

## Taller de Introducción a la Programación Grafica Interactiva

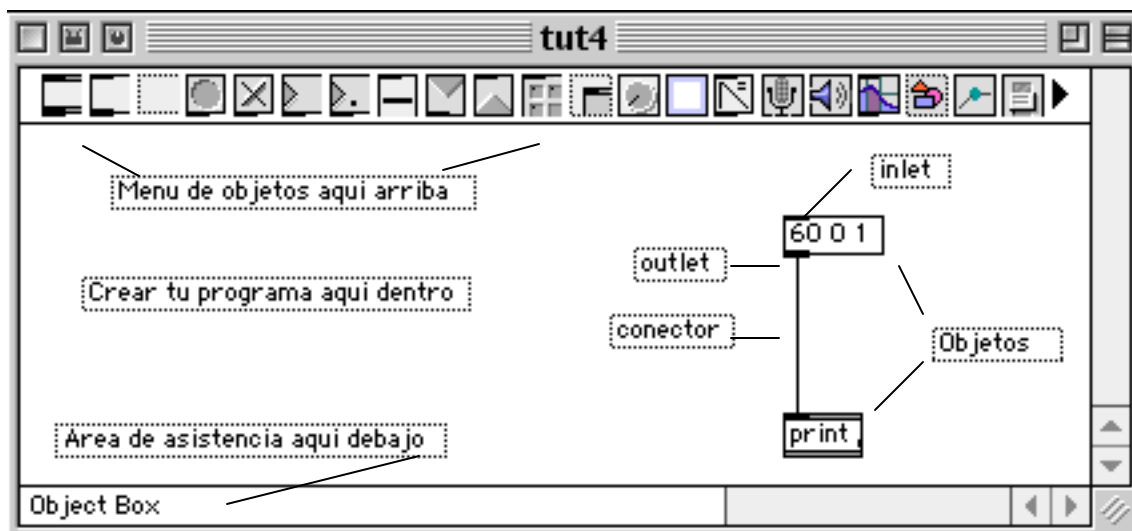
### Hardware y software requerido:

Macintosh			
			Software: MAX/MSP

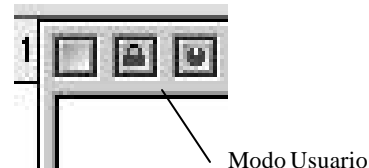
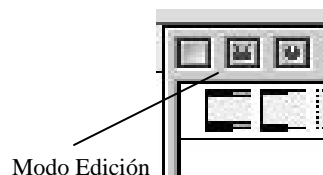
### Modulo 1:

Manipulación de objetos, asignación de datos y sintaxis del programa.

- Abrir programa MAX/MSP.
- Menú: File → New → *Patcher*
- Sin hacer clic en el mouse, arrastrar lentamente el mismo sobre los diferentes Objetos que están en la parte superior del *Patcher*. Observe al mismo tiempo el texto que aparece al pie del *Patcher*. Allí se describe que tipo de Objeto es en el área de asistencia.



- **VOCABULARIO:**
  - **Patcher** - sector en donde insertamos los diferentes Objetos programables y con los cuales creamos nuestro programa en Max.
  - **Objetos** - iconos gráficos que permiten conectarse unos con otros y pueden programarse.
  - **Inlets & Outlets** – conectores de entrada o salida entre un objeto y otro.
- **COMENTARIOS:** Clic sobre el tercer Objeto (*Comment*) y lo arrastramos dentro del *Patcher*. Escribimos allí “Programación en Max”. Este Objeto nos permite escribir comentarios.
- **MODO USUARIO VS EDICIÓN:** Clic sobre el segundo icono ubicado en el extremo superior izquierdo en forma de candado. Al cerrarse estamos en el Modo Usuario (observe como desaparece el menu de Objetos), al abrirse estamos en el Modo Edición. Este ultimo nos permite programar en Max, el primero nos permite hacer uso de dicha programación.



- **LISTA DE OBJETOS:** Clic sobre el primer Objeto (*Object Box*) . Lo ubicamos dentro del *patcher* y con un solo clic se desplegara un menu. Clic en *All Objects* y en la columna de la derecha se verán todas las instrucciones posibles con las que puede ser programado el objeto *Box*. En cambio, si comenzamos a partir del segundo ítem -de la izquierda- (Ej.: *control*) encontraremos que estas mismas instrucciones estarán agrupadas por categorías.

- **BORRADO DE OBJETOS:** Para borrar un Objeto, seleccione uno ya insertado. Menu → *Edit* → *Clear*
- **IMPRESION DE TEXTO:** Clic sobre *Object Box*. Ubicar en *Patcher*. Clic en *all objects* → *Print* (columna de la derecha), o bien escriba directamente la palabra *print* en minúsculas.
- Clic sobre el segundo objeto (*Message Box*). Ubicar en *Patcher*. Escriba tres palabras. Conectar su *outlet* con el *inlet* del Objeto *Print*.
- Ir a Modo Usuario. Clic sobre el segundo icono ubicado en el extremo superior izquierdo en forma de candado.
- Clic sobre el texto escrito. Observe como se imprime el texto en la interfase de Max ubicada detrás del *Patcher* (Max - Windows).
- **AUTOMATIZACION:** Clic sobre *Object Box*. Ubicar en *Patcher*. Clic en *All objects* → *metro* (columna de la derecha). Escriba a continuación de *metro* la cifra 1000.
- Clic sobre el quinto Objeto (*Toggle*). Ubicar en *Patcher*. Conectar su *outlet* con el *inlet* del Objeto *metro*, el *outlet* de *metro* conectarlo con el *inlet* del Objeto con el texto anteriormente ingresado.
- Ir a Modo Usuario. Clic sobre el primer Objeto (*toggle*). Cada vez que se active se imprimirá el texto cada 1000 milisegundos (1 seg.) en forma automática.
- También puede hacer clic sobre el texto y se imprimirá independientemente de la automatización programada.
- **OCULTAMIENTO DE OBJETOS:** Seleccione todos los objetos ingresados al *Patcher* **menos** el Objeto *toggle*. Para seleccionar todos puede también hacer clic en cada uno apretando al mismo tiempo la tecla shift (haga clic también sobre los conectores). Suelte la tecla shift y diríjase al menú → *Objetc* → *Hide on Lock*.
- Ir a Modo Usuario y observe como se ocultaron los objetos seleccionados. Clic en el único objeto visible. Veamos como el programa aun funciona (asómbrese!).
- **Programar:** Realice un programa **-DATA-1-** que imprima, en forma automatizada, diferentes palabras o frases que se alternen constantemente. Solo se deberán ver los objetos que disparan los eventos.
- **Atención:** Poner diferentes tiempos a los *metros*, poner frases o palabras que puedan combinarse, poner al final un solo objeto *print*, y al principio poner un solo objeto que active a todos los objetos *metro*.
- *Muestre el programa concluido al docente.*

## Modulo 2:

Inspector de objetos, formas de desplegar y controlar objetos

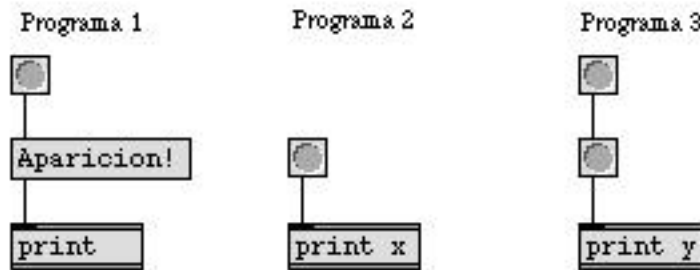
- **NUMBER BOX:** Clic sobre el sexto objeto (*Number Box*). Ubicar en *Patcher*. Ir a Modo Usuario y arrastre el mouse sobre el objeto *Number Box* y verá que los valores cambian indefinidamente.
- **INSPECTOR DE OBJETOS:** Retorne al Modo Edición y presione la tecla *ctrl.*, haga clic sobre el objeto y vaya a *Get Info*. En *Range* (rango de valores entre los que se moverán las cifras) y asigne valores *min.* 0 y *max.* 127. Ir a Modo Usuario y observe como los valores se mueven dentro de dicho rango (0 a 127).
- Retorne al Modo Edición. *Get Info* nuevamente del objeto y modifique en *Display Style* a hex. (Hexagecimal). Ir a Modo Usuario y observe como los valores se mueven en dicho rango pero con valores en hexagecimal.
- **NUMERACIONES ALTERNATIVAS:** Retorne al Modo Edición e inserte 3 Objetos *Number Box* iguales al primero. Asigne en el inspector de cada uno un diferente *Display Style* con valores en decimal, binario y en MIDI Note Names respectivamente. Conecte el *outlet* del primer objeto (que despliega valores hexagecimals) con el *inlet* de cada uno de los tres objetos previamente incorporados. Coloque un comentario (objeto *Comment*) que indica; decimal, midi notas, hexagecimal y binario respectivamente. Ir a Modo Usuario y modifique los valores en el objeto con despliegue numérico en hexagecimal.
- **Programar:** Realice un programa **-DATA-2-** (derivado del anterior) que: a) al modificar valores en decimal se modifican los tres restantes. B) al modificar valores en Binario se modifican los tres restantes. C) al modificar midi notas se modifican las tres restantes. Indique sus respectivos valores con comentarios adjuntos.
- *Muestre el programa concluido al docente.*

## Modulo 3:

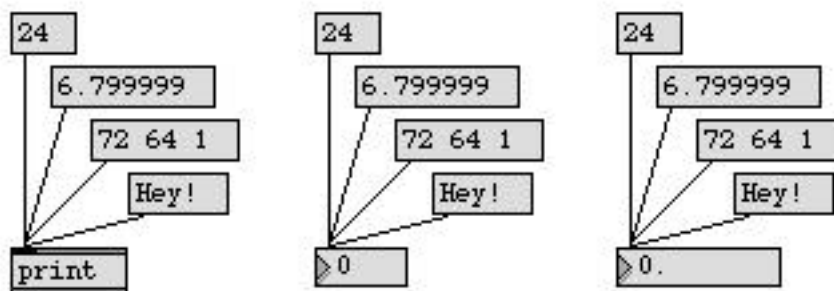
Análisis de diferentes ejemplos tutoriales.

- **TRIGGERS:** Abra el file → **tutorial 1**. Vera 3 programas que imprimen diferentes eventos por medio de variantes en los métodos de disparo activados con el objeto *button*.
- **Descripción de 1, 2 y 3:**
- **Programa 1:** El botón dispara la impresión del texto encontrado en mitad de su camino.
- **Programa 2:** El nombre del objeto *print* puede llevar allí un argumento para imprimir (ejemplo: *print x* o bien y o cualquier texto o numero). Observar que se imprime el texto y la palabra *bang*. *Bang* es el botón mismo.

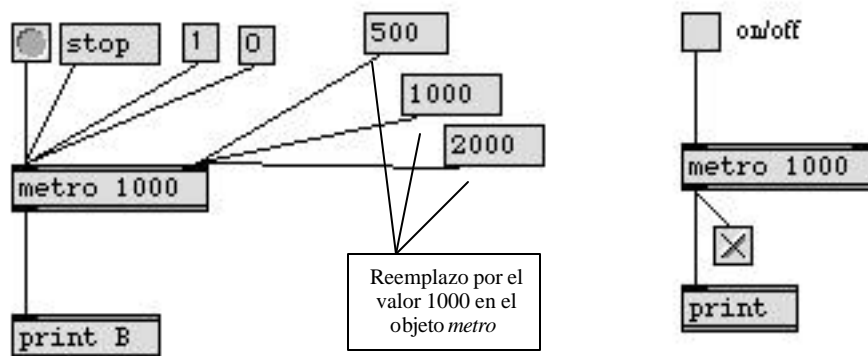
- Programa 3:** Igual al anterior salvo que ponemos un botón en mitad del camino a modo de testeo de disparo. Es decir que se iluminará cuando lo atraviese la acción de disparar algo. En definitiva, el objeto *button* es una especial forma de trigger que causa que un objeto ejecute su acción.



- NÚMEROS INT, FLOAT Y LIST:** Abra el file → **tutorial 2**. Max distingue números enteros (int = 1,4 , 77, etc), decimales con una parte fraccional (flota = 2.3308, 0.49, etc) y listas de números (list = 22 555 7826, etc). Es importante saber que en Max, cuando un numero decimal es convertido a entero, su fraccional parte no es tenida en cuenta. Ejemplo: 6.799999 será convertido a 6 y no 7 (ver ejemplo en el segundo programa). Adicionalmente observe que los *Number Box* y *Float Number Box* también distinguen entre enteros y fraccionales. Preste atención con lo que ocurre al ejecutarlos con el texto en los dos últimos programas.

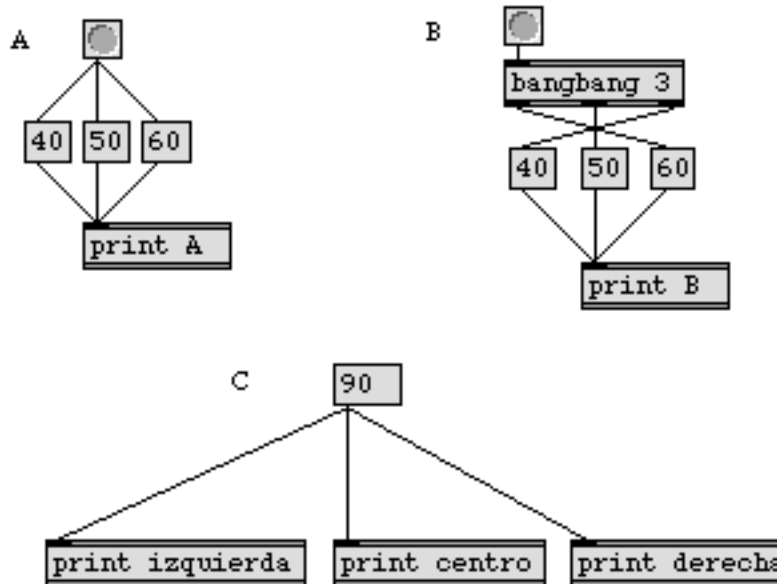


- METRO:** Abra el file → **tutorial 3**. Como ya hemos visto, el objeto *metro* permite automatizar un disparo a tiempos asignados en su argumento indicados en milisegundos. Ahora, los ejemplos dados presentan diferentes formas para activarlo y desactivarlo. Con un *stop* o un cero o un *toggle* se desactiva. Un 1, o un botón, o un *toggle* lo activará. Observe atentamente que en el primer ejemplo, un *inlet* en la parte superior-derecha permite variar el tiempo de ejecución del objeto *metro*.

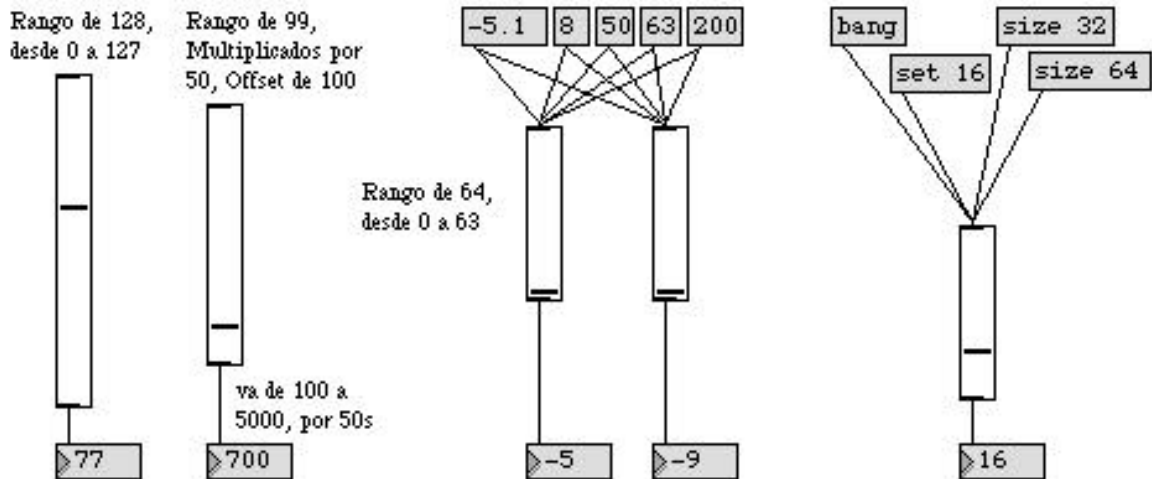


- Respuesta:** ¿Que función cumple el objeto *toggle* que esta sobre el objeto *print* del segundo ejemplo?  
 -----  
 -----
- Muestre el texto al docente.

- **ORDEN DE MENSAJES**: Abra el file → **tutorial 4**. En Max los mensajes se envían con una prioridad de derecha a izquierda. Los *outlets/inlets* también responden a dicha prioridad.



- **Descripción de A y B:**
  - **A:** El orden de impresión será 60, 50 y 40. A modo de comprobación, cambie de posición los diferentes objetos numéricos, por ejemplo, ubique el 60 al medio, el 50 a la derecha y observe luego como se imprimirán los valores.
  - **B:** El orden será 40, 50 y 60 ya que ahora se ejecuta de acuerdo al orden de sus *outlet* quienes están dispuestos también de derecha a izquierda.
  - **Responda:** ¿Describe como funciona **C**?
- 
- *Muestre el texto al docente.*
- 
- **SLIDER:** Abra el file → **tutorial 5**.



- **Descripción de 1 y 2:**
- **Slider 1:** El objeto *slider* determina valores entre un rango de 0 a 128. (ver *Get Info* del objeto).
- **Slider 2:** El rango es de 99 valores (0 a 98) pero estos son multiplicados por su *Multiplier*, que no es de 1 como en el primer caso, sino que es de 50 y el *offset* es de 100. El proceso es el siguiente: cuando el *slider* esta en el mas bajo valor  $(0 * 50) + 100$  va a enviar por el *outlet* un 100. Cuando esta en el mas alto valor  $(98 * 50) + 100$  va a enviar un valor de 5000.
- **Slider 3 y 4:** En estos objetos, los comportamientos son disímiles.
- **Respuesta:** ¿Porque muestran diferentes valores los dos *sliders* afectados por los mismos objetos teniendo ambos el mismo rango de valores? (detalle que ocurre en cada caso con los valores -5.1, 8, 50, 63 y 200).

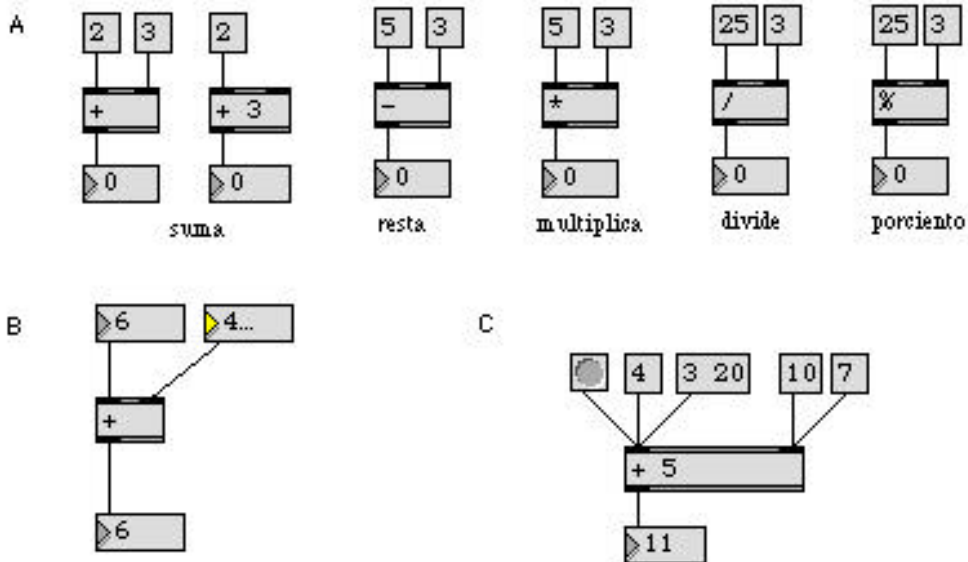
-----  
 -----

- *Muestre el texto al docente.*
- **Respuesta:** ¿Que ocurre con cada uno de los parámetros del objeto Slider 5?
- **set :** -----
- **size :** -----
- **bang :** -----
- *Muestre el texto al docente.*

- **CALCULOS ARITMETICOS:** Abra el file → **tutorial 6**. Max permite realizar diferentes operaciones matemáticas. Algunos ejemplos de operaciones básicas y modos de empleo de los objetos utilizados se dan a continuación.

\* los numeros del inlet izquierdo disparan el calculo

\* los numeros del inlet derecho son almacenados y se utilizan como operadores



- **Descripción de A y B:**
- **A:** Sumas, restas, multiplicaciones, divisiones y porcentajes. Observar el primer ejemplo con dos modalidades de suma: el *inlet* de la izquierda (operador = 2) dispara el calculo sumando así el valor del *inlet* de la derecha (operando = 3). También podemos ubicar directamente el operando dentro del objeto suma (ver segundo ejemplo de sumas).
- **B:** Otro modo de realizar operaciones matemáticas es la de modificar en forma deslizable valores que se van a ir calculando continuamente (cambie el valor del operador y el operando).
- **Respuesta:** ¿Como funciona el ejemplo C con cada uno de sus operadores y operandos (es decir, cual es el comportamiento de los objetos con los valores 4, 3 y 20, 10 y 7 respectivamente)?

- 
- *Muestre el texto al docente.*
  - **ATAJOS:** 7 inmediatas formas de editar en Max (compruebe cada una de ellas).
  - **Modo Usuario / Edición:** Mientras presiona la tecla **Command** y hace un clic sobre el *patcher* (sobre ningún objeto en particular) se habilitara el modo Usuario o Edición.
  - **Ultimo / nuevo objeto:** Mientras presiona la tecla **Ctrl** y hace un clic sobre el *patcher* (sobre ningún objeto en particular) bajara un menú de propiedades. *Active Lock*, o *Select All*, u *Object Palette* o bien *Recent Object*.
  - **Duplicación:** Mientras presiona la tecla **Shift** y hace clic sobre cualquier objeto ubicado en la barra del menú (salvo los tres primeros) y luego lo inserta en el *patcher* haciendo clic varias veces, este objeto se duplicara en forma reiterada.
  - **Selección de Objetos y Conectores:** Mientras presiona la tecla **Option** y selecciona varios objetos con sus conectores, estos últimos también se seleccionaran.
  - **Propiedades de Objetos:** Mientras presiona la tecla **Ctrl** y selecciona un objeto ya insertado en el *patcher* vera las propiedades de este objeto tal como si fuera al menú principal de *Objetc* .
  - **Help de un Objeto:** Ingrese un Objeto *Box* con argumento *Print*. Luego, mientras presiona la tecla **Option** y selecciona el objeto *print* vera un ejemplo de su funcionamiento en un nuevo *patcher*.
  - **Segmentar conectores:** Inserte al *patcher* 4 objetos que puedan conectarse entre si. Antes de conectarlos presione la tecla **Shift** y haga clic sobre el *outlet* de un objeto, ahora conéctelo a otro objeto pero trazando líneas rectas de 90 grados para llegar al *inlet* deseado (quiebre las líneas haciendo clic al trazar la línea misma).

**Hardware y software requerido:**

Macintosh	Modulo MIDI	Controlador MIDI	Amplificador de auriculares
Cable Mini DIN	Cable DIN	Cable Mini Plug Stereo – 2	<b>Software:</b> MAX/MSP

		Plug Mono	
--	--	-----------	--

**Modulo 4:**

Objetos básicos de transmisión de datos MIDI

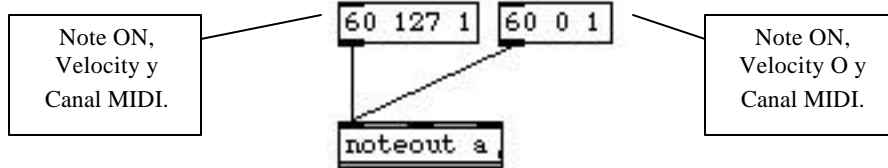
- Hay varios objetos en Max que transmiten y reciben data MIDI, hacia y desde un equipo MIDI.
- Objetos bastante comunes son los *midin* y *midout*. Estos reciben data MIDI, byte por byte, sin analizar los mensajes (solo los envían o reciben tal cual fueron enviados o recepcionados).



- **NOTEOUT-IN:** Mas especializados objetos MIDI, que filtran la data que ingresa por su *inlet* y sale por su *outlet* son los objetos *notein* y *noteout*.

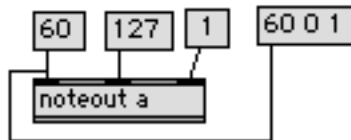


- **Programar:** Realice el siguiente programa **MIDI-1** graficado a continuación.



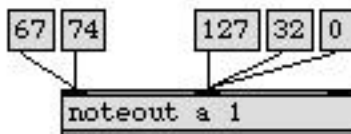
- Ejecute el objeto con la lista de valores 60 127 1. Ejecute el otro objeto con la lista de valores 60 0 1.
- **Descripción:** Lo enviado en el primer objeto fue: 60 = pitch, 127 = velocity, 1 = canal midi. El otro objeto envía un Note on con velocity 0 (al mismo canal). La letra “a” indica que la data midi ingresa por el puerto del modem, o bien el puerto “b” perteneciente a la impresora.
- **Programar:** Cambie de valores al pitch y al velocity.
- *Muestre el programa al docente.*

- **Programar:** Realice el siguiente programa **MIDI-2** graficado a continuación.



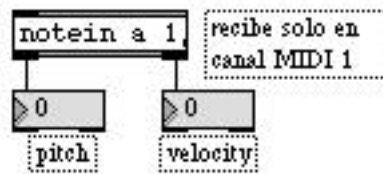
- **Descripción:** Como se ve, uno puede enviar individualmente los mensajes midi por su correspondiente *inlet*. De ser así deberán ser disparados secuencialmente, siendo el primero el numero de canal y el ultimo el pitch. El note off (en este caso note on con velocity 0) es enviado de una sola vez en una lista como en ejemplo **MIDI-1**.
- **Programar:** Al programa **MIDI-2** conectarle un botón para que dispare todos los eventos del note on al mismo tiempo.
- *Muestre el programa al docente.*

- **Programar:** Realice el siguiente programa **MIDI-3** graficado a continuación.



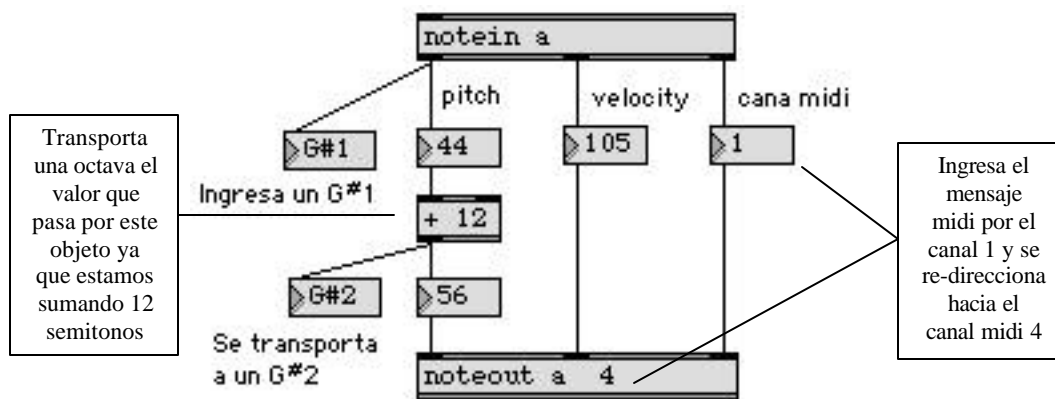
- **Respuesta:** ¿Como funciona el programa realizado al ejecutar los diferentes valores?  
 -----  
 -----
- *Muestre el programa y el texto al docente.*

- **NOTEIN:** Este objeto recibe data midi por su inlet desde un controlador midi conectado al sistema.
- **Programar:** Realice el siguiente programa **MIDI-4** graficado a continuación.



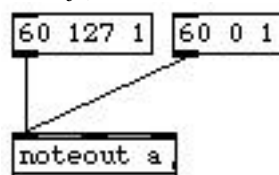
- Ejecute un controlador midi enviando alguna nota. Se deberá ver el valor correspondiente de la nota y su correspondiente velocity.
- **Programar:** Elimine el valor 1 que esta a continuación de la *a* (puerto del modem).
- **Responda:** ¿Que ocurrió y que apareció en el objeto de nuevo? (observe los *outlets*!)

- *Muestre el programa y texto al docente.*
- **Programar:** Realice el siguiente programa **MIDI-5** graficado a continuación.

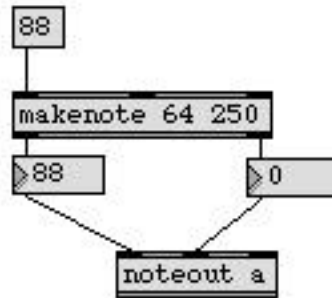


- **Atención:** El objeto *notein* no tiene especificado por que canal deja ingresar la mensajes midi (ejemplo contrario del programa **MIDI-4** ). Por consiguiente, al no indicarse puede recibir mensajes midi por cualquier canal (modo OMNI). Pero a pesar de recibir data midi en cualquier canal (ver objeto *notein*) obliga a estos mensajes a re-direccionarse y salir únicamente por el canal midi numero 4 (ver objeto *noteout*).
- Ejecute un controlador midi enviando alguna nota.
- **Responda:** ¿Como funciona el programa realizado?

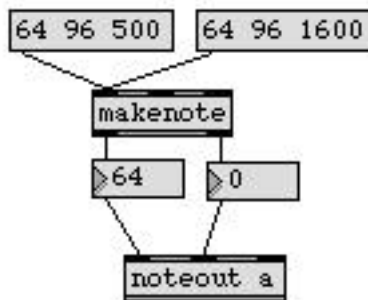
- *Muestre el programa y texto al docente.*
- **NOTE-ON Y OFF:** Uno de los problemas al enviar un mensaje de nota es que requiere que se envíe posteriormente su correspondiente note off, a la misma nota y por el mismo canal, tal como fue el caso presentado anteriormente (ver grafica debajo).



- **MAKENOTE:** Max tiene un objeto que envía automáticamente el correspondiente note off de la nota que fue previamente enviada. Este objeto se llama *makenote*.
- **Programar:** Realice el siguiente programa **MIDI-6** graficado a continuación.



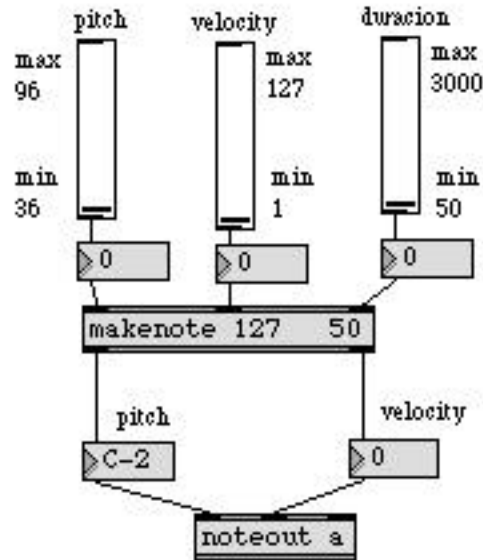
- Clic en el objeto numérico superior con el numero 88.
- **Descripción:**
  1. *makenote* recibe un numero en su *inlet* izquierdo y lo interpreta como el pitch,
  2. lo combina con el velocity (Ej.: 64),
  3. y envía a cada valor por los 2 *outlets* (el 88 = pitch y 64 = velocity).
  4. Después de un específico tiempo de delay (Ej.: 250msg) automáticamente envía el mismo pitch pero con velocity 0.
- **Programar:** Realice el siguiente programa **MIDI-7** graficado a continuación.



- **Responda:** ¿Cuáles son las diferencias de funcionamiento de los dos últimos programas y asegúrese de entender porque el objeto *makenote* en este caso no incorpora parámetros numéricos y porque en el programa anterior **MIDI-6** si ?

.....  
 .....

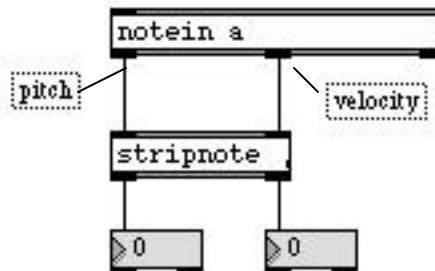
- *Muestre el programa y texto al docente.*
- **Programar:** Realice el siguiente programa **MIDI-8** graficado a continuación.
- **Atención:** Observe los rangos *max.* y *min.* de cada *slider*.



- **Responda:** ¿Que valor tendrá el parámetro de *range* en los diferentes *slider* en base a los datos dados? a: el *slider* del *pitch*: *offset* = 36 y *multiplier* = 1. El *slider* del *velocity*: *offset* 1, *multiplier* = 2. El *slider* de *duración*: *offset* = 50, *multiplier* = 50.

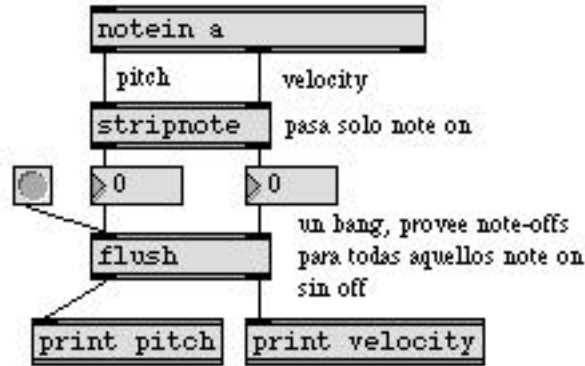
- *Muestre el programa y texto al docente.*

- **STRIPNOTE:** Otro objeto que filtra data MIDI es el *stripnote*. Este objeto es similar al objeto *makenote* con la siguiente diferencia. *Stripnote* recibe mensajes de *pitch* y *velocity* en sus correspondientes *inlets*, pero deja pasar estos datos a sus *outlet* si el **velocity NO es de un valor igual a 0**. Esto quiere decir que este objeto filtra los *note off* y deja pasar solo los mensajes *note on*. Este tipo de objeto es útil cuando, por ejemplo, queremos capturar los datos del teclado cuando este es presionado, pero no deseamos obtener sus valores MIDI cuando las teclas son levantadas generando el consiguiente *note off*.
- **Programar:** Realice el siguiente programa **MIDI-9** graficado a continuación.
- **Atención:** Observe como llegan solo los valores de *note on* a los diferentes objetos numéricos.

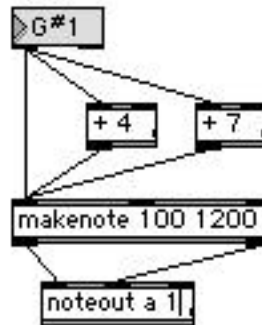


- *Muestre el programa al docente.*

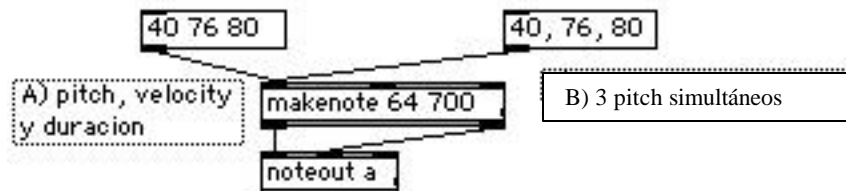
- **FLUSH:** Este objeto envía mensajes *note off*, pero a diferencia de *makenote*, que enviaba *note off* automáticamente luego de un tiempo x, el objeto *flush* envía *note off* a todas los mensajes *note on* que han pasado a través de el y que aun no han recibido su correspondiente *note off*. *Flush* se dispara con un objeto *bang* (botón) conectado en su *inlet* izquierdo.
- **Programar:** Agregue al programa **MIDI-9**, realizado anteriormente, los objetos necesarios para completar el siguiente programa graficado a continuación.



- *Muestre el programa al docente.*
- **ACORDES**: Supongamos que deseamos crear un programa que ejecute acorde de triadas mayores a partir del envío de una sola nota.
- **Programar**: Realice el siguiente programa **MIDI-10** graficado a continuación.
- Los objetos intervinientes son: 1 *number box*, y 4 *object box*.



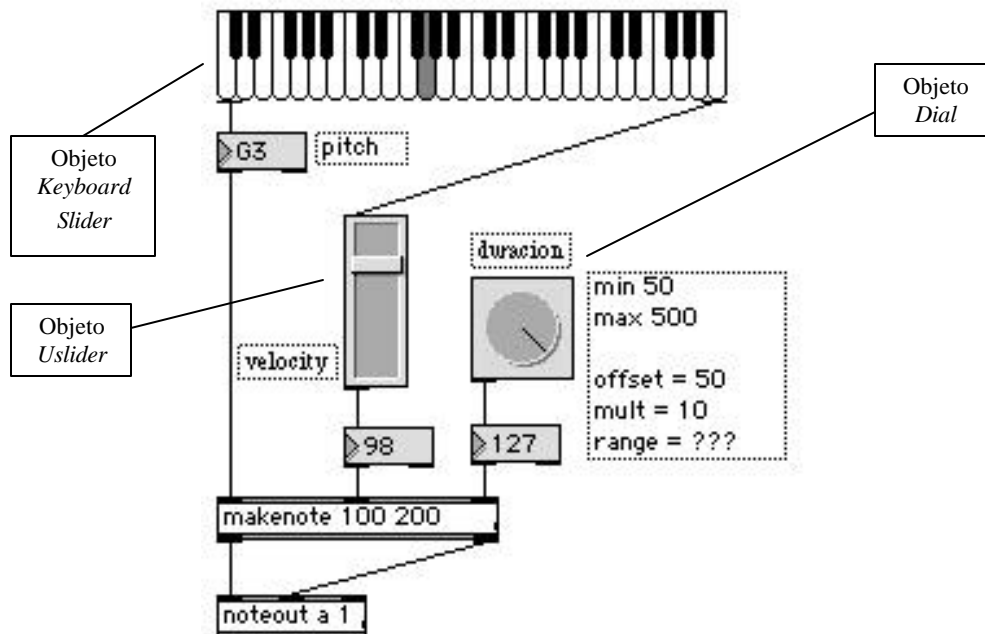
- **Descripción**: El + 4 agrega 4 semitonos al valor de cada nota ingresada. Idem con + 7. Recuerde los parámetros de *makenote*. (100 = velocity, 1200 = tiempo en msg en enviar el off)
- *Muestre el programa al docente.*
- **Programar**: En el mismo archivo realice las siguientes variaciones adicionales: a) modifique el valor de semitonos (intervalos) y b) agregue dos notas mas al acorde.
- *Muestre el programa al docente.*
- **NOTAS SIMULTANEAS**: Como ya hemos visto, con el objeto *message*, hemos enviado un número o listas de números al objeto *makenote* (ver MIDI-1 o MIDI-7) interpretando estos valores como pitch, velocity y duración. Ahora, si enviamos una lista de números, pero en este caso, separados por comas, la interpretación de un objeto como *makenote* será diferente.
- **Programar**: Realice el siguiente programa **MIDI-11** graficado a continuación.



- **Respuesta**: Si primero disparamos la lista numérica A y luego la lista B ¿qué velocity y duración de nota tendrán los pitch con valor decimal 40, 76 y 80?

- *Muestre el programa y texto al docente.*

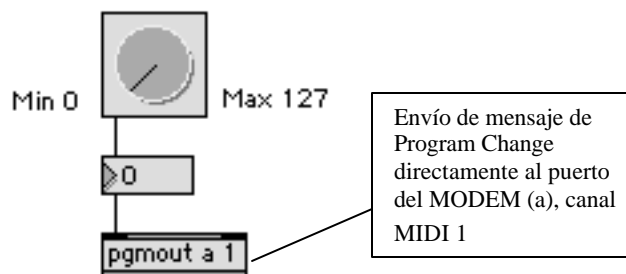
- **KEYBOARD SLIDER, USLIDER, DIAL:** Podemos utilizar diferentes tipos de *slider* para enviar o recibir datos midi. A continuación 3 nuevos objetos que se diferencian fundamentalmente por sus parámetros.
- **Programar:** Realice el siguiente programa **MIDI-12** graficado a continuación.
- **Atención:** Los objetos intervinientes son: *keyboardslider*, 1 *uslider*, 1 *dial*, 3 *number box*, 2 *objetc box*.



- **Responda:** ¿Cual es el valor de *range* del objeto *dial*?

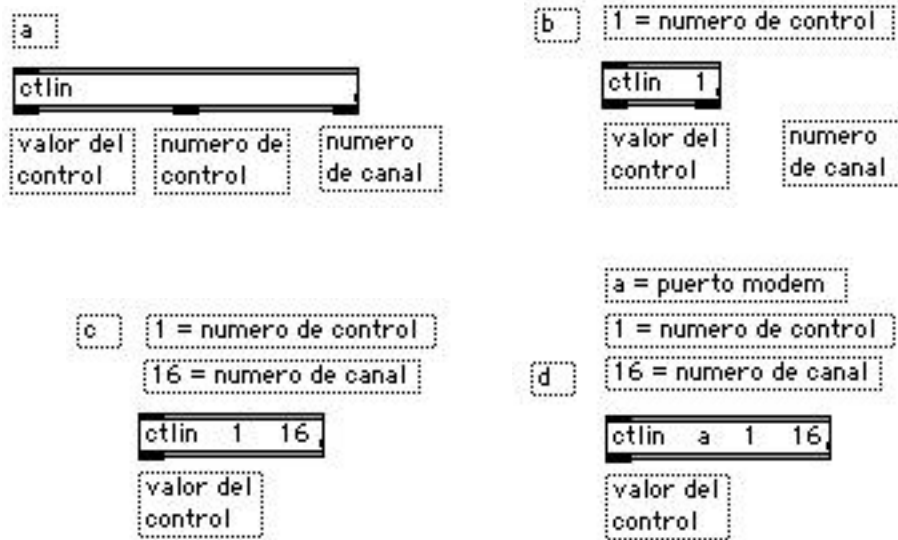
---

- *Muestre el programa y texto al docente.*
- **PGMOUT:** El objeto *pgmout* envía mensajes de cambios de programa (Program Change) al sintetizador que se encuentre conectado. Suponiendo que tenemos un banco de 128 programas por canal, como es el caso en los equipos GM, vamos a manejar este rango para el objeto *dial* de 0 a 127. Esto se modificará de acuerdo al equipo que quisiéramos enviar su correspondiente numero de programa.
- **Programar:** Realice el siguiente programa **MIDI-13** graficado a continuación.

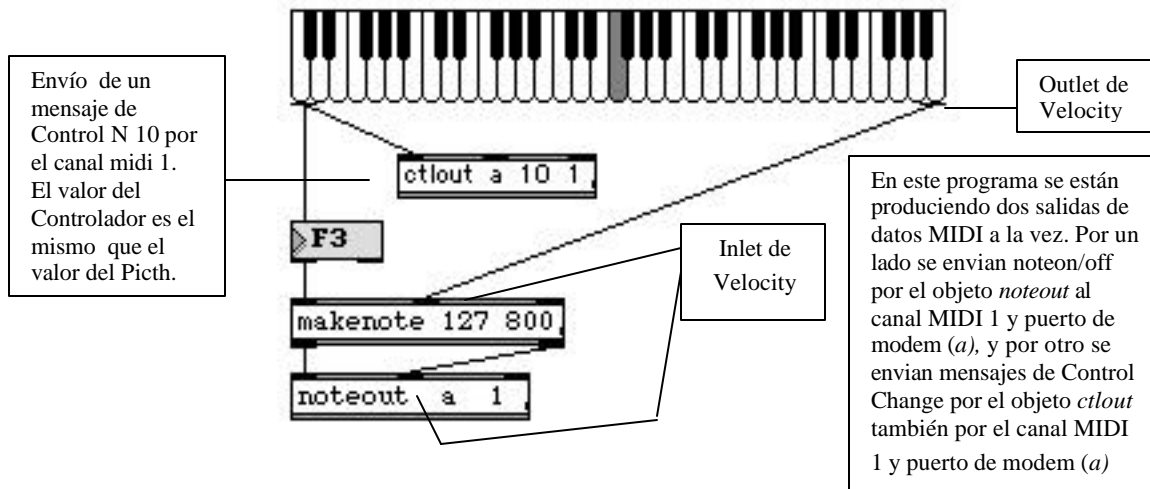


- *Muestre el programa al docente.*
- **Programar:** ¿Que objetos le debo agregar al programa **MIDI-13**, anteriormente realizado, para que con cada giro del objeto *dial* continúe enviando cambios de programa y a la vez, estos mismos valores de programa, sean interpretados como mensajes note on? (resolver el envío automático de sus correspondientes note off).
- *Muestre el programa al docente.*

- **CTLOUT – CTLIN:** El objeto *ctlout* y *ctlin* envía y recibe mensajes de Control Change al sintetizador que se encuentre conectado.
- La sintaxis que utiliza responde a 4 alternativas diferentes de programación (tanto para *ctlout* y *ctlin*):



- **Programar:** Realice el siguiente programa **MIDI-14** graficado a continuación.



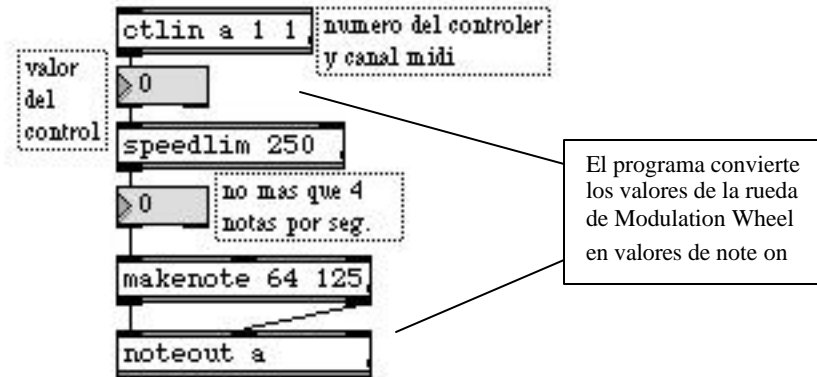
- **Responda:** ¿Que mensaje de Control Change se esta enviando a la vez que se ejecuta el *keyboard slider*?

-----

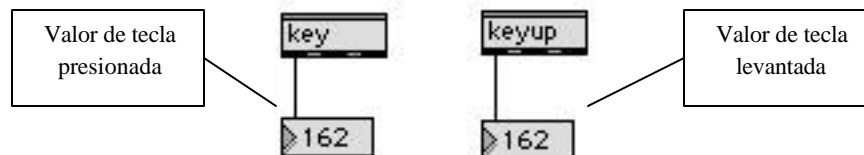
- *Muestre el programa y texto al docente.*

- **Programar:** Realizar el programa -**MIDI-15**- que envíe un mensaje midi de *MODO* como el *All note off* (numero 123) .
- *Muestre el programa al docente.*

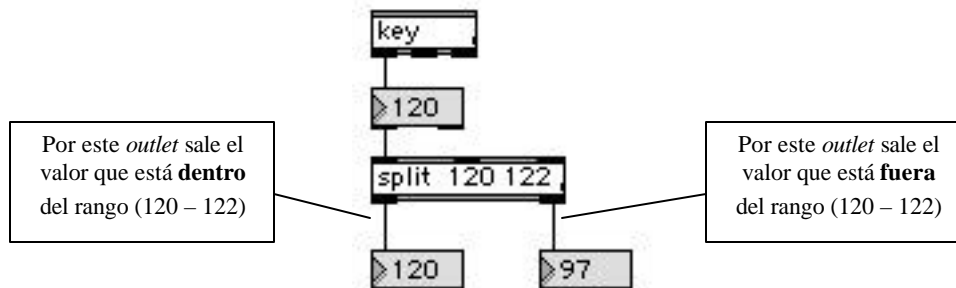
- **REASIGNAR DATOS:** Max puede reasignar los valores recibidos por un mensaje de Control Change continuo y enviarlo para que ejecute valores de notas, como en forma similar hicimos con los cambios de programa (ver MIDI-13).
- **Programar:** Realice el siguiente programa **MIDI-16** graficado a continuación.
- Aquí encontramos un nuevo objeto llamado *speedlim*. Cuando este objeto recibe un numero, este envía este mismo numero a su *outlet* pero espera cierto tiempo calculado en *msg* antes de recibir otro mensaje.
- **Atención:** Ejecute el programa enviando mensajes midi con la rueda de *modulation wheel*.



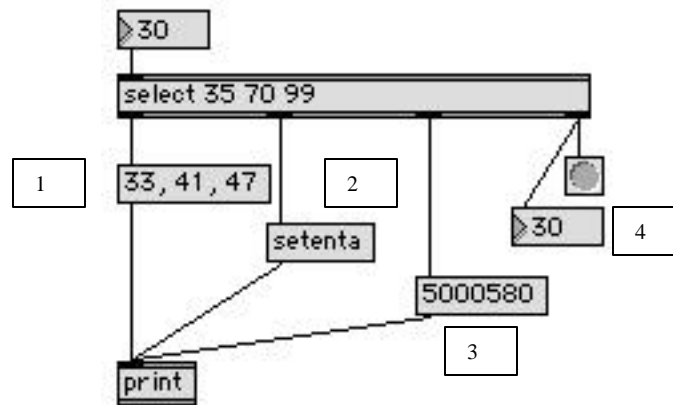
- *Muestre el programa al docente.*
- **KEY – KEYUP (ASCII):** Cuando tipiamos en el teclado de la Macintosh podemos interpretar estos mensajes como datos para nuestros programas. El *American Stándar Code for Information* (ASCII) es el standart para asignar un numero a cada tecla del teclado. Max tiene un objeto que puede interpretar estos números.
- El objeto que interpreta los valores del teclado es el objeto *key*. A su vez el objeto *keyup* interpreta los mismos valores pero cuando cada tecla es dejada de presionar.
- **Programar:** Realice el siguiente programa **MIDI-17** graficado a continuación.



- *Muestre el programa al docente.*
- **SPLIT:** Si uno quisiera filtrar un rango de valores, en este caso de ciertas teclas de la Macintosh, podríamos utilizar el objeto *split*. Este objeto deja pasar por su *outlet* izquierdo un rango de valores asignados (en este caso las teclas 120 al 122, es decir  $x - y - z$ ). Por otro lado, envía por su *outlet* derecho los restantes valores que no estén dentro de dicho rango.
- **Programar:** Agregar al anterior programa el objeto *split* según se detalla.

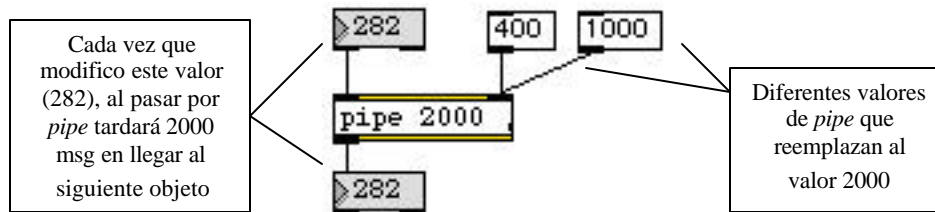


- *Muestre el programa al docente.*
- **Programar:** Agregue los objetos necesarios para que los valores tipeados en el teclado de la Macintosh sean interpretados como mensajes MIDI note on/off
- **Atención:** Los mensajes de note on deberán ser interpretados con el objeto *key*, y los note off con *keyup*. No utilizar el objeto *makenote* para enviar automáticamente los note off.
- *Muestre el programa al docente.*
- **SELECT:** El objeto *select* es otro filtro de datos numérico, mas específico que el objeto *split*, el cual nos permite seleccionar un valor o varios en forma individualizada.
- **Programar:** Realice el siguiente programa **MIDI-18** graficado a continuación.

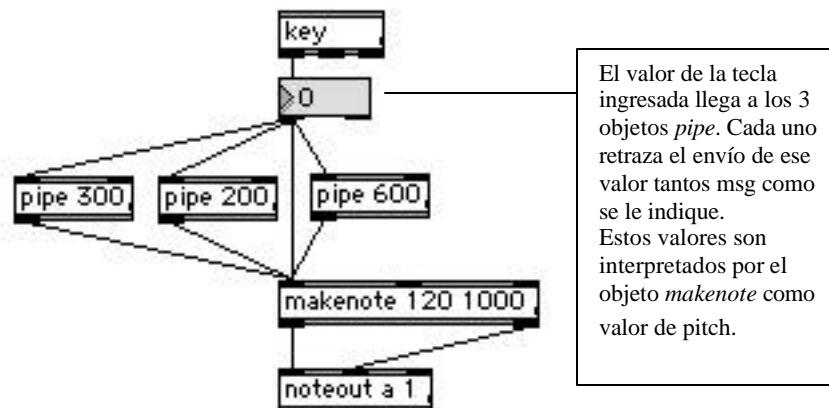


- **Descripción:** En el ejemplo, el objeto *select* selecciona 3 diferentes valores (35, 70 y 99).
  1. En el caso de recibir por su *inlet* un valor de 35 disparará un *trigger* por su 1er *outlet* (en este caso se imprimirá la lista de valores 33, 41 y 47).
  2. Si el valor es de 70 disparará un *trigger* por su 2do *outlet* (en este caso se imprimirá el texto “setenta”).
  3. Si el valor es de 99 disparará un *trigger* por su 3er *outlet* (en este caso se imprimirá la cifra 5000580).
  4. Y si el valor no es ninguno de los tres, enviará estas cifras por su ultimo *outlet* que no se imprimirán.
- **Atención:** Observe que en el caso que los valores coincidan con las cifras indicadas en el objeto *select*, el resultado implicará que se dispare un *trigger* por los diferentes *outlets* y no que se retransmita el valor indicado por *select*, salvo en el ultimo *outlet*.
- *Muestre el programa al docente.*
- **Programar:** Agregar y modificar diferentes objetos del programa **MIDI-18**, anteriormente realizado, para que:
  - un objeto *dial* envíe mensajes note on al girar el mismo,
  - y en el caso de recibir los valores 40 y 80 respectivamente se envíen dos diferentes mensajes de Program Change.
- **Atención:** Resolver el envío automático de sus correspondientes note off.
- *Muestre el programa al docente.*
- **PIPE:** En Max hay varios objetos que retrasan el envío de datos. Ya vimos el caso de *speedlimit* que espaciaba los envíos de datos de acuerdo a un tiempo dado. Ahora tenemos el objeto *pipe* que retrasa también un mensaje que llega a su *inlet* pero en este caso solo retrasa el tiempo del envío, transmitiendo los datos llegados a su *inlet* a la misma velocidad tal cual fueron recibidos luego de un tiempo *x* indicado en *msg*.

- **Programar:** Realice el siguiente programa **MIDI-19** graficado a continuación.



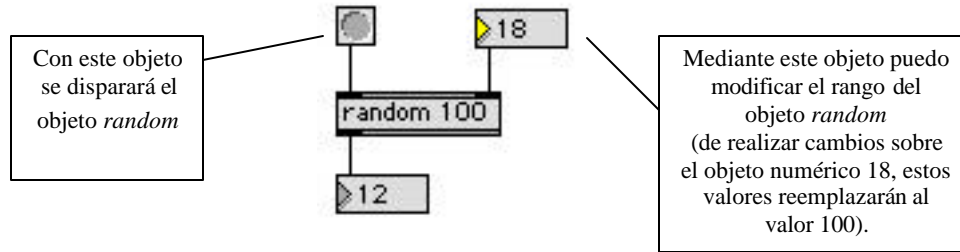
- **Atención:** Asegúrese de tener en claro la diferencia entre los objetos *speedlimit* y *pipe*.
- *Muestre el programa al docente.*
- **Programar:** Realice el siguiente programa **MIDI-20** graficado a continuación en donde un mismo dato numérico, enviado por el teclado de la Macintosh, es repetido en 3 direcciones con retrazos asignados por cada *pipe* para finalmente ser interpretados como pitch por un objeto *makenote*.



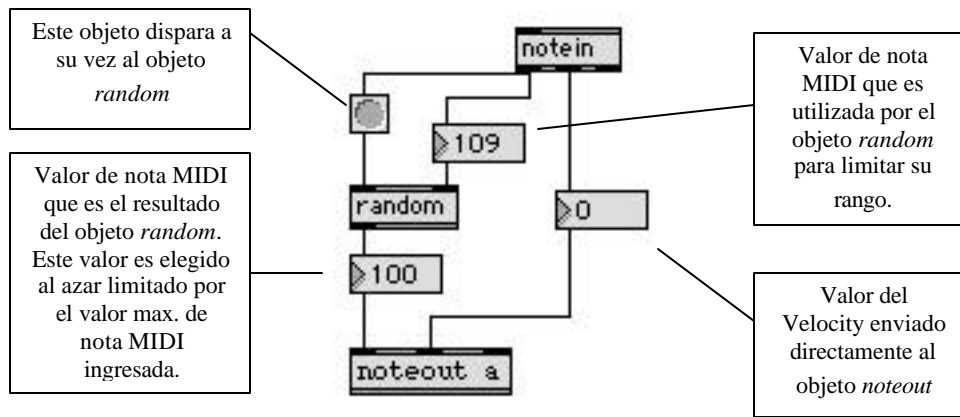
- **Responda:** Si modifica los valores de tiempo de los 3 objetos *pipe* ¿Qué escucharemos de diferente?

---

- *Muestre el programa y texto al docente.*
- **Programar:** Agregue diferentes objetos al programa **MIDI-20**, anteriormente programado, para que realice las siguientes funciones:
  1. que cada nota que llega a *pipe* sea previamente transportada formando arpeggios,
  2. aplicar a cada tiempo de *delay* de cada *pipe* un control variable de tiempo,
  3. y finalmente, que el ingreso de datos no sea a través del teclado de la Macintosh (mediante el objeto *key*) sino que se utilice el teclado del controlador de un sintetizador.
- *Muestre el programa al docente.*
- **RANDOM:** En el caso de necesitar enviar números aleatoriamente, Max dispone de un objeto llamado *random* que al recibir un *bang* en su *inlet* elige un numero al azar y envía este por su *outlet*.
- **Programar:** Realice el siguiente programa **MIDI-21** graficado a continuación.



- **Descripción:** En el ejemplo dado, cada vez que se ejecute el objeto *random* se enviará por su *outlet* un valor que estará dentro del rango de 0 a 100.
- *Muestre el programa al docente.*
- **Programar:** Realice el siguiente programa **MIDI-22** graficado a continuación utilizando el objeto *random*.



- **Respuesta:** ¿Cuál es el inconveniente que presenta este programa relativo a los mensajes midi de note off? ¿Cómo lo resolvería?  
-----  
-----
- *Muestre el programa y texto al docente.*
- **Programar:** Realice el programa **MIDI-23** que deberá realizar las siguientes acciones:
  1. Que en forma automática envíe regularmente valores de notas y velocity de manera aleatoria (el rango deberá ser, para el mínimo de 20 y para el máximo de 127).
  2. El programa se deberá activar con solo tocar la tecla “a” y se detendrá con la tecla “s” del teclado de la Macintosh.
  3. En el caso que reciba un valor decimal de nota entre 40 a 50 ejecutará dicho valor en forma de acorde (Fundamental / 3M / 5J),
  4. y cuando reciba un valor decimal de nota entre 60 a 70 ejecutará dicho valor en forma de arpeggio (Fundamental / 4J / 5J).
- **Atención:** Los objetos intervinientes en este programa involucran a la mayoría de los vistos en la guía de trabajos prácticos, como por ejemplo: *metro*, *toggle*, *button*, *random*, *makenote*, *pipe*, *split*, *select*, *key*, etc.
- *Muestre el programa al docente.*

- **RESUMEN DE OBJETOS:** Breve referencia de algunos objetos vistos hasta el momento.
- **print:** imprime algún mensaje en el *Max Windows*.
- **comment:** permite escribir comentarios a modo de texto.
- **button:** dispara el mensaje que se encuentra en otro objeto, o bien imprime un *bang*.
- **bangbang:** es como el *button* pero con tantos *outlets* como se le indique numéricamente.
- **number box:** despliega un valor (en decimal, octal, binario, hexadecimal, MIDI, etc).
- **metro:** envía un mensaje a intervalos regulares en milisegundos.
- **toggle:** cambia de estado entre un on (1) u off (0).
- **slider:** envía un numero por su *outlet* mediante el deslizamiento del mismo.
- **midiiin:** permite el ingreso y retransmisión por sus *outlets* de data MIDI.
- **midiiout:** envía data MIDI por el puerto asignado (modem, printer, usb, etc)
- **notein:** permite el ingreso y retransmisión por sus *outlets* de data MIDI, solo de mensajes de nota.
- **noteout:** envía data MIDI, solo de mensajes de nota, por el puerto asignado (modem, printer, usb, etc).
- **makenote:** genera un note off de los noteon ingresados, a un tiempo establecido en milisegundos.
- **striptime:** filtra los note off, y deja pasar y retransmite solo los note on.
- **flush:** genera note off de aquellas notas que no han recibido este mensaje. Se dispara con un *bang*.
- **keyboard slider, uslider, dial:** diferentes tipos de *slider* o interfase de controles numéricos.
- **pgmout:** envía los mensajes MIDI de Program Change
- **ctlin – ctout:** toma o envía los mensajes MIDI de Control Change.
- **key – keyup:** interpretan como valores numéricos cada tecla de la computadora.
- **speedlimit:** envía datos que llegan a su *inlet* pero espera a tomar otro dato luego de cierto tiempo.
- **split:** deja pasar por su *outlet* izquierdo lo que esta dentro de un rango, y por el derecho el resto de valores.
- **select:** es un filtro numérico que nos permite seleccionar un valor o varios en forma especifica.
- **pipe:** retrasa el envío de datos y los envía, a la velocidad en la que fueron enviados.
- **random:** elige un numero al azar el cual es enviado por su *outlet*.