Práctica 8: OSC. Open Sound Control

¿Qué es OSC?

Open Sound Control, abreviado OSC, es un **protocolo de comunicación** entre dispositivos (ordenadores, sintetizadores ...) que está optimizado para tecnologías conectadas en red (Hutchinson, 2022).

¿Y qué es un protocolo? Un protocolo es un lenguaje, un código que tanto emisor como receptor comprenden y nos va a servir para enviar datos estructurados que una vez recibidos podamos descifrar.

En este curso vamos a utilizar este protocolo para comunicar **Pure data** con otros programas y/o dispositivos. Esto nos va a permitir, por ejemplo, controlar Pure data desde el móvil. Para entender cómo funciona, primero vamos a utilizar un ejemplo muy sencillo que nos va a permitir controlar desde un móvil, tablet u otro programa del ordenador el Kick drum que hemos hecho en Práctica 7: Kick drum ... | Librería CATEDU . Vamos a utilizar una aplicación o app para ello. El ordenador y el movil que queramos utilizar para este ejercicio deberán de estar conectados a la **misma red**, por ejemplo ambos deben de estar conectados a la wifi de casa. Para asegurarnos de esto podemos apagar los datos de nuestro movil y solamente conectarlo a traves de la wifi.

Etiquetas y datos

¿Os acordáis de la diferencia entre datos e información que veíamos en Programación en genera... | Librería CATEDU? Vamos a ver como toma relevancia a la hora de enviar datos. Para que los datos que recibimos puedan ser interpretados como información tendremos que ponerlos en contexto y para ello vamos a utilizar **etiquetas** que nos indiquen el origen y contexto de cada dato.

https://giphy.com/embed/OKPtqN7dlflYcw8gPu

Figura 1. El programa emisor poniéndole una etiqueta a un mensaje osc, para saber que contiene.

Copyright 2026

Los datos van a llegar por el mismo sitio y pueden ser diferentes como, por ejemplo, la hora de llagada y la hora de salida de un tren. Si solo recibimos la hora: "16:45" y nada mas no podremos saber si es la hora de salida o de llegada necesitaremos recibir un mensaje con otro dato que nos proporcione esa información: "salida 16:45". Ahora ya sabemos que la hora que hemos recibido es la de salida.

¡Vamos a ello! Comencemos por instalar y configurar la app en nuestro móvil o tablet:

> Si no tienes un dispositivo Android más abajo te damos una alternativa para el ordenador y otros dispositivos.

Movil-Tablet

Android

Vamos a instalar la aplicación "**OSCcontroller**", podéis encontrarla en el siguiente enlace.

Una vez instalada, la abrimos y configuramos la IP de la red a través de la cual queremos enviar datos. ¿Qué es la IP? Una IP es la dirección que utilizan los dispositivos conectados a red para identificarse. En nuestro caso, vamos a tomar la **IP local o privada**.

¿Y cómo sabemos cual es nuestra IP? Podremos consultarlo en nuestro ordenador, accediendo a la configuración de red:

Copyright 2026

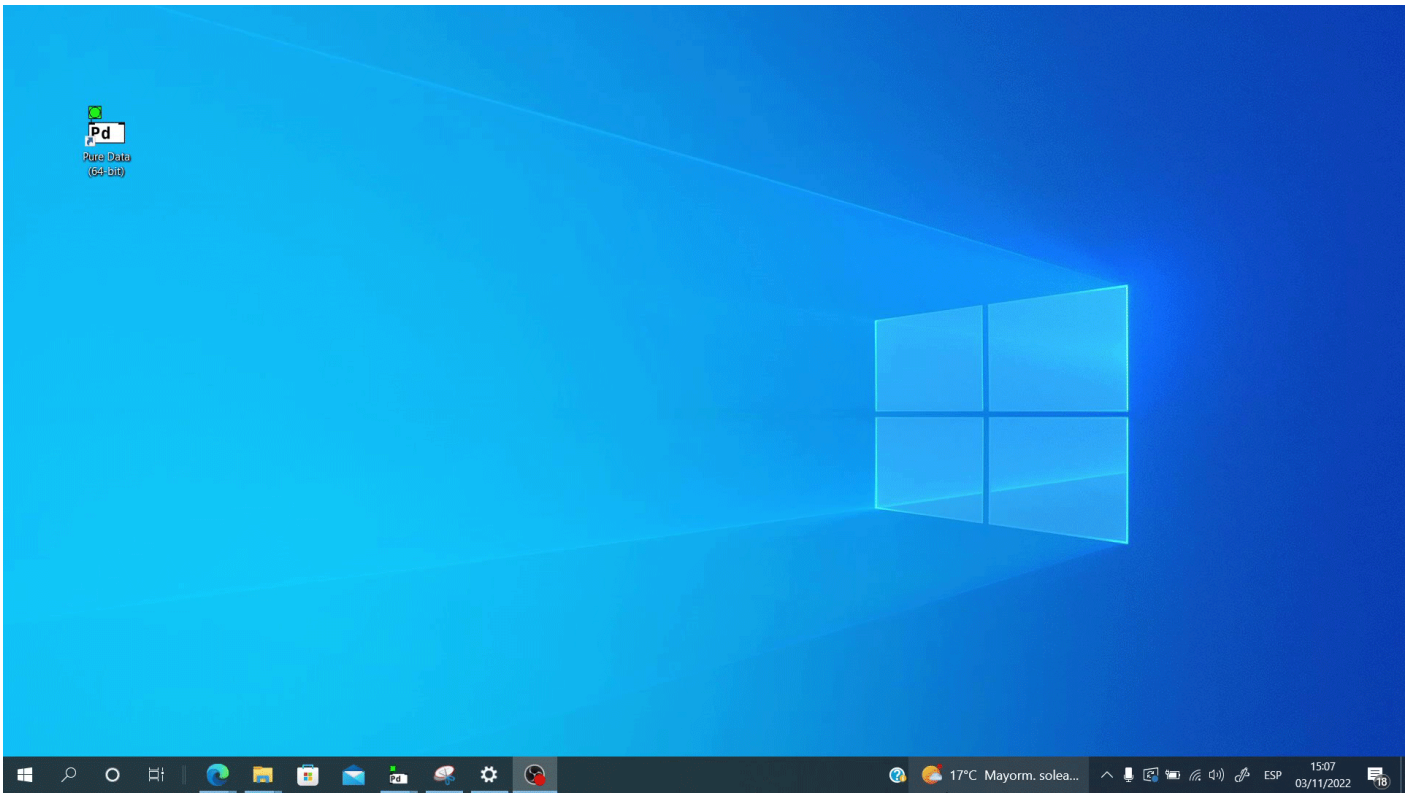


Figura 2. Consultar la IP local en Windows 10.

Esta IP es la que vamos a introducir en la app "OSCcontroller". Tambien vamos a configurar el **puerto** a través del cual nos vamos a comunicar, que sera el mismo que configuraremos en Pure data.

El **puerto** es el **canal especifico** en el que vamos a enviar los datos. Para este ejemplo utilizaremos el 8000, pero podéis utilizar otro puerto que tengáis libre. Nuestra IP puede cambiar así que, si vemos que no funciona, nos aseguraremos de que la IP de nuestra red coincide con la que configuramos en la app.

Copyright 2026

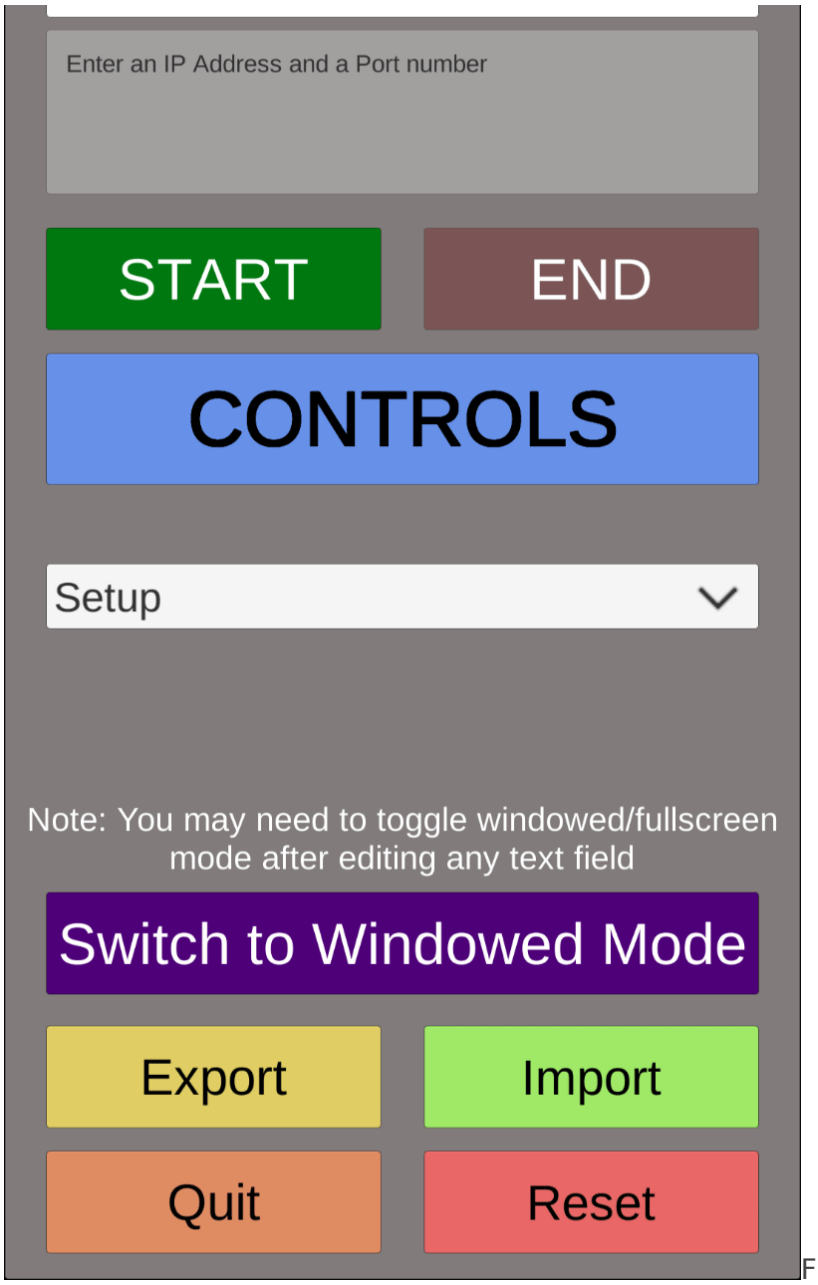


Figura 3. Configuración de IP, puerto, inicio de la conexión y acceso a los controles que enviaran los mensajes osc en la app OSCcontroller.

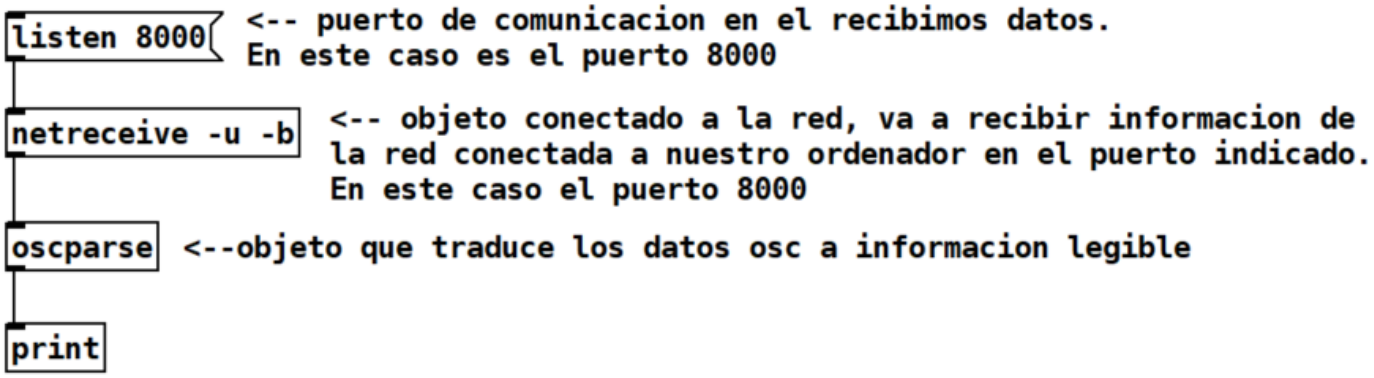
Una vez tengamos IP y puerto configurados pulsaremos Start para iniciar la conexión. El botón azul de control nos llevara a una ventana en la que encontraremos sliders, botones e interruptores y son los valores de estos controles los que se enviaran por osc.

Copyright 2026

¡Nuestro móvil o tablet ya está listo para enviar datos! Ahora vamos a configurar Pure data:

Configuremos Pure data

Para recibir esta comunicación a través de la red, vamos a utilizar el objeto "**netreceive**" de la siguiente manera:



Vamos a configurar el **puerto de escucha** en Pure data enviando un mensaje con el texto "**listen**" y el numero del puerto. Pondremos dos argumentos en el objeto "netreceive" para asegurar su correcto funcionamiento: "**-u**" y "**-b**".

Los datos recibidos por "netreceive" serán **traducidos** por el objeto "**oscparse**" a un language legible. Tras la traducción vamos a imprimir lo que estamos recibiendo.

¡Comencemos a enviar datos! Vamos a tocar los botones y sliders que vemos en la ventana de control del móvil:

Copyright 2026

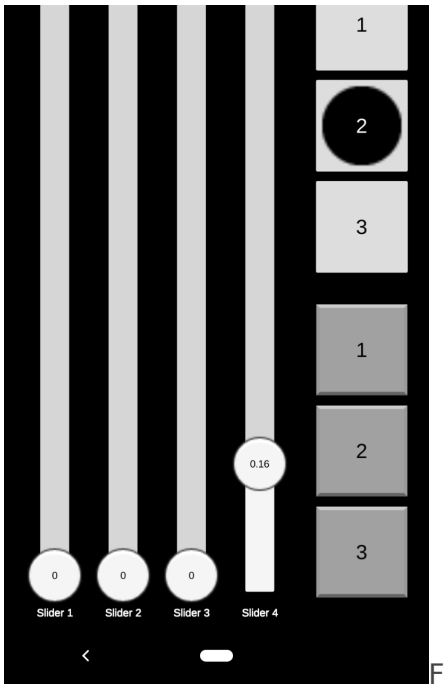


Figura 4. ventana de los controles en la app OSCcontroller.

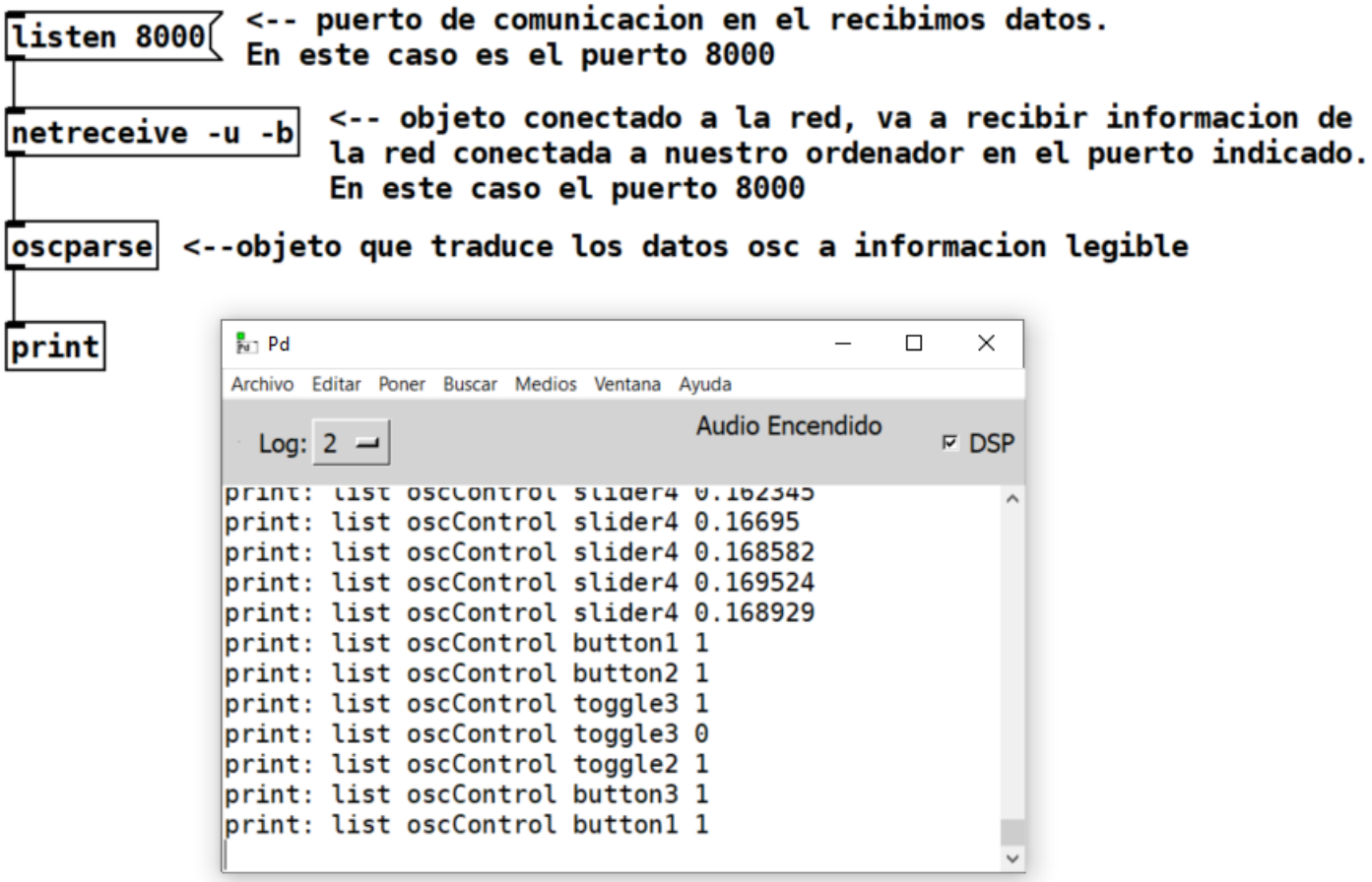


Figura 5. Pure data imprime los datos recibidos por osc y que se han enviado desde la app OSCcontroller.

Vamos a fijarnos en la interfaz del móvil en la figura 4. Tenemos 4 sliders en la parte izquierda, en la parte derecha superior tenemos tres interruptores y debajo de estos tres pulsadores. Cada uno de estos sliders, botones y pulsadores van a tener un código identificativo, una **etiqueta** que nos permitirá saber desde que objeto de control proviene la información recibida. Vamos a fijarnos ahora en las líneas impresas en la ventanita de Pd. En las primeras líneas podemos leer lo siguiente:

print: list oscControl slider4 0.16

Encontramos palabras y números separados por un espacio. Vamos a ver qué indican cada uno de estos grupos. Como bien sabeis el "**print:**" nos indica que esta linea que aparece en la ventanita son datos impresos desde un patch de pd utilizando el objeto print.

La palabra "**list**" nos indica que lo que se está imprimiendo es una lista de elementos.

La palabra "**oscControl**" es una etiqueta que identifica el emisor de estos datos, en nuestro caso es la app OSCcontrol. Podemos configurar el nombre de esta etiqueta en la app:

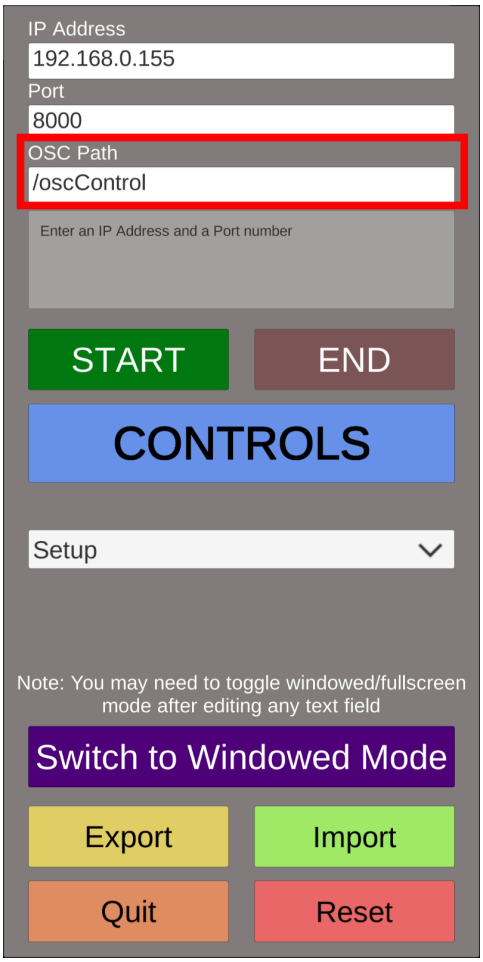


Figura 6. Marcado en rojo se muestra donde cambiar la etiqueta del emisor en la app "OSCcontroller".

La siguiente palabra es una **etiqueta** que nos va a indicar desde **elemento de control** se envían los datos. Que botón, interruptor o slider envía esos datos: slider1, slider2, slider3, slider4, toggle1 toggle2 toggle3, button1, button2 o button3.

print: list oscControl slider4 0.16

Tras esta etiqueta que indica desde que botón, interruptor o slider se envían esos **datos**, encontraremos un valor numérico que corresponde con el estado del botón, interruptor o slider. Este envío se va a realizar cuando cambie el estado del botón, interruptor o slider, por ejemplo, cuando pulsemos un botón o movamos un slider.

Como podéis ver en la figura 5, en la consola de Pd recibimos información de varios actuadores/elementos de control. El slider4 en la posición 0.16, el button1 y el button2 han sido pulsados, el toggle3 ha encendido y apagado, el toggle2 ha sido encendido y permanece encendido y finalmente el button3 y el button1 han sido pulsados.

Copyright 2026

> Ejercico 9: Montad el sistema que acabamos de ver y haced pruebas con los actuadores y el print.

¿Funciona?

https://giphy.com/embed/XoPxquIwsYXpC

Figura 7. El paquete del slider4 de camino a Pure data cuando no enviamos el mensaje "listen 8000" al objeto "netreceive".

Una vez recibimos los **datos en Pd** tendremos que **procesarlos** un poco para poder utilizarlos para modificar parámetros en nuestro patch. Como hemos visto estamos recibiendo una lista con etiquetas y un valor numérico, vamos a desgranar y clasificar esa lista. Vamos a empezar por quitar la palabra "list" de nuestra lista de datos, utilizando el objeto "**list trim**"

Copyright 2026

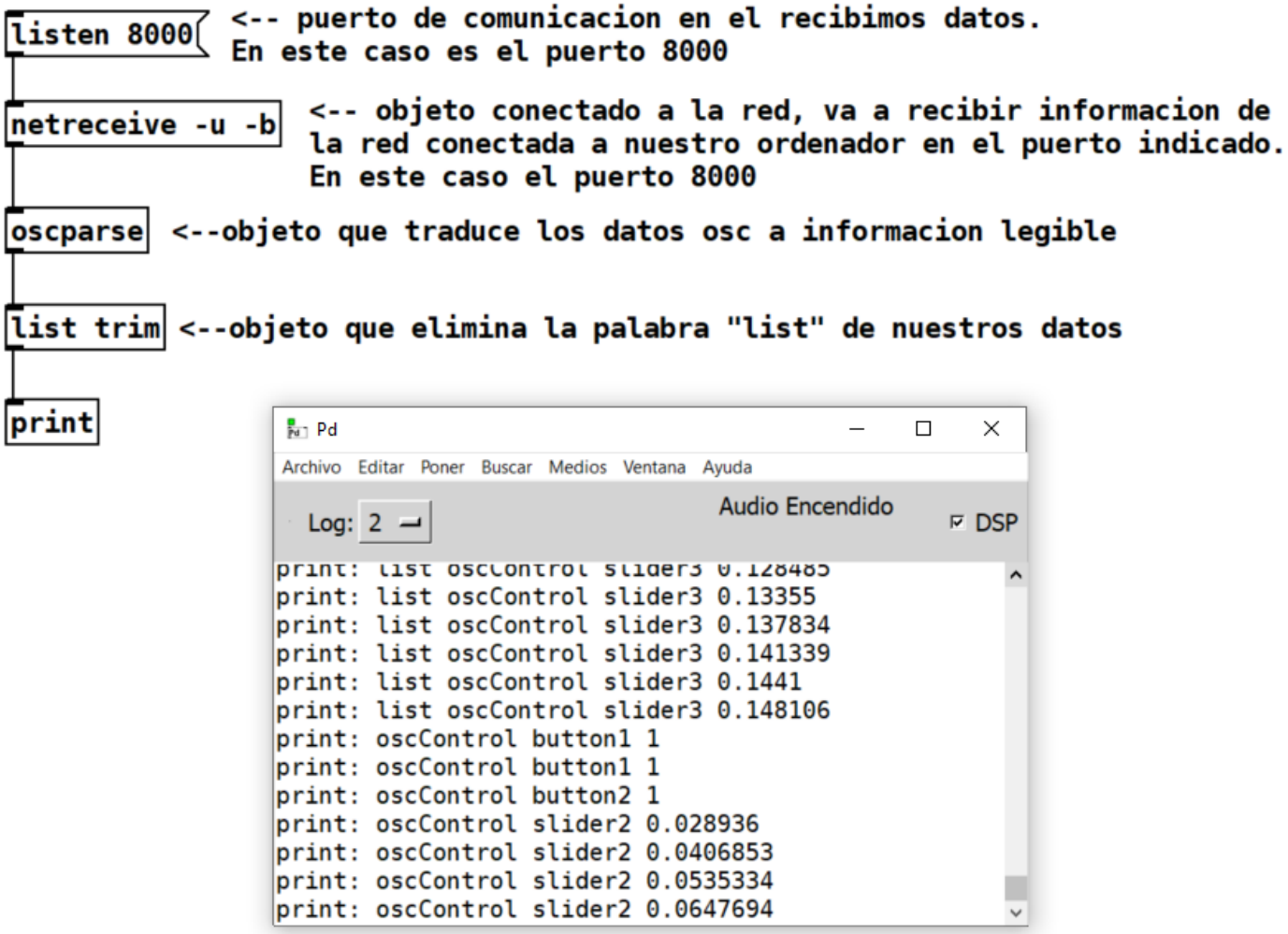


Figura 8. Uso del objeto "list trim" para eliminar la palabra "list" de la lista en Pure data.

Ahora, vamos a ver cómo clasificar los datos en base a sus etiquetas con el objeto "**route**". Al tener solo datos enviados de desde el mismo origen, todas las listas contienen la palabra oscControl que hace referencia al origen de los datos. Si recibiéramos datos de dos fuentes distintas tendríamos una etiqueta diferente para identificar cada fuente, como en el siguiente ejemplo:

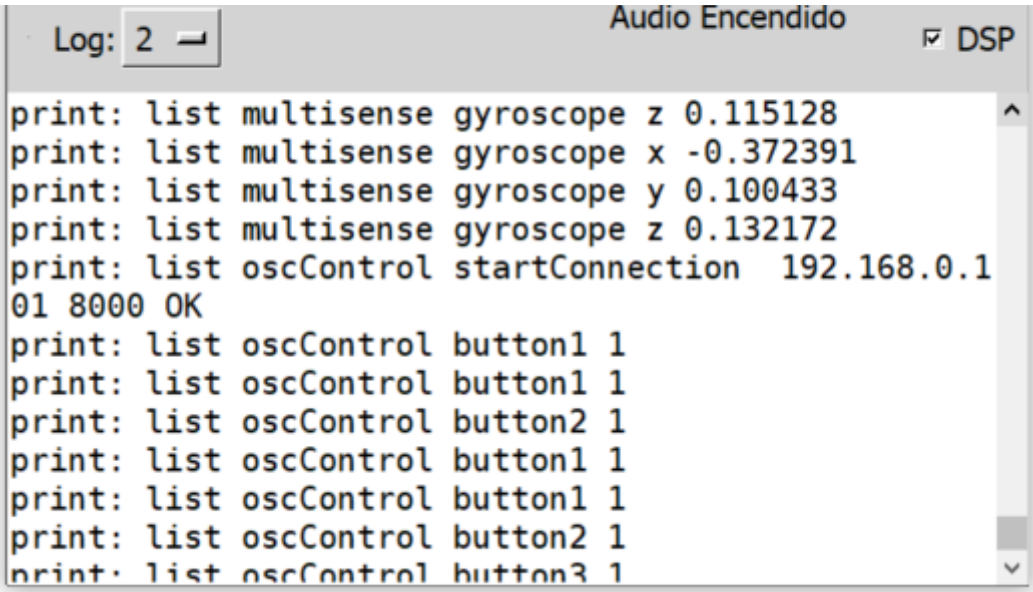


Figura 9. mensajes

osc recibidos de dos emisores distintos en Pure data.

Para separar los datos de cada emisor, utilizaremos el objeto **route** cuyos **argumentos** serán el nombre de la **etiqueta** de cada emisor. En el ejemplo de las figuras 9 y 10, he utilizado dos aplicaciones distintas para enviar mensajes osc y mostraros como seria, pero vosotros de momento solo utilizareis OSCcontrol.

Copyright 2026

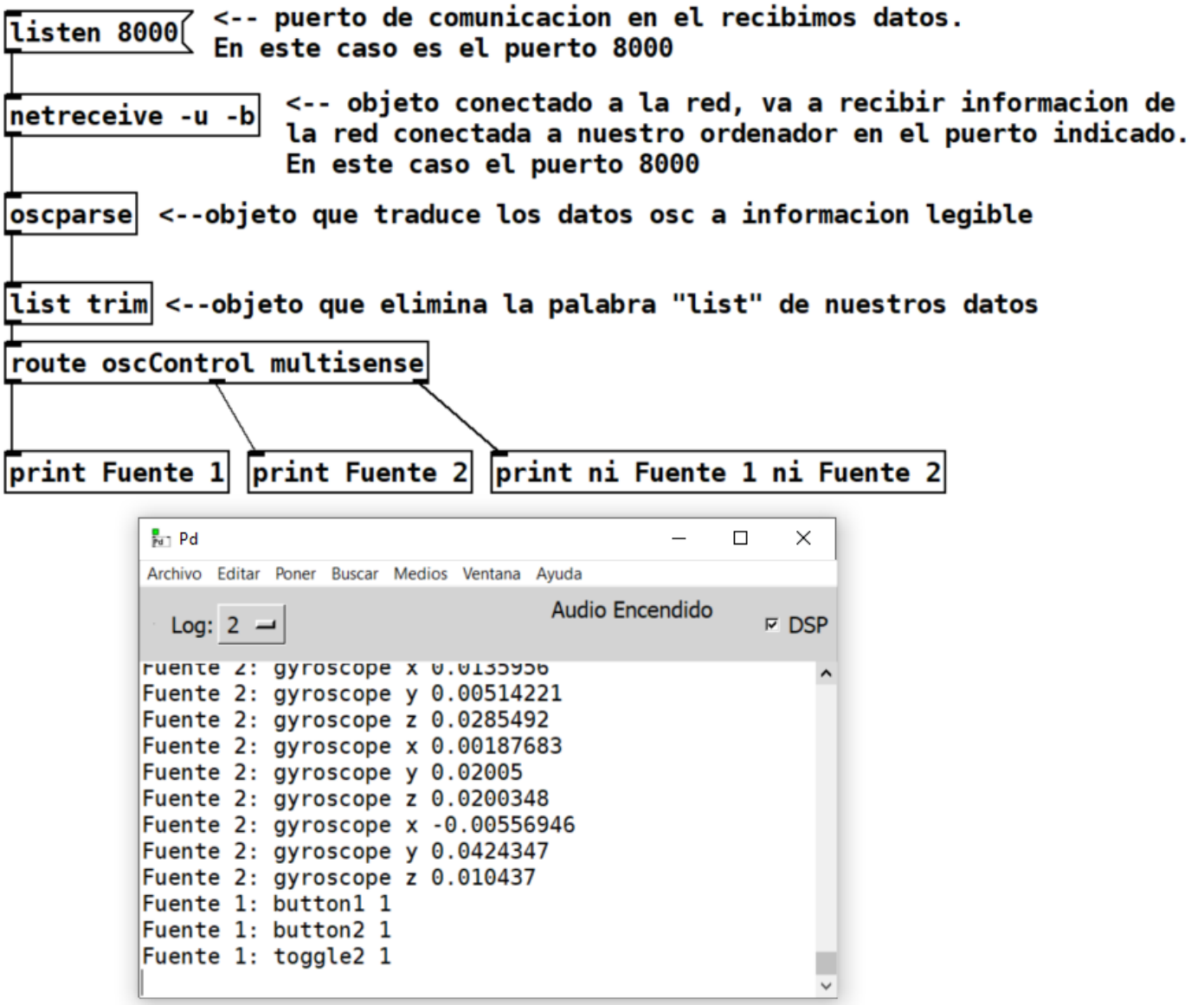
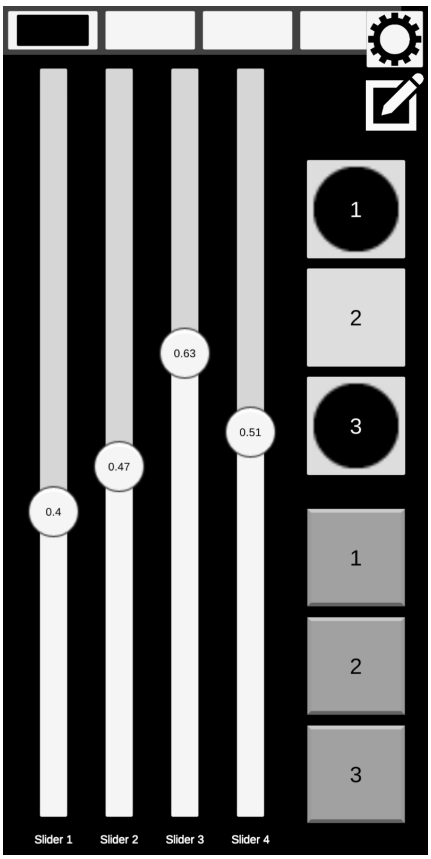


Figura 10. Uso del objeto "route" para clasificar los mensajes osc procedentes de dos emisores en Pure data. Fuente 1=oscControl, Fuente 2=multisense.

Por la **primera salida del route**, saldran las listas con la **etiqueta oscControl,** que es el **primer argumento** del "route"; por la segunda salida, las listas con la etiqueta multisense; y por la tercera salida, las listas que no contengan ninguna de las etiquetas anteriores. Como podéis observar, **tras clasificar** estas listas el route **elimina la etiqueta** de la lista, que era el primer elemento de la lista. La figura 10 era un ejemplo con dos fuentes, aunque en el ejercicio que estamos haciendo solo vamos a trabajar con una fuente, vamos a utilizar el route para quitar la etiqueta oscCcontrol.



Ahora vamos a clasificar los datos que vienen de cada botón utilizando el objeto **route**. Para los botones podemos ver en la lista que hemos impreso previamente que las etiquetas que utilizan son **button1, button2 y button3**. Utilizaremos esas etiquetas en el route.

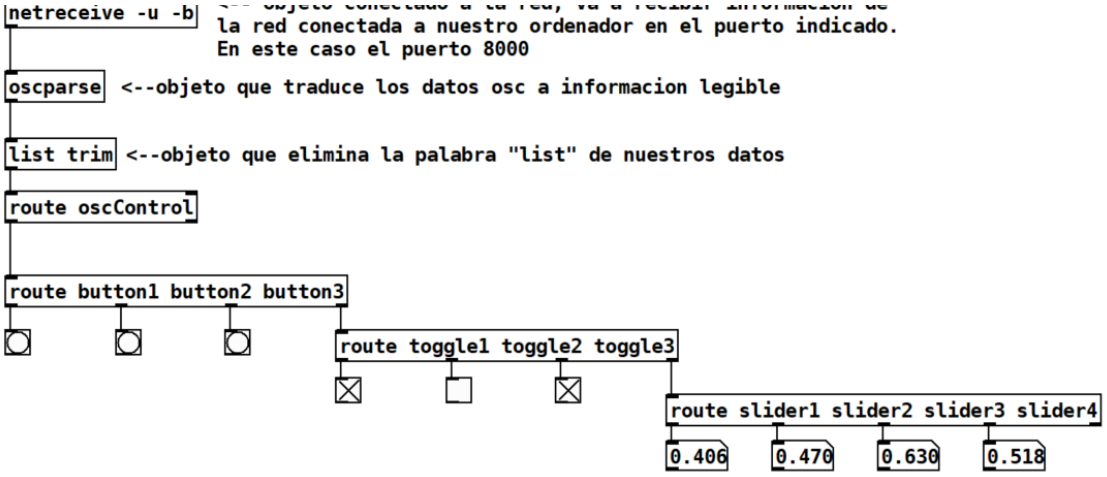


Figura 11.

Clasificación de los datos recibidos por osc en Pure data.

Vemos que al pulsar el botón 1 se envía un 1 a traves de la primera salida del route y al pulsar el botón 3 se envía un 1 a traves de la tercera salida del route. Al encender un toggle se envía un 1 y al apagarlo un 0. Los sliders envían un valor correspondiente a la posición del indicador en un rango de 0 a 1.

> Ejercicio 10 : Una vez hemos conseguido enviar valores desde nuestro móvil a nuestro patch estamos listos para incluir estos controles en el kick drum que hemos hecho en la Práctica 7: Kick drum ... | Librería CATEDU. Vamos a utilizar los tres botones para controlar cada uno de los valores iniciales del envelope que regula la frecuencia del kick drum. Si os fijáis, en el patch que hicimos era un random el que controlaba los valores iniciales del envelope, ahora van a ser los botones de la app oscControl.

> Ejercicio 11: Ahora que ya controlas el envelope de la frecuencia con los botones, intenta controlar la frecuencia general con el slider1 y el volumen con el slider2.

¿Y qué pasa con los que no tenéis un dispositivo Android?

Os vamos a dar una alternativa que es la aplicación TouchOSC. Esta aplicación está disponible para iphone y android pero en estos dispositivos es una aplicación de pago. La version de prueba de ordenador de esta aplicación es gratuita y podréis utilizarla para aprender a enviar mensajes osc desde otro programa a Pure Data.

Aunque utilicemos una aplicación distinta para enviar los mensajes de osc, estos funcionaran de la misma manera que en el ejemplo con Android. Pure Data nos llegaran listas con etiquetas y valores que tendremos que procesar.

Movil-Tablet-Ordenador

MacOSX

Vamos a descargar e instalar la última version para nuestro sistema operativo desde este link:

https://hexler.net/touchosc/bridge-releases

TouchOSC: Abrimos el ejemplo **BeatMachine Mk2**, en **Help > Examples**.
Salimos del modo de edición con Cmd + E
Configuración: En **TouchOSC > Connections**

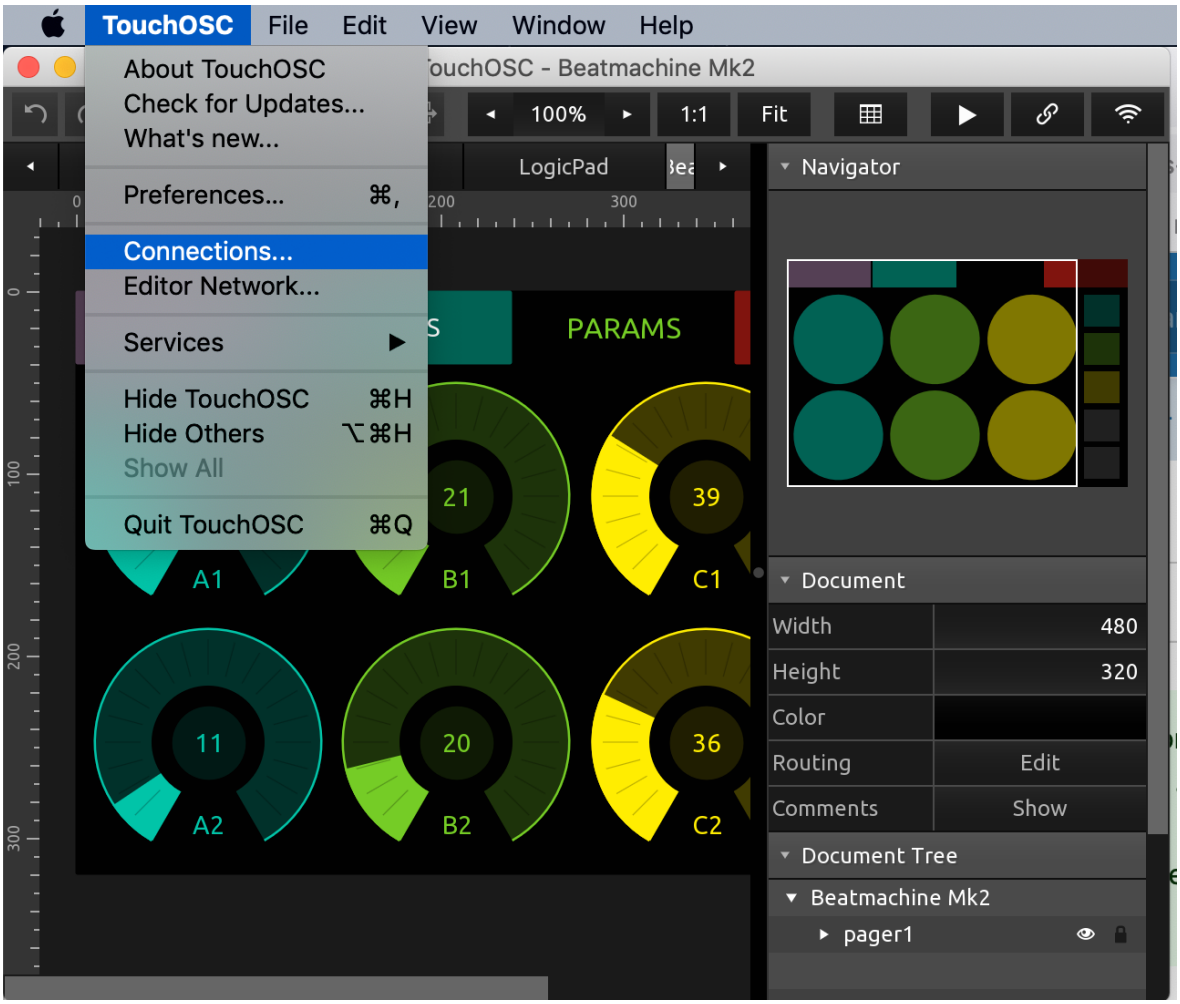


Figura 12. programa "Touch OSC"

En la pestaña **OSC** configuramos en Host nuestra **ipv4** y en **Send Port: 8000**

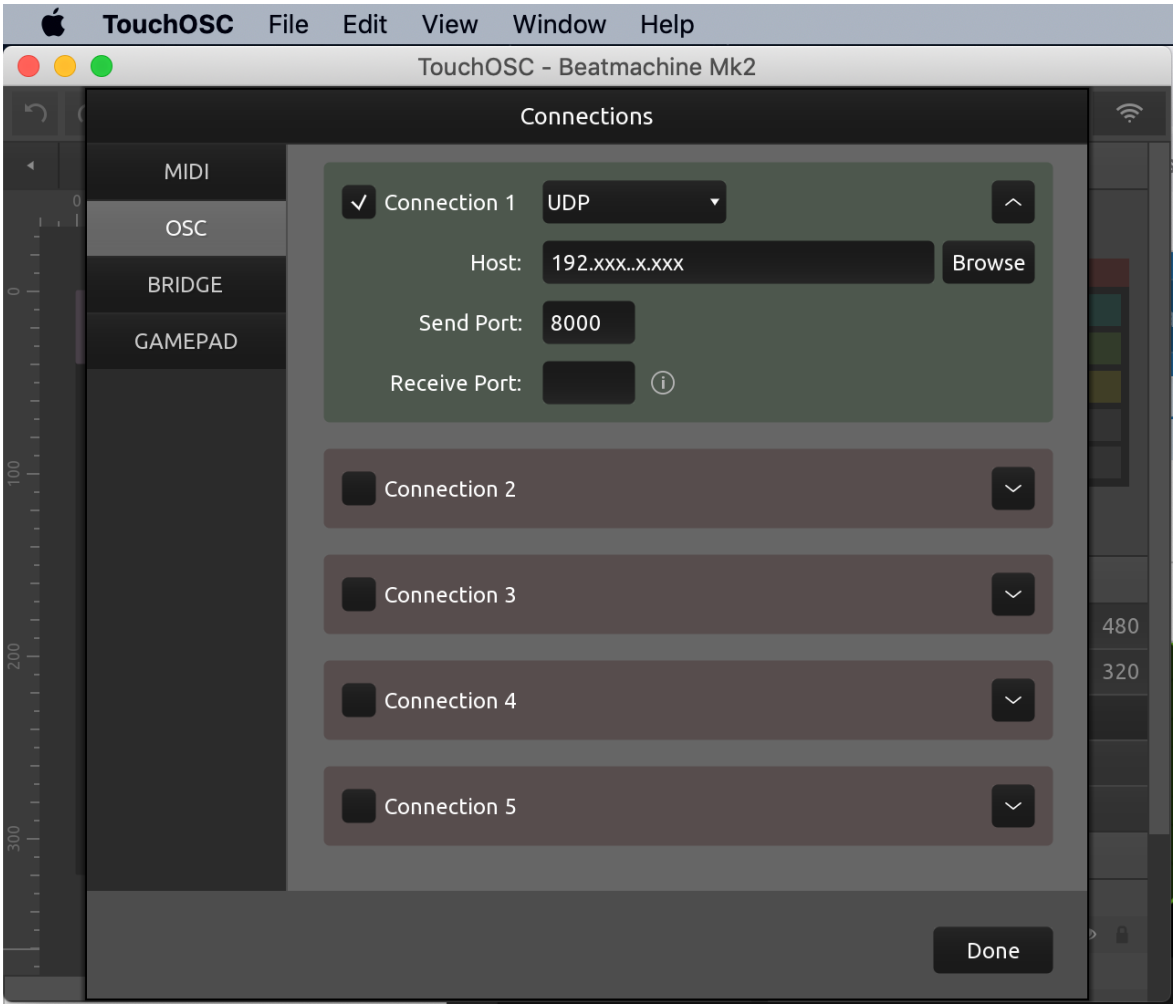


Figura 13. Configuración de IP y puerto en "Touch OSC".

Abrimos nuestro patch de **Pure Data** y hacemos click sobre **listen 8000**. Si no lo hacemos, Pure Data no recibirá información.
Una vez hayamos hecho click, podemos comenzar a girar los botones en TouchOSC y veremos cómo los valores se imprimen en la consola de Pure Data, como puedes ver en el siguiente .gif:

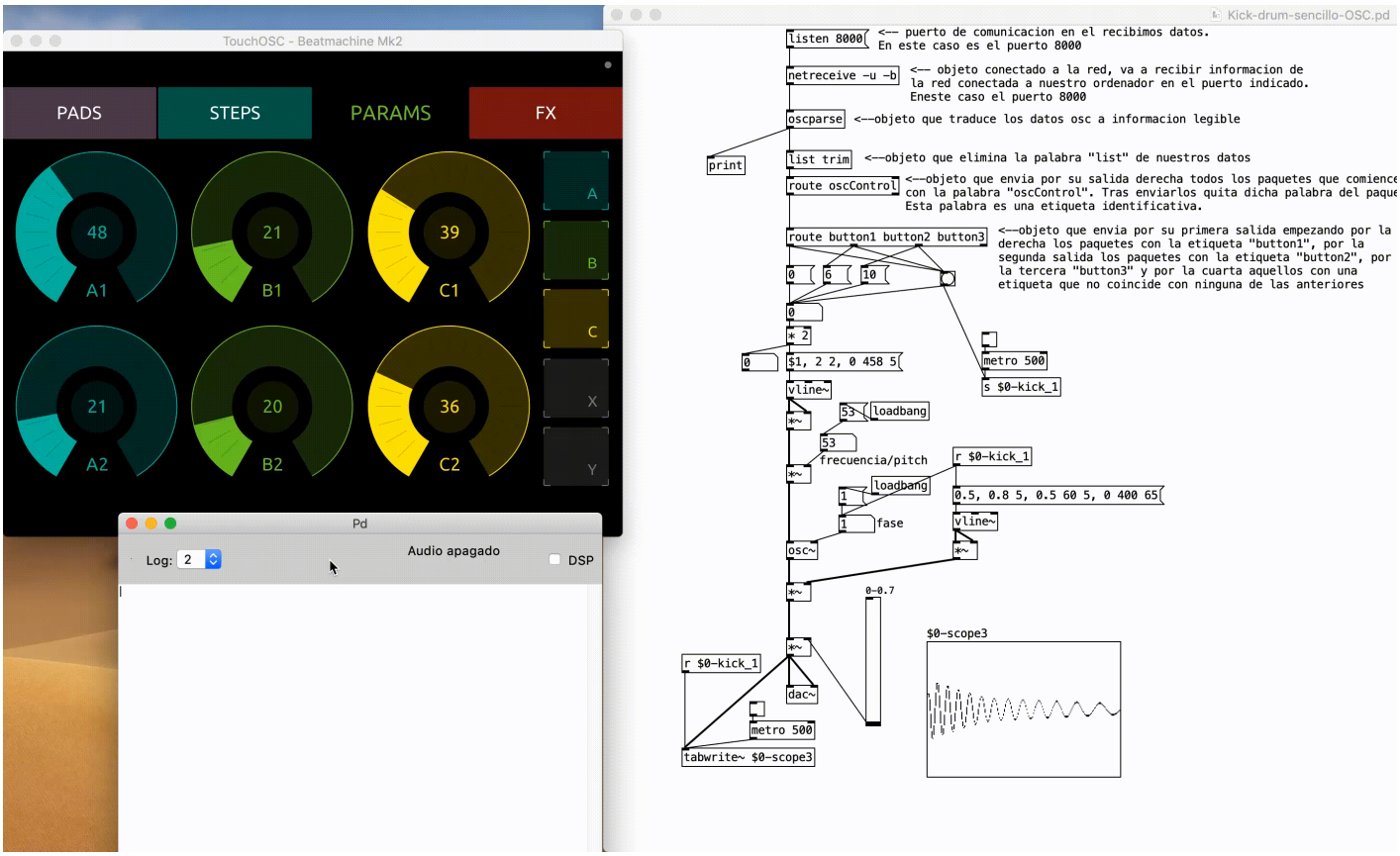


Figura 14. "Touch OSC" enviando mensajes osc a Pure data.

Aqui encontrareis un video de como configurar el programa si tenéis alguna duda. La parte de la aplicación Protokol no es necesaria ya que solo visualiza esos mensajes:

https://hexler.net/touchosc/manual/getting-started-osc

¿Qué tenemos que entregar?

El ejercicio 10 y 11. Sube a la carpeta del Moodle de la Practica 8:

- Ejercicio 10: Un video en el que se vea y escuche como desde otra aplicación se controla el kick drum en Pd. El video deberá llamarse: **practica8-video_vuestronombre_vuestroapellido**.
- Ejercicio 11: una captura de pantalla del patch y el patch que has creado. El patch deberá llamarse: **practica8_vuestronombre_vuestroapellido.pd** y la imagen **practica8-imagen_vuestronombre_vuestroapellido**.

Referencias:

Hutchinson, S. [Simon Hutchinson]. (2022, 7 febrero). *Open Sound Control (OSC) in Pure Data Vanilla | Simon Hutchinson.* [Video]. https://youtu.be/tJ2Kocl-2m4

Figuras:

Figura 1. El emisor poniéndole una etiqueta a un mensaje osc, para saber que contiene. https://giphy.com/gifs/GLSSpain-box-parcel-paquete-OKPtqN7dlflYcw8gPu

Figura 2. Consultar la IP local en Windows 10.

Figura 3. Configuración de IP, puerto, inicio de la conexión y acceso a los controles que enviaran los mensajes osc.

Figura 4. ventana de los controles en la app OSCcontroller.

Figura 5. Pure data imprime los datos recibidos por osc y que se han enviado desde la app OSCcontroller.

Figura 6. Marcado en rojo se muestra donde cambiar la etiqueta del emisor en la app "OSCcontroller".

Figura 7. El paquete del slider4 de camino a Pure data cuando no enviamos el mensaje "listen 8000" al objeto "netreceive". https://giphy.com/gifs/page-usps-XoPxquIwsYXpC

Figura 8. Uso del objeto "list trim" para eliminar la palabra "list" de la lista en Pure data.

Figura 9. mensajes osc recibidos de dos emisores distintos en Pure data.

Figura 10. Uso del objeto "route" para clasificar los mensajes osc procedentes de dos emisores en Pure data. Fuente 1=oscControl, Fuente 2=multisense.

Figura 11. Clasificación de los datos recibidos por osc en Pure data.

Figura 12. programa "Touch OSC"

Figura 13. Configuración de IP y puerto en "Touch OSC".

Figura 14. "Touch OSC" enviando mensajes osc a Pure data.

Copyright 2026

Revision #1
Created 2023-02-17 11:30:16 CET by Julia del Río
Updated 2023-02-17 11:30:16 CET by Julia del Río

Copyright 2026