

# Música por computadora

Ernesto Romero y Hernani Villaseñor  
Centro Multimedia 2012

## Sesión 10

### Conceptos musicales

Hasta ahora hemos revisado conceptos de la teoría del sonido en general. En esta sesión estudiaremos los principales conceptos de la teoría de la música. La sesión no pretende constituir un tratado de música. El propósito de esta sesión es que se obtenga un lenguaje a través del cual podamos expresar lo que estamos haciendo y ligarlo a los conceptos de teoría de la música por computadora que veremos más adelante.

### Altura

La altura se refiere a la frecuencia de un sonido. En lenguaje musical nos referimos a las notas. Las notas corresponden a frecuencias específicas. Existen 12 notas en la música tradicional occidental:

Nombre de la nota	Sistema inglés	Frecuencia	MIDI
Do	C	261.625	60
Do sostenido	C#	277.182	61
Re	D	293.664	62
Mi bemol	Eb	311.126	63
Mi	E	329.627	64
Fa	F	349.228	65
Fa sostenido	F#	369.994	66
Sol	G	391.995	67
La bemol	Ab	415.304	68
La	A	440	69
Si bemol	Bb	466.163	70
Si	B	523.251	71

Entre cada nota consecutiva se dice que hay  $\frac{1}{2}$  tono. Cada una de estas notas se pueden tocar más graves (multiplicándolas por  $\frac{1}{2}$ ) o más agudas (multiplicándolas por 2).

Dentro de este conjunto de notas se pueden hacer diferentes escalas. Las escalas son un conjunto de notas específicas. Este conjunto de notas definen una tonalidad. A partir de ahora usaremos el sistema inglés para referirnos a las notas.

La escala mayor de C es:

C D E F G A B C

Si analizamos la distancia en  $\frac{1}{2}$  tonos que hay entre cada nota obtenemos la siguiente estructura

1 1  $\frac{1}{2}$  1 1 1  $\frac{1}{2}$

Esto es: Entre C y D hay 1 tono, entre D y E un tono, entre E y F  $\frac{1}{2}$  tono, etc.

Esta escala tiene 7 notas. Si se agrega una nota más volvemos a tener C pero se dice que esta una octava arriba.

La tabla 1 tiene las notas dentro de la octava central del piano y se le conoce como la octava índice 5. Las mismas notas pero una octava arriba estarán en el índice 6. Los índices de las notas cambian a partir de cada C.

La escala menor armónica de C es:

C D Mb F G Ab Bb C

sus intervalos son

1  $\frac{1}{2}$  1 1  $\frac{1}{2}$  1 1

Si usamos la raíz doceava de dos podemos obtener un medio tono a partir de cualquier frecuencia. La raíz doceava de 2 en SuperCollider se escribe así:

$2^{**(1/12)}$

C es 261.625. Si queremos obtener  $\frac{1}{2}$  tono arriba de este C, es decir C# lo hacemos así:

$261.625 * 2^{**(1/12)}$

Y para obtener D lo multiplicamos otra vez:

$261.625 * 2^{**(1/12)} * 2^{**(1/12)}$

Es más práctico elevar al cuadrado  $2^{(1/12)}$  que multiplicarlo dos veces:

$$261.625 * ((2^{(1/12)})^{**2})$$

La potencia a la que estamos elevando  $2^{(1/12)}$  indica el número de medios tonos que obtendremos arriba de una nota.

Ejemplo:

261.625 \* ((2<sup>(1/12)</sup>)<sup>\*\*0</sup>) // cero medios tonos sobre C = C  
261.625 \* ((2<sup>(1/12)</sup>)<sup>\*\*2</sup>) // dos medios tonos sobre C = D  
261.625 \* ((2<sup>(1/12)</sup>)<sup>\*\*4</sup>) // cuatro medios tonos sobre C = E  
261.625 \* ((2<sup>(1/12)</sup>)<sup>\*\*5</sup>) // cinco medios tonos sobre C = F  
261.625 \* ((2<sup>(1/12)</sup>)<sup>\*\*7</sup>) // siete medios tonos sobre C = G  
261.625 \* ((2<sup>(1/12)</sup>)<sup>\*\*9</sup>) // nueve medios tonos sobre C = A  
261.625 \* ((2<sup>(1/12)</sup>)<sup>\*\*11</sup>) // once medios tonos sobre C = B  
261.625 \* ((2<sup>(1/12)</sup>)<sup>\*\*12</sup>) // doce tonos sobre C = C una octava arriba

Si escribimos en orden las potencias obtenemos esta serie:

0, 2, 4, 5, 7, 9, 11, 12

SuperCollider tiene el método .midiratio que simplifica el uso de  $2^{(1/12)}$

Así la tabla anterior se puede escribir de la siguiente forma:

261.625 \* 0.midiratio // cero medios tonos sobre C = C  
261.625 \* 2.midiratio // dos medios tonos sobre C = D  
261.625 \* 4.midiratio // cuatro medios tonos sobre C = E  
261.625 \* 5.midiratio // cinco medios tonos sobre C = F  
261.625 \* 7.midiratio // siete medios tonos sobre C = G  
261.625 \* 9.midiratio // nueve medios tonos sobre C = A  
261.625 \* 11.midiratio // once medios tonos sobre C = B  
261.625 \* 12.midiratio // doce medios tonos sobre C = C una octava arriba

También existe el método midicps que convierte el código de notas midi a frecuencias. El código de notas midi con su correspondiente nota se encuentran en la tabla 1. El código se extiende por debajo del 60 y por arriba del 71 alcanzando las octavas inferiores y superiores respectivamente

60.midicps=261.625

Podemos hacer sonar una escala en SuperCollider así:

```

p=ProxySpace.push(s)
// mayor
~escala={Demand.kr(Impulse.kr(2),0,Dseq([0,2,4,5,7,9,11,12].midiratio,inf))}
// menor Armonica
~escala={Demand.kr(Impulse.kr(2),0,Dseq([0,2,3,5,7,8,10,12].midiratio,inf))}
// menor Melodica
~escala={Demand.kr(Impulse.kr(2),0,Dseq([0,2,3,5,7,8,11,12].midiratio,inf))}

~sonido={SinOsc.ar(~escala.kr*60.midicps)}
~sonido.play
~sonido.stop

```

## Dinámica

La dinámica se refiere a la intensidad con la que se emite un sonido. Coloquialmente se utilizan palabras como “quedito”, “duro” o “fuerte” para hablar de esto. En la terminología musical clásica los términos que se usan no son muy diferentes, simplemente están en italiano. Para hablar de un sonido suave se usa la palabra “piano” que en español literalmente significa “plano” y cuyo símbolo es la letra *p*. Para los sonidos fuertes se usa la palabra “forte” que significa lo mismo en español y se usa la letra *f* para representarla. El instrumento de teclas negras y blancas que conocemos como piano originalmente se llamó *pianoforte* ya que podía tocar notas suaves y fuertes a diferencia del clavecín que solo podía tocar notas con una misma intensidad. Entre estas dos dinámicas, *p* y *f*, existe la subjetividad. Lo normal es que para un sonido muy suave se use *pp* (pianísimo), que es más suave que *p*, y para sonidos muy fuertes se use *ff* (fortísimo). Así mismo, para un sonido ni suave ni fuerte se usa *mf* (mezzoforte) que es medio fuerte. Existen compositores que incluso llegan a usar *ppp* y *fff*.

En SuperCollider la intensidad de los sonidos está normalizada. Es decir, 1 es el máximo y 0 el mínimo. El valor de la intensidad del sonido se le llama amplitud, ya que es la amplitud de una onda la que implica la fuerza que lleva, y mientras mayor la fuerza, mayor su intensidad. En los UGens el argumento *mul* es el que se encarga de asignarle valor a la amplitud de las ondas y por default se le asigna el valor de 1: la máxima intensidad antes de que comience a distorsionarse. Si queremos hacer una correspondencia entre valores de amplitud y dinámicas podríamos hacerlo de la siguiente forma:

```

{SinOsc.ar(440,0,mul:1)}.play           // ff
{SinOsc.ar(440,0,mul:0.75)}.play       // f
{SinOsc.ar(440,0,mul:0.5)}.play        // mf
{SinOsc.ar(440,0,mul:0.25)}.play       // p
{SinOsc.ar(440,0,mul:0.125)}.play      // pp

```

El uso de un valor normalizado para la amplitud a veces no refleja la experiencia del oído humano. Para mejorar esta respuesta se usan los decibelios que son una medida del volumen, siendo este una característica física del movimiento de partículas de aire en el espacio. 0

decibeles, o 0 *db* es el mximo de volumen. El oido humano percibe diferencias de volumen de 3 *db*, asi que si vamos bajando el volumen de un sonido obtenemos la siguiente regla:

0 *db*  
-3 *db*  
-6 *db*  
-9 *db*  
-12 *db*  
-15 *db*  
-18 *db*

El descenso de decibeles puede llegar al infinito, pero estando en -60 *db* practicamente ya no se escucha nada. En Supercollider se puede hacer uso de los decibeles con el mensaje *dbamp* que convierte decibeles a amplitud normalizada

```
{SinOsc.ar(440,0,mul:0.dbamp)}.play // ff  
{SinOsc.ar(440,0,mul:-10.dbamp)}.play // f  
{SinOsc.ar(440,0,mul:-20.dbamp)}.play // mf  
{SinOsc.ar(440,0,mul:-35.dbamp)}.play // p  
{SinOsc.ar(440,0,mul:-50.dbamp)}.play // pp
```

La dinamica tambien incluye cambios progresivos en la intensidad del sonido. Para un sonido cuya intensidad vaya creciendo se usa el termino *crescendo* y se usa la contraccion *cresc*. Para un sonido cuya intensidad vaya decreciendo se usa el termino *decrescendo* o *diminuendo* y se usan la contracciones *decr* o *dim*. Tambien se usan los simbolos conocidos como reguladores < y > para el *cresc* y el *dim* respectivamente.

```
{WhiteNoise.ar(Line.kr(0,1,5,1,0,2)).play // crescendo  
{WhiteNoise.ar(Line.kr(1,0,5,1,0,2)).play // diminuendo
```

Traten de escuchar si hay una diferencia entre los reguladores con amplitud normalizada y los que usan decibeles

```
{WhiteNoise.ar(Line.kr(-60.dbamp, 0.dbamp,5,1,0,2))}.play // crescendo  
{WhiteNoise.ar(Line.kr(0.dbamp, -60.dbamp,5,1,0,2))}.plot // diminuendo
```

## Densidad

La densidad se puede entender como la cantidad de eventos sonoros que suceden en un determinado tiempo.

Poca densidad:

```
{Dust.ar(1)}.play // poca densidad
```

```
{Dust.ar(20)}.play // mayor densidad
```

La densidad también se puede observar en el espectro de un sonido

```
{Dust.ar(20)}.play // poca densidad espectral  
{Decay2.ar(Dust.ar(20),0.01,1,BrownNoise.ar)}.play // mayor densidad espectral
```

O en el espectro armónico

```
{Mix((Pulse.ar(4)*SinOsc.ar(300*[0,7].midiratio,0,0.1)))}.play // poca densidad armónica  
{Mix((Pulse.ar(4)*SinOsc.ar(300*[0,4,7,11,14,17,21].midiratio,0,0.1)))}.play // mayor densidad armónica
```

### **Textura**

Las texturas son las diferentes combinaciones que se pueden hacer de los elementos mencionados anteriormente. Al hablar de textura se suele referir a elementos físicos, como cuando se habla de la textura de un muro liso o rasposo o con formas que emergen o con estrias... el elemento subjetivo aparece con mayor fuerza que en ningún otro de los parámetros que hemos descrito anteriormente. La densidad es parte importante a la hora de describir una textura. Se habla de texturas densas o texturas poco densas. La duración de los eventos sonoros pueden dar origen también a la percepción de diferencias en la textura. El contraste así mismo puede definir la textura de un gesto sonoro. En la música instrumental académica la textura es a veces vista como la cantidad de instrumentos que tocan en una pieza.

### **Contraste**

El contraste es uno de los elementos sonoros más importantes en cuanto a la claridad de las ideas se refiere. Hay que tener conciencia del contraste para poder hacer una pieza que contenga cierto balance. Para obtener un mínimo acercamiento a la idea musical o sonora que se desea es imprescindible tener siempre en cuenta el contraste. El contraste se puede encontrar o provocar en las alturas, en las dinámicas, en las densidades, en las texturas, en las formas... en todo lo hay. También es importante (igual de importante) tener claridad en cuanto a la no existencia del contraste, en caso de que sea deseado, o en caso de que sea un resultado secundario...

### **Espacialización (estereofonía y multicanal)**

Como vimos, el cerebro humano es capaz de formar mapas de la localización a partir del tiempo y fuerza con que llega el sonido a ambos oídos.

### **El sistema estéreo**

Consta de dos canales, uno izquierdo y uno derecho. Funciona con dos bocinas que emiten información diferente de la misma fuente sonora, si alguien se sienta en medio de dos bocinas

podrá percibir una imagen fantasma del sonido que sale justo en medio si el sonido es reproducido con la misma fuerza en ambas bocinas. Si se incrementa el volumen en una de ellas, por ejemplo la bocina derecha, percibimos que la imagen sonora se mueve a esa dirección. Esto sucede por la capacidad que tenemos de percibir la diferencia de intensidad.

Para colocar el sonido en un lugar determinado entre las dos bocinas usamos la el objeto Pan2.ar o panorama estéreo.

Pan2.ar(señal, panorama, amplitud)

```
//Sonido al centro
{Pan2.ar(Pulse.ar(4,0.5,0.5), 0, 1)}.play

//Sonido a la izquierda
{Pan2.ar(Pulse.ar(4,0.5,0.5), -1, 1)}.play

//Sonido a la derecha
{Pan2.ar(Pulse.ar(4,0.5,0.5), 1, 1)}.play

//Sonido moviendose de derecha a izquierda con Line.kr
{Pan2.ar(Pulse.ar(4,0.5,0.5), Line.kr(1,-1,5,doneAction:2),
1)}.play
```

## **Sonido Multicanal**

SC es capaz de gestionar varios canales. Con el objeto Out.ar es como direccionamos el sonido hacia el canal deseado.

A la práctica de colocar sonido en varias bocinas se le llama espacialización. La espacialización en SC la podemos gestionar a partir de varios objetos:

Pan2.ar // forma un panorama estéreo  
Pan4.ar // forma una cuadrifonía  
PanAZ.ar // realiza una distribución del sonido en un horizonte circula o azimhut  
PanB2.ar // señal Ambisonics en formato B

Más acerca de gestión de múltiples bocinas:

BEASTmulch: software basado en SuperCollider que controla el sistema BEAST  
Birmingham ElectroAcoustic Sound Theater

<http://www.birmingham.ac.uk/facilities/BEAST/index.aspx>