



# Introducción a Pure Data - 01

Introducción, bloques básicos y visualización.

---



FACULTAD DE  
INGENIERÍA

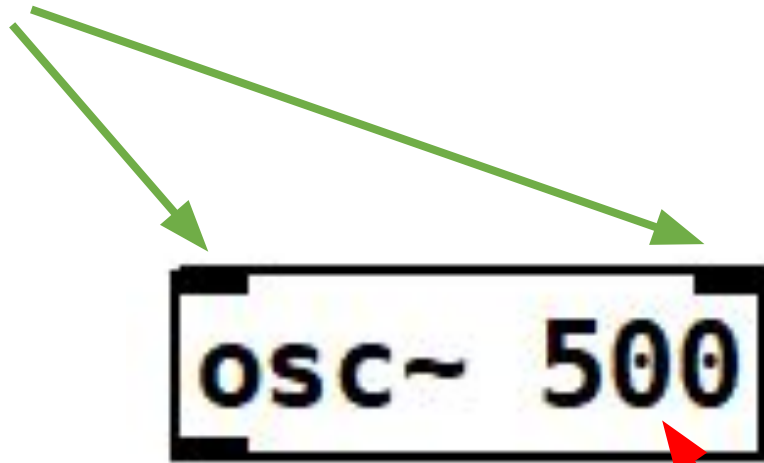


UNIVERSIDAD  
DE LA REPÚBLICA  
URUGUAY

Taller de Procesamiento de Audio y Video con Pure Data/Gem

# Composición de los objetos

**Inlets**



**Outlets**

**Parámetros por defecto**

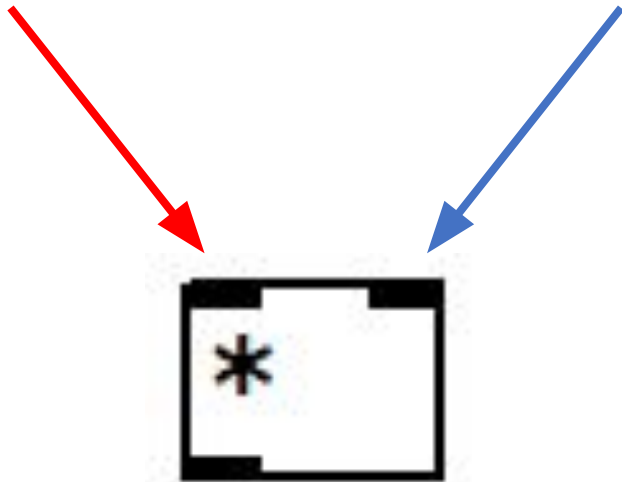
**Nota:**

La conexión entre objetos se hace haciendo click sobre el outlet del primer objeto y arrastrando hasta el inlet del siguiente.

# Composición de los objetos

Hot Inlet

Cold Inlet



**Nota:**

El **Hot Inlet** es el que activa el objeto, ya sea con un *bang* o actualizando su valor.

Si se modifica el valor en el **Cold Inlet**, el objeto no se activará.

## Objetos de datos numéricos y comunes



Número



Suma



Multiplicación



Resta



División

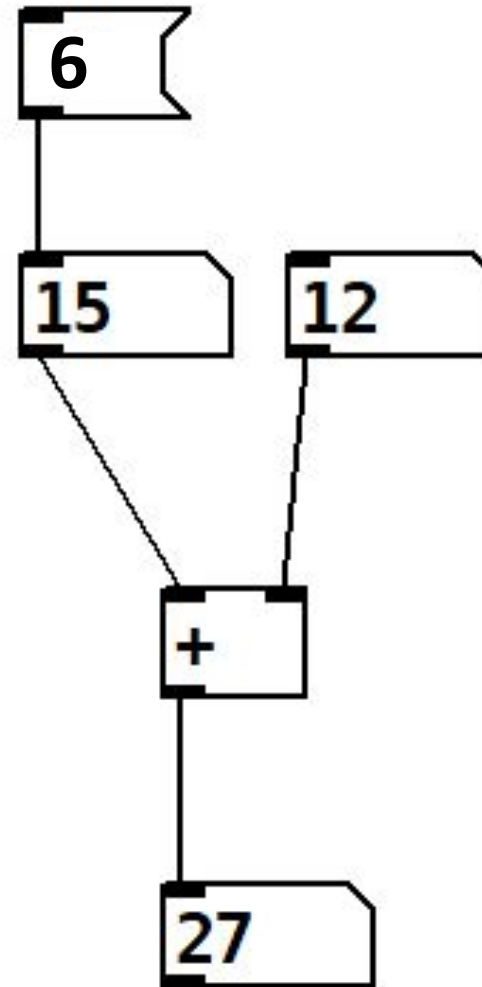


Mensaje

# Ejemplo: Sumador

## Objetivos:

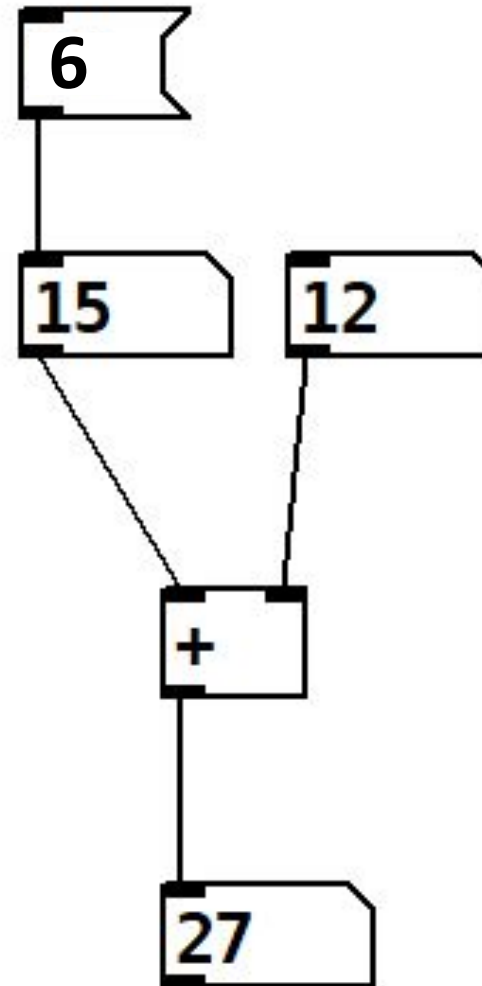
- Identificar cómo interactúan los bloques de números (hot y cold inlet).
- Ver como funciona el mensaje.



# Ejemplo: Sumador

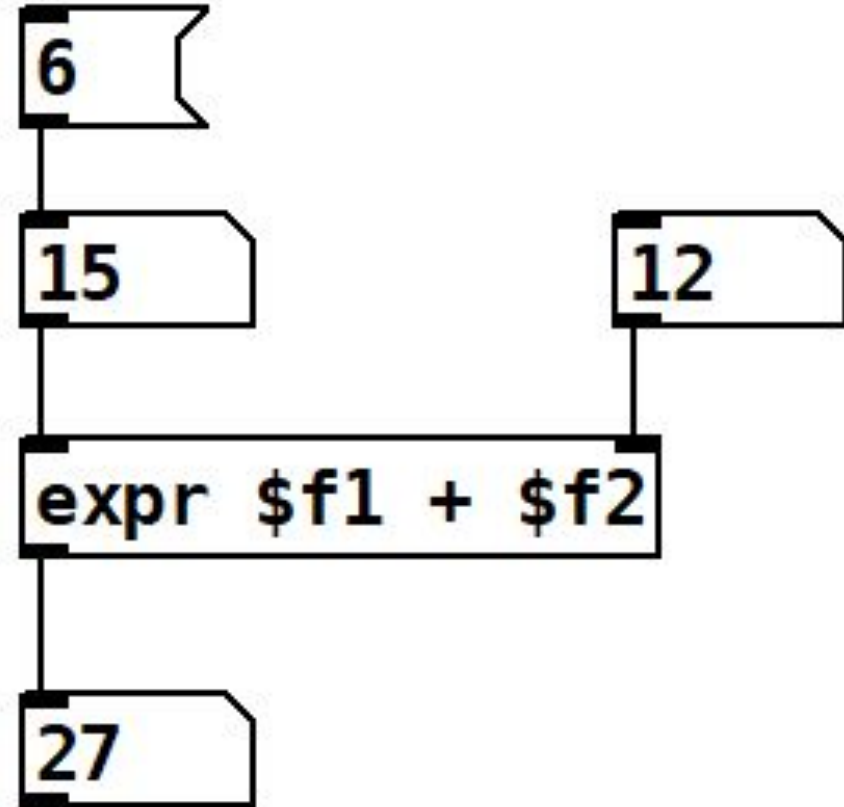
## Observaciones:

- Modificando el número en el Hot inlet se modifica el resultado
- Modificando el número en el Cold inlet no se modifica el resultado hasta activar el Hot inlet.
- El mensaje envía el número que tiene escrito hacia el bloque objeto que tiene conectado.



## Ejemplo: Sumador con *expr*

- También se pueden expresar fórmulas matemáticas mediante el objeto *expr*.
- \$f1 es el valor que recibe por el primer inlet, \$f2 el que recibe por el segundo, etc.
- La expresión entiende de tipos. \$f# se usa para flotantes, \$i# para enteros y \$s# para símbolos.
- También existe *expr~* para trabajar con señales, en ese caso se utiliza \$v#.



## Objetos de señales de audio



Oscilador (generador de tonos)



Generador de ruido



Multiplicador (modifica amplitud)



**DAC** - Conversor digital a analógico  
- Salida de audio (parlantes)

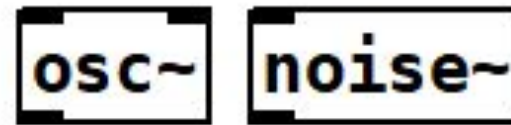


**ADC** - Conversor analógico a digital  
- Entrada de audio (micrófono)



# Objetos de datos numéricos VS señales de audio

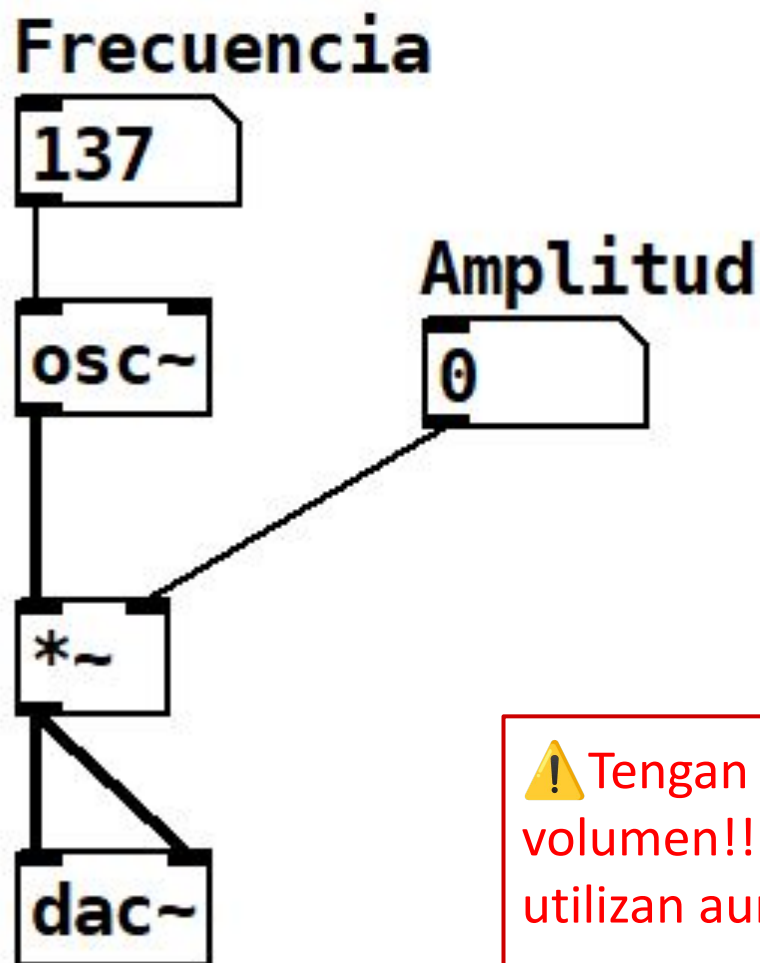
Los bloques de señales siempre incluyen ~



## Ejemplo: Oscilador

Un objeto *número* varía la frecuencia de la señal sinusoidal creada por el *osc~*.

Un objeto *número* (entre 0 y 1) que multiplica a la señal para controlar su volumen.

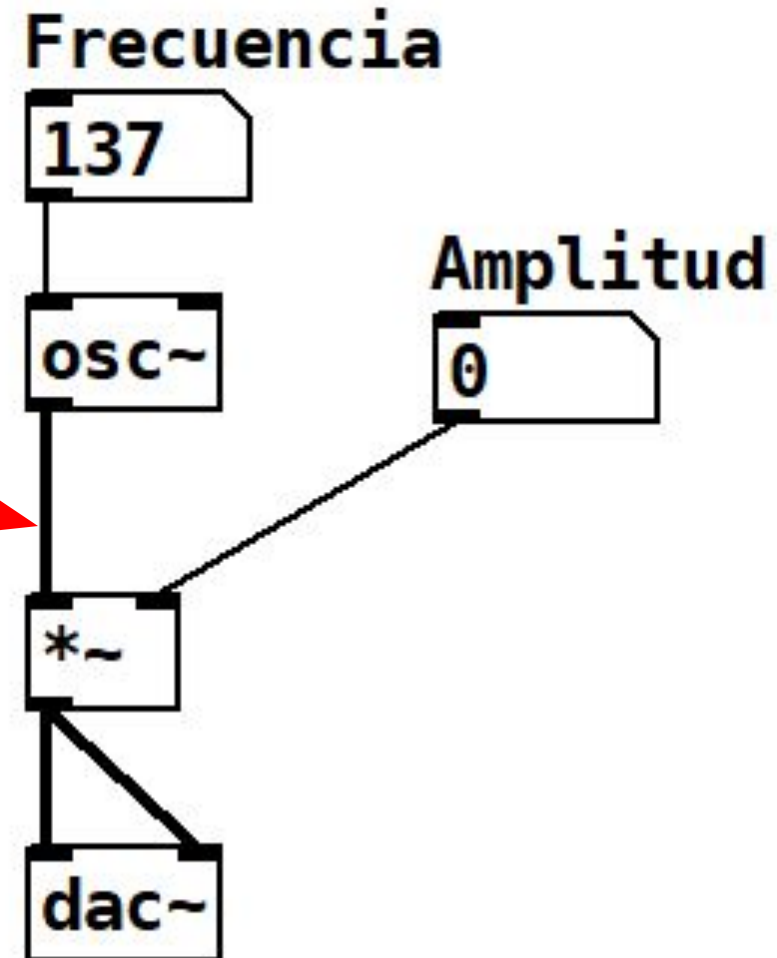


⚠ Tengan cuidado con el volumen!! En especial si utilizan auriculares.

Sugerencia:  
Poner amplitud entre 0 y 1.

# Mas diferencias entre datos numéricos y señales de audio

Los enlaces por donde “viajan” señales de audio, se marcan con un **enlace grueso**.



## Bloques controladores



**Bang:**  
Envía un pulso



**Toggle:**  
Habilita o no algunos objetos

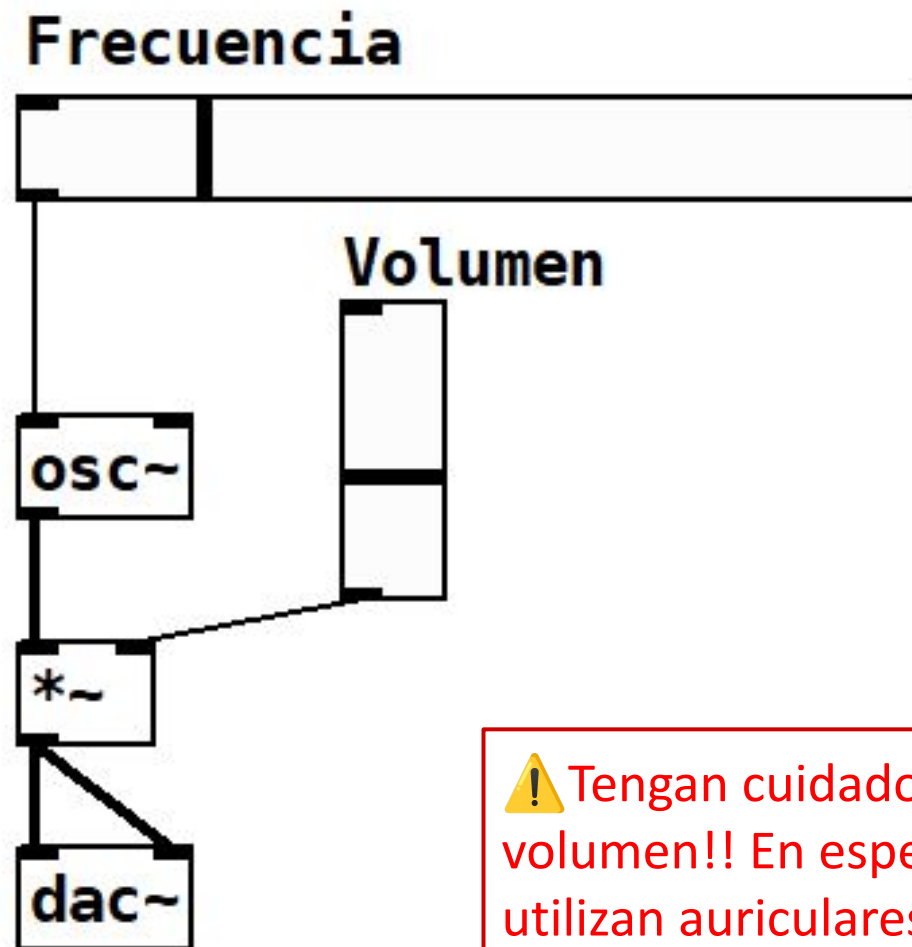


**Slider:**  
Similar al bloque número, pero el número se define de manera gráfica dentro de los márgenes que se configuren.

# Ejemplo: Oscilador

(Variación)

Variación utilizando sliders  
en lugar de los objetos  
*Número*.



⚠ Tengan cuidado con el volumen!! En especial si utilizan auriculares.

Sugerencia:  
Poner amplitud entre 0 y 1.

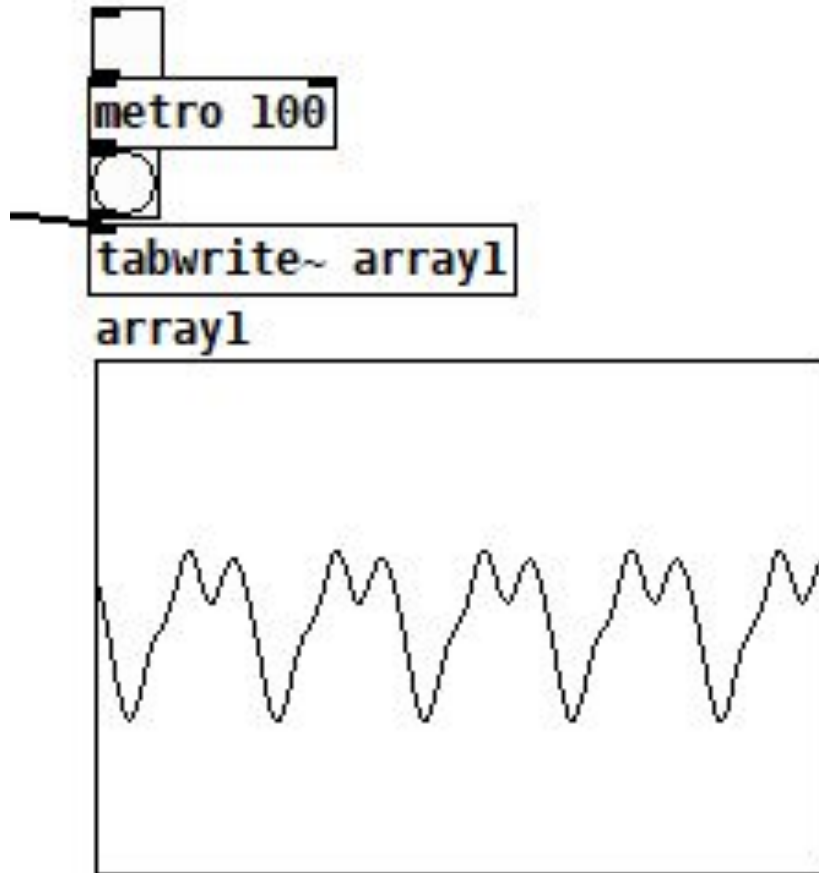
# Visualización con arrays

## Observación:

Para poder visualizar el array hay que “actualizarlo”.

Por ese motivo hay que poner un *bang* o *metro* conectado al `tabwrite~`. No es necesario poner ambos.

El *bang* sería actualización manual y el *metro* automática.



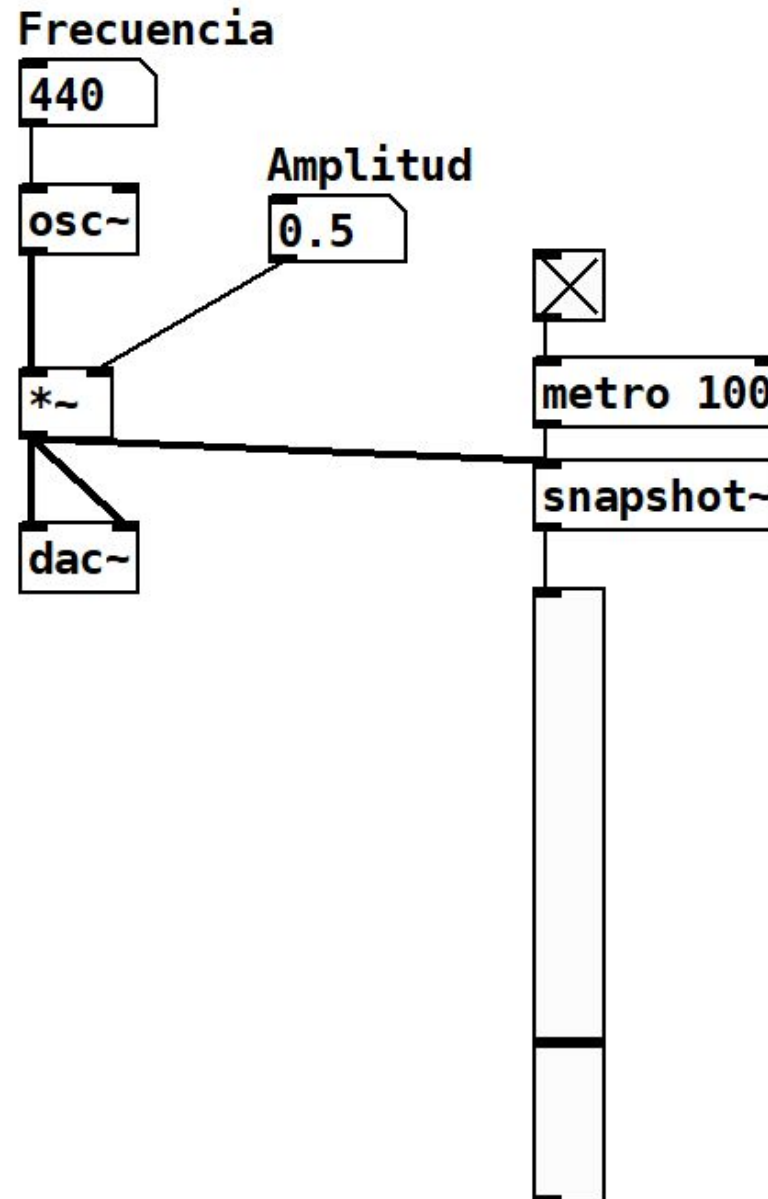
**Nota:** se profundizará en los arrays más adelante.

# Visualización con sliders

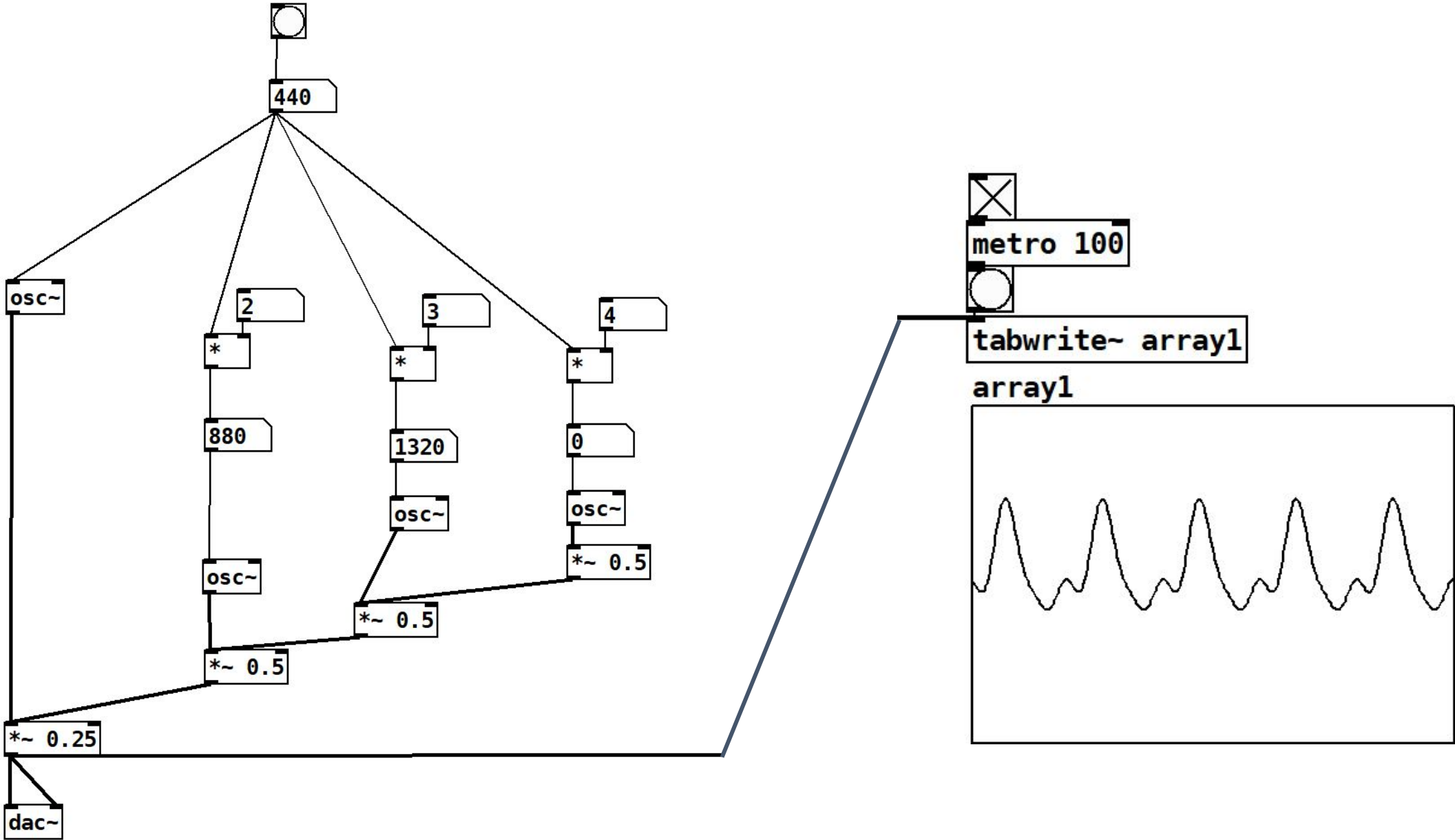
Se puede visualizar un número conectándolo al inlet de un slider.

Esta opción no permite conectar señales de audio, pero podemos observar su amplitud si la conectamos a través de un `snapshot~` como muestra el ejemplo.

Resumidamente, `snapshot~` convierte señales a números.



# Ejemplo





# Gracias



FACULTAD DE  
INGENIERÍA



UNIVERSIDAD  
DE LA REPÚBLICA  
URUGUAY