

# Música por computadora

Ernesto Romero y Hernani Villaseñor  
Centro Multimedia 2012

## Sesión 7

### Patterns

Los Patterns son patrones con los que podemos crear secuencias, los patrones o Patterns se pueden expresar de varias maneras y dependen a su vez de Patterns capaces de empaquetar el flujo de datos producido por distintos tipos de Patterns como es el caso de Pbind.

### Pbind (llave, patrón1, llave2, patrón2...)

Combina una serie de flujos (*streams*) provenientes de varios valores, en un solo *stream* de datos. La estructura Pbind se crea de la siguiente manera:

```
// en este ejemplo usamos un Synth que viene en SC, y comenzamos  
un stream con el valor de duración  
Pbind(\dur, 0.25).play
```

```
// agregamos el argumento freq  
Pbind(\dur, 0.25, \freq, 440).play
```

Lo que suena es un Synth establecido internamente en SC, los argumentos dur y freq ya están programados para crear flujos de datos concernientes a parámetros definidos, en este caso duración y frecuencia.

Hay varios argumentos que podemos utilizar y que están previamente programados:  
\dur, \freq, \amp, \pan, \sustain, \note, \midinote

Los métodos que utiliza Pbind son entre otros: .play, .stop, .clock, .quant

Podemos usar patrones para generar secuencia por ejemplo variar la duración de cada nuevo evento del *stream* así como su frecuencia, para lograr estos patrones usamos: Pseq, Prand, Pcollect, etc. Veremos dos.

### **Pseq ( [lista], repeticiones)**

Pertenece a ListPatterns y deposita valores en una lista de manera secuencial. Esta formado por una lista de donde tomará los valores en el orden que esten colocados, después indicamos cuantas veces queremos que suceda la secuencia de valores.

```
//controlamos la duración mediante una lista de valores
Pbind(\dur, Pseq([0.125,0.25,0.5,1],inf)).play
```

### **Prand ( [lista], repeticiones)**

Pertenece a ListPatterns y deposita valores en una lista de manera aleatoria. Es practicamente igual que Pseq pero con la variante de que los valores de la lista son leídos de manera aleatoria.

```
//controlamos la duración mediante una lista de valores
Pbind(\dur, Prand([0.125,0.25,0.5,1],inf)).play
```

### **Pdef**

Pdef es una rutina donde podemos anidar un Pbind y la cual podemos modificar y evaluar en tiempo real sin tener que detener SC, ya que es una clase que funciona mediante una referencia a un flujo de datos. Esta clase es parte de la librería JITLib.

```
// este es un modelo muy sencillo de un Pdef
Pdef (\secuencia,
      Pbind (\dur, Pseq([0.25,0.5],inf),
            \freq, Prand([200,250,300,350],inf)).play
```

```
// igualamos el Pdef a una variable
a=Pdef (\secuencia,
        Pbind (\dur, Pseq([0.25,0.5],inf),
              \freq, Prand([200,250,300,350],inf))
```

```
a.play
a.stop
```

Podemos modificar y evaluar en tiempo real el código interno del `Pdef` sin que tengamos que parar el servidor.

## Silencios

Los silencios son aplicables a llaves con modelos de altura como frecuencia o nota, cualquier símbolo es interpretado como silencio.

```
// nota como los siguientes símbolos dan silencio: \, \silencio
a=Pdef (\secuencia,
        Pbind (\dur, Pseq([0.25,0.5,0.5],inf),
              \freq, Pseq([300,\,350,400,450,\silencio],inf)))

a.play
a.stop
```

Otra forma de generar silencios es usar una lista antecedida del caracter #, y colocar una r donde queremos el silencio.

```
// nota como en el argumento freq usamos la r para indicar
silencio
a=Pdef (\secuencia,
        Pbind (\dur, Pseq([0.25,0.5,0.5],inf),
              \freq, Prand(#[200,r,250,300,350],inf)))

a.play
a.stop
```

## Patrones con nuestros instrumentos

Para usar sonidos diseñados mediante `SynthDef`, es necesario mandarlos al servidor con el mensaje `.add` por convención.

```
SynthDef(\triangulo,{|frec=400, amp=1, paneo=0|
  var sen, pan, env;
  sen=LFTri.ar(frec, 0, amp);
  pan=Pan2.ar(sen, paneo, amp);
  env=EnvGen.kr(Env.perc(0.1, 1), doneAction:2);
  Out.ar(0,pan*env)
}).add
```

Ahora solo necesitamos usar el argumento `\instrument` y el nombre de nuestro sintetizador, en este caso `\triangulo`, para añadirlo a la cadena de argumentos y valores de la estructura `Pdef`.

```
// nota que usamos \instrument con el nombre del synth para decirle a Pdef que reconozca nuestro sonido
```

```
(  
c=Pdef (\secuencia,  
      Pbind (\instrument, \triangulo,  
            \dur, Pseq([0.25,0.5,0.5],inf),  
            \freq, Prand([200,250,300,350],inf)))  
)  
  
c.play  
c.stop
```



Esta obra está sujeta a la licencia Attribution-NonCommercial-ShareAlike 3.0 Unported de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nc-sa/3.0/> o envíe una carta a Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.