Elementos básicos

ELEMENTOS

En Pure Data trabajaremos con 4 tipos/clase de elementos o cajitas: objetos, mensajes, GUI, y comentarios. Estas cajitas pueden tener 0 o más entradas y/o salidas (Pure Data, 2022). Estas cajitas van a ser contenedores de procesos o datos y las conectaremos unas a otras para crear la estructura que nos permita obtener el resultado deseado.

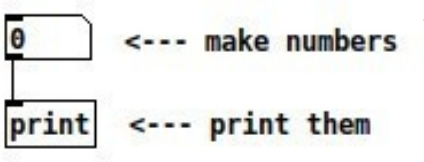

Figura 1. patch que imprime un numero

<--- en este patch hay cuatro cajitas de texto: una cajita de numero (que contiene un cero), una caja de objeto que contiene la palabra "print" y dos comentarios que indican la función de las otras dos cajas.

Un **clic derecho** sobre cualquier elemento nos abre un menu con tres opciones: **Propiedades, Abrir y Ayuda**.

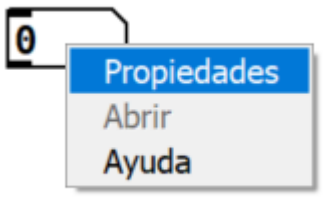

Figura 2. Ventana de opciones que aparece al hacer click derecho sobre cualquier elemento de Pure data.

Ayuda nos va a abrir un patch sobre ese elemento con una explicación sobre su funcionamiento y ejemplos. Esta explicación está en inglés, si no la entendéis podéis utilizar deeple para traducirla:

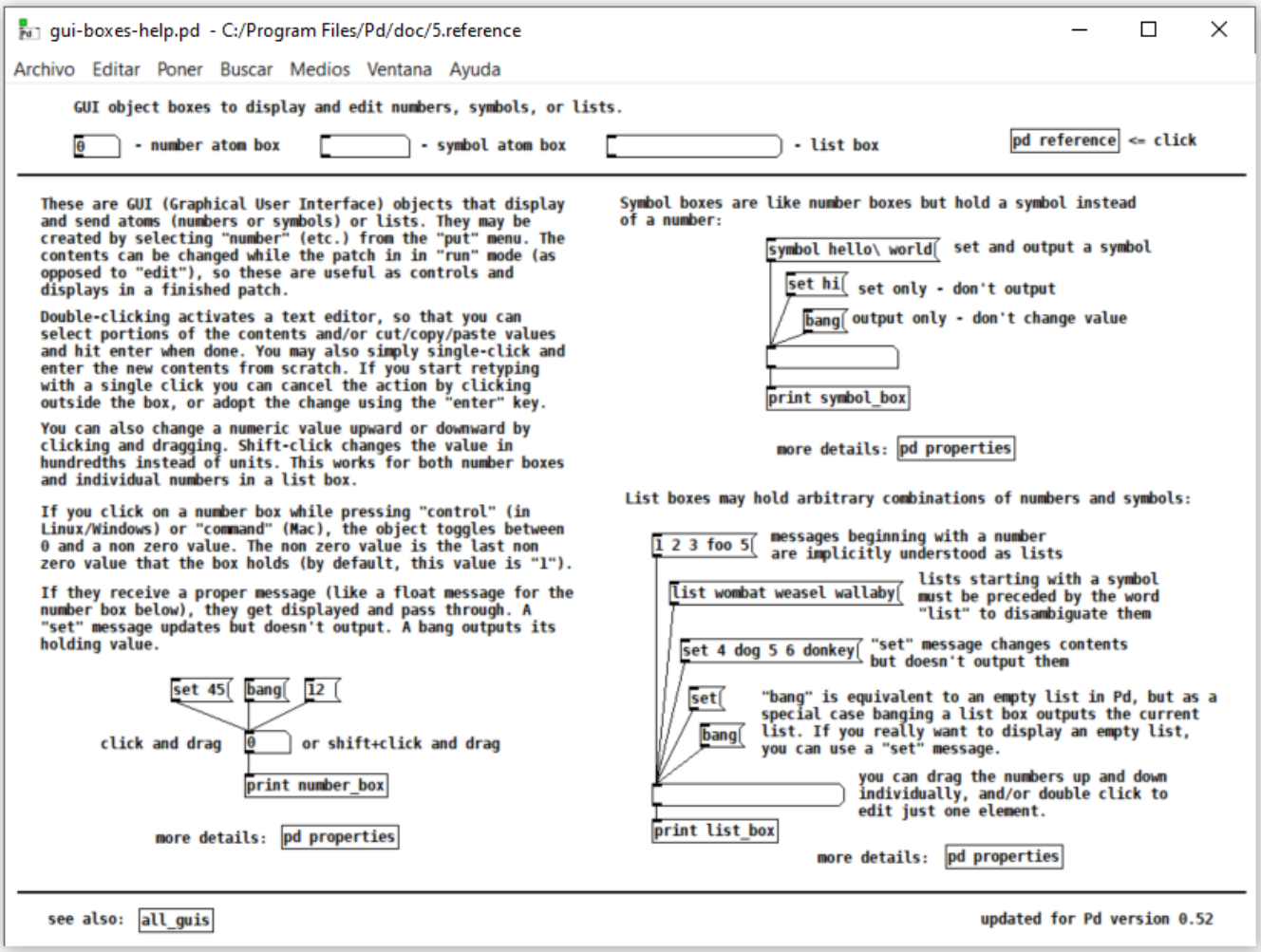

Figura 3. Patch de ayuda de la interfaz gráfica de cajita de numero variable

Mensaje

La cajita de Mensaje se representa por un polígono irregular, un rectángulo con dos piquitos en los vertices derechos. El contenido que introduzcamos en una cajita de mensaje en el modo edición no se puede cambiar mientras el programa está fuera del modo de edición, para modificar el contenido de un mensaje tenemos que hacerlo en el modo edición del patch y este contenido se conserva cuando guardamos, cerramos y volvemos a abrir el patch.

Los mensajes envían su contenido cuando son activados (generalmente utilizaremos un Bang para activar un mensaje, si queremos que un mensaje se envié dos veces tendremos que activar el bang dos veces). Tambien podemos enviarlo clicando sobre él, esto hace que el mensaje sea

considerado también una interfaz gráfica de usuario (GUI). Conectaremos la salida de un mensaje con la entrada de un objeto que sera el destinatario del mensaje. Las salidas de cualquier elemento de pure data se encuentran en la parte inferior de la cajita, y las entradas en la parte superior.

Por defecto los caracteres numéricos, por ejemplo, el numero 5, en un mensaje serán considerados **float**. La palabra "perros" sera considerada **symbols**, y cuando un mensaje contenga varios elementos separados por un espacio tendremos una **lista**. Las listas pueden contener elementos de diferentes clases por ejemplo "5 perros"

Para colocar un mensaje en nuestro patch: **menu horizontal>Poner>Mensaje** o utilizando en comando **"Ctrl+2"**

Objeto

Los objetos se crean escribiendo texto en una cajita de objeto, y es ese **texto** el que va a **crear un objeto** determinado con una función específica. Este texto se divide en ***atoms*** separados por un espacio. El primer *atom* indica que tipo de objeto sera creado, los *atoms* siguientes, llamados ***argumentos*** *de creación,* indican a Pd como inicializar el objeto, por ejemplo que valor/valores o en qué estado comienza el objeto (Pure Data, 2022). Hacerlo mas claro, añadir imagen con explicacion

La cajita de objeto se representa por un rectángulo, en la parte superior del rectángulo se encuentran las **entradas**, también llamadas inputs o **inlets**, marcadas con una pequeña línea gruesa. En la parte inferior se encuentran las **salidas**, outputs o **outlets**, marcadas cada una de ellas también con una pequeña línea gruesa. Dependiendo del objeto el número de entradas y salidas variara, y hay objetos que no tienen salida, solo tienen entrada como el "print". Para referirnos a las entradas o salidas empezaremos a contar por la parte izquierda, por ejemplo, el objeto suma que veis a continuación tiene dos entradas, la primera entrada es la que se encuentra a la izquierda y la segunda entrada es la que se encuentra a la derecha.

el "+" especifica el tipo de objeto que sera esta cajita, indica que es un objeto de suma y su función sera esa, sumar. Este objeto tiene un argumento que es el "13" indica que cantidad se va a sumar al número que llegue por la entrada izquierda del objeto (Pure Data, 2022):

Entrada/input izquierda + 13 = output/salida

La cantidad a sumar sera el argumento inicial del objeto ,13, a no ser que introduzcamos otro número por la entrada derecha del objeto. El numero introducido remplazara al 13 y quedara almacenado en el objeto suma. Aunque visualmente en el objeto sigamos viendo "+ 13", su

contenido es "+ Ultima Entrada/inlet derecha"

Entrada/input izquierda + Ultima Entrada/input derecha = output/salida

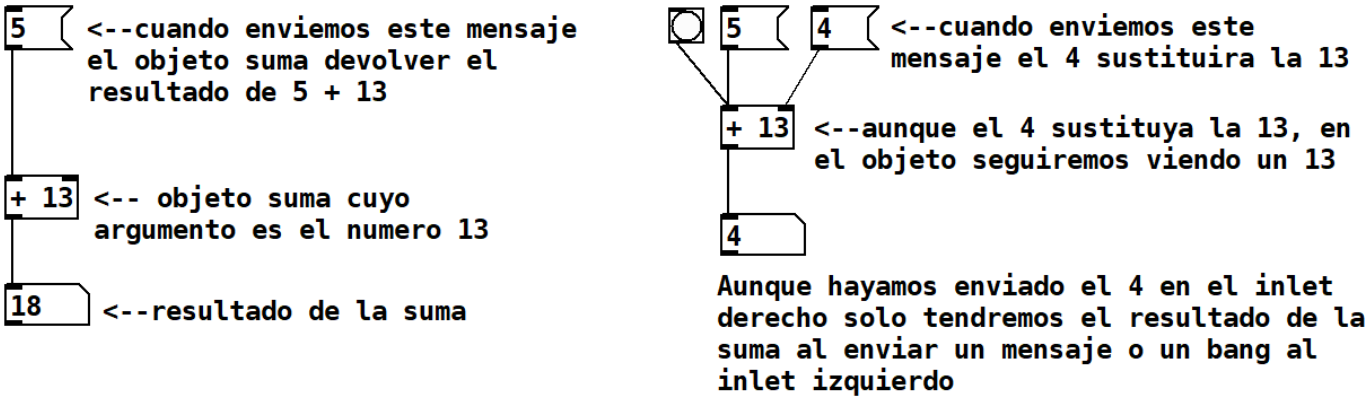

Figura 4. patch *Suma.pd*

¿Hay alguna diferencia entre la entrada izquierda del objeto suma y la entrada derecha?

Si las hay, y es importante que las conozcamos. Como regla general el primer inlet de un objeto sera un **"hot" inlet** o **entrada caliente** y el resto de inlets serán **"cold" inlets** o **entradas frías**. Un valor enviado a una entrada fría sera almacenado por el objeto, pero no activará el proceso de ejecución del objeto y este no emitirá ningún valor por su outlet hasta que la entrada caliente reciba un bang o un valor/mensaje.

Los atoms son números o símbolos. Todo aquello que no sea considerado un numero valido sera interpretado por el programa como símbolo. Los números validos pueden contener decimales (12, 15.6, -783) o estar escritos con notación exponencial (7.8e3 = quiere decir 78 multiplicado por 10 tres veces = 7800) (Pure Data, 2022). Los exponenciales negativos dividen entre 10 (69.3e-4= 0,00693) (Pure Data, 2022).

Entre números no validos que son leídos como símbolos podemos encontrar expresiones como "+5" "0..6" "casa" y el programa reconoce como símbolos diferentes "hola" , "Hola" y "HOLA" luego diferencia entre mayúsculas y minúsculas (Pure Data, 2022).

Los números validos pueden ser de dos tipos:

- Integer o int = números enteros, sin decimales. Como el numero 5 o el 22.
- Float = números con decimales. Como el numero 10,85 o el 22,6.

Para colocar un objeto en nuestro patch: **menu horizontal>Poner>Objeto** o utilizando en comando **"Ctrl+1"**

Utilizando un objeto podemos crear una variable que nos permita almacenar datos. ¿Os acordáis de lo que era una variable? Lo vimos en la página "Elementos que debemos de conocer"

Tendremos que saber qué tipo se variable queremos almacenar, ya que crearemos un objeto para ese tipo de variable. Si queremos almacenar un numero entero crearemos un objeto "int" o "i", son el mismo objeto, pero podemos escribirlo de estas dos maneras, una de ellas ocupa mucho menos espacio y nos va ser mucho más cómoda cuando tengamos muchos objetos en nuestro patch. Si queremos almacenar un numero decimal crearemos un objeto "float" o "f".

E :h *Variables-Integer-F*

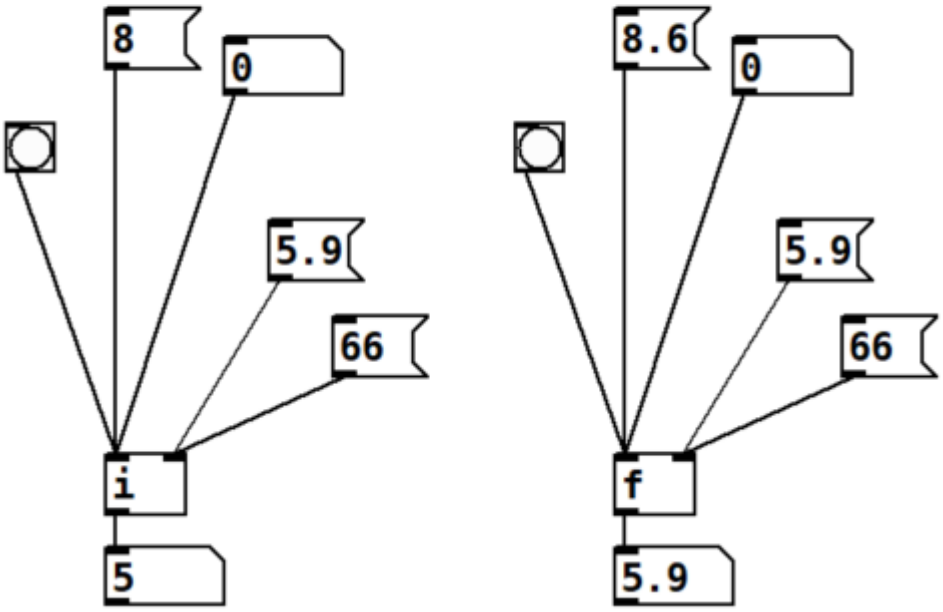

Figura 5. patch *Variables-Integer-Float.pd*

Los objetos de variables int y float funcionan siguiendo el esquema de entrada caliente y entrada fría que explicábamos antes. Los valores que lleguen al inlet derechos serán almacenados, pero no serán enviados hasta que el inlet izquierdo reciba un bang. Los valores recibidos en el inlet izquierdo serán almacenados y enviados inmediatamente al ser recibidos. Podéis comprobarlo en el patch *Variables-Integer-Float.pd.*

Para más información de cualquier elemento de Pure Data accederemos a el **patch de ayuda** que encontraremos clicando sobre el elemento con el botón derecho y seleccionando Ayuda. Este patch en ingles explica la funcionalidad y posibilidades del elemento.

Cuando trabajemos con **señales** de audio utilizaremos **objetos específicos de audio**. Estos objetos siempre contienen el símbolo de **virgulilla : "~"** y a diferencia de los objetos de control/Data flow que realizan su función únicamente cuando sucede un evento y los activa (ejemplo: un el clic sobre un mensaje, la llegada de un símbolo a un print), los objetos de audio procesan y envian señal constantemente sin necesidad de eventos que los activen. La señal de audio se diferenciar como una **línea más gru**

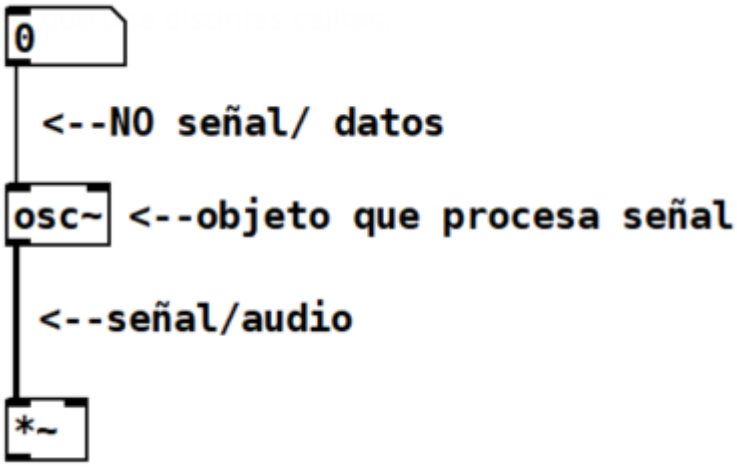

Figura 6. patch *Senial.pd*

Cada objeto dependiendo de su función va a esperar un tipo de inlets, y si intentamos conectar un envió con un inlet cuyo tipo no corresponde, la conexión no se hara, veremos que la línea no queda conectada, o veremos un error marcado en rojo en la zona de impresión de la ventanita de pd:

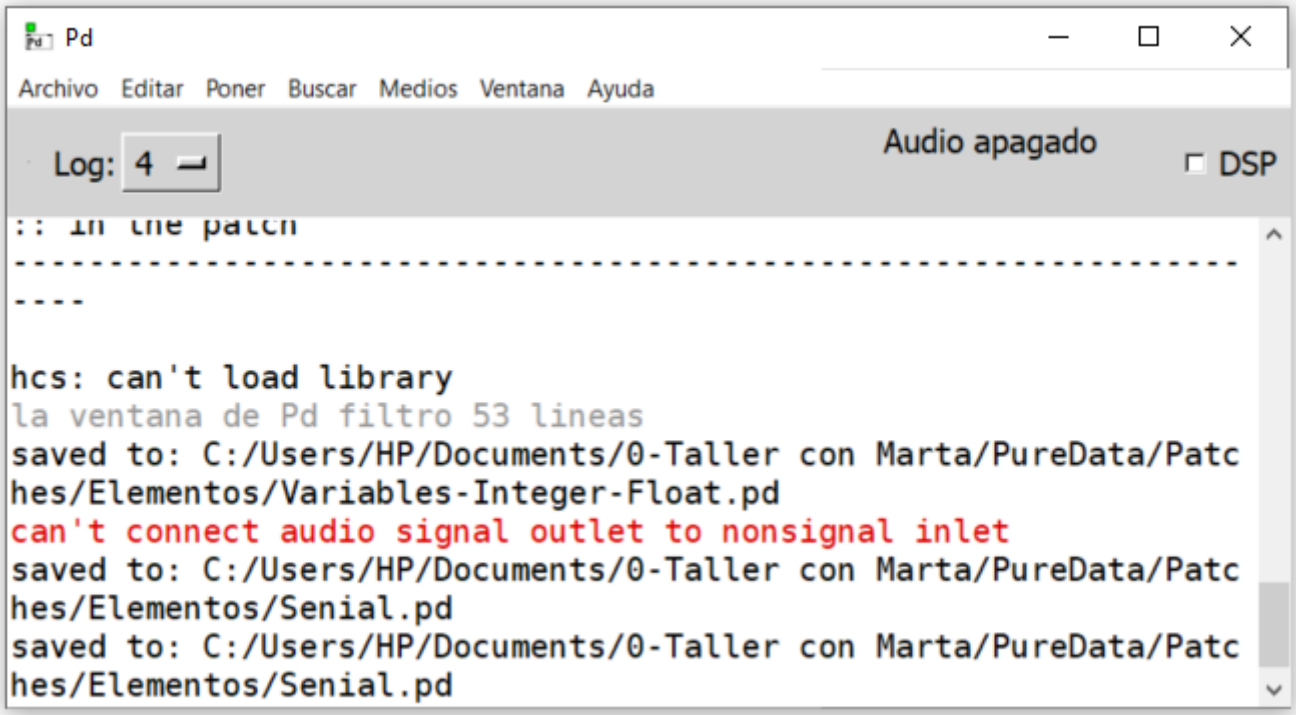

Figura 7. Ventana principal de Pd con mensajes en la consola.

En la ventana superior el mensaje en rojo dice que no podemos conectar una salida de una señal de audio a una entrada que no es de señal. Este mensaje ha aparecido al intentar conectar la salida del objeto "osc~" con un bang.

Si no sabéis el tipo que inlets que espera un objeto podéis entrar en su patch de ayuda y ahi abrir el subpatch "pd reference":

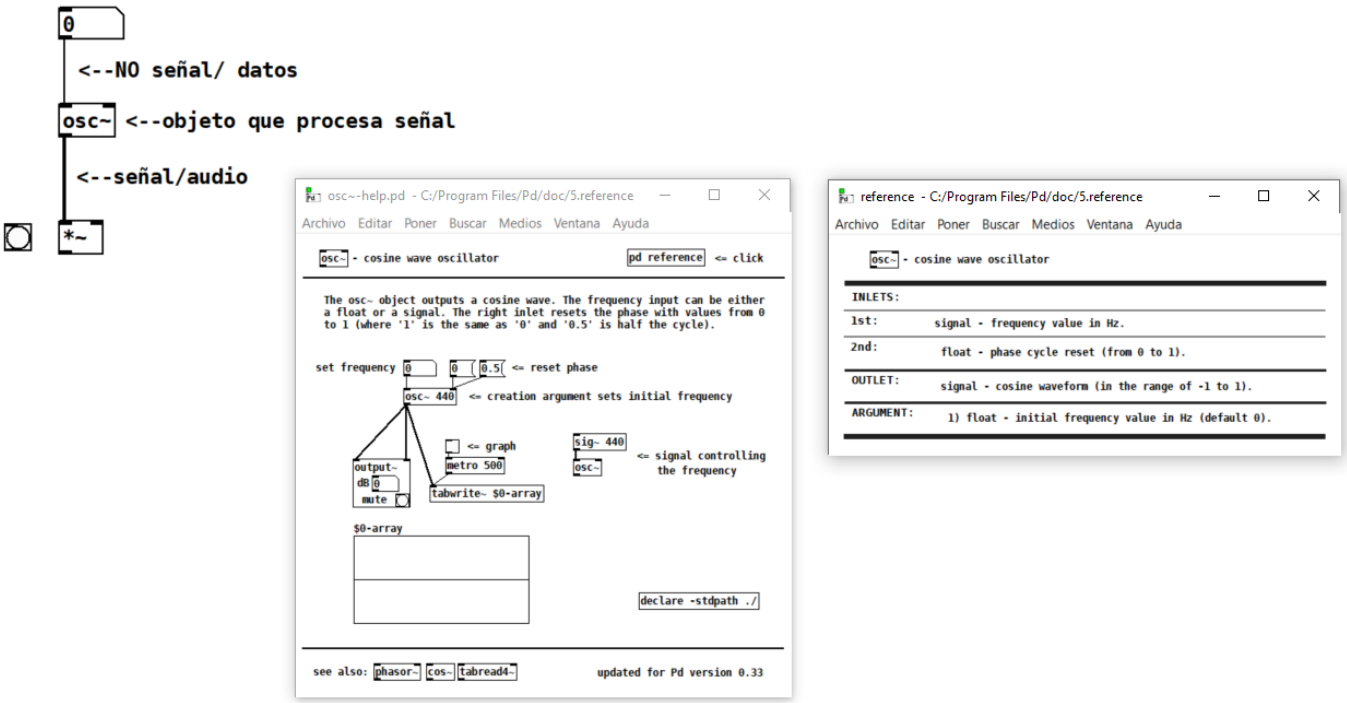

Figura 8. patch de ayuda y subpatch "pd reference" del objeto "osc~"

Lista de objetos:

Lista de objetos de Pure Data Vanilla + librerías externas:

https://puredata.info/docs/ListOfPdExternals/

Aqui os dejo una lista con objetos de Pure Data y su función. Tambien podréis ver si pertenecen a Pure Data Vanilla, que es el programa base sobre el que trabajamos o si forman parte de alguna librería, si forman parte de alguna librería los llamaremos objetos extended y tendremos que instalar esa librería para poder utilizar esos objetos. A utilizar librerías aprenderemos más adelante.

GUI (Graphical User Interface/ Interfaz Gráfica de Usuario)

Las interfaces graficas de usuario (GUI) son elementos cuya apariencia puede variar cuando el programa está en funcionamiento, a diferencia de los objetos y mensajes que por lo general tienen una apariencia estática. Alguno de estos elementos es interactivo y nos van a permitir modificar parámetros desde el ratón y/o visualizar cambios o estados cuando el programa este en funcionando. Tienen diferentes formas y funciones. Estas son algunas de las más utilizadas:

Cajita de número

La cajita de número se representa por un rectángulo truncado en su esquina superior derecha. Este elemento nos permite almacenar y visualizar el número que llegue por el inlet a través de un mensaje u objeto. Podremos modificar el numero almacenado con el ratón clicando y arrastrando arriba o abajo cuando el programa este en ejecución. Este elemento enviara el número que contiene cuando este varíe o cuando reciba un evento ej. (bang). Si el número no varía y no recibe ningún bang, la cajita de número no envía nada y a diferencia del mensaje que con un clic hacemos que se envíe, sobre este, el click de ratón no hará que la cajita de numero envíe su contenido.

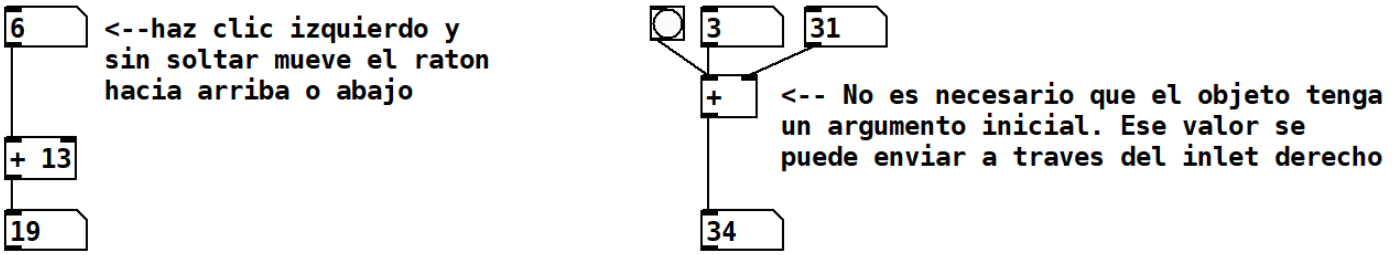

Figura 9. patch *Suma.pd*

Clic derecho en el objeto>**Propiedades** = nos permite modificar tamaño, **limites**, etiquetas ...

Para colocar una caja de numero en nuestro patch: **menu horizontal>Poner>Numero** o utilizando en comando **"Ctrl+3"**

> Ejercicio 1: Realizar una suma de dos números contenidos en dos mensajes e imprimir el resultado en la ventana principal.

> Ejercicio 2: Realizar una suma utilizando dos cajitas de numero variable. Ves alguna diferencia entre el inlet derecho de la suma y el inlet izquierdo? Si la hubiere como afecta esta diferencia al output del objeto suma?

Bang

Este elemento es un botón que nos permite enviar un **"impulso"** o "**mensaje de bang"** cuando es activado: bien clicando sobre él, o cuando recibe cualquier mensaje en su entrada, da igual el contenido del mensaje. Frecuentemente utilizaremos el bang para **activar otros elementos** y nos va permitir **visualizar cuando hay flujo** de datos entre otros objetos. El mensaje de bang le dice a otros objetos: actúa! para que realicen su función. Y a nosotros nos dice que algo ha llegado.

Su forma GUI se representa con un cuadrado con un circulo circunscrito, cuando el bang se activa el circulo cambia momentaneamente de color, nos indica que ha recibido cualquier mensaje y enviado un mensaje de bang. En la parte superior izquierda tiene una entrada y en la parte inferior izquierda una salida.

Clic derecho en el objeto>**Propiedades** = nos permite modificar **tamaño, color, etiquetas** ...

Para colocar un bang en nuestro patch: **menu horizontal>Poner>Bang** o utilizando en comando **"Shift+Ctrl+B"**

El bang se puede crear también como objeto: "bang" o "b" y realiza la misma función que su version GUI pero no permite al usuario activarlo desde el ratón ni visualiza su actividad. Podremos encontrarlo también en forma de mensaje con el texto "bang"

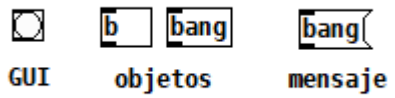

Figura 10. Posible formas que tiene el bang en Pd. (Interfaz grafica, Objetos y Mensaje)

Ejercicio 3: Abrir el patch de ayuda del objeto suma (binops-help.pd). Dentro del patch de ayuda del objeto suma, abrir el subpatch "pd reference" para obtener más información acerca del funcionamiento de los objetos "operadores aritméticos". Utiliza deeple para traducir el texto en ingles del patch de ayuda que no comprendas. Responde las siguientes preguntas:

3.1 - Que sale del output derecho del objeto trigger ([t b f]) en el primer ejemplo del patch? ¿Y del output izquierdo? Utiliza el objeto print para comprobarlo.

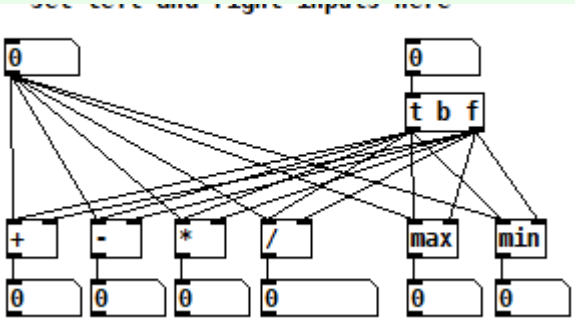

Figura 11. primer ejemplo del patch de ayuda del objeto "+"

3.2 - Que utilidad tiene el objeto trigger ([t b f])? Abre el patch de ayuda de este objeto para descubrilo.

3.3 - Qué pasaría si prescindiéramos del objeto trigger en este ejemplo y conectáramos la cajita de numero variable del lado derecho directamente al inlet derecho de los operadores?

Ejercicio 4: En el caso anterior sustituye el objeto trigger ([t b f]) por un bang y reconecta las cajitas para obtener la misma funcionalidad que con el objeto trigger ([t b f]).

Ejercicio 5: ¡Reto! Construye un programa en Pd que imprima el resultado de la siguiente ecuación: (2X + 4) x (3Y -20). Pista: X e Y serán cajitas de numero variable.

Cuando queramos que un bang se envíe automáticamente al abrir un patch utilizaremos el objeto "**loadbang**":

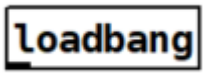

Figura 12. Objeto "loadbang"

Toggle

Este objeto GUI tiene **dos estados: apagado y encendido**. Podemos cambiar el estado del objeto clicando sobre él o enviando un valor o mensaje a su inlet.

Cuando recibe un valor igual a 0 en su entrada, el estado del toggle cambiara a apagado y emitira por su la salida el valor 0. Visualmente reconoceremos el estado de apagado como un cuadrado blanco vacío.

Al cambiar el estado a encendido, clicando sobre el o cuando recibe en su entrada un numero distinto de 0, la salida del toggle emitirá ese numero distinto de cero y reconoceremos este estado por una cruz dentro del cuadrado. Un toggle encendido, por defecto emitirá un 1, a no ser que reciba otro valor numérico.

Un clic de ratón sobre el objeto o un mensaje de bang cambian el estado del objeto, si este está encendido cambiara a apagado y emitirá un 0, y si esta apagado cambiara a encendido y emitirá un 1. Un input numérico distinto de zero posiciona siempre el toggle en estado de encendido y un input igual a cero lo cambia o mantiene en estado apagado. El toggle emite un valor por su salida

cada vez que el valor de entrada cambie aunque este valor no haga que cambie su estado. Frecuentemente utilizaremos el toggle para **activar o cambiar de estado otros elementos** y nos va permitir **visualizar el estado** de esos elementos.

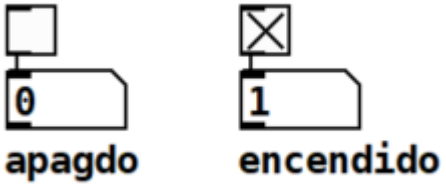

Figura 13. patch *Toggle.pd.* Toggle apagado y su output, Toggle encendido y su output predeterminado.

Si el input del toggle es un numero distinto de 0 y distinto de 1, por ejemplo, el 5, el toggle estará en el estado de encendido y emitirá un 5 por su output, si después de ese 5 enviamos un 6 al input del toggle el estado de este no cambiará y permanecerá encendido, pero emitirá el numero 6.

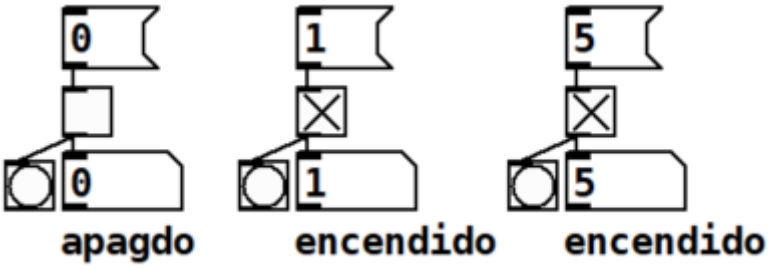

Figura 14. patch *Toggle.pd.* Estado de Toggle y que outlet proporciona en función de su inlet.

Para cambiar el estado del toggle sin que emita ningún valor a su salida se utiliza el mensaje "set 0" para apagarlo y "set (valor distinto de 0)" para encenderlo. (hacer un gif de esto)

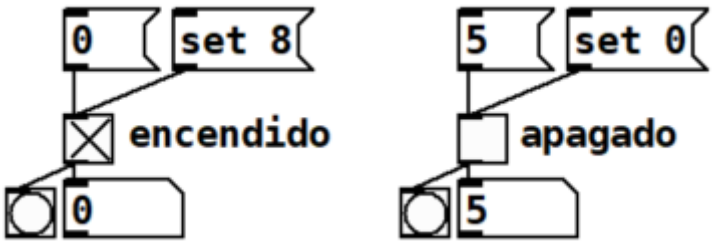

Figura 15. patch *Toggle.pd.* Mensaje "set" para cambiar el estado del toggle sin que emita ningún valor.

Para colocar un toggle en nuestro patch desde el menu horizontal en **Poner>Toggle** o utilizando en comando **"Shift+Ctrl+T"**

Clic derecho en el objeto>**Propiedades** = nos permite modificar **tamaño, color, etiquetas** ...

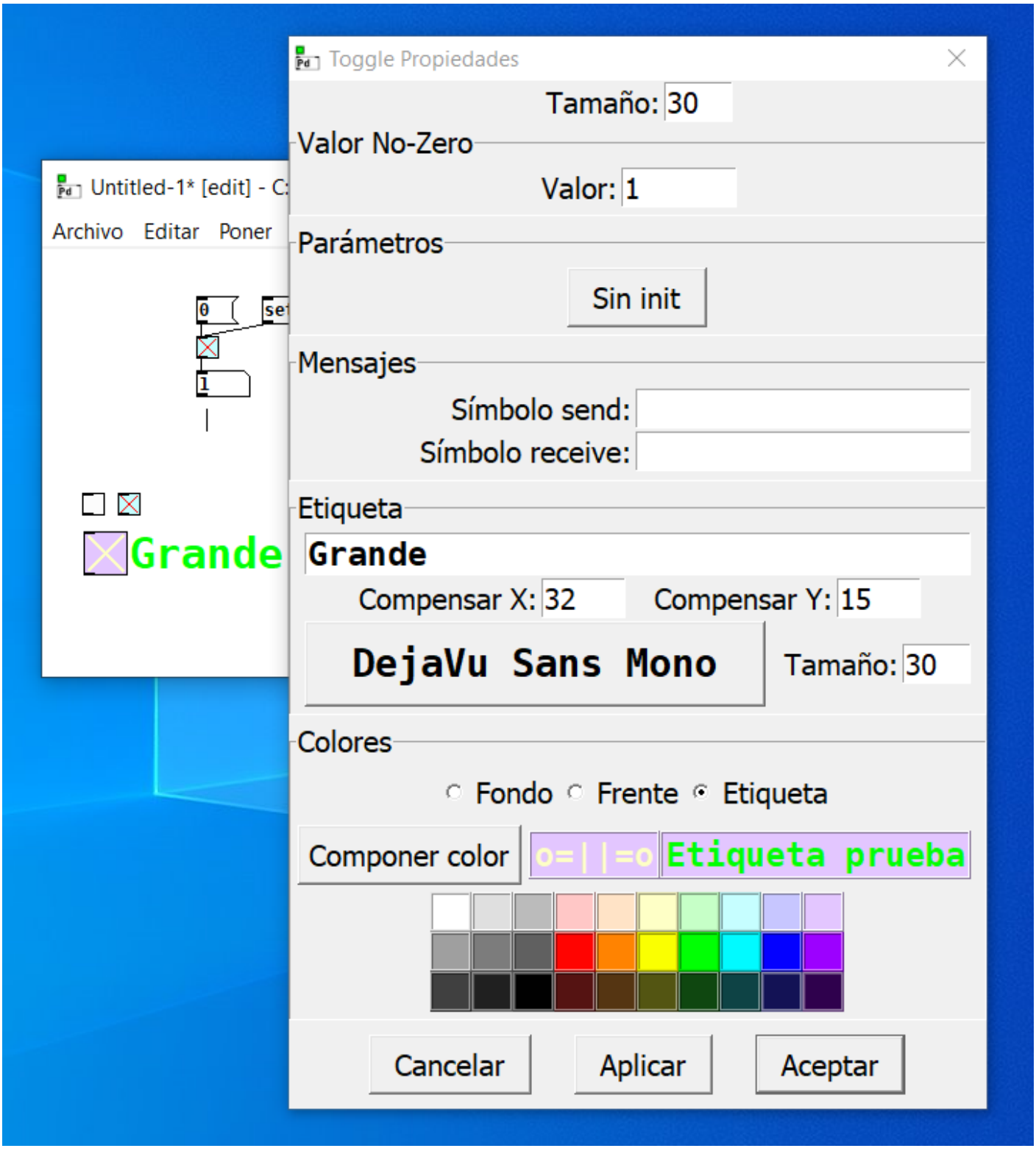

Figura 16. Ventana de propiedades de Toggle.

En el patch de Lista de objetos (help-intro.pd) que se encuentra en **menu horizontal>Ayuda>Lista de objetos** hay un sub-patch llamado **"all_guis"** que nos muestra

todas las interfaces graficas de Pd Vanilla.

comment Comentarios

Los comentarios visualmente no aparecen en ninguna cajita ni tienen entradas/inputs ni salidas/outputs. El texto introducido en los comentarios es información que no sera interpretada por la maquina como elementos transformadores del programa/patch que estamos construyendo. Estos comentarios contienen **información dirigida a las personas**, notas que tomareis para vosotros mismos relacionadas con el funcionamiento del patch que estéis construyendo o notas para facilitar a otras personas la comprensión del patch que habéis creado. Son muy útiles para escribir **recordatorios o explicaciones** durante el proceso de trabajo y para **etiquetar** las diferentes partes de las estructuras que hagáis.

comment

Para colocar un comentario en nuestro patch desde el menu horizontal en **Poner>Comentario** o utilizando en comando **"Ctrl+5"**. Para editar un comentario clicamos sobre el cuando estemos en el modo edición.

Este capítulo ha sido redactado tomando como guía el Capitulo 2 del Manual de Pd.

Referencias:

Pure Data (2022). Pd Manual chapter 2: theory of operation. In *Pure Data* (Version 0.52.2). [Computer software]. Pure Data. https://puredata.info/downloads/pure-data

http://puredata.info/docs/manuals/pd/

Figuras:

Figura 1. patch que imprime un numero.

Figura 2. Ventana de opciones que aparece al hacer click derecho sobre cualquier elemento de Pure data.

Figura 3. Patch de ayuda de la interfaz gráfica de cajita de numero variable

Figura 4. patch *Suma.pd*

Figura 5. patch *Variables-Integer-Float.pd*

Figura 6. patch *Senial.pd*

Figura 7. Ventana principal de Pd con mensajes en la consola.

Figura 8. patch de ayuda y subpatch "pd reference" del objeto "osc~"

Figura 9. patch *Suma.pd*

Figura 10. Posible formas que tiene el bang en Pd. (Interfaz grafica, Objetos y Mensaje)

Figura 11. primer ejemplo del patch de ayuda del objeto "+"

Figura 12. Objeto "loadbang"

Figura 13. patch *Toggle.pd.* Toggle apagado y su output, Toggle encendido y su output predeterminado.

Figura 14. patch *Toggle.pd.* Estado de Toggle y que outlet proporciona en función de su inlet.

Figura 15. patch *Toggle.pd.* Mensaje "set" para cambiar el estado del toggle sin que emita ningún valor.

Figura 16. Ventana de propiedades de Toggle.

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