



**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO**

---

---

Maestría en Música, campo de Tecnología Musical

***MAX-RHYTHMATIST: Instrumento Virtual  
Compuesto generador de patrones rítmicos  
programado en max/MSP.***

**TESIS**

Que Para Obtener el Título de

**Maestro En Música**

Con Especialidad en

**Tecnología Musical**

P R E S E N T A

**Eric Pérez Segura**



Tutor: Dr. Felipe Orduña Bustamante

OCTUBRE 2011



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## INDICE

<b>INTRODUCCIÓN.....</b>	<b>5</b>
PROPÓSITO Y OBJETIVO DEL ESTUDIO.....	5
DESCRIPCIÓN DEL TRABAJO DE TESIS.....	5
<b>CAPÍTULO 1 ASPECTOS TEÓRICOS DEL RITMO.....</b>	<b>8</b>
1.1 ASPECTOS CONCEPTUALES DEL RITMO.....	8
1.1.1 Terminología básica.....	8
1.1.2 Diversas concepciones del ritmo.....	11
1.2 DESARROLLO HISTÓRICO MUSICAL DEL RITMO EN EL SIGLO XX.....	15
1.2.1 Procedimientos de elaboración rítmica en la música de concierto del siglo xx.....	23
1.3 INVESTIGACIONES EN EL ÁMBITO PSICOACÚSTICO.....	28
1.3.1 Sincronización.....	29
1.3.2 El presente subjetivo.....	29
1.3.3 Visión científica de los principales atributos del ritmo.....	31
1.3.4 Modelos teóricos de representación del ritmo.....	38
1.4 COMPOSICIÓN ALGORÍTMICA E INSTRUMENTOS VIRTUALES.....	43
1.4.1 Composición asistida por computadora.....	43
1.4.2 Criticas a la composición asistida por computadora.....	47
1.4.3 Instrumentos Virtuales.....	49
1.5 ANÁLISIS Y OPINIONES ACERCA DEL ESTADO DEL ARTE.....	63
1.5.1 Sobre la concepción y las investigaciones referentes al ritmo.....	63
1.5.2 Sobre las implicaciones que el estado del arte tuvo en la concepción del modelo	

**MAX-RHYTHMATIST**

aritmético en esta tesis.....	64
1.5.3 Sobre las implicaciones que el estado del arte tuvo en el diseño de la aplicación informática.....	66
1.5.4 Sobre las implicaciones que el estado del arte tuvo en la elección del lenguaje de programación.....	67
1.6 OBJETIVOS DEL TRABAJO DEL TRABAJO DE TESIS.....	69
1.7 PREMISAS DE TRABAJO.....	70
<b>CAPÍTULO 2 MÉTODOS DE GENERACIÓN RÍTMICA.....</b>	<b>71</b>
2.1 PRINCIPIOS BÁSICOS.....	71
2.2 REPRESENTACIÓN MATEMÁTICA Y GRÁFICA.....	72
2.3 CLASIFICACIÓN DE LAS SERIES PARA GENERAR PATRONES RÍTMICOS.....	74
2.3.1 Series regulares.....	74
2.3.2 Series Irregulares.....	76
2.3.3 Series transformadas (Snt).....	78
2.3.4 La Serie Rítmica Combinada (SRCn).....	80
<b>CAPÍTULO 3 IMPLEMENTACIÓN INFORMÁTICA DE LOS MÉTODOS DE GENERACIÓN RÍTMICA.....</b>	<b>82</b>
3.1 IMPLEMENTACIÓN DE LAS SERIES REGULARES SIMPLES Y COMPUESTAS.....	82
3.2 IMPLEMENTACIÓN DE LAS SERIES IRREGULARES.....	83
3.2.1 Implementación de Series Irregulares de duraciones acumuladas.....	83
3.2.2 Implementación de Series Irregulares de duraciones individuales.....	84
3.3 IMPLEMENTACIÓN DE LAS SERIES TRANSFORMADAS.....	85
3.4 GENERACIÓN DE SECUENCIAS MIDI Y AUDIO.....	88

## MAX-RHYTHMATIST

3.4.1 Integración de los submódulos de generación rítmica en un módulo general.....	88
3.4.2 Submódulos MIDI.....	89
3.4.3 Submódulos de Audio.....	90
3.5 EJECUCIÓN DE PATRONES RÍTMICOS MIDI POR MEDIO DE UN CONTROLADOR EXTERNO.....	92
3.5.1 Descripción de módulo externo de ejecución.....	92
3.5.2 Descripción del submódulo aleatorio.....	93
3.5.3 Descripción del submódulo por alturas.....	94
3.5.4 Descripción del submódulo por intervalos.....	95
3.5.5 Ejecución de muestras de audio con el módulo externo de ejecución.....	97
<b>CAPÍTULO 4 LA INTERFAZ DE USUARIO.....</b>	<b>99</b>
4.1 ELEMENTOS CONSTITUTIVOS DE LA INTERFAZ DE USUARIO.....	99
4.2 COMENZANDO A UTILIZAR EL MAX-RHYTHMATIST.....	102
4.2.1 Instalación y guía rápida.....	102
4.2.2 Panel inicial.....	106
4.3 GENERANDO SERIES MANUALMENTE.....	107
4.4 GENERANDO SERIES ALEATORIAS IRREGULARES.....	109
4.5 GENERANDO SERIES ALEATORIAS UNIFORMES.....	111
4.6 GENERANDO SERIES REGULARES.....	112
4.7 CAMBIANDO SERIES CON EL MÓDULO DE TRANSFORMACIONES.....	113
4.8 USANDO EL MULTI-TRACK MIDI.....	115
4.9 USANDO EL MULTI-TRACK AUDIO.....	117
4.10 USANDO EL MÓDULO EXTERNO DE EJECUCIÓN.....	119

MAX-RHYTHMATIST

<b>CAPÍTULO 5 EVALUACIÓN DE MAX-RHYTHMATIST.....</b>	<b>125</b>
5.1 TRANSCRIPCIÓN DE UN RITMO ESCRITO EN NOTACIÓN MUSICAL A NOTACIÓN NUMÉRICA.....	125
5.2 TRANSCRIPCIÓN DE UN RITMO ESCRITO EN NOTACIÓN NUMÉRICA A NOTACIÓN MUSICAL.....	127
5.2.1 Transcripción de una serie regular compuesta.....	127
5.2.2 Transcripción de series irregulares.....	129
5.3 ELABORACIÓN Y TRANSCRIPCIÓN DE TEXTURAS MUSICALES.....	133
5.3.1 Textura polifónica con poli-tempos, aceleraciones y desaceleraciones.....	133
5.3.2 Textura con compases compuestos y reacomodo de acentos.....	134
5.4 EVALUACIÓN DE MAX-RHYTHMATIST.....	136
5.4.1 Ventajas de Max-Rhythmatist.....	136
5.4.2 Limitaciones de Max-Rhythmatist.....	139
<b>CONCLUSIONES: LINEAS FUTURAS DE INVESTIGACIÓN Y DESARROLLO DE MAX-RHYTHMATIST.....</b>	<b>141</b>
EXPANSIÓN DE LAS FUNCIONES DE MAX-RHYTHMATIST.....	141
INVESTIGACIÓN TEÓRICA A REALIZAR.....	143
CONSIDERACIONES FINALES.....	144
<b>BIBLIOGRAFIA.....</b>	<b>146</b>

## INTRODUCCIÓN

### **PROPÓSITO Y OBJETIVO DEL ESTUDIO**

Este trabajo de tesis plantea:

*La programación de un generador de ritmo complejo, basado en la técnicas de elaboración rítmica más representativas de la música de concierto del siglo XX.*

Los objetivos del trabajo son:

- Definir las características generales que definen al ritmo como elemento musical, sin importar el tipo de lenguaje musical.
- Definir los procedimientos que caracterizan la elaboración rítmica en la música de concierto del siglo XX.
- Elaborar modelos aritméticos que ayuden a representar los aspectos que conforman las estructuras rítmicas.
- Seleccionar el programa o lenguaje de programación más adecuado para la implementación de modelos aritméticos.

Todo esto con el propósito de elaborar un programa de cómputo que permita diseñar, manipular, ejecutar y grabar estructuras rítmicas, tanto de carácter controlado, así como aleatorio.

### **DESCRIPCIÓN DEL TRABAJO DE TESIS**

El primer capítulo es una aproximación a las investigaciones teóricas relacionadas con el ritmo. En primera instancia, se abordan los aspectos puramente musicales, definiciones de términos básicos, y una semblanza de aquellos compositores del siglo XX cuyas obras involucraron un manejo especial de los aspectos rítmicos. Luego, se hace una abstracción de las técnicas de manejo rítmico derivadas del análisis de obras de los compositores de esa época. Se abordan aspectos científicos del ritmo,

## MAX-RHYTHMATIST

muchos de ellos relacionados con la psicoacústica, es decir, la forma en que son percibidos los fenómenos rítmicos. Posteriormente, se analizan las características de varios modelos de representación rítmica con el fin de evaluar cuál de ellos es más conveniente para iniciar el diseño de una implementación informática. Luego, se hace una recopilación de aspectos relacionados con la Composición Asistida por Computadora y se describen las características de algunos lenguajes de programación enfocados a este aspecto. Finalmente, se habla de las características de los Instrumentos Musicales Virtuales, se hace un recuento de lenguajes de programación para su diseño. Para concluir el capítulo, se hace un análisis de la información recabada y se establecen los criterios para teóricos y prácticos para desarrollar la aplicación informática.

En el segundo capítulo se plantea una forma de representar eventos rítmicos de manera gráfica y numérica. Para ello se clasifican diversos tipos de series numéricas para representar patrones rítmicos, desde aquellos cuyos eventos ocurren de manera regular, hasta en los que ocurren aleatoria e irregularmente. Se define una nomenclatura para representar las series, así como las características de cada una.

En el tercer capítulo se procede a describir la manera en que se generarían las series en el programa de cómputo, así como la manera de integrarlos para crear secuencias MIDI o audio. En cada paso descrito se coloca un diagrama de flujo representando las funciones principales de cada módulo programado.

En el cuarto capítulo se describe el funcionamiento de la interfaz gráfica. Dicha descripción puede ser utilizada como un manual de usuario.

En el quinto capítulo se evalúa la eficiencia del programa al momento de transcribir patrones rítmicos transcritos con la escritura musical tradicional, así como el paso inverso, la transcripción a escritura tradicional de una serie numérica generada por el programa. De igual forma, se pone a



### *MAX-RHYTHMATIST*

prueba el programa cuando este se utiliza para generar texturas musicales específicas. Al final, se incluye una descripción detallada de las fortalezas y limitaciones del programa de cómputo.

En las conclusiones se plantea la manera de expandir las funciones del programa generador de ritmo.

Se describen los aspectos teóricos que podrían investigarse para integrar aspectos relacionados con el manejo de alturas, timbres y ataques. Así mismo, se describen las características que deberá cumplir el programa de cómputo para integrar dichos aspectos, así como las conclusiones.

## CAPÍTULO 1

### ASPECTOS TEÓRICOS DEL RITMO

#### 1.1 ASPECTOS CONCEPTUALES DEL RITMO

##### 1.1.1 Terminología básica

Comencemos por establecer el término "estructura rítmica" para referirnos a todos los aspectos objetivos y subjetivos relacionados con el ritmo como elemento musical. Este término deriva de los estudios realizados por Leonard B. Meyer y Grosvenor Cooper en su libro "La estructura rítmica de la música", en donde utilizan la palabra "estructura" para referirse a la generalidad total, y al término "ritmo" como un aspecto constituyente de la estructura rítmica. Ambos autores mencionan que hay una confusión generalizada entre los conceptos de "ritmo" y "métrica". Esta confusión es explicada por Meyer y Cooper como resultado de haber llamado "rítmicos" a los aspectos relacionados con la organización temporal que se ocupan de la medida, siendo que estos aspectos debieron haberse denominado "métricos" (Meyer & Cooper, 2000. Pág 9, 13). Meyer y Cooper enumeran y explican los siguientes aspectos, que nos servirán para conformar una terminología básica para el análisis de la estructura rítmica de la música:

- *Niveles arquitectónicos*: Se entiende este concepto como el análisis de los diversos niveles de subdivisión o multiplicación rítmica, que dan como resultado la creación de una estructura métrica. Por ejemplo, un compás de 3/4 se diferencia de uno de 12/8 en que el primero está compuesto por tres unidades de compás de 2/8 en un nivel inferior, mientras que el segundo está compuesto por dos unidades de compás de 3/8 en un nivel inferior. Para que los diferentes niveles arquitectónicos que conforman a la estructura métrica puedan ser percibidos, entran en juego otros elementos, que Meyer y Cooper definen como "modos de

### MAX-RHYTHMATIST

organización temporal", estos son el pulso, el compás y el ritmo. (Meyer & Cooper, 2000. Págs 10,11)

- *Pulso*: Meyer y Cooper definen este concepto como "cada uno de los estímulos exactamente equivalentes, que se repiten en una serie regular", a partir de esto concepto, ellos comienzan a enfatizar la relación entre los estímulos objetivos, es decir, los sonidos, y las sensaciones que pueden existir de manera subjetiva en el oyente. Para ejemplificar esto, ellos mencionan que incluso después de que un sonido breve, emitido en intervalos de tiempo regulares, se haya detenido, la sensación de regularidad se mantiene tanto en la mente como en la musculatura del oyente. Es decir, el pulso tiene un carácter ambivalente porque representa tanto un hecho psicológico como un hecho físico. (Meyer & Cooper, 2000. Págs 12,13)
- *Compás*: En palabras textuales de Meyer y Cooper, el compás debe ser entendido como "la medida del número de pulsos existentes entre los acentos que aparecen con una recurrencia más o menos regular" (Meyer & Cooper, 2000. Pág 13). Esto nos muestra, la intrincada relación que hay entre los conceptos que se han enumerado y los que están por enumerarse, ya que el concepto de acento, cómo se discutirá posteriormente, también tiene el mismo carácter ambivalente que el pulso, y sirve para definir el concepto de compás, de la misma manera en que los conceptos de pulso y compás sirven para definir el concepto de niveles arquitectónicos, y a su vez, el compás conforma la estructura métrica que es de naturaleza arquitectónica, puesto que "las partes que miden el compás pueden dividirse en unidades iguales, o ser dispuestas en unidades métricas mayores" (Meyer & Cooper, 2000. Pág 14).
- *Ritmo*: Definir el concepto de ritmo parece ser una tarea más complicada de lo que parece. Por ello, el limitar la definición de este término a la dada por Meyer y Cooper tiene la simple finalidad de dar consistencia al vocabulario que se utilizará a lo largo de esta tesis. Para ello

### MAX-RHYTHMATIST

se tiene que introducir, en primera instancia, el concepto de pie poético, normalmente relacionado con la poesía, que es un patrón de acentuación que contiene sílabas cortas y largas. Los cinco pies rítmicos tradicionalmente asociados a la prosodia greco-latina son:

Yambo	corto + largo
Anapesto	corto + corto + largo
Troqueo	largo + corto
Dáctilo	largo + corto + largo
Anfibraco	corto + largo + corto

Lo más interesante en Meyer y Cooper es que basan su definición en el uso de estos pies poéticos para justificar que el ritmo es independiente, tanto del compás como del pulso. En el caso de su independencia respecto al compás, mencionan que el uso de los pies poéticos puede existir sin la necesidad de concebir un compás regular. Y en el caso del pulso, mencionan que la acentuación de estos pies y su combinación con elementos no acentuados se puede dar sin que exista una regularidad en la aparición de los acentos (Meyer & Cooper, 2000. Pág 16,17). Un ejemplo histórico de esta independencia, y del cual nos ocuparemos más adelante, es el canto gregoriano. En consecuencia, Meyer y Cooper definen al ritmo como "el modo en el cual una o más partes no acentuadas son agrupadas en relación con otra que sí lo está". (Meyer & Cooper, 2000. Pág 15)

- *Acento*: El problema de este concepto es que intervienen muchos elementos en su percepción al momento de escuchar música, entre los que podemos citar la duración, la intensidad, el perfil melódico, y la regularidad. En otras palabras, nos encontramos con un fenómeno subjetivo cuya percepción es relativa. Meyer y Cooper mencionan que "únicamente puede haber acentos si hay partes débiles, y viceversa", pero a pesar de la relatividad del fenómeno,

### MAX-RHYTHMATIST

elaboran una definición de acento en los siguientes términos: "un acento es un estímulo (dentro de una serie de estímulos) que está marcado para ser advertido de algún modo" (Meyer & Cooper, 2000. Pág 18, 19). La subjetividad y relatividad de este fenómeno, y por tanto, la dificultad de expresarlo en términos cuantificables y objetivos, fue un factor importante para excluirlo como un elemento a considerar en el diseño del patch en Max/MSP, tema principal de esta tesis.

- *Énfasis dinámico*: Comúnmente conocido en el medio musical como dinámica, este concepto no debe ser confundido con el de acento, ya que el énfasis dinámico se puede dar tanto en partes acentuadas, como en no acentuadas. Normalmente este énfasis se mediante un cambio gradual (de ahí el término 'dinámico') en la intensidad sonora de un evento musical. En palabras de Meyer y Cooper, este fenómeno es "la intensificación dinámica de una parte, acentuada o no". (Meyer & Cooper, 2000. Pág 19).
- *Agrupamiento*: Nuevamente nos encontramos con un fenómeno de carácter subjetivo, caracterizado por una relatividad similar a la que se discutió sobre el fenómeno de la acentuación. Meyer y Cooper mencionan que el la percepción de agrupamiento es resultado de "la semejanza y la diferencia, la proximidad y la separación de los sonidos percibidos por los sentidos y organizados por la mente" (Meyer & Cooper, 2000. Pág 20). Este término, a pesar de provocar divergencias en su interpretación, resultará ser uno de los más reiterados a lo largo de su libro y de importancia capital para las disertaciones teóricas de ambos.

#### **1.1.2 Diversas concepciones del ritmo**

Dentro de la bibliografía disponible, también se encontraron autores e ideas que en ciertos aspectos reforzaban las ideas de Meyer y Cooper discutidas en las páginas anteriores, y en otros aspectos eran divergentes.

### MAX-RHYTHMATIST

Platón en "Las Leyes" (Platón, 1988. Pág. 96) describe el ritmo como "un orden en movimiento" o un orden dentro del movimiento". A pesar de que no explica qué tipo de orden o movimiento, la formulación de Platón tiene similitudes con la de Dewey (Dewey,1980. Pág 61), el cual dice que el ritmo es una "variación ordenada de cambios". Dewey reconoce un aspecto "dinámico pero no humanístico" que concibe al ritmo como un "orden orgánico". La concepción orgánica de este autor tiene sus orígenes en el orden rítmico en aspectos genéticos y fisiológicos, como por ejemplo, la existencia del pulso o la respiración, así como en la alternancia entre tensión y relajación características de estos procesos corporales. En resumen, Dewey propone que el ritmo es un orden dentro del movimiento que puede ser percibido a través de uno o más de nuestros sentidos, y que tiende a expresar o generar una participación o involucramiento de la persona que lo produce o lo experimenta; el ritmo se lograría cuando se impongan acentos en una secuencia de sonidos o movimientos regulares.

En oposición, Malcom Budd (Budd, 2003. Págs. 209-223), propone que el ritmo se puede concebir como una conjunción entre duraciones y acentos, completamente independiente de la persona que los genera o percibe, y que por lo tanto, implica la existencia del ritmo de como algo abstracto y estático. En esa misma línea de pensamiento se encuentra Boghossian (Boghossian, 2002.Págs. 49-55), que menciona que el hablar de "movimiento musical" es una metáfora que no explica nada y que puede ser eliminable, y sostiene que el ritmo en música es un proceso intencionalmente diseñado, lo cual no implica una concepción humanista, ya que Boghossian concluye que este proceso es independiente de los participantes.

Un tercer punto de vista proviene de Hamilton (Hamilton, 2007. Págs. 50-72), en oposición a la visión orgánica, que sugiere que el ritmo es una concepción humana, envuelta en comportamientos y prácticas humanas. "El ritmo de las estaciones", según este autor, es una metáfora, ya que para poder

### MAX-RHYTHMATIST

percibir un fenómeno como rítmico, uno debe de estar familiarizado con el ritmo tal como se manifiesta en música, danza o en los movimientos corporales. Esta concepción humanista trata al ritmo como un "orden distintivo de movimientos humanos o movimientos dentro del sonido", en otras palabras, un orden generado a través de la imaginación del ser humano y proyectado en procesos que no "poseen" dicho orden de manera literal. Esta idea de "proyecto" se refiere, según Hamilton, a que las personas experimentan la sucesión de sonidos o movimientos como rítmicos en buena parte porque ellos imaginan que es así.

Sobre la relación entre imaginación y percepción, Scruton (Scruton, 1999. Págs. 45-56), en una línea similar a Hamilton, dice que nosotros "proyectamos el ritmo sobre sonidos regulares", es decir, una vez que percibimos regularidad, nuestra imaginación impone la idea de ritmo. Sin embargo, menciona que puede llegar a haber confusión cuando dos tipos de imposición o proyección imaginativa entran en conflicto: la percepción cotidiana, y la percepción técnico y filosófica. Es decir, si el ritmo es lo suficientemente complejo, la imaginación, más que la percepción, se ve comprometida, ya que requiere un "esfuerzo imaginativo la percepción de ritmos provenientes de la Nueva Complejidad o de la música Balinesa para gamelán". En estos casos la percepción del ritmo puede ser "multi-aspecto".

En cuanto a los atributos del ritmo (pulso, acento, metro, etc.), Hamilton (Hamilton, 2007. Págs. 50-72) sostiene que son tan difíciles de definir como el ritmo. Estas son nociones internamente relacionadas y forman un "holismo conceptual", tanto como lo es el ritmo y la música. Es decir, los dos conceptos son adquiridos gradual y simultáneamente. La declaración de un holismo conceptual entre A y B implica entonces que para definir o entender A se asume un entendimiento de B, y viceversa. Esta es una "circularidad benigna, esencial al lenguaje" en palabras de Hamilton. En conclusión, Hamilton dice esta circularidad le confiere al ritmo no sólo un carácter humanista, sino

### MAX-RHYTHMATIST

también un carácter dinámico, en concordancia con la definición de Platón mencionada al principio de este apartado.

Por otro lado, Clayton (Clayton, 1996. Págs. 323-332) introduce el concepto de "ritmo libre", al mencionar que existe un gran acervo de música y poesía que no parecen ser métricas. Mientras que la música occidental tiende a ser explícitamente métrica, con pulsos claramente articulados, en música no occidental, este modelo es más ambiguo, y el ritmo libre o el tempo libre es común, entendiendo al "ritmo libre" como un pulso flexible, es decir, una aceleración o una desaceleración relativa a un pulso o a varios pulsos básicos. En este sentido, el concepto de ritmo libre sería similar al "rubato", término muy usado dentro de la música occidental para referirse a la alteración expresiva del ritmo o el tempo. Este concepto de ritmo libre no excluye la idea de pulso, pero cuestiona la idea de la regularidad de su aparición para que la idea de ritmo aparezca.

Un concepto importante que une la concepción dinámica con la idea de ritmo libre es la "estásis", definida por Hamilton como un punto intermedio entre el continuum cinético y el ritmo, y a su vez, lo define como el mínimo tipo de ritmo (Hamilton, 2007. Págs. 50-72). Para aclarar esto, Hamilton menciona que los límites del orden rítmico son, por un lado, el caos cinético, como el que se da en una avalancha, una explosión o en el movimiento aleatorio Browniano de las partículas suspendidas en un fluido, o el ruido generado aleatoriamente; y por otro lado, el continuum cinético, como en el movimiento suave de un yate, o en un glissando musical. Considera que el caos es demasiado variante como para ser considerado como ordenado, mientras que el continuum es el polo opuesto al presentar la mínima variación. Mientras que el ejemplo del yate y el glissando son demasiado continuos para ser tomados como un orden rítmico, el cual envuelve un elemento de discontinuidad o discreción de los elementos, la Estásis es discontinua y más cercana al ritmo. Para ejemplificar la Estásis, Hamilton menciona al canto llano, en donde hay puntos de reposo y movimiento, de tal



## MAX-RHYTHMATIST

forma que la música no es fuertemente acentuada, sin embargo, los ejecutantes inevitablemente hacen, al menos, un "mínimo de propulsión". Rítmicamente, el canto llano tiene una serie de reposos, creando un tipo de rubato, y está fuertemente influenciado por los patrones de acentuación y entonación del lenguaje hablado, el cual muestra más variaciones que el metro musical, ya que tanto en oratoria como en una conversación normal, el pulso varía. Esta idea de la mínima propulsión acomoda el concepto del ritmo libre dentro de la concepción dinámica del ritmo.

### **1.2 DESARROLLO HISTÓRICO MUSICAL DEL RITMO EN EL SIGLO XX**

Un autor que pone en claro la importancia de la estructura rítmica en todos los niveles de construcción musical es Fernando Martínez Soques, en su libro "Método de canto gregoriano", coincidiendo con muchas de las ideas de Meyer y Cooper. En su libro se menciona cómo en los cantos gregorianos la concepción rítmica se da desde la generación de los melismas hasta la generación del canto en su totalidad, a manera de una construcción arquitectónica, que él denomina síntesis rítmica (Martínez Soques, 1943. Pág 46, 192). En la figura 1 se muestra un esquema rítmico de un fragmento de un canto gregoriano.

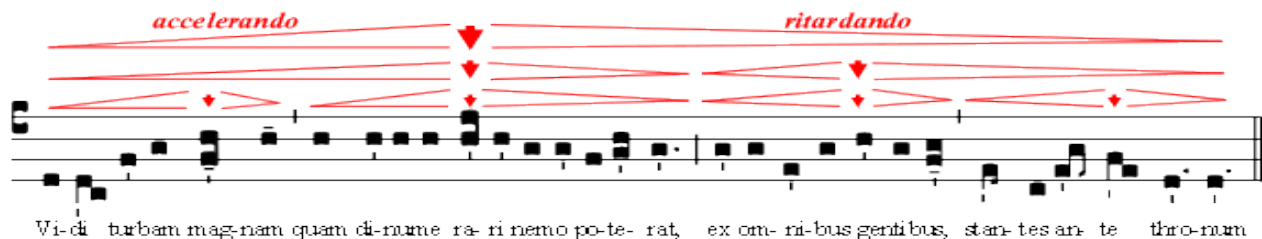


FIG.1. Esquema rítmico de un fragmento de canto gregoriano.

Este esquema muestra que una línea melódica gregoriana está formada por elementos ársicos ó de carácter ascendente, y téticos o de carácter descendente. La unión de pequeños elementos que combinan ambas características nos dan como resultado elementos más grandes que siguen conservando las características ya mencionadas, dando como resultado una línea melódica que llega

### MAX-RHYTHMATIST

a un punto culminante o ápice, como resultado de la interacción de elementos más pequeños. La figura 1 muestra de manera esquemática como se da un proceso de aceleración y desaceleración, siendo entonces el canto gregoriano un ejemplo claro de ritmo libre de acuerdo a la definición de Clayton, explicada en el apartado anterior. La concepción del tiempo libre también se presenta en la música del siglo XX, tal como lo señala Abromont, al mencionar que el compositor Pierre Boulez distingue entre un *temps lisse* (tiempo liso) y *temps strie* (tiempo estriado: que se contrae). Abromont señala que la concepción rítmica de Boulez se relaciona con el canto llano, que ellos lo denominan "canto libre", ya que el ritmo de este canto se compone tanto de notas largas y breves como de sílabas acentuadas, pero dentro de un "tiempo subjetivo", no cuantificado, un tiempo relacionado con el lenguaje. (Abromont, et al, 2005. Pág. 273)

Dentro de las investigaciones musicológicas que describen el desarrollo histórico del ritmo en el siglo XX, ya sea de manera directa o tangencialmente, hay un acuerdo general en señalar a varios compositores a partir de los cuales se dan manifestaciones rítmicas específicas. Entre ellos podemos mencionar a:

1. Stravinsky y Bartok, con la incorporación de patrones rítmicos tomados de la música popular de Europa del Este, con un marcado juego de cambios de acentuación, compases compuestos y superposición de patrones con niveles de subdivisión distinta. La influencia de ambos compositores se dejó sentir en la rítmica usada por los compositores representantes del nacionalismo mexicano, especialmente Chávez y Revueltas (E. Antokoletz, 1998. Págs. 218-228).

## MAX-RHYTHMATIST

The musical score consists of three systems of piano music. The first system begins with a treble clef staff and a bass clef staff. The first staff has a treble clef and the second has a bass clef. The music is written in a complex, irregular meter. The first system includes dynamic markings: *p ma espr.*, *pp*, and *p dolce*. The second system includes *mf quasi parlando* and *(non legato)*. The third system includes *cresc.*, *mf*, *mf*, *molto espr.*, and *più lento (♩ = 56) espr.*. The score is marked with a '5' above a measure in the third system and a '\*' below a measure in the same system.

FIG. 2. Bartok, Improvisations op. 20, no. 3, compases 16-30

La figura 2 muestra un extracto de Bela Bartok en donde podemos observar cambios de compás a todo lo largo de este pasaje, dentro de ellos hay compases compuestos, como el de 5/8. De igual manera se presentan cambios de metrónomo, como en el último compás del pasaje, y en el compás previo un acento en el tiempo débil del compás. También es importante notar el quintillo del primer compás del último sistema<sup>1</sup>. Todo esto da como resultado una obra en donde el aspecto rítmico es ampliamente desarrollado.

2. Olivier Messiaen, quien en su libro "La técnica de mi lenguaje musical", describe la forma en que genera patrones rítmicos mediante la adición o sustracción de valores, retrogradación, y aumentación o disminución por factores no enteros (Messiaen, 1944. Págs. 16-17). En su obra tardía también surge la idea de serialización de todos los parámetros musicales, incluyendo el ritmo, idea que seguirían numerosos compositores de la posguerra. (E. Antokoletz, 1998. Págs. 370-373).

<sup>1</sup> Se usa el término "sistema" para designar la agrupación de compases en un mismo renglón.

MAX-RHYTHMATIST

Un peu vif

Père Fils (aiment - verbe aimer)

Père Fils (ablatif: par)

Saint Esprit (amour - verbe aimer)

p r o c é d a n t Père

Fils (aiment - verbe aimer) r a c e

h u m a i n e (ablatif: par)

FIG. 3. Messiaen, Meditations No.7, p. 61

En el ejemplo mostrado (Fig. 3), Messiaen utiliza patrones rítmico-melódicos a los cuales les asigna un nombre, estos se van sucediendo uno tras otro. Cada patrón está dividido por una barra de compás; sin embargo, la duración de cada compás no es la misma, por lo que la subdivisión de compases es sólo una referencia para identificar cada patrón rítmico. Por otro lado, los motivos rítmicos que conforman a cada patrón contienen combinaciones de figuras rítmicas cuyos valores sumados no corresponden a los tradicionales múltiplos de 2 o 3. Por ejemplo, en el primer compás, el primer motivo rítmico agrupa a un octavo, un dieciseisavo, y un octavo, dando como resultado un motivo de 5 dieciseisavos, seguido de otro de otro de siete dieciseisavos, y otro de un cuarto, por lo que si asumimos que cada motivo recibe una

### MAX-RHYTHMATIST

acentuación inicial, los acentos caen a intervalos irregulares. El resultado final, es una melodía en donde el escucha no percibe un pulso regular.

En la figura 4, Messiaen utiliza series de sonidos, dinámicas, valores rítmicos y articulaciones. Cada evento, por lo tanto, es una entidad individual, cuya relación con el resto de los eventos estará justificada por la serie utilizada. En el caso del ritmo, cada valor de la serie es distinto en duración, y cuando alguno de ellos llega a repetirse como resultado de una nueva utilización de la serie, este valor rítmico se encuentra en un contexto diferente. Por ejemplo, la primera nota de la partitura (mi bemol con valor de treintaidosavo con dinámica en ppp) se vuelve a repetir en el último compás del extracto, y aunque las notas que le suceden en la mano derecha son las mismas, dado que pertenecen a una misma serie, las notas de la mano izquierda son distintas, además de que dicha nota cae en un momento del compás distinto, por lo que el resultado sonoro en ambos compases es diferente. Por otra parte, las barras de compás se vuelven una mera referencia para el intérprete, ya que aunque indican espacios de 2/4, los eventos rítmicos no ocurren necesariamente en los tiempos fuertes de dicho compás.

**Mode de valeurs et d'intensités**

Modéré (♩ = 120)

© 1990 Éditions Durand  
© 2000 Éditions Durand (édition révisée)  
Paris, France

DF 15848

Dépôt légal 2133  
Tous droits réservés pour tous pays.

FIG. 4. Messiaen, extracto de la obra "Modo de valores e Intensidades", para piano solo.

3. Conlon Nancarrow, quien con la ayuda de pianolas diseñó patrones rítmicos sobre melodías que aceleraban y desaceleraban en distintos momentos, o bien, iban a velocidades distintas. Posterior a la elaboración de los rollos de pianola, la obra de Nancarrow fue transcrita a notación tradicional, y ejecutada por humanos (C. Sandoval, 1994. Págs. 148-175).

MAX-RHYTHMATIST



FIG. 5. Nancarrow, detalle de la transcripción del segundo movimiento de la obra "Three movements for chamber orchestra".

En este ejemplo (Fig. 5), las tres líneas melódicas diseñadas por Nancarrow van a una velocidad distinta, haciendo imposible la utilización de barras de compás, a eso agreguemos que cada uno de los eventos rítmicos en cada línea melódica van cambiando su valor o duración. Para poder lograr una transcripción usando notación tradicional, se utilizan pequeñísimas unidades de tiempo para poder insertar los eventos rítmicos en un lugar preciso de la partitura.

4. Bryan Feneyhough, quien diseña patrones rítmicos por superposición de patrones en distintos niveles, con diferente subdivisión, y que además juega con cambios de compás y métrica; todo ello para generar un alto nivel de complejidad en su ejecución (S. Schick, 1994. Págs. 132-153)

*MAX-RHYTHMATIST*

"Tout est hiéroglyphique" (Baudelaire)

Lemma-Icon-Epigram

For Massimiliano Damerini

The image displays a complex musical score for piano solo, titled "Lemma-Icon-Epigram" by Brian Ferneyhough. The score is written in a single system with two staves, the right and left hands. It features a variety of rhythmic patterns and dynamic markings. The tempo is marked as "♩ = ca. 50". The score is divided into measures, with some measures containing multiple rests or complex rhythmic figures. The dynamics range from *ppp* (pianississimo) to *fff* (fortississimo). The score includes various performance instructions such as "leggero", "poco", "agitato", and "rit.". The score is published by Edition Peters, No. 3223, and is copyrighted by Hinrichsen Edition, Peters Edition Ltd., London, in 1992.

FIG 6. Ferneyhough, extracto de su obra "Lema-Icon-epigram" para piano solo.

En la figura 6 observamos una obra para piano de Brian Ferneyhough, en donde podemos observar cambios de compás continuos, y una proliferación de polirritmos. Por ejemplo, en el compás 1, la mano derecha va pasando por ritmos subdivididos en 12, 11 y 10



### MAX-RHYTHMATIST

sesentaicuatrosavos, dentro de estas subdivisiones, la misma mano derecha hace dentro de la subdivisión de 10, un fragmento melódico subdividido en 6 dieciseisavos en el espacio correspondiente a 4. Mientras que la mano izquierda, hace tresillos en cada gran subdivisión de la mano derecha, por lo que cada tresillo se toca de manera distinta, dado que están dentro de subdivisiones distintas. Procesos similares al descrito se dan en cada uno de los compases de este extracto.

5. El jazz también ha tenido influencia en la música contemporánea (C. Folio, 1995. Pág 103-134), así como los ritmos derivados de culturas no occidentales, como los ritmos de las culturas africanas, los patrones rítmicos hindús, así como aquellos provenientes de Oriente Medio (N. Sorrel, 1980. Pág 190). En el caso específico del Jazz, el efecto de "swing", heredado del Blues, permite que aunque la partitura esté escrita en compás binario (2/4, 3/4 o 4/4), el intérprete la ejecute como si estuviera en compás ternario, siendo además este efecto de "ternarización", variable de ejecutante a ejecutante, por lo que la partitura se vuelve una guía flexible para la ejecución rítmica (C. Abromont, 2005. Págs, 220-232).

#### **1.2.1 Procedimientos de elaboración rítmica en la música de concierto del siglo xx.**

Basándose en el análisis de los fenómenos rítmicos de los compositores mencionados con anterioridad, Garwood Whaley (G. Whaley, 1974. Pags 3-70) y Arthur Wesberg (A. Wesberg, 1993. Pags 3-69) definen los siguientes procedimientos de elaboración rítmica.

- a) *Cambio de métrica*. La razón primaria de un cambio de métrica es la colocación de acentos. Normalmente esta técnica se usa cuando hay una sucesión de denominadores métricos (sólo octavos, o sólo dieciseisavos, por ejemplo), de manera que este denominador se perciba todo el tiempo, y el cambio de métrica juegue con la colocación de acentos sobre la serie de valores.

MAX-RHYTHMATIST

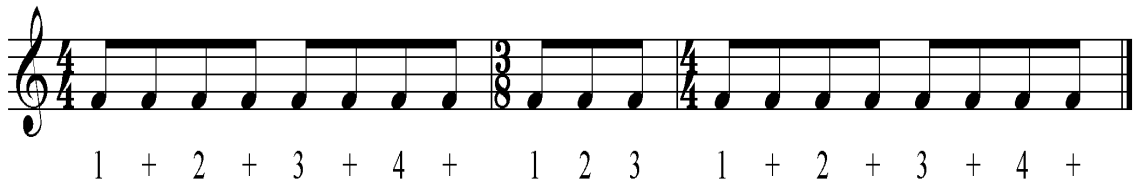


Fig. 7. Cambio de métrica

- b) *Compases compuestos*. Estos generalmente se basan en un numerador primo (5, 7, 11, etc.), y debido a esta propiedad la colocación de la acentuación es variable e irregular. Por ejemplo, un compás de 5/8, normalmente puede acentuarse como 2+3, 3+2, sin embargo, nada impide que pueda acentuarse como 1+4, 4+1, o simplemente 5. De aquí deriva la siguiente técnica.

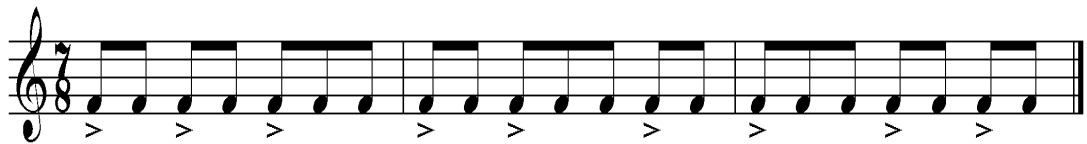


Fig 8. Compás compuesto

- c) *Reacomodo de acentos*. Si bien esto se puede conseguir mediante el cambio de métrica, o jugando con las posibilidades de los ritmos compuestos, es posible realizarlo también dentro de un mismo compás simple. Por ejemplo, un compás de 4/4 subdividido en 8 normalmente se agruparía 2+2+2+2, pero se puede reacomodar su acentuación como 2+3+3, 3+2+3, o 3+3+2.

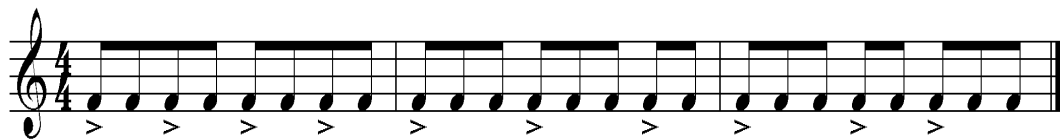


Fig. 9 Reacomodo de acentos.

- d) *Subdivisiones métricas distintas a las potencias de dos*. Normalmente, a medida que vamos a niveles arquitectónicos inferiores, el número de valores rítmicos que pueden

### MAX-RHYTHMATIST

colocarse en una unidad de tiempo aumenta en potencias de 2. Ejemplo, en un compás de 2/4, en un nivel primario hay una mitad, un nivel de subdivisión abajo hay 2 cuartos, después 4 octavos, luego 8 dieciseisavos, etc. El uso de otros divisores nos da los tresillos, quintillos, sietillos, etc., los cuales pueden ser utilizados en cualquier nivel de subdivisión, y que tienden a producir un efecto de cambio de velocidad en el pulso.

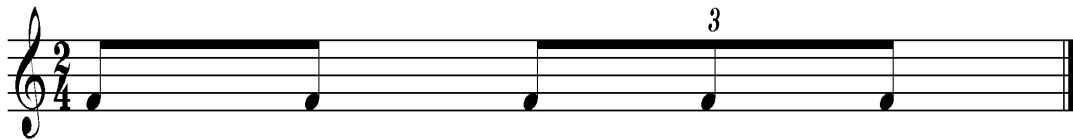


Fig. 10. Subdivisiones métricas distintas a las potencias de dos.

- e) *Agrupamiento de niveles con divisores distintos*. El combinar un quintillo de octavos, con un tresillo de cuartos y un cuatrillo de dieciseisavos, ya sea en voces musicales distintas o incluso en una misma voz, es una técnica muy recurrente dentro de la nueva complejidad de la música contemporánea. Es importante decir que estos agrupamientos, debido a su extrema dificultad, implican un trabajo muy duro del intérprete para “descifrar” la manera de abordarlo, dando como resultado, casi siempre, una interpretación que tiende a parecerse a lo escrito, pero nunca llega a ser exacta.

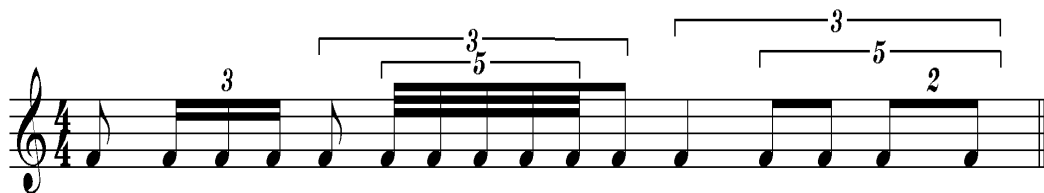
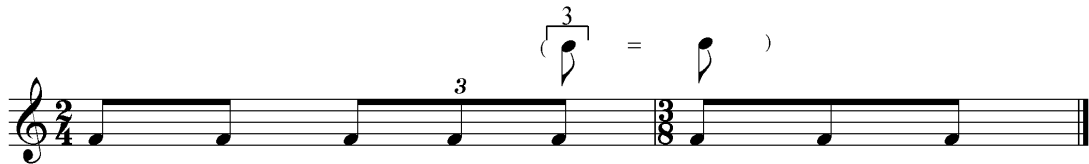


Fig. 11. Agrupamiento de niveles con divisores distintos

- f) *Modulación métrica*. Ésta técnica es usada para alterar el pulso básico. El valor de las notas no siempre se mantiene constante con un cambio de métrica. En el punto en donde hay un cambio en el pulso (modulación), dos valores rítmicos lo indican sobre la partitura,

**MAX-RHYTHMATIST**

el primero indica el pulso precedente, y el segundo el nuevo pulso. Por ejemplo, si en un compás de 2/4 se presenta un tresillo de octavos al final, y hay un cambio de métrica a 3/8, la modulación métrica puede darse para que el octavo del tresillo en el primer compás tenga el mismo pulso que el octavo del compás de 3/8, luego entonces el pulso aumenta de velocidad.



*Fig. 12. Modulación métrica*

- g) *Aceleración y/o desaceleración del pulso.* Normalmente de carácter libre, dado que se deja al intérprete la libertad de hacer la desaceleración o la aceleración a su gusto. Esto se manifiesta también en la notación, que no establece valores rítmicos precisos, tan sólo se limita a indicar la velocidad de inicio, y la de llegada. A diferencia de la modulación rítmica, aquí no hay un cambio de velocidad súbito, sino gradual. Estos cambios graduales pueden darse en distintos niveles arquitectónicos, llegándose a dar incluso el caso de que un nivel inferior puede estar cambiando, mientras que los niveles superiores se mantienen constantes.



*Fig. 13. Aceleración y desaceleración*

MAX-RHYTHMATIST

- h) *Apoyaturas*. Generalmente usadas como adornos, estos valores pueden darse sobre el pulso o antes del pulso, y su valor rítmico específico puede variar dependiente de qué tan rápido o lento lo ejecute el músico.



Fig. 14. *Apoyaturas*

- i) *Cambio de tempo*. Cambio súbito en la velocidad del pulso, normalmente indicado como un valor de metrónomo en la partitura.

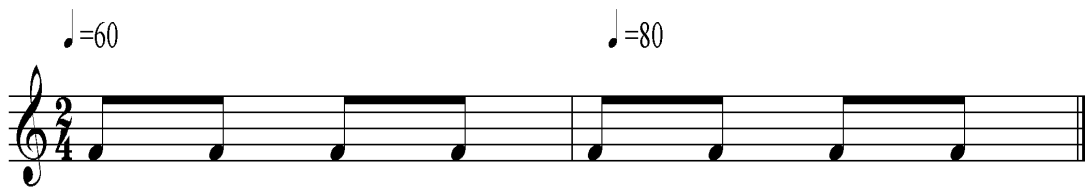


Fig. 15. *Cambio de tempo*

- j) *Adición o sustracción de valores rítmicos*. Técnica que resulta eficaz sobre todo cuando se tienen patrones rítmicos que agrupan valores de distintos niveles arquitectónicos, dando como resultado un cambio en la duración del patrón rítmico. A diferencia del cambio de compás, que busca sólo el reacomodo de acentos sin hacer perder la percepción de un pulso, la adición o sustracción de valores rítmicos hace que la idea de pulso llegue incluso a ser ambigua.



Fig. 16 *Adición y sustracción de valores rítmicos*.

### MAX-RHYTHMATIST

- k) *Serie de valores rítmicos*. Al igual que las series de notas, las series de valores consisten en un conjunto de valores rítmicos provenientes de distintos niveles arquitectónicos, ordenados al gusto del compositor, y que pueden ser utilizados siguiendo el orden establecido, retrogradarlos, segmentarlos para usarlos por grupos, etc.

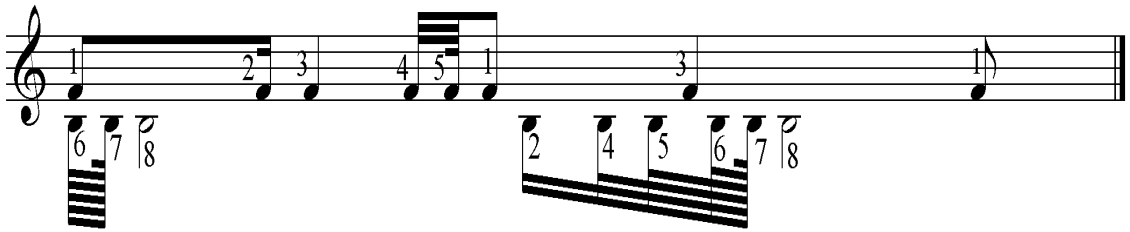


Fig. 17. Serie de 8 valores rítmicos

## **1.3 INVESTIGACIONES EN EL ÁMBITO PSICOACÚSTICO**

Definiciones de ritmo desde un punto de vista acústico difieren entre autores. Clarke define al ritmo como “una organización de grupos de duraciones relativas sin consideraciones de periodicidad” (E. F. Clarke, 1985. Pág. 209). La definición de Dowling es “un patrón temporal de relaciones entre duraciones y acentuaciones” (Dowling y Harwood, 1986. Pág. 29). Parncutt dice que ritmo musical es una secuencia acústica que evoca una sensación de pulso (Parncutt, 1994. Pág. 127).

Estas definiciones son algunos ejemplos de cómo la investigación en el terreno de la percepción rítmica ha estado marcada por la complejidad y por teorías contrastantes; lo cual a su vez refleja la complejidad de los procesos de percepción temporal. La sensibilidad en cierto contexto, las interrelaciones entre melodía y temporalidad, y la ausencia de características perceptivas invariables son factores que contribuyen a la complejidad del análisis rítmico (Handel, 1989. Pág. 597).

Algo fundamental para cualquier concepción de ritmo es la percepción de temporalidad del escucha. La percepción del tiempo musical está determinada por la capacidad del escucha de percibir eventos audibles. Podemos dividir las percepciones audibles entre estímulos de sonidos breves, como serían

## MAX-RHYTHMATIST

los impulsos o “clicks”, y los tonos sostenidos (H. Woodrow, 1951. Págs. 1224-1236). La concepción de tiempo acciona dos comportamientos significativos: la sincronización con un ritmo percibido, y la noción de presente.

### **1.3.1 Sincronización**

Estudios acerca del desarrollo biológico de un ser humano a lo largo de su vida indican que la percepción de sincronía es potencialmente biológica; niños de cuatro meses de edad prefieren la sincronización entre un estímulo visual y un sonido relacionado con dicho estímulo con relación a estímulos no sincronizados. Canciones interpretadas espontáneamente por niños de temprana edad tienen un patrón de pulso regular dentro de sus frases (G. Widmer, 1996. Págs. 179-205).

Como una regla de percepción, el escucha reacciona siguiendo un estímulo, luego entonces la sincronización produce respuestas al mismo tiempo que el estímulo. La regularidad parece ser menos importante para la sincronización que la habilidad del escucha para anticipar y predecir, de manera que incluso la aceleración o desaceleración de los ritmos puede también ser sincronizada. Para patrones regulares, la sincronización puede ser establecida muy rápidamente desde el tercer sonido, y así mismo la sincronización de patrones repetitivos se consigue a partir del tercer patrón (P. Fraisse, 1982. Págs. 149-180). De modo que un modelo de tiempo musical debe de ser capaz de generar sincronizaciones después de dos o tres pulsos y por tanto moldear las expectativas del escucha.

### **1.3.2 El presente subjetivo**

El presente subjetivo es un termino caracterizado por “la sensación de el ahora” (U. Seifert, 1993. Pág. 288), un lapso de atención, una ventana en el tiempo, o el intervalo de percepción gestalt temporal (P. Fraisse, 1982. Pág: 149). Se considera que es un intervalo donde todas las percepciones y sensaciones están simultáneamente disponibles para ser procesadas como una unidad (R. Parncutt, 1994. Pág. 127). El presente subjetivo ha sido considerado por Dowling y Harwood como la

### MAX-RHYTHMATIST

sensación percibida de una memoria acústica sensorial o “ecóica” (en comparación con el término memoria icónica visual), es decir un breve almacenamiento de información auditiva no procesada. (Dowling y Harwood, 1986. Pág. 29)

#### ***A) Lapsos de tiempo (time-spans) del presente subjetivo***

Las evidencias presentadas por Cowan sugieren que el almacenamiento de percepciones auditivas se integra de manera continua y no discreta (N. Cowan, 1984. Págs. 341-370). Seifert sugiere que para que una frase musical sea percibida como una sola estructura, la duración total debe estar comprendida dentro del lapso de tiempo del presente subjetivo (U. Seifert et al, 1995. Págs: 164-195), citando el hecho de que los ciclos rítmicos en la música árabe y africana oscilan entre 2800 y 3000 ms. Woodrow reportó una diferencia apenas notable (JND, por sus siglas en inglés) para la discriminación de un solo intervalo dado por pulsos audibles de corta duración en el rango de 200 a 1500 ms (H. Woodrow, 1951. Págs. 1224-1236).

Tal como lo notan Dowling y Harwood, esta JND es significativamente más alta cuando se reproducen intervalos aislados que cuando se discriminan intervalos en el contexto de un patrón rítmico. Ambos reportan que el tamaño de la ventana que limita la percepción del presente oscila entre los 2 segundos y los 5 segundos. Un máximo de 10 a 12 segundos de presente sólo se consigue agrupando largas secuencias rítmicas en subsecuencias.

De acuerdo al modelo propuesto por Parncutt, la duración máxima de la memoria ecóica es de 500 ms, y la del presente subjetivo es de 4 segundos. De acuerdo a sus investigaciones, la percepción de un pulso está limitada por el lapso de tiempo del presente subjetivo (R. Parncutt, 1994. Págs. 127-138).

#### ***B) Jerarquías de los quanta temporales.***

Seifert propone la teoría de los quanta temporales para describir el comportamiento de la percepción



### MAX-RHYTHMATIST

de los pulsos (R. Parncutt, 1994. Págs. 127-138). Dicha teoría supone una categorización de experiencias temporales elementales con respecto a la percepción rítmica como un medio para describir los niveles de dichos cuanta:

- a) Los pulsos entre los 0 y los 2-5 ms son percibidos como simultáneos, indistinguibles, incluso con diferente sonoridad, pero con misma duración, es decir, se perciben como un solo evento.
- b) Los pulsos entre 2-5 y 40 ms pueden ser distinguidos, pero ninguna relación de orden puede ser indicada.
- c) Los pulsos entre los 30 y los 50 ms pueden producir una relación de orden, es decir, el orden en que ocurren dichos eventos es distinguible.

#### **1.3.3 Visión científica de los principales atributos del ritmo**

Las dimensiones del ritmo musical son revisadas en este punto para caracterizar procesos musicales que cualquier modelo computacional pueda interpretar. La información rítmica es más importante para la cognición musical que la altura. Investigaciones tempranas mostraron que melodías familiares podían ser reconocidas al presentarse únicamente su patrón rítmico (Dowling y Harwood, 1986. Págs. 30-73). De ahí el interés científico que ha despertado su estudio.

##### ***A) Acentuación***

La acentuación produce diferencias entre notas musicales, distinguiendo eventos sonoros acentuados de eventos temporales adyacentes. Tanguiane menciona que un ritmo sólo ocurre cuando una secuencia periódica puede ser segmentada en grupos (Tanguiane, 2007. Pág. 89). Fraisse dice que la base del ritmo es el ordenamiento en el tiempo de relaciones temporales entre eventos, y no algo que surja de patrones de pulsos acentuados. Evidencias sobre esta última opinión se encuentran en las variaciones entre escuchas al identificar cuáles pulsos son acentuados suficientemente para indicar

### *MAX-RHYTHMATIST*

un tiempo fuerte<sup>2</sup> (P. Fraisse, 1982. Págs. 149-180).

Lerdahl y Jackendoff han distinguido acentos métricos, fenomenológicos, y estructurales, de acuerdo con el efecto sobre los grupos que conforman al ritmo (Lerdahl y Jackendoff, 1983. Pág 17). Los acentos métricos ocurren cuando el pulso enfatizado se relaciona con el patrón métrico (la repetición de acentos en los pulsos de manera regular). Los acentos fenomenológicos se considera que existen en la superficie musical, enfatizando un momento específico, permitiendo que las síncopas (acentos fuera de fase con respecto al metro) sean percibidas. Los acentos estructurales son definidos como “un acento causado por puntos melódico-armónicos de atracción en una frase o sección, especialmente provocado por la cadencia, que es el objetivo de un movimiento tonal” (Lerdahl y Jackendoff, 1983. Pág 17). Un acento estructural es entonces percibido en relación al desarrollo de la frases y de la estructura de la música, que son más largos que un compás.

La acentuación se consigue por medio de diferencias objetivas entre eventos sonoros, permitiendo con ello la agrupación de sonidos en el tiempo. Como ahora describiremos, un escucha también acentúa de manera subjetiva sonidos que son de hecho isócronos, aún cuando estos eventos carecen de acentos objetivos.

### *Metrización subjetiva*

La acentuación subjetiva es un término histórico usado para describir el agrupamiento de trenes de pulso isócronos en grupos de dos, tres o cuatro. El primer elemento del grupo es percibido como acentuado, y el intervalo entre el último elemento y el primero del siguiente grupo es percibido como alargado (P. Fraisse, 1982. Págs- 149-180). En términos modernos, el término metrización subjetiva es ahora más apropiado (E. W. Large, 1994. Págs. 177-208). Este fenómeno evoca una sensación de pulso cuyo período es más largo que el del estímulo (R. Parncutt, 1994. Págs. 127-138).

---

<sup>2</sup> El tiempo fuerte es el primer pulso de un patrón métrico que usualmente es percibido como acentuado.

### **MAX-RHYTHMATIST**

La longitud relativa de un intervalo de silencio que sigue a un tono en una secuencia equitonal es un factor determinante para percibir un acento sobre ese tono. Povel y Okkerman (Povel y Okkerman, 1981. Págs. 565-572) variaron tanto el primero como el segundo IOI<sup>3</sup> entre pares de tonos en secuencias isócronas y equitonalas. Con esto pretendían determinar los intervalos de tiempo en que los acentos son percibidos sobre el primero o el segundo tono como resultado de la metrización subjetiva. Cuando la diferencia entre los intervalos es corta, el acento fue percibido sobre el primer tono del par, cuando el intervalo se incrementaba por encima de 220 ms, el acento se percibía con mayor frecuencia sobre el segundo tono del par.

El fenómeno de metrización subjetiva demuestra que el proceso de agrupamiento de elementos temporales en estructuras más largas ocurre aún cuando no esté apoyada por diferencias objetivas. Esto sugiere que los intervalos temporales por sí mismos son responsables de la acentuación y la determinación de las estructuras rítmicas.

### ***Acentuación y metrización objetivas.***

A continuación se muestra un resumen de las diferencias objetivas producidas por un ejecutante humano para inducir el agrupamiento (Povel y Okkerman, 1985. Págs. 441-440):

- a) Alargamiento de un IOI entre dos eventos
- b) Incremento en la intensidad
- c) Variación de la articulación (legato/staccato)
- d) Cambio en la altura o en los extremos de trayectorias melódicas
- e) Inicios y términos de progresiones armónicas
- f) Diferencia o cambio de timbre o instrumentación

---

<sup>3</sup> IOI: Inter-Onset Interval. Espacio de tiempo considerado desde el inicio de un sonido hasta su total decaimiento, seguido de un período de silencio, siendo este silencio considerado como articulación entre el primer sonido y el subsecuente.

### MAX-RHYTHMATIST

- g) Sincronía entre voces con los mismos instrumentos
- h) Sincronía entre voces con instrumentos distintos
- i) Densidad de eventos en tiempos muy breves (time-spans), por ejemplo, trémolos, apoyaturas, etc.
- j) Alargamiento al final de una frase o sección, como en los efectos de rubato.

### ***B) Agrupamiento***

Una regla general a toda percepción es que los elementos tienden a ser colocados en grupos de igual tamaño, siendo estos grupos de dos o más elementos (S. Handel, 1989. Pág. 597). El agrupamiento en el ritmo es la asignación de una estructura temporal a un conjunto de elementos audibles; se considera que el agrupamiento es responsable de los conceptos de frases o motivos. El agrupamiento aparece a temprana edad, y parece ser un comportamiento espontáneo (P. Fraisse, 1982. Págs. 149-180). A consecuencia del efecto de metrización subjetiva, el agrupamiento ocurrirá aún cuando no haya acentos objetivos.

Formar grupos de 2, 3 y 4 elementos es significativamente más fácil que hacerlo en grupos de 5 o 7, lo cual sugiere que la economía de atención y representación, y los límites de la memoria auditiva, más que los acentos por sí solos, contribuyen a la organización en grupos (S. Handel, 1989. Pág-597). Estos ritmos más largos tienden a ser agrupados como subgrupos de 2 y 3, lo cual a dado a entender que las jerarquías de agrupamiento son espontáneamente organizadas (Dowling y Harwood, 1986. Págs. 30-73). Fraisse ha estimado un valor máximo de 25 sonidos sucesivos que pueden ser percibidos como una unidad (P. Fraisse, 1982. 149-180)

### ***C) Metro y pulso***

El metro es la ocurrencia de una acentuación objetiva o subjetiva, mientras que impulsos isócronos sólo producen una sensación de periodicidad o pulso. Una sensación de pulso es necesaria para el

### MAX-RHYTHMATIST

metro o el ritmo, pero requiere a su vez una sensación de diferenciación de los impulsos para que estas estructuras superiores hagan su aparición. La alternancia regular de impulsos perceptivamente débiles y fuertes produce una sensación de metro (Palmer y Krumhansl, 1990. Págs. 505-518), que Dowling y Harwood han descrito como “el nivel más básico de organización rítmica” (Dowling y Harwood, 1986. Págs. 30-73).

El metro es un medio para la interpretación rítmica, al medir el tiempo y anticipar eventos por ocurrir, y es un auxiliar en la organización de puntos en el tiempo con características equivalentes. Estos puntos tienen similitud con la organización de alturas por octavas, el primer pulso en un compás (tiempo fuerte) asume una equivalencia funcional a lo largo del tiempo, ayudando a establecer juicios de similitud y memoria (Palmer y Krumhansl, 1990. Págs. 505-518).

Sin embargo, como lo señala Frigyesi (J. Frigyesi, 1993), el ritmo descrito como “libre, “carente de un pulso claro”, o “fluido”, es de gran importancia en la música no occidental. En la música de la India, el concepto occidental de pulso es equivalente a matra -la unidad de tiempo básica- que es el tiempo mínimo para una sílaba hablada (Sorrell y Narayan, 1980. Pág. 190). Una cierta tala –un ciclo- es la estructura de agrupamiento impuesta sobre los pulsos y es típicamente menor que 20 pulsos de duración, pero puede llegar a ser pensada de manera teórica en valores más largos. La tala es subdividida, frecuentemente en subgrupos desiguales e impares.

La ocurrencia frecuente de silencios en pulsos temporales implica que el ritmo de la tala puede llegar a ser más implícito que explícito y “el interés estético yace en el grado en el que los músicos y su audiencia pueden seguir la tala de manera precisa a lo largo de pasajes de elaborada improvisación rítmica” (Dowling y Harwood, 1986. Págs. 30-73). Una sólida representación mental es requerida para proveer sentido musical.

Luego entonces, la asignación de fuerza a un pulso, el agrupamiento de acuerdo con esta fuerza, y el

### MAX-RHYTHMATIST

efecto del metro sobre la estructura de percepción temporal, deben ser considerados por una teoría del ritmo cuyo objetivo sea su aplicación informática.

#### *Interacción entre metro y agrupamiento*

Mientras que el concepto de agrupamiento auditivo tiene sus orígenes en analogías psicológicas visuales, el metro es un concepto propio de la teoría musical, teniendo una notación característica, como por ejemplo: 4/4. El metro representado de un ritmo no es necesariamente el que el escucha percibe.

El metro y las estructuras de agrupamiento interactúan, pero no siempre se mantienen en fase; de manera que notas importantes dentro de un grupo pueden caer en tiempos débiles, así que el juego entre metro y agrupamiento determina en última instancia el ritmo percibido (L. B. Meyer, 1956. Págs. 103, 307). El metro afectará la percepción de la secuencia entera. Povel demostró que al tocar 12 secuencias de elementos con diferentes metros se hacía imposible identificar secuencias idénticas (Povel y Okkerman, 1985. Págs. 441-440). Handel demostró que ritmos difíciles que no corresponden a la acentuación natural; es decir, la métrica típica, eran oídos en términos de agrupamiento de elementos, en lugar de un metro con variaciones agógicas (S. Handel, 1989. Pág. 597).

#### *Tactus*

Tactus es un término renacentista adoptado por Lerdahl y Jackendoff para describir el tempo y el nivel jerárquico por los cuales el escucha puede acompañar a un ritmo con el pie. Se puede definir al tactus como el “nivel de la estructura métrica donde los pulsos se suceden en un rango moderado”. De acuerdo a esta definición, los rangos de tiempo del tactus se centran alrededor de los 600 ms (Lerdahl y Jackendoff, 1983. Pág. 17)

## *MAX-RHYTHMATIST*

### *El papel de la repetición en la formación del metro*

La repetición es un auxiliar importante para agrupar; y de la misma manera, la repetición de secuencias es una importante guía para la formación del metro. Longuet-Higgins y Lee (Lee y Longuet-Higgins, 1985. Págs. 53-69) le han dado gran importancia a la repetición de patrones rítmicos y melódicos, aún sujetos a transposición temporal o de alturas en la formación del metro.

#### ***D) Polirritmos***

En ejemplos musicales de Indonesia, la India y Africa (Dowling y Harwood, 1986. Págs. 30-73), y ejemplos de Jazz (C. Folio, 1995. Pág. 103-134), la estructura rítmica es comúnmente organizada en capas que involucran patrones que no están subdivididos por un mismo pulso. Handel define a un polirritmo como “la presentación simultánea de dos patrones isócronos que no comparten un mismo denominador” (S. Handel, 1989. Pág. 597).

#### ***E) Tempo***

El tempo es un término musical que describe el rango de presentación de los pulsos. Sin embargo, sin importar la cultura a la que nos refiramos, es posible entender al tempo como un rango de presentación de un nivel de pulsos perceptibles y significativos.

### *Interacción entre tempo y estructura*

Las investigaciones de Clarke con referencia al tempo en la ejecución, han mostrado una interrelación entre la estructura y el tempo (E. F. Clarke, 1987. Págs. 209-237). La música tiende a ser agrupada en menos unidades cuando el tempo es alto. Clarke dice que esta dependencia entre tempo y agrupación se debe a la limitación del presente subjetivo. Este límite provoca la segmentación por el ejecutante y por lo tanto provoca la subdivisión de grupos más largos, en puntos acordes con las propiedades estructurales de la obra.

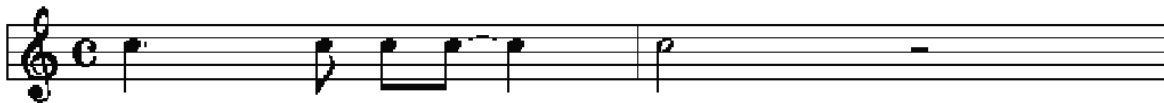
### 1.3.4 Modelos teóricos de representación del ritmo

Aquí se hace una revisión a las distintas aproximaciones al problema del ritmo desde el punto de vista de su representación dentro de un sistema o modelo, y a su vez, se revisan las diversas soluciones que dichos enfoques plantean para el análisis del ritmo.

#### *A) Modelo de estratos rítmicos*

Yeston propone una concepción y representación del ritmo como una jerarquía de estratos (M. Yeston, 1976, Pág. 155), cada uno de ellos incrementando lapsos de tiempo con proporcionales disminuciones en los rangos de pulsos. Yeston considera su modelo como un análisis de la estructura musical, y no como un modelo de análisis de la percepción directa. El metro es considerado por él como un fenómeno resultado de los acentos creados por la interacción entre los niveles estratificados de una manera jerárquica, en este aspecto coincide con las ideas propuestas por Cooper y Meyer. Además, Yeston describe cinco criterios mediante los cuales aislar el ritmo en sub-patronos. Estos sub-patronos son muy importantes para encontrar el estrato rítmico en una pieza. Los criterios son:

- *Punto de ataque.* Derivado de la partitura, se refiere a la distancia (duración) entre ataques, en términos de la unidad local más pequeña. Por ejemplo,



tiene la secuencia 3 1 1 3 4, tomando al octavo como la unidad.

- *Timbre.* Los sub-patronos rítmicos se distinguen por cambios en la instrumentación o por desplazamientos tímbricos, como en un desplazamiento de registro abrupto de un solo instrumento.
- *Dinámica.* Ya sean los acentos notados, o el cambio equivalente a nivel dinámico, puede



### MAX-RHYTHMATIST

determinar un sub-patrón rítmico sin interpretar.

- *Densidad*. Se refiere a los cambios en ya sea la “cantidad” de sonido o el número de voces simultáneas en la textura general.
- *Patrón recursivo*. Búsqueda de los patrones de unidades repetidas, ya sea en figuras de duración repetidas, o contornos de altura, o combinaciones del mismo. Inicialmente se observa la música como una “estructura sin interpretar” porque aún no se ha establecido ningún agrupamiento interno.

Los sub-patrones rítmicos identificados por los cinco criterios son utilizados para crear el primer estrato, desde el cual se pueda abstraer los eventos estructurales. En cada nivel, los eventos estructurales están dados por las duraciones combinadas de los eventos desde los cuales son abstraídos. En cada nivel, la nueva representación es considerada como una nueva estructura sin interpretar, y se repite el proceso.

Yeston ve la interacción de estos niveles como el factor determinante en la determinación del metro.

De hecho dice: “el metro nunca aparece sobre cualquier estrato simple, sino que surge de la interacción de dos estratos, uno de los cuales debería ser siempre de un nivel intermedio”.

La idea de los sub-patrones y sus criterios para determinarlos, muy en especial el referente a los puntos de ataque, fueron importantes para decidir que la aplicación informática generaría patrones rítmicos a partir de pequeños elementos, que a su vez pudieran ser contruidos mediante la suma de eventos rítmicos con una duración específica. Sin embargo, el modelo de Yeston no proporciona una manera sistemática de generar dichos eventos, de forma que no sea necesario generar uno por uno.

#### ***B) Teoría jerárquica del metro***

Las gramáticas generativas han sido utilizadas tanto en composición algorítmica, tema que se tratará posteriormente en esta tesis, como en análisis musical. El modelo lingüístico básico fue desarrollado

### MAX-RHYTHMATIST

por Noam Chomsky en 1957 (Chomsky, 2004. Págs. 33-41), es el punto inicial de aplicación de este y otros principios de generación en aspectos musicales. En lingüística, el término gramática generativa se refiere a un conjunto de marcos teóricos para el estudio de la sintaxis de las lenguas. Lerdahl y Jackendoff han propuesto su “Teoría Generativa de la Música Tonal” (GTTM, por sus siglas en inglés), aplicando una teoría de gramáticas generativas como un mecanismo descriptivo de las relaciones musicales (F. Lerdahl & R. Jackendoff, 1983. Pág. 368). Ellos proponen tanto una jerarquía métrica como una jerarquía de agrupamientos. De acuerdo con ellos, la jerarquía métrica es responsable de la asignación de una importancia a los pulsos en un lapso de tiempo corto. En lapsos de tiempo largos, es la jerarquía de agrupamientos la responsable de la organización en frases, temas y secciones. La percepción del metro y el agrupamiento ocurren simultáneamente, y de acuerdo con Handel, la separación de ambos parámetros es imposible (S. Handel, 1989. Pág. 597).

Dentro del discurso musical distinguimos sonidos que se encuentran perceptualmente acentuados, es decir, que sobresalen y se distinguen por sobre otros. En estas acentuaciones confluyen generalmente más de un factor que contribuyen a la delineación del acento como tal.

Según Lerdahl y Jackendoff, estos factores producen tres tipos de acentos: los *fenoménicos* describen las acentuaciones que ocurren evento a evento, lo que se acentúa en el discurrir de la música, los *estructurales* describen las acentuaciones que se producen en la estructura armónica o tonal, y los *métricos*, entendidos como la coincidencia de distintos niveles de pulsaciones de la estructura métrica. Estos acentos generan relaciones de convergencia y divergencia, configurándose así agrupamientos o grupos de sonidos que colaboran en la comprensión del ritmo (F. Lerdahl & R. Jackendoff, 1983. Pág. 368).

#### **C) Modelo basado en series**

Hablando de modelos aritméticos que sirvan para el análisis o la estructuración rítmica, llama la

### MAX-RHYTHMATIST

atención Augusto Novaro, que en su libro "El sistema natural de la música" analiza la generación de escalas a partir de series numéricas, y en uno de sus apartados habla del ritmo; pero sólo de manera muy tangencial, ya que no es el tema principal del libro (A. Novaro, 1953. Págs. 27-55). Sin embargo, las series numéricas aplicadas a la generación de escalas de sonidos parecen ser, en primera instancia, adaptables al ritmo, de ahí que valga la pena detenerse un poco a analizarlas.

Novaro introduce su investigación aclarando que la mayoría de los términos que utilizará son de su invención, dado que la mayoría de los estudios realizados sobre microtonalidad carecían de una terminología clara en aquel tiempo. Denota como "Escalas Geométricas" a aquellas derivadas de la combinación de fracciones que dividen al espacio comprendido entre 1 y 2 (que representan al sonido inicial y su octava). Al multiplicar esta serie de números por una frecuencia en Hertz cualquiera, el resultado era una escala no temperada, es decir, una escala en la que los intervalos perceptibles entre cada elemento consecutivo de la escala no eran iguales. Novaro distingue tres tipos de escala geométrica:

1. Escalas fundamentales, en donde el intervalo entre 1 y 2 es dividido por fracciones con un denominador común. Por ejemplo: 1,  $\frac{3}{2}$ , 2 ó 1,  $\frac{4}{3}$ ,  $\frac{5}{3}$ , 2
2. Escalas recíprocas, que son inversiones de las escalas fundamentales. Por ejemplo: 1,  $\frac{4}{3}$ , 2, que es recíproca de 1,  $\frac{3}{2}$ , 2, dado que la inversión de  $\frac{3}{2}$  es  $\frac{2}{3}$ , multiplicado por 2, para que este número esté en el intervalo comprendido entre 1 y 2, nos da  $\frac{4}{3}$ ; de la misma manera para la escala fundamental 1,  $\frac{4}{3}$ ,  $\frac{5}{3}$ , 2, su recíproca será 1,  $\frac{3}{2}$ ,  $\frac{6}{5}$ , 2.
3. Escalas complejas, que son combinaciones de escalas fundamentales y recíprocas.

Tras una exhaustiva explicación de las series aritméticas que pueden generar escalas no temperadas, la segunda parte de su libro ahonda en las escalas temperadas, generadas a partir de raíces N de 2, dando por resultado series numéricas exponenciales, y describe los diferentes instrumentos y

### MAX-RHYTHMATIST

experimentos generados a partir de dichas escalas, de tal forma que cuando toca el tema del ritmo, las escalas numéricas aritméticas (geométricas, de acuerdo a su terminología) pasaron a ser sólo una referencia inicial, y en su afán por seguir usando raíces, trata de aplicarlas al tema del ritmo. Al hacer esto, Novaro comete un error, ya que la percepción de intervalo es de carácter exponencial, es decir, la cantidad de Hertz que comprende una octava, depende de qué tan grave o agudo se sitúa este intervalo, ya que responde a potencias de 2, mientras que en el ritmo la percepción de tiempo y duración es lineal, no importa en qué momento se escuche un tren de pulsos, la duración entre cada uno siempre será la misma, más allá de los problemas de metrización subjetiva que ya se mencionaron.

Por lo tanto, las escalas aritméticas de Novaro pueden no haber sido útiles en sus experimentos finales con escalas sonoras, pero pueden ser el punto de partida para la generación de los patrones rítmicos en una aplicación informática. Es decir, se puede emplear un proceso similar al descrito para generar escalas de sonidos, pero obviamente adaptado al ritmo, se tomaría al espacio entre 0 y 1 como intervalo para generar series numéricas y luego se multiplicarían por una duración, y se podrían generar patrones rítmicos mediante la combinación de series numéricas generadas por distintos medios.

## **1.4 COMPOSICIÓN ALGORÍTMICA E INSTRUMENTOS**

### **VIRTUALES**

Dentro del ámbito informático relacionado con los problemas musicales, básicamente se encontraron dos tipos de enfoque al momento de abordar el diseño de implementaciones informáticas, la Composición Algorítmica o Composición asistida por Computadora y los Instrumentos Virtuales.

#### **1.4.1 Composición asistida por computadora**

El objetivo de la Composición Asistida por Computadora (CAC) es proveer los medios que ayuden en el proceso creativo de un compositor. CAC está frecuentemente basada en un modelo matemático, ejemplos de esta estrategia incluyen la aplicación de modelos estocásticos (Por ejemplo, Xenakis, 1992. Págs. 131-154), o de modelos matemáticos de caos para generar una partitura (Por ejemplo, Supper, 2001. Págs: 48-53). Otras estrategias transforman datos para colocarlos sobre una partitura, que es obtenida mediante el análisis de los cambios espectrales de un sonido grabado. Algunas estrategias de CAC implementan estrategias de composición ya existentes que fueron desarrolladas fuera del ámbito de la CAC, por ejemplo la composición serial (Laske, 1981. Págs: 84-93.). La mayoría de las estrategias de CAC trabajan en tiempo diferido, sin embargo, en años recientes ha habido un gran número de compositores que comienzan a experimentar el uso de estas técnicas en el campo de la música interactiva (Rowe, 2001. Págs. 277-310). Papadopoulus y Wiggins ofrecen una visión sistemática de la CAC enfocándose en sistemas basados en técnicas provenientes de la Inteligencia Artificial (Papadopoulus y Wiggins, 1999. Págs. 1-8).

#### ***A) El ritmo en la composición asistida por computadora***

En lo referente a software musical, Nierhaus (Nierhaus, 2009. Págs. 157-184) menciona que algunas aplicaciones informáticas musicales hacen uso de algoritmos genéticos, que consisten en transformar

### MAX-RHYTHMATIST

una idea ya existente para generar una idea nueva. Estos algoritmos hacen uso de bloques virtuales que interactúan entre sí, cada bloque tiene su propio patrón musical, y es capaz de enviar dicho patrón a otros bloques, así como recibir patrones de otros; cuando un patrón es pasado a otro, sirve como patrón de referencia para el bloque que lo recibe, y este último genera un nuevo patrón musical basado en la comparación de ambos patrones (el recibido y el que tenía ya alojado). Entre los sistemas enfocados al ritmo, Nierhaus menciona que Damon Horowitz desarrolló un sistema informático en el cual patrones rítmicos son generados por medio de un cromosoma (término usado para definir una idea generada mediante algoritmos genéticos) que representa un compás cuyos valores rítmicos pueden incluir tanto notas como silencios, con valores rítmicos que pueden variar entre valores de 1/16 de tresillo y valores 1/4. Siguiendo la línea de los algoritmos genéticos enfocados a la generación rítmica, Nierhaus también menciona los trabajos de Alejandro Pazos, Santos del Riego, Julián Dorad y J.J. Romero-Carnalda, quienes además hacen uso de redes neurales artificiales.<sup>4</sup>

### ***B) Sistemas informáticos***

Muchos sistemas informáticos han sido desarrollados como herramientas para compositores que quieran usar estrategias de CAC. Con frecuencia, los compositores desarrollan sus propias herramientas según sus necesidades, pero también hay sistemas que son usados por cientos de compositores. Roads hace una lista histórica de muchos de estos sistemas (Roads, 1996. Págs.819-852). A continuación se mencionan algunos de los que todavía se encuentran en uso.

### ***Open Music***<sup>5</sup>

OpenMusic es un lenguaje de programación gráfico enfocado a CAC con muchas extensiones a

---

<sup>4</sup> Una red neural artificial (RNA) es un modelo de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales. El objetivo es conseguir que las máquinas den respuestas similares a las que es capaz de dar el cerebro.

<sup>5</sup> <http://repmus.ircam.fr/openmusic/home> "Sitio oficial de OpenMusic"

## MAX-RHYTHMATIST

manera de librerías para diversas estrategias de composición. En la página oficial del programa se menciona que está basado en CommonLisp. En lo que respecta al manejo del ritmo, la función `omquantify` convierte una lista de milisegundos en un árbol rítmico, en donde los números enteros representan duraciones y los negativos silencios. En la figura se muestra un ejemplo del uso de esta función y su visualización en OpenMusic.

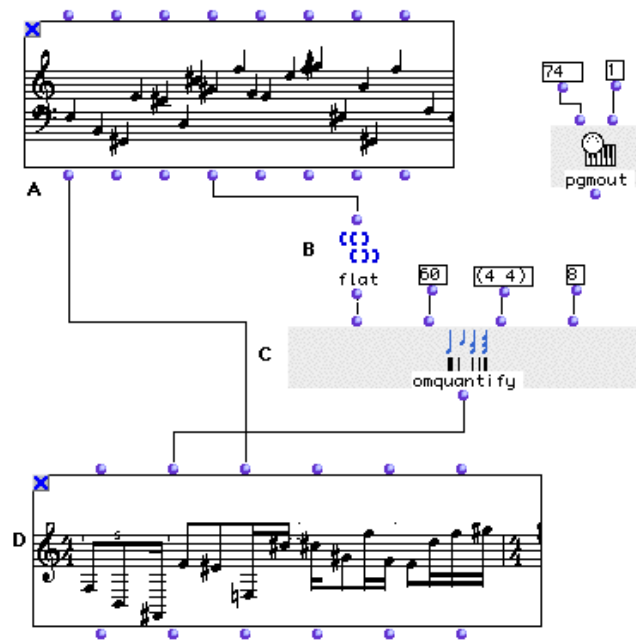


Figura 18. Visualización de un "patch" en OpenMusic

En esta figura, se está cuantificando una secuencia de acordes, puestos a manera de arpeggios, de tal manera que a la secuencia de notas se le asigna un valor rítmico determinado.

### *PatchWork*<sup>6</sup>

PatchWork, el predecesor de OpenMusic, que sigue en desarrollo de manera paralela. También está basado en CommonLisp. Al igual que OpenMusic, consiste en ventanas en las cuales hay diferentes tipos de elementos que pueden ser colocados, movidos y editados de diversas maneras para que las acciones o mecanismos programados puedan ser disparados mediante alguna acción externa, como el

<sup>6</sup> <http://recherche.ircam.fr/equipes/repmus/RMPapers/CMJ98/> "Computer Assisted Composition at IRCAM: PatchWork and OpenMusic".

### **MAX-RHYTHMATIST**

click del mouse o la presión de una tecla.

### ***Common Music***<sup>7</sup>

Common Music, es un lenguaje de programación enfocado a CAC en un entorno de programación orientada a objetos. Está basado en los lenguajes Scheme y C++. Permite la creación y manipulación de información musical (objeto), que puede ser transformada en sonido, partitura, u otras representaciones gráficas. Este sistema puede ser ampliado mediante el uso de paquetes de Lisp, estableciendo de esta manera comunicación con otros programas como CSound, Supercollider (de los cual se hará referencia posteriormente), Common Music Notation (para edición de partituras) y Common Lisp Music (sintetizador de sonido).

### ***Haskore***<sup>8</sup>

Haskore (Hudak) es un lenguaje de programación enfocado a CAC que soporta tanto la composición como la interpretación de una partitura. Es una colección de módulos diseñados para expresar estructuras musicales. Los objetos musicales en Haskore consisten en nociones simples como notas y silencios, operaciones para transformar objetos como la transposición, la escala de tiempo y operaciones que combinan objetos musicales para crear objetos más complejos, como una composición secuencial o concurrente (no secuencial o simultánea).

### ***Symbolic Composer***<sup>9</sup>

Este programa utiliza dos "gramáticas" para definir estructuras formales y de instrumentación: La "gramática de sección" y la "gramática de instrumentación". Ambas son estructuras en forma diagrama de árbol, en donde cada nodo contiene definiciones de "propiedades de clase". De tal forma que una sección consiste de un número de clases que definen símbolos, duraciones, tonalidades,

---

<sup>7</sup> <http://commonmusic.sourceforge.net/> "Sitio Oficial de Common Music".

<sup>8</sup> <http://www.haskell.org/haskellwiki/Haskore> "Sitio oficial de Haskore".

<sup>9</sup> [http://www.symboliccomposer.com/page\\_main.shtml](http://www.symboliccomposer.com/page_main.shtml) "Sitio oficial de Symbolic Composer".



### MAX-RHYTHMATIST

zonas, velocidades canales, programas, controladores, entre otros.

#### **1.4.2 Críticas a la composición asistida por computadora**

Nierhaus (Nierhaus, 2009. Págs. 259-273) menciona que las técnicas de composición algorítmica tiene una limitación en cuanto al estilo de música que puede generarse, y que las publicaciones referentes a las técnicas algorítmicas están, en buena medida, condicionadas por la posición estética y el ambiente profesional de los compositores que desarrollan o trabajan con CAC. Destaca además que la evaluación de los resultados en la composición musical están sujetos a preferencias subjetivas, aunque admite que el estudio de dichas técnicas es importante en la formación de los músicos, tanto como el estudio del Contrapunto o la Armonía. Menciona además que los sistemas de composición algorítmica generan, de manera análoga a las partituras de la música occidental, un nivel simbólico, y que más allá de este nivel hay un fenómeno complejo de interpretación y recepción. Pero sobre todo, establece que existe el peligro de tener una "creatividad automatizada", no porque la composición algorítmica *per se* así lo condicione, sino porque los algoritmos muchas veces son usados para generar un discurso musical restringido en cuanto a su estilo, por las razones mencionadas en líneas anteriores, por lo que en muchas ocasiones el problema no es la generación de algoritmos, sino el uso creativo de esos algoritmos por parte del compositor.

En cuanto a los sistemas informáticos, Miller Puckette (Puckette, 2006. Págs. 143-152) menciona que en la semántica de OpenMusic y Patchwork no hay una noción de eventos en tiempo real, y que en su lugar, el contenido de un "patch" (programación generada con estos lenguajes) se compone de datos estáticos, aunque reconoce que OpenMusic provee una función para secuenciar, que ejecuta el resultado obtenido en el programa, ya sea por medio del puerto MIDI hacia algún instrumento externo, o a través de un sintetizador virtual. Sin embargo, más allá de la posibilidad de ejecutar los resultados obtenidos en el programa, Puckette menciona que OpenMusic es ideal si lo que se busca

### MAX-RHYTHMATIST

es la creación musical de un resultado en tiempo diferido, de la misma manera en que la composición tradicional trabaja "fuera de tiempo" para la creación de una partitura, aunque el compositor use un piano para ejecutar sus ideas durante este proceso. Sin embargo, este proceso sería de poca utilidad para los ejecutantes, por lo que ellos encontrarían poca utilidad en OpenMusic, y que en su lugar se tendría que buscar una aplicación informática que busque la manera de romper la barrera entre la computación en tiempo-real y los datos, que es lo que buscó en primera instancia con Max y posteriormente con Pure Data (programas de los cuales se hará referencia posteriormente). En este último sentido, Puckette menciona que espera que la computadora sea capaz de funcionar más como un instrumento musical (y menos como una computadora) que como lo ha hecho hasta ahora, y que en la cultura de la música electrónica subyace la idea de hacer música por computadora como si se estuviera trabajando en un estudio, en donde se graba, corta, pega, mezcla y después se reproduce el resultado. Finalmente, menciona en este mismo sentido que los usuarios de programas enfocados a CAC, e incluso usuarios de Max/MSP y Pure Data, buscan en buena medida automatizar procesos, pero que él espera que estos últimos tiendan a una labor más "manual", contrario a la concepción de la CAC que privilegia la composición "pre-planeada" o "pre-programada", buscando en su lugar una exploración y experimentación del "patch", tal como se haría con un instrumento musical tradicional.

En lo que se refiere a los sistemas informáticos enfocados a la generación de patrones rítmicos empleando algoritmos genéticos, Nierhaus (Nierhaus, 2009. Págs. 157-184) menciona que Marcus Pearce critica, después de una serie de pruebas para comparar patrones rítmicos generados por humanos y por diferentes sistemas basados en algoritmos genéticos, que el resultado de estos últimos fue considerado como inexacto, en comparación con aquellos generados por humanos; esto debido a que el buen resultado de un algoritmo genético yace en pequeñas tareas, pero que estas no son

### MAX-RHYTHMATIST

capaces de generar ideas musicales como un todo, de manera que tengan sentido.

#### **1.4.3 Instrumentos Virtuales**

De acuerdo con Jordà (Jordà, 2004. Pág. 321), el término "instrumento musical virtual", o más simplemente "instrumento virtual" se usa para describir un sistema de cómputo diseñado para facilitar la expresión y la exploración musicales que respondan a la acción de un ejecutante para producir sonidos. De acuerdo con Rowe (Rowe, 1993. Pág. 40), el término "instrumento", referido a sistemas de cómputo, implica la idea de una interacción, que puede incluir aspectos tales como la grabación y la edición digital, y el uso de secuenciadores.

De acuerdo con Goto (Goto, 2000. Pág. 217), algunos instrumentos virtuales pueden producir, en menor o mayor medida, algún tipo de resultado visual que está relacionado con la acción del ejecutante y/o el audio generado. Este mismo autor menciona que debido a que no hay necesidad de establecer una relación directa, o bien una relación de efecto, entre la acción del ejecutante y los sonidos generados, la línea entre la composición y el instrumento se vuelve borrosa. Goto considera que podríamos hablar de composición musical en aquellos instrumentos en donde una sola acción del ejecutante dispara múltiples eventos musicales, por ejemplo, presionando un botón para generar varias notas o incluso varios minutos de música.

De acuerdo con Schnell y Battier (Schnell y Battier, 2002. Págs. 138-142), cuando se añaden más botones y otras posibilidades de interacción que dan al ejecutante más influencia sobre la música generada, se podría empezar a hablar del sistema como un "instrumento compuesto" (entendiendo el término "compuesto" derivado de la palabra composición musical), o bien, como una especie de "arreglo musical en tiempo real". Este instrumento-compositor puede usar procesos (algoritmos que generen material musical) y tal vez, fragmentos de música pre-compuesta, que son disparados y afectados por la acción del compositor-ejecutante.

### MAX-RHYTHMATIST

Según Wessel y Wright (Wessel y Wright, 2002. Pág. 11), a medida que se le da más control al ejecutante sobre el instrumento virtual, y sobre el tamaño de los eventos musicales disparados, ya sea por la duración o por la cantidad de información que se genera, entonces el instrumento virtual comienza a comportarse a un instrumento musical acústico, trabajando como un modelo de "un gesto por evento acústico".

#### **A) Criterios de diseño de instrumentos virtuales**

##### *Facilidad de uso y potencial expresivo*

Wessel y Wright (Wessel y Wright, 2002. Págs. 11-22) argumentan que los instrumentos necesitan proveer "una cuota de entrada baja sin límites sobre la virtuosidad" para poder considerarse como "expresivos". En otras palabras, necesitan proveer una facilidad de uso que no sacrifique la flexibilidad que le permita a un ejecutante, lo suficientemente motivado, desarrollar habilidades de alto nivel. Sin embargo, mencionan que hay cierta ambigüedad en lo que se refiere a la facilidad de uso, ya que hay un varios instrumentos tradicionales sobre los que se podría decir que hay una "alta cuota de entrada", que requieren cierto tiempo para desarrollar la habilidad de tocarlos, pero que proveen la suficiente motivación al ejecutante para iniciar su estudio y desarrollar habilidades de alto nivel; esto aunado al hecho de que tienen una rica historia de ejecuciones virtuosísticas previas -un ejemplo es el violín, que tiene una historia de grandes ejecutantes- por lo que el éxito de un instrumento no puede ser dissociado del contexto social.

El concepto de virtuosidad tampoco escapa a la ambigüedad, en lo que se refiere al diseño de herramientas de cómputo enfocados a la virtuosidad. Tanto Wessel y Wright, Jordà, Fels y Dobrian, y Koppelman (Wessel y Wright, 2002. Págs. 11-22) (Jordà, 2004. Págs. 321-341) (Fels, 2004. Págs. 672-685) (Dobrian y Koppelman, 2006. Págs. 277-282), coinciden en que el instrumento debe dar al ejecutante un amplio rango de posibilidades expresivas, es decir, no debe ser demasiado restrictivo,

### *MAX-RHYTHMATIST*

ni responder en formas que sean demasiado simples. Al mismo tiempo, los instrumentos deben ser capaces de producir un rango amplio de sonidos ricos y complejos, pero esto no debe darse a expensas de la transparencia y la repetibilidad. Paine aclara aún más esto al decir que el instrumento debe ser diseñado de tal forma que pueda darse una extensa exploración y descubrimiento de nuevos resultados, pero dichos resultados deben poder ser nuevamente obtenidos, es decir, la repetición de los resultados debe confirmar el "mapa cognitivo" que el ejecutante está desarrollando, al mismo tiempo que su relación con el sistema interactivo se profundiza.

### *Intimidad y Transparencia*

Fels (Fels, 2004. Págs. 672-685) argumenta que la "Intimidad" es la clave para el éxito en el diseño de un instrumento. La intimidad, en este caso, se refiere al grado en el que un ejecutante se siente en control del instrumento. Ahondando en el argumento de Moore, Fels dice que la intimidad ("intimidad de control" en términos de Moore) ocurre cuando el ejecutante siente una conexión cercana entre la acción física y el sonido producido por el instrumento. De manera conceptual, Fels añade que hay cuatro tipos de relación que se puede formar entre la persona y los objetos:

- *Respuesta:* El objeto es percibido como una entidad separada que responde a alguna acción. La interacción en esta relación puede verse como una especie de conversación simple en donde la persona ejecuta algún movimiento o gesto y observa como responde el sistema. El punto aquí es que el gesto y la respuesta se perciben como separados uno del otro. La persona puede observar que el sistema responde a sus acciones, pero ella aún está descubriendo de qué manera responde este.
- *Control:* El objeto es percibido como una extensión del cuerpo. En esta relación, la acción y la respuesta se encuentran unidas y se tiene la sensación de controlar y ejecutar el sistema, y no de una respuesta como una entidad separada.

### MAX-RHYTHMATIST

- *Reflexión:* El objeto es percibido como una entidad separada que no responde a ninguna acción. Aquí se puede poner como ejemplo a una exhibición en una galería de arte. La persona puede tener una respuesta compleja ante las pinturas, se puede decir incluso que se da cierta comunicación, pero la pintura por sí misma no es afectada por esta interacción.
- *Pertenencia:* La persona se siente una extensión del objeto. En otras palabras, las acciones son controladas por el objeto. Fels dice que "para este tipo de relación, el objeto debe ser capaz de controlar a la persona y la persona debe estar dispuesta a ser controlada", y nota que es muy difícil diseñar dispositivos que tengan este tipo de relación.

Con referencia a los conceptos enunciados por Fels, Costello (Costello et al, 2005. Págs. 49-55) menciona que estas relaciones forman un continuo y que pueden coexistir, y que con frecuencia las personas de mueven entre la relación de respuesta y de control durante la interacción. Fels dice que construir una fuerte relación de respuesta es una pre-condición necesaria para moverse después a una íntima relación de control. Es decir, aprender cómo funciona un sistema es necesario antes de desarrollar habilidades expertas. Un buen sistema, por lo tanto, debe facilitar y motivar la transición entre las relaciones de respuesta y control.

Un instrumento que facilita la intimidad debe exhibir transparencia en su operación. Esto es, las conexiones entre la ejecución del ejecutante y los sonidos producidos deben ser claras y comprensibles, tanto para el ejecutante como para la audiencia. Fels (Fels et al, 2002. Págs. 109-126) argumenta que el grado de transparencia permite prever, a su vez, tener un cierto grado de expresividad.

Tanto Fels, como Jordá, Schnell y Battir (Fels et al, 2002. Págs. 109-126) (Jordà, 2002. Págs. 23-29) (Schnell y Battier, 2002. Págs. 138-142), señalan que una consecuencia de la tendencia actual a usar computadoras portátiles en ejecuciones en vivo es que hay una carencia de relación entre lo que la

### MAX-RHYTHMATIST

audiencia percibe, las acciones de los ejecutantes y los sonidos y/o imágenes resultantes. De acuerdo con ellos, las ejecuciones en vivo que presentan a ejecutantes sentados frente a sus laptops creando y manipulando sonidos y/o imágenes son ahora muy comunes, y critican el hecho de que desde la perspectiva de la audiencia, las acciones físicas de los ejecutantes están prácticamente disociadas de los sonidos que producen.

Collins (Collins, 2003. Págs. 67-79) menciona que la práctica del "Live Coding" (programación en vivo) es una técnica que ha surgido recientemente para restablecer la conexión entre la percepción de la audiencia y la generación audiovisual por computadora. En el Live Coding, la pantalla de la computadora del ejecutante se hace visible a la audiencia, y esta puede discernir, en cierta medida, como las acciones de los programadores causan un impacto en la música o la imagen.

Sobre esta problemática, Miller Puckette (Puckette, 2006. Págs. 143-152) señala que la relación entre las acciones del ejecutante y el resultado que envía la computadora es significativa no sólo para la audiencia, sino también para la construcción de tradiciones de ejecución:

"Para hacer posible el compartir ideas musicales, debe haber una relación directa y comprensible entre los controles que usamos y los sonidos que escuchamos. (Esto tampoco sería malo desde el punto de vista de la audiencia). Un ejecutante que presiona un botón y comienza una secuencia no está mostrándonos como se hizo la música; todo lo que podemos aprender sobre esa música es lo que nuestros oídos nos pueden decir acerca de ella. Pero si las acciones del ejecutante corresponden de manera más cercana a los sonidos mismos, entonces podemos ver algo del contenido gestual de la música, y este contenido puede dar mayor información sobre la música generada. De esta manera podemos desarrollar una tradición mucho más fuerte y significativa en la creación de música por computadora." (Puckette)

## **B) Lenguajes de Programación para el diseño de instrumentos virtuales**

En este punto se listan lenguajes de programación, tanto de propósito general, como enfocados directamente en aspectos musicales.

### *Lenguaje C*

Dennis M. Ritchie en su texto "El desarrollo del lenguaje C"<sup>10</sup> menciona la forma en que este lenguaje de programación se ha desarrollado y las variantes que ha tenido. El desarrollo inicial del lenguaje C se llevó a cabo en los Laboratorios Bell de AT&T entre 1969 y 1973; según Ritchie, el periodo más creativo tuvo lugar en 1972. Se le dio el nombre "C" porque muchas de sus características fueron tomadas de un lenguaje anterior llamado "B".

Uno de los objetivos de diseño del lenguaje C es que sólo sean necesarias unas pocas instrucciones en lenguaje máquina para traducir cada elemento del lenguaje, sin que haga falta un soporte intenso en tiempo de ejecución. Es muy posible escribir C a bajo nivel de abstracción; de hecho, C se usó como intermediario entre diferentes lenguajes.

En parte a causa de ser de relativamente bajo nivel y de tener un modesto conjunto de características, se pueden desarrollar compiladores de C fácilmente. En consecuencia, el lenguaje C está disponible en un amplio abanico de plataformas (seguramente más que cualquier otro lenguaje). Además, a pesar de su naturaleza de bajo nivel, el lenguaje se desarrolló para incentivar la programación independiente de la máquina. Un programa escrito cumpliendo los estándares e intentando que sea portátil puede compilarse en muchos computadores.

C se desarrolló originalmente (conjuntamente con el sistema operativo Unix, con el que ha estado asociado mucho tiempo) por programadores para programadores. Sin embargo, ha alcanzado una popularidad enorme, y se ha usado en contextos muy alejados de la programación de sistemas, para

---

<sup>10</sup> <http://cm.bell-labs.com/cm/cs/who/dmr/chist.html> "The Development of C Language"



## MAX-RHYTHMATIST

la que se diseñó originalmente.

Desde el inicio del lenguaje han surgido varias ramas de evolución que han generado varios lenguajes:

- **Objective C** es un primer intento de proporcionar soporte para la programación orientada a objetos en C, de escasa difusión, pero actualmente usado en Mac OS X y GNUstep.
- **C++** (pronunciado *C Plus Plus*) diseñado por Bjarne Stroustrup fue el segundo intento de proporcionar orientación a objetos a C y es la variante más difundida y aceptada. Esta versión combina la flexibilidad y el acceso de bajo nivel de C con las características de la programación orientada a objetos como abstracción, encapsulación y ocultación.

De igual manera, se han desarrollado librerías especializadas en diversas áreas, que permiten al lenguaje C enfocarse en soluciones especializadas. Dentro del ámbito musical está:

- **The CLAM project**<sup>11</sup>. La librería para Audio y Música en C++ (CLAM, por sus siglas en inglés) es, como su nombre lo indica, una librería enfocada en la programación de aplicaciones musicales en C++. El proyecto se originó en 2000 por iniciativa de Xavier Amatriain y Maarten de Boer en la Universidad Pompeu Fabra de Barcelona, y puede ser usado en Linux, MacOS y Windows. Dentro de las características que sus desarrolladores describen está la posibilidad de crear aplicaciones que puedan ser empleadas en tiempo real.

### *Lenguaje Java* <sup>12</sup>

**Java** es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos

---

<sup>11</sup> [http://clam-project.org/wiki/Frequently\\_Asked\\_Questions](http://clam-project.org/wiki/Frequently_Asked_Questions) "The CLAM project"

<sup>12</sup> [http://es.wikibooks.org/wiki/Programaci%C3%B3n\\_en\\_Java](http://es.wikibooks.org/wiki/Programaci%C3%B3n_en_Java) "Programación en Java"

### MAX-RHYTHMATIST

errores, como la manipulación directa de punteros o memoria.

El lenguaje Java se creó con cinco objetivos principales:

1. Debería usar la metodología de la programación orientada a objetos.
2. Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
3. Debería incluir por defecto soporte para trabajo en red.
4. Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
5. Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

La primera característica, orientado a objetos (“OO”), se refiere a un método de programación y al diseño del lenguaje. Aunque hay muchas interpretaciones para OO, una primera idea es diseñar el software de forma que los distintos tipos de datos que usen estén unidos a sus operaciones. Así, los datos y el código (funciones o métodos) se combinan en entidades llamadas objetos.

Java se ha convertido en un lenguaje con una implantación masiva en todos los entornos (personales y empresariales).

Dentro de las librerías enfocadas en la programación de aplicaciones musicales se encuentra:

- **The JMusic Library**<sup>13</sup>. De manera similar al proyecto CLAM, la librería JMusic es una librería enfocada en el desarrollo de aplicaciones musicales. Fue desarrollada en la Universidad Tecnológica de Queensland en Australia, y fue diseñada para ayudar en procesos de composición musical a través de un ambiente estructurado para la exploración musical, de igual manera, puede ser usada para análisis musical y en educación musical por computadora.

### *CSound*<sup>14</sup>

**Csound** es un paquete de software orientado a crear, editar, analizar y componer música y sonido.

También es llamado así el propio lenguaje de programación que se usa para controlar el software.

<sup>13</sup> <http://jmusic.ci.qut.edu.au/> "The JMusic Library"

<sup>14</sup> <http://csounds.com/about> "Página oficial de CSound"

### MAX-RHYTHMATIST

Funciona correctamente en máquinas pequeñas y lentas, aunque si se lo corre en equipos más potentes, es posible utilizarlo en tiempo real crear interfaces gráficas de usuario, y conectarlo a otros softwares vía drivers o red.

El funcionamiento del Csound (así como el de los programas Music1, Music2, etc.) es trabajar con dos clases de objetos relevantes en la composición.

Por un lado se debe crear la «orquesta» (en inglés *orchestra*), que se compone de los instrumentos que se utilizarán en la composición.

En Csound hay que crear los instrumentos, es decir, realizar una descripción completa de cómo son y cómo funcionan (cómo suenan): esto puede ir desde un oscilador que genere un tono puro (sinusoide) de un frecuencia determinada (de sonido similar a un diapasón real) a un instrumento complejo cuyo timbre varía estadísticamente.

Por otro lado está la partitura (en inglés *score*), que es el segundo objeto relevante. Esta no es más que una tabla o gráfica donde se especifica el orden de actuación de los instrumentos a lo largo del tiempo.

Desde la versión 5.0, se da un cambio significativo en código de Csound. Aparecen características como: mejora de la gestión de "tiempo real", APIs<sup>15</sup> para otros lenguajes de programación como Python o Lisp, Interfaces gráficas definidas por el usuario, protocolos de hosting a VST y LADSPA, protocolo OSC, opcodes definidas por el usuario y un gran número de nuevos opcodes de orquesta.

Además en la versión para linux, se acopla perfectamente con Jack<sup>16</sup>.

---

<sup>15</sup> Una **interfaz de programación de aplicaciones** o **API** (del inglés *Application Programming Interface*) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usados generalmente en las bibliotecas.

<sup>16</sup> Conector de audio virtual para interconectar software que reciba, envíe y procese señales o archivos de audio.

## MAX-RHYTHMATIST

### *Supercollider*<sup>17</sup>

Supercollider es un lenguaje y ambiente de programación originalmente lanzado por James McCartney en 1996 para realizar síntesis de audio en tiempo real y composición algorítmica, a partir de 2002 es distribuido bajo los términos de la GNU (General Public License), por lo que SuperCollider se convirtió en software libre. La última versión (3.4) fue lanzado en julio de 2010. Desde su origen ha ido evolucionando hasta convertirse en un sistema usado y desarrollado por científicos y artistas que trabajan con el sonido. Es un lenguaje de programación dinámica que provee un espacio de trabajo interesante para la investigación acústica, la música algorítmica y programación interactiva, y puede ser usado para generar interfaces gráficas de usuario.

### *Max/MSP/Jitter*<sup>18</sup>

**Max** es un entorno de desarrollo gráfico para música y multimedia desarrollado y mantenido por Cycling '74, una empresa de programas situada en San Francisco. El programa ha sido usado durante más de quince años por compositores, artistas y diseñadores de programas interesados en la creación de programas interactivos.

Miller Puckette originalmente escribió Max, en ese entonces nombrado *Patcher*, un editor para el sistema Macintosh en el IRCAM a mediados de los años 80 para que los compositores tuvieran acceso a un sistema de composición de música interactiva hecha con ordenadores. Primero fue usado en una composición de piano y ordenador llamada *Plutón*, sincronizando el ordenador con el piano y controlando un sintetizador Sogitec 4X, el cual procesaba el audio.

En 1989, IRCAM desarrolló y mantuvo una versión de Max que fue trasladada a la Estación de informática musical (este nombre proviene del francés, «Station d'informatique musicale») para el sistema NeXT (y después para los sistemas SGI y Linux), llamada Max/FTS (las siglas FTS

<sup>17</sup> <http://supercollider.sourceforge.net/> "Página de Supercollider"

<sup>18</sup> <http://cycling74.com/products/maxmspjitter/> "Página de Cycling 74"

### MAX-RHYTHMATIST

significan «Faster Than Sound», literalmente, «Más rápido que el sonido», y es una analogía a su precursor MSP; este último fue mejorado al implementar el soporte físico DSP al ordenador).

Después se autorizó la venta del programa por Opcode Systems, Inc., el cual publicó una versión comercial del programa en 1990 llamado Max/Opcode (desarrollado por David Zicarelli). Desde el 1999, la versión actual de Max es distribuida por la empresa de Zicarelli, Cycling74 (fundada en 1997).

En 1996, Puckette publicó un software libre completamente rediseñado llamado Pd (*Pure Data*), el cual contiene varias diferencias fundamentales con el IRCAM original, sin embargo, sigue siendo un sustituto interesante para aquellos que no quieran gastar cientos de dólares en Max/MSP.

Max tiene numerosas extensiones y encarnaciones; en particular, las extensiones de audio que aparecieron en 1997, trasladadas de Pure Data. Estas fueron llamadas **MSP** (las iniciales de Miller S. Puckette, autor de Max y Pd). Estas adiciones para Max permitieron que el audio digital sea manipulada en tiempo real, y a la vez, permitía a los usuarios la creación de sus propios sintetizadores y efectos-procesadores. Previamente, Max había sido diseñado con el fin de tener un terreno común con sintetizadores reales, samplers, etcétera, como un «lenguaje de control» usando MIDI o algún otro protocolo de red.

En 1998, un «descendiente directo» de Max/FTS fue desarrollado en Java (llamado jMax) y fue publicado como un programa de código abierto, este se podía descargar de la página del IRCAM, pero actualmente ha dejado de desarrollarse.

En el 2003, la segunda adición para el Max/MSP, llamada *Jitter*, fue publicada, agregándole la posibilidad de procesar vídeo, 3D, y matrices en el programa. 2 años más tarde se incorporan al software los objetos *mxj* y *js*. Estos objetos son "externals" que ejecutan código Java y javascript en Max.

### MAX-RHYTHMATIST

Las unidades donde se programa el código se llaman “patch” o abstracciones, son utilizadas como programas independientes y compartidos libremente entre la comunidad de usuarios de Max/MSP. Los patches constan de diferentes objetos interconectados entre ellos. En su parte superior encontraremos las entradas, donde se les enviarán valores numéricos u otros tipos de datos, y en la posterior la salida de estos. También existe la posibilidad de crear patches secundarios conocidos como subpatches. Están dentro del patch principal. Tanto los patches, como los objetos externos, ambos desarrollados en su mayor parte por usuarios de Max pueden ser compartidos con la comunidad de usuarios.

Entre los patches y objetos relacionados con el manejo rítmico llama la atención el siguiente ejemplo:

- **Real-Time Composition Library.**<sup>19</sup> Es una colección de módulos y objetos para Max/MSP, desarrollados en 1992 por Karlheinz Essl, que sirven como un ambiente gráfico interactivo para multimedia, música y MIDI. Ofrece la posibilidad de experimentar con varias técnicas compositivas tales como procedimientos seriales, permutaciones y aleatoriedad controlada. Dentro del aspecto rítmico, se manejan funciones basadas en el tiempo, las cuales son manejadas en los siguientes módulos u objetos:
  - a) La generación de ritmo Browniano. Módulo basado en el movimiento browniano, que es en pocas palabras, un movimiento aleatorio definido por una función probabilística.
  - b) Escalas rítmicas. Módulo para generar escalas rítmicas, definidas como una secuencia de valores rítmicos que van de un valor mínimo a un máximo en un número de pasos o divisiones elegidos por el usuario, de manera lineal, logarítmica o exponencial.
  - c) Super-ritmo. Objeto externo que tiene una serie de retardos (delays) cuyo valor puede ser

---

<sup>19</sup> <http://www.essl.at/works/rtc.html#abs> "Página oficial de Karlheinz Essl"

### *MAX-RHYTHMATIST*

asignado por medio de los principios seriales, secuenciales o aleatorios, y que genera valores rítmicos con los valores asignados por cualquiera de los principios antes mencionados.

Esta librería resultó ser muy interesante por la manera en que generaba ideas rítmicas, especialmente por el uso de escalas rítmicas, y por la manera en que hacía interactuar el ritmo con otros parámetros musicales como la altura. Sin embargo, la mayor parte de sus procesos implicaban aleatoriedad, y la aplicación que en esta tesis se describe busca, por el contrario, tener un control muy preciso en los patrones e ideas rítmicas que se generen.

### *Pure Data*<sup>20</sup>

**Pure Data (o Pd)** es un lenguaje de programación gráfico desarrollado por Miller Puckette durante los años 90 para la creación de música computerizada interactiva y obras multimedia. Aunque Puckette es el principal autor del software, Pd es un proyecto de código abierto y tiene una gran base de desarrolladores trabajando en nuevas extensiones al programa.

Pd es muy similar en alcance y diseño al programa original de Puckette, Max(desarrollado cuando él estaba en IRCAM), y es hasta cierto grado interoperable con Max/MSP, el sucesor comercial del lenguaje Max. Ambos Pd y Max son ejemplos discutibles de lenguajes de programación de "flujo de datos". En este tipo de lenguajes, funciones u "objetos" son conectados o "parcheados" unos con otros en un ambiente gráfico que modela el flujo del control y el audio.

Como Max, Pd tiene una base modular de código con externos u objetos que son utilizados como bloques de construcción para programas escritos en el software. Esto hace el programa arbitrariamente extensible a través de una API pública, y alienta a los desarrolladores a añadir sus propias rutinas de audio y control, ya sea en el lenguaje de programación C o, con la ayuda de otros

---

<sup>20</sup> <http://puredata.info/> "Página oficial de Pure Data"

### *MAX-RHYTHMATIST*

externos, en Python, Javascript, Ruby, y potencialmente otros lenguajes también. Sin embargo, Pd es un lenguaje de programación en sí mismo. Unidades de código modulares y reusables, escritas nativamente en Pd, llamadas "parches" o "abstracciones", son usadas como programas independientes y compartidas libremente entre la comunidad de usuarios de Pd.

Con la adición del externo "Entorno Gráfico para Multimedia" (GEM, por su nombre en inglés), y otros externos diseñados para trabajar con él (como el Pure Data Packet "PiDiP" para Linux, "Framestein" para Windows, "GridFlow" para proceso de matrices n-dimensionales que integra Pure Data con el lenguaje de programación Ruby, etc.), es posible crear y manipular vídeo, gráficos OpenGL, imágenes, etc, en tiempo real con aparentemente infinitas posibilidades de interactividad con audio, sensores externos, etc.

Adicionalmente, Pd está diseñado para permitir colaboración en vivo a través de redes o de Internet, permitiendo a músicos conectados vía LAN, o incluso en distintas partes del mundo, hacer música juntos en tiempo real.

### *Proyectos informáticos relacionados con el ritmo*

En lo relacionado a proyectos informáticos que impliquen algo similar a lo que este trabajo de tesis plantea, el único ejemplo relevante fue el **Intelligent Rhythm Generator**<sup>21</sup>; se trata de proyecto aún en desarrollo ideado por Eduardo Reck Miranda, Marcelo Gimenes y Chris Johnson en la Universidad de Plymouth, Reino Unido. Consiste en un programa que recibirá información rítmica proveniente de un instrumento musical interpretado por un ser humano, de tal forma que el programa lo pueda decodificar. Una vez analizado, el programa se encargará de realizar una improvisación rítmica, basada en la información previamente recibida. Este proyecto se inició en 2003, y a la fecha de la elaboración de esta tesis (2011) aún se encuentra en desarrollo, lo que muestra la complejidad

---

<sup>21</sup> [http://cmr.soc.plymouth.ac.uk/publications/Gimenes\\_Glasgow\\_def.pdf](http://cmr.soc.plymouth.ac.uk/publications/Gimenes_Glasgow_def.pdf) "Proyecto Intelligent Rhythm Generator"



de su elaboración.

## **1.5 ANÁLISIS Y OPINIONES ACERCA DEL ESTADO DEL ARTE**

### **1.5.1 Sobre la concepción y las investigaciones referentes al ritmo**

La información mostrada en este capítulo muestra que la concepción del ritmo, y de los elementos que lo constituyen es un tema controversial, dado que dependiendo del punto de vista desde el cual se analice, las definiciones pueden llegar a ser diferentes. Por un lado, no se ve una línea de pensamiento continua entre las concepciones de Meyer y Cooper de 1960, y la concepción orgánica del ritmo de Dewey de 1934, o la concepción dinámica de Hamilton del presente siglo.

Concepciones estáticas, más en concordancia con las ideas de Meyer y Cooper, también se encuentran presentes en este siglo con Budd y Boghossian, y la idea del ritmo libre de Clayton de 1996 está más en concordancia con las ideas de Hamilton. El aspecto del ritmo libre mencionado por Clayton también se encontró presente en la descripción de la estructura rítmica del canto gregoriano de Martínez Soques y también es mencionado en las investigaciones de Frigyesi, todos ellos concordando con la idea de que el ritmo puede aparecer aún sin la presencia de un pulso claro. Se puede concluir que el problema de la concepción del ritmo y su definición desde el punto de vista musicológico, filosófico y técnico es un tema que aún no logra un consenso entre los especialistas. Para efectos de esta tesis, se tomaron más en cuenta las definiciones de Meyer y Cooper para iniciar la exposición teórica referente al ritmo porque, entre todos los investigadores mencionados en este párrafo, ellos fueron los que hicieron un trabajo teórico mucho más meticuloso para definir con la mayor claridad posible todos los elementos que conformaban al ritmo, y a partir de estas definiciones fue más fácil presentar las ideas de los otros teóricos.

La información recabada en el ámbito de la investigación científica reveló que existe una amplia

### MAX-RHYTHMATIST

investigación en la percepción psicoacústica del ritmo, y que el conocimiento generado hasta el momento sigue aún en evolución.

Un punto en común entre Meyer, Cooper y Hamilton fue la concepción del acento como resultado de varios factores, lo cuál hace de este aspecto del ritmo algo subjetivo, en concordancia con hallazgos científicos en el ámbito psicoacústico, mencionados en la segunda sección de este capítulo. En lo que a este último aspecto se refiere, las investigaciones de Fraisse, Large, Parncutt, Povel y Okkerman referentes al fenómeno de metrización subjetiva demuestran que el proceso de agrupamiento de elementos temporales en estructuras más largas ocurre aún cuando no esté apoyado por diferencias objetivas. Esto sugiere que los intervalos temporales por sí mismos son responsables de la acentuación y la determinación de las estructuras rítmicas.

#### **1.5.2 Sobre las implicaciones que el estado del arte tiene en la concepción del modelo aritmético en esta tesis**

El aspecto multi-factorial del acento llevó a la premisa de pensar en el aspecto de la acentuación y del agrupamiento como elementos que podían ser secundarios en la concepción de la implementación informática. Esto permite que la implementación se centre básicamente en el control de la generación de las duraciones de los eventos para generar patrones y secuencias de patrones rítmicos.

Esta simplificación en la forma de generar patrones rítmicos, por tanto, no implica que se elimine la posibilidad de crear acentos o agrupamientos de manera intencional, tan sólo significa que existe la opción de no crearlos, dejando que la interacción de los patrones rítmicos con los diferentes elementos musicales, tales como las alturas, las intensidades, las simultaneidades, etc., vaya creando la percepción de acentos y agrupamientos, y por tanto, de jerarquización.

Esta decisión se apoya en el concepto de "puntos de ataque", proveniente del Modelo de Estratos

### MAX-RHYTHMATIST

Rítmicos de Yeston, que sugiere la posibilidad de traducir un ritmo escrito como una secuencia de números, sin importar la jerarquía que tuvieran los valores rítmicos implicados, lo cual sugiere que era necesario buscar una manera sistemática de realizar esta traducción numérica. Aquí es importante aclarar, más allá de lo ya mencionado en este capítulo, que si bien Novaro no concibió sus series numéricas para la generación rítmica, su utilización en esta tesis resultó de gran utilidad. Dado que el fenómeno de percepción de los intervalos, referente a las alturas musicales, es un fenómeno exponencial, el uso de escalas exponenciales permite dividir una octava en partes iguales, no así las escalas geométricas, más propias para fenómenos lineales, como la percepción temporal. Las escalas exponenciales de Novaro no pueden dividir espacios temporales en partes iguales, pero sus escalas geométricas, representadas por series numéricas, sí lo pueden hacer; está es la razón principal por la que las series de Novaro son tomadas en cuenta para la traducción de eventos rítmicos en series numéricas. En consecuencia, una serie numérica puede representar un patrón rítmico, y para ello, se toma al 1 como unidad básica para hacer una abstracción de este, de tal forma que una serie numérica dividida a la unidad en partes, y la serie generada pueda ser usada para generar un patrón rítmico, cuya duración pueda ser decidida posteriormente. Este es el primer algoritmo de generación de patrones rítmicos.

Además de series rítmicas de carácter regular, basadas en las series de Novaro, y gracias a la concepción de los patrones rítmicos como una división de la unidad en partes, se abre la posibilidad de trabajar con el concepto de ritmo libre de Clayton. Esto implica que se puede dividir la unidad de generación rítmica (el número 1) de forma que esta división no necesariamente provenga de una división regular o de algún algoritmo basado en series; para lograr esto, se decidió que el uso de divisiones aleatorias es la forma más simple y eficaz.

### **1.5.3 Sobre las implicaciones que el estado del arte tiene en el diseño de la aplicación informática**

Para enfocar el problema, se decidió que lograr imitar las técnicas rítmicas desarrolladas en la música de concierto del siglo XX debía ser un requisito inicial del instrumento. Dado que estas técnicas son muy variadas, pretender imitarlas a todas ellas podía implicar que el programa fuera demasiado complicado de usar. El programa entonces debía contener dos partes, una en la que los patrones rítmicos pudiera secuenciarse, acercándose más al acto de componer, y otra en donde pudieran ejecutarse directamente a través de una acción del usuario, acercándose más al acto de ejecutar. Por otro lado, el acceso al código fuente del programa también era un requisito indispensable para que el usuario pudiera adaptar la aplicación informática tanto a sus necesidades expresivas, como al logro de una mayor sutileza en la imitación de técnicas rítmicas contemporáneas.

Se decidió programar la aplicación informática como un instrumento virtual por varias razones. En primera, al hacer una evaluación de los lenguajes de programación enfocados a la CAC, se encontraron varias dificultades. Por ejemplo, en el momento de inicio de esta tesis, OpenMusic aún no estaba disponible para Windows, lo cual impedía a los usuarios de este sistema poder utilizar la aplicación. Además, la ayuda que proveía el lenguaje al usuario estaba escrita de manera fragmentada en varios idiomas, lo cual dificultaba el trabajo de programación; estos dos problemas se fueron resolviendo en versiones recientes de OpenMusic mientras se desarrollaba esta tesis.

La concepción rítmica en OpenMusic partía de la idea de extraer información de una partitura, o bien, de trasladar datos numéricos a ella, por lo que la idea de ritmo libre se veía comprometida, y también entraba en contradicción con la decisión de concebir la generación de patrones rítmicos a partir de una división de la unidad en partes, sin que estas representen una idea preconcebida de jerarquía. Finalmente, la labor realizada por diversos lenguajes de programación y aplicaciones

### MAX-RHYTHMATIST

informáticas enfocadas a CAC contenían ya una labor importante en el aspecto rítmico, por lo que una nueva aplicación informática relacionada con este tema y tomando el enfoque de la CAC carecería de relevancia.

Concebir la aplicación informática de esta tesis como instrumento virtual de cualquier manera implicaba resolver varios problemas. A pesar de los esfuerzos de Wessel, Wright y Fels por definir los requisitos que un instrumento virtual debe de tener, como lo son la expresividad, la flexibilidad, la intimidad y la transparencia, dichas características son difíciles de evaluar debido al alto contenido de subjetividad implícito en ellos.

#### **1.5.4 Sobre las implicaciones que el estado del arte tiene en la elección del lenguaje de programación**

Se realizó una revisión de las coincidencias que había entre las aplicaciones informáticas listadas en este capítulo y las utilizadas por diversas instituciones dentro de sus diferentes cursos, encontrándose la siguiente información:

El IRCAM<sup>22</sup> (acrónimo de *Institut de recherche et coordination acoustique/musique*), utiliza OpenMusic y Max/MSP para sus cursos de formación para artistas. Esta misma institución proporciona cursos de capacitación y maestrías para programadores enfocados al arte digital utilizando los lenguajes C y Java

El Instituto Universitario del Audiovisual<sup>23</sup>, perteneciente a la Universidad Pompeu Fabra en Barcelona, dentro de sus cursos de preparación para artistas utiliza Lisp, Max/MSP, Pure Data, Reaktor y CSound.

La Universidad Pompeu Fabra<sup>24</sup>, por otro lado, en su maestría en Cómputo Musical y Sonido,

---

<sup>22</sup> <http://www.ircam.fr/> "Página oficial del IRCAM"

<sup>23</sup> <http://iua.upf.edu/> "Página oficial del Instituto Universitario del Audiovisual"

<sup>24</sup> <http://www.upf.edu/smc/> "Página de la Maestría en Cómputo Musical y Sonido de la Universidad Pompeu Fabra"

### MAX-RHYTHMATIST

incluye los lenguajes C y Java.

El CNMAT<sup>25</sup> (acrónimo de *The Center for New Music and Audio Technologies*) perteneciente a la Universidad de Berkeley en Estados Unidos utiliza y desarrolla investigación con Max/MSP.

El CMMAS<sup>26</sup> (acrónimo de *Centro Mexicano para la Música y las Artes Sonoras*) de Morelia ha realizado la mayoría de sus cursos de formación para artistas utilizando Max/MSP.

El Centro Multimedia<sup>27</sup> del Centro Nacional de las Artes de la Ciudad de México ha realizado cursos de capacitación para artistas utilizando Max/MSP, Pure Data, y de manera más reciente y frecuente, Supercollider.

Esta revisión reveló que el uso de lenguajes de propósito general, como C o Java, estaba poco extendido dentro de las instituciones revisadas, eso implicaba que el acceso al código fuente al usuario no le representaría una posibilidad realista de adaptar la aplicación informática a sus necesidades si se decidía utilizar cualquiera de estos lenguajes, por lo que su uso se descartó.

Los lenguajes de programación más comunes dentro de las instituciones revisadas eran Max/MSP, Pure Data y Supercollider. Cualquiera de los tres daba la posibilidad al usuario de tener acceso a su código, por lo que eran elecciones factibles. Supercollider se descartó debido a la poca eficacia que tenía en la generación de interfaces gráficas, y a la dificultad de usar el protocolo MIDI. Max/MSP y Pure Data son lenguajes de programación muy similares; la facilidad de acceso y modificación del código de un "patch" programado en cualquiera de ellos, y la mayor facilidad de diseño de interfaces gráficas fueron factores determinantes para que estos dos lenguajes fueran las dos opciones finales. Ambos lenguajes tenían la dificultad de manejar listas de datos de manera eficiente, característica que era esencial para el buen funcionamiento de la aplicación informática dada la forma en que se

---

<sup>25</sup> <http://cnmat.berkeley.edu/> "Página oficial del CNMAT"

<sup>26</sup> <http://www.cmmas.org/> "Página oficial del CMMAS"

<sup>27</sup> [www.cenart.gob.mx/html/cmmp.html](http://www.cenart.gob.mx/html/cmmp.html) "Página del Centro Multimedia del CENART"

## MAX-RHYTHMATIST

planeaba generar, almacenar y utilizar los patrones rítmicos generados. Al hacer una revisión de los objetos externos disponibles para ambos lenguajes, se encontraron los "LObjects", objetos para Max/MSP que permitían manejar listas de muy variadas maneras; por el lado de Pure Data, no se encontraron objetos similares. Esto último inclinó la balanza hacia Max/MSP, a eso se agregaba la posibilidad, no explorada en esta tesis, de incluir más objetos externos, tanto relacionados con el manejo rítmico (como los desarrollados por Karheinz Essl), así como otros enfocados al manejo de alturas, efectos sonoros, etc, si es que el usuario así lo deseaba.

### **1.6 OBJETIVOS DEL TRABAJO DE TESIS**

De lo expuesto en este capítulo, el modelo aritmético de representación rítmica que se desarrolla en esta tesis debe:

1. Limitarse a describir aquellos parámetros puramente temporales del fenómeno rítmico, que son la sucesión y duración de los valores rítmicos. Dejando de lado aspectos como las alturas, que son un elemento agregado, o la posible acentuación que del patrón rítmico se perciba, dado que ya se ha aclarado que el fenómeno de la acentuación tiene aspectos subjetivos, tanto por Cooper y Meyer, como por Fraisse, Large y Parncutt.
2. Definir la duración de un valor rítmico y su posición dentro de una estructura rítmica más grande, manteniendo con ello la idea de construcción arquitectónica del ritmo de acuerdo a las ideas de Cooper y Meyer que ya se han expuesto.
3. Poder combinar todas las características anteriores para analizar y/o generar estructuras rítmicas en varios niveles de subdivisión, desde elementos de duración muy pequeña, hasta conjuntos de elementos que conformen una duración mayor.

En cuanto a la aplicación informática, ésta debe:

### *MAX-RHYTHMATIST*

1. Implementar el modelo de representación rítmica, con las características mencionadas en los tres puntos anteriores.
2. Hacer que este modelo pueda interactuar con otros parámetros musicales, como son las alturas, los timbres, las dinámicas, etc.
3. Tener flexibilidad de uso para que cualquier músico lo pueda adaptar o modificar de acuerdo a sus necesidades específicas.
4. Ser capaz de manejar todas las técnicas de estructuración rítmica presentes en la música de concierto del siglo XX mencionadas en esta tesis, ya sea mediante la secuenciación de patrones rítmicos, la ejecución directa del instrumento virtual, o la modificación de su código por parte del usuario

### **1.7 PREMISAS DE TRABAJO**

De la exposición realizada en este capítulo se establecen las siguientes premisas de trabajo:

1. Las series aritméticas de Novaro y la división aleatoria de unidades de tiempo son la base de desarrollo del modelo aritmético de representación rítmica
2. Se implementa el modelo de forma que pueda manejar tanto el protocolo MIDI, como el uso de archivos de audio.
3. El software a utilizar es Max/MSP/Jitter.



## CAPÍTULO 2

### MÉTODOS DE GENERACIÓN RÍTMICA

#### 2.1 PRINCIPIOS BÁSICOS

Tanto el modelo de estratos rítmicos de Yeston (capítulo 1), como el uso de series numéricas en Novaro (capítulo 1) plantean una idea común, que es la de ver al ritmo como una serie de estratos. A partir de ello, se plantean los siguientes puntos:

- a. El ritmo se puede analizar como una serie de estratos.
- b. La unión de varios estratos provoca que una jerarquización de los mismos.
- c. Dicha jerarquización es traducida por el metro.
- d. El metro es resultado de los acentos creados por la interacción de los diversos estratos rítmicos.
- e. La acentuación es un fenómeno subjetivo, al estar sujeto a la interacción no sólo del ritmo en sí, sino también de las alturas, la dinámica y en lo general, de todos los elementos que conforman la música.
- f. Los estratos pueden ser representados por series de valores numéricos, independientes de la acentuación, y por esa misma razón, no representan ninguna jerarquización predeterminada.
- g. El uso de valores rítmicos numéricos es una manera eficaz de trasladar las diversas técnicas de elaboración rítmica en la música contemporánea al ámbito informático, ya que por un lado, permite trabajar con listas de datos precisos que pueden ser aritméticamente transformadas para su manejo y desarrollo, además de que se libera de la notación rítmica tradicional que en algunas ocasiones resulta confusa o inadecuada para representar varias de las técnicas rítmicas contemporáneas.

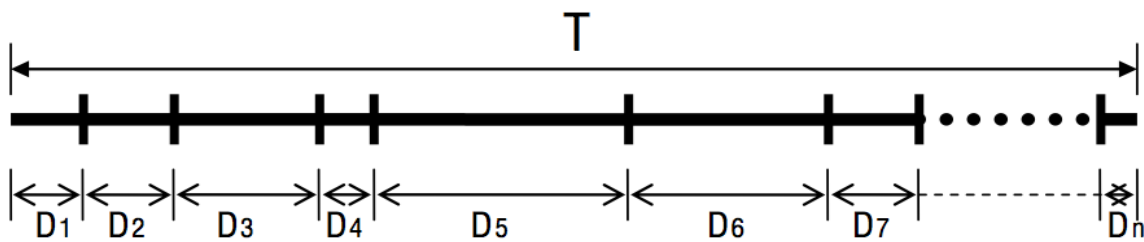
## 2.2 REPRESENTACIÓN MATEMÁTICA Y GRÁFICA

Una vez enunciados los principios básicos sobre los cuales se desarrolla el modelo de generación rítmica, se pasa ahora a su representación numérica. Puesto que se ha decidido trabajar con datos numéricos, el primer paso será formalizar todos los elementos involucrados matemáticamente.

Empecemos por acotar un elemento esencial, que es el tiempo. Para poder trabajar con series numéricas, se debe tener claro que los valores contenidos en las series serán equivalentes a pequeños lapsos de tiempo, que a su vez, una vez generados y conjuntados con valores de otras series, darán un resultado final que se traducirá en un evento rítmico con un número  $n$  de eventos individuales. Estos ocurrirán en un espacio de tiempo total  $T_n$ , donde cada uno de los valores rítmicos individuales durará una fracción del tiempo total,  $D_k$ . Esta fracción de tiempo ( $D_k$ ) será a su vez, la diferencia de los tiempos de ocurrencia ( $T_k$ ) inicial y final del evento rítmico.

$$(1) \quad T_n = \sum_{k=1}^n (D_k) \quad , \quad D_k = T_k - T_{k-1} \quad , \quad n, k \in \mathbb{N}$$

con  $T_{\text{inicial}}=0$ ,  $T_1=D_1$ ,  $T_2=D_1+D_2$ , etc., y  $T_n= D_1+D_2+\dots+D_n= T$



*Fig. 18. Representación gráfica del intervalo de tiempo T*

Sin embargo, para que dichos valores individuales  $D_k$  puedan ser entendidos como un conjunto de valores que forma parte de una serie numérica  $S_n$  de  $n$  valores temporales sucesivos, cada evento  $T_k$  de la serie  $S_n$  será representado como la duración actual más los valores  $D_k$  previamente ocurridos, es decir:

MAX-RHYTHMATIST

$$(2) \quad S_n = \{T_1, T_2, \dots, T_n\} \quad , \quad T_n = \sum_{k=1}^n (D_k) = T$$

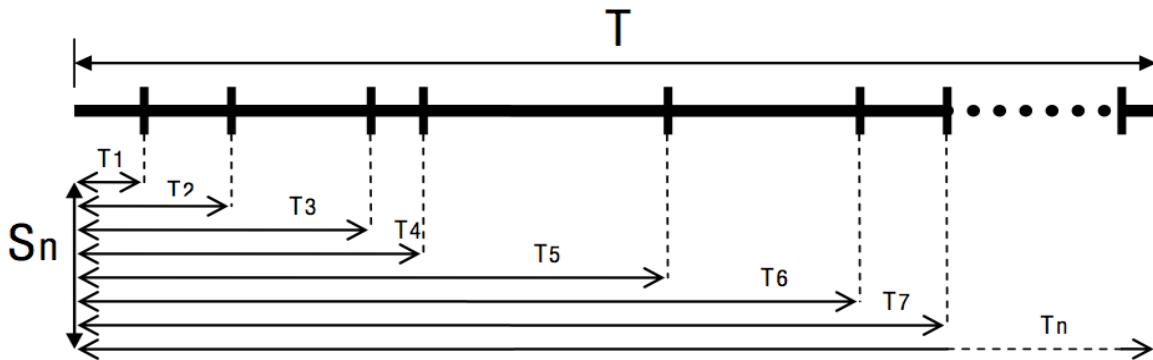


Fig. 19. Representación gráfica de la serie de valores rítmicos  $S_n$ .

De lo cual se deduce que el valor final  $T_n$  de la serie  $S_n$  será siempre igual a la duración total de tiempo  $T$ . Por lo tanto, para simplificar el manejo de las series numéricas será conveniente que consideremos al espacio  $T$  como una **duración total**, y en consecuencia, con un valor predeterminado de 1. Al hacer esto, simplificamos las expresiones previamente mostradas y podemos finalmente definir los elementos de nuestro modelo:

- a.  $T = \text{Duración total} = 1$
- b.  $D_k = \text{Duración de un elemento rítmico individual}$ , cuyos valores son números reales.
- c.  $T_k = \text{Valor sucesivo de duración acumulada (o tiempo de ocurrencia)}$
- d.  $S_n = \text{Serie numérica de tiempos sucesivos de ocurrencia}$

Finalmente, las expresiones finales para definir el modelo general de generación rítmica, así como su representación gráfica, quedan de la siguiente manera:

$$(3) \quad S_n = \{T_1, T_2, \dots, T_n\} \quad , \quad T_n = \sum_{k=1}^n (D_k) = 1 \quad , \quad D \in \mathbb{R}$$

MAX-RHYTHMATIST

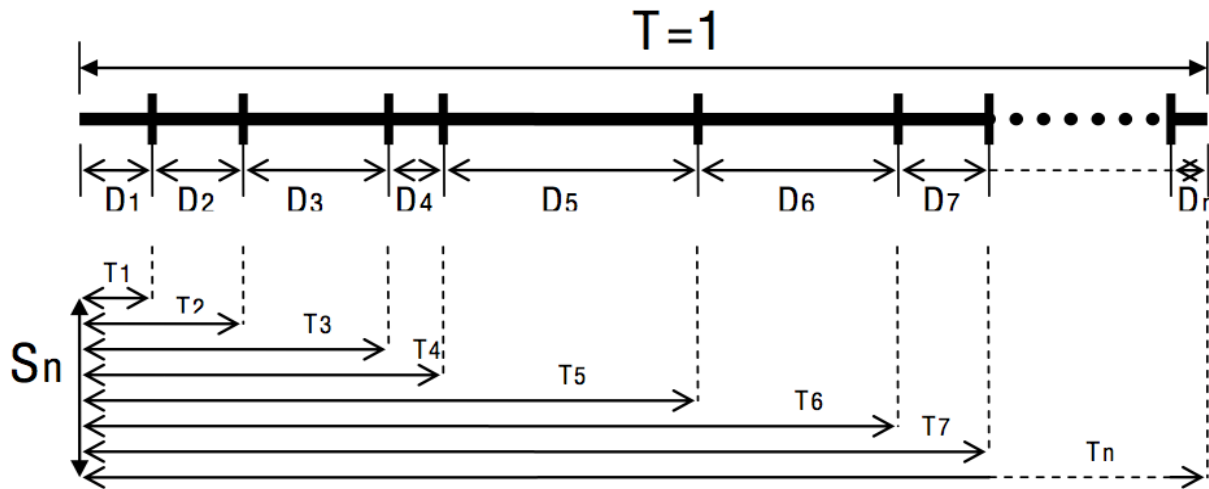


Fig. 20. Representación gráfica del modelo de generación rítmica

## 2.3 CLASIFICACIÓN DE LAS SERIES PARA GENERAR PATRONES

### RÍTMICOS.

Dado que las series segmentan la duración total  $T=1$  en partes, de ello se derivan los siguientes tipos de series:

#### 2.3.1 Series regulares

##### *Series regulares uniformes ( $S_n$ )*

Este tipo de series dividen a  $T$  en  $n$  duraciones  $D$  de igual valor.

$$(4) \quad S_n = \{T_1, T_2, \dots, T_n\}, \quad T_n = \sum_{k=1}^n (kD) = 1, \quad D = \frac{1}{X}, \quad X = n$$

Ejemplo:  $S_{u3} = \{1/3, 2/3, 3/3\} = \{0.3333, 0.6666, 1\}$

<b>Tabla de valores de la serie uniforme <math>S_{u3}</math></b>			
<b>k</b>	<b>1</b>	<b>2</b>	<b>3</b>
<b>T</b>	0.3333	0.6666	1
<b>D</b>	0.3333	0.3333	0.3333

MAX-RHYTHMATIST

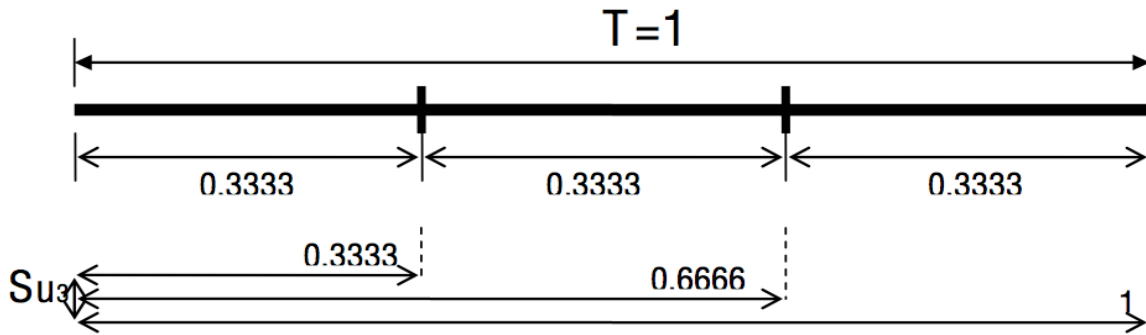


Fig. 21. Representación gráfica de la serie uniforme  $Su_3$

***Series regulares compuestas ( $Sc_n$ )***

Estas series son la unión de dos o más series regulares uniformes, y cuyas duraciones  $D_k$  combinadas dan como resultado un patrón rítmico simétrico.

$$(5) \quad Sc_{a+b} = Su_a \cup Su_b \dots$$

Ejemplo:  $Sc_{(3+5)} = Su_3 \cup Su_5 = \{1/3, 2/3, 3/3\}, \{1/5, 2/5, 3/5, 4/5, 5/5\}$

Para calcular los valores  $D_k$  derivados de la serie, que sumados nos den por resultado  $T=1$ , se tendrán que ordenar los elementos  $T_k$  de ambas series, para posteriormente encontrar las diferencias entre ellos.

$$Sc_{(3+5)} = \{1/5, 1/3, 2/5, 3/5, 2/3, 4/5, 5/5, 3/3\}$$

$$T = (1/5 - 0) + (1/3 - 1/5) + (2/5 - 1/3) + (3/5 - 2/5) + (2/3 - 3/5) + (4/5 - 2/3) +$$

$$(5/5 - 4/5) + (5/5 - 3/3) = 1$$

$$T = 0.2 + 0.1333 + 0.0666 + 0.2 + 0.0666 + 0.1333 + 0.2 = 1$$

<b>Tabla de valores de la serie <math>Sc_{(3+5)}</math></b>							
<b>k</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>
<b>T</b>	0.2	0.3333	0.4	0.6	0.6666	0.8	1
<b>D</b>	0.2	0.1333	0.0666	0.2	0.0666	0.1333	0.2

MAX-RHYTHMATIST

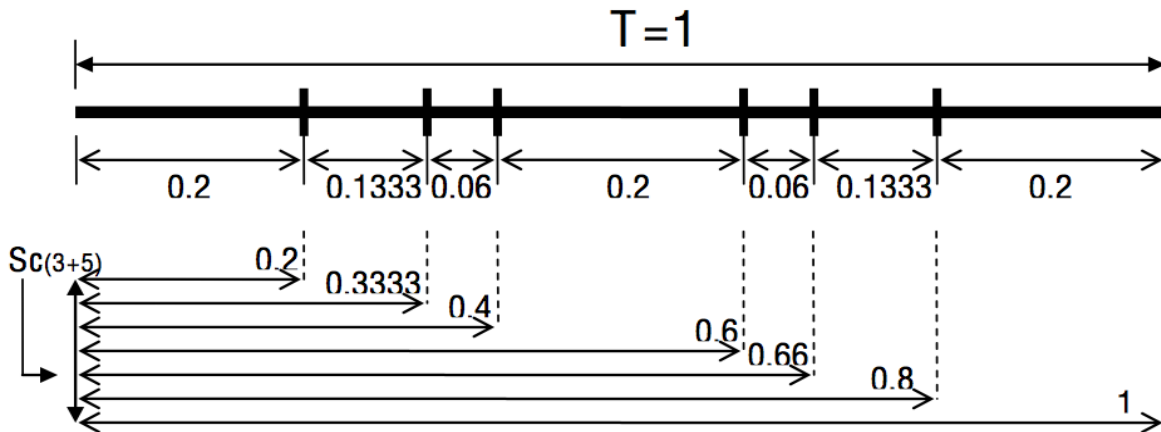


Fig. 22. Representación gráfica de la serie compuesta  $S_{c(3+5)}$

### 2.3.2 Series Irregulares

*Series irregulares de duraciones acumuladas ( $S_{yn}$ )*

Se definió a  $T_k$  como un valor sucesivo de duración acumulada. Si se considera que una serie irregular se basa en una serie de valores que no pueden ser referidos a una unión de series regulares, entonces es posible considerarla como una serie de números aleatorios.

La serie irregular de duraciones acumuladas se define como el ordenamiento de menor a mayor de un conjunto  $n$  de números aleatorios, cada uno de los cuales representará un valor  $T_k$ , y cuyo valor máximo será  $T_n = T = 1$ .

$$(6) \quad S_{y_n} = \{ T_1, T_2, \dots, T_n \} \quad , \quad 0 = T_k \quad , \quad T \in \mathbb{R}$$

Ejemplo: Dados los números aleatorios: 0.491, 0.61, 0.356, 0.848, más el valor máximo 1, luego entonces,  $S_{y_5} = (0.356, 0.491, 0.61, 0.848, 1)$ .

Las diferencias entre los valores  $T_k$  son,

$$T = (0.356 - 0) + (0.491 - 0.356) + (0.61 - 0.491) + (0.848 - 0.61) + (1 - 0.848) = \\ 0.356 + 0.135 + 0.119 + 0.238 + 0.152 = 1$$

MAX-RHYTHMATIST

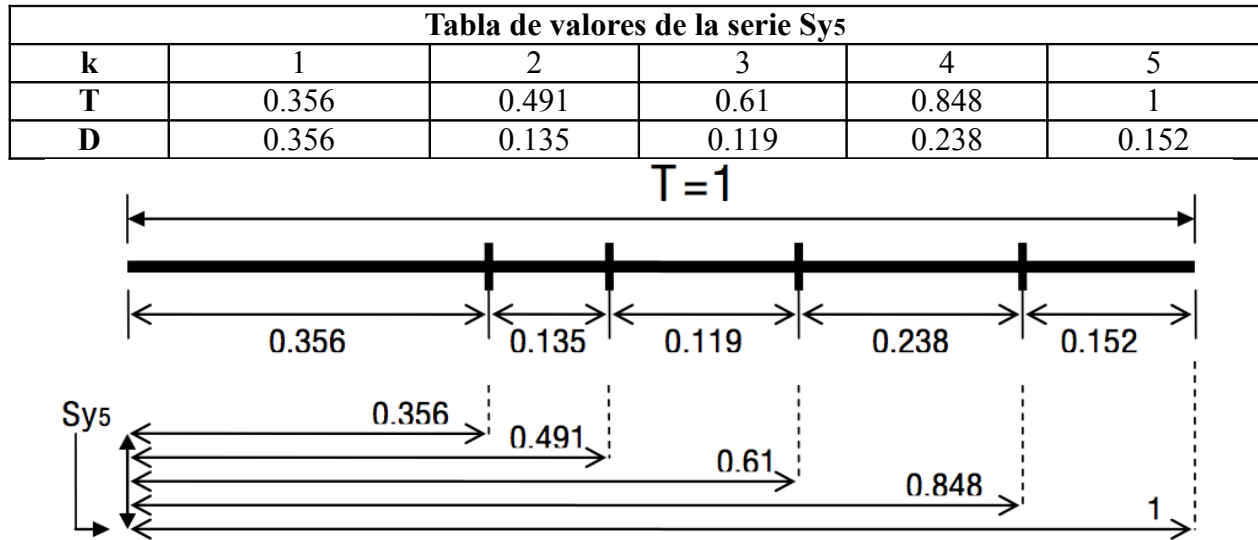


Fig. 23. Representación gráfica de la serie de duraciones acumuladas Sy5

Es interesante notar que a medida que se incrementa la cantidad de números aleatorios para generar la serie, y dado que se están siempre ordenando de menor a mayor, la serie irregular tenderá a parecerse a una serie regular uniforme, ya que las duraciones  $D_k$  serán cada vez más parecidas.

$$(7) \lim_{n \rightarrow \infty} Sy_n = Su_n$$

***Series irregulares de duraciones individuales (Sdn)***

De la misma manera que la serie previamente expuesta, la serie irregular de duraciones individuales se genera a partir de  $n$  números aleatorios, pero a diferencia de aquella, los valores no serán ordenados de mayor a menor, sino que cada uno de ellos representará una duración individual  $D_d$ . Esto tendrá como consecuencia que la suma de valores generados dará un valor  $T$  distinto a 1, por lo que para expresar los valores  $T_k$  y  $D_k$  definitivos se tendrá que dividir cada valor aleatorio entre su suma total ( $D_d/T_d$ ).

**MAX-RHYTHMATIST**

$$(8) \quad Sd_n = \{T_1, T_2, \dots, T_n\}, \quad T_n = \sum_{k=1}^n (D_k) = 1, \quad D_k = \frac{D_d}{T_d}, \quad D_d \in \mathbb{R}$$

Ejemplo: Dados los números aleatorios: 0.491, 2.61, 0.356, 3.848, se tendrá lo siguiente:

$$T_d = 0.491 + 2.61 + 0.356 + 3.848 = 7.305$$

Dividiendo cada valor entre el total  $T_d$  obtendremos:

$$T = 0.0672 + 0.357 + 0.048 + 0.526 = 1, \text{ y por lo tanto}$$

$$Sd_4 = \{0.0672, 0.4242, 0.4722, 1\}$$

<b>Tabla de valores de la serie Sd4</b>				
<b>k</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
<b>T</b>	0.0672	0.4242	0.4722	1
<b>D</b>	0.0672	0.357	0.048	0.526

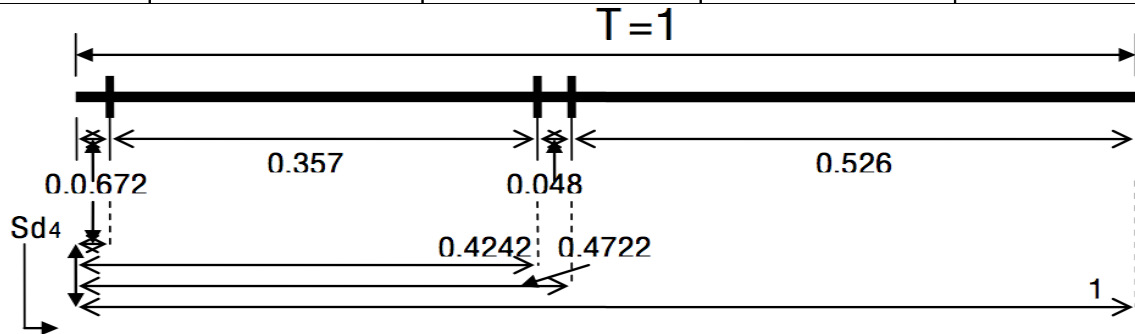


Fig. 24. Representación gráfica de la serie de duraciones individuales  $Sd_4$

A diferencia de la serie irregular expuesta en el punto anterior, la serie irregular de duraciones individuales nunca tenderá a hacerse uniforme, ya que sin importar la cantidad de números aleatorios generados, estos siempre serán diferentes entre sí. Lo que es más, entre más grande sea el rango de números aleatorios disponibles para generar la serie, mucho más irregular será, al haber mayor diferencia entre duraciones individuales.

### **2.3.3 Series transformadas (Snt)**

Se considerará una serie transformada a toda aquella serie derivada de la modificación de cualquier serie regular o irregular por medio de una función  $F(T_{ik})$ , cuya variable independiente equivale a los



**MAX-RHYTHMATIST**

valores T iniciales (Ti), y que como resultado dé una serie de valores acelerados y/o desacelerados (Tf) con respecto a los valores originales. El valor de la pendiente en cada punto de la función elegida representará la velocidad instantánea, por lo que un cambio en la pendiente representará un incremento o un decremento de velocidad que se traducirá en una aceleración o desaceleración del patrón rítmico expresado por la serie. Finalmente, puesto que el tiempo total se puede ver afectada por la transformación, todos los valores Tf se dividirán entre el valor máximo obtenido para mantener la igualdad T=1.

$$(9) \quad S_{n_t} = \{Tf_1, Tf_2, \dots, Tf_n\}, \quad Tf_k = F \frac{(Ti_k)}{Tf_n}, \quad S_n = \{Y_1, Y_2, \dots, Y_n\}$$

Ejemplo: Dada la serie  $S_{u3} = 0.3333 + 0.6666 + 1$ ,

y la función  $Tf_k = (Ti_k)^2$ ,

se obtiene:

<b>Tik</b>	<b>Tft</b>
0.3333	0.1111
0.6666	0.4444
1	1

Puesto que  $Tfn=1$ ,

$S_{tu3} = \{0.1111, 0.4444, 1\}$

<b>Tabla de valores de la serie transformada <math>S_{tu3}</math></b>			
<b>k</b>	<b>1</b>	<b>2</b>	<b>3</b>
<b>T</b>	0.1111	0.4444	1
<b>D</b>	0.1111	0.3333	0.6666

MAX-RHYTHMATIST

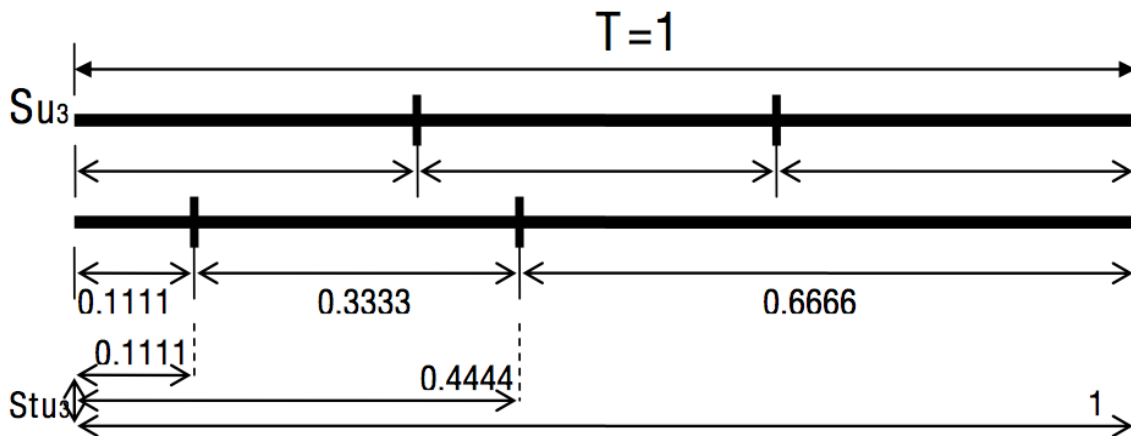


Fig. 25. Representación gráfica de la serie  $Su_3$  y su transformación  $Stu_3$

### 2.3.4 La Serie Rítmica Combinada (SRCn)

La serie combinada fue el resultado de la descripción del modelo aritmético en el punto 4.2, definida en lo general por las expresiones matemáticas en (3), y representada gráficamente en la figura 19.

Una vez definidos los distintos tipos de series regulare e irregulares con o sin transformación, se puede redefinir a la serie combinada  $SRC_n$  como una unión de dos o más series de distinto tipo:

$$(\cdot) \quad SRC_n = Su_a \cup Sc_b \cup Sy_c \cup Sd_d \cup Sn_t \dots$$

Esta serie tendrá como característica el contener tanto elementos que presentan cierta regularidad o simetría, así como elementos carentes de cualquier relación.

Ejemplo:  $SRC_7 = Su_3 \cup Sy_5 =$

$$\{0.3333, 0.6666, 1\} \cup \{0.356, 0.491, 0.61, 0.848, 1\} =$$

$$\{0.3333, 0.356, 0.491, 0.61, 0.6666, 0.841, 1\}$$

Tabla de valores de la serie $SRC_7$							
k	1	2	3	4	5	6	7
T	0.3333	0.356	0.491	0.61	0.6666	0.841	1
D	0.3333	0.0227	0.135	0.119	0.0566	0.1744	0.159

*MAX-RHYTHMATIST*

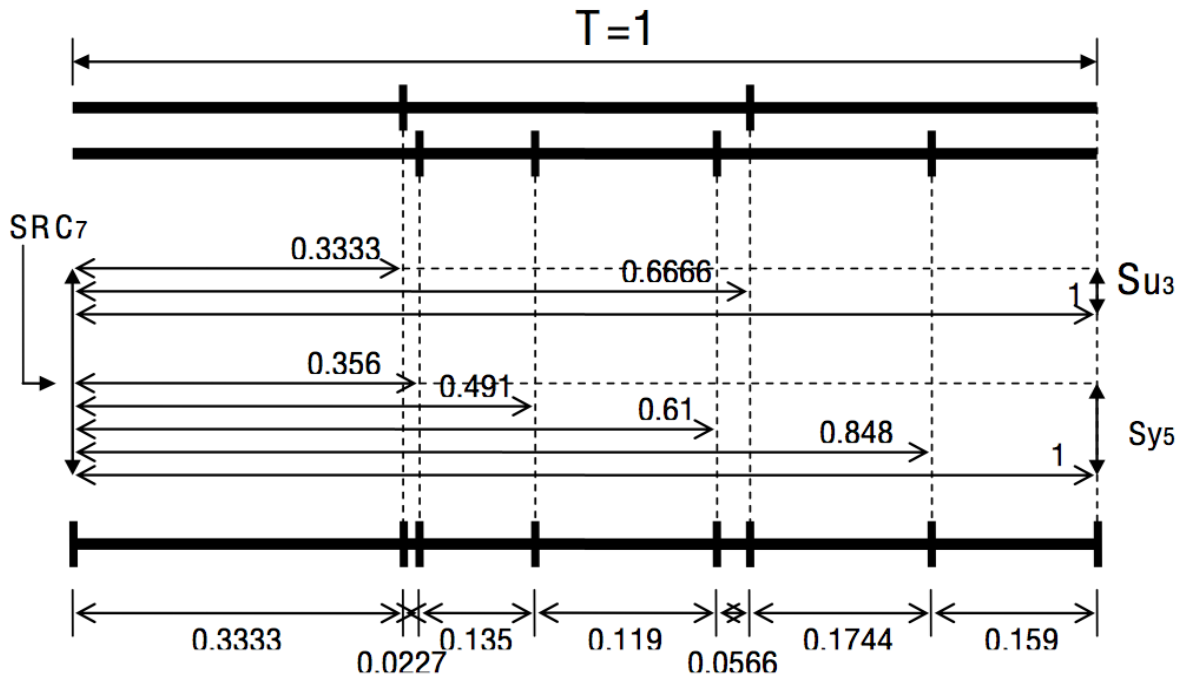


Fig. 26. Representación gráfica de la serie SRC7.

## CAPÍTULO 3

# IMPLEMENTACIÓN INFORMÁTICA DE LOS MÉTODOS DE GENERACIÓN RÍTMICA

Una vez que el modelo de representación rítmica está definido y que se utilizarán diversos tipos de series para la generación rítmica se pasa al problema de su implementación en Max MSP. En la mayor parte de la programación se utilizó el paquete L objects, objetos informáticos diseñados por Peter Elsea en la Universidad de California<sup>28</sup>. Dichos objetos permiten el manejo, transformación e interacción de listas de datos, lo cual es muy útil dada la cantidad de información en forma de lista que se necesita manejar para la creación de las series de valores rítmicos.

### **3.1 IMPLEMENTACIÓN DE LAS SERIES REGULARES SIMPLES Y COMPUESTAS.**

Para la creación de series regulares se utilizaron esencialmente los siguientes objetos:

- a) *Counter*. Para contar desde 1 hasta un valor X, previamente introducido
- b) *Metro*. Para accionar el objeto counter
- c) Operador aritmético  $/$ . Para dividir los valores de counter entre su valor máximo, para obtener los valores de duración acumulada  $T_k$ , resultado de dividir los valores de counter entre el valor máximo introducido
- d) *Lin*. L object que recibe los valores  $T_k$  provenientes del operador aritmético para crear una lista. Cabe decir que el proceso de generación de una serie regular puede repetirse tantas veces se quiera con el número de elementos deseados, por lo que *Lin* puede almacenar un número indeterminado o arbitrario de series regulares uniformes.

<sup>28</sup> [http://www.maxobjects.com/?v=libraries&id\\_library=29](http://www.maxobjects.com/?v=libraries&id_library=29) "Base de datos de objetos externos de Max/MSP"

## MAX-RHYTHMATIST

- e) `Lsort`. Otro L object que ordena de menor a mayor los valores de `Lin`, especialmente útil cuando se introduce más de una serie regular uniforme.
- f) `Lunique`. L object que elimina los valores iguales y que da como resultado la serie regular compuesta final.
- g) `Unlist`. L object que permite enviar los valores de una lista uno por uno, y que sirve para restar los valores  $T_k$  de la serie final por medio de un operador aritmético  $-$ , para obtener con ello los valores de duración individual  $D$ .

Finalmente, se vuelve a utilizar un objeto `Lin` para reagrupar los valores obtenidos a manera de una sucesión de valores no acumulada.

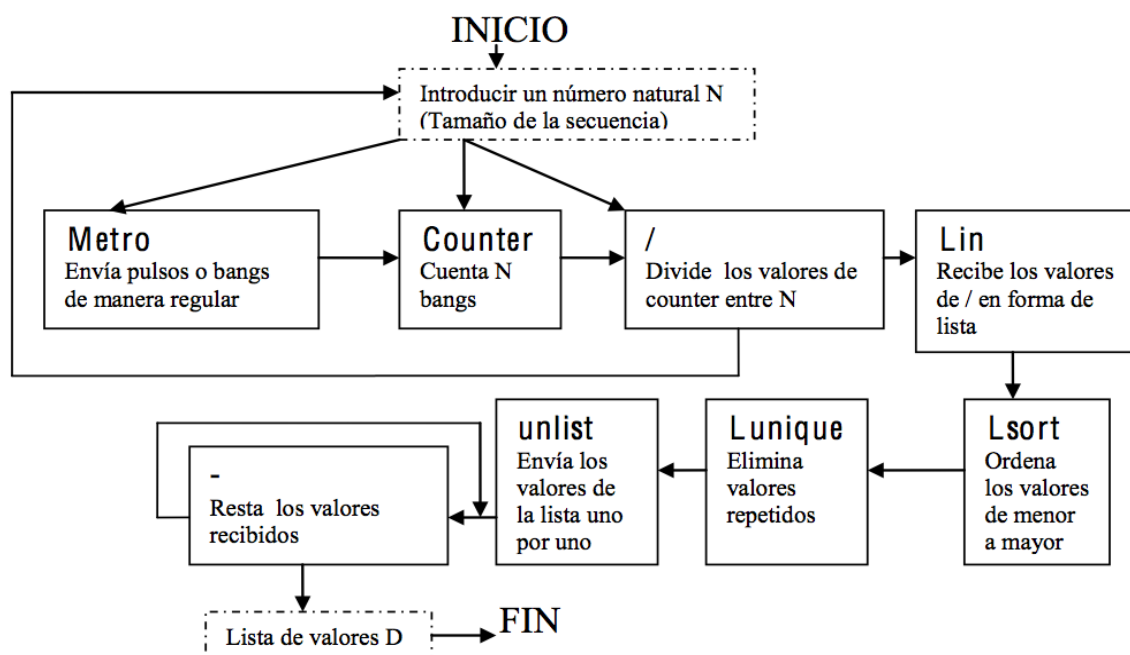


Fig. 27. Implementación informática para generar series regulares.

## **3.2 IMPLEMENTACIÓN DE LAS SERIES IRREGULARES.**

### **3.2.1 Implementación de Series Irregulares de duraciones acumuladas.**

Para esta implementación se usaron, principalmente, los siguientes objetos:

### MAX-RHYTHMATIST

- a) `Random`. Utilizado con un valor predeterminado de 1000, para generar números aleatorios entre 0 y 999.
- b) `Lin`. Para almacenar en forma de lista los valores recibidos de `random`.
- c) `Lsort`. Para ordenar de menor a mayor los valores de `Lin`
- d) `Ldiv`. L object usado para dividir todos los valores de la lista obtenida de `Lsort` entre el valor máximo predeterminado, de manera que el resultado sea una lista de valores ordenados  $T_k$ , cuyo valor máximo sea 1, es decir  $T_n$ .
- e) `Unlist`. Usado en combinación con el objeto `-`, para obtener las diferencia entre valores, y con `Lin` para reagrupar los valores a manera de duraciones individuales `D`.

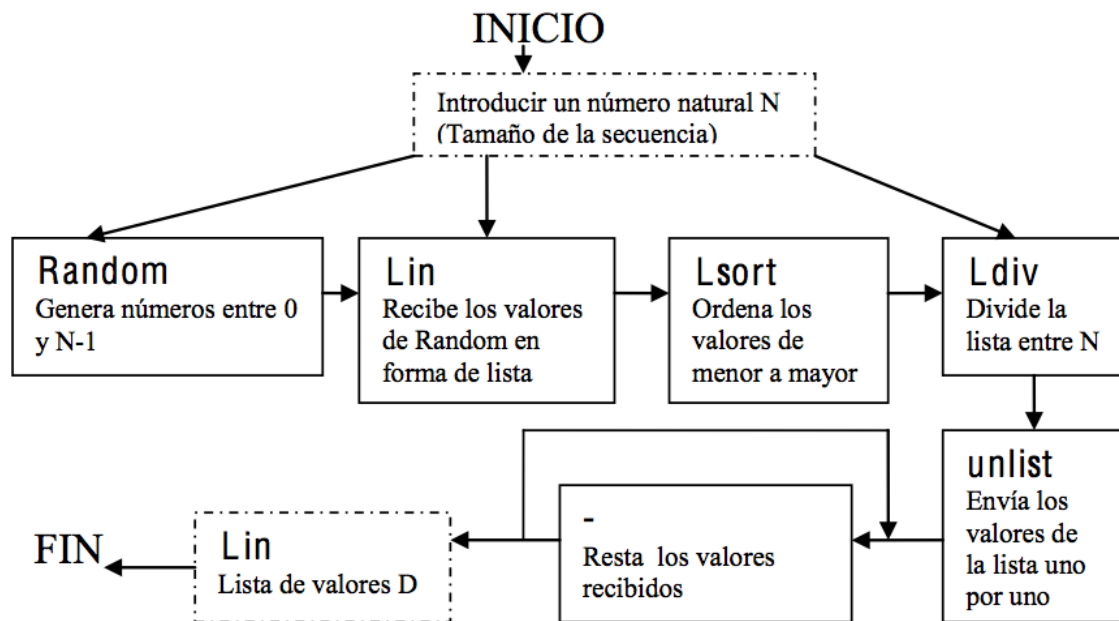


Fig. 28. Implementación informática para generar series irregulares de duraciones acumuladas.

### **3.2.2 Implementación de Series Irregulares de duraciones individuales.**

De manera similar al submódulo previamente explicado, en esta implementación nuevamente el objeto `Random` genera una serie de números aleatorios predeterminados entre 0 y 999. Sin

### MAX-RHYTHMATIST

embargo, a diferencia del submódulo anterior, en este los valores no son ni ordenados, ni alterados, ya que por sí mismos representan duraciones individuales  $D$ .

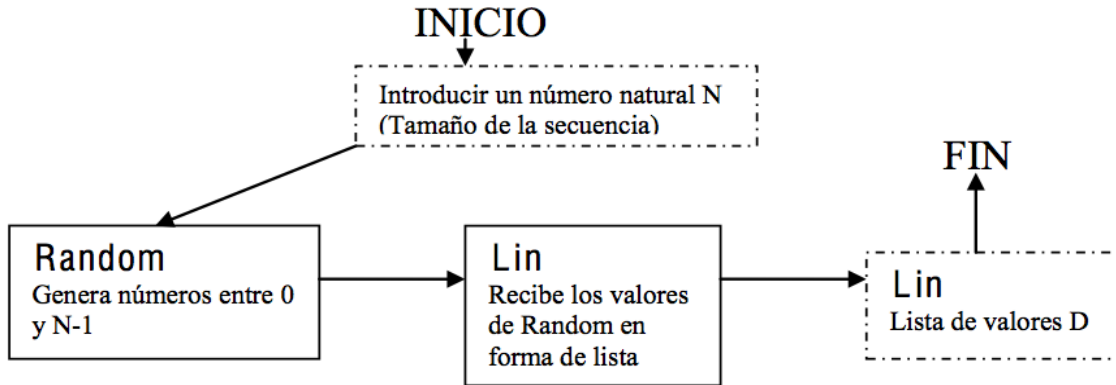


Fig. 29. Implementación informática para generar series irregulares de duraciones individuales.

### **3.3 IMPLEMENTACIÓN DE LAS SERIES TRANSFORMADAS.**

Para tener una visión clara de la transformación se consideró, en primera instancia, la relación entre la gráfica de los valores de duración acumulada ( $T_{ok}$ ) de la serie original contra los valores transformados ( $T_{k'}$ ) de la nueva serie.

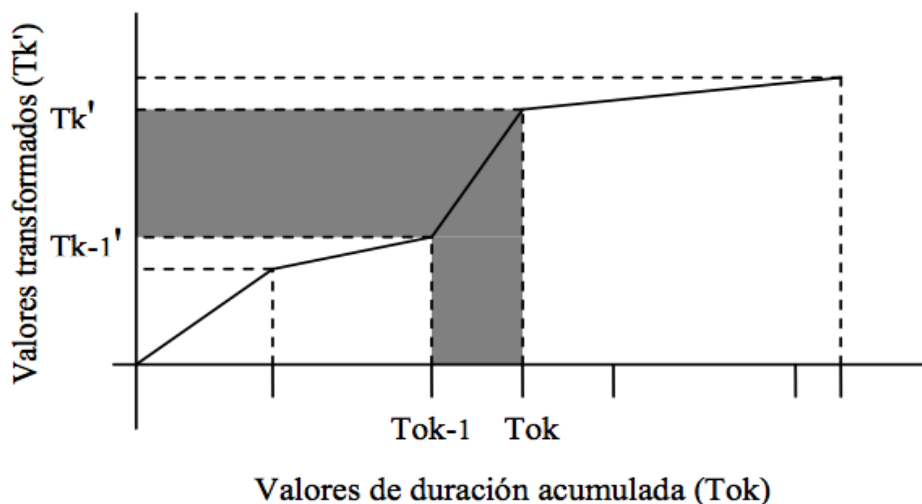


Fig. 30. Gráfica de valores de duración acumulada originales contra valores transformados

La curva resultante muestra los cambios de velocidad que se aplican a la serie original, por lo que la

### MAX-RHYTHMATIST

pendiente de la gráfica nos da como resultado el inverso la velocidad instantánea (i.e. la dilatación o expansión de la escala de tiempo) en cada punto evaluado, de ahí que la derivada de la función  $F(T_k)$  nos muestre los cambios en la pendiente y por tanto la velocidad ( $1/V$ ):

$$\frac{d F(T_k)}{d T_k} = \frac{T'_k - T'_{k-1}}{T_{ok} - T_{ok-1}} = \frac{1}{V}$$

De ahí que para obtener los valores de duración individual ( $Dtr$ ) de la serie transformada ( $Str$ ) se tenga que:

$$T'_k - T'_{k-1} = \frac{T_{ok} - T_{ok-1}}{V} = \frac{D_k}{V} = D'_k$$

Dado que las series se obtienen de la sumatoria de sus valores  $D_k$ , luego entonces, para obtener los valores  $T_k$  de la serie transformada ( $Str$ ), bastará con dividir los valores  $T_{ok}$  de la serie original entre la velocidad instantánea para cada uno de ellos:

$$(11) \quad S_{tr} = \{T_1, T_2, \dots, T_n\} \quad , \quad T_k = \sum_{k=1}^n \left( \frac{T_{ok}}{V_k} \right)$$

Gráficamente, esto muestra más claramente el cambio a realizar, ya que una línea descendente representará una desaceleración del patrón rítmico; una línea horizontal, la misma velocidad; y una línea ascendente, una aceleración.



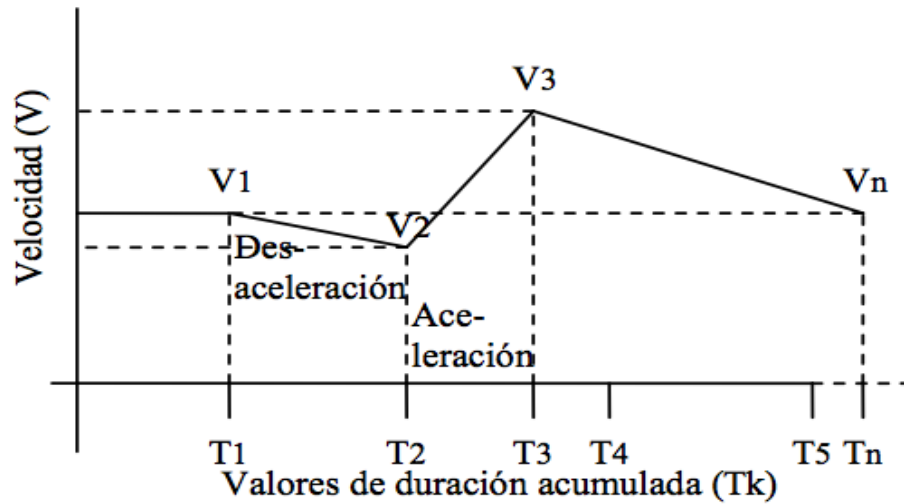


Fig. 31. Gráfica de valores de duración acumulada originales contra el cambio de velocidad

Para transformar una serie se interconectaron las implementaciones previas, es decir, los módulos de series regulares e irregulares a:

- a) Objeto `graphic`, sobre el cual se puede trazar una curva, los valores de la serie recibida son mapeados a través de la gráfica trazada, dando como resultado una serie transformada.

Posteriormente se utilizaron los siguientes objetos:

- b) `Lin`. Para agrupar los valores  $V_n$  obtenidos del objeto `Graphic`.
- c) `Ldiv`. Para dividir los valores de la serie original entre los de la serie  $V_n$ .

Finalmente, para mantener la identidad  $T=1$ , que sale afectada incluso cuando  $V_n=$ constante, se hace pasar la serie transformada por:

- d) Objeto `Lscale`. para cambiar la proporción de una lista de datos. Al poner el valor de 1 como predeterminado en dicho objeto la serie transformada contendrá valores  $T_k$  entre 0 y 1.
- e) `Unlist`. Para sacar los nuevos valores  $D$ , en conjunción con el objeto `-` y `Lin`.

## MAX-RHYTHMATIST

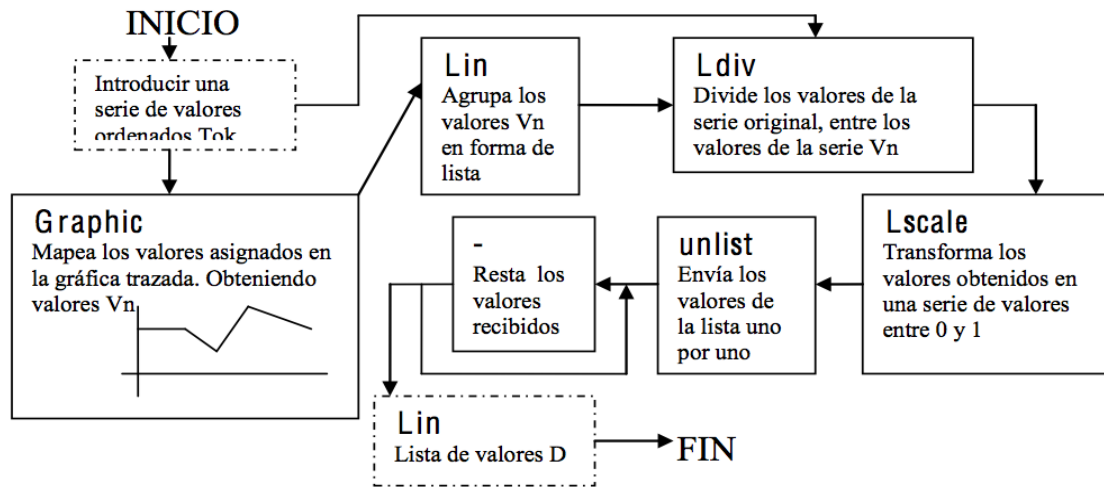


Fig. 32. Implementación informática para generar series transformadas.

### **3.4 GENERACIÓN DE SECUENCIAS MIDI Y AUDIO**

#### **3.4.1 Integración de los submódulos de generación rítmica en un módulo general.**

Una vez lograda la implementación de los diversos métodos de generación de ritmo, cada uno de ellos fue colocado en submódulos dentro de un módulo general.

Las acciones realizadas en el módulo general son las siguientes:

- Recibe las listas de valores de duración individual ( $D_k$ ) provenientes de cualquiera de los generadores.
- Las listas son pasadas a través de un objeto `Lscale` para ajustarlas a una duración en milisegundos que es decidida por el usuario.
- Una vez transformadas las listas en valores en milisegundos son enviadas a los submódulos MIDI y audio.
- Se tienen a su vez dos objetos `matrixctrl`, que son dos matrices para controlar los subpatches MIDI y audio respectivamente. Cada vez que se enciende un punto de la matriz, esta envía la información del canal o muestra de audio usado, y el patrón rítmico a utilizar.

## MAX-RHYTHMATIST

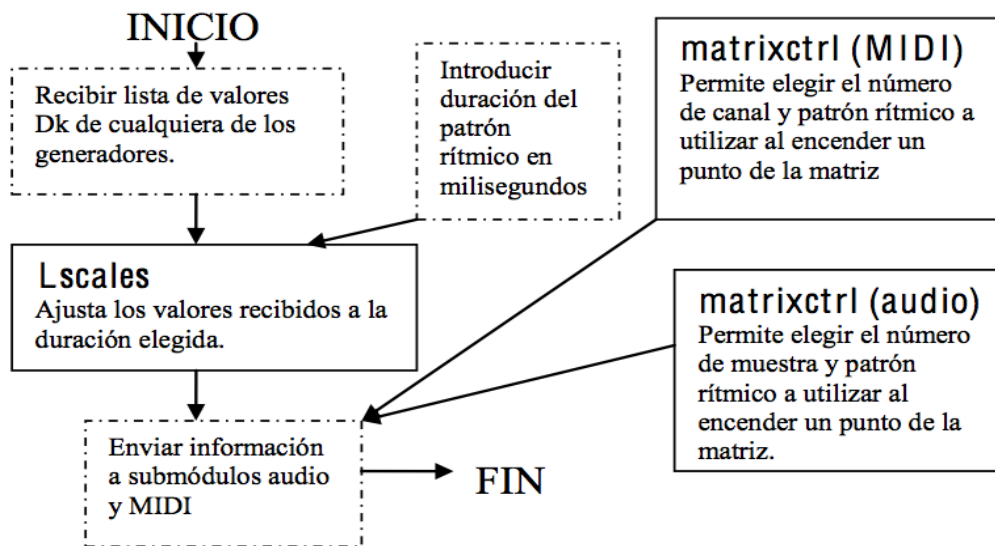


Fig. 33. Diseño del módulo principal.

### 3.4.2 Submódulos MIDI

Se crearon 16 submódulos MIDI, uno para cada canal. Las acciones realizadas por estos submódulos son las siguientes:

- Reciben las listas enviadas por el módulo general
- Las listas se almacenan en el objeto `Text` en el orden recibido
- Al ser accionado el objeto `Text` por el objeto `matrixctrl` del módulo principal, este envía la lista seleccionada a un objeto `Unlist`.
- Cada valor salido del objeto `Unlist` es enviado a un objeto `Random`, que genera un valor aleatorio que representa el número o altura de la nota.
- Tanto el objeto `Unlist` como el `Random` se conectan al objeto `Makenote`, que genera una nota MIDI con el número de nota enviado por `Random` y la duración enviada por `Unlist`. Una vez que transcurre la duración enviada, se genera un `note-off` que acciona nuevamente al objeto `Unlist` para enviar un nuevo valor. Una vez que se termina de leer la lista de valores

## MAX-RHYTHMATIST

seleccionada, esta se vuelve a iniciar y no se detiene hasta que se apaga el punto seleccionado en la matriz, o bien, se selecciona otro punto para enviar una nueva lista.

- f) Finalmente, el objeto `Makenote` se conecta tanto al objeto `Noteout`, para poder escuchar la nota a través del sintetizador interno de la computadora o al dispositivo MIDI conectado a la misma, así como al objeto `Detonate` (presente en el módulo principal), para almacenar la secuencia de sonidos generada de manera gráfica y que posteriormente pueda ser grabada en un archivo MIDI.

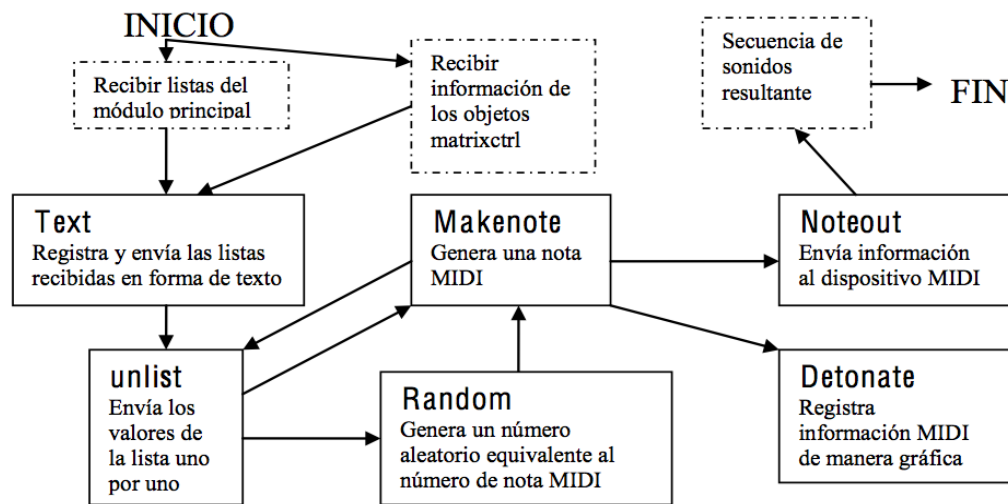


Fig. 34. Diseño de los submódulos MIDI.

### 3.4.3 Submódulos de Audio.

Se crearon 16 submódulos Audio, uno para cada muestra de audio, de manera que hubiera una equivalencia entre el número de muestras de audio que se pudieran usar simultáneamente en la matriz de audio, y el número de canales MIDI usados en la otra matriz. Las acciones realizadas por estos submódulos son las siguientes:

- Reciben las listas enviadas por el módulo general
- Las listas se almacenan en el objeto `Text` en el orden recibido

### MAX-RHYTHMATIST

- c) Al ser accionado el objeto `Text` por el objeto `matrixctrl` del módulo principal, este envía la lista seleccionada a un objeto `Unlist`.
- d) Cada valor salido del objeto `Unlist` es enviado a un objeto `Random`, que genera un valor aleatorio que representa la velocidad en que será reproducida la muestra. El valor de 1 representa la velocidad normal, 0.5 la mitad de la velocidad, y 2 el doble.
- e) El objeto `Random` se conecta al objeto `Groove`, que sirve para reproducir la muestra de audio seleccionada.
- f) El objeto `Unlist` se conectan al objeto `Makenote`, que genera una nota MIDI de duración igual al valor enviado por `Unlist`. Una vez que transcurre la duración enviada, se genera un `note-off` que acciona nuevamente al objeto `Unlist` para enviar un nuevo valor. Una vez que se termina de leer la lista de valores seleccionada, esta se vuelve a iniciar y no se detiene hasta que se apaga el punto seleccionado en la matriz, o bien, se selecciona otro punto para enviar una nueva lista.
- g) El objeto `Makenote` también se conecta al objeto `Groove` para detener la reproducción de la muestra al mandar una velocidad de nota igual a cero (`note-off`).
- h) Finalmente, el objeto `Groove` se conecta tanto al objeto `Dac`, para convertir la señal digital a una señal analógica que pueda ser escuchada a través de bocinas, así como al objeto `Sfrecord` (presente en el módulo principal), para grabar la secuencia de audio generada en un archivo AIFF.

## MAX-RHYTHMATIST

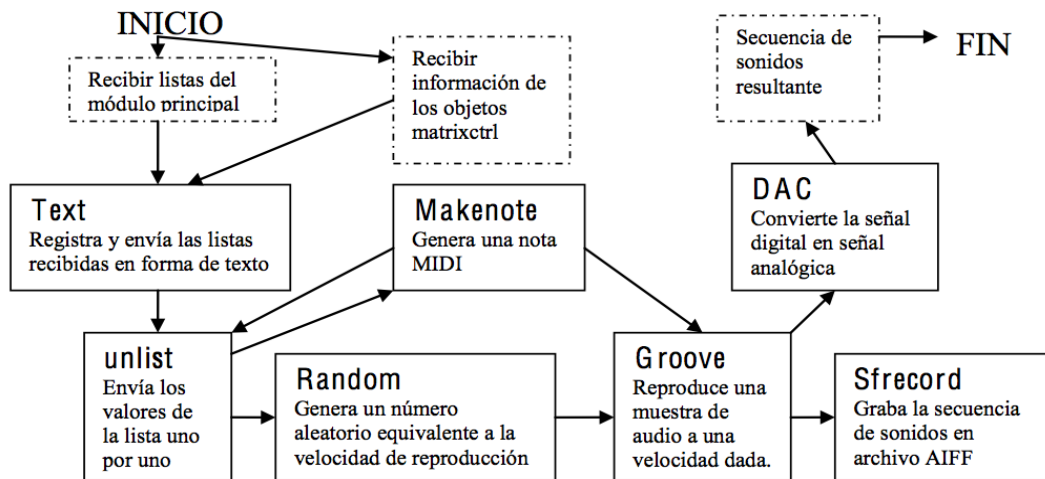


Fig. 35. Diseño de los submódulos audio.

### **3.5 EJECUCIÓN DE PATRONES RÍTMICOS MIDI POR MEDIO DE UN CONTROLADOR EXTERNO**

Para poder tener un instrumento compuesto, de acuerdo a la definición de Schnell y Battier (Schnell y Battier, 2002, Págs. 138-142), discutida en el primer capítulo, además de la posibilidad de secuenciar los patrones, también se incluyó la posibilidad de manipular los patrones rítmicos a través de un controlador MIDI externo. Esto permitiría que el usuario pudiera disparar un patrón rítmico cada vez que presionara una tecla.

#### **3.5.1 Descripción de módulo externo de ejecución**

El módulo externo de ejecución está adaptado para que, al conectarse un controlador MIDI externo, se pueda ejecutar un patrón rítmico al presionar cualquier tecla. Se tomó al controlador MIDI UMX25 marca Behringer<sup>29</sup> para hacer esta adaptación; la razón para elegir a este controlador es por que era muy austero, al tener pocos botones, deslizadores y perillas, y cuenta sólo con 25 teclas, lo cual ayudaría a que el usuario no estuviera condicionado a tener un controlador más sofisticado para

<sup>29</sup> <http://www.behringer.com/ES/Products/UMX25.aspx> "Página de productos Behringer".

### MAX-RHYTHMATIST

poder utilizar el módulo externo. De cualquier modo, el usuario podría modificar el código para adaptarlo al controlador MIDI que esté utilizando.

El módulo está dividido en 2 partes o "layers", uno que abarca las primeras 12 teclas, y el segundo para las otras 13, cada layer está conectado a su vez a tres submódulos:

- **Un submódulo aleatorio**, que permite que las alturas asignadas a los patrones rítmicos tengan un grado de aleatoriedad controlada por el usuario.
- **Un submódulo por alturas**, que permite que el usuario asigne notas específicas que serán asignadas a los patrones rítmicos ejecutados.
- **Un submódulo por intervalos**, que permite que el usuario asigne los intervalos que se asignarán al patrón rítmico partiendo de la nota ejecutada.

#### **3.5.2 Descripción del submódulo aleatorio**

En este submódulo, las series almacenadas, provenientes de los generadores rítmicos, pueden ser llamadas una a la vez, las notas asignadas se dan de manera aleatoria, desde una sola nota hasta 127.

Si la tecla se mantiene presionada, sólo se ejecuta el primer valor rítmico del patrón. En cambio, si se suelta la tecla, se ejecuta todo el patrón rítmico una sola vez.

Para lograr lo anteriormente descrito, se siguen los siguientes procedimientos:

- a) Se asigna un canal, un programa (banco de sonido) y el rango de notas por encima de la tecla presionada con las cuales jugará el patrón rítmico; esto se hace para cada "layer", pudiendo dejar los mismos parámetros en ambos.
- b) Se llama un patrón rítmico, el cual ya fue previamente creado con cualquiera de los generadores y almacenado en un objeto `Text`.
- c) Cada vez que se presione una tecla, el objeto `Unlist` enviará una por una las duraciones del

### MAX-RHYTHMATIST

patrón rítmico seleccionado al objeto *Makenote*. Este objeto, a su vez, recibe el resultado enviado por el objeto *+*, que suma al número de nota presionado los números aleatorios provenientes del objeto *Random*, de tal manera que por cada número de nota generado, se va asignando una duración del patrón rítmico. Los dos grupos de datos recibidos por *Makenote* son enviados al objeto *Noteout* para poder escuchar el resultado sonoro via MIDI.

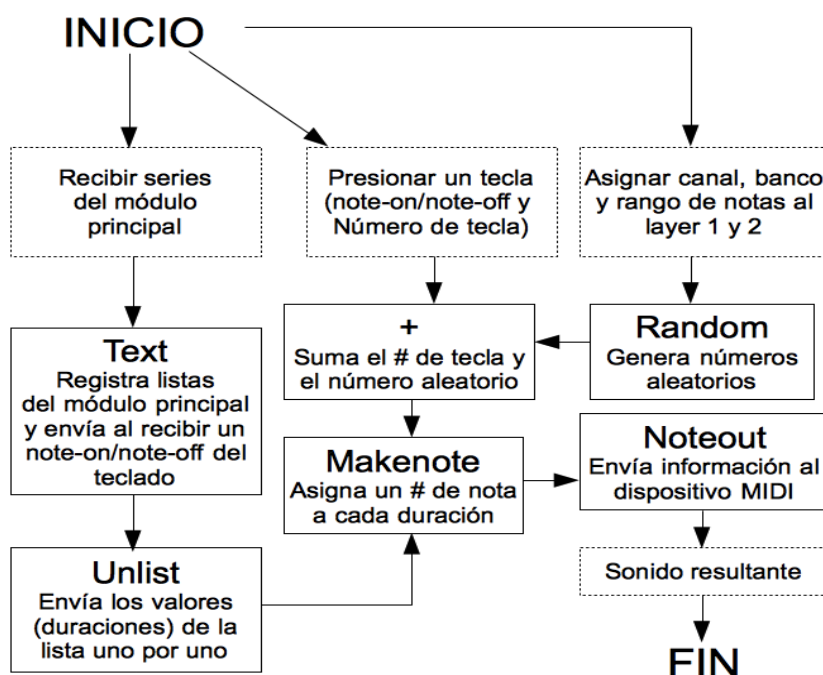


Figura 36. Submódulo aleatorio.

### 3.5.3 Descripción del submódulo por alturas

En este submódulo, el usuario es libre de elegir las notas específicas con las cuales quiera que el patrón rítmico juegue, dichas notas son tocadas aleatoriamente, el usuario tiene la elección de elegir el grado de aleatoriedad, de forma tal que si la cantidad de números aleatorios es mayor que las notas escogidas, entonces a algunas duraciones del patrón rítmico se le pueden asignar silencios.

Para lograr lo anteriormente descrito, se siguen los siguientes procedimientos:



### MAX-RHYTHMATIST

- a) Mediante el objeto `Lin`, se introducen los valores de las notas a ejecutar (número de nota MIDI).
- b) Cada vez que se presiona una tecla, nuevamente el objeto `Unlist` envía al objeto `Makenote` una duración.
- c) Simultáneamente, el objeto `Random`, elige de manera aleatoria una de las notas introducidas por el usuario y la envía al objeto `Makenote`.
- d) El objeto `Makenote` envía al objeto `Noteout` las notas y duraciones recibidas previamente para poder escuchar el resultado via MIDI.

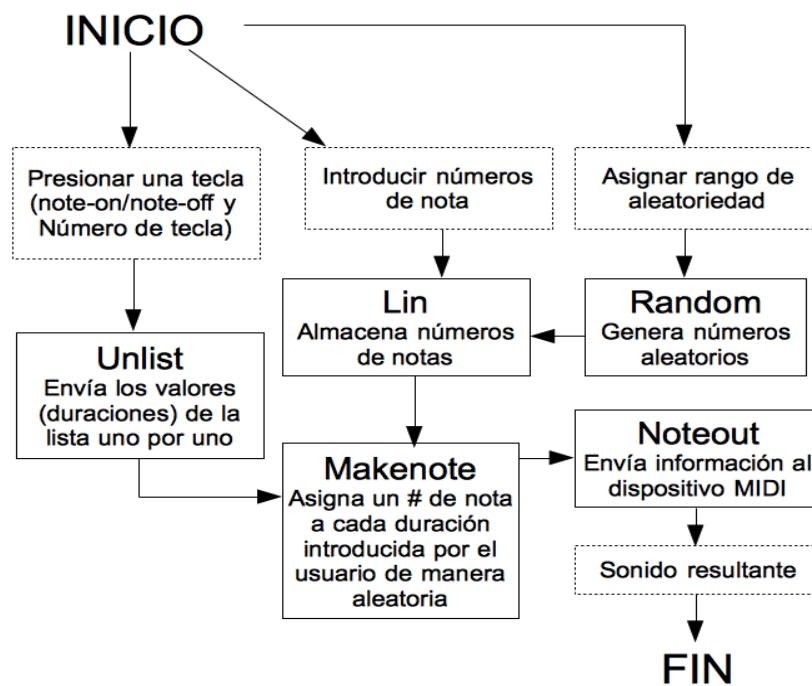


Figura 37. Submódulo por notas.

#### 3.5.4 Descripción del submódulo por intervalos

En este submódulo, el usuario introduce los intervalos, medidos a partir de la nota que se haya presionado, pudiendo estos ser positivos (por encima de la nota) o negativos (por debajo de la nota).

### MAX-RHYTHMATIST

Nuevamente, como ocurrió con el submódulo por notas, existe un grado de aleatoriedad controlado por el usuario, con la posibilidad de insertar silencios si se elige una cantidad de números aleatorios mayor a los intervalos elegidos.

Para lograr lo anteriormente descrito, se siguen los siguientes procedimientos:

- Se introducen los intervalos a utilizar por medio del objeto `Lin`.
- Cada vez que se presiona una tecla, nuevamente el objeto `Unlist` envía al objeto `Makenote` una duración.
- Simultáneamente, el objeto `Random`, elige de manera aleatoria uno de los intervalos introducidos por el usuario y lo envía al objeto `+` para sumarlo al número de nota que se haya presionado, y el resultado se envía al objeto `Makenote`.
- El objeto `Makenote` envía al objeto `Noteout` las notas y duraciones recibidas previamente para poder escuchar el resultado via MIDI

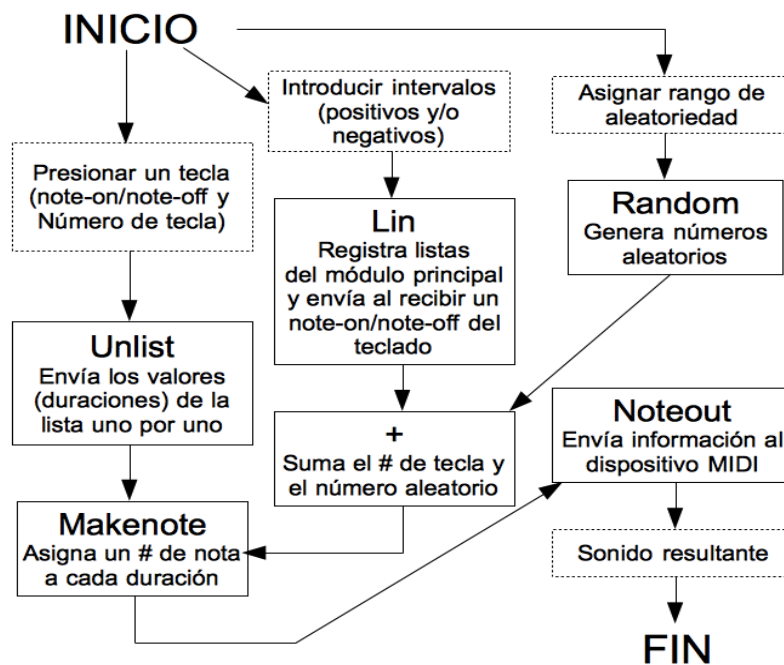


Figura 38. Submódulo por intervalos

### 3.5.5 Ejecución de muestras de audio con el módulo externo de ejecución

Dentro de la programación se incluyó una versión *alfa*<sup>30</sup> para lograr que los procedimientos aplicados para ejecutar patrones rítmicos via MIDI pudieran funcionar con muestras de audio. La razón principal por la que no se incluyó en la interfaz de usuario fue la producción de "clicks" de manera ocasional. Estos "clicks" se producen sobre todo cuando se tocan notas con mayor rapidez que la duración total del patrón rítmico, provocando una interrupción súbita del patrón rítmico en ejecución, y por consecuencia, un click. A pesar de experimentarse con diferentes envolventes de amplitud para evitar este problema, la producción de clicks continuó, por lo que se decidió dejar esta opción fuera de la interfaz de usuario. Sin embargo, su inclusión en el código es importante por si el usuario quiere experimentar con esto submódulo. Por el momento, el submódulo de audio está conectado al submódulo aleatorio (sólo en el "layer" 2), y se tiene también la opción de transponer aleatoriamente la muestra para generar diferentes alturas sobre el patrón rítmico.

Para lograr lo anteriormente descrito, se siguen los siguientes procedimientos:

- a) Se abre una muestra de audio (se toman por default 2000 ms de dicha muestra) con el objeto `Buffer~`.
- b) A cada duración enviada por el objeto `Unlist` se le restan 30 ms con el objeto `-`, con el fin de evitar "click" entre cada muestra ejecutada dentro de un mismo patrón rítmico.
- c) El resultado de dicha operación aritmética es insertado en un mensaje que ejecuta dos objetos `Line~` de manera alternada, los cuales ejecutan dos muestras por medio de los objetos `Groove~`. Esta alternancia se logra con un objeto `Gate`. El uso de dos objetos para la ejecución de la muestra se debió a los "click" generados con la utilización de uno solo.

---

<sup>30</sup> Algunos equipos de desarrollo utilizan el término **alfa** informalmente para referirse a una fase donde un producto o parte de un producto todavía es inestable, aguarda todavía a que se eliminen los errores o a la puesta en práctica completa de toda su funcionalidad, pero satisface la mayoría de los requisitos.

MAX-RHYTHMATIST

- d) Simultáneamente, un objeto `Sig~` se conecta a cada objeto `Groove~`, este objeto envía el resultado de la resta de los números de nota MIDI ejecutada por el módulo aleatorio menos 60 (la nota más grave del "layer" 2) obtenida del objeto `-`, y que pasa por el objeto `Expr` para transformar la diferencia obtenida en un intervalo musical basado en el temperamento igual de 12 sonidos.
- e) Los sonidos generados por los objetos `Groove~` se envían al objeto `Ezdac~` para poder escuchar el resultado.

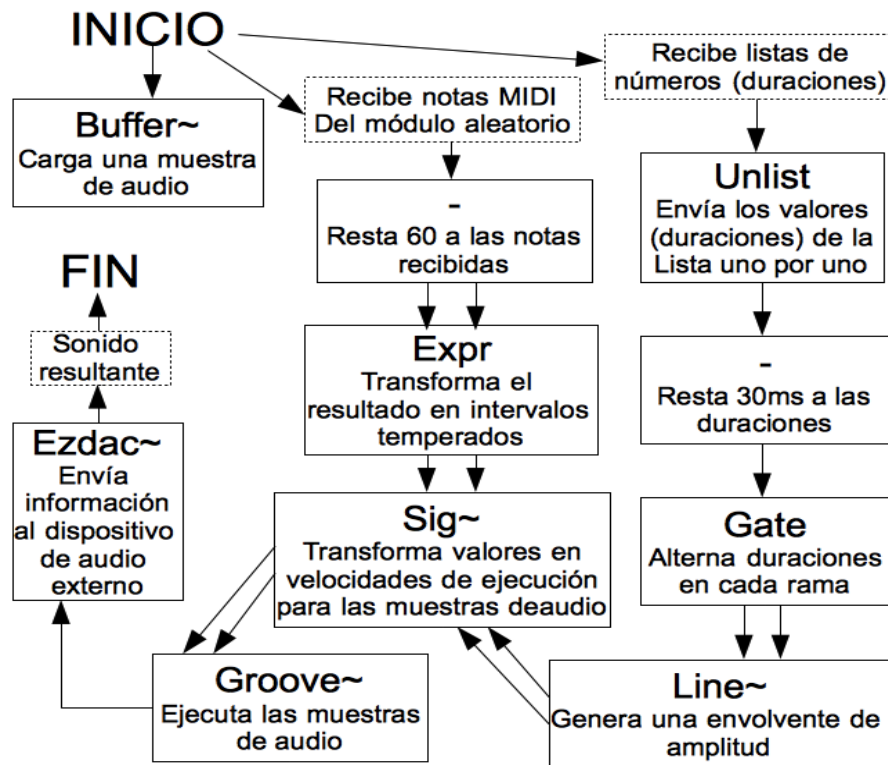


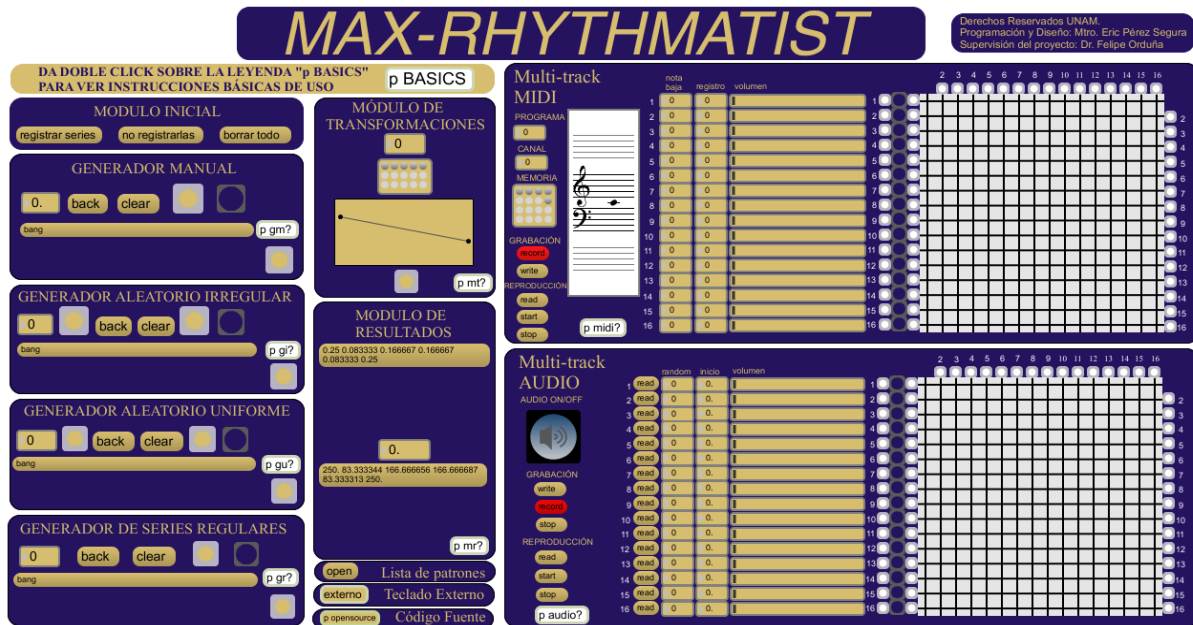
Figura 39. Ejecución de muestras de audio

## CAPÍTULO 4

### LA INTERFAZ DE USUARIO

#### 4.1 ELEMENTOS CONSTITUTIVOS DE LA INTERFAZ DE USUARIO

Se decidió darle el nombre de **Max-Rhythmatist**<sup>31</sup> al patch programado en Max/MSP, nombre con el que se le denominará de aquí en adelante. Dada la cantidad de submódulos y la extensión del módulo principal, se creó una interfaz que agrupara todas las acciones a tomar en una sola pantalla.



Dicha interfaz consta de los siguientes elementos:

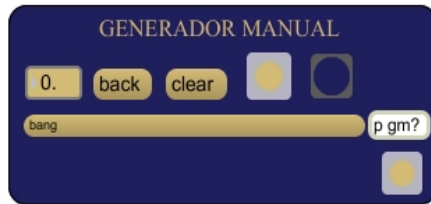
- Un panel inicial
- Un módulo inicial para registrar las series generadas.



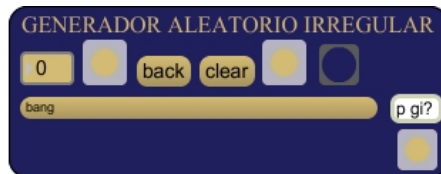
<sup>31</sup> Nombre inspirado en "The Rhythmatist", sobrenombre que el percusionista Stewart Copeland (ex-baterista del grupo The Police) utilizó para la elaboración del soundtrack de la película "Rumble Fish" (1983) de Francis Ford Coppola, y que posteriormente también fue utilizado como título de su disco solista en 1985.

MAX-RHYTHMATIST

- Un generador manual de valores rítmicos.



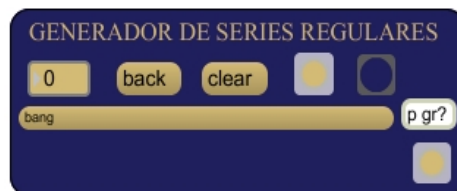
- Un generador aleatorio irregular.



- Un generador aleatorio uniforme.



- Un generador de series regulares.

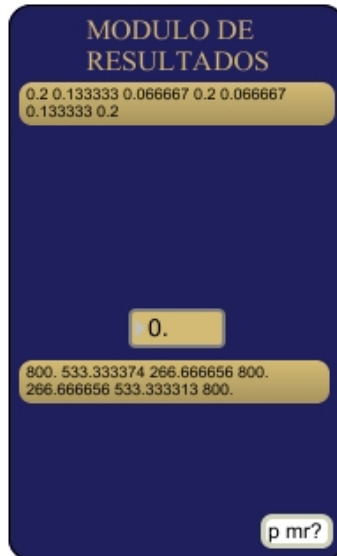


- Un módulo de transformaciones



## MAX-RHYTHMATIST

- Un módulo de resultados.



- Un acceso a la lista de patrones rítmicos almacenados



- Un acceso al módulo externo de ejecución

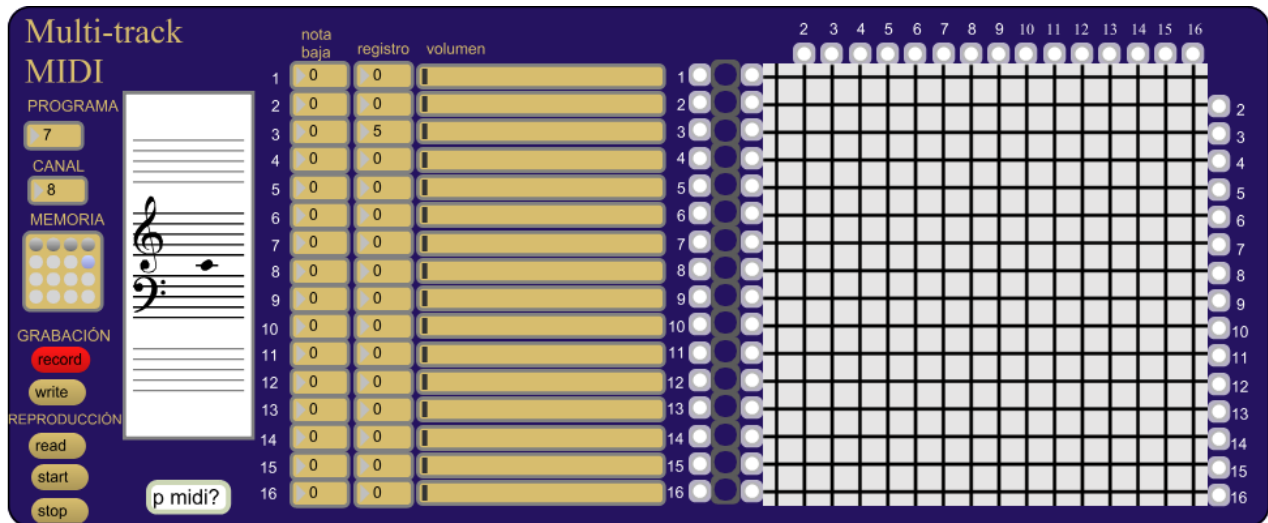


- Un acceso al código fuente o patch principal

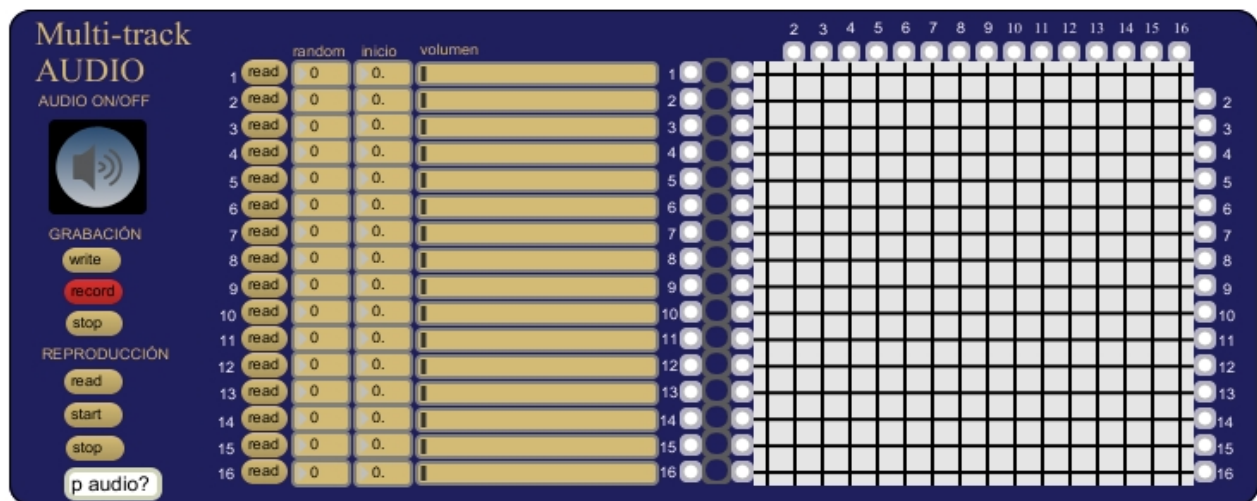


## MAX-RHYTHMATIST

- Un multi-track MIDI



- Un multi-track audio



## **4.2 COMENZANDO A UTILIZAR EL MAX-RHYTHMATIST**

### **4.2.1 Instalación y guía rápida**

Dentro de la carpeta que contiene al programa se encuentra una Guía Rápida de Instalación y Prueba, que se colocó con la finalidad de que el usuario pueda probar el sistema en pocos minutos antes de entrar de lleno en la exploración del mismo. El texto colocado es el siguiente:



## MAX-RHYTHMATIST

### INSTRUCCIONES DE INSTALACIÓN Y GUÍA RÁPIDA DE USO

- Descargar el demo de Max/MSP/Jitter de la página: <http://cycling74.com/downloads/>
- Instalar
- Abrir la carpeta de Max Rhythmatist.
- Poner los carpetas de los LObjects (objetos externos) de la siguiente manera:
  - Para usuarios de Windows
    - Colocar la carpeta con el nombre "LObjects Windows" en la ruta *Archivos de programas/cycling'74/Max 5.x/Max-externals*.
    - Colocar la carpeta con el nombre "LObjects help Windows" en la ruta *Archivos de programas/cycling/Max 5.x/Max-help*.
  - Para los usuarios Mac
    - Colocar la carpeta con el nombre "LObjects Mac" en la ruta *Aplicaciones/cycling'74/Max 5.x/Max-externals*.
    - Colocar la carpeta con el nombre "LObjects help Mac" en la ruta *Aplicaciones/cycling/Max 5.x/Max-help*.
- Abrir el archivo "Max-rhythmatist 1.0".
- Verificar que en la ventana de Max los LObjects estén reconocidos, en caso contrario, verificar que los objetos estén guardados en las carpetas correspondientes.
- Presionar el botón "registrar series" colocado en el "módulo inicial".
- Presionar el "bang" (recuadro gris con amarillo) colocado en el lado izquierdo del "Generador aleatorio irregular" en 5 ocasiones, presionar el "bang del lado izquierdo colocado sobre el mismo renglón. Se encenderá el LED, y se verá la secuencia de números generada en el

### MAX-RHYTHMATIST

"modulo de resultados".

- En el recuadro inferior (con el número 0.) dar click y anotar una duración en segundos, dar enter. Se podrá escuchar a un volumen bajo un patrón rítmico con alturas aleatorias.
- En el "multi-track MIDI" , sobre el renglón marcado con el número 1, introducir el valor 60 en el primer recuadro (nota baja) y dar enter, deslizar la barra de volumen hasta su extremo derecho. Presionar sobre el nodo superior izquierdo de la matriz colocada en el lado derecho. Presionar el nodo superior izquierdo de la matriz y luego presionar el "bang" colocado inmediatamente después de la barra de volumen. Se deberá escuchar el patrón rítmico repetido sobre notas aleatorias a manera de loop, y se encenderá un LED en el mismo renglón. En caso contrario, verificar en la barra de menú de Max en "Options/MIDI setup" que todas las opciones de "input" y "output" estén seleccionadas.
- En el "multi-track MIDI" , presionar el botón "récord" del multi-track MIDI. Presionar el bang colocado al lado derecho del LED encendido en el primer renglón de la matriz para detener el loop. Presionar el botón "write" e introducir un nombre de archivo, adjuntando la extensión .midi.
- En el "multi-track MIDI" , presionar el botón "read", seleccionar el archivo previamente guardado. Presionar "start", se deberá escuchar el loop. En caso contrario, repetir los dos puntos anteriores.
- En el "multi-track AUDIO, verificar que el símbolo de la bocina se encuentre en color amarillo, en caso contrario, presionarlo. Dicho color indica que el audio está activo.
- En el "multi-track AUDIO", presionar el botón "read" del primer renglón, seleccionar un archivo de audio (IMPORTANTE: tiene que ser una muestra de audio cuyos primeros dos

### MAX-RHYTHMATIST

segundos tengan suficiente volumen, ya que el programa sólo toma este tiempo de la muestra; se ha incluido en la carpeta de Max Rhythmatist un archivo de audio con el nombre archivoprueba.aiff). Insertar el valor de 1. en el recuadro de la columna "inicio", deslizar el volumen hasta el extremo derecho. Seleccionar el primer nodo del renglón (tal como se hizo con el multitrack MIDI), y presionar el "bang" colocado después de la barra de volumen. Se deberá escuchar el patrón rítmico en loop sobre la muestra seleccionada y un LED se encenderá. En caso contrario, verificar en la barra de menú de Max en "Options/DSP status" que el audio está en "On" y que se tiene alguna opción seleccionada para "output destination".

- En el "multi-track AUDIO", presionar el botón "write", asignar un nombre de archivo, presionar el botón "récord". Posteriormente, apagar el loop con el "bang" colocado al lado derecho del LED encendido. Presionar el botón "stop" colocado debajo del botón "récord".
- Presionar el botón "read", seleccionar el archivo previamente grabado. Presionar "start", se deberá escuchar el archivo de audio con el loop. En caso contrario, repetir los dos puntos anteriores.
- Conectar un controlador MIDI, y dar doble click sobre el objeto con el nombre "externo" en el recuadro titulado "teclado externo" colocado debajo del "modulo de resultados".
- En el Layer 1, dar click sobre el recuadro "on/off" del "módulo random" , presionar cualquier tecla, soltándola casi inmediatamente. Se deberá escuchar un patrón rítmico por cada tecla que se presione. En caso contrario, verificar que el controlador MIDI no requiera algún driver de instalación previo a su uso. De igual manera verificar que la barra de volumen colocada al lado extremo derecho no tenga la línea horizontal de color negro hasta abajo, si este es el caso, deslizar el controlador 7 del controlador a un punto intermedio (se puede verificar el

### MAX-RHYTHMATIST

número de control que se está utilizando en el recuadro superior derecho.

- Si todo ha funcionado como se ha descrito, Max-Rhythmatist está listo para usarse.
- Es muy importante checar las ayudas colocadas dentro de la interfaz, dando doble click sobre cualquier objeto blanco con la letra "p". es estas ayudas se explica de manera detallada la manera de utilizar y explotar las posibilidades de max-rhythmatist.

#### **4.2.2 Panel inicial**

Al dar doble click sobre el panel inicial de color amarillo se podrá observar el siguiente texto, dando instrucciones básicas al nuevo usuario:

*"Este es un patch programado en Max/MSP para generar, almacenar y grabar patrones rítmicos tanto en archivos MIDI, como en archivos de audio, para su posterior secuenciación.*

*Cada panel colocado del lado derecho es un generador de patrones rítmicos con características distintas.*

*Del lado derecho de cada uno podrás observar un recuadro blanco con la letra p algunas siglas y un signo de interrogación, si das doble click en cualquiera de ellos encontrarás una explicación detallada del modo de operación de cada módulo.*

*En la segunda columna se encuentran los módulos de transformación y de resultados, que reciben información de los generadores de la primera columna. De igual manera, en su parte inferior derecha se encuentra un recuadro blanco en el cual se puede dar doble click para ver instrucciones de uso.*

*Finalmente, del lado izquierdo de la pantalla se encuentran los multi-tracks para secuenciar, mezclar y grabar los patrones rítmicos generados y almacenados. En la parte inferior izquierda de cada uno podrás encontrar el recuadro blanco*

## MAX-RHYTHMATIST

*correspondiente a las instrucciones.*

*Antes de comenzar a explorar el patch decide si lo que vas a generar se almacenará o no en el módulo inicial*

*Oprime “REGISTRAR SERIES”, si es que se quiere enviar los patrones rítmicos a los multi-track MIDI y audio. Por default, el programa no registra las series introducidas hasta que se le indica hacerlo. Una vez seleccionada esta opción, seguirá registrando las series introducidas de cualquier generador hasta que se le indique no hacerlo con la opción “NO REGISTRARLAS”. Cuando se quiera borrar todas las series registradas, se selecciona la opción “BORRAR TODO”.*

### **4.3 GENERANDO SERIES MANUALMENTE**

Si se quiere generar una serie de valores muy específicos, entonces es conveniente usar el generador manual. Dado que cada valor introducido representa una duración individual que divide a una unidad de tiempo ( $T=1$ ), luego entonces los valores a introducir se consideran divisores de dicha unidad. Cada fracción podría ser equivalente a una figura rítmica en la notación tradicional, es decir, una negra se representará como  $1/4$ , pero en el caso de una negra con puntillo, cuyo valor equivale a  $3/8$ , deberemos convertir esta fracción en términos de  $1/X$ , por lo que el valor a considerar será  $1/2.6666$  (el denominador 2.6666 fue obtenido dividiendo en forma inversa, es decir,  $8/3=2.6666$ ).

Por ejemplo: Para introducir la serie  $1/2, 1/4, 1/7, 1/3, 1/5, 1/3.3$ , se siguen los siguientes pasos.

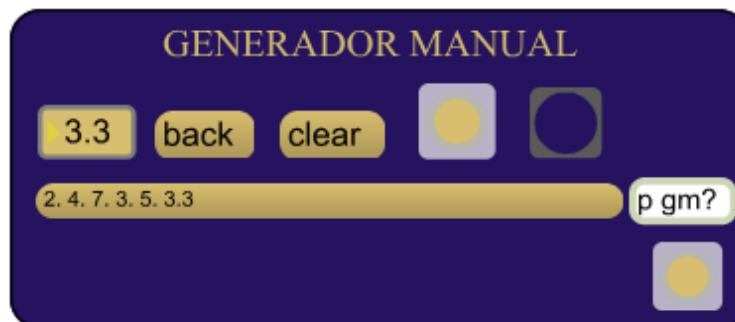
1) Oprimir “registrar series”, si es que se quiere enviarla a los multi-track MIDI y audio. Por default, el programa no registra las series introducidas hasta que se le indica hacerlo. Una vez seleccionada esta opción, seguirá registrando las series introducidas de cualquier generador hasta que se le indique no hacerlo con la opción “no registrarlas”. Cuando se quiera borrar todas las series registradas, se

## MAX-RHYTHMATIST

selecciona la opción “borrar todo”



2) Introducir los denominadores de la serie dando “click” en el recuadro superior izquierdo y anotándolos uno por uno, oprimiendo “enter” cada vez, para introducirlos en el espacio inferior. Oprimir “back” para borrar el último valor introducido, “clear” para borrar toda la serie escrita, y “bang” (círculo amarillo, al lado de "clear") para enviar la serie al patch principal, y registrarla en su caso.

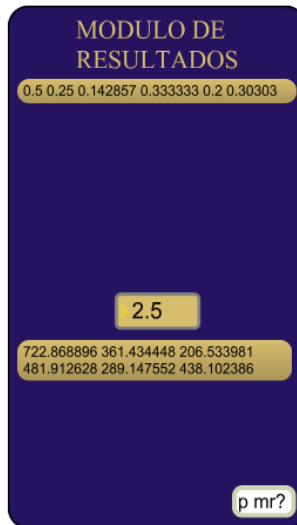


NOTA IMPORTANTE: Una vez oprimido el bang, se encenderá el “LED” del lado superior derecho indicando que este módulo fue el último que se utilizó. Esto es particularmente útil para saber que se tiene la posibilidad de enviar la secuencia previamente generada al módulo de transformaciones (que se explicará más adelante), con el bang que se encuentra en la parte inferior derecha. Cada módulo contiene un “LED” del lado superior derecho que se encenderá cada vez que se genere una secuencia rítmica en dichos módulos. Así mismo, cada módulo tiene un “bang” colocado en la parte inferior derecha para enviar la secuencia rítmica generada al módulo de transformaciones, que se explicará en la ayuda correspondiente a este módulo)

## MAX-RHYTHMATIST



3) La lista resultante se ve en el módulo de resultados en forma de decimales. En este módulo se anota la duración deseada para esta serie en segundos (Ejemplo: 2.5 segundos), dando “click” en el recuadro correspondiente, anotando la duración y oprimiendo “enter”. Esto nos dará como resultado la equivalencia de cada valor en milisegundos. Al momento de terminar este proceso se podrá escuchar la secuencia de sonidos MIDI correspondiente a esta única serie. **NOTA IMPORTANTE:** En ocasiones, dependiendo del módulo del cual se envíen las series, el módulo de resultados puede tardar algunos segundos en mostrar el resultado.



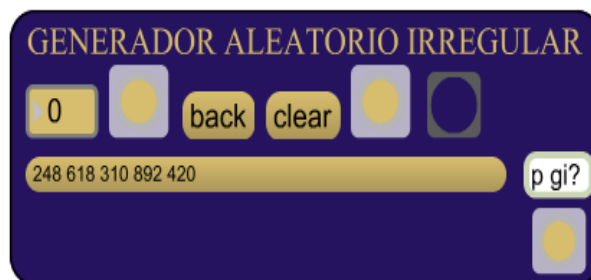
### **4.4 GENERANDO SERIES ALEATORIAS IRREGULARES**

En el generador aleatorio irregular se generan números aleatorios que por sí solos ya representan una duración individual. Estos números se generan, por default, entre 0 y 999, es decir 1000 números

### MAX-RHYTHMATIST

aleatorios, pero puede ser variado tecleando el rango total de números aleatorios deseado en el recuadro colocado en el ángulo superior izquierdo, y dando “enter”.

1) Para generarlos, basta con presionar el “bang” colocado al lado izquierdo de "back" tantas veces como números se deseen. “Back”, sirve para borrar el último número aleatorio generado, y “clear” para borrar toda la serie. Una vez generada la serie de números correspondiente se presiona el “bang” al lado derecho de "clear", lo cual encenderá el “LED”.



2) La serie se envía al módulo de resultados. Se teclea la duración en segundos deseada, se oprime “enter”, y el resultado de cada valor en milisegundos aparecerá en la parte inferior, a la vez que se escucha el resultado vía MIDI



## MAX-RHYTHMATIST



### **4.5 GENERANDO SERIES ALEATORIAS UNIFORMES**

Al igual que el otro generador aleatorio, se puede variar el rango de números aleatorios que puede usarse en este generador, para crear series irregulares de duraciones acumuladas  $T_k$  (Capítulo 5).

1) Como primer paso, se presiona el “bang” izquierdo tantas veces como números aleatorios se deseen. Al teclear el “bang” superior derecho, la serie se envía al módulo de resultados.



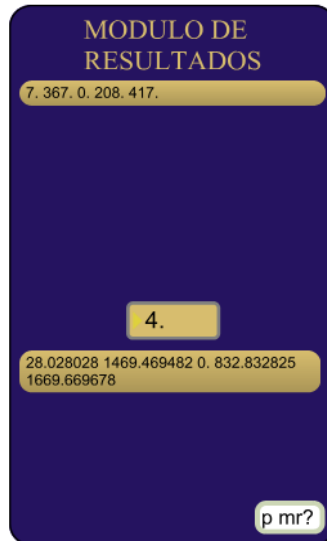
2) A diferencia de los generadores previos, la serie mostrada es el resultado de ordenar los valores



previamente generados de menor a mayor, para luego sacar la diferencia entre cada uno de ellos y

### MAX-RHYTHMATIST

obtener una serie de valores individuales, a la cual se le puede asignar una duración total en segundos.



#### **4.6 GENERANDO SERIES REGULARES**

En este generador se pueden agrupar varias series regulares.

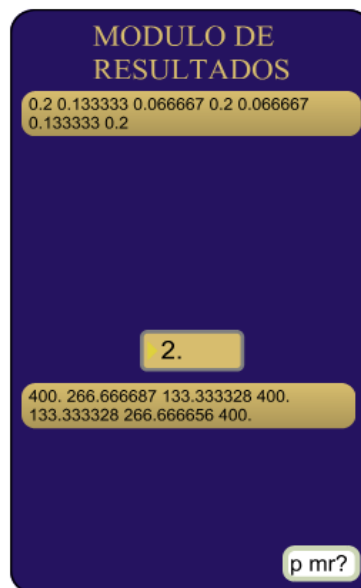
1) Primeramente, se introducen un denominador, equivalente al número de divisiones de la unidad de tiempo ( $T=1$ ), y se da “enter” para generar automáticamente una serie en valores decimales. Se pueden introducir más denominadores oprimiendo “enter” después de teclear cada uno de ellos. En el ejemplo mostrado, se agruparon una serie de tercios y una serie de quintos, que en notación musical tradicional podría equivaler a un quintillo y a un tresillo. Se oprime el “bang” superior derecho para enviar los valores al módulo de resultados.



### MAX-RHYTHMATIST



2) El módulo muestra el resultado de ordenar todos los valores agrupados de mayor a menor, para luego sacar la diferencia entre cada uno y mostrarla como una serie de valores individuales. Posteriormente se le asigna la duración en segundos deseada.



### **4.7 CAMBIANDO SERIES CON EL MÓDULO DE TRANSFORMACIONES**

El módulo de distribuciones tiene cinco gráficas almacenadas en “presets”, que pueden ser elegidas al presionar los cuadros marcados con un punto en el cuadro que se presenta arriba del espacio destinado para la gráfica.

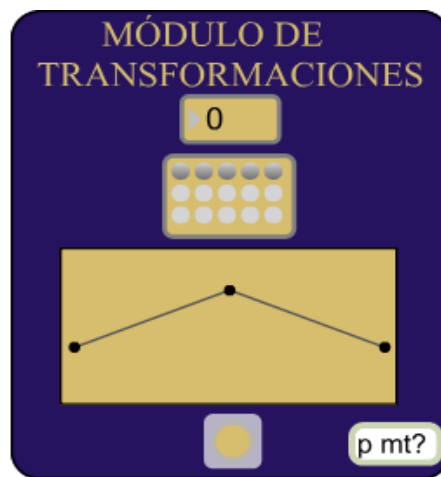
1) Si se desea hacer una gráfica propia, basta con seleccionar alguna de las previamente almacenadas y con el cursor del mouse seleccionar un punto de la misma sin dejar de oprimirlo y arrastrarlo en la

### MAX-RHYTHMATIST

dirección deseada, o bien, presionar en cualquier lugar dentro de una recta para crear un punto, para luego arrastrarlo al lugar deseado. Para almacenar la gráfica trazada se encuentran disponibles los “presets” 6, 7 y 8, por lo que bastará teclear dicho número en el recuadro colocado arriba de los “presets” y oprimir “enter”. NOTA IMPORTANTE: Una línea recta ascendente implicará una aceleración, una descendente una desaceleración y una horizontal una velocidad estable. Entre más se aproximen las líneas a una vertical el cambio de velocidad será más pronunciado.

2) Para transformar una serie, primero hay que generarla por cualquiera de los módulos de generación previamente explicados. Sólo se podrá transformar la serie proveniente del último generador utilizado, mismo que se identifica por tener encendido su respectivo “LED”.

3) Posteriormente se selecciona o traza la gráfica deseada.



4) Una vez hecho esto, hay que oprimir el “bang” que se encuentra en la parte baja del generador utilizado para enviar la serie generada al módulo de transformaciones.



NOTA IMPORTANTE: Cada generador contiene un “LED” de color que se enciende al momento de utilizarlo y se apaga al momento de usar un módulo diferente. Esto es importante notarlo, ya que sólo

### MAX-RHYTHMATIST

se podrá enviar una serie al módulo de transformaciones si esta proviene del último módulo utilizado.

5) Finalmente, se oprime el “bang” colocado dentro del módulo de transformaciones. La transformación será mostrada en el módulo de resultados tras algunos segundos de espera.

NOTA IMPORTANTE: Si se manda la serie antes de seleccionar la gráfica, el resultado final de la transformación será incorrecto, pero bastará con oprimir nuevamente el “bang” del módulo de transformaciones para realizar nuevamente la transformación.

NOTA IMPORTANTE: Una vez enviada una serie al módulo de transformaciones, esta puede ser transformada varias veces al seleccionar diferentes gráficas y oprimiendo el “bang” del módulo.

## **4.8 USANDO EL MULTI-TRACK MIDI**

Si se decidió registrar las series generadas, estas quedarán registradas en el orden en que hayan sido generadas y quedarán almacenadas simultáneamente tanto el multi-track MIDI, como en el multi-track audio. Para utilizar el multi-track MIDI se siguen los siguientes pasos:

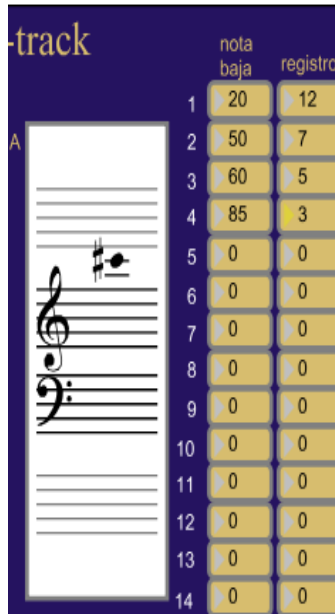
1) Seleccionar el programa (sonido) a usar y el canal al que corresponda. Inmediatamente esto quedará almacenado en los “presets” colocados bajo el título “memoria”.



2) Seleccionar la nota a partir de la cual se generarán las notas en cada canal. Cada vez que se hace esto, el recuadro donde se encuentran los pentagramas mostrará la altura que corresponde al valor introducido. Posteriormente, introducir el rango sobre el cual se generarán las notas (12= doce

### MAX-RHYTHMATIST

notas= 1 octava). En el ejemplo mostrado, los primeros cuatro canales están siendo utilizados.



3) Determinar el volumen de cada canal. Este puede ser cambiado mientras se ejecuta la secuencia de sonidos.



4) Si se desea grabar la secuencia de sonidos oprimir el botón “record” o la tecla “R” del teclado.

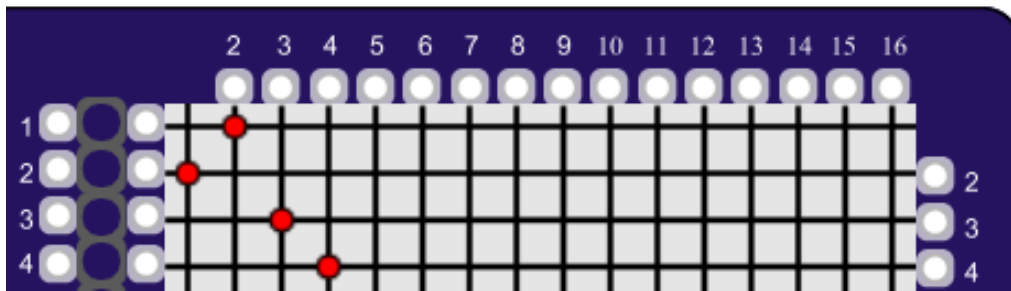


5) Seleccionar en la matriz el número de patrón rítmico que utilizará el canal (eje horizontal). Para iniciar la secuencia se tiene la opción de iniciar canal por canal, para lo cual se presionarán los “bang” colocados a la izquierda de los “LEDs” azules, o bien, iniciar todos los canales de manera simultánea, para lo cual se presionará el “bang” de la línea superior que corresponda al número de canales a iniciar. **NOTA IMPORTANTE:** Si se decidió grabar la secuencia, hay que realizar estas

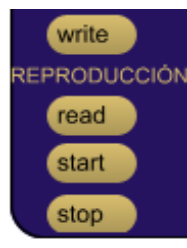
### MAX-RHYTHMATIST

acciones lo más inmediatamente posible tras haber oprimido el botón “record”.

6) Para detener la secuencia, igualmente se puede detener canal por canal con los “bangs” a la derecha de los “LEDs” azules, o todos al mismo tiempo con los “bang” colocados a la derecha de la matriz en el número correspondiente al número de canales que se quiera apagar simultáneamente.



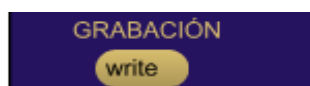
7) Una vez detenida la secuencia sonora, se dejará de grabar automáticamente. Para guardarla en archivo MIDI, hay que presionar el botón “write”; para leer este o algún otro archivo MIDI, presionar “read” para seleccionarlo; “start”, para escuchar el archivo seleccionado y “stop” para detenerlo.



## **4.9 USANDO EL MULTI-TRACK AUDIO**

El multi-track audio sigue una secuencia de pasos ligeramente diferente al del multi-track MIDI.

1) Si se desea grabar la secuencia de audio, hay que oprimir el botón “write” para dar nombre al archivo en el cual será grabado.






2) Para cada track, primero se oprime el botón “read” para seleccionar la muestra de audio a cargar,

### MAX-RHYTHMATIST

luego introducir el factor de transposición en la columna "transp" a partir de la cual se leerá la muestra. En el ejemplo se colocó 0.5 (mitad de la velocidad normal= 1 octava abajo) en el primer track, 1 en el track 2 para leerla a partir de la velocidad original, y 1.5 (una quinta justa por encima de la altura original) para el track 3.

3) Luego se asigna el valor random para determinar el rango de lectura. Los valores random generados se dividen entre 1000 y se suman al valor de inicio para obtener distintas velocidades de lectura de la muestra, que se traducen en una transposición aleatoria de la misma. En el ejemplo mostrado, el track 1 será leído de 0.5 a 1.5, es decir, entre una octava abajo y una quinta arriba de la velocidad original; el track 2, entre 1 y 1.5, entre la velocidad original y una quinta arriba; el track 3, entre 1.5 y 4, entre una quinta arriba y dos octavas arriba de la velocidad original.

4) Se asigna el volumen en que se tocará cada track, pudiéndose variar durante la ejecución de la secuencia sonora.

	random	Transp	volumen
1 read	1000	0.5	
2 read	500	1.	
3 read	2500	0.5	

5) Oprimir el botón con la figura de la bocina para habilitar el audio del equipo al que esté conectada la computadora.



6) Si se desea grabar oprimir el botón "record" o la tecla "G". Botón "Stop" o tecla "S" para dejar de

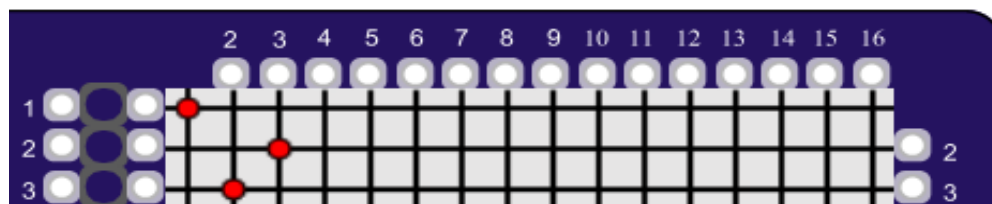


## MAX-RHYTHMATIST

hacerlo.



7) En la matriz, seleccionar los patrones rítmicos para cada track. La manera de iniciar y detener los tracks funciona igual que en el multi-track MIDI, por lo que se podrán iniciar los tracks tanto individualmente, como de manera simultánea. NOTA IMPORTANTE: Al igual que en el multi-track MIDI, si se decidió grabar la secuencia, hay que realizar estas acciones lo más inmediatamente posible tras haber oprimido el botón “record”.



8) Finalmente, una vez detenida la grabación con “Stop” o tecla “S”, se puede volver a escuchar esta u otra secuencia en audio con el botón “read”, para cargar un archivo AIFF, y con los botones “start” y “stop” para iniciarla y detenerla, respectivamente.



9) Una vez que se termine de utilizar el multi-track audio, se recomienda deshabilitar el equipo conectado apretando nuevamente el botón con la figura de la bocina.

### **4.10 USANDO EL MÓDULO EXTERNO DE EJECUCIÓN**

Con el módulo externo se puede ejecutar cualquiera de los patrones rítmicos ya almacenados mediante un controlador MIDI externo. Entre las características principales de este módulo se puede

## MAX-RHYTHMATIST

mencionar la capacidad de asignar notas o intervalos específicos a los patrones ejecutados; de igual manera, existe la posibilidad de insertar silencios; y además, por cada tecla se presionada se ejecuta el patrón rítmico una sola vez, por lo que hay una relación directa entre la acción del ejecutante y el sonido generado. De igual manera, se puede grabar lo ejecutado, pudiendo además interactuar ya sea con la secuenciación realizada con los multi-tracks o con archivos ya grabados.

Para abrir el módulo externo hay que dar doble click sobre el objeto amarillo con el nombre "externo".



Una vez abierto se podrá visualizar la siguiente pantalla.



El módulo externo está adaptado a un controlador MIDI Behringer UMX-25. Dentro de la interfaz se

### MAX-RHYTHMATIST

indica el número de control asignado a los diferentes parámetros que pueden manipularse. Para que el usuario final no tenga problemas con este aspecto al momento de conectar un controlador diferente, se ha colocado un recuadro en donde se puede observar el controlador que está siendo asignado para que el usuario pueda identificar en qué parte de su controlador se encuentran. En este mismo recuadro está colocado un botón "panic", para mandar note-offs a notas que pudieran quedar sonando mientras se manipulan los diferentes submódulos.



El volumen general se puede controlar exclusivamente mediante el control número 7 del controlador MIDI.



En la parte superior izquierda, se encuentra un recuadro con el título "teclado solo". Si se da click sobre el switch con el subtítulo "On/off", o bien, si se presiona el controlador 97, se podrá escuchar una sola nota por tecla presionada, es decir, el controlador externo funciona como un teclado convencional, que puede interactuar con los otros submódulos si el usuario así lo desea.. Se puede asignar el sonido (programa) y el canal por el que se enviará la información de este submódulo.

### MAX-RHYTHMATIST



Los otros dos paneles tienen los títulos "Layer 1" y "Layer 2", cada uno está asignado a rangos diferentes del teclado. El Layer 1 está asignado de las notas 60 a la 71, y el Layer 2 de las notas 72 a la 84. Se puede cambiar esta asignación editando el código de este módulo. Cada panel está dividido en tres partes:

- El módulo random, que se acciona presionando el switch colocado en el lado extremo izquierdo (se encenderá el LED que está a su lado), se asigna un registro, que representa el número de notas aleatorias por encima de la nota ejecutada que se podrían usar para ser asignadas al patrón rítmico. Por ejemplo, si se toca la nota 60, y se asigna un random de 7, entonces el patrón rítmico ejecutado asignará notas entre 60 y 67 de manera aleatoria. El # de patrón es el patrón rítmico que se ejecutará, el número corresponde al orden ascendente en que fue guardado en módulo principal (número 1, para el primer patrón guardado). Finalmente, los recuadros de programa y canal sirven para asignar el sonido y el canal por el cual se enviará la información.



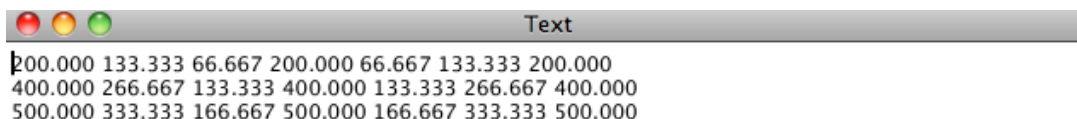
- Para facilitar la elección del número de patrón rítmico, se ha incluido en el módulo principal un acceso a la lista de patrones rítmicos guardados.



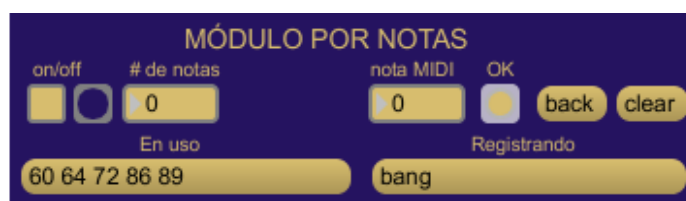
- Si se da click sobre el botón "open" se podrán observar las listas almacenadas, separadas por renglones, y en el orden en que fueron almacenadas. **Es importante decir que el número de**

### MAX-RHYTHMATIST

**patrón seleccionado podrá ser utilizado simultáneamente por todos los submódulos que componen al módulo externo.**



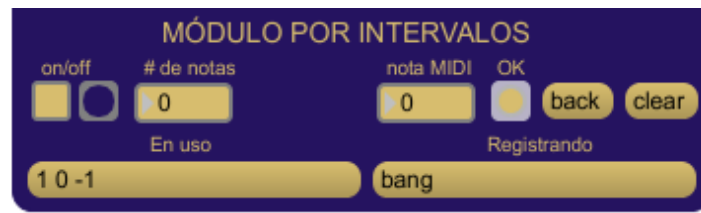
- El módulo por notas funciona de manera similar al módulo anteriormente explicado. Una vez activado, en el recuadro con el subtítulo "nota MIDI" se insertan los números de notas específicos que se quieran utilizar para ser asignados aleatoriamente al patrón rítmico. Una vez insertados se da click sobre el botón con la leyenda "OK". En el ejemplo, se asignaron las notas 60, 64, 72, 86 y 89. Una vez asignadas las notas, se asigna un número aleatorio en el recuadro con el subtítulo "# de notas", si este número es igual al número de notas insertados (en este ejemplo, 5), entonces por cada valor rítmico siempre se asignara una nota, si el valor es menor, por ejemplo 2, se asignarán las dos primeras notas insertadas, y si es mayor, se insertarán silencios dentro del patrón rítmico, entre mayor sea el número de notas, habrá mayor probabilidad de ocurrencia de silencios.



- El módulo por intervalos funciona de manera similar al anteriormente explicado. Una vez activado, en el recuadro con el subtítulo "nota MIDI", se insertan los intervalos, medidos a partir de la tecla que se presione, que serán asignados al patrón rítmico. Estos intervalos se miden por semitonos (1=un semitono), y se pueden incluir número negativos si es que se quiere que haya intervalos por debajo de la tecla presionada. Una vez insertados los valores,

### MAX-RHYTHMATIST

se da click sobre el botón con la leyendo "OK". En el ejemplo, se insertaron los valores 1, 0 y -1 (semitono hacia arriba, nota original, semitono hacia abajo). El grado de aleatoriedad, que se asigna en el recuadro con el subtítulo "# de notas" funciona de igual manera que en el módulo por notas.



Es importante decir que el usuario tiene la libertad de encender todos los submódulos para experimentar con todos ellos simultáneamente, o bien encender uno y luego apagar otro. La razón por la cual no se implementó una forma de encender un submódulo y apagar el resto, fue el hecho de que al hacer eso, la posibilidad de dejar notas sin *note-off* (notas que se mantienen sonando) era muy alta. Finalmente, si se quiere grabar lo ejecutado, se tendrá que ir a la parte de grabación colocada en el multi-track MIDI para llevarlo a cabo, esto a su vez permite que lo grabado en el multi-track pueda interactuar con lo ejecutado en el módulo externo.

Finalmente, dentro del código del módulo externo se ha incluido un subpatch en versión *alfa* para poder adaptar todo este módulo al manejo de muestras de audio. Por el momento, la generación de clicks en su manejo hizo que se tomara la decisión de no incluir este subpatch en la interfaz final.



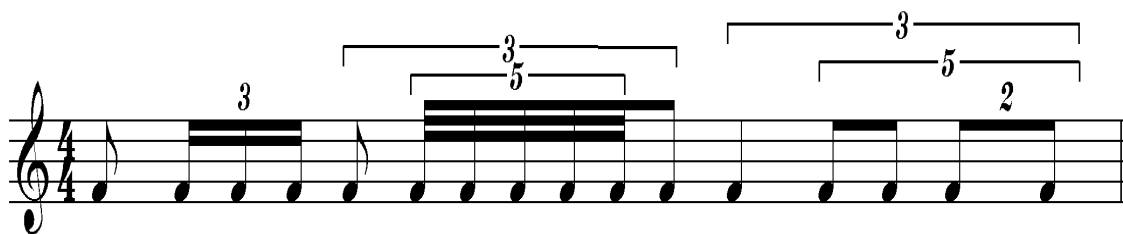
## CAPÍTULO 5

### EVALUACIÓN DE MAX-RHYTHMATIST

Este capítulo evalúa la eficacia del programa al momento de transcribir patrones rítmicos transcritos con la escritura musical tradicional, así como el paso inverso, transcribir a escritura tradicional una serie numérica generada por el programa. De igual forma, se evalúa la pertinencia del programa cuando este se utiliza para generar texturas musicales específicas. La evaluación final incluye una descripción detallada de las características y limitaciones de **Max-Rhythm**.

#### **5.1 TRANSCRIPCIÓN DE UN RITMO ESCRITO EN NOTACIÓN MUSICAL A NOTACIÓN NUMÉRICA.**

Tomemos como ejemplo el ritmo de la figura 11 presentado en el capítulo 2, referente a los elementos rítmicos en la música del siglo XX, específicamente al agrupamiento de figuras rítmicas con potencias distintas.



Puesto que la notación rítmica tradicional ya está basada en una representación gráfica de fracciones numéricas, bastará con hacer el proceso inverso, es decir, cambiar la representación gráfica a una representación en fracciones, que pueda ser posteriormente introducida en **Max-Rhythm**.

El primer valor rítmico es:  $1/8$ .

Los siguientes tres representan un tresillo de dieciseisavos, o bien, una subdivisión ternaria de un octavo, por lo que cada valor será el resultado de dividir ese octavo entre tres:  $(1/8)/3 = 1/24$ .

### MAX-RHYTHMATIST

El siguiente tresillo es la subdivisión en tres partes de un cuarto, por lo que cada tiempo de ese tresillo estará representado por:  $(1/4)/3=1/12$

Por lo tanto, cada valor del quintillo que se encuentra dentro de este último tresillo será:

$$(1/12)/5=1/60.$$

El siguiente tresillo se encuentra en el espacio de dos cuartos, por lo que cada tiempo de este tresillo está representado por la fracción:  $(2/4)/3= 2/12=1/6$ .

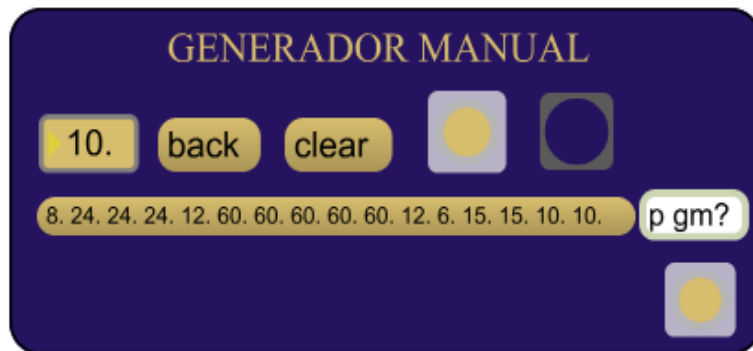
El último quintillo abarca el espacio de 2 tiempos del último tresillo, por lo que cada tiempo dentro de este quintillo se expresará de la siguiente manera:  $(2/6)/5=2/30=1/15$ .

Finalmente, el dosillo colocado dentro de éste último quintillo abarca un espacio de tres tiempos de este quintillo, por lo que cada tiempo del dosillo será:  $(3/15)/2=3/30=1/10$ .

La serie de valores individuales de este ritmo será por tanto:

$$S= 1/8, 1/24, 1/24, 1/24, 1/12, 1/60, 1/60, 1/60, 1/60, 1/60, 1/12, 1/6, 1/15, 1/15, 1/10, 1/10.$$

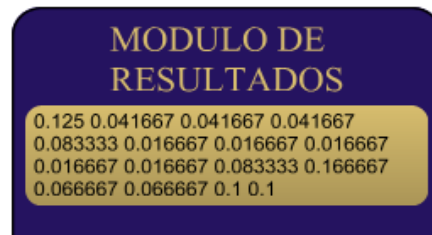
Esta serie puede introducirse fácilmente en **Max-Rhythmatist** a través del generador manual, registrando los denominadores de cada uno de los valores de la misma.



El módulo de resultados arrojará la siguiente serie, expresada en números decimales:



## MAX-RHYTHMATIST



Cabe mencionar que en caso de que la transformación del ritmo escrito diera como resultado una fracción numérica cuyo numerador sea distinto a 1, por ejemplo  $3/8$ , dicho resultado se puede igualmente expresar de la forma  $1/X$ , siendo  $X$  no necesariamente un entero. Por ejemplo:  $3/8 = 1/2.6666$ ; de tal forma que el denominador  $2.6666$  será el valor a introducir en el generador; obtenido de dividir en forma inversa  $8/3 = 2.6666$ ).

Por lo tanto, se puede decir que cualquier ritmo escrito con la notación tradicional, por intrincado que pueda parecer, es factible de ser traducido en términos de fracciones, y por tanto, de ser registrado en **Max-Rhythmatist** a través del generador manual.

## **5.2 TRANSCRIPCIÓN DE UN RITMO ESCRITO EN NOTACIÓN NUMÉRICA A NOTACIÓN MUSICAL.**

### **5.2.1 Transcripción de una serie regular compuesta.**

La transcripción de una serie generada por **Max-Rhythmatist** a la notación musical tradicional no representa ningún problema cuando esta proviene del generador de series regulares.

Como ejemplo, tomemos la serie generada en el capítulo 4, punto 4.3.1 (figura 22), que es una serie compuesta que representa la unión de dos series regulares, una dividida en 3 partes y la otra en 5.

$$Sc_{(3+5)} = Su_3 \cup Su_5 = \{0.2, 0.1333, 0.0666, 0.2, 0.0666, 0.1333, 0.2\}$$

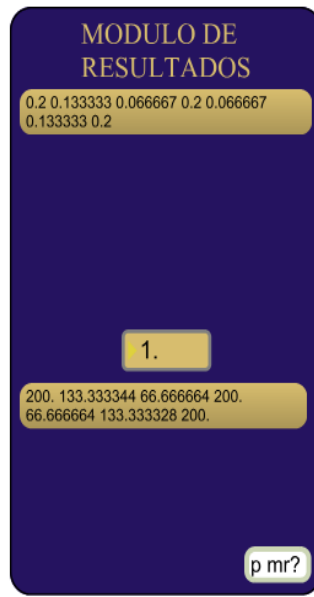
Dependiendo de la duración en segundos asignada a esta serie, se puede escribir este ritmo en un tiempo de un compás si es de corta duración, o como un compás completo, en el caso de una larga duración.

MAX-RHYTHMATIST

Ejemplo:

Si a la serie de valores resultante les asignamos una duración total de 1 segundo, dará como resultado las duraciones siguientes en milisegundos:

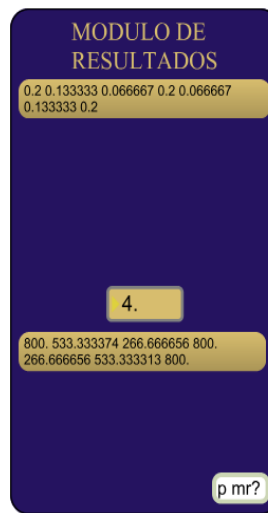
200, 133.33, 66.66, 66.66, 133.33, 200



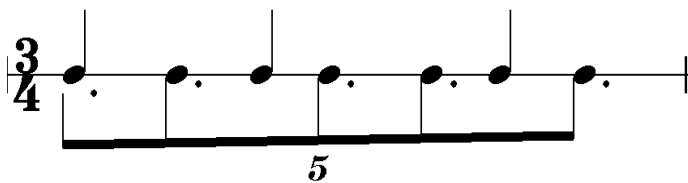
Por otro lado, si le asignamos una duración mayor, sería poco práctico transcribir dicho ritmo en un tiempo. Por ejemplo, si le asignamos al patrón ya mencionado una duración de 4 segundos, dará como resultado las siguientes duraciones en milisegundos:

800, 533.33, 266.66, 266.66, 533.33, 800

## MAX-RHYTHMATIST



De tal forma que resulta más práctico expandir el patrón resultante en un compás, tal como se muestra a continuación.



Por lo tanto, la representación musical de cualquier serie regular compuesta no representa mayores dificultades.

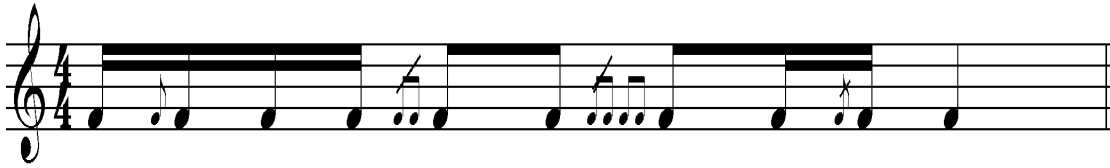
### **5.2.2 Transcripción de series irregulares.**

Las series irregulares, creadas con cualquiera de los dos generadores aleatorios, no tienen ninguna base regular sobre la cual interpretarlos. Además, su generación busca precisamente no sujetarse a ninguna base, y por tanto, liberarse de la notación rítmica tradicional, cuyos símbolos representan duraciones relacionadas entre sí.

Sin embargo, si entendemos la naturaleza de estos ritmos, se puede realizar una notación musical de carácter aproximado. Esto ayudaría, incluso si se quiere usar estos ritmos en combinación con ejecutantes en vivo que busquen imitarlos. Para ello, podemos auxiliarnos del uso de las apoyaturas,

### MAX-RHYTHMATIST

mostradas en el Capítulo 1, figura 14.



Para realizar una transcripción de este tipo debemos partir de ciertas convenciones. Tomando como referencia el procedimiento que usan algunos editores de partituras en el que se transcribe un archivo MIDI a notación gráfica, (proceso que se denomina cuantización del tiempo), y que considera cierto rango de tolerancia para asignar un valor rítmico a cierto evento temporal, podríamos asignar una serie de símbolos a nuestra transcripción de la siguiente manera<sup>32</sup>:

- a) Apoyatura rápida antes de tiempo. Valor rítmico que ocurre antes del valor rítmico correspondiente a un tiempo fuerte, y que cuya rapidez equivale a valores por debajo del cuarto nivel de subdivisión. (Por ejemplo, valores menores a los treintadosavos en un compás subdividido en cuartos)

NOTA: El hecho de tomar al cuarto nivel como límite para asignar el adorno será también una convención para evitar que la transcripción resultante contenga valores rítmicos exactos que sean demasiado pequeños y que puedan complicar su lectura.



- b) Apoyatura rápida después de tiempo. Valor rítmico que ocurre después del valor rítmico correspondiente a un tiempo fuerte, y que cuya rapidez equivale a valores por debajo del cuarto

<sup>32</sup> Esta notación la usé en la obra "La Vista", para guitarra amplificada y voz (2001).

*MAX-RHYTHMATIST*

nivel de subdivisión.

- c) Apoyatura lenta antes de tiempo. Valor rítmico que ocurre antes del valor rítmico correspondiente al tiempo fuerte, y cuya rapidez equivale a valores iguales o superiores al cuarto nivel de subdivisión.
  
- d) Apoyatura lenta después de tiempo. Valor rítmico que ocurre después del valor rítmico correspondiente al tiempo fuerte, y cuya rapidez equivale a valores iguales o superiores al cuarto nivel de subdivisión.

Ejemplo:

Al usar el generador aleatorio irregular se obtuvieron los siguientes valores:

$$S=591+251+83+14+383+859+857$$

La suma total de duraciones del ejemplo seleccionado es: 3032.

Comencemos por asignar un valor convencional de 1000 a cada cuarto en un compás de 3/4, de tal forma que el valor total previamente anotado equivale a un patrón rítmico que sobrepasa ligeramente la duración del compás ya mencionado. Cada tiempo ó cuarto tendrá un valor de 1000, cada octavo de 500 y cada dieciseisavo de 250. Las apoyaturas lentas tendrán un valor de 125 +/- 25 (20%), para aproximarlos a un treintaidosavo, para dejar todos los valores fuera de este rango a las apoyaturas rápidas.

Consideremos ahora la serie de duraciones acumuladas:

$$S= 591, 842, 925, 939, 1323, 2181, 3038$$

Aproximando dichas duraciones a los valores asignados para cuartos, octavos y dieciseisavos nos dará como resultado:

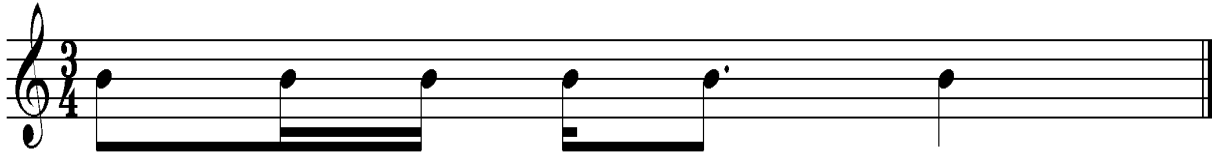
$$S= 500, 750, 1000, 1000, 1250, 2000, 3000$$

MAX-RHYTHMATIST

Que en valores de duración individual es igual a:

S= 500, 250, 250, 250, 750, 1000

Equivalente a la siguiente notación rítmica:



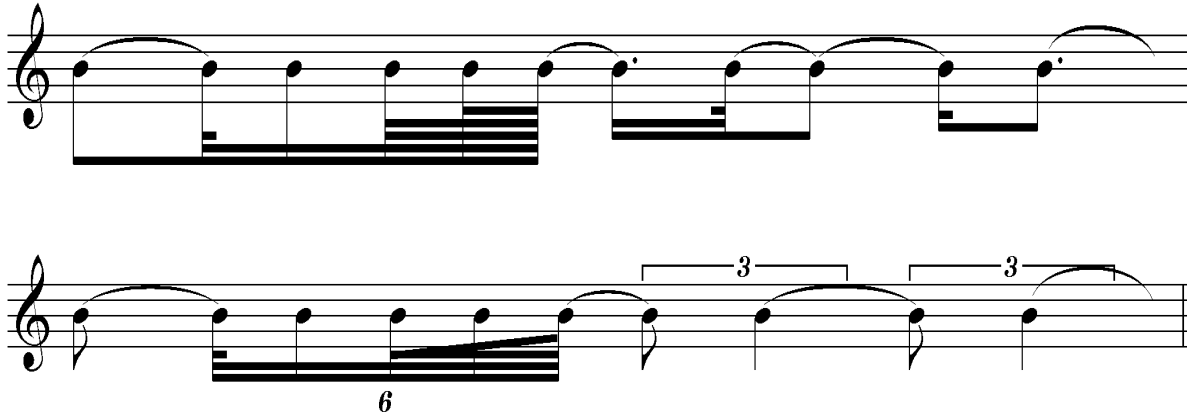
De la serie de valores acumulados del ejemplo, el valor 591 representa una duración ligeramente mayor a 500, por lo que puede representarse como un octavo, ligado al primer dieciseisavo con una apoyatura rápida después de tiempo. El valor 852 equivale a colocar una apoyatura lenta ligada después del segundo valor de dieciseisavo (750). Los valores 925 y 929, representan dos apoyaturas rápidas ligadas al valor de dieciseisavo del segundo tiempo, que comienza en el valor 1000. El valor 1323 se representaría con una apoyatura rápida colocada después del octavo con puntillo (1250), ligada al siguiente valor, y finalmente, el valor 2181, como una apoyatura lenta colocada después del último cuarto (200), que durará hasta el valor 3038, es decir, un poco después del término del compás.

Agregando las apoyaturas correspondientes nos dará como resultado:



Esto, evidentemente, nos da como resultado una escritura aproximada, pero de lectura más convencional, que en caso de usarse para hacer interactuar a un intérprete con el programa le permitiría leer este tipo de ritmos con cierta flexibilidad. Como ejemplos, se muestran dos posibles interpretaciones del ritmo previamente anotado:

## MAX-RHYTHMATIST



### **5.3 ELABORACIÓN Y TRANSCRIPCIÓN DE TEXTURAS MUSICALES.**

#### **5.3.1 Textura polifónica con poli-tempos, aceleraciones y desaceleraciones.**

Tal como se explicó en el capítulo 1, Conlon Nancarrow compuso, con la ayuda de pianolas, estructuras poli-rítmicas cuyas voces podían realizar cambios de tempo de manera independiente a la estructura total.

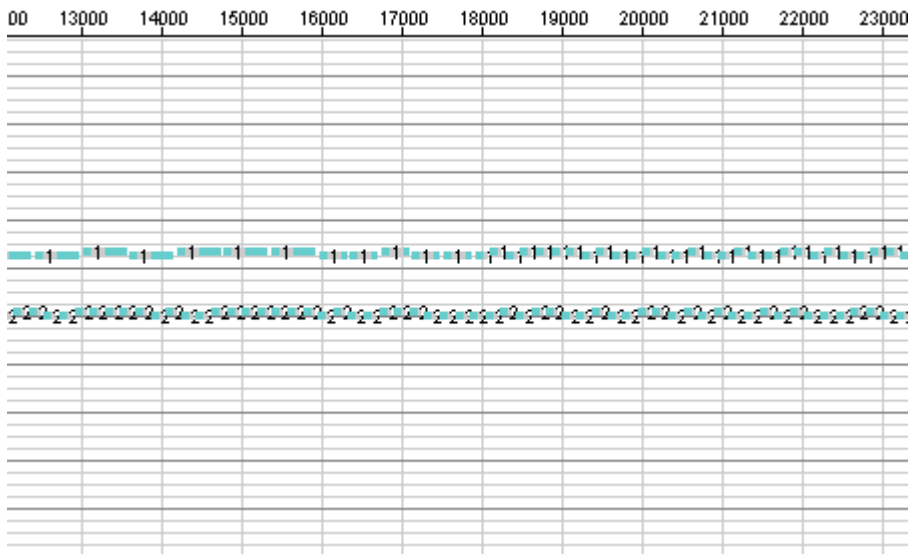
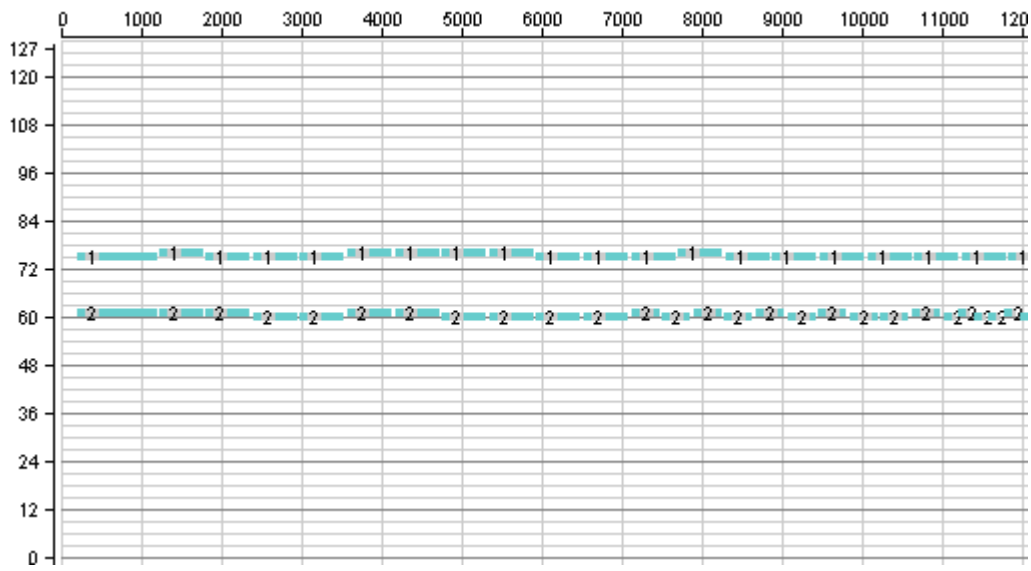
Con **Max-Rhythmatist** esto se puede lograr generando un solo patrón rítmico, y registrarlo con diferente duración, o incluso transformarlo con las gráficas de aceleración o desaceleración. Una vez registradas todas las series, con el multi-track MIDI del programa podemos elegir el número de canales y sonidos a utilizar. Hacemos iniciar los canales elegidos al mismo tiempo y con el mismo patrón rítmico, y vamos variando los patrones voz por voz, de tal manera que las voces que en un principio hacían el mismo patrón rítmico a la misma velocidad, se vayan desfasando poco a poco.

Una partitura o representación de este tipo de textura se podía obtener con la ayuda del objeto

`Detonate` que en Max/MSP almacenaba la secuencia MIDI generada en una gráfica que de altura contra tiempo, y coloca el número de canal al lado de cada nota.<sup>33</sup>

<sup>33</sup> En la versión 5 de Max/MSP esta visualización MIDI dejó de estar disponible en el objeto `DETONATE`. Sin embargo, si se graba la secuencia generada en Max/MSP en un archivo, este puede ser visto de manera similar en cualquier editor externo, como por ejemplo, `MidiSwing`, software libre para edición MIDI que funciona en Windows, MacOS y Linux.

### MAX-RHYTHMATIST



El mismo procedimiento se puede aplicar a una secuencia de audio, teniendo cuidado en seleccionar las muestras que nos permitan diferenciar las diferentes capas rítmicas. Gráficamente, su representación sólo es posible con la ayuda de un espectrograma de amplitud contra tiempo.

#### **5.3.2 Textura con compases compuestos y reacomodo de acentos.**

En el capítulo 1, vimos cómo Bartók y Stravinsky usaban cambios de acentuación, y compases compuestos. El problema que plantea este tipo de textura radica en la acentuación, que como se vio



### MAX-RHYTHMATIST

en el Capítulo 1 y 2, es un fenómeno que no solamente implica la duración de los eventos rítmicos, sino que depende de la interacción de estos con otros elementos. Con **Max-Rhythm** podemos valernos de las acotaciones que podemos hacer al número de notas empleadas o a la amplitud del registro para poder lograr la acentuación deseada.

Como ejemplo, tomemos un compás de 4/4 u 8/8, sobre el cual deseamos hacer las acentuaciones, 2+3+3, 3+3+2, 3+2+3, siendo 2 equivalente a un valor rítmico de 1/4 y 3 a 1/4 con puntillo, o bien 3/8.

Estos cambios de acentuación pueden ser introducidos en el generador manual. Recordemos que en este los valores introducidos representan un número que divide a 1, por lo que el valor de  $3/8=1/2.6666$ . Luego entonces, los valores a introducir serán:

(4, 2.6666, 2.6666); (2.6666, 2.6666, 4); y (2.6666, 4, 2.6666)

Una vez introducidas estas series de acentos y registradas con una duración dada, se procede a generar una serie con el generador de series regulares. Bastará con introducir el valor de 32, para generar un ritmo regular de 32/32.

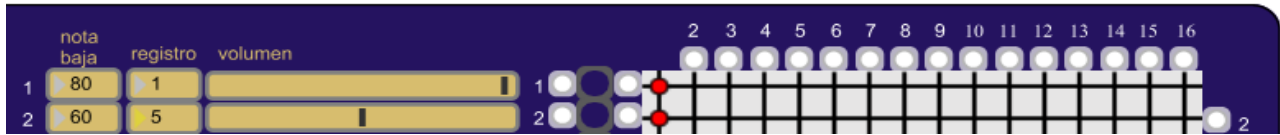
Las series de acentos corresponderán a un canal MIDI con un sonido a elegir y un número de notas restringido, y la serie regular a otro canal MIDI con un registro más que no se empalme con el registro de los acentos, y que además tendrá un volumen menor al del otro canal. Al ejemplificar esto usando notación musical tendríamos lo siguiente:



La voz superior representa el canal MIDI de acentos generados por la serie (2.6666, 4, 2.6666), y la melodía inferior al canal MIDI regular.

## MAX-RHYTHMATIST

Una vez asignados los parámetros mencionados, se accionan los canales al mismo tiempo y uno decide en qué momento se cambia la acentuación seleccionando el nodo correspondiente.



Para el multi-track audio el procedimiento es similar, tan sólo difiere en los valores que se asignan para delimitar los registros correspondientes, que están dados por rangos numéricos y no por número de notas.

### **5.4 EVALUACIÓN DE MAX-RHYTHMATIST**

Una vez ejemplificados algunos usos del programa, se procede a evaluar las fortalezas y debilidades de **Max-Rhythmatist**.

#### **5.4.1 Ventajas de Max-Rhythmatist**

El patch ofrece las siguientes ventajas:

a) Facilidad de introducir, registrar y manejar cualquier ritmo escrito en notación tradicional una vez que este ha sido traducido a términos numéricos. Es importante decir que la misma notación tradicional nos da elementos suficientes para poder traducirla numéricamente, dado que todos los valores rítmicos representan fracciones numéricas por sí solos.

b) Por otro lado, con los generadores irregulares o aleatorios podemos crear patrones rítmicos que difícilmente se podrían obtener a partir de una notación tradicional. Esto debido a que la notación tradicional se basa en la subdivisión de niveles rítmicos regulares, mientras que la generación irregular considera un segmento de tiempo como un único nivel a dividir.

c) Derivado del punto anterior, la generación irregular o aleatoria nos libera de la notación tradicional, y esto a su vez permite que los valores numéricos generados puedan ser posteriormente

### MAX-RHYTHMATIST

representados gráficamente con una notación musical de carácter aproximado, pero más convencional, que permitiría a un intérprete ejecutar estos mismos patrones rítmicos, teniendo con ello la posibilidad de interactuar con **Max-Rhythmatist**.

d) El manejo de poli-ritmos y poli-tempos. Una vez registrados los patrones rítmicos a utilizar permite al usuario la creación de secuencias MIDI o audio, tanto de manera estructurada, es decir, planeando exactamente los momentos en que se cambiarán los patrones rítmicos a utilizar y los canales MIDI o muestras de audio a utilizar, así como de manera improvisada, manejando el programa en tiempo real.

e) Las secuencias MIDI y audio grabadas durante la sesión de trabajo en **Max-Rhythmatist** pueden ser utilizadas sin problema en otros programas de cómputo, o bien, por dispositivos MIDI conectados a la computadora. En el proceso de composición de una pieza electroacústica, esto permite tener un material sonoro inicial que ya cuenta con un proceso de elaboración, desarrollo y transformación previo.

f) La programación en Max MSP permite al usuario modificar o añadir los elementos que considere necesarios para adaptar **Max-Rhythmatist** a una situación muy específica, tanto para su utilización en tiempo real, como en estudio.

g) Se tiene mayor control en los patrones rítmicos generados que en los módulos y objetos de Essl, los cuales fueron descritos en el Capítulo 1 (*Real-Time Composition Library*). Esto se debe a que en **Max-Rhythmatist**, se conocen los valores generados (incluso en los patrones rítmicos aleatorios), se tiene la oportunidad de omitir y volver a generar valores cuando las características del patrón no se ajusten a las necesidades del usuario, se puede modificar un patrón rítmico ya generado, se puede asignar una duración arbitraria a cada patrón, y finalmente, se puede mezclar el uso de patrones rítmicos regulares y aleatorios.

### MAX-RHYTHMATIST

h) De igual manera, el uso flexible de los patrones rítmicos, tanto en MIDI como en audio, y la posibilidad de tenerlos una colección disponible en multi-tracks para su mezcla final en un archivo MIDI o AIFF, son aspectos de los cuales carecen los módulos y objetos de Essl; a menos que a estos últimos se les añadan los objetos y módulos necesarios para realizar las funciones descritos en el punto incisos g y h, lo cual necesitaría de un trabajo de programación igual o mayor al realizado para elaborar **Max-Rhythmatist**.

i) Se tiene un gran control de los ritmos a utilizar y de la textura musical que se obtendrán, a diferencia del prototipo de Reck Miranda, mencionado en el Capítulo 1 (*Intelligent Rhythm Generator*), en donde los ritmos y texturas dependen de lo que el ejecutante haga, y de la manera en cómo el programa lo interprete. Sin embargo, es importante mencionar que la concepción de ambas aplicaciones es distinta, ya que el programa de Reck Miranda busca una generación rítmica derivada del análisis de la información recibida, y **Max-Rhythmatist** está basado en un enfoque de diseño rítmico predeterminado.

El módulo externo de ejecución, por otro lado, tiene características complementarias que no contiene el módulo principal.

a) Primeramente, la posibilidad de tener una relación entre la acción del usuario y el sonido generado, ya que por cada tecla presionada se ejecuta sólo un patrón rítmico.

b) Se tiene un mayor control sobre las alturas manejadas y también se tiene la posibilidad de manejar silencios. Algo que no es posible con el módulo principal.

c) Finalmente, la posibilidad de generar texturas musicales en tiempo real se hace mucho más fácil con este módulo. Mientras que con el módulo principal se hace más cómoda la labor de composición.

d) En conjunto ambos elementos del patch (el módulo principal y el módulo externo) se

### MAX-RHYTHMATIST

complementan, y dadas sus características distintas permiten decir que **Max-Rhythmatist** es en efecto un instrumento virtual de acuerdo a la descripción dada por Schnell y Battier (Schnell y Battier, 2002. Págs. 138-142), mencionada en el capítulo 1.

#### **5.4.2 Limitaciones de Max-Rhythmatist**

El módulo principal tiene las siguientes limitaciones:

a) Por muy simple que sea traducir la notación musical tradicional en números, siempre tendrán que realizarse las operaciones aritméticas necesarias para ello, antes de empezar a usar **Max-Rhythmatist**, esto puede resultar incómodo para el usuario. La misma consideración se aplica para los cálculos que el usuario debe hacer para determinar el rango de frecuencias en los que oscilarán las muestras de audio en el Multi-track Audio.

b) A diferencia de las secuencias MIDI, que contaban con el objeto `Detonate` en la versión 4 de Max/MSP o de algún editor MIDI externo que permita ver duraciones, alturas y canales de manera clara, en Max/MSP no se cuenta con ningún elemento o herramienta para representar gráficamente los eventos rítmicos en una secuencia de audio.

c) Puesto que en los objetivos de este trabajo se planteó centrar la programación en términos estrictamente rítmicos, es decir, trabajando sobre duraciones, el manejo de otros elementos musicales dentro del módulo principal, como la altura o la articulación, es muy restringido.

d) El programa funciona como un generador de material rítmico, con el cual se pueda trabajar posteriormente, por lo tanto, su uso en tiempo real interactuando con otros elementos informáticos o con instrumentistas requerirá necesariamente de su adaptación, lo cual implica trabajo de programación que podría realizarse a futuro.

e) Derivado del punto anterior, a medida que se tengan que agregar o modificar elementos del programa, el espacio en pantalla irá creciendo, haciendo con ello más incómodo su uso. Este

### MAX-RHYTHMATIST

problema se puede solventar con la utilización de sub-módulos, de manera que se empaqueten los elementos agregados; sin embargo, la interfaz usuario cuenta ya con poco espacio, incluso para nuevos sub-módulos, por lo que una reelaboración profunda del programa implicaría trabajar directamente en el módulo principal.

El módulo externo de ejecución tiene las siguientes limitaciones:

a) El intercambio entre submódulos puede llegar a ser incómodo, ya que se tienen que presionar dos diferentes switches, uno para encender el submódulo que se va a utilizar y otro para apagar el otro. Esto en el momento de una ejecución en vivo puede llegar a ser confuso.

b) A pesar de contar con un mayor control sobre alturas, silencios y volúmenes, todavía hay un grado de aleatoriedad en su manejo, por lo que repeticiones de notas en una secuencia fija, o silencios colocados de manera regular no pueden ser manejados en este módulo.

c) La imposibilidad de eliminar la generación de clicks al manejar muestras de audio, razón principal por la que no se incluyó la programación dentro de la interfaz de este módulo, hace que este módulo quede incompleto.

## CONCLUSIONES

# LINEAS FUTURAS DE INVESTIGACIÓN Y DESARROLLO DE MAX-RHYTHMATIST

Derivado de la evaluación realizada en el capítulo previo, en este capítulo se plantea la manera de expandir las funciones del programa generador de ritmo. Se describen los aspectos teóricos que podrían investigarse para integrar aspectos relacionados con el manejo de alturas, timbres, y ataques. Así mismo, se describen las características que deberá cumplir **Max-Rhythmatist** para integrar dichos aspectos, así como las consideraciones finales.

### ***EXPANSIÓN DE LAS FUNCIONES DE MAX-RHYTHMATIST***

Para mejorar las funciones de **Max-Rhythmatist** será necesario:

1) Dentro del módulo principal, hacer que los patrones rítmicos generados interactúen con otros parámetros musicales, especialmente:

- a) Alturas. El programa podría ofrecer una mayor variedad en el manejo de estas mediante el uso y selección de diversos tipos de escalas, tanto predeterminadas como aleatorias, ordenadas ascendente o descendentemente, o colocadas en cualquier otro orden determinado por el usuario. De igual manera, se deberá implementar el uso de series y células interválicas, así como operaciones de transformación de las mismas.
- b) Articulaciones. Se deberá integrar el manejo de articulaciones como el *legatissimo*, *legato*, *non-legato*, *stacatto*, etc., con el fin de facilitar la creación de texturas musicales específicas.
- c) Dinámicas. Aunque ya se cuenta con controles de volumen, estos se tienen que manejar individualmente mientras se generan las secuencias. Por lo tanto, se deberán integrar los

### MAX-RHYTHMATIST

controles necesarios para determinar dichos niveles y los cambios (crescendo y decrescendo) que puedan tener durante la creación de la secuencia de manera predeterminada.

#### 2) Dentro del módulo externo:

a) Resolver los problemas de clicks generados en la versión alfa para el manejo de muestras de audio.

b) Integrar la opción de manejar patrones fijos de alturas, intervalos y/o silencios, y la posibilidad de invertirlos y retrogradarlos.

#### 2) Rediseño de la interfaz usuario:

Los elementos incluidos hasta el momento en la interfaz usuario están contenidos dentro de una sola pantalla. Sin embargo, la inclusión de más elementos se empezaría a complicar, por su número, por lo que a medida que se necesiten será necesario colocar menús y sub-menús.

#### 3) Uso de otros lenguajes de programación.

Este es un punto crucial para el desarrollo de **Max-Rhythmatist**. Max/MSP ha demostrado que puede implementar procesos musicales que serían más complicados con otros lenguajes de programación de carácter más general. Sin embargo, cuestiones estrictamente de control o transferencia de datos, y el espacio ocupado en pantalla a medida que se requieren más elementos, hacen que la expansión del programa pueda presentar obstáculos si se mantiene la programación en Max/MSP. Se tienen por tanto dos alternativas:

a) Reprogramar todas las funciones con otro lenguaje que pueda facilitar el manejo de datos.

Dentro de de las opciones de esta alternativa se encuentra Supercollider, que es un lenguaje de texto diseñado para manejar procesos musicales, y que ha resultado ser de manejo más sencillo que otros lenguajes de este tipo, como CSound. Otra opción se encuentra en lenguajes de programación más generales como C o Java, que cuentan con colecciones de



## MAX-RHYTHMATIST

comandos enfocados a aplicaciones musicales, y que aunque resultarían menos accesibles para ser modificados por el usuario, facilitarían la expansión de las funciones de **Max-Rhythmatist**.

b) Combinar el uso de Max MSP con otros lenguajes. Es importante decir que la última versión de Max permite la interacción de este con Java, Javascript, C y otros lenguajes. Otra alternativa es el uso de lenguaje C, que también permitiría el diseño de objetos Max con funciones muy específicas, como es el caso de los L objects usados en **Max-Rhythmatist**.

## **INVESTIGACIÓN TEÓRICA A REALIZAR**

Para desarrollar los aspectos técnicos de **Max-Rhythmatist** se necesitará ahondar en los siguientes aspectos teóricos.

1) Manejo de alturas. Ahondar en la teoría referente a los sistemas seriales, especialmente la teoría desarrollada por George Perle respecto a la tonalidad de doce sonidos (*Twelve-tone Tonality*)

2) Manejo de intervalos y clases de alturas. Muy relacionado con el punto anterior, se deberá explorar el manejo de grupos de notas y sus relaciones interválicas, muy especialmente la *Set-Theory* desarrollada por Allen Forte.

3) Propiedades acústicas de diversos modos de ataque. Un aspecto acústico más específico a estudiar será el relacionado con la forma de imitar los ataques, o modos de articulación de los instrumentos musicales conocidos, empezando por los más comunes: *legato*, *no legato* o *suelto*, *staccato*, *staccatissimo*, *portato*, y *tenuto*.

4) Comandos, objetos, e instrucciones aplicables a parámetros musicales contenidos en otros lenguajes de programación. Dado que la última versión de Max/MSP acepta programación tanto en Javascript como en Java, será importante estudiar estos lenguajes para ver la manera en que ellos

### *MAX-RHYTHMATIST*

puedan simplificar la programación a desarrollar. Tampoco se descarta utilizar el lenguaje C para crear objetos Max. Igualmente importante será evaluar la viabilidad de otros lenguajes de programación enfocados a aspectos musicales como Nyquist, CSound o Supercollider, para lograr los objetivos de expansión de **Max-Rhythmatist**.

### **CONSIDERACIONES FINALES**

**Max-Rhythmatist** cumple con los objetivos iniciales de la investigación, que básicamente se pueden resumir en la posibilidad de generar patrones rítmicos emparentados con las diversas técnicas rítmicas de la música de concierto del siglo XX para poder utilizarlos en la creación de material sonoro, tanto en archivos MIDI como en muestras de audio, y que ellos fueran de utilidad para la elaboración de composiciones electroacústicas. También cumple con las características de un instrumento virtual compuesto, ya que aunado a las características ya mencionadas en este párrafo, se integra también la posibilidad de ejecutar los patrones rítmicos con un controlador externo, como si fuera un instrumento acústico.

Otro aspecto importante en el desarrollo de esta investigación fue el establecimiento de una base teórica, así como una notación específica, para transcribir patrones rítmicos mediante el recurso de series numéricas, o bien mediante generación aleatoria.

En lo que respecta a la interfaz usuario, se logró que el manejo del material generado fuera lo suficientemente sencillo con el uso de las matrices colocadas en la interfaz, y con los parámetros que el usuario puede predeterminar. Esto permite al usuario un buen control sobre las características del material, a la vez que se le ayuda a ser flexible al momento de manejarlo.

Sin embargo, desde la investigación teórica se percibió que reducir el desarrollo del programa al aspecto meramente rítmico necesariamente limitaría también las posibilidades del programa, lo cual

*MAX-RHYTHMATIST*

abre la puerta a su desarrollo futuro considerando los aspectos mencionados en estas conclusiones para la expansión del **Max-Rhythmatist**.

## BIBLIOGRAFIA

### LIBROS:

- **Abromont, Claude, et al.** *Teoría de la Música, una guía*. México: FCE, 2005. Págs. 220-232, 273.
- **Antokoletz, Elliot.** *Twentieth-century music*. EUA: Prentice Hall, 1998. Págs. 218-228, 370-373.
- **Chomsky, Noan.** *Estructuras Sintácticas*. México: Siglo XXI Editores, 2004. Págs 33-41.
- **Cooper, Grosvenor W. y Meyer, Leonard B.** *Estructura rítmica de la música*, España: Idea Books, S.A., 2000. Págs. 9-23.
- **Dewey, John.** *Art as experience*. EUA: Penguin Group, 1980. Pág 61.
- **Dowling, W.J y Harwood, D.L.** *Music Cognition*. EUA: Academic Press, 1986. Págs. 29-73.
- **Hamilton, Andy.** *Aesthetics and Music*. Reino Unido: Continuum, 2007. Págs: 60-72.
- **Handel, Stephen.** *Listening: An introduction to the perception of auditory events*. EUA: MIT Press, 1989. Pág. 597.
- **Lerdahl, Fred y Jackendoff, Ray.** *A generative theory of tonal music*. EUA: MIT Press, Cambridge, Mass, 1983. Págs. 17, 368.
- **Martínez Soques, Fernando.** *Método de canto gregoriano*. España: Editorial Pedagógica, 1943. Págs. 46, 192.
- **Messian, Olivier.** *Technique de mon langage musical*. Francia: Alphonse Leduc et Cie, Éditions Musicales, 1944. Págs. 16, 17.

**MAX-RHYTHMATIST**

- **Meyer, Leonard B.** *Emotion and meaning in music*. EUA: University of Chicago Press, 1956. Págs. 103, 307.
- **Nierhaus, Gerhard.** *Algorithmic Composition. Paradigms of Automated Music Generation*. Alemania: Springer-Verlag/Wien, 2009. Págs. 157-184, 259-273.
- **Novaro, Augusto.** *Sistema natural de la música*. México: [Augusto Novaro], 1951. Págs. 27-55.
- **Platón.** *Las leyes*. España: Ediciones Akal, 1988. Pág: 96.
- **Roads, Curtis.** *The Computer Music Tutorial*. EUA: MIT Press, 1996. Págs. 819-852
- **Rowe, Robert.** *Machine Musicianship*. EUA: MIT Press, 2001. Págs. 277-310.
- **Rowe, Robert.** *Interactive Music Systems*. EUA: MIT PRESS, 1993. Pág. 40.
- **Scruton, Roger.** *The Aesthetics of Music*. Reino Unido: Oxford University Press, 1999. Págs. 45-56.
- **Sorrell, Neil y Narayan, Ram.** *Indian music in performance: A practical introduction*. Reino Unido: Manchester University Press, 1980. Pág. 190.
- **Wesberg, Arthur.** *Performing twentieth century music*. EUA: Yale University Press, 1993. Págs. 3-69.
- **Whaley, Garwood, et al.** *Rhythmic patterns of contemporary music*. EUA: J. R. Publications, 1974. Págs. 3-70.
- **Woodrow, H.** *Time Perception en Handbook of Experimental Psychology* (Editor S.S. Stevens). EUA: Wiley and Sons, 1951. Págs. 1224-1236.
- **Xenakis, Iannis.** *Formalized Music: Thought and Mathematics in Composition*. EUA: Pendragon Press. 1992. (Revisited Edition). Págs. 131-154.

### MAX-RHYTHMATIST

- **Yeston, Maury.** *The stratification of musical rhythm*. EUA: Yale University Press, 1976. Pág. 155.

### **REVISTAS Y PUBLICACIONES:**

- **Boghossian, Paul.** "On hearing the music in Sound: Scruton on Musical Expression" en *Journal of Aesthetics and Art Criticism*. EUA: Temple University, 2002. Págs: 49-55.
- **Budd, Malcom.** "Musical Movement and Aesthetic Metaphors" en *British Journal of Aesthetics*. Reino Unido: Oxford Journals, 2003. Págs: 209-223.
- **Clark, E.F.** "Structure and expression in Rhythmic Performance" en *Musical structure and Cognition*. Reino Unido: Academic Press, 1985. Págs: 209-237
- **Clayton, Martin.** "Free Rhythm: Ethnomusicology and the Study of Music without Metre" en *Bulletin of the School of Oriental and African Studies*. Reino Unido: University of London, 1996. Págs: 323-332.
- **Collins, Nick.** "Generative music and laptop performance" en *Contemporary Music Review* 22(4). Reino Unido: Taylor & Francis Ltd, 2003. Págs. 67-79.
- **Cowan, Nelson.** "On short and long auditory stores" en *Psychological Bulletin* No. 96. EUA: American Psychological Association, 1984. Págs: 341-370.
- **Costello, B., Muller, L., Amitani, S. y Edmonds, E.** "Understanding the experience of interactive art: Iamascope in beta\_space" en *Interactive Entertainment (IE2005)*. Australia: Creativity and Cognition Studios Press, 2005. Págs. 49-55.
- **Dobrian, C y Koppelman, D.** "The 'e' en nime: musical expression with new computer interfaces" en *NIME '06: Proceedings of the 2006 conference on New interfaces for musical expression*. Francia: IRCAM, 2006. Págs. 277-282.

**MAX-RHYTHMATIST**

- **Fels, Sidney.** "Designing for intimacy: creating new interfaces for musical expression" en *Proceedings of the IEEE 92(4)*. Canadá: IEEE, 2004. Págs. 672-685.
- **Fels, Sidney, Gadd, A. y Mulder, A.** "Mapping transparency through metaphor: Towards more expressive musical instruments" en *Organised Sound 7(2)*. EUA: Cambridge University Press New York, 2002. Págs. 109-126
- **Folio, Cynthia.** "An analysis of polyrhythm in selected improvised jazz solos" en *Concert music, rock, and jazz since 1945: Essays and analytical studies*. Págs, 103-134. EUA: University of Rochester Press, 1995. Págs: 103-134.
- **Fraisse, Paul.** "Rhythm and tempo" en *The psychology of music*. Reino Unido: Academic Press, 1982. Págs. 149-180.
- **Goto, Suguru.** "Virtual musical instruments: Technological aspects and interactive performance issues" en *Trends in Gestural Control of Music*. Francia: IRCAM, 2000. Págs. 217-230.
- **Jordà, Sergi.** "Instruments and players: Some thoughts on digital lutherie" en *Journal of New Music Research No. 33(3)*. EUA: Routledge, 2004. Págs. 321-341.
- **Jordà, Sergi.** "Fmol: Toward user-friendly, sophisticated new musical instruments" en *Computer Music Journal 26(3)*. EUA: MIT Press, 2002. Págs. 23-39.
- **Large, Eduard W. y Kolen, John F.** "Resonance and the perception of musical meter" en *Connection science, número 6*. EUA: Taylor & Francis, 1994. Págs. 177-208.
- **Lee C.S. y Longuet-Higgins H.C.** "The Rhythmic Interpretation of Simple Musical Sequences: Towards a Perceptual Model" en *Musical Structure and Cognition*. Reino Unido: Academic Press, 1985. Págs. 53-69.

**MAX-RHYTHMATIST**

- **Laske, Otto.** "Composition theory in koenig's project one and project two" en *Computer Music Journal*, 5(4). EUA: MIT Press, 1981. Págs: 84-93.
- **Palmer, Caroline y Krumhansl, Carol. L.** "Pitch and temporal contributions to musical phrase perception-effects of harmony, performance timing, and familiarity" en *Perception and Psychophysics No. 41*. EUA: Psychonomic Society, 1987. Págs. 505-518.
- **Papadopoulos, George y Wiggins, Geraint.** "AI methods for algorithmic composition: A survey, a critical view and future prospects" en *Proceedings of the AISB '99 Symposium on Musical Creativity*. Reino Unido: University of Edinburgh, 1999. Págs. 1-8.
- **Parncutt, Richard.** "The perception of pulse in musical rhythm" en *Action and Perception in Rhythm and Music*, volumen 55. Suecia: Royal Swedish Academy of Music, Stockholm, 1994. Págs. 127-138.
- **Povel, Dirk-jan y Okkerman, Hans.** "Accents in equitone sequences" en *Perception and Psychophysics No. 30*. EUA: Psychonomic Society, 1981. Págs. 565-572.
- **Povel, Dirk-jan y Okkerman, Hans.** "Perception of temporal patterns" en *Music Perception No. 2*. EUA: University of California Press, 1985. Págs. 441-440.
- **Puckette, Miller.** "A divide between Compositional and Performative aspects of Pure Data" en *Bang Pure Data* (based on the First International Pure Data-Convention 2004, Graz/Austria). Austria: Wolke Verlag, Hofheim, 2006. Págs. 143-152.
- **Sandoval, Carlos.** "Conlon + tiempo= Nancarrow" en *Pauta No.50-51*. México: Cenidim-INBA, 1994. Págs. 148-175.
- **Schnell, N. y Battier, M.** "Introducing composed instruments, technical and musical implications" en *New Interfaces for Musical Expression (NIME-02)*. Reino Unido: University



### MAX-RHYTHMATIST

of Limerick, Department of Computer Science and Information Systems, 2002. Págs. 138-142.

- **Seifert, Udo.** "Time constraints and cognitive modelling" en *Proceedings of the First International Conference on Cognitive Musicology*. Finlandia: Department of Music, University of Jyväskylä, 1993. Págs: 288-299.
- **Seifert, Udo, Olk. F y Schneider A.** "On rhythm perception: Theoretical issues, empirical findings" en *Journal of New Music Research No. 24*. EUA: Routledge, 1995. Págs: 164-195.
- **Schick, Steven.** "Developing an interpretative context: Learning Brian Ferneyhough's Bone Alphabet" en *Perspectives of New Music, Vol. 32, No. 1*. EUA: University of Washington, 1994. Págs. 132-153.
- **Supper, Martin.** "A few remarks on Algorithmic Composition" en *Computer Music Journal No. 25(1)*. EUA: MIT Press, 2001. Págs: 48-53.
- **Tanguiane, Andranick S.** "Artificial perception and music recognition" en *Lecture notes in artificial intelligence No. 746*. EUA: Springer-Verlag, 2007. Pág 89.
- **Wessel, D y Wright, M.** "Problems and prospects for intimate musical control of computers" en *Computer Music Journal No. 26(3)*. EUA: MIT Press, 2002. Págs. 11-22
- **Widmer, Gerhard.** "Learning Expressive Performance: The Structure-Level Approach" en *Journal of New Music Research No. 27*. EUA: Routledge, 1996. Págs: 179-205.

### **SITIOS DE INTERNET:**

- **[http://clam-project.org/wiki/Frequently\\_Asked\\_Questions](http://clam-project.org/wiki/Frequently_Asked_Questions).** "The CLAM project", verificado el 1 de Marzo de 2011.
- **<http://cm.bell-labs.com/cm/cs/who/dmr/chist.html>.** "The Development of C Language",

**MAX-RHYTHMATIST**

verificado el 1 de Marzo de 2011.

- [http://cmr.soc.plymouth.ac.uk/publications/Gimenes\\_Glasgow\\_def.pdf](http://cmr.soc.plymouth.ac.uk/publications/Gimenes_Glasgow_def.pdf). "Proyecto Intelligent Rhythm Generator", verificado el 1 de Marzo de 2011.
- <http://cnmat.berkeley.edu/> "Página oficial del CNMAT", verificado el 1 de Octubre de 2011.
- <http://commonmusic.sourceforge.net/>. "Sitio Oficial de Common Music", verificado el 1 de Septiembre de 2011.
- <http://csounds.com/about>. "Página oficial de CSound", verificado el 1 de Marzo de 2011.
- <http://cycling74.com/products/maxmspjitter/>. "Página de Cycling 74", verificado el 1 de Marzo de 2011.
- [http://es.wikibooks.org/wiki/Programaci%C3%B3n\\_en\\_Java](http://es.wikibooks.org/wiki/Programaci%C3%B3n_en_Java). "Programación en Java", verificado el 1 de Marzo de 2011.
- <http://iua.upf.edu/> "Página oficial del Instituto Universitario del Audiovisual", verificado el 1 de Octubre de 2011.
- <http://jmusic.ci.qut.edu.au/>. "The JMusic Library", verificado el 1 de Marzo de 2011.
- <http://puredata.info/>. "Página oficial de Pure Data", verificado el 1 de Marzo de 2011.
- <http://recherche.ircam.fr/equipes/repmus/RMPapers/CMJ98/>. "Computer Assisted Composition at IRCAM: PatchWork and OpenMusic", verificado el 1 de Septiembre de 2011.
- <http://repmus.ircam.fr/openmusic/home>. "Sitio oficial de OpenMusic", verificado el 1 de Septiembre de 2011.
- <http://supercollider.sourceforge.net/>. "Página de Supercollider", verificado el 1 de Marzo de 2011.

*MAX-RHYTHMATIST*

- <http://www.behringer.com/ES/Products/UMX25.aspx>. "Página de productos Behringer", verificado el 1 de Septiembre de 2011.
- <http://www.cenart.gob.mx/html/cmmp.html> "*Página del Centro Multimedia del CENART*", verificado el 1 de Octubre de 2011.
- <http://www.cmmas.org/> "Página oficial del CMMAS", verificado el 1 de Octubre de 2011.
- <http://www.essl.at/works/rtc.html#abs>. "Página oficial de Karlheinz Essl", verificado el 1 de Marzo de 2011.
- <http://www.haskell.org/haskellwiki/Haskore>. "Sitio oficial de Haskore", verificado el 1 de Septiembre de 2011.
- <http://www.ircam.fr/> "Página oficial del IRCAM", verificado el 1 de Octubre de 2011.
- [http://www.maxobjects.com/?v=libraries&id\\_library=29](http://www.maxobjects.com/?v=libraries&id_library=29). "Base de datos de objetos externos de Max/MSP", verificado el 1 de Marzo de 2011.
- [http://www.symboliccomposer.com/page\\_main.shtml](http://www.symboliccomposer.com/page_main.shtml). "Sitio oficial de Symbolic Composer", verificado el 1 de Septiembre de 2011.
- <http://www.upf.edu/smc/> "Página de la Maestría en Cómputo Musical y Sonido de la Universidad Pompeu Fabra", verificado el 1 de Octubre de 2011.