Práctica 12: Caja de ritmos

Ahora que tenemos los instrumentos de una batería y sabemos hacer un secuenciador vamos a juntarlo todos para crear una caja de ritmos de 8 pasos/tiempos. Os dejo un video de una caja de ritmos de 16 tiempos para que os hagáis una idea de lo que permite:

https://www.youtube.com/embed/gFgITfT2Ihw

¿Qué necesitamos y que estructura va a tener este programa?

Este será nuestro algoritmo general:

1- Necesitamos un **contador** de **8 pasos**.

2- Necesitaremos **enviar** esos **pasos** a cada uno de nuestros instrumentos.

3- Necesitaremos indicar en **cada instrumento** en cual o cuales de los 8 **pasos** queremos que se activen.

4- Necesitaremos que los **instrumentos** generen una señal.

5- Necesitaremos poder **regular el volumen** de **cada instrumento** para ajustar la mezcla.

6- Necesitaremos **mezclar la señal** de los tres instrumentos y **regular el volumen** de ma mezcla (master).

Vamos a ver que tenemos que hacer en cada paso de nuestro algoritmo para conseguir nuestro objetivo:

1- Contador de 8 pasos.

Esto ya lo hemo[illegible] Práctica 3: Contador y... | Librería CATEDU. Y sería algo asi:

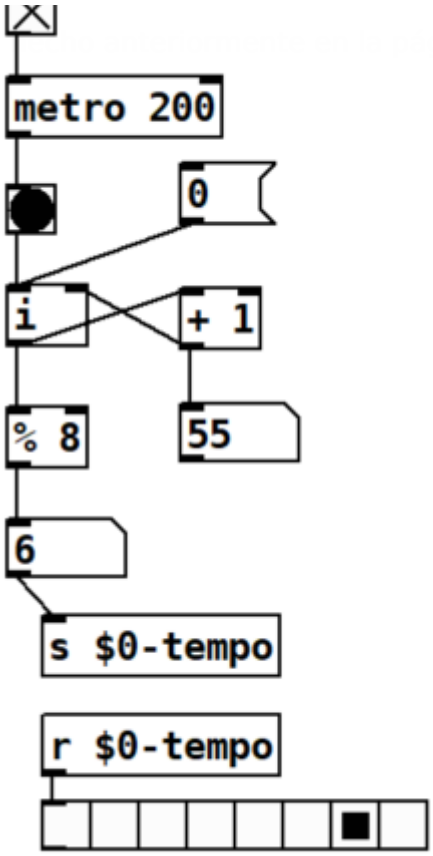

Figura 1. Contador de 8 tiempos.

2- Enviar esos pasos a cada uno de nuestros instrumentos.

Para facilitar el orden y la legibilidad de nuestro patch utilizaremos los objetos "**send**" and "**receive**" para enviar el resultado de nuestro contador (que serán valores del 0 al 7) a cada uno de nuestros instrumentos. Utilizaremos un "**receive**" para cada instrumento

3- Indicar en cada instrumento en cual o cuales de los 8 pasos queremos que se activen.

Este paso va a ser el más novedoso para nostros ya que aún no hemos construido un patch así. ¿Como podemos hacer esto y que necesitamos? escribamos un algoritmo para plantear la solución a nuestro problema.

3.1- Necesitamos saber en qué **paso** está el **contador**.

3.2- Necesitaremos una **configuración** que indique en que **pasos** queremos que se active el **instrumento**.

3.3 - Necesitamos saber **si** el **paso** en que se encuentra el **contador** está **activado** en un **instrumento**.

3.4 - Si el **paso** en el que se encuentra el contador esta **activado** enviaremos un **bang** para activar el instrumento

3.5 - Si el **paso** en el que se encuentra el contador **no** está **activado no** enviaremos un **bang** para activar el instrumento

Vamos a incluir el sub-algoritmo 3 en un subpatch que tendrá 8 inlets y 1 outlets.

3.1- Necesitamos saber en qué paso está el contador.

Habíamos dicho en el punto dos que enviaríamos el resultado del contador con un objeto "send", asique crearemos un objeto "receive" para que a cada uno de nuestros instrumentos les llegue la información de en qué paso está el contador.

3.2- Necesitaremos una configuración que indique en que pasos queremos que se active el instrumento.

En nuestro caso vamos a utilizar 8 objetos **toggle** para cada instrumento, habíamos visto su funcionamiento en la página tal. los colocaremos en fila y cada uno de ellos controlara el estado de un paso del contador. Cuando el toggle este abierto enviara un 1 por su salida y cuando esté cerrado un 0.


Figura 2. Ocho toggles en fila.

3.3 - Necesitamos saber si el paso en que se encuentra el contador está activado en un instrumento.

Para ello vamos a **comparar** la posición en la que se encuentra el contador con el estado de un instrumento para esa posición. Los valores que genera el contador van de 0 a 7 y los valores que genera el toggle que nos indica el estado de cada posición son **0** o **1**. Vamos a buscar una manera de comparar estos dos valores.

Primero vamos a utilizar el objeto "**select**" o "**sel**" para **separar** cada uno de los **pasos del contador**. ¿Cómo funciona este objeto? En el objeto "sel" podremos introducir tantos argumentos como elementos queramos evaluar, cuando un **elemento recibido** por el sel **coincida** con alguno de su **argumento** enviara un **bang** a traves del outlet correspondiente con la posición del argumento.

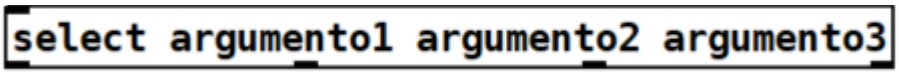

Figura 3. Objeto select.

Esto quiere decir si coincide con el **segundo argumento**, enviara un bang por el **outlet segundo** . "select" se puede utilizar también con símbolos, pero no se pueden mezclar símbolos y numero. En nuestro caso vamos a utilizar números. Pero vemos que para tres argumentos tenemos 4 outlets. Esto es porque por el **ultimo outlet** se enviarán aquellos valores que **no coincidan** con ninguno de los argumentos. Abre el patch *select.pd* para probar el objeto.

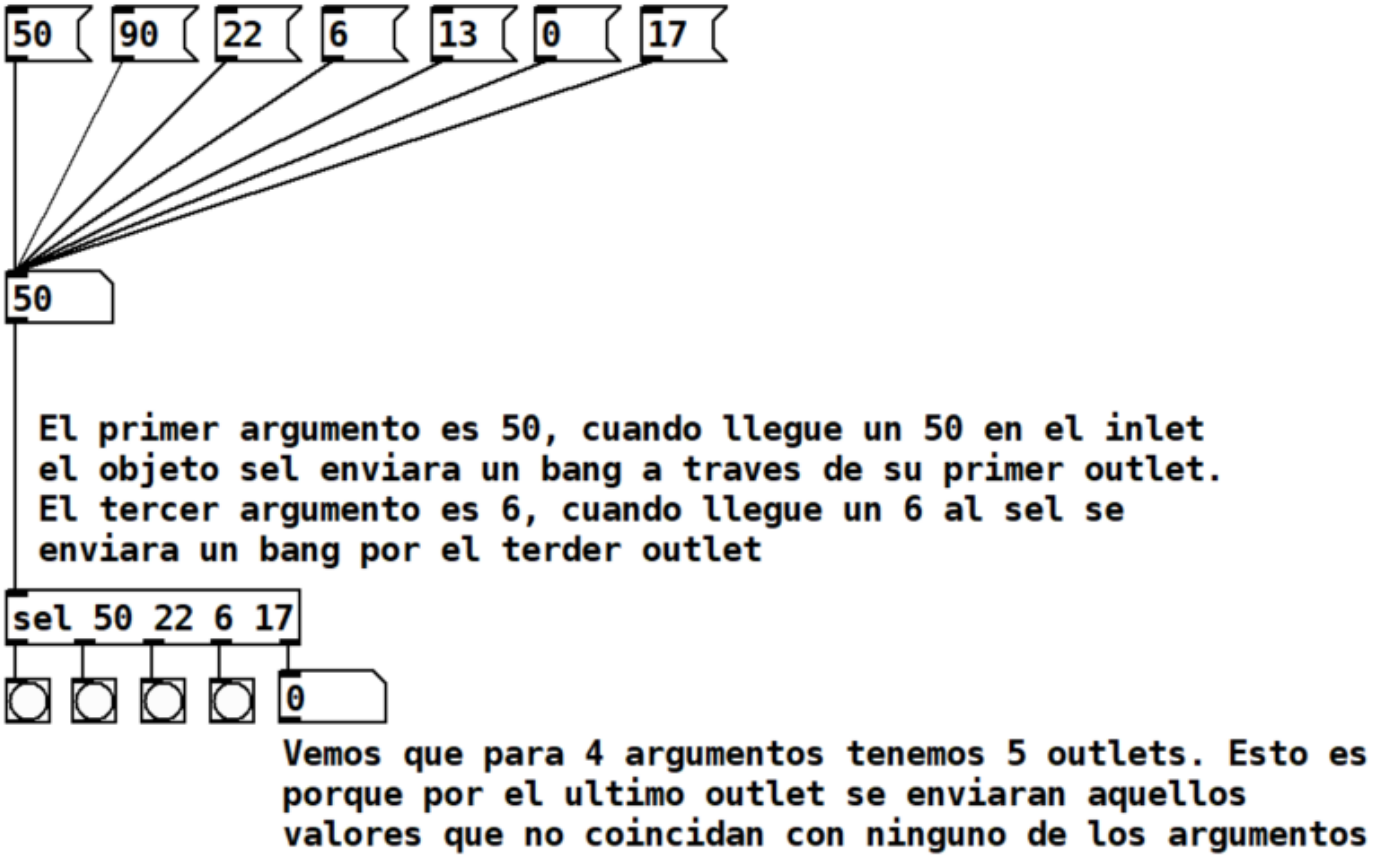

Figura 4. Patch *select.pd.*

Ahora que ya conocemos el objeto select vamos a utilizarlo para separar cada uno de los pasos del contador. Como el contador genera números de 0 a 7 vamos a crear un "select" con 8 **argumentos**, que **coinciden** con los **valores** que genera nuestro **contador**:

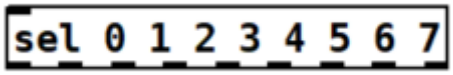

Figura 5. Objeto "select" que vamos a utilizar en esta práctica.

Cada vez que llegue al inlet del "select" de la figura 5, un 6, se enviara un bang por la séptima salida. El objeto "r $0-tempo" recibe los valores que genera el contador.

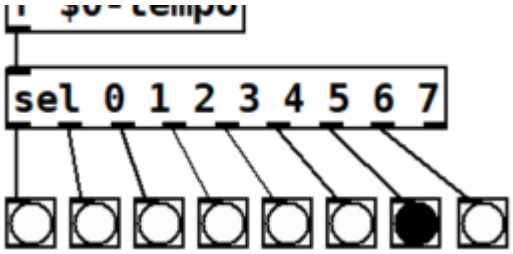

Figura 6. Objeto "sel" que acaba de recibir un 6 en su inlet y envía un bang por el séptimo outlet. (el 6 es el séptimo argumento).

Ahora vamos a crear una **estructura que compare** cada paso del contador con el estado del instrumento en ese paso. Empezamos por el **primer paso**, que en los valores que nos proporciona el **contador** corresponde con el **numero 0**, y que es el primer argumento de nuestro "select". Cada vez que el contador se encuentre en la posición 0 nuestro select enviara un bang por su primer outlet. Habíamos dicho que utilizamos 8 objetos toggle para marcar los pasos en los que queremos que este activado el instrumento. Como el toggle activado envía un **1** vamos a utilizar ese valor como **parámetro de comparación**. Vamos a utilizar el objeto **"=="** para comparar el estado de un instrumento en un paso determinado. Cada vez que el contador este en el paso que estamos evaluando queremos que se realice una comparación con el estado del instrumento en ese paso. Es algo así como responder a la pregunta. Estamos en el paso 1, en qué estado está el instrumento en el paso 1? Si esta activado activa el instrumento.

> Recordar que el los operadores matemáticos el inlet izquierdo introduce valor y da la orden de realizar la operación, por el contrario, el inlet derecho solo introduce valor.

Como queremos que la comparación se realice cuando el **contador** este en el paso que estamos del **inlet izquierdo de "=="** cada vez que el contador se ando.

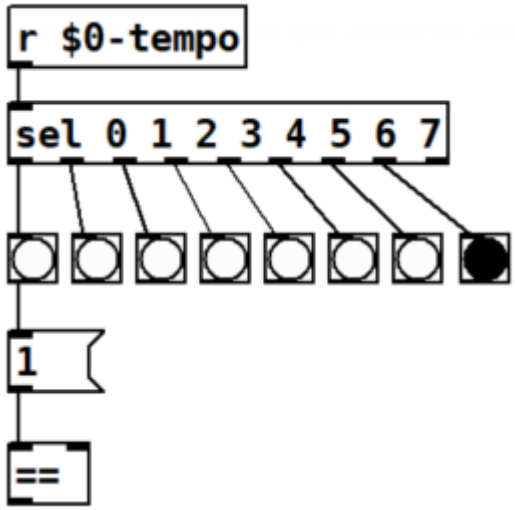

Figura 7. Cuando el objeto "select" reciba un 0 por su inlet, un bang saldrá por su primer outlet y enviara el mensaje que contiene "1" al inlet izquierdo del

objeto "==".

Por el **inlet derecho de "=="** vamos a recibir el **estado del instrumento** en ese paso. Como es el p[illegible]ggle, que cuando este **activado** sera un **1** y cua[illegible]

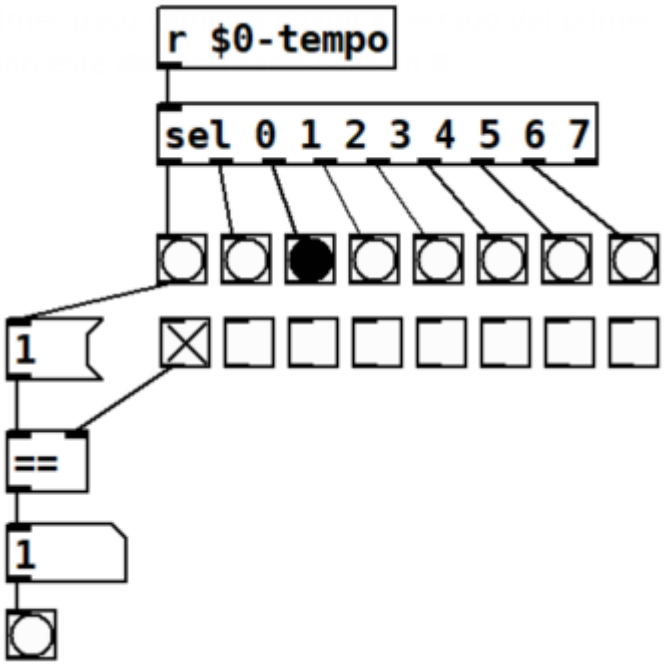

Figura 8. El primer paso del control de este instrumento esta activado. Cuando el primer toggle este activado se enviará un "1" al inlet derecho del objeto "==", cuando el toggle este desactivado enviará un "0".

Cuando el primer toggle este activado se enviará un "1" al inlet derecho del objeto "==", cuando el primer toggle este desactivado enviará un "0". Cuando el objeto "select" reciba un 0 por su inlet, un bang saldrá por su primer outlet y se enviara un "1" al inlet izquierdo del objeto "==". Cuando los valores que recibe el objeto "==" sean iguales, enviara un 1 por su outlet, si los valores procedentes del contador y del estado del instrumento no coinciden, enviara un 0.

La operación de comparación en el objeto "==" se realizará cada vez que llegue algo al inlet izquierdo. En nuestro caso al inlet izquierdo llega siempre un 1 y sera el **estado del instrumento** recibido en el inlet derecho el valor que **alterne** entre **0 y 1**, dependiendo de si este activado ese paso o no.

3.4 - Si el paso en el que se encuentra el contador esta activado enviaremos un bang para activar el instrumento + 3.5 - Si el paso en el que se encuentra el contador no está activado no enviaremos un bang para activar el instrumento

Cuando el toggle este activado y el objeto "==" se realize la comparación **"1 ==1"** el objeto "==" emitirá por su **outlet** un **1**, ya que sus dos inlets cumple la condición de igualad que establece el operador matemático. Cuando el toggle este desactivado y el objeto "==" se realize la comparación **"1 ==0"** el objeto "==" emitirá por su **outlet** un **0**, ya que la condición de igualdad no se cumple. Para activar el instrumento nos interesa tener un bang cada vez que esta condición de cumpla por lo que volveremos a utilizar el objeto **"select"** para emitir un **bang** cada vez que el

objeto "==" envíe un 1 por su outlet y este **bang** sera el que **active** el **instrumento**.

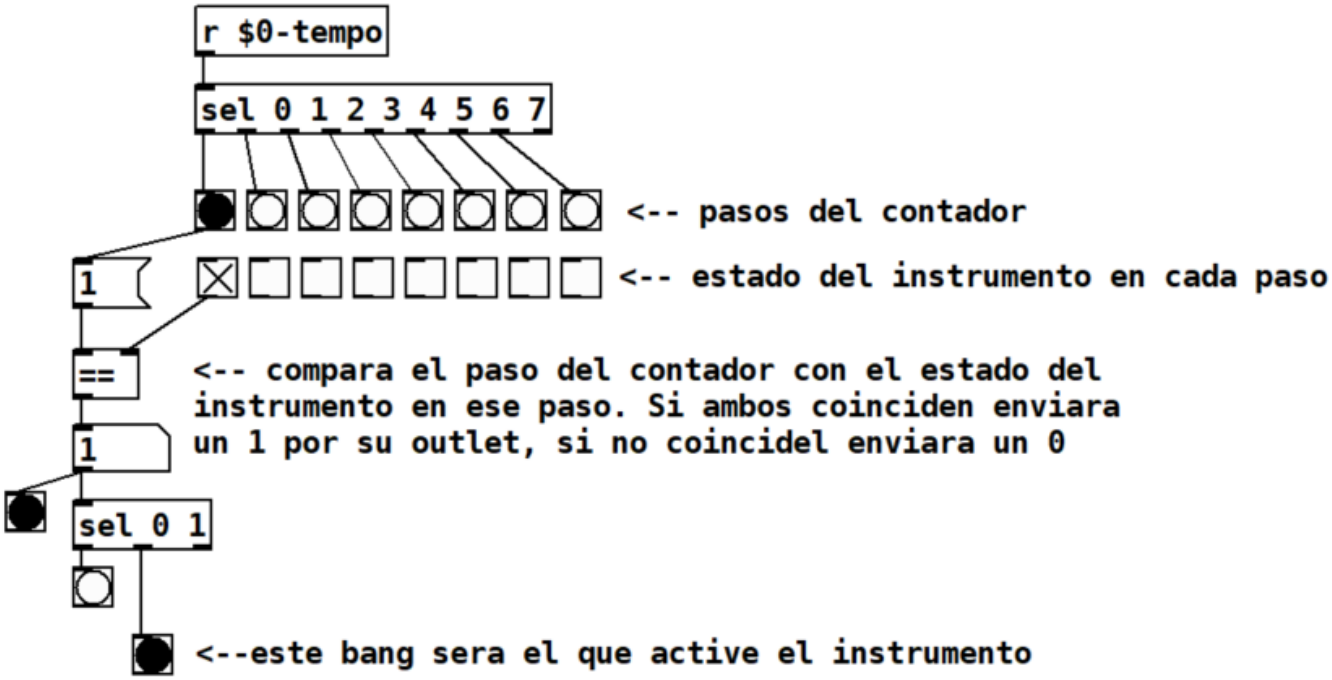

Figura 9. Comprobación del estado del instrumento en el paso en el que se encuentra el contador (primer paso). En ese paso el instrumento esta activado por lo que tras comparar con el "==" y clasificar con el "sel 0 1" obtenemos un bang por el segundo outlet de objeto "sel 0 1".

Lo que hemos hecho hasta el momento evalúa solo el estado del instrumento en el primer paso, tendremos que **repetir** lo mismo **para cada paso**:

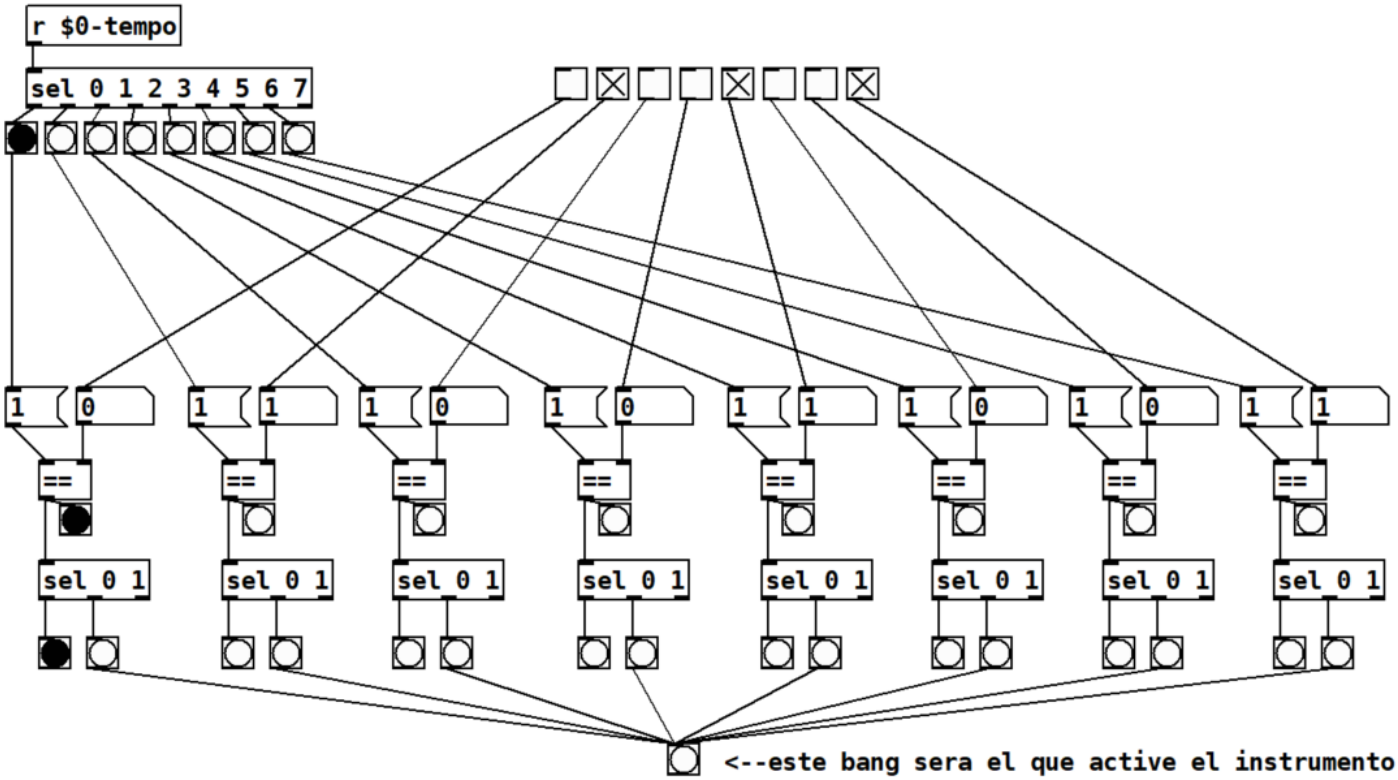

Figura 10. Patch que compara el estado del instrumento en cada uno de los 8 pasos que genera el contador. Cada vez que un paso del instrumento este activado se enviara un bang para activar el sonido.

Una vez lo tengamos hecho para cada paso tendremos completado el control de 1 instrumento, para controlar el resto de los instrumentos, tendremos que **repetir la misma estructura** y lo haremos creando un **subpatch** donde meteremos toda esta estructura de control.

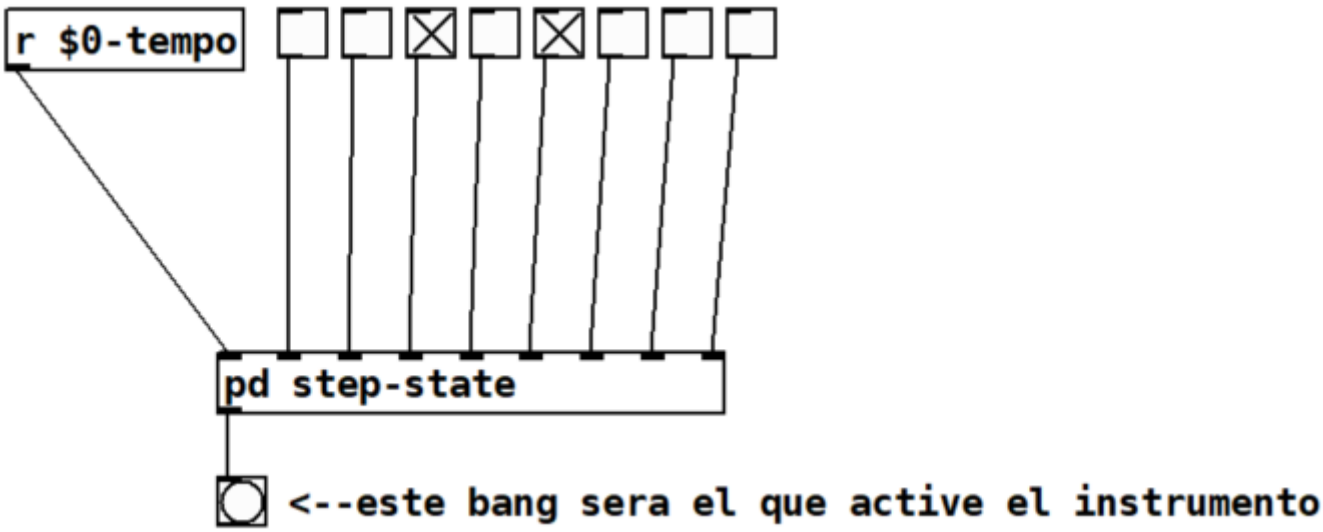

Figura 11. Subpatch que contine gran parte de la estructura de la figura 10.

Este **subpatch** tendra **9 inlets** y **1 outlet**. 8 de los inlets enviaran el estado del instrumento en cada paso que controlan los toggles y 1 de los inlets enviara el valor del contador. En el outlet

tendremos el bang que controla el instrumento.

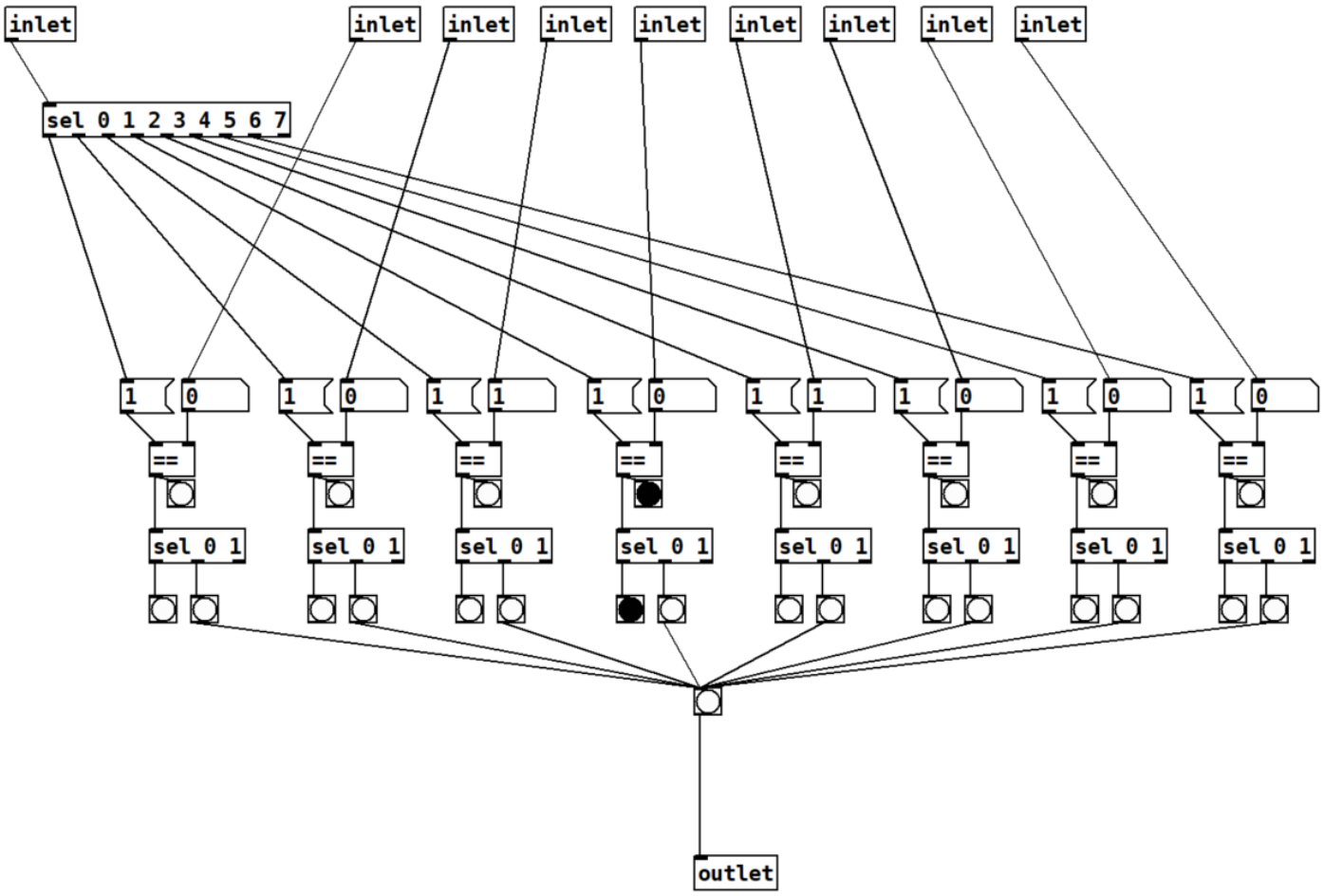

Figura 12. Estructura de control de la figura 10 metida en un el subpatch "pd step-state" de la figura 11.

Ya tenemos nuestro sistema de control para los instrumentos. ¡Traigamos ahora los instrumentos!

https://giphy.com/embed/11MS2pksWxB8SA

Figura 13. La orquesta de camino a vuestro patch de pd.

4- Instrumentos generen una señal, o que la reproduzcan.

Los hemos hecho en lecciones anteriores: Snare drum y Hi hat | Librería CATEDU y Micrófono: Grabar y re... | Librería CATEDU

Podemos utilizar los instrumentos que hemos hecho en la lección anterior o también reproducir sonidos grabados, vamos a combinar estos. Vamos a **meter los instrumentos en subpatches** y dejar fuera del subpatch solo los parámetros que queramos o necesitemos. En previos capítulos hemos creado el **kick drum**, **snare drum** y **hi hats.** Una vez hayamos creado los subpatches en el patch de cada instrumento, vamos a coger esos subpatches y traerlos al patch en el que estamos creando nuestra caja de ritmos.

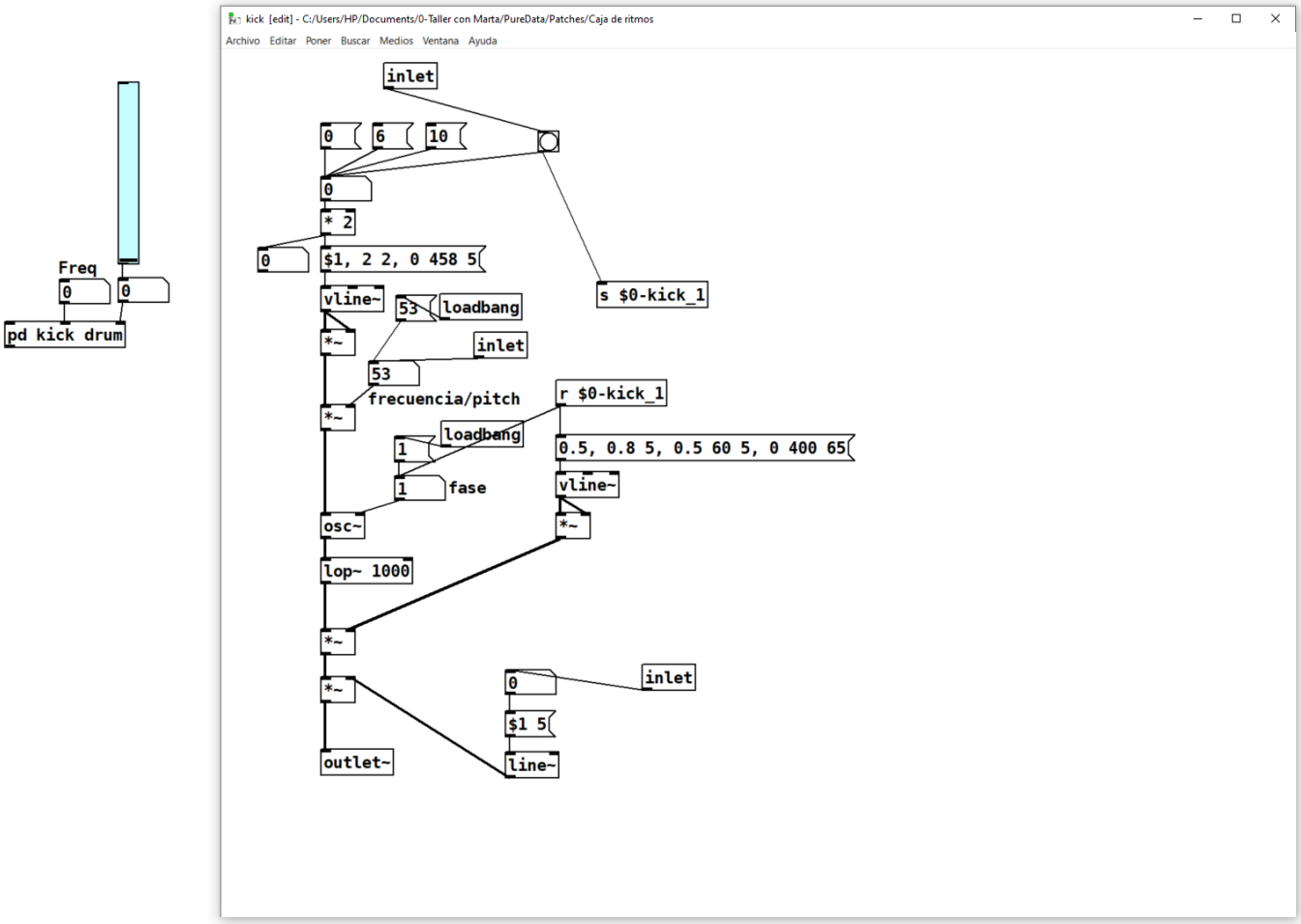

Figura 14. Subpatch para el kick drum o bombo.

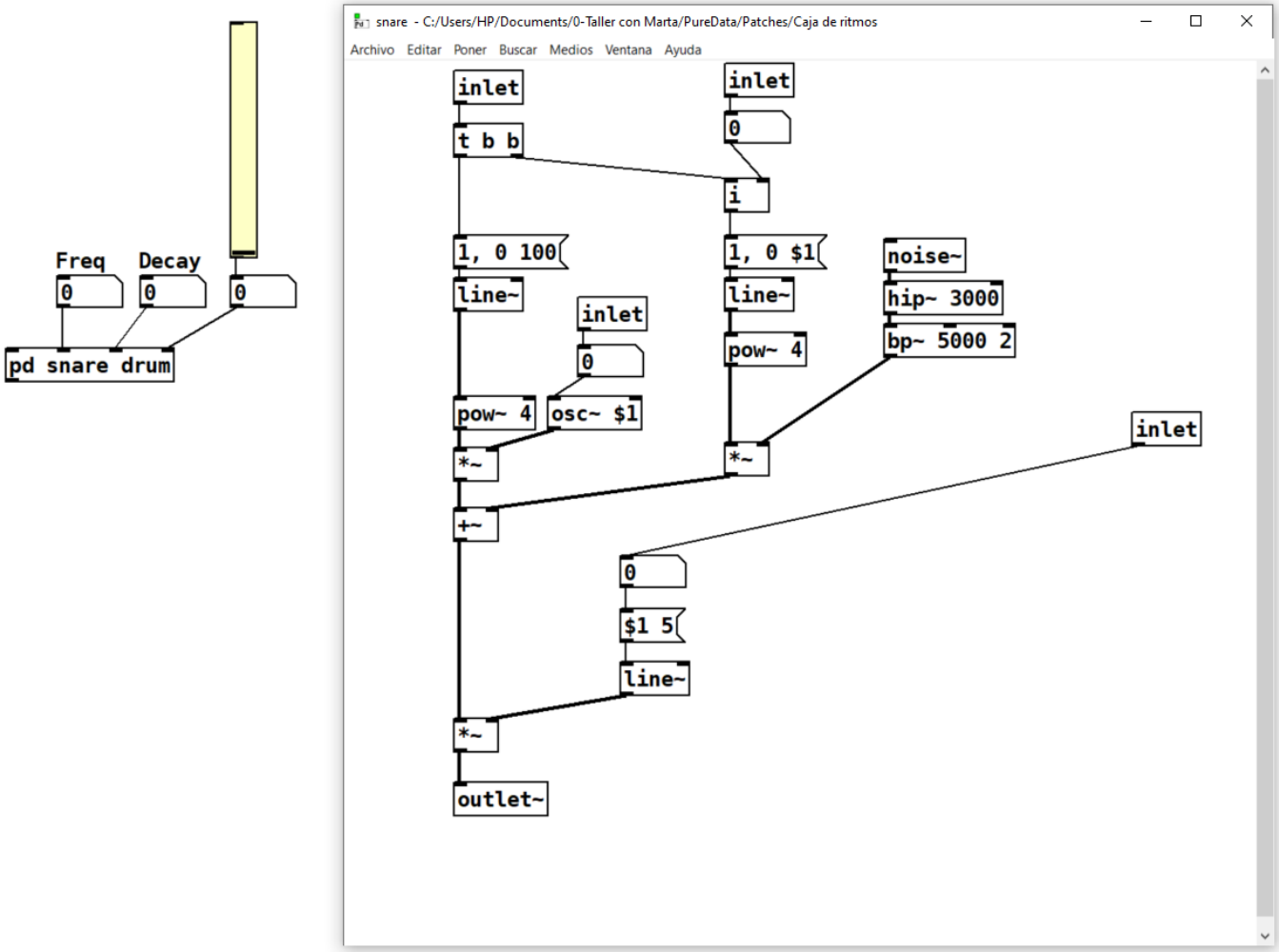

Figura 15. Subpatch para snare drum o caja.

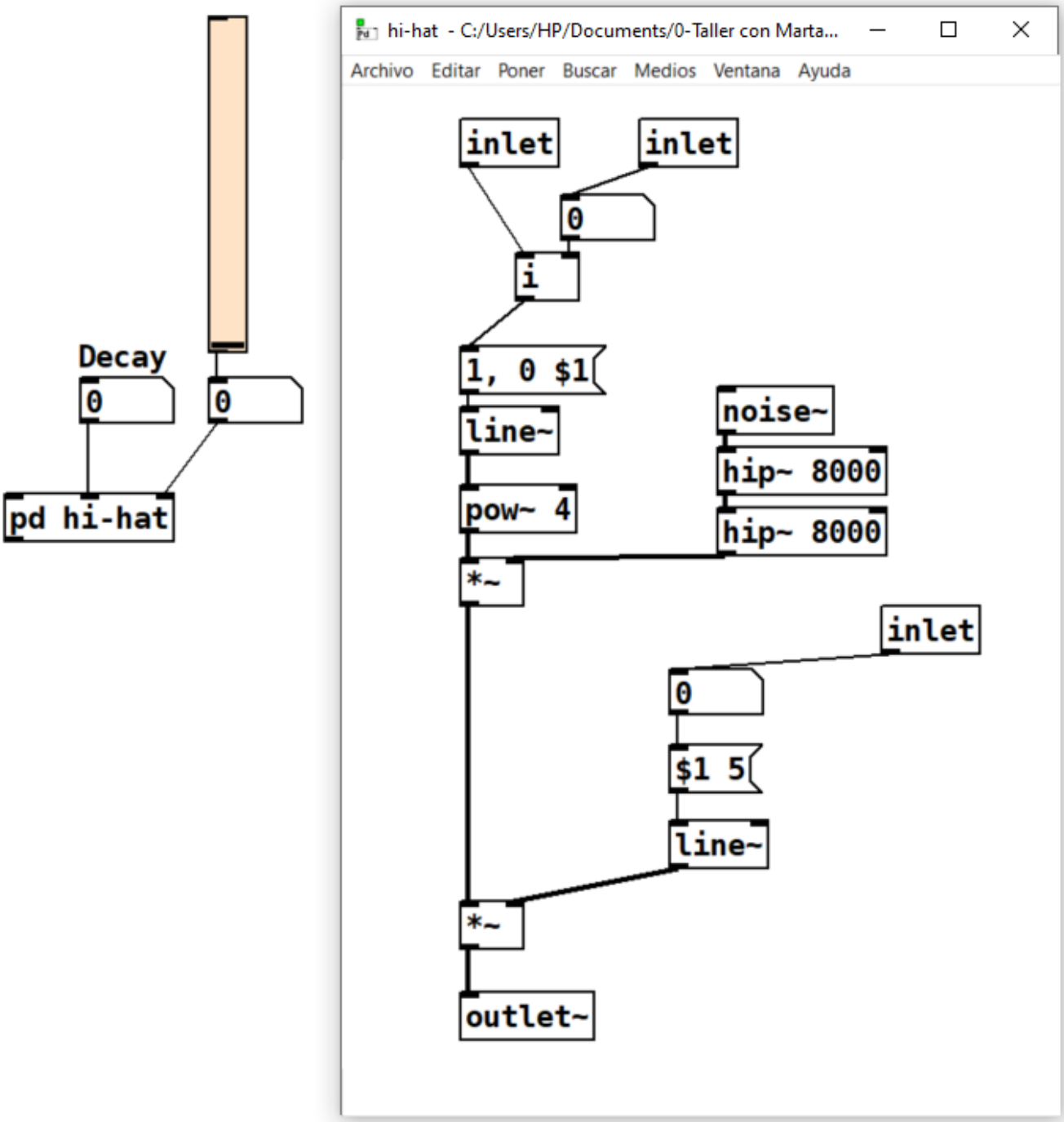

Figura 16. Subpatch para hi hats o platillos.

Copiamos los subpatches de nuestros instrumentos, seleccionándolos en el modo edición, clicamos y sin soltar movemos el rato cubriendo el area donde se encuentren los objetos que queremos seleccionar, una vez seleccionados se pondrán en azul). Los copiámos (Ctrl + C) y los pegamos (Ctrl + V) en el patch que queramos:

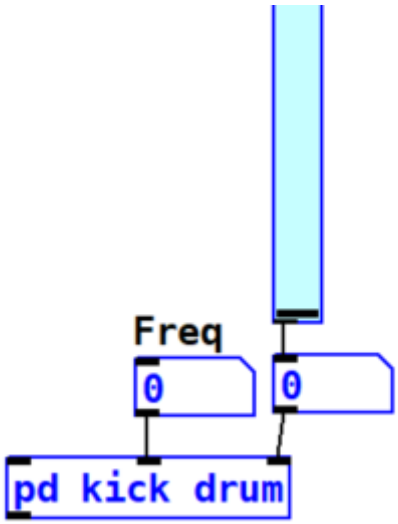

Figura 17. El subpatch del kick drum seleccionado.

Una vez tengamos los instrumentos, conectamos cada uno con un subpatch de control.

Vamos a añadir también un instrumento utilizando un sample, hemos visto cómo hacerlo en Micrófono: Grabar y re... | Librería CATEDU:

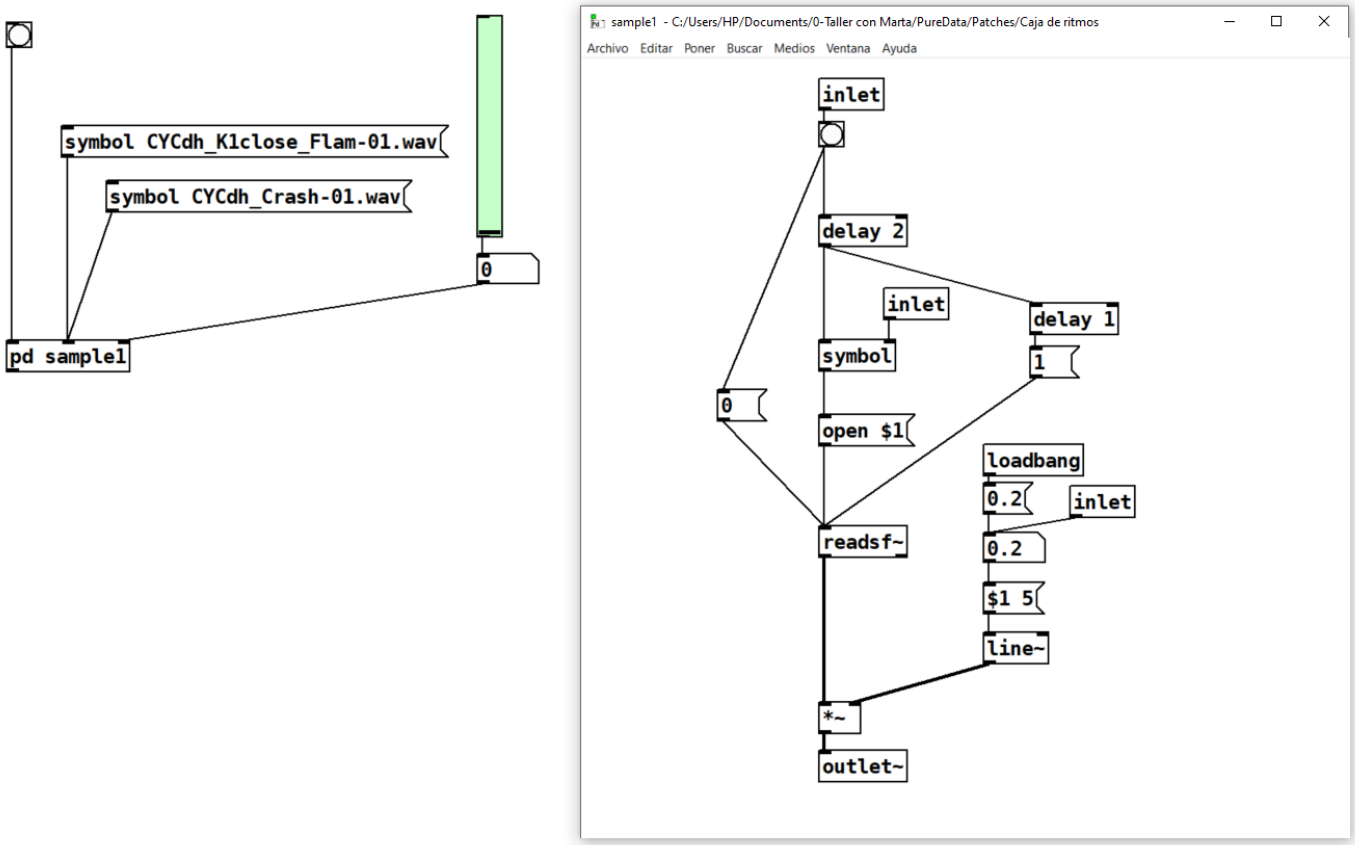

Figura 18. patch *Sample-subpatch.pd*. Subpatch para el sonido procedente de un archivo.

En el mensaje de open hemos sustituido el nombre del archivo por **$1** para poder cargar diferentes archivos enviado un mensaje con el nombre del archivo que queremos reproducir. Hemos añadido al mensaje la palabra symbol para que se identifique como tal. Recordar que los archivos tienen que estar en la misma carpeta que el patch.

> Recordar que el orden en el que conectáis los objetos importa. El orden en que el bang activa en 0 y el delay importan. Para controlar el orden también podéis utilizar un delay muy rápido para aseguraros de que el orden en que se ejecutan los comandos es el que queréis.

5- Regular el volumen de cada instrumento para ajustar la mezcla.

Como podréis observar en las figuras 14, 15, 16, 17 y 18, he añadido un control de volumen que se puede modificar desde fuera del subpatch a todos los instrumentos. Estos nos va a permitir ajustar la mezcla.

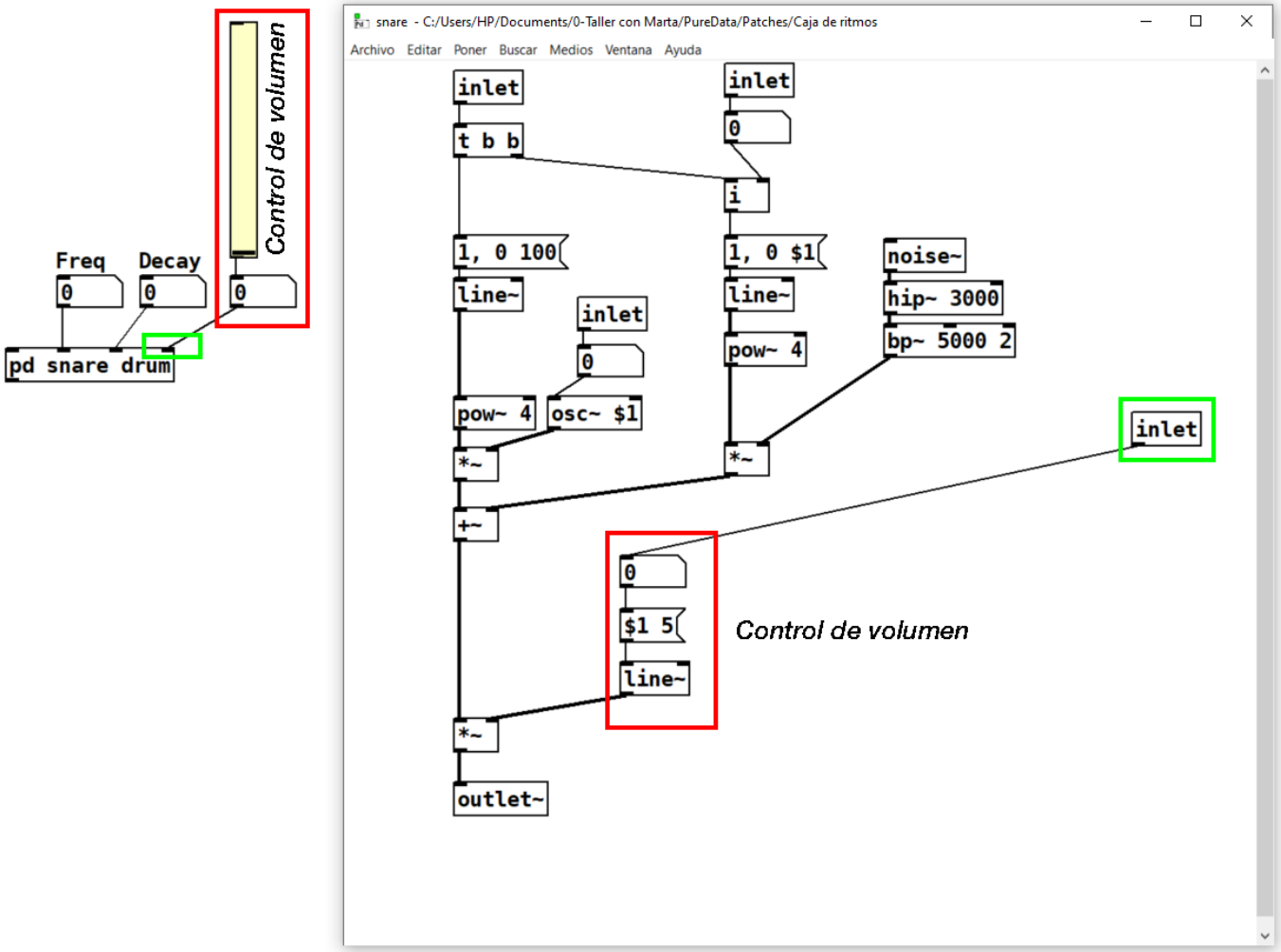

Figura 19. Subpatch para snare drum o caja con el control de volumen general del instrumento marcado en rojo.

6- Mezclar la señal de los tres instrumentos y regular el volumen de la mezcla (master).

Una vez tengamos todos los instrumentos que queramos vamos a **mezclarlos** utilizando el **objeto "*~"** . Enviaremos las señales al inlet izquierdo y a traves del inlet derecho controlaremos el volumen. El **master** es lo que enviamos a nuestros altavoces "**dac~**".

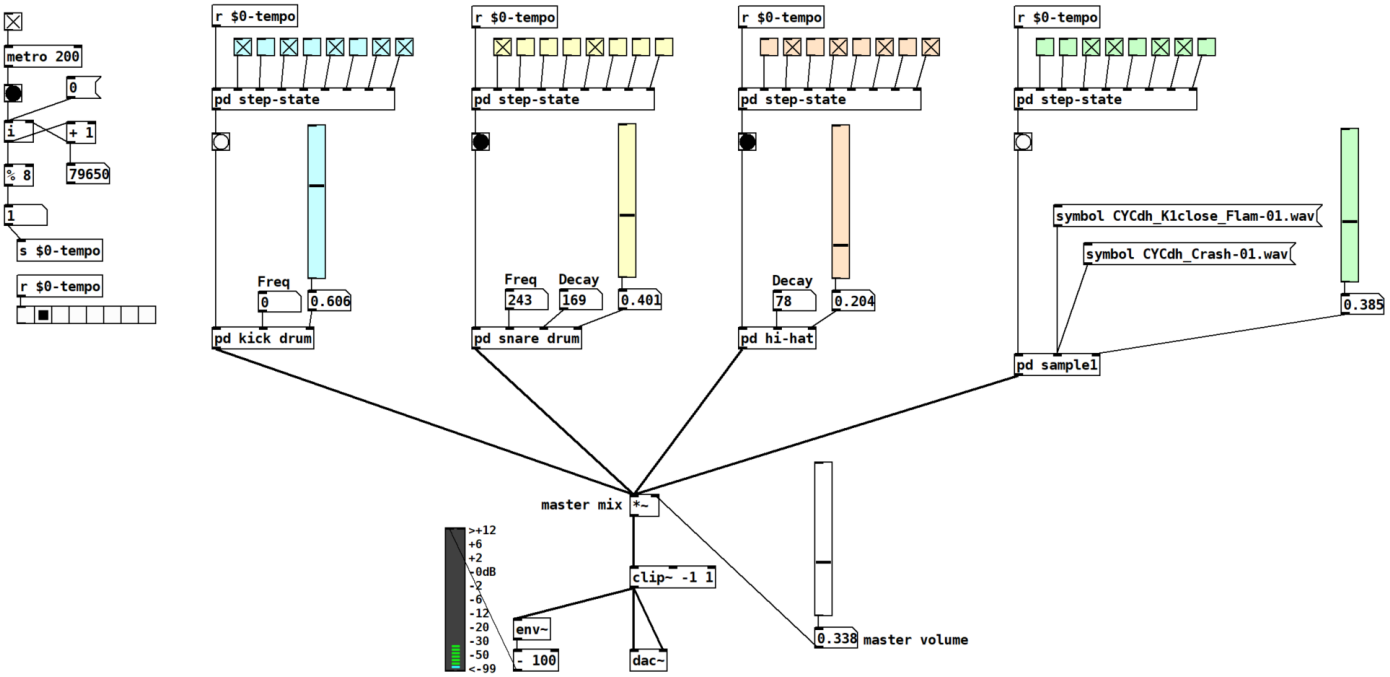

Figura 20. El patch que habéis construido para vuestra caja de ritmos.

> Ejercicio 1: Construye la caja de ritmos como se acabamos de ver. Ha de tener un mínimo de cuatro instrumentos. Puedes incluir más instrumentos o sonidos grabados, incluso una melodía. Explora, **experimenta** y **diviértete**. Una vez tengas tu caja de ritmos graba tres audios utilizando el objeto "**writesf~**" con diferentes patrones de ritmo. Recuerda vimos el objeto "writesf~" en Micrófono: Grabar y re... | Librería CATEDU. Sube tu patch y esas tres grabaciones.

Ejemplos: caja de ritmos de grabaciones de estornudos, de pedos, de palabras grabadas... Explora, **experimenta** y **diviértete**

¿Qué tenemos que entregar?

El ejercicio 1 de la lección. Sube a la carpeta del Moodle de la Practica 12 los siguientes archivos:

- patch ***caja_de_ritmos-vuestronombre_vuestroapellido.pd.***
- grabacion **grabacion_caja_de_ritmos-*vuestronombre_vuestroapellido-1***
- grabacion **grabacion_caja_de_ritmos-*vuestronombre_vuestroapellido-2***
- grabacion **grabacion_caja_de_ritmos-*vuestronombre_vuestroapellido-3***

Si has hecho varios patches diferentes o alguna grabación mas también puedes subirlo. Máximo 3 patches y 6 grabaciones.

Figuras:

Figura 1. Contador de 8 tiempos.

Figura 2. Ocho toggles en fila.

Figura 3. Objeto select.

Figura 4. Patch *select.pd.*

Figura 5. Objeto "select" que vamos a utilizar en esta práctica.

Figura 6. Objeto "sel" que acaba de recibir un 6 en su inlet y envía un bang por el séptimo outlet. (el 6 es el séptimo argumento).

Figura 7. Cuando el objeto "select" reciba un 0 por su inlet, un bang saldrá por su primer outlet y enviara el mensaje que contiene "1" al inlet izquierdo del objeto "==".

Figura 8. El primer paso del control de este instrumento esta activado. Cuando el primer toggle este activado se enviará un "1" al inlet derecho del objeto "==", cuando el toggle este desactivado enviará un "0".

Figura 9. Comprobación del estado del instrumento en el paso en el que se encuentra el contador (primer paso). En ese paso el instrumento esta activado por lo que tras comparar con el "==" y clasificar con el "sel 0 1" obtenemos un bang por el segundo outlet de objeto "sel 0 1".

Figura 10. Patch que compara el estado del instrumento en cada uno de los 8 pasos que genera el contador. Cada vez que un paso del instrumento este activado se enviara un bang para activar el sonido.

Figura 11. Subpatch que contine gran parte de la estructura de la figura 10.

Figura 12. Estructura de control de la figura 10 metida en un el subpatch "pd step-state" de la figura 11.

Figura 13. La orquesta de camino a vuestro patch de pd. https://giphy.com/gifs/perfect-band-member-11MS2pksWxB8SA

Figura 14. Subpatch para el kick drum o bombo.

Figura 15. Subpatch para snare drum o caja.

Figura 16. Subpatch para hi hats o platillos.

Figura 17. El subpatch del kick drum seleccionado.

Figura 18. Subpatch para el sonido procedente de un archivo.

Figura 19. Subpatch para snare drum o caja con el control de volumen general del instrumento marcado en rojo.

Figura 20. El patch que habéis construido para vuestra caja de ritmos.

Revision #1
Created 17 February 2023 11:30:16 by Julia del Río
Updated 17 February 2023 11:30:16 by Julia del Río