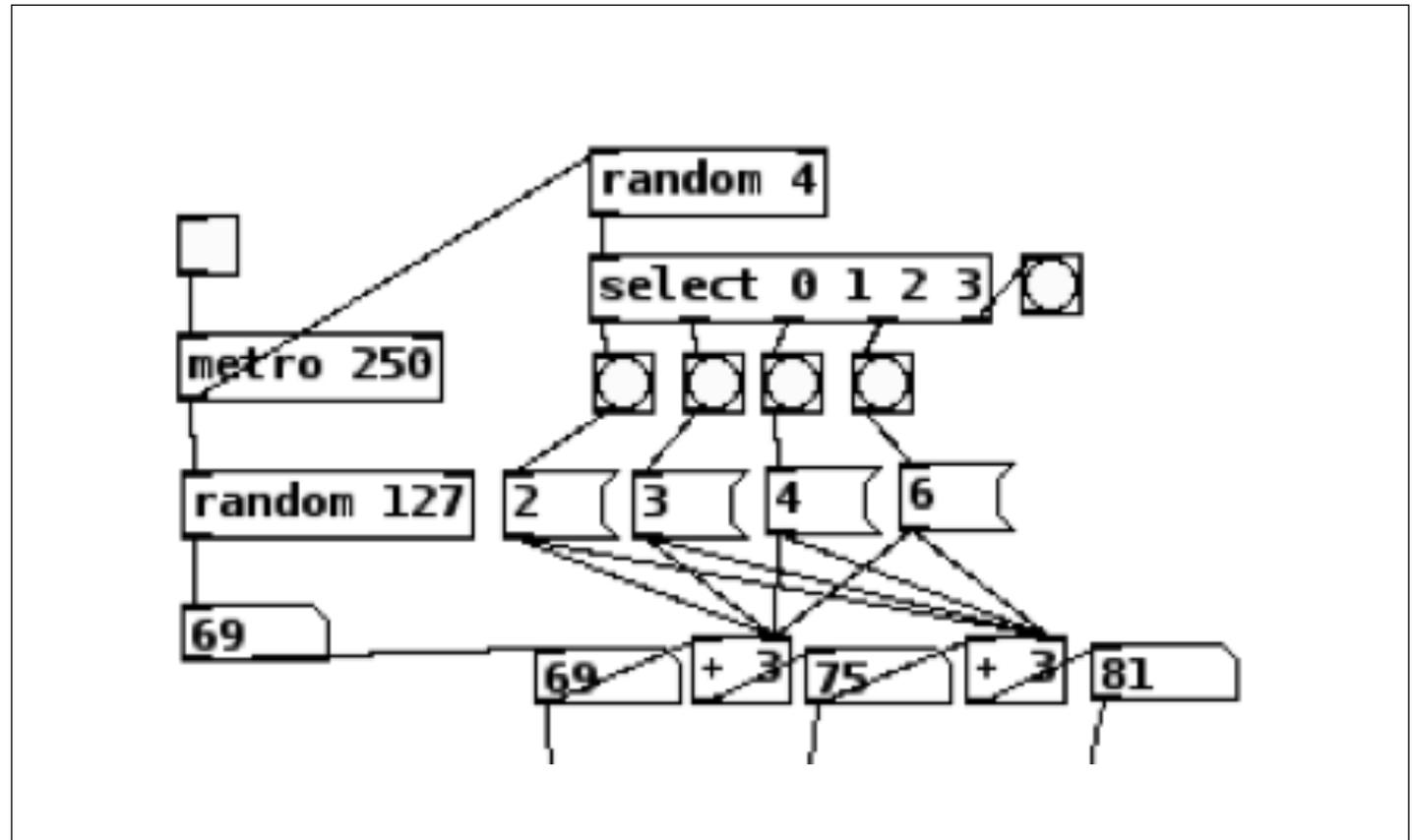


Si activamos el **toggle** podremos ver como los bangs se activan de forma aleatoria. Para asegurar un poco más de aleatoriedad cambiaremos el argumento de **random** a **4** ya que el objeto **select** al tener un número fuera de sus valores esperados (sus argumentos), mandará un pulso al último bang, el cual no corresponde a ningún argumento. También conectaremos los 4 primeros bangs a los mensajes que habíamos creado anteriormente.



ALSA y ALSA MIDI

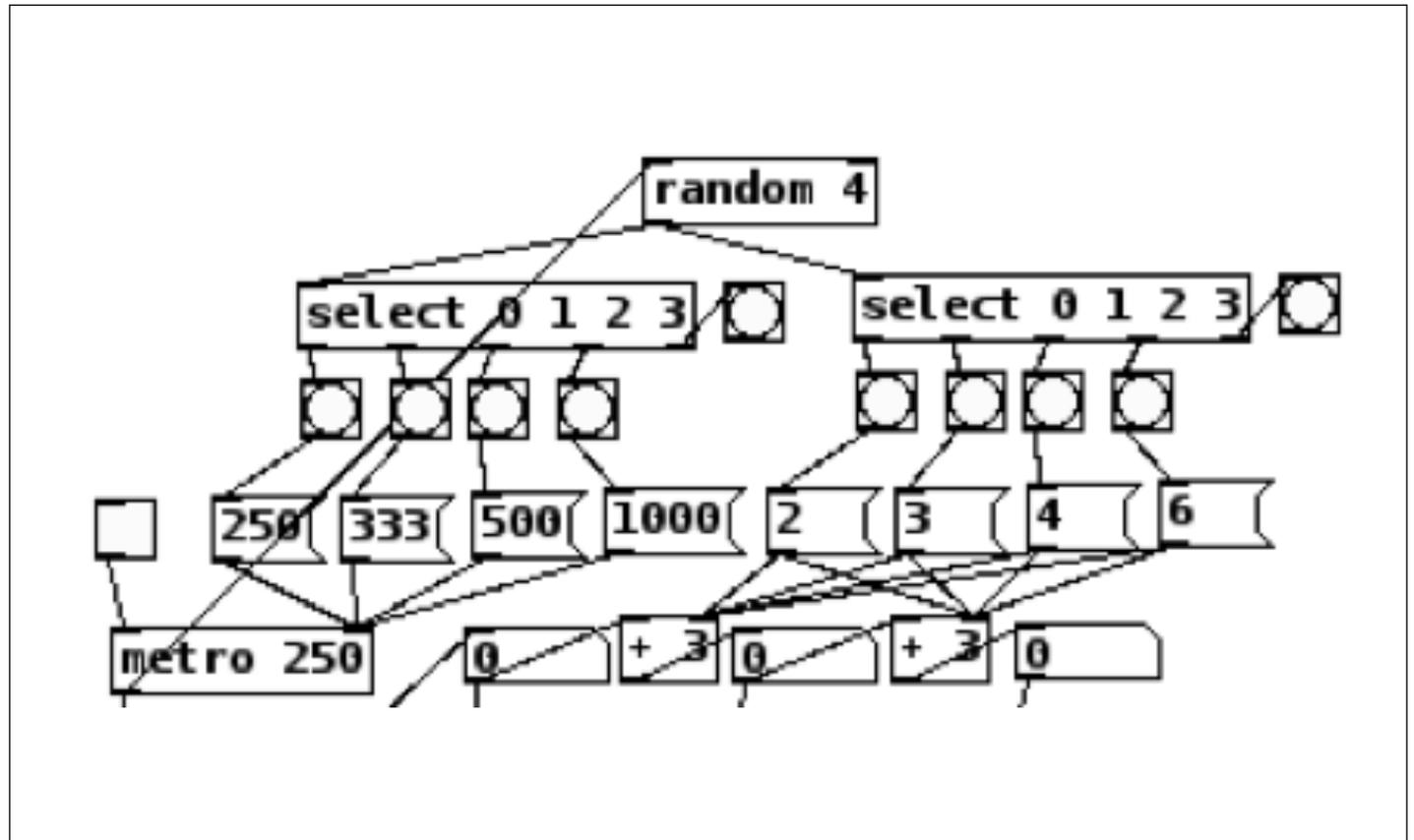
Al inicializar el parche con el **toggle** notarás que ahora el cambio en las distancias de las notas cambiará dinámicamente. Para demostrar la versatilidad de este estilo de programación en ambientes gráficos crearemos una variación rítmica para el parche. Selecciona el objeto **select**, los **bangs** y los **mensajes** para duplicarlos. Sustituyendo los números **2, 3, 4 y 6** de los mensajes por **250, 333, 500 y 1000**, conecta todos los *outlets* de estos mensajes al *cold inlet* del objeto **metro**. Mientras que el *inlet* de nuestro **select** duplicado se conectará al *outlet* de nuestro **random** como podemos ver a continuación.



ALSA y ALSA MIDI

Esta simple modificación logró hacer que nuestro parche no sólo cambie dinámicamente su timbre, sino que de igual forma cambie la duración de las notas aleatoriamente.

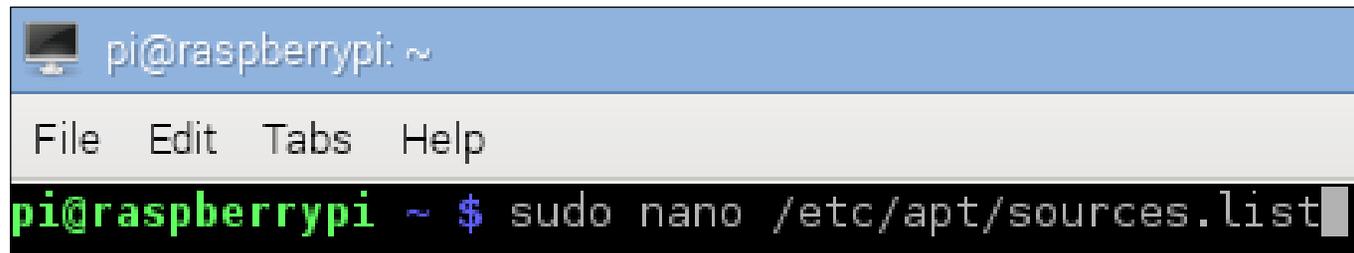
Este parche finalmente te permitirá entender las formas básicas de este lenguaje de programación con entorno gráfico así como las ventajas de este tipo de software. El aspecto final de nuestro parche será este.



Apéndice

Aquí aprenderás cómo instalar Pure Data Extended desde la terminal de Raspberry y hacer que el parche funcione de forma *headless* sin necesidad de hacer nada más que conectarlo a la alimentación de 5v y a la salida de audio de 3.5 mm.

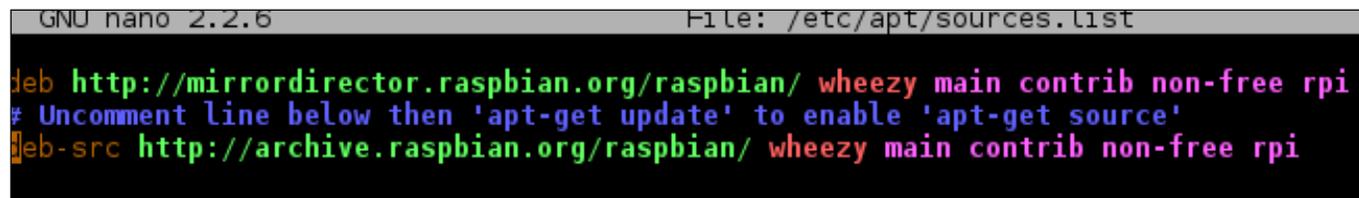
Antes que nada prende el Raspberry Pi y conéctalo al monitor o usando el servidor de acceso remoto **xrdp** (ver parte II). Abre una terminal, ve al directorio raíz usando **cd.** y ejecuta los siguientes comandos.



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi ~ $ sudo nano /etc/apt/sources.list
```

Este comando ejecutará un programa de edición de texto llamado **nano** dentro de nuestra terminal, el cual abrirá un archivo para poder añadir un nuevo repositorio. En **Linux**, los repositorios sirven como archivadores donde el sistema operativo busca los paquetes para instalar programas, ahí están los datos de todos los programas que puedes instalar en el Raspberry Pi.

Agrega o quita el signo de **#** de la línea que se puede ver en la pantalla. Al terminar presiona **ctrl + o**, te preguntará si deseas hacer los cambios y pondrás una "y". Luego presiona **ctrl + x** para salir del editor de texto **nano**.



```
GNU nano 2.2.6 File: /etc/apt/sources.list  
deb http://mirrordirector.raspbian.org/raspbian/ wheezy main contrib non-free rpi  
# Uncomment line below then 'apt-get update' to enable 'apt-get source'  
deb-src http://archive.raspbian.org/raspbian/ wheezy main contrib non-free rpi
```

Apéndice

Teclea el siguiente comando:

```
sudo wget https://puredata.info/downloads/pd-extended-0-43-3-on-raspberry-pi-raspbian-wheezy-armhf/releases/1.0/  
Pd-0.43.3-extended-20121004.deb
```

El comando **wget** te permitirá bajar el paquete que contiene los datos para instalar un programa. Para poder abrir el paquete usaremos el comando **dpkg**.

```
pi@raspberrypi ~ $ sudo dpkg -i Pd-0.43.3-extended-20121004.deb
```

Una vez instalado, procederemos al paso más importante. Es necesario darle los permisos de ejecución más elevados a este programa, ya que para poder abrir los **drivers** de sonido y usar una prioridad de procesamiento en tiempo real es necesario notificar al sistema.

```
pi@raspberrypi ~ $ sudo apt-get -f install
```

Con esto tendremos nuestro programa de Pure Data Extended instalado.

```
pi@raspberrypi ~ $ sudo chmod 4755 /usr/bin/pd-extended
```

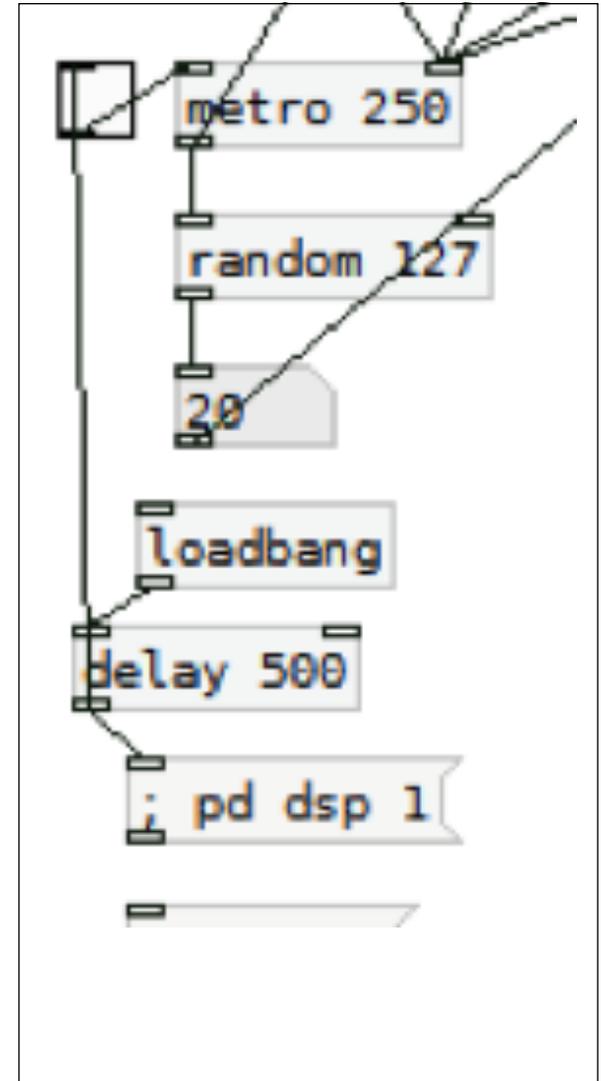
Apéndice

Para comenzar, necesitas abrir el parche de Pure Data o Pure Data Extended, donde generarás dos nuevos objetos **loadbang** y **delay**. Estos objetos serán de suma utilidad, ya que el **loadbang** mandará un bang al abrir el parche, lo cual permite inicializar todas aquellas partes del parche que necesiten ser inicializadas. También generarás un mensaje con el siguiente contenido **; dsp 1**.

También usaremos el objeto **delay** que, tras ponerle un argumento de 500 esperará 500 ms para mandar un bang. Esto es para darle tiempo al sistema para inicializar los drivers de sonido antes de mandar el mensaje. Este mensaje es de suma importancia ya que al ponerle ";" como prefijo estaremos mandando instrucciones directamente al software de Pure Data. En este caso mandaremos a que prenda la salida de audio o el DSP.

Haz las conexiones como se muestra en el parche de la derecha donde el **loadbang** ira conectado al *inlet* del objeto **delay**, mientras que del objeto **delay** tendremos conectados el mensaje a Pure Data y el toggle del objeto metro.

Una vez terminado guarda el parche como **ejemplo1.pd** en el escritorio (Desktop), cierra Pure Data y abre una terminal.



Apéndice

Usa el comando **pwd** para saber en que directorio te encuentras, busca el directorio **/home/pi**. Si no estuvieras en ese directorio usa **cd /home/bin** para ir a ese directorio. A continuación crea una carpeta **bin** con el comando **mkdir** y muévete a ella con el comando **cd**.

```
pi@raspberrypi ~ $ pwd
/home/pi
pi@raspberrypi ~ $ cd /home/pi
pi@raspberrypi ~ $ mkdir ./bin
pi@raspberrypi ~ $ cd /bin
pi@raspberrypi /bin $ █
```

Creas un archivo de texto en blanco con el editor de texto **nano**, esto abrirá un archivo en blanco.

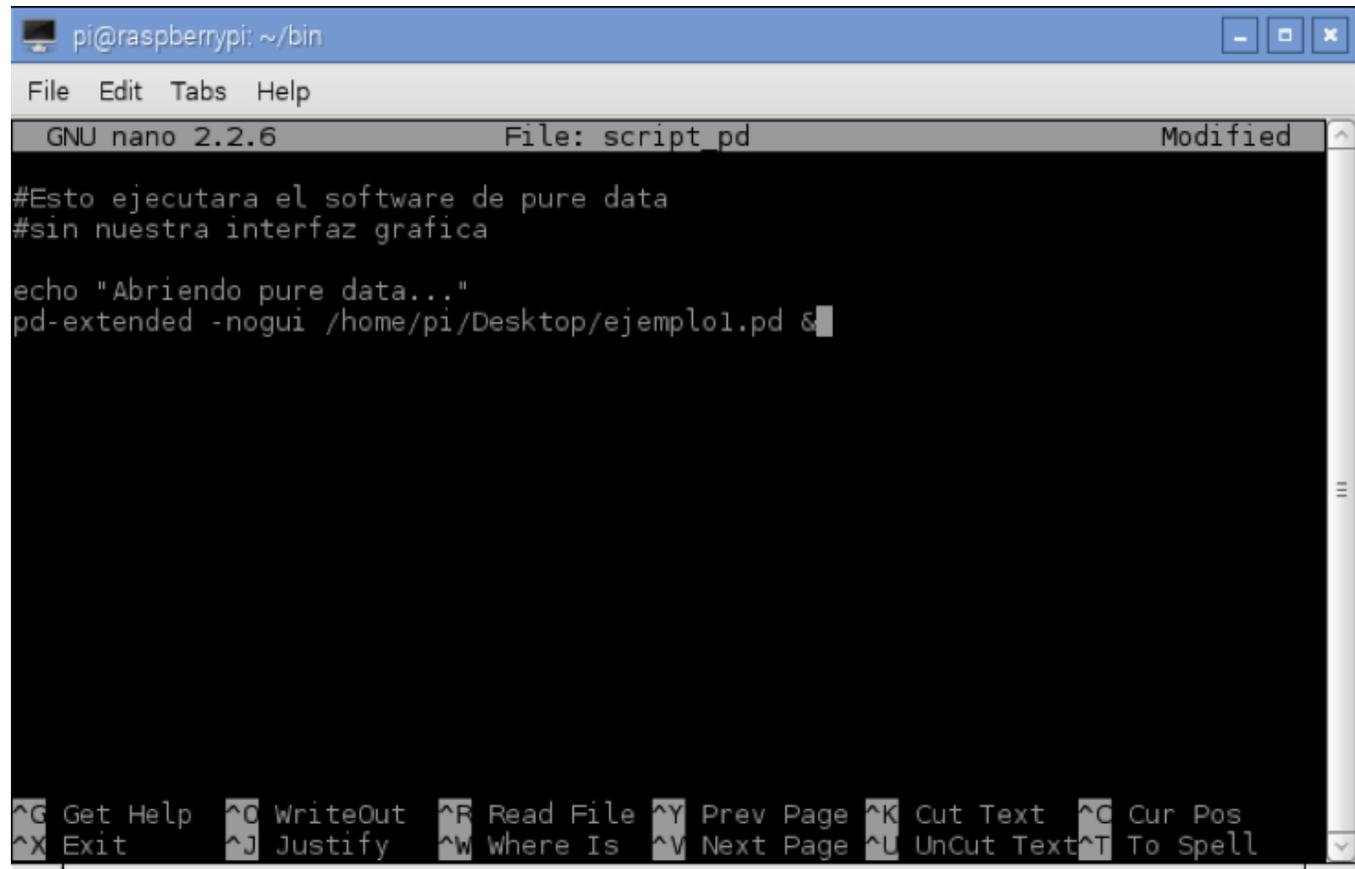
```
pi@raspberrypi /bin $ sudo nano script_pd
```

Apéndice

Guarda y sal del editor con **ctrl + o** y **ctrl + x**

En caso de estar usando Pure Data en vez de Pure Data Extended debes sustituir **pd-extended** con **pd**.

Para lograr que nuestro *script* sea ejecutable será necesario escribir este comando.



```
pi@raspberrypi: ~/bin
File Edit Tabs Help
GNU nano 2.2.6 File: script pd Modified
#Esto ejecutara el software de pure data
#sin nuestra interfaz grafica

echo "Abriendo pure data..."
pd-extended -nogui /home/pi/Desktop/ejemplo1.pd &

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Apéndice

Para comprobar que nuestro *script* funcione escribe en la terminal **script_pd** y verás cómo la terminal empieza a inicializar el software de Pure Data sin interfaz gráfica y luego escucharás el parche funcionando.

```
pi@raspberrypi ~/bin $ sudo chmod 755 script_pd
```

Escribire la línea justo después de los comentarios (**#**). Guarda y cierra.

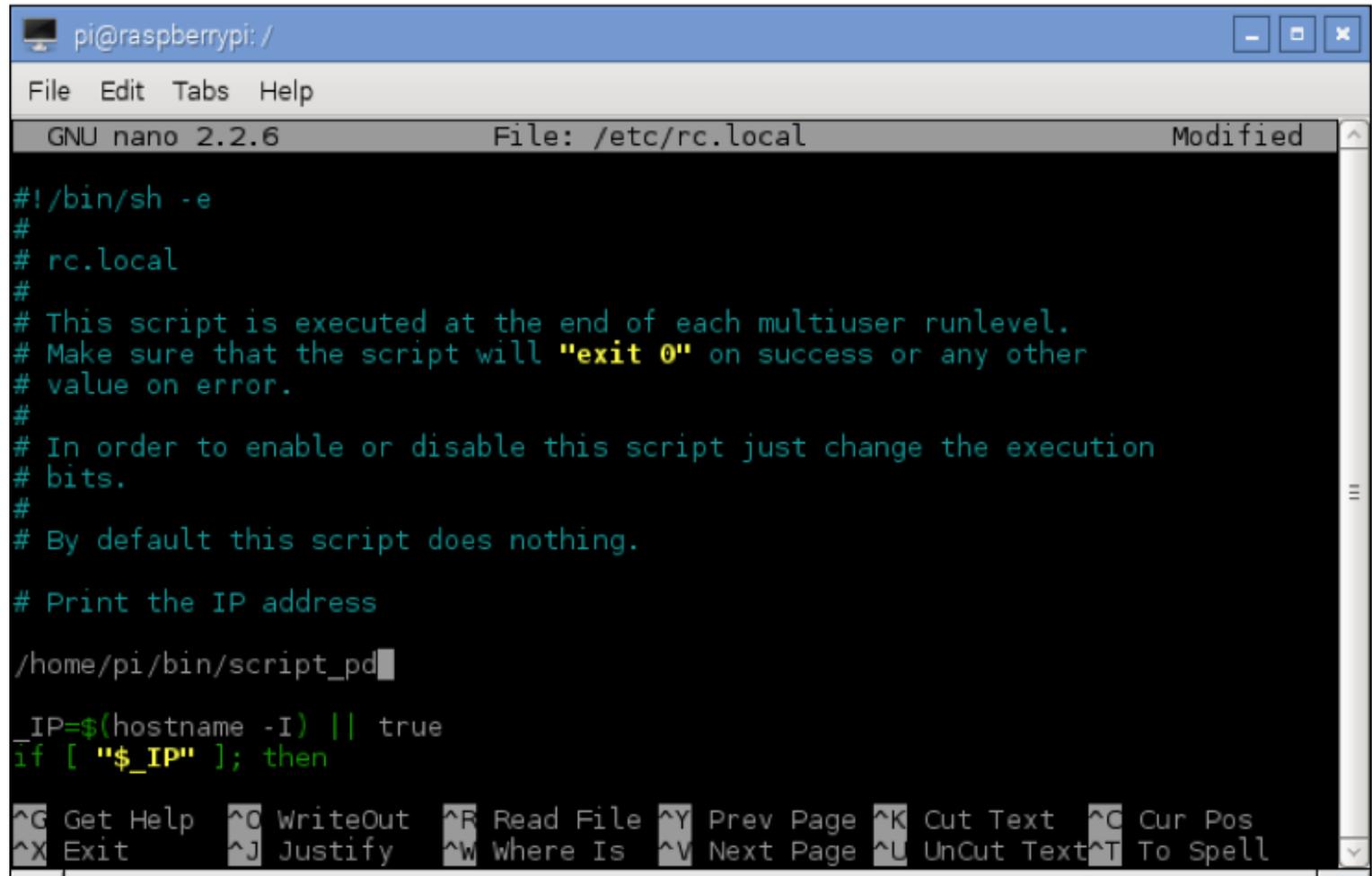
Esta última instrucción provocará que al reiniciar la Raspberry Pi automáticamente se ejecutará el **script** donde mandarás a abrir el parche de Pure Data sin interfaz gráfica, y se prenderá la salida de sonido sin necesidad de más interacción por parte del usuario.

Para poder detener la ejecución del parche abre una terminal y teclea **sudo killall puredata** o **sudo killall pd-extended** en caso de estar usando Pure Data Extended.

```
pi@raspberrypi / $ sudo nano /etc/rc.local
```

Esto finaliza completamente nuestro tutorial, en este punto no necesitaremos más que el cable de alimentación y unos audífonos o bocinas para poder usar nuestro generador de sonidos aleatorios.

Apéndice



```
pi@raspberrypi: /
File Edit Tabs Help
GNU nano 2.2.6 File: /etc/rc.local Modified
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
# Print the IP address
/home/pi/bin/script_pd
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^L UnCut Text ^T To Spell
```

Adiós y gracias por haber completado el tutorial.

Colección: Manuales

Autor: Manuel Guerrero Nava

Título: Cómo programar una caja de música con RaspberryPi

Producción:

Secretaría de Cultura / Centro de Cultura Digital

Julio, 2016, Ciudad de México

www.centroculturadigital.mx

www.librosmexico.mx

©2016, Manuel Guerrero Nava

©2016, Ana Laura Alba por diseño

©2016, Dirección General de Publicaciones

Paseo de la reforma 175,

Colonia, Cuauhtémoc, C.P. 06500

Ciudad de México.



Cómo programar una caja de música con RaspberryPi de Manuel Guerrero Nava
está bajo una licencia [Creative Commons Atribución-NoComercial-NoDerivados 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/)
[Licencia internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Hecho en México

CULTURA
SECRETARÍA DE CULTURA



**CENTRO
DE CULTURA
DIGITAL**