



## **UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

Programa de Maestría y Doctorado en Música

FACULTAD DE MÚSICA

Intersecciones entre la Nueva Media, Composición Asistida por Ordenador y Obra Pre-informática de Xenakis: Implementaciones de Analogique A y CAO en Supercollider

TESIS

QUE PARA OPTAR POR EL GRADO DE:

MAESTRÍA EN MÚSICA (TECNOLOGÍA MUSICAL)

PRESENTA:

JORGE EDUARDO GÓMEZ ELIZONDO

TUTOR:

DR. ROBERTO MORALES MANZANARES  
(UNIVERSIDAD DE GUANAJUATO)

México, D. F., a enero del 2017



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.



*A mis padres y a mi Mango.*



# ÍNDICE

AGRADECIMIENTOS.....	7
<b>CAPÍTULO 1.....</b>	<b>10</b>
<b>INTRODUCCIÓN .....</b>	<b>10</b>
1.1 PLANTEAMIENTO Y OBJETIVOS PRINCIPALES.....	19
1.2 DESCRIPCIÓN DE PROYECTO PRÁCTICO .....	21
1.3 ANTECEDENTES .....	23
1.4 ESTRUCTURA GENERAL .....	25
<b>CAPÍTULO 2.....</b>	<b>28</b>
<b>ESTUDIOS DE SOFTWARE, NUEVA MEDIA Y.....</b>	<b>28</b>
<b>COMPOSICIÓN ALGORÍTMICA .....</b>	<b>28</b>
2.1 ESTUDIOS DE SOFTWARE Y NUEVA MEDIA.....	28
2.2 TENDENCIAS Y PRINCIPIOS DE LA NUEVA MEDIA SEGÚN MANOVICH.....	32
2.2.1 REPRESENTACIÓN NUMÉRICA.....	33
2.2.2 MODULARIDAD.....	40
2.2.3 AUTOMATIZACIÓN .....	49
2.2.4 VARIABILIDAD .....	63
2.2.5 TRANSCODIFICACIÓN .....	68
<b>CAPÍTULO 3.....</b>	<b>71</b>
<b>OBRAS ORIGINALES .....</b>	<b>71</b>
3.1 OBRAS IV Y VI.....	72
3.1.1 DETALLES DE IMPLEMENTACIÓN DE <i>IV</i> Y <i>VI</i> .....	73
3.1.2 ANÁLISIS RETROSPECTIVO DE LAS OBRAS <i>IV</i> Y <i>VI</i> .....	86
3.2 ANAMNESIS Y 27 DE A 40 GRADOS.....	93
3.3.1 DETALLES DE IMPLEMENTACIÓN .....	95
3.2.2 INTERVENCIÓN DE ANAMNESIS.....	114
3.2.3 ANÁLISIS RETROSPECTIVO DE ANAMNESIS Y 27 DE A 40 GRADOS.....	124
3.3 AUTOPO(I)ÉTICA .....	128
3.3.1 DETALLES DE IMPLEMENTACIÓN EN AUTOPO(I)ÉTICA.....	128
3.3.2 ANÁLISIS RETROSPECTIVO <i>AUTOPO(I)ÉTICA</i> .....	135
<b>CAPÍTULO 4.....</b>	<b>141</b>
<b>ANÁLISIS DE ANALOGIQUE A OBRA PRE-INFORMÁTICA DE XENAKIS: HACIA UNA PRE- COMPOSICIÓN ASISTIDA POR ORDENADOR .....</b>	<b>141</b>
4.1 MODELOS DE MARKOV .....	142
4.2 TEORIZACIÓN DE MÚSICA ESTOCÁSTICA MARKOVIANA EN FORMALIZED MUSIC.....	146
4.2.2 DESCRIPCIÓN DEL MECANISMO MTPZ. ....	154
4.3 FASES ESTACIONARIAS Y PERTURBACIONES.....	163
4.4 ANÁLISIS POR FASES ESTACIONARIAS Y PERTURBACIONES .....	169

4.4.1 FASE ESTACIONARIA E .....	169
4.4.2 FASE DE PERTURBACIÓN PA0 COMPASES 16 - 30 .....	190
4.4.3 PERTURBACIÓN PA' .....	191
4.4.4 FASE PC' COMPASES 65 -80 .....	193
4.4.5 FASE PC0 COMPASES 80 – 95 .....	195
4.4.6 FASE PB0 COMPASES 95 – 109.....	196
4.4.7 FASE PB' COMPASES 110-124.....	197
4.5 FORMA GENERAL.....	199
4.6. HACIA UNA PRE-COMPOSICIÓN ASISTIDA POR ORDENADOR EN ANALOGIQUE A.....	204
<b>CAPÍTULO 5.....</b>	<b>215</b>
<b>COMENTARIOS FINALES Y FUTUROS PROYECTOS.....</b>	<b>215</b>
5.1 FUTUROS PROYECTOS.....	218
<b>APÉNDICE I.....</b>	<b>219</b>
PARTITURAS OBRAS ORIGINALES.....	219
<b>APÉNDICE II .....</b>	<b>259</b>
CÓDIGO FUENTE FOMUS2 Y OBRAS .....	259
<b>CÓDIGO FUENTE FOMUS2 .....</b>	<b>260</b>
<b>BREVE EJEMPLO FOMUS2 .....</b>	<b>269</b>
<b>CÓDIGO FUENTE IV (PIANO).....</b>	<b>273</b>
<b>CÓDIGO FUENTE ANAMNESIS.....</b>	<b>283</b>
<b>CÓDIGO FUENTE ANAMNESIS INTERVENCIÓN.....</b>	<b>304</b>
<b>CÓDIGO FUENTE AUTOPO(I)ÉTICA.....</b>	<b>335</b>
<b>APÉNDICE III.....</b>	<b>363</b>
DATOS ANÁLISIS POR FASES .....	363
<b>APÉNDICE IV.....</b>	<b>374</b>
<b>APÉNDICE V .....</b>	<b>444</b>
EJEMPLO DE FASE ESTACIONARIA E POR IMPLEMENTACIÓN.....	444
<b>BIBLIOGRAFÍA .....</b>	<b>448</b>

## AGRADECIMIENTOS

A mi tutor principal el Dr. Roberto Morales Manzanares por brindarme su invaluable enseñanza, apoyo, confianza y amistad.

A todos los sinodales involucrados por todas sus revisiones y recomendaciones que hicieron de este un mejor trabajo.

A mi maestro de composición de Licenciatura el Dr. Ricardo Martínez Leal por sus consejos, apoyo y amistad.

A todos mis maestros y compañeros de posgrado, con quienes recorrí y de quienes aprendí mucho en este grato camino.

A la Universidad Nacional Autónoma de México, a la Facultad de Música y al departamento de Posgrado en Música por la oportunidad y privilegio de estudiar en esta máxima casa de estudios y todo el apoyo y las enseñanzas que me brindaron.

Al ensamble Liminar, al Cuarteto de Cuerdas Cromano, a Juanmanuel Flores por su trabajo y entrega en la interpretación de las obras propuestas en este trabajo.

A Eduardo Caballero y al Centro de Compositores de Nuevo León.

A mis padres por su inagotable paciencia, amor y apoyo.

A Natalia Neira Nieto por todo su amor, paciencia, fuente de ánimo y motivación, por siempre brindarme su luz y alegría, y por el jardín que cultivamos.



*“sonic discourse is nothing but a  
perpetual fluctuation of entropy in all its forms.”*

*Xenakis*



# CAPÍTULO 1

## INTRODUCCIÓN

*“colorless green ideas sleep furiously”*

—Noam Chomsky

La composición algorítmica se podría definir como el proceso de crear música<sup>1</sup> concebida a partir de la construcción de esquemas y procesos de mapeo o transcodificación de algún sistema o proceso formalizado a parámetros musicales. Estos esquemas o modelos generalmente son tomados de diversas disciplinas científicas, apropiadas de alguna manera y a veces modificadas tanto para ser de utilidad en el campo musical, como también, modificando y reordenando resultados para que se ajusten de manera más satisfactoria al concepto sónico que tiene el compositor, dando lugar a una compleja dialéctica entre formalización, intuición, creatividad y/o contexto.

La computadora se ha convertido en el medio ideal para construir estos sistemas y llevar a cabo las operaciones necesarias dentro de un esquema de composición algorítmica, regularmente dentro de entornos de programación y/o paquetes de software especializados con lo cual ya se entraría al terreno de la composición

---

<sup>1</sup> A lo largo de este trabajo cuando se haga referencia a la música se estará haciendo alusión a la música clásica de tradición occidental, al menos de que se indique lo contrario.

asistida por ordenador<sup>2</sup> (CAC por sus siglas en inglés, *Computer Assisted Composition*).

Entonces partiendo de la definición anterior, podríamos hacer la suposición de que a la definición de la composición asistida por ordenador solamente le agregaríamos, como está implícito en su nombre, el uso de la computadora como herramienta mediadora para llevar a cabo los procesos realizados en la composición algorítmica, es decir, la computadora cumpliendo una función de “remediación” o como una “máquina de remediación” (Bolter y Grusin, 2000), en la que la computadora es:

[...] una máquina que representa expertamente una variedad de media más antigua  
 [...] Bolter y Grusin definen la remediación como ‘la representación de un medio en otro’. De acuerdo a su argumento, la nueva media siempre *remedia* a la media más antigua y por lo tanto no deberíamos de esperar que las computadoras funcionen de alguna manera distinta. (Manovich, 2013: 58-59, traducción propia.)

A pesar de que en efecto sí puede concebirse como una “máquina de remediación”, al introducirse el elemento de software, más aún, entornos de programación, entran en juego una serie de características y procesos que forman parte de la informática y que tienen el potencial de inscribirse de diversas maneras en los tres niveles del hecho musical, siendo estos *poiético*, *estésico* y *neutro*. Correspondiendo esto *grosso modo* a todos los procesos y variables dentro de la música, siendo articulada como una forma simbólica, vinculadas a la creación, la partitura o trazo y la recepción de la obra respectivamente.

---

<sup>2</sup> En español se podrían utilizar las siglas CAO (Composición Asistida por Ordenador).

A partir de la segunda mitad del siglo XX la tecnología y la informática han permeado de manera paulatina, en mayor o menor medida virtualmente cada aspecto de nuestra vida profesional y cotidiana. Con el surgimiento y desarrollo de software nuestras herramientas de comunicación o de trabajo se han extendido, inmersas en un mismo espacio virtual cuando antes era inimaginable, incluso creando medios completamente nuevos o *born digital*; como afirma Manovich en su libro *Software Takes Command*, se podría decir que se ha agregado una nueva dimensión a nuestra sociedad, vivimos en una sociedad de software. (Manovich, 2008 y 2013)

Uno de los ejes sobre los cuales gira este trabajo de investigación es la noción de que el medio en el cual se formulan ideas musicales, o se ejerce al menos parte del trabajo creativo, tiene la capacidad de influenciar la manera en que son concebidos los elementos que forman un discurso musical, tratándose en esta ocasión específicamente de cómo puede llegar a suceder esto a través de software de programación.

Esta noción no es nada nueva, numerosos estudiosos y compositores han expuesto ya esta particularidad dentro de la música, como por ejemplo la manera en que la notación musical prioriza ciertos parámetros por encima de otros, algunas técnicas compositivas que son claras y evidentes en transcripción pero no auditivamente, como lo son el retrógrado o la inversión melódica, o lo que es más, modelos compositivos como el serialismo integral en el cual se establece una secuencia para cada uno de los parámetros establecidos en la notación.

A su vez, también se parte en este trabajo de la noción de que las ideas musicales tienen el poder de modificar o influenciar la notación según las necesidades y prioridades requeridas, lo cual ha sido quizá más evidente a partir de mediados del siglo XX, como bien lo expresa el compositor Brian Ferneyhough:

La notación (*particularmente* la notación) nos muestra dos facetas: trazable y analizable en términos de su desarrollo histórico, [...] es sin embargo casi imposible separarlo, incluso en principio, de las metas compositivas que se ha propuesto un artista. No es accidente el hecho de que tantos trabajos de las últimas tres décadas sean quizá inmediatamente catalogables en términos de sus características visuales en vez de sus características audibles. (Boros y Toop Eds., 1978 : 2, traducción propia)

Se esté de acuerdo o no con la afirmación de que actualmente la música es más catalogable en términos de notación que en términos audibles, lo cierto es que hoy en día es habitual que los compositores tengan la necesidad de expandir algunos aspectos de la notación mediante distintos métodos (algunos de los cuales son ya relativamente estandarizados). Esto generalmente ocurre cuando se plantea una estética o un discurso sonoro que va más allá del paradigma del binomio de altura y ritmo, ampliamente priorizado en el sistema de notación musical occidental.

El compositor británico Trevor Wishart adopta una postura un tanto más radical en su libro *On Sonic Art* donde sugiere que exclusivamente la transcripción en la música es la que le adhiere la importancia del paradigma altura – ritmo: “Es la transcripción en la música la que determina la importancia de altura, ritmo y duración y no viceversa” (Wishart, 1996: 6, traducción propia) y reafirmando más adelante:

[...] las prioridades de la notación no reflejan prioridades musicales – las crean [...] la notación tiene una construcción orientada a un enrejado; [refiriéndose a los parámetros] hay aspectos fundamentales del sonido [...] que no se pueden transcribir de manera convencional y por lo tanto no son incluidos en la partitura. [...] de hecho la música no tiene que tener esta orientación enrejada [...] Una preocupación con la notación convencional nos puede llevar a la formalización, una situación donde ya no hay un medio experiencial verificable para poner a prueba nuestras teorías de cómo componer música. (Ibíd., 11)

¿Por qué hablar de notación?

Porque considero que el código se incluye dentro del nivel inmanente del hecho musical y evidentemente también dentro del polo poiético por ser parte del proceso creativo. Es una representación simbólica *dinámica*, por ser entre otras cosas ejecutable, de otra representación simbólica que es la partitura, que a su vez es una representación simbólica de la materialización del sonido en un discurso musical dado, dando lugar a un diálogo entre creador y medio y/o notación con un material que tiene elementos *performativos* que adquiere nuevas propiedades y tienen el potencial de ser inscritas en uno o todos los polos del hecho musical.

Por lo tanto si tomamos en cuenta que el código ejecutable en la computadora es una representación simbólica de otra representación (la partitura), hasta cierto punto puede concebirse como parte de la notación *dentro del proceso poiético*, abarcando también el polo inmanente.

En este trabajo de investigación se partirá de la noción que se mencionó al inicio de este apartado, la notación y/o el medio tienen el poder de influenciar los esquemas a partir de los cuales se conciben construcciones musicales y con ello la música que se

está generando, y la música en turno puede obligar a plantear distintas metodologías de transcripción para con ello capturar con mayor eficacia y exponer con mayor claridad las prioridades e intención musical del compositor. Se podría hacer una analogía entre diferencias entre media tradicional y nuevos medios como lo hace Walter Ong en su libro *Orality and Literacy* con la tradición oral y la escritura:

Many of the features we have taken for granted in thought and expression in literature, philosophy and science, and even in oral discourse among literates, are not directly native to human existence as such but have come into being because of the resources which the technology of writing makes available to human consciousness. We have had to revise our understanding of human identity. (Ong, 1982: 1)

Ahora bien, si en la composición asistida por ordenador se pueden construir esquemas musicales representadas en forma de código ejecutable las cuales a su vez arrojan datos que obedecen las reglas de construcción de los algoritmos involucrados, que finalmente representarán parámetros musicales o describirán un comportamiento musical determinado, ¿cuáles son las características, procesos y/o formas que se adhieren al *softwarizarse*<sup>3</sup> la composición algorítmica? ¿Cómo y en qué sentido pueden llegar a manifestarse, modificar o extender alguna idea o medio, e influenciar algunos o todos los niveles del hecho musical?

Son estas preguntas fundamentales del presente trabajo que se buscan responder, como se verá más adelante, principalmente a partir de la praxis desempeñando mi

---

<sup>3</sup> Aún y siendo un anglicismo se ha optado por utilizar el neologismo de “Softwarización” que aparece en los escritos de Lev Manovich al sentir que describe con mayor precisión el fenómeno que su traducción directa ‘informatización’.



labor como compositor explorando la CAO, e insertándolos dentro de un marco de los estudios de software mediante modelos teóricos propuestos por Manovich; como por ejemplo las tendencias que le adjudica a la nueva media que son representación numérica, modularidad, automatización, variabilidad y la transcodificación. Estas son planteadas como características o principios que pueden o no presentar los *objetos*<sup>4</sup> de la nueva media que son expuestas a manera de diferenciación (y también analogías) entre la nueva media y la media tradicional. Buscaré hacer algo similar en el ámbito de la CAO partiendo de las obras compuestas para este trabajo de investigación para después situarlas dentro del modelo tripartito adaptado por Nattiez en su semiología musical para intentar aprehender lo planteado.

Estos principios expuestos por Manovich son categorizados siguiendo una lógica de creciente dependencia, es decir, que los últimos tres principios dependen de los primeros dos, y no son presentadas como un cisma entre media tradicional y media nueva, sino como puntos de partida para poder comprender sus intersecciones y transformaciones, sobre estos principios nos dice:

[Estos principios] Deberían de ser considerados no como leyes absolutas sino como tendencias generales que sigue una cultura que está siendo sometida a una computarización. A medida de que la computarización vaya llegando a capas de

---

<sup>4</sup> Manovich define como *objeto* a “I use the term *new media object*, rather than *product*, *artwork*, *interactive media* or other possible terms, A new media object may be a digital still, digitally composited film, virtual 3-D environment, computer game, self contained hypermedia DVD, hypermedia Web site, or the Web as a whole. The term thus fits with my aim of describing the general principles of a new media that hold true across all media types, all forms of organization, and all scales. *I also use object to emphasize that my concern is with the culture at large rather than with new media art alone.*” (Manovich, 2001: 14, mi énfasis)

cultura cada vez más hondas, estas tendencias serán mayormente evidenciadas. (Manovich, 2001: 27, traducción propia).

Reitero que considero que algunas o todas estas tendencias pueden estar presentes dentro de la composición asistida por ordenador y ser manifestadas en mayor o menor grado tanto en el código ejecutable, en el nivel poético, estético y/o neutro.

Al formular una obra dentro de determinados software de programación donde seguramente existen distintas librerías u objetos de diversas naturalezas y funciones que interactúan entre sí, la música o el acto creativo se puede ver enriquecida o influenciada por esta interacción y diversidad invitando a mezclar algoritmos y técnicas para plantear algún discurso musical. En este proceso es común que sean tomados prestados conceptos de múltiples disciplinas y modelos que son implementados sin importar a fin de cuentas si son 'mal utilizados' o si no se tiene una profunda comprensión (aunque esto podría tener sus ventajas técnicas y filosóficas en su implementación) del algoritmo más que quizá algunos fundamentos básicos y detalles los cuales pudieron haber motivado o inspirado su utilización para establecer una forma o un comportamiento musical determinado, siendo este quizá para algunos el fin de ulterior importancia. Por esta misma razón los datos y construcciones arrojadas por los algoritmos son generalmente manipulados e intervenidos tomando libertades, cambiando algunas de las estructuras formales establecidas en pro del discurso sónico concebido por el compositor, entendida aquí como un proceso de *bricolage*.

Bricolage es un concepto utilizado por el etnólogo y antropólogo francés Claude Lévi-Strauss (1908 - 2009) que adopta el musicólogo griego Makis Solomos (1962)

para aprehender procesos dialécticos de formalización en intuición en el imaginario Xenakiano:

Para darle un nombre en este artículo a estas intervenciones manuales, utilicé la palabra francesa *bricolage*. Lo tomo en el sentido que lo utiliza Claude Lévi-Strauss en 'El Pensamiento Salvaje', con ello buscando elucidar la manera en que específicamente Xenakis lleva a cabo estas intervenciones manuales [...] Con Xenakis, la idea de que su *bricolage* se lleva a cabo en un 'universo instrumental cerrado' [...] [es atinada] si nos limitamos a un sistema formal específico que utiliza si durante su implementación musical no está satisfecho con los resultados que está dando, el no busca alguna otra herramienta mejor adaptada, ni busca su corrección sistémica. El 'se las ingenia con los medios que tiene a su alcance' [Lévi-Strauss, 1962: 31]. Además, este universo instrumental consiste de 'herramientas y materiales heteróclitas' (Solomos, 2005: 15, traducción propia)

Puede aparentar curiosa la decisión de incluir la obra de *Analogique A* de Xenakis dentro de una tesis que incluye estudios de software en su marco teórico, tomando en cuenta que en el momento en que creó la obra (1958) aún no tenía contacto con una computadora, cosa que sucedería hasta el año 1962 en una IBM 7090 donde comenzaría a trabajar sobre las obras de la famosa serie *ST*, retomando y expandiendo ideas y conceptos que había tenido en piezas que ya había compuesto en años precedentes como *Achorripsis* (1957).

Las razones por las cuales decidí incluirla en el presente trabajo, a pesar de la aparente discrepancia, podrían resumirse en que considero que al abordar esta obra puedo encontrar puntos de contacto entre características de la media tradicional y la nueva media, al estar Xenakis en contacto con investigaciones científicas de la época que abarcaban física cuántica que lo acercaría a una concepción cuántica del

sonido (Gabor, Einstein) , termodinámica (Boltzmann), estadística , *teoría de la información* (Shannon), entre otros, aborda aspectos de la composición a partir de conceptos que toma prestados de estas variadas ramas insertándolos dentro de su imaginario sonoro (y visual), además de abordar la composición partiendo de diversas hipótesis y axiomatizaciones como si se tratara de una rama de las ciencias. Construye sistemas y mecanismos con diversas *mapeos* de diversas variables que son sujetas a transformaciones *estocásticas* a parámetros y transformaciones musicales, marcos sistémicos que son homologados a diversas obras, hace distinciones entre *data* y *algoritmo*, propone un método de trabajo que se asemeja al proceso creativo en la CAO<sup>5</sup>, en fin se podría decir que Xenakis hace una *pre composición asistida por ordenador*.

## 1.1 PLANTEAMIENTO Y OBJETIVOS PRINCIPALES

El presente trabajo de investigación se divide primordialmente en las siguientes dos vertientes:

- 1) La creación de obras originales que involucran la composición asistida por ordenador (CAO) principalmente haciendo uso del lenguaje de programación *SuperCollider*<sup>6</sup>. A partir de estas piezas se explorará cómo es que el uso de la tecnología, en este caso específicamente un lenguaje de programación basado en texto ejecutable, puede influenciar y adherir procesos, características o

---

<sup>5</sup> En el capítulo 2 abordaré una analogía entre el proceso creativo encontrado en *“Bricolage Programming in the Creative Arts”* (McLean y Wiggins, 2010) y un método de trabajo propuesto por Xenakis en su libro *Formalized Music* (Xenkis, 1992)

<sup>6</sup> <http://supercollider.github.io>

tendencias que son inherentes al software y a la programación (ubicado dentro de un marco teórico de estudios de software), a alguno o todos los niveles del modelo tripartito propuesto por Jean Molino del hecho musical (Molino, 1995), siendo estos el nivel poiético (el acto de creación), el nivel estésico (*procesos* de recepción) y el nivel neutro (el resultado sonoro, transcripción, partitura), lo cual será ahondado en su sección correspondiente.

- 2) Por otro lado presentaré un análisis de la obra *Analogique A et B* del compositor Iannis Xenakis, que a pesar de ser ampliamente abordada en su libro *Formalized Music* (Xenakis, 1992) y notablemente por el compositor italiano Agostino Di Scipio (1962) en *Formalization and Intuition in Analogique A et B (with some remarks on the historical-mathematical sources of Xenakis)* (Di Scipio, 2005), considero pertinente hacer revisión de la obra por dos razones: por un lado al hacer un análisis preliminar de la partitura encontré ciertas incongruencias con respecto a la formalización e implementación detallada por Xenakis, a partir de lo cual se encontraron datos nuevos que a su vez sirvió de motivación para elaborar una implementación dentro de *SuperCollider*, lo cual permitirá comparar un resultado completamente apegado a su concepción teórica original reconstruyendo múltiples versiones de *Analogique A*. Además esto permitirá explorar qué es lo que involucra una *softwarización* (Manovich, 2001) de esta obra como herramienta metodológica para su análisis, además del potencial de modificar o expandir el código fuente para composiciones originales.

## 1.2 DESCRIPCIÓN DE PROYECTO PRÁCTICO

En seguida me limitaré a exponer una descripción sintetizada del proyecto práctico que se propone, ya que será discutido a profundidad en su sección correspondiente.

He planteado componer tres familias de obras originales a lo largo de la maestría a partir de las cuales analizaré cuáles son los procesos adquiridos al introducirse el elemento informático y cómo pueden influenciar o no el hecho musical, a la vez que se expondrán sus detalles de implementación.

Las primeras obras propuestas se tratan de dos miniaturas que fueron concebidas dentro de *SuperCollider* utilizando máquinas de estado finito: *IV* para flauta, clarinete y piano, y la siguiente obra *VI* para fagot, trombón, piano, viola y contrabajo. Ambas obras tienen una forma y un discurso muy similar por el hecho de ser compuestas a partir de un mismo algoritmo, sin embargo, se puede decir que tienen una evolución sónica propia, al incluir dentro del algoritmo esquemas estocásticos, lo cual hace cada obra única con su símil “reflejo” o “eco ” distorsionado.

A primera vista el resultado seguramente puede asemejarse a un esquema tradicional de partitura, intérpretes y formato de concierto, sin embargo, al ubicar el hecho musical dentro de un marco de estudios de software buscaré evidenciar por qué considero que no es así, analizando los procesos involucrados principalmente dentro del nivel poético.

La siguiente familia de obras comprende *Anamnesis* para violín solo y *27 de a 40 grados* para quinteto de cuerdas. Ambas obras parten de un mismo algoritmo que involucra principalmente máquinas de estado finito, gradientes o *tendency masks* (Koenig, 1971), movimientos brownianos y sistemas de Lindenmayer<sup>7</sup> (sistemas-L). Las dos obras están estrechamente vinculadas al utilizar algoritmos esencialmente iguales y un discurso muy similar. Sucede algo semejante como en *IV* y *VI* en el sentido de que son una especie de reflejo de si mismas, pero ahora con la diferencia de que se puso a prueba el concepto de la obra al hacerla para ensamble y otra para instrumento solista.

Además en *Anamnesis* propongo dos discursos sonoros distintos, por un lado la ejecución tradicional de la obra y por otro una intervención en la cual después de la ejecución (aunque se pueden presentar y se han presentado como formatos independientes) se crea un segundo discurso derivado del primero. Partiendo de la grabación de la interpretación (*in situ*<sup>8</sup> o *in absentia*) se manipulan y transforman tanto el sonido como su desenvolvimiento en el tiempo a partir de procesos automatizados que toman decisiones en base a distintos análisis de propiedades acústicas y/o psicoacústicas<sup>9</sup> de los archivos de audio, dando lugar a una intervención del formato de concierto, y así desenvolverse en un discurso más afín a la de una instalación sonora. Igualmente se analizarán los procesos presentes dentro

---

<sup>7</sup> Todos estos conceptos serán abordados en su apartado correspondiente dentro el capítulo 3.

<sup>8</sup> Propuesta para un trabajo futuro.

<sup>9</sup> Utilizando el plug-in SCMIR (SuperCollider Music Information Retrieval Library) escrito por Nick Collins <http://composerprogrammer.com/code.html>.

del hecho musical y cómo es afectado no solamente el nivel poiético pero también el nivel estésico.

Finalmente en la obra *Autopo(i)ética* para cuarteto de cuerdas propongo un modelo para la generación de partituras en tiempo real. Las partes de cada instrumento son calculadas dentro de los software *SuperCollider*, *Lilypond*<sup>10</sup> y *FOMUS*<sup>11</sup> y visualizadas en formato PDF en la pantalla de cada computadora utilizando el software *Skim*<sup>12</sup>, cada ejecutante tiene en escena una laptop y está leyendo a primera vista.

Cada vez que se ejecuta el programa se da una versión distinta de una pieza y la posibilidad de que el programa tome otras decisiones para formar obras diferentes. Aquí el nivel neutro del cual forma parte la partitura adquiere elementos dinámicos, la partitura no puede existir, entre otras cosas, sin el acto performativo del código. Analizaré igualmente cual es el impacto de la informática sobre el hecho musical en este caso, así como el planteamiento de trabajos a expandir, siendo este otra característica fundamental del software, o como lo cataloga el matemático, científico informático y pionero del *Computer Art* alemán Frieder Nake (1938) una “*aesthetic of the unfinished*”.

### 1.3 ANTECEDENTES

Los antecedentes de la composición algorítmica<sup>13</sup> y sus tempranas manifestaciones pre-informáticas ya han sido exhaustivamente abordadas en una amplia gama de

---

<sup>10</sup> <http://www.lilypond.org/index.es.html>

<sup>11</sup> <http://fomus.sourceforge.net>

<sup>12</sup> <http://skim-app.sourceforge.net/>

<sup>13</sup> Aunque frecuentemente los términos composición algorítmica y composición asistida por ordenador han sido utilizados de manera intercambiable, considero pertinente hacer una



libros, publicaciones académicas y tesis de posgrado, como por ejemplo (Neuhaus, 2010), (Miranda, 2009), (Roads, 1995), (Loy, 2006), (Wang, 2008), por mencionar algunos. Se presentan remontándose desde el siglo XI<sup>14</sup>, pasando por el desarrollo de la notación musical, lo cual permitió entre otras cosas la proliferación de un pensamiento musical paramétrico. Sistemas de composición algorítmica desarrollados por Athanasius Kircher en el siglo XV en su tratado *Musurgia Universalis*<sup>15</sup>, los “juegos musicales de dados” del siglo XVIII; así como también han sido ampliamente cubiertos los antecedentes de la CAO retomando desde conceptualizaciones tempranas de la computadora como Charles Babbage, las primeras especulaciones sobre un software capaz de generar música de manera automatizada por Ada Lovelace, el desarrollo tecnológico, informático y de software desde la segunda mitad del siglo XX que permitieron la proliferación de la tecnología actual y que a la par con la creciente accesibilidad de la computadora ha permitido a su vez en general una mayor “democratización del software” (Manovich, 2013),

---

distinción entre ambas debido a que estrictamente hablando una composición algorítmica no requiere del uso de una computadora. Los compositores se han valido de diversos arquetipos para la construcción de ideas musicales a través de la historia, como por ejemplo el sistema tonal, reglas contrapuntísticas o armónicas de determinada época, el serialismo de la segunda escuela vienesa, el serialismo integral con Pierre Boulez, entre otros, dentro de los cuales se excluye cualquier tipo de software. Bien ciertamente la CAO es por naturaleza algorítmica, sin embargo al agregar la capa informática adquiere nuevas características y procesos que se buscarán elucidar en este trabajo y por lo cual considero necesario hacer la distinción. Más adelante en los capítulos dos y tres se ampliará esta discusión.

<sup>14</sup> Cuando el monje benedictino Guido d'Arezzo en los capítulos XV y VII de su tratado *Micrologus de Disciplina Artis Musicae* desarrolla un sistema para la generación automática de melodías a partir de un texto con el mapeo de diversos componentes de un texto, como por ejemplo de sílabas a notas musicales.

<sup>15</sup> Todos estos ejemplos serán abordados en el capítulo siguiente.

dentro del cual su intersección con el quehacer artístico y musical no es una excepción.

Aprovecho este espacio para aclarar que es por esta razón que he tomado la decisión de omitir un capítulo dedicado específicamente a los antecedentes de la composición algorítmica o de la *CAO*. En su lugar me limitaré a mencionarlos donde considere que sea pertinente para fines ilustrativos, o para exponer trabajos anteriores que estén estrechamente vinculados con la temática en turno en sus respectivos apartados.

#### 1.4 ESTRUCTURA GENERAL

En el capítulo dos introduciré los conceptos básicos del marco teórico en el cual está inserto el presente trabajo relacionándolos directamente con la *CAO*, abordaré procesos, características o tendencias expuestas por el teórico de estudios de software y nuevos medios Lev Manovich, principalmente en *The Language of New Media* (Manovich, 2001) y *Software Takes Command* (Manovich, 2013), así como también por otros conceptos de los estudios de software y nuevos medios que serán articulados en el siguiente capítulo.

Posteriormente en el capítulo tres plantearé las piezas que comprenden la parte práctica del proyecto, por un lado detallando sus respectivas implementaciones en el lenguaje de programación y paquetes de software involucrados, y por el otro abordándolas desde un marco teórico de los estudios de software y ubicándolas dentro del hecho musical, buscando con esto elucidar cómo es que considero que se manifiestan las tendencias o procesos expuestos en el capítulo anterior.

En el capítulo cuatro abordaré el análisis de *Analogique A* de Xenakis, introduciendo los detalles analíticos expuestos por el mismo compositor en *Formalized Music* y en el artículo *Formalization and Intuition in Analogique A et B* de Agostino Di Scipio. Posteriormente explayaré los nuevos datos encontrados a partir de un análisis minucioso de la obra y su respectiva implementación desarrollada dentro de SuperCollider. También discutiré lo que involucra la *softwarización* de una obra que no fue originalmente concebida dentro de un lenguaje de programación, las propiedades que adquiere, posibilidades que ofrece y cómo nos ayuda en este caso a comprender mejor esta obra.

En los apéndices y anexos correspondientes se podrán encontrar:

- el código fuente, partitura e instrucciones de ejecución o montaje de todas las obras compuestas para este trabajo;
- código fuente de la implementación de *Analogique A*;
- fase estacionaria generada con implementación de *Analogique A*;
- tablas con los datos del análisis de la obra.

Me parece importante constatar que el presente trabajo está sumergido en un marco teórico de estudios de software y se auxilia de algunas de las herramientas semiológicas y los conceptos básicos del modelo tripartito de Jean Molino expuestas por Jean Jaques Nattiez en su libro *Music and Discourse: Towards a Semiology of Music* (Nattiez, 1990), los conceptos y terminología semiológica será abordada en su momento cuando lo considere pertinente, ya que no se trata de un análisis

semiológico exhaustivo *per se* y no considero que sea necesario incluir un capítulo para detallar esta parte del marco teórico.

Por el momento basta con afirmar que lo utilizo para elucidar tendencias y procesos que de otra manera podrían pasar inadvertidas, así como también me parece oportuno siendo el código fuente de programación una forma simbólica, en este caso específico ligado a su vez a otra forma simbólica que es la música.

## CAPÍTULO 2

### ESTUDIOS DE SOFTWARE, NUEVA MEDIA Y COMPOSICIÓN ALGORÍTMICA

*“Welcome to the world of permanent change  
—the world that is now defined not by heavy industrial  
machines that change infrequently, but by software that is always in flux”  
—Lev Manovich*

En este capítulo quisiera exponer de manera lacónica algunas de las ideas principales que hay detrás de los *estudios de software* esperando con ello elucidar los procesos o tendencias que Lev Manovich le adjudica a los objetos de la nueva media. Considero que estas tendencias se manifiestan en la composición asistida por ordenador de diversas maneras y en distintos niveles que pueden ser evidenciados bajo el preámbulo de los estudios de software para después integrarlos tanto al análisis de las obras compuestas para este trabajo como también al análisis de *Analogique A* de Xenakis.

#### 2.1 ESTUDIOS DE SOFTWARE Y NUEVA MEDIA

A partir de la actual proliferación tecnológica e informática y la manera en que esto ha permeado la cultura contemporánea en diversos aspectos y niveles, recientemente han surgido una serie de disciplinas teóricas y esfuerzos que proponen metodologías interdisciplinarias para analizar distintos objetos y

procesos culturales partiendo del software (*software studies*), plataformas (*platform studies*), código fuente (*critical code studies*) desde un marco que propone vincular conocimientos teóricos, procesos y estructuras propias de las ciencias computacionales con las de teorías críticas y las humanidades para así encontrar nuevas maneras de aprehender fenómenos y *objetos* culturales.

Dadas las crecientes intersecciones entre las artes y humanidades con diversos software y lenguajes de programación de alto nivel dentro y fuera de la academia, se ha encontrado la necesidad de expandir la comprensión del software, no sólo en un sentido tecnológico – instrumental, sino también buscando cómo es que diversos discursos o manifestaciones culturales y sociales pueden ser afectados o moldeados a partir de procesos involucrados dentro del software y cómo son manifestadas e implementadas en el software en turno, ya que el software mismo es también una construcción cultural. Uno de los pioneros en los estudios de software, Mathew Fuller, bien expresa esta necesidad en el prefacio del libro *10 PRINT CHR\$(205.5+RND(1)); : GOTO 10*:

Software is deeply woven into contemporary life—economically, culturally, creatively, politically—in manners both obvious and nearly invisible<sup>16</sup>. Yet while much is written about how software is used, and the activities that it supports and shapes, thinking about software itself has remained largely technical for much of its history. Increasingly, however, artists, scientists, engineers, hackers, designers, and scholars in the humanities and social sciences are finding that for the questions they face, and the things they need to build, an expanded understanding of software is necessary. For such understanding they can call upon a strand of texts in the history of computing and new media, they can take part in the rich implicit culture of software,

---

<sup>16</sup> Mi énfasis.

and they also can take part in the development of an emerging, fundamentally transdisciplinary, computational literacy. These provide the foundation for Software Studies. (Fuller, 2013: xi)

El término de estudios de software aparece por primera vez en el año 2001 dentro del seminal libro de Manovich *The Language of New Media*, donde expresa:

New media calls for a new stage in media theory whose beginnings can be traced back to the revolutionary works of Robert Innis and Marshall McLuhan of the 1950s. To understand the logic of new media we need to turn to computer science. It is there that we may expect to find the new terms, categories and operations that characterize *media that became programmable*<sup>17</sup>. From media studies, we move to something which can be called software studies; from media theory—to software theory. (Manovich, 2001: 59)

Una de las principales contribuciones de Manovich en su libro es el de abordar los estudios de media y distintos *objetos* culturales a partir del software, auxiliado con conceptos de las ciencias computacionales vinculándolas a las humanidades a la vez que explora como ha transformado nuestra vinculación con los media o los nuevos medios a través del software. Esto se asemeja de cierta manera a la célebre postulación del filósofo de teorías de la comunicación canadiense Marshall McLuhan (1911 - 1980) 'el medio es el mensaje' en *Understanding Media: The Extensions of Man* (McLuhan, 1964) donde aseveraba que el medio se inscribía en el mensaje y le daba forma, y es el mismo medio el que debe de ser el foco de estudio para tener una mayor comprensión del contenido.

Por otro lado Manovich también plantea que al abordar el estudio a partir del software, se habilita analizar objetos que fueron concebidos dentro de la

---

<sup>17</sup> Mi énfasis.

computadora que vistos superficialmente no aparentan reflejar cambio alguno comparado con medios tradicionales, pero que dentro de su 'hábitat digital' sí tienen el potencial de ser hondamente transformados:

Si una fotografía digital es convertida en un objeto físico en el mundo –una ilustración en una revista, un cartel en la pared, un estampado en una camiseta– funciona de la misma manera que sus predecesor. Pero si dejamos la misma fotografía dentro de su nativo medio, la computadora, [...] la cual permite al usuario editar la fotografía y pasarla a otros dispositivos o el internet–puede funcionar de maneras en las cuales, desde mi punto de vista, la hacen radicalmente distintos de su equivalente tradicional. (Manovich, 2008: 37, traducción propia)

Esta noción es de primordial importancia dentro de este trabajo, ya que como veremos más adelante, se presenta el caso en el cual alguna obra, en su gestación, bajo la forma de código ejecutable posee atributos de un objeto de nueva media y que ya en el resultado o formato final es manifestada como media tradicional. Por ejemplo una partitura que a simple vista bien pudo haber sido concebida sin utilizar algún lenguaje de programación en su concepción, puede presentar transformaciones significativas. El código con el que se hizo la pieza aún sin materializarse tiene diversas maneras de expandirse hacia otros formatos a la vez que involucra procesos que terminan teniendo influencia sobre el acto creativo, por otro lado, el código ejecutable que está ligada al nivel neutro de la pieza, puede fungir como documento histórico analizable que revela algunos aspectos de la construcción de la obra y puede ser analizado como un texto, como sucede con las herramientas metodológicas de los *Critical Code Studies* (CCS) que hacen un exégesis del código para abordar distintas temáticas partiendo del código fuente mismo.



Ahora bien, que algún objeto haya sido construido dentro de algún software no necesariamente involucra algún cambio ontológico significativo o que se haya producido algún cisma entre media tradicional y nuevos medios, sin embargo, considero que definitivamente tiene al menos el potencial de ser transformado de alguna manera dentro de su manifestación o la manera en que nos vinculamos a él, aunque quizá en algunas instancias sea esto solamente visible y analizable dentro de su representación digital y con impacto perceptible bajo escrutinio.

Después de la aparición de *The Language of New Media* ha habido una expansión considerable de literatura de estudios de software diseminándose en distintos campos abordado por una vasta variedad de disciplinas, Mathew Fuller, destacado teórico y pionero de los estudios de software en el prefacio del ya mencionado resume de la siguiente manera :

the ability to mix the formalized with the more messy—non-mathematical formalisms, linguistic, and visual objects and codes, events occurring at every scale from the ecological to the erotic and political—which gives computation its powerful effects, and which folds back into software in its existence as culture. (Fuller, 2010: 5)

## 2.2 TENDENCIAS Y PRINCIPIOS DE LA NUEVA MEDIA SEGÚN MANOVICH

A continuación expondré una serie de ‘principios’ o tendencias que propone Manovich que están presentes dentro de los objetos de nueva media, siendo estos representación numérica, modularidad, automatización, variabilidad y transcodificación. Los tomo como punto de partida para abordar las obras compuestas dentro de la parte práctica de este trabajo en el siguiente capítulo. En este apartado expondré como considero que estos principios permiten evidenciar

paralelismos históricos con procedimientos algorítmicos en la concepción musical a la vez que sirve de herramienta para comprender cómo es que se ha inscrito el software en ellas.

Cabe mencionar que estos no son principios universales o leyes absolutas, sino “tendencias generales de una cultura que está siendo sujeta a una computarización.” Y que “Si colocamos a la nueva media dentro de una perspectiva histórica más amplia, veríamos que muchos de los principios no son exclusivos de la nueva media, pueden ser encontrados en tecnología de medios más antiguos de igual manera.” (Manovich, 2001: 27, 50, traducción propia). Por lo tanto pueden estar presentes en medios que involucran o no el ordenador, sin embargo, tienen al menos el potencial para tener diferencias significativas lo cual puede o no repercutir en cómo se desenvuelve el objeto dentro de algún contexto específico.

Además estos principios no son excluyentes, es decir, un objeto puede presentar todos o solamente algunos de estos principios, por consiguiente algunos de los ejemplos que expondré podrán ser incluidas en varias categorías. También es importante notar que es posible que algunos de los principios dependen de la existencia de otro(s).

### 2.2.1 REPRESENTACIÓN NUMÉRICA

Todos los objetos de nueva media, ya sea el fuente código de un programa, fotografías digitales, o la importación de algún medio análogo a formato digital, tienen en común que son representados numéricamente dentro del ordenador. Esto repercute en el objeto principalmente de dos maneras:

1. Un objeto de nueva media puede ser descrito de manera formalizada.
2. El objeto de nueva media al ser representado numéricamente, puede ser sujeto a manipulación algorítmica, “la media se torna programable” (Manovich, 2001: 27)

Cuando objetos de nueva media son creadas directamente desde la computadora su modo de existencia en un medio digital ya involucra una posible reducción de su representación a datos numéricos siendo esto algo que es *born digital*<sup>18</sup>.

Por otro lado también se puede dar el caso en el que se almacena una señal análoga, que por naturaleza es *continua* y es convertida a un medio digital discreto, o sea digitalizado, representadas en unidades discretas cuantificables que son a su vez sujetas a distintos tipos de manipulación propios del medio.

La digitalización *grosso modo* involucra dos pasos, el muestreo digital<sup>19</sup> y la cuantificación<sup>20</sup>, los datos son reducidos a unidades discretas que son muestreados

---

<sup>18</sup> El anglicismo *born digital* se refiere a *objetos* que fueron concebidos bajo una forma digital, en contraste a convertir un objeto análogo a su representación digital.

<sup>19</sup> “The core concept in digital audio recording is *sampling*, that is, converting continuous analog signals (such as those coming from a microphone) into discrete *time-sampled* signals. The theoretical underpinning of sampling is the *sampling theorem*, which specifies the relation between the sampling rate and the audio bandwidth [...] This theorem is also called the *Nyquist theorem* after the work of Harold Nyquist after the work of Harlod Nyquist of Bell Telephone Laboratories (Nyquist, 1928), but another form of this theorem was fist stated in 1841 by the French mathematician Augustin Louis Cauchy (1789 1857). The British researcher A. Reeves developed the first patented *pulse-code-modulation* (PCM) system for transmission of messages in ‘amplitude-dichotomized, time-quantized’ (digital) form (Reeves 1938; Licklider 1950; Black 1953). [...] The development of *information theory* contributed to the understanding of digital audio transmission (Shannon 1948)” (Roads, 1995: 9, 10)

<sup>20</sup> “Sampling at discrete time intervals [...] constitutes one of the major differences between digital and analog signals. Another difference is *quantization*, or discrete amplitude

frecuentemente dentro de intervalos regulares, sean estos por ejemplo pixeles o tiempo. Además, como se dijo anteriormente, los datos son reducidos a unidades discretas cuantificadas, se les asigna un valor numérico dentro de un rango definido, como por ejemplo los valores de 0 a 127 para representar intensidad (*velocity*) en una nota MIDI o valores de 0 – 255 dentro de escalas de colores en imagen.

Ahora bien, que los objetos creados dentro de una computadora sean inherentemente reducibles a datos discretos numéricos no quiere decir que objetos creados fuera de ella no puedan poseer esta propiedad de ser reducible a unidades mínimas discretas, aunque no necesariamente numéricas. Un ejemplo que se mencionaba en el capítulo de introducción es la notación musical occidental, que reduce el complejo continuo del sonido en unidades discretas como lo son valores de altura dentro de una gama cromática o valores de duración rítmica dentro de un tempo establecido (que a su vez puede ser *mapeado* a valores numéricos). Lo mismo sucede con el lenguaje o también con la comunicación escrita: “Language is, as it were, that which divides reality (for instance, the continuous spectrum of the colors is verbally reduced to a series of discontinuous terms).” (Barthes, 1968: 63, citado en Manovich, 2001: 29)

Estos valores pueden estar sujetos a manipulación algorítmica o a procedimientos de mapeo. Desde tiempos medievales se pueden rastrear instancias de procedimientos algorítmicos o de mapeo sencillos en la concepción musical cuando

---

resolution. The values of the sampled signal cannot take on any conceivable value. This is because digital numbers can only be represented within a certain range and with a certain accuracy”(Ibíd., 33)

alrededor del año 1026 d.C. Guido d'Arezzo (991/92 - 1050), monje benedictino y destacado teórico de la música, escribe uno de los primeros tratados teóricos musicales bajo el título de "*Micrologus de disciplina artis musicae*". En los capítulos 15 y 17 de dicho tratado, desarrolla un sistema para la generación automática de melodías a partir de un texto con el mapeo de diversos componentes de un texto, como por ejemplo de sílabas a notas musicales. Aquí no hay una representación de datos de manera numérica, sin embargo una reducción a unidades discretas que posibilitan un mapeo o *transcodificación*, siendo este último otro principio de los objetos de la nueva media que también será abordado. En este ejemplo me parece importante resaltar el hecho de que podríamos encontrar varias de las tendencias características de la nueva media, como recién indiqué existe una transcodificación al convertir elementos de un sistema simbólico (vocales de un texto) a música (alturas específicas) por medio de una correspondencia (mapeo) en mayor o menor medida arbitraria, por otro lado, esto posibilita una pseudoautomatización, ya que la intervención manual sigue siendo requerida. Además existe cierto grado de modularidad, ya que el sistema de vocales es relativamente independiente al sistema de alturas, sin embargo, lo que me parece fundamental en este ejemplo es que para permitir todo esto primero fue necesario reducir el fenómeno análogo a unidades discretas manipulables.

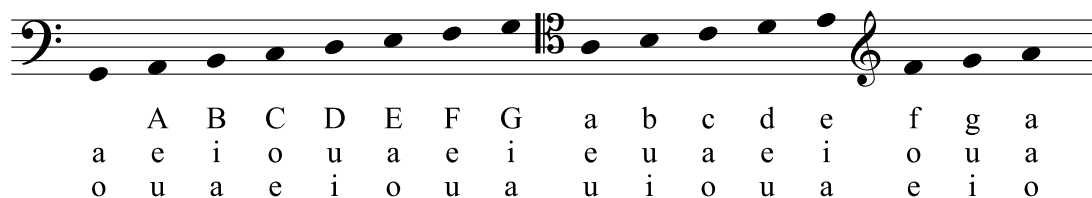


Figura 2.1 Mapeo de sílabas a notas musicales por Guido d'Arezzo.

Una instancia del mapeo de alturas a valores numéricos es la de la noción de *pitch class* o *pitch class sets* dentro de las herramientas de análisis de teoría de música atonal introducidas y desarrolladas principalmente por los musicólogos estadounidenses Allen Forte (1926-2014) y David Lewin (1933-2003). Esto a su vez está íntimamente ligada a desarrollos posteriores de música post serial ya que posibilita una colección de manipulaciones sobre las alturas abordadas analíticamente o dentro de su composición, como conjuntos de clases de alturas dentro de la gama cromática representados mediante números del 0 al 11.

También se podrían mencionar los procedimientos estocásticos que aplicaba Iannis Xenakis (1922 - 2001) en obras donde no hacía uso del ordenador, dentro de las cuales era necesario la reducción de distintos parámetros musicales a datos numéricos para su mapeo a variables dentro de diversas funciones de probabilidad.



Figura 2.2 Micrologus ejemplo musical.

Ahora bien si como hemos visto que previamente a la incorporación del software en la música ha existido de cierta manera una reducción a datos numéricos o al menos a unidades discretas, ¿cuáles son las diferencias fundamentales de la representación numérica digital con respecto a las previas manifestaciones numéricas, discretización y parametrización en la música? ¿Existen diferencias?

Considero que *sí* hay diferencias primordiales entre la representación numérica digital y su contraparte análoga, que radican principalmente en que el modo de existencia dentro de un medio digital es *necesariamente* reducible a datos numéricos, lo cual permite un manejo de material previamente inconcebible y puede tornarse como un elemento fundamental para la estructuración o la manera de abordar una obra. Además de que su misma representación numérica puede facilitar su manipulación de manera matemática y/o algorítmica.

Se pueden encontrar casos claros de esto dentro de algunas de las diversas técnicas de síntesis y procesamiento digital de señales (DSP por sus siglas en inglés), como por ejemplo la síntesis estocástica de Xenakis, también conocida como síntesis no estandarizada (Holtzman, 1978), conceptualizada desde la década de los 60s y llevada a cabo hasta años posteriores en la década de los 70s (Luque, 2006) debido principalmente a limitantes tecnológicos de la época.

Xenakis propuso abordar un tipo de síntesis de micro sonido utilizando distintas distribuciones de probabilidad para manipular muestras digitales individuales, es decir, crear texturas de sonido a partir de la manipulación de sus elementos indivisibles dentro del ordenador. El *sample* o la muestra individual es punta de partida para la construcción del material sonoro, se puede decir que se tiene una concepción cuántica del sonido. (Rocha, 1999)

Dentro de lenguajes de programación como SuperCollider, por un lado la representación misma de valores dentro de argumentos o variables son con

frecuencia literalmente<sup>21</sup> valores de números, que pueden estar representando parámetros para determinado tipo de síntesis, operaciones matemáticas, coordenadas de píxeles en la pantalla, valores MIDI, mensajes OSC, valores que serán mapeados a parámetros musicales, entre otras posibilidades. Por otro lado, todo es teóricamente reducible a datos numéricos, a muestras discretas que pueden ser utilizadas en un sin fin de maneras para conformar un objeto, como en el caso que acabamos de ver en el cual el sonido y su comportamiento es construido a partir de la manipulación de muestras individuales. La representación misma del complejo musical es ya desde la interfaz presentada en una forma simbólica que involucra procesos de mapeo entre valores musicales y valores numéricos.

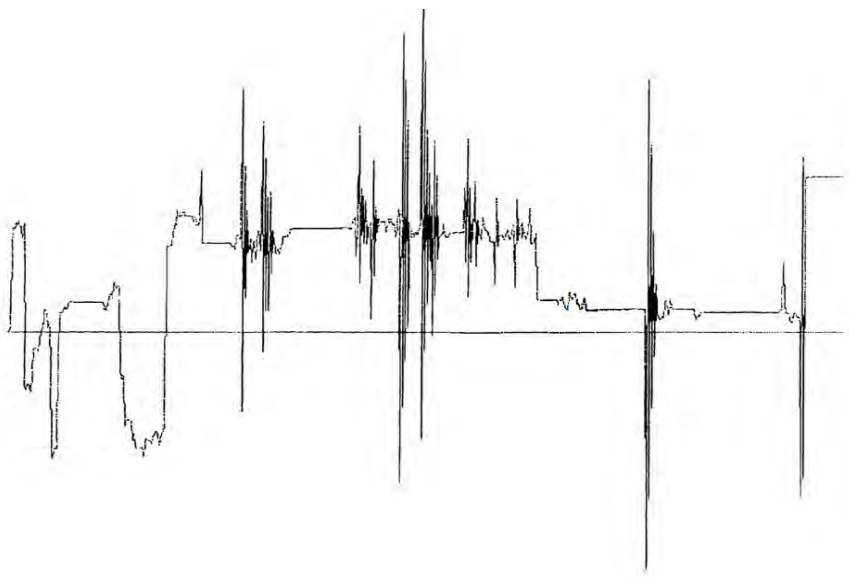


Figura 2.3 Representación de densidades Cauchy exponenciales en ejes de tiempo y presión (Xenakis, 1992: 251)

---

<sup>21</sup> *Literals* en ciencias de la computación son aquellos valores que tienen una representación sintáctica directa, como los números enteros, números de punto flotante, caracteres, *strings* (cadenas de caracteres), entre otros.



### 2.2.2 MODULARIDAD

Los objetos de la nueva media están hechos de tal manera que contienen una estructura modular en el sentido de que pueden ser fragmentados en objetos más pequeños, desde muestras individuales discretas a imágenes, formas, sonidos, entre otros que a pesar de formar un todo que es el objeto resultante, no deja de ser un elemento independiente con sus características propias y que pueden ser apropiadas, extendidas o reutilizadas para conformar otros objetos.

En los lenguajes de programación que operan bajo el paradigma de la programación orientada a objetos<sup>22</sup>, que fue desarrollado con el fin de hacer más explícitos los procesos involucrados en la programación como la interconexión modular entre funciones para trabajar sobre variables, “se utilizan objetos y clases como bloque esenciales de construcción. Una clase define un grupo de métodos (funciones) y campos (variables). Un objeto es una instancia específica de una clase” (Reas, Fry; 2007: 422), es una entidad que contiene datos los cuales representan el estado de dicho objeto y una serie de operaciones que puede ser utilizados para algún fin en particular.

Es común el uso de analogías cuando se está enseñando a programar dentro de un lenguaje OOP entre objetos del software y objetos cotidianos, por ejemplo, podríamos tener una clase hipotética llamada Cerveza, cuyos métodos podrían ser malta, levadura, agua, otrosIngredientes, e instancias u objetos de la clase Cerveza podrían ser Lager, Pale Ale, Pilsner, etc.

---

<sup>22</sup> Comúnmente referida simplemente como OOP por sus siglas en inglés *Object Oriented Programming*.

También hay una serie de conceptos dentro de la programación orientada a objetos que de alguna manera involucran modularidad, como lo son: herencia, polimorfismo, abstracción y encapsulamiento<sup>23</sup>.

Los objetos, siendo en sí una especie de mini programas relativamente independientes, pueden ser interconectados de una manera modular para crear programas o códigos más grandes y complejos, a su vez, fragmentos de estos programas pueden ser reutilizados, recombinados con otra serie de objetos o apropiados para crear programas con comportamientos o fines muy distintos.

Al igual que como sucede con el principio de representación numérica, la modularidad no es una propiedad o principio exclusivo de los nuevos medios, se pueden encontrar instancias de estructuras modulares dentro de los antecedentes tempranos de la algoritmia musical, como por ejemplo, Philippe de Vitry (1291-1361) compositor, teórico musical y poeta francés, mientras iba floreciendo una creciente complejidad polifónica así como perfeccionándose el sistema de notación musical para satisfacer sus necesidades, concibió el concepto de *isorritmia*, el cual consiste en el uso de modelos rítmicos conocidos como *talea* y melódicos llamados *color* que se repiten a lo largo de la pieza desfasándose e interactuando en distintas proporciones. La noción misma de separar el ritmo y la melodía en categorías propias y relativamente independientes para generar un todo ya comienza a asemejarse a una especie de modularidad, a la vez que es necesaria su reducción a unidades más pequeñas para la conformación de un todo.

---

<sup>23</sup> <https://www.processing.org/tutorials/objects/>

Podemos encontrar otro ejemplo de modularidad que aún no involucra los nuevos medios dentro del serialismo integral. El compositor y director francés Pierre Boulez (1925 - 2016) en su artículo "*Schönberg ha muerto*" (Boulez, 1952) destaca al compositor austriaco Arnold Schönberg (1874 - 1951) por haber creado y teorizado el sistema dodecafónico, pero lo critica por no haber explotado sus potenciales seriales al máximo, argumentando que trató la serie dodecafónica como material musical temático tradicional en vez de abordarlo como una configuración abstracta de intervalos. Boulez propuso abordar "todos" los elementos musicales, no solamente las alturas sino también ritmo, dinámicas, textura y la forma musical misma de acuerdo a "procedimientos estrictos" de serialización, y de esta manera al menos en teoría, lograr apartarse por completo de concepciones musicales previas. (Morgan, 1991: 334)

Dentro de su influyente obra *Structures I* (1952) para dos pianos, en la obra *la* utiliza una serie de 12 notas, 12 duraciones (ambos fueron concebidas a partir de la obra *Mode de valeurs et d'intensités* de su maestro Olivier Messiaen que veremos más adelante) 10 ataques y 12 dinámicas (Ligeti, 1960), extendiendo el principio de la dodecafónica a la "serialización total" de los parámetros musicales, o más bien, de los parámetros musicales que son sujetos a una clasificación de unidades discretas posibilitadas dentro de nuestro sistema de notación musical occidental, lo que facilita su implementación dentro de sistemas de esta naturaleza estando ligado al primer principio recién expuesto en el apartado anterior. Esto posibilita la utilización de grupos *modulares* de instancias de parámetros que pueden ser sujetos a una combinatoria relativamente arbitraria e independiente.

A continuación se presentan las mencionadas series, que fueron expuestas por el análisis que hizo Ligeti (1960: 36-62) de la obra, también se puede obtener mas información concerniente al modo de implementación y mapeo en (Smith, 1989).

Original:



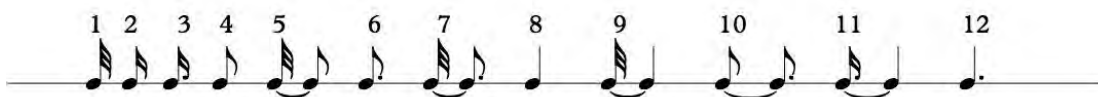
Inversión:



Mapeo de alturas con RI<sup>5</sup> (12, 11, 9, 10, 3, 6, 7, 1, 2, 8, 4, 5)





Valores de alturas, 1 el valor mínimo representa una duración de fusa, a la cual se le va sumando duraciones,  $2 = 0.125 + 0.125 = 0.25 =$  semicorchea.



Valores numéricos mapeados a dinámicas:

1	2	3	4	5	6	7	8	9	10	11	12
<i>pppp</i>	<i>ppp</i>	<i>pp</i>	<i>p</i>	<i>quasi p</i>	<i>mp</i>	<i>mf</i>	<i>quasi f</i>	<i>f</i>	<i>ff</i>	<i>fff</i>	<i>ffff</i>

Finalmente, la serie de 10 ataques:

1	2	3	5	6	7	8	9	11	12
>	.>	.	<i>normal</i>	. 	∨	<i>sfz</i> ^	>	∴	

El compositor alemán Karlheinz Stockhausen (1928 - 2007) que dentro de sus fases iniciales como compositor en la década de los 50s fue otro de los principales propulsores del serialismo integral, en 1956 concibe la controversial obra para piano *Klavierstück XI*. En esta obra, que está compuesta de diecinueve fragmentos musicales dispuestos en una sola hoja que comprende la partitura el ejecutante escoge el camino a seguir dentro de estos *a piacere*, con algunas cuantas indicaciones y reglas generales, se crea la obra a través de módulos musicales denotando una estructura modular distinta a la de Boulez, ya que se integra a la interpretación, además de estar presente en los fragmentos musicales que poseen estructuras proporcionales (Maconie, 1976).

Cabe mencionar que tanto Boulez como Stockhausen fueron alumnos del influyente compositor francés Olivier Messiaen (1908 - 1992), que ya desde su seminal libro *Technique de mon langage musical* (Messiaen, 1944) muestra una inclinación a un tratamiento estructural relativamente independiente de los parámetros musicales para luego ser integrados como un todo dentro de un discurso musical. Dentro de



En el capítulo 4 abordaré las estructuras modulares dentro de la obra de Xenakis en su obra pre-informática *Analogique A*.

A pesar de que se pueden encontrar numerosas manifestaciones de estructuras modulares dentro de distintas etapas de la literatura musical, habiendo aquí expuesto dos instancias considerablemente lejanas diacrónicamente entre sí, opino que existen diferencias importantes con respecto a las estructuras modulares que se pueden encontrar dentro de un medio digital, específicamente dentro de un lenguaje de programación utilizado para crear música. Consisten principalmente en que el texto es dinámico y puede ser implementado e interconectado dentro del mismo hábitat digital con diferentes lenguajes de programación. Por ejemplo se podría imaginar la situación hipotética en la que se *transforma* lo que inicialmente era un *fragmento* de código para la composición de una obra con instrumentos acústicos a uno que controle (el mismo tipo de) comportamientos y distribuciones a parámetros de síntesis de sonido que a su vez es conectado con otro lenguaje de programación para controlar transformaciones de una imagen, el *mapeo* en las estructuras modulares de la nueva media son ampliamente magnificadas y abarcan una primacía que no solamente posibilita sino que por su misma naturaleza se podría decir que *invita* a dichos procedimientos.

Es común que en libros y manuales de programación para artistas se cubra el tema explorando la interconexión de programas de distinta naturaleza, *apropiación* de aparatos tecnológicos como un control *Wii* o los sensores del *Kinect* explorando distintas técnicas de mapeo para lograr una mayor gestualidad expresiva entre un

sin fin de posibilidades que desde luego que depende del artista si decide incluirlos o no en su discurso, pero que de alguna manera ya están implícitos en su formación.

A su vez, al existir este lenguaje de programación en un *metamedio*<sup>24</sup> que es la computadora, coexiste con otros programas sean estos de audio, video, animación, para programar micro controladores, entre otros, con los que puede interactuar por medio de algún protocolo de comunicación, como por ejemplo OSC<sup>25</sup>. Todo esto incita a combinar distintos discursos o medias, sea esto de manera superpuesta conservando cada media sus atributos sin ser modificados en lo absoluto o mezclando elementos de distintas medias las cuales alteran o entrelazan sus atributos característicos creando *híbridos*. “en la media híbrida los lenguajes de medias previamente distintas se funden. Intercambian propiedades, crean estructuras nuevas e interactúan en el nivel más hondo.” (Manovich, 2008: 76, traducción propia)

Por otro lado, también existen estructuras modulares que involucran fragmentos de código escritos por diversos programadores alrededor el mundo para otros

---

<sup>24</sup> El Ideal del *metamedio* del influyente informático estadounidense Alan Kay (1940-) lo aborda Manovich de la siguiente manera: “computer metamedium is simultaneously a set of different media and a system for generating new media tools and new types of media. In other words, a computer can be used to create new tools for working in the media it already provides as well as to develop new not-yet-invented media [...]The previously unique properties and techniques of different media became the elements that can be combined together in previously impossible ways.” (Manovich, 2008: 68, 78)

<sup>25</sup> Open Sound Control (OSC) es un protocolo de comunicación desarrollado en los laboratorios CNMAT Berkeley California que permite comunicar instrumentos de música, computadoras y otros dispositivos multimedia, para compartir información musical en tiempo real sobre una red. <http://vcresearch.berkeley.edu/research-unit/center-new-music-and-audio-technologies>



propósitos, que nuevamente son *apropiados* para crear discursos y objetos completamente distintos, siendo esto de práctica común en los lenguajes de programación. Por ejemplo en SuperCollider los *quarks*<sup>26</sup> y *plug-ins*, que son programas escritos por terceros y que implementamos modularmente en nuestros códigos. “New media objects are rarely created completely from scratch; usually they are assembled from ready-made parts [...] Pulling elements from databases and libraries becomes the default; creating them from scratch becomes the exception.” (Manovich, 2001: 124, 130)

Mas aún, el mismo programa SuperCollider tiene una estructura modular, *grosso modo* comprende dos programas principales que son *sclang* y *scsynth* (siendo *sclang* cliente de *scsynth*) que se ocupan respectivamente de lenguaje (aquí también está el interpretador) y por otro lado del síntesis de sonido, siendo *conectados* por medio del protocolo de comunicación TCP (*Transmission Control Protocol*) o UDP<sup>27</sup> (*User Datagram Protocol*) en el cual intercambian mensajes OSC.

---

<sup>26</sup> Quarks are packages of SuperCollider code containing classes, extension methods, documentation and server UGen plugins. The Quarks class manages downloading these packages and installing or uninstalling them. (Documentación de ayuda de SuperCollider versión 3.7.0 beta-1)

<sup>27</sup> “User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) [...] have a common way of addressing services on a particular machine [...] Computers (hosts) within the same network segment are uniquely identified by their IP address [...] UDP protocol is message oriented (i.e., communication takes place by sending individual packets of limited size, without any guarantee of correct delivery). It is used in many audio and video applications because of its lower overhead compared with TCP and its ease of use. It is also the default protocol used by *sclang* and *scsynth*. TCP, on the other hand, is connection oriented (i.e., conceptually, communication is an exchange of infinite Streams of unstructured binary data between 2 connected hosts.)” (Wilson; Cottle; Collins; 2011: 114-115)

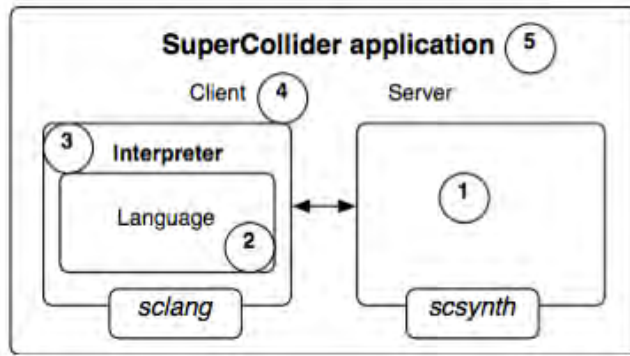


Figura 2.11 Comunicación de las partes que integran el lenguaje de programación SuperCollider. (Documentación de ayuda de versión 3.7.0 beta-1)

En el siguiente capítulo abordaré cómo entra en juego la modularidad dentro de las obras que compuse para este trabajo y reforzaré las nociones hasta aquí expuestas.

### 2.2.3 AUTOMATIZACIÓN

Los dos principios anteriores, representación numérica y modularidad, propician la automatización de procesos involucrados en la creación, manipulación y acceso de distintos objetos de nueva media. Antes de comenzar a abordar la automatización en objetos de nueva media quisiera explorar algunas de sus manifestaciones y especulaciones tempranas dentro de la concepción musical, con el fin de elucidar sus similitudes y diferencias con respecto a su contraparte digital.

Athanasius Kircher (1601 ó 1602 - 1680) un sacerdote jesuita, erudito e importante científico de la época barroca, desarrolló un sistema de composición algorítmica entre muchas otras cosas, dentro de su extenso tratado *"Musurgia Universalis"* escrito en 1650 (Cramer, 2005; Nierhaus, 2009). Este sistema consiste de tres categorías de pequeñas tablas de madera etiquetadas, en los cuales están inscritos valores numéricos y rítmicos. Su sistema permite la generación automática de música polifónica en el estilo de contrapunto simple y florido, que describe de la

siguiente manera: “The mechanical production of music is nothing other than a certain closely defined method I have invented, by which anyone, even if he has no musical knowledge, may, by varied application of music making instruments, compose tunes”. (Kircher: 1650, citado en Nierhaus, 2009)

En el “*Arca Musarithmica*” de Kircher, columnas de cuatro líneas de números pueden ser combinadas con patrones rítmicos a cuatro voces, los números representan el grado dentro de determinado modo y están agrupados en unidades de dos a doce con el propósito de transferir correctamente pasajes de texto representando una sílaba cada uno. Cada grupo de alturas de tamaño fijo puede ser combinado con un grupo de patrones rítmicos del mismo tamaño para producir música en el estilo de contrapunto simple a cuatro voces (Ibíd., 2009), aquí también podemos ver cómo está implícita la modularidad en la “automatización”.

En el “*Arca Musurgia*” que está incompletamente descrito por Kircher, los grupos de alturas y patrones rítmicos están contruidos de manera que se pueda generar música en cinco géneros distintos: monodia, madrigal, fuga, motete y música sacra. (Ibíd., 25)

Por otro lado desarrolla en el año 1661 lo que él llamó “*organum mathematicum*” para el adiestramiento del joven archiduque Karl Joseph de Austria. El mecanismo realmente no era un órgano sino una caja con tablas de madera con información inscrita en ellas elaborado de tal manera que habilita la manipulación de los datos para hacer cálculos, en este caso era un obsequio con fines didácticos que contenía

una treintena de distintas disciplinas, entre ellas composición musical abordado de manera similar a la expuesta. (Ibíd.)

No está de más mencionar que una versión menos extensa de esta idea de combinatoria “automatizada” ya se había concebido alrededor de dos siglos antes por Ramón Lull (1232-1315) con su “*Ars Magna*”, donde por medio de una combinatoria algorítmica es posible generar expresiones generalmente aceptadas, en su mayoría bajo el dogma cristiano, con el objetivo principal de la conversión del mundo islámico al cristianismo por medio de la racionalidad. (Cramer, 2005)

bb. 9

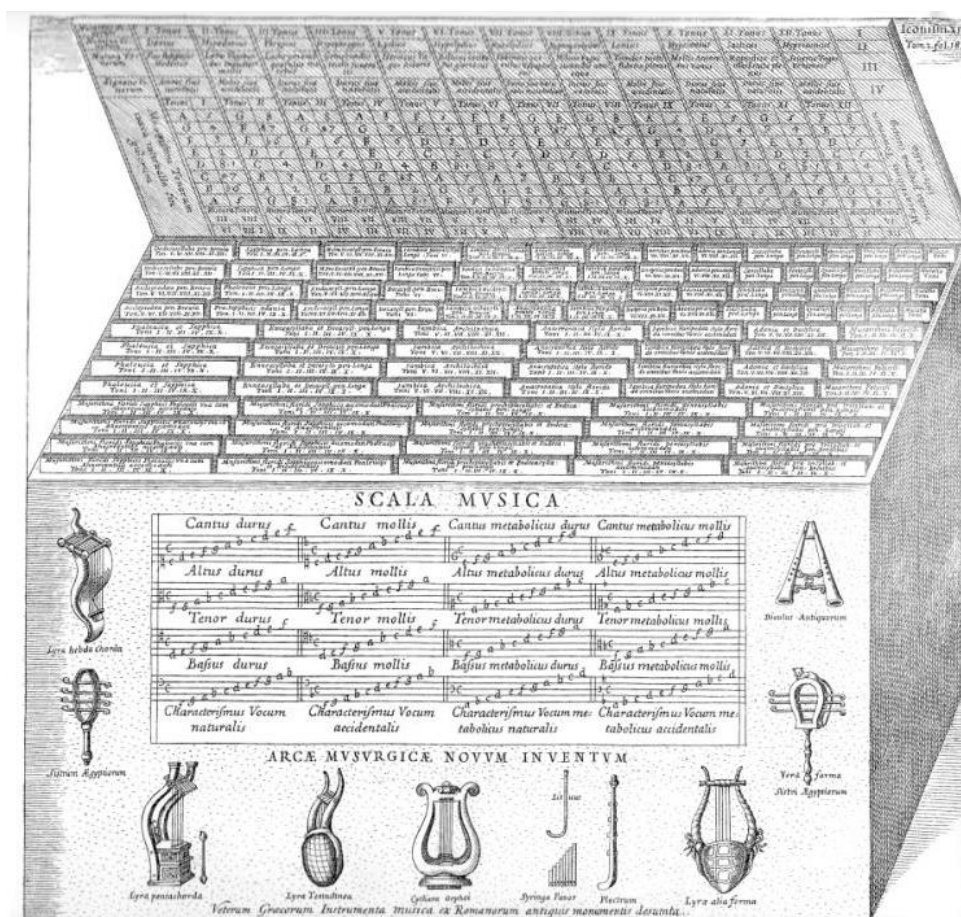
MU B, Iconismus XIV, nach S. 186, *Arca musurgica*

Figura 2.12 Arca Musurgica, Musurgia Universalis, 1650, Kircher.

En el siglo XVIII apareció un juego musical que se popularizó rápidamente, el llamado “juego musical de dados”. En pocas palabras, el juego consistía en la generación de música de diversos géneros populares de la época, a partir de una combinatoria de posibilidades de compases pre-compuestos por el autor del juego. Las posibilidades de combinación estaban dispuestas de una manera en que no existieran “incongruencias musicales”, como por ejemplo, que no hubiesen sucesiones melódicas que fueran consideradas inapropiadas para la época o progresiones armónicas igualmente inadecuadas, además que respetara el género sobre el cual se estaba “componiendo”. Para alrededor del año 1812 ya se habían compuesto una veintena de juegos de diversos autores, el primero del que se tiene récord se le atribuye a Johann Philipp Kirnberger en el año de 1757, Carl Philipp Emanuel Bach escribe uno el próximo año, igualmente se le conoce un juego a Maximilian Stadler del año 1780. Después de 1812 aparecieron juegos bajo la autoría de Mozart y Haydn, sin embargo no hay pruebas sustanciales que corroboren su veracidad (Nierhaus, 2009). Aunque el carácter de estos juegos es puramente lúdico se requiere destreza para su creación, reflejo de ello está en que compositores de renombre hayan estado involucrados en la elaboración de algunos ejemplares.

Por otro lado, también en el siglo XVIII dentro de las cortes europeas las curiosidades tecnológicas y una variedad de autómatas eran cada vez más populares, con el objetivo de entretener a la nobleza. Jacques de Vaucanson uno de los inventores más destacados de estos aparatos, concibió la idea de un telar

completamente automatizado en el año de 1745<sup>28</sup>, pero fue Joseph Marie Jacquard quién construyó el primer modelo listo para la fabricación en masa para la industria textil hacia el año de 1805. (Manovich, 2001; Nierhaus, 2009)

El concepto de utilizar tarjetas perforadas para el control automatizado fue utilizado posteriormente para la creación de diversos mecanismos, entre ellos la máquina analítica que fue concebida por Charles Babbage alrededor del año 1833. Dicha máquina tenía la función de resolver cualquier problema matemático y es considerada por muchos como el predecesor de la computadora, siendo Babbage referido como el “padre de la computadora”. El diseño no pudo llevarse a cabo en la vida de Babbage debido principalmente a los enormes costos y a la falta de una tecnología adecuada que facilitara su manufactura, sin embargo, sus diseños y principios básicos eran correctos y fueron desarrollados por Konrad Zuse, Howard Hathaway Aiken, entre otros, en tiempos posteriores.

Ada Lovelace (1815 - 1852) una matemática autodidacta que trabajó junto con Babbage en la conceptualización de la máquina analítica, en 1843 tradujo un texto del matemático italiano Luigi Federico Menabrea “*Notions sur la machine analytique de Charles Babbage*” al inglés donde además agregó un texto suyo donde explica detalladamente un algoritmo para ser ejecutada en la máquina. Por esta razón se le ha referido como la primera programadora del mundo. No está de más mencionar que Lovelace sugirió en determinado momento hacer una implementación en la

---

<sup>28</sup> En 1725 Basile Bouchon había desarrollado un sistema semi-automatizado mediante un sistema binario (este sistema binario había sido previamente teorizado por Leibniz en 1679) con tarjetas de papel perforado.

máquina analítica para componer piezas musicales, lamentablemente la idea nunca fue puesta en práctica, sin embargo, deja huella de una premonición certera.

[...] suponiendo, por ejemplo, que las relaciones fundamentales de los sonidos...en la ciencia de la armonía y de la composición musical fueran susceptibles de tal expresión y adaptación, la Máquina podría componer piezas de música de cualquier grado de dificultad y extensión. (Berron, 2007: 10; citado en Gutiérrez, 2014: 19)

Regresando brevemente a la obra *Structures al* mencionada en el apartado de modularidad, Boulez utiliza distintas series de alturas, ataques y dinámicas, los cuales organizó de manera en que después de crear su estructura y sus mapeos correspondientes, prácticamente todos los datos podían ser vaciados a la partitura.

Se podría hacer una analogía con la generación automática de música dentro de las cajas musicales de Kircher y algunos de los distintos ejemplos hasta aquí expuestos, en el sentido de que existe una composición que podría ser llamada “automática” y definitivamente *algorítmica* pero aún no *automatizada*.

La primera pieza generada completamente por computadora fue hecha por Lejaren Hiller y Leonard Isaacson entre 1955 y 1956 en la computadora Illiac de la universidad de Illinois. Se trata de la llamada “*Illiad Suite*” para cuarteto de cuerdas, dividida en cuatro movimientos a los que titularon “experimentos” donde en cada uno desarrollaron un concepto musical distinto. Por ejemplo en los experimentos 1 y 2 aplicaron conceptos y técnicas contrapuntísticas de Josquin des Près y Giovanni Pierluigi da Paestrina mientras que en el experimento 4 implementaron cadenas de Markov, de las que hablaré posteriormente en el capítulo 4 donde se abordará el análisis de Xenakis.

Este logro fue fundamental para el inicio de la composición asistida por computadora y la generación de diversos software de programación que se desarrollaron conforme se creaban lenguajes de programación de alto nivel. En 1963 Hiller y Robert Baker desarrollaron Musicomp el primer lenguaje diseñado para la CAO, ya desde 1957 Max Mathews desarrolló una serie de programas Music para la generación de sonido a partir de la computadora y junto con Richard F. Moore crearon el sistema Groove de los 70s. Existe una amplia variedad de lenguajes de programación que germinaron a partir de estos avances en años posteriores que se utilizaron para la síntesis de sonido o la composición asistida por ordenador; Csound (1986 - actualidad) de Barry Vercoe, Common Lisp Music (1991 - actualidad) de Bill Schottstaedt y programas más recientes como Pure Data (Miller Puckette), Max/MSP (Puckette y Zicarelli), Chuck (Wang), SuperCollider (James McCartney), solo por mencionar algunos.

Cabe destacar que en 1955, el mismo año en que se comenzó a trabajar sobre la ya mencionada *Illiad Suite*, Iannis Xenakis estrenó su impactante obra *Metastasis* bajo la batuta de Hermann Scherchen. La obra se caracteriza por el entretejido de glissandi a distintas velocidades, clusters que abarcan un rango de 46 notas y la noción de sonoridades globales como un ente que se puede transformar en otro, esto último llevaría a Xenakis a pensar en dichas sonoridades con una concepción estadística y lo llevaría a formular lo que él llamó música estocástica, (Nierhaus, 2009) lo cual describe de la siguiente manera:

For if, thanks to complexity, the strict, deterministic causality which the neo-serialists postulated was lost, then it was necessary to replace it by a more general causality, by



a probabilistic logic which would contain strict serial causality as a particular case. This is the function of stochastic science. "Stochastics" studies and formulates the law of large numbers [...] the laws of rare events, the different aleatory procedures, etc. As a result of the impasse in serial music, as well as other causes, I originated in 1954 a music constructed from the principle of indeterminism; two years later I named it "Stochastic Music." The laws of the calculus of probabilities entered composition through musical necessity. (Xenakis, 1992: 8)

A mediados de los 50s del siglo XX acontecieron dos sucesos que marcarían hondamente el rumbo de la composición, se materializaron ideas que se venían vislumbrando desde siglos anteriores, abriendo las puertas a una infinidad de posibilidades creativas que se siguen desarrollando y ampliándose a múltiples disciplinas donde la música, el arte, la tecnología y la ciencia hacen intersección.

Cabe mencionar que dentro del contexto de los 50s y 60s, que si bien los software de programación para audio digital empezaron a surgir en estos tiempos con el programa MUSIC, desarrollado por Max Mathews en los laboratorios Bell (lo cual desencadenó una familia de programas del linaje MUSIC-N). Estos programas no corrían en tiempo real, para hacer una función podían tardar horas o incluso días y por otro lado eran ejecutadas en computadoras que costaban millones de dólares en laboratorios ultra especializados. En los 60s se distribuyó el sucesor MUSIC-4 a algunas universidades privadas de Estados Unidos como Princeton y Stanford donde seguirían siendo desarrollados, aún en laboratorios especializados y con equipo que estaba fuera del alcance de la gran mayoría de la población. Con el advenimiento de computadoras personales cada vez más poderosas con precios más accesibles fue creciendo exponencialmente el uso y desarrollo de software de programación para audio digital dentro y fuera de la academia para una amplia gama de usos como por

ejemplo, síntesis de sonido, composición musical asistida por computadora, investigación de acústica, psicoacústica, entre otros. Sobre esto Manovich nos dice:

Between its invention in the mid 1940s and the arrival of PCs in the early 1980s, the digital computer was mostly used for military, scientific, and business calculations and data processing. It was not interactive. It was not designed to be used by a single person. In short, it was hardly suited for cultural creation. (Manovich, 2013: 55)

En un mismo tono Wang menciona el contexto en el cual se realizaron los trabajos pioneros en la composición asistida por ordenador y en la síntesis de sonido:

Our first era of computer-based music programming systems paralleled the age of mainframes (the first generations of “modern” computers in use from 1950 to the late 1970s) and the beginning of personal workstations (mid 1970s) The mainframes were gigantic, often taking up rooms or even entire floors. Early models had no monitor or screens, programs had to be submitted via punch cards, and the results delivered as printouts. Computing resources were severely constrained. It was difficult even to gain access to a mainframe – they were not commodity items and were centralized and available mostly at academic and research institutions (in 1957 the hourly cost to access a mainframe was \$200!)

The composer / programmer would design their software (usually away from the computer), create punch cards specifying the instructions, and submit them as jobs during scheduled mainframe access time (also referred to as batch-processing) – sometimes traveling far to reach the computing facility. The process was extremely time-consuming. A minute of audio might take several hours or more to compute, and turn-around times of several weeks were not uncommon. Furthermore, there was no way to know ahead of time whether the result would sound anything like what was intended. After a job was complete, the generated audio would be stored on computer tape and then be digital-to-analog converted, usually by another computer. Only then could the composer actually hear the result. It would typically take many such iterations to complete a piece of music. (Wang, 2008: 11-15)

Hasta este punto he expuesto varias instancias de automatizaciones o pseudoautomatizaciones previas a la llegada de la computadora personal, es evidente que ya con el ordenador lo “automático” deja de ser “manual” y se convierte en una automatización concreta, lo cual ya conlleva diferencias significativas. Xenakis veía estas automatizaciones como algo liberador y como el siguiente eslabón dentro del desenvolvimiento de la composición en un marcado tono modernista: “computers that accelerate the mental processes necessary for clearing the way for new fields by providing immediate experimental verifications at all stages of musical construction.” Y más adelante, en las conclusiones sobre su serie de obras *ST* que fueron compuestas en una computadora IBM 7090

Freed from tedious calculations the computer is able to devote himself to the general problems that the new musical form poses and to explore the nooks and crannies of this form while modifying the values of the input data. For example, he may test all instrumental combinations from soloists to chamber orchestras, to large orchestras. With the aid of electronic computers the composer becomes a sort of pilot: he presses the buttons, introduces coordinates, and supervises the control of a cosmic vessel sailing in the space of sound, across sonic constellations and galaxies that he could formerly glimpse only as a distant dream. Now he can explore them at his ease, seated in an armchair [...] The program, i.e., the list of sequential operations that constitute the new musical form, is an objective manifestation of this form. The program may consequently be dispatched to any point on the earth that possesses computers of the appropriate type, and may be exploited by any composer pilot. (Xenakis, 1992: 133, 144)

Manovich comienza su apartado de automatización dentro del capítulo de principios de la nueva media con la siguiente afirmación:

The numerical coding of media (principle 1) and the modular structure of a media object (principle 2) allow for the automation of many operations involved in media creation, manipulation, and access. *Thus human intentionality can be removed from the creative process, at least in part*<sup>29</sup>. (Manovich, 2001: 32)

Sobre la última parte de esta afirmación considero que la *intención humana* no está ausente, sino que su articulación y vinculación con el objeto es trastocado. Por ejemplo, Alex McLean y Geraint Wiggins en "*Bricolage Programming in the Creative Arts*" (McLean y Wiggins, 2010) exponen el proceso creativo del artista y lo sitúan dentro de un estilo de programación de *bricolage*<sup>30</sup>; buscan explorar la relación entre el artista, el proceso creativo, su programa y el resultado o la obra a través de psicología, lingüística cognitiva, ciencias computacionales y creatividad. Hacen énfasis en la retroalimentación inmediata que percibe el artista, describiendo un *feedback loop* (que se posibilita en gran parte gracias a la automatización):

[...] bricolage programming as a creative feedback loop [encompasses] the written algorithm, its interpretation, and the programmer's perception and reaction to its output or behavior. The addition of the algorithmic component in the creative feedback loop makes an additional inner loop explicit between the programmer and the text. At the beginning, the programmer may have a half-formed concept, which only reaches internal consistency through the process of being expressed as an algorithm. The inner loop is where the programmer elaborates upon their imagination of what might be, and the outer where this trajectory is grounded in the

---

<sup>29</sup> Mi énfasis.

<sup>30</sup>"The bricoleur resembles the painter who stands back between brushstrokes, looks at the canvas, and only after this contemplation, decides what to do next. Bricoleurs use a mastery of associations and interactions. For planners, a program is an instrument for premeditated control; bricoleurs have goals but set out to realize them in the spirit of a collaborative venture with the machine. For planners, getting a program to work is like 'saying one's piece'; for bricoleurs, it is more like a conversation than a monologue." (Turkle y Papert, 1990: 136, citado en McLean y Wiggins, 2010: 2)

pragmatics of what they have actually made. Through this process both algorithm and concept are developed, until the programmer feels they accord with one another or otherwise judges the creative process to be finished. (Ibíd., 2)

Por lo tanto lejos de que se haya perdido alguna parte de la “intencionalidad humana”, dentro del proceso creativo se ha adoptado la automatización. La parte algorítmica quizá ya existía en media tradicional, sin embargo al volverse dinámico y ejecutable se ha trastocado su articulación, “The addition of the algorithmic step would appear to affect the creative process as a whole” (Ídem.)

Florian Cramer hace referencia al mismo fenómeno del *feedback loop* refiriéndose a esto como algo que genera una imaginación expectativa y que es incrustada en nuestra cultura de software (Cramer, 2005: 9). Ge Wang también hace referencia a esto en su tesis doctoral donde creó y desarrolló el lenguaje de programación de audio *Chuck*,

[...] audio programming, in both computational acoustics research and in music composition and performance, is necessarily an experimental and empirical process; it requires rapid experimentation, verification/rejection/workshopping of ideas and approaches, and takes the forms of both short-term and sustained prototyping. It can greatly benefit from the ability to modify, or even create, parts of the software system ‘on the fly’ — as it runs. We believe that rapid prototyping, in and of itself, is a uniquely useful approach to programming audio, with its own benefits (and different ways of thinking about problems). (Wang, 2008: 3)

Es interesante ver la analogía existente entre estas afirmaciones, sobre todo dentro de las de Wiggins y McLean refiriéndose al proceso creativo del artista/programador, con el método de trabajo expuesto por Xenakis que etiqueta como “fases fundamentales de un trabajo musical” al exponer su *música estocástica*

*libre* compuesta (aborda principalmente *Pithoprakta* y *Achorripsis*) *antes de entrar en contacto con un ordenador:*

1. *Initial conceptions* (intuitions, provisional or definitive data);
2. *Definitions of the sonic entities* and of their symbolism communicable with the limits of possible means (sounds of musical instruments, electronic sounds, noises, sets of ordered sonic elements, granular or continuous formations, etc.);
3. *Definitions of the transformations* which these sonic entities must undergo in the course of the composition (macrocomposition: general choice of logical framework, i.e., of the elementary algebraic operations and the setting up of relations between entities, sets and their symbols as defined in 2.); and the arrangement of these operations in lexicographic time with the aid of succession and simultaneity);
4. *Microcomposition* (choice and detailed fixing of the functional or stochastic relations of the elements of 2.), i.e., algebra outside-time, and algebra in-time;
5. *Sequential programming* of 3. and 4. (the schema and *pattern* of the work in its entirety);
6. *Implementation of calculations*, verifications, feedbacks, and definitive modifications of the sequential program;
7. *Final symbolic result* of the programming (setting out the music on paper in traditional notation, numerical expressions, and graphs, or other means of solfeggio);
8. *Sonic realization* of the programs (direct orchestral performance, manipulations of the type of electromagnetic music, computerized construction of the sonic entities and their transformations). (Xenakis, 1992: 22)

Por otro lado son interesantes los resultados de la investigación de Petre y Blackwell (1999) en las que las construcciones algorítmicas dentro de la programación evocan analogías y metáforas. Se le dota una dimensión antropomorfa al código. En dicho estudio le dieron a sus sujetos algunas tareas de programación, y se les pidió que

analizaran en qué pensaban mientras estaban trabajando. Dentro de los resultados se pueden encontrar descripciones como las siguientes:

[...] landscape of solutions, dealing with interacting creatures, transforming a dance of symbols, hearing missing code as auditory buzzing, combinatorial graph operations, munching machines, dynamic mapping and conversation [...] The above research suggests that programmers understand the operation of their programs by metaphorical relation to their experience as a human [...] by embedding the development of an algorithm in a human creative process, the algorithm itself becomes a human expression. (Petre y Blackwell, citados en McLean y Wiggins, 2010)

Finalmente Manovich distingue entre “*Low level automation*” algoritmos que automáticamente mejoran calidad de imagen, *templates* de Word o de alguna página web, software de AL (*artificial life*) en películas, filtros de *instagram*. “*High level automation*” con la que básicamente se refiere a Inteligencia artificial y finalmente automatizaciones en búsquedas de información o de datos, *batch processing*, motores de búsqueda. Sobre este último, me gustaría ahondar en los siguientes dos capítulos, ya que expondré como interactúan las estructuras paradigmáticas de las bases de datos con dimensiones sintagmáticas<sup>31</sup> de discursos temporales, como lo es la música.

---

<sup>31</sup> “Saussure emphasized that meaning arises from the differences between signifiers; these differences are of two kinds: *syntagmatic* (concerning positioning) and *paradigmatic* (concerning substitution) [...] The plane of the syntagm is that of the combination of ‘this-and-this-and-this’ [...] while the plane of the paradigm is that of the selection of ‘this-or-this-or-this’ ” (Chandler, 2007: 83-84)

## 2.2.4 VARIABILIDAD

La variabilidad es el cuarto principio que puede estar presente en los objetos de la nueva media. Manovich comienza el apartado de variabilidad de la siguiente manera:

A new media object is not something fixed once and for all, but something that can exist in different potentially infinite versions. This is another consequence of numerical coding of media (principle 1) and the modular structure of a media object (principle 2).

Old media involved a human creator who manually assembled textual, visual, and/or audio elements into a particular composition or sequence. This sequence was stored in some material, its order determined once and for all. Numerous copies could be run off from the master, and, in perfect correspondence with the logic of an industrial society, they were all identical. New media, in contrast, is characterized by variability. (Other terms often used in relation to new media and that might serve as appropriate synonyms of *variable* are *mutable* and *liquid*.) Instead of identical copies, a new media object typically gives rise to many different versions. And rather than being created completely by a human author, these versions are often in part automatically assembled by a computer [...] Thus the principle of variability is closely connected to automation [...] Variability would also not be possible without modularity. Stored digitally, rather than in a fixed medium, media elements maintain their separate identities and can be assembled into numerous sequences under program control. (Manovich, 2001: 36)

Más adelante menciona algunas instancias particulares del principio de la variabilidad, dentro de los cuales incluye la *base de datos*, diciendo que funciona como una forma cultural por sí misma ofreciendo un modo particular de la experiencia humana afectando la manera en que el usuario concibe los datos que contiene. Además incluye las estructuras ramificadas encontradas dentro de los



*hyperlinks*, actualizaciones de software y la escalabilidad, en el cual diferentes versiones del mismo objeto pueden ser generados en una amplia gama de tamaños y niveles de detalle. (Ibíd., : 37-38)

Antes de abordar el principio de variabilidad específicamente dentro de los nuevos medios y algunas de sus implicaciones dentro de la CAO, me gustaría mencionar un par de ejemplos de este principio dentro de instancias pre-informáticas.

Es común encontrar analogías de las estructuras ramificadas dentro de *hyperlinks*, que de alguna manera no implican narrativas lineales dentro de la literatura, como por ejemplo, el concepto del poeta y crítico francés Mallarmé (1848 - 1898) dentro de *Le Livre*, o también los 100, 000 billones de poemas del poeta, novelista, escritor y cofundador de OuLiPo<sup>32</sup> Ramond Queneau (1903 - 1976) que consiste en un soneto combinatorio en diez variaciones. "It was printed in a book whose lines were individually sliced so that each line of a poem could be turned like a page and picked from ten alternatives. From ten alternatives fro the twelve sonnet lines,  $10^{12}$  possible poem combinations result." (Cramer, 2005: 89) También se podría mencionar la famosa *Librería de Babel* del escritor argentino Jorge Luis Borges (1899 - 1986) en el cual vislumbra una librería cuyos libros son generados por un fantástico mecanismo de combinatoria dispuestos en colecciones prácticamente infinitas de hexágonos. Un último ejemplo literario en el cual podríamos ver tanto variabilidad (además de (seudo)automatización y modularidad) son los poemas

---

<sup>32</sup> Oulipo (por sus siglas en francés *Ouvroir de littérature potentielle*, Taller de Literatura potencial) cuyos miembros pioneros eran Raymond Quenaue y el matemático François Le Lionnais.

Dada de 1923 de Tristan Tzara (1896 - 1963) : *To Make a Dadaist Poem* en los cuales expone un “algoritmo” para generar poemas partiendo del “procesamiento” de entradas arbitrarias:

Tome un periódico.

Tome unas tijeras.

Escoja de este periódico un artículo de las dimensiones que desee para su poema.

Corte el artículo.

Después cuidadosamente corte cada una de las palabras que conforman este artículo y colóquelas en una bolsa.

Agite gentilmente.

Luego retire cada una de las palabras, una después de la otra.

Copie con consciencia el orden en que fueron retirados de la bolsa.

El poema se asemejará a usted.

Y ahí lo tiene—un autor infinitamente original de sensibilidad encantadora, a pesar de ser despreciado por la manada vulgar.<sup>33</sup>

Dentro de la música del siglo XX podemos encontrar también instancias del principio de variabilidad dentro de la música aleatoria o indeterminista siendo el compositor estadounidense John Cage (1912 - 1992) una referencia por excelencia, quien comenzó a incorporar operaciones aleatorias en sus composiciones musicales

---

<sup>33</sup> Tristan Tzara. Pour faire une poème dadaïste. Consultado en (Cramer, 2005: 76, traducción propia).

en la década de los 50s, notoriamente empleando el antiguo libro *I Ching* como un mecanismo para producir números aleatorios de entre 1 y 64, teniendo estas técnicas aleatorias sus raíces en la filosofía Zen.

Por otro lado Iannis Xenakis dentro de su concepción de la música estocástica, comienza su travesía solamente imaginando el uso de la computadora dentro de sus composiciones y estructuras en la década de los 50s.

Algunas de las principales diferencias en la variabilidad con la media tradicional y los nuevos medios se podrían resumir en los siguientes puntos:

Los puntos 4 y 5 que expone Manovich (2001: 37, 38) sobre principios de variabilidad hablan de estructuras ramificadas interactivas como los que se pueden encontrar en menús o *hyperlinks*, en nuestro contexto hablaremos de ramificaciones automatizadas, siendo las implicaciones sobre la automatización expuestas en el apartado anterior.

Con respecto a la escalabilidad, los mismos algoritmos empleados para algún instrumento pueden ser utilizados para otros instrumentos o comportamientos de parámetros de determinado tipo de síntesis sonora, el mapeo casi siempre involucra la escalabilidad.

Although I deduce the principle of variability from more basic principles of new media –numerical representation and modularity of information– the principle can also be seen as a consequence of the computer’s way of representing data —and modeling the world itself— as variables rather than constants. As new media theorist and architect Marcos Novak notes, a computer—and computer culture in its wake—substitutes every constant with a variable. In designing all functions and data structures, a

computer programmer tries always to use variables rather than constants. (Manovich, 2001: 43)

Esta es otra manera en que la variabilidad y/o elementos estocásticos entran en contacto con la composición por medio de la programación. La misma sintaxis y la práctica común de declarar variables y argumentos dentro del código invita a crear estructuras con elementos basados en probabilidades o fluctuantes, las cuales pueden ser probadas de manera heurística, interactuando todos los principios de la nueva media hasta aquí expuestos.

If we apply this principle to culture at large, it would mean that every choice responsible for giving a cultural object a unique identity can potentially remain always open. Size, degree of detail, format, color, shape, interactive trajectory, trajectory through space, duration, rhythm, point of view, the presence or absence of particular characters, the development of plot—to name just a few dimensions of cultural objects in different media—can all be defined as variables, to be freely modified by a user [...] Do we want, or need such freedom? As the pioneer of interactive filmmaking Grahame Weinbren argues, in relation to interactive media, making a choice involves a moral responsibility. By passing these choices to the user, the author also passes on the responsibility to represent the world and the human condition in it. (Ibíd., 44 )

Sobre esta última cita me parece importante notar que no se está tomando en cuenta que las variables y argumentos no son definidos de manera aleatoria, sino especificando qué y dónde se quiere dejar que el comportamiento, sea este sonoro, gráfico, numérico, etc. se desenvuelva de acuerdo a nuestras necesidades expresivas a la cual se le adhieren las automatizaciones y variabilidades implícitas.

### 2.2.5 TRANSCODIFICACIÓN

La transcodificación, el quinto y último principio de los nuevos medios que aborda Manovich involucra un término fundamental que ha aparecido a lo largo de este trabajo: *mapeo*, que está íntimamente ligado a la transcodificación “in new media lingo, to ‘transcode’ something is to translate it into another format”. (Ibíd., 47)

Manovich afirma que la nueva media se puede percibir consistiendo de dos capas distintas: “la capa cultural” y “la capa de computadora”. Como ejemplo de la primera categoría menciona la enciclopedia, el cuento corto, historia y trama, mimesis y catarsis, comedia y tragedia. Para la segunda categoría considera proceso y paquete (como datos transmitidos a través de una red), funciones y variables, lenguaje de programación y estructura de datos, entre otros. (Ibíd., 46)

Más adelante dice que la capa de la computadora y la capa cultural son entrelazados: “the computer layer and the culture layer influence each other [...] we can say that they are being composited together. The result of this composite is a new computer culture—a blend of human and computer meanings, of traditional ways in which human culture modeled the world and the computer’s own means of representing it.” (Ídem.)

Dentro de este último principio, que puede ser considerado como punto de convergencia del resto de los principios expuestos, es abordada la cuestión fundamental en la que, como Manovich afirma, la computarización de la cultura gradualmente crea una transcodificación en relación a todas las categorías y conceptos, “that is cultural categories and concepts are substituted, on the level of

meaning and/or language, by new ones that derive from the computer's ontology, epistemology , and pragmatics. New media thus acts as a forerunner of this more general process of cultural reconceptualization" (Ídem.) Quizá se pueda considerar que las afirmaciones son excesivas, sin embargo, creo que como he mencionado a lo largo de este trabajo, definitivamente existen cambios, influencias e intercambios entre el contacto que se tiene con la representación de esquemas musicales a través de estructuras dinámicas en un lenguaje de programación dentro de la computadora, definitivamente se le agrega una nueva dimensión, en la cual existe una compleja dialéctica entre la formalización y la intuición.

Al momento de estar leyendo los escritos de Manovich también estaba leyendo sobre la semiología musical de Nattiez, me pareció un símil de lo recién expuesto lo que menciona con respecto a la relación de un análisis con respecto a la obra: "From a tripartitional point of view, what, metalinguistically speaking is an analysis? I would claim, with Molino (1975a: 48), that it is a *surrogate symbolic behavior*—that is, a symbolic behavior that feeds upon another, primary symbolic behavior" (Nattiez, 1990: 153) Con la diferencia fundamental de que no es una forma simbólica que "se alimenta de otro", sino que construye y moldea la otra forma simbólica.

Tomando en cuenta que dentro del principio de la transcodificación convergen el resto de los principios expuestos hasta este punto, me parecería redundante seguir con más ejemplos dentro de la literatura musical o de las artes. Sin embargo

retomaré el concepto de este apartado dentro del análisis de las obras compuestas para ahondar en su discusión.

Ya habiendo expuesto en los apartados precedentes algunos de los principios característicos de los nuevos medios y su relación con manifestaciones culturales previas, en las siguientes secciones del presente capítulo quisiera abordar cómo son puestos en práctica y cómo se han adherido al proceso musical dentro de las obras compuestas para este trabajo. Además de los conceptos expuestos en este capítulo, hay una serie de distintos conceptos relacionados que serán abordados en su lugar correspondiente de los análisis, en aras de una mejor continuidad de discurso. Estos conceptos engloban básicamente: narrativa/bases de datos (que ya había mencionado anteriormente), cuestiones sobre autoría de textos y código, la expansibilidad permanente de código y la inminente obsolescencia del mismo. Finalmente, serán nuevamente tomados en cuenta para abordar la sección final del análisis de Xenakis en el capítulo 4.

## CAPÍTULO 3

### OBRAS ORIGINALES

*“Do I contradict myself?  
Very well then I contradict myself,  
(I am large, I contain multitudes.)”<sup>34</sup>*

— *Walt Whitman*

Dentro de este capítulo abordaré las piezas que compuse para este trabajo de la siguiente manera.

- Abarcando los detalles de implementación dentro de sus respectivos códigos y algoritmos empleados.
- Analizando distintos procesos que resultan a partir de algunos de los principios de la nueva media expuestos en la sección precedente, es decir, insertando las obras dentro de un marco de los estudios de software.
- Ubicar dichos procesos dentro del modelo semiológico tripartito del hecho musical de Jean Molino y semiología musical de Nattiez, con la finalidad de elucidar cómo es que el software y la programación se han inscrito en el hecho musical, no queda inafectado, ya que como se ha expuesto anteriormente, se adhieren características y tendencias propias del software al quehacer musical, y viceversa.

---

<sup>34</sup> ¿Me contradigo a mi mismo?

Muy bien, entonces me contradigo.

(Yo soy grande, contengo multitudes.)

(1855)



- Finalmente también plantearé motivaciones e ideas involucradas en las obras, cómo fueron germinando a lo largo del proceso creativo que resultó, dificultades con las que me encontré y conclusiones generales sobre la experiencia.

### 3.1 OBRAS IV Y VI

La implementación de procesos estocásticos dentro de un modelo compositivo implica que los resultados de las operaciones sean de alguna manera siempre distintas, siguen las mismas reglas, el mismo principio, pueden dar resultados similares o un comportamiento global semejante, sin embargo cada serie de iteraciones sobre un modelo estocástico genera una distribución distinta y estrictamente hablando única. Esta característica paradójica me cautivó profundamente por diversas razones las cuales resumiré en los siguientes tres puntos.

1. Por un lado lo considero como una analogía de la condición humana en el sentido de que cualquier decisión, por más trivial que parezca, puede marcar hondamente el camino de una vida;
2. de manera similar manifiesta cierta irrepetibilidad o unicidad de los objetos o eventos temporales, aunque en esencia o apariencia pueden ser semejantes;
3. me pareció sumamente interesante la obtención de resultados musicales distintos con estructuras esencialmente iguales, la infinidad de posibilidades del fluir de un discurso generado a partir de un mismo engranaje.

Es con esta motivación con la cual comencé a componer las obras *IV* y *VI*, siendo mis primeras exploraciones en la composición asistida por ordenador.

En un principio había planteado que formarían parte de una serie de doce obras que derivan de seis modelos algorítmicos que involucran procesos estocásticos, por lo tanto cada algoritmo se emplearía en dos piezas distintas que tendrían en esencia la misma estructura pero una evolución o distribución sónica propia. Además pretendía hacer una instalación sonora o intervención derivada de las piezas con material grabado del mismo concierto para ser difundidas en el recinto como si la energía acústica de las piezas se hubiese impregnado a las paredes. Sin embargo, se tomó la decisión de postergar el proyecto para un futuro debido principalmente a cuestiones de practicidad y la evolución de distintas inquietudes que explayaré en las demás obras. De igual manera se quedaron estas dos piezas dentro del trabajo en cuestión, así como la idea de aplicar una intervención sobre la obra *Anamnesis* para violín solo y también de alguna manera el concepto de aplicar los mismos procedimientos algorítmicos a dotaciones distintas o a distintas escalas, creando obras que son emparentadas, pero finalmente independientes.

### 3.1.1 DETALLES DE IMPLEMENTACIÓN DE *IV* Y *VI*

Cabe mencionar que en el momento de escribir el código para estas obras, llevaba relativamente poca experiencia programando en general y particularmente en SuperCollider, ya que mis primeras indagaciones en la programación eran principalmente dentro del lenguaje de programación visual Max/MSP<sup>35</sup>. Decidí

---

<sup>35</sup> <https://cycling74.com/>

mantener el código en su forma original, con todo y *bichos*<sup>36</sup> y fallas estructurales porque considero que a partir de ellos se pueden abordar fielmente detalles dentro del proceso poético, y asimismo ver cómo es que se adhirieron dichas fallas a este.

Las obras *IV* y *VI* fueron compuestas utilizando el siguiente software:

- SuperCollider
- FOMUS
- Sibelius

Para el planteamiento del desarrollo de las obras y para sus estructuraciones utilicé extensamente la librería de patrones de SuperCollider y de la clase Fomus<sup>37</sup> que es la que permite la comunicación entre SuperCollider y FOMUS. A dicha clase le hice unas modificaciones para resolver algunas cuestiones en las otras dos familias de obras, lo cual abordaré en su sección correspondiente. Para la edición de partitura utilicé el software Sibelius ya que estaba muy familiarizado con el programa, y tuvo un peso significativo en parte del proceso creativo.

El concepto básico del desarrollo de la obras *IV* y *VI* está concebida a manera de procesos acumulativos, disgregativos y de gradientes o *tendency masks* en los parámetros de ritmo, altura, espacio o índices acústicos y de evento no evento o sonido/silencio. Utilicé máquinas de estados para la toma de decisiones

---

<sup>36</sup> Es de práctica común dentro de la programación llamarle *bugs* o *bichos* a errores de programación dentro de un software o programa, causando errores, caídas del sistema o resultados inesperados, lo cual dentro del contexto artístico no es sinónimo de indeseado y puede influenciar decisiones *poéticas*.

<sup>37</sup> Escrita por Bernardo Barros, la cual modifiqué y amplié a Fomus2 para implementarlas en todas las obras de este trabajo.

automatizada en el desarrollo de cada uno de los atributos de manera relativamente independiente, aunque fuertemente inclinados a un desenvolvimiento deseado buscando establecer un balance entre procesos estocásticos y deterministas.

Una máquina de estado finito o autómata finito es un modelo matemático que realiza cálculos en forma automática sobre una entrada para producir una salida. Este modelo está conformado por un alfabeto, un conjunto de estados, que por su parte son un conjunto de situaciones o las condiciones en las que se halla un objeto en determinado momento; no se puede estar en más de un estado al mismo tiempo. También incluye un conjunto de posibles transiciones entre dichos estados siendo estas las probabilidades de encadenamiento entre los posibles distintos estados, y un estado final o de aceptación que finaliza el proceso y representa la salida. En este caso particular no fue deseable un estado final por lo que se dejó abierto a iteraciones *ad infinitum*.

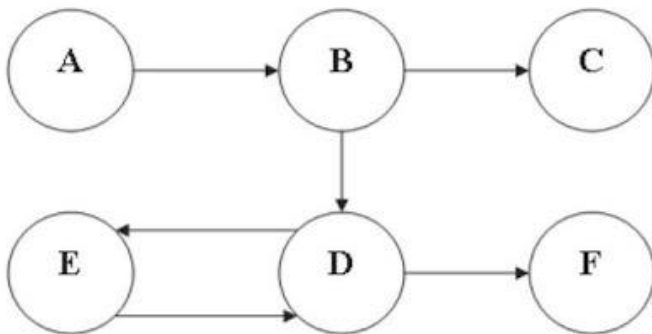


Figura 3.1 Ejemplo gráfico de una máquina de estados finitos.

Para la implementación dentro del lenguaje de programación, vinculé varios diccionarios (con la clase *Dictionary*) con varias instancias del objeto de máquinas de estados implementada dentro de la librería de patrones *Pfsm*.

Los diferentes diccionarios contienen arreglos de valores de duración, conjuntos de alturas y silencios, y operadores para establecer el índice acústico en el que se mapeará la altura, los cuales son asociados respectivamente a sus máquinas de estados donde se definen las probabilidades de transición. De esta forma, cada uno de los parámetros mencionados tiene un desenvolvimiento independiente, las máquinas van avanzando en la toma de sus decisiones de manera desfasada en cada uno de los parámetros que representa siendo tres máquinas por instrumento. En el caso de la obra *IV* nueve y en la obra *VI* doce máquinas de estado avanzan de manera independiente a la vez que conforman un comportamiento musical global compuesto de dichas estructuras y procesos modulares.

Para el diccionario de ritmos se emplearon unidades rítmicas derivadas de la sección áurea<sup>38</sup> de la duración total de la pieza, la cual a su vez en un inicio sería una sección áurea de la duración total del proyecto que se había concebido preliminarmente, sin embargo, ya en marcha, perdió prioridad esta relación, pero permanecieron los valores rítmicos y una proporción dorada entre ellas aunque redondeada e imperfecta, conservando hasta cierto punto propiedades de auto similitud como mosaicos de duración pseudofractales:

$$(1/13) \cdot 1.618033 = 0.12446407692308$$

---

<sup>38</sup> “The *golden ratio* is an irrational number defined to be  $(1 + \sqrt{5})/2$ . It has been of interest to mathematicians, physicists, philosophers, architects, artists and even musicians since antiquity. It has been called the *golden mean*, the *golden section*, the *golden cut*, the *divine proportion*, the *Fibonacci number* and the *mean of Phidias* and has a value of 1.618033...” (Dunlap, 2003: 1) Posee propiedades de auto-similitud proporcional y ha sido ampliamente utilizado en diversas áreas desde la antigüedad, siendo la música y las artes notorias.

quedando el diccionario con su respectiva máquina de estados de la siguiente manera<sup>39</sup>:

```

4 // Ejemplo 1
5 // p. 77, 78
6 (
7
8 // Máquina de estados fuertemente tendenciado
9
10 ~pfsmC = Pfsm([
11 [0], // 0 estado inicial
12
13 \j, [0,1], // 50% de probabilidad de quedarse en el estado 0 que mapea a 0.0769 en diccionario y 50% de irse a 0.125.
14 \k, [0,1,1,2,3], // 20 % quedarse en 0.125, 40% de irse a 0.2, y 20 para irse a 0.333 ó 0.5.
15 \l, [1,2,2,2,3,4], // 16.6% regresar a 0.125, 50% de quedarse en 0.2, 16.6% de irse a 0.333 ó 0.5.
16 \m, [0,1,2,3,4], // 20% de probabilidad de irse a los estados del 0 al 4, (0.0769, 0.125, 0.2, 0.333, 0.5, 1.0).
17 \n, [1,2,4,5], // 25% de probabilidad de quedarse en 1.0, regresar a los estado 1 o 2, o avanzar a 1.5.
18 \o, [1,2,3,4,5,6], // igual probabilidad de irse a los estados 1 al 6. 16.6% de seguir avanzando.
19 \p, [7,6,5,4,2], // 20% de seguir avanzando.
20 \q, [8,7,6,5,4,3,2], // 0.1428% de probabilidad de avanzar a último estado.
21 \r, [7,6,5,4,3,2,1], // si llega retrocede, distribución uniforme
22 nil, nil
23 ]);
24
25 ~pfsmC.asStream;
26
27 // duraciones de figuras derivadas de sección áurea
28 ~diccionarioC = Dictionary[
29 \j -> [1/13],
30 \k -> [1/8],
31 \l -> [0.2],
32 \m -> [0.333],
33 \n -> [0.5],
34 \o -> [1.0],
35 \p -> [1.5],
36 \q -> [2.4],
37 \r -> [4.0]
38 ];
39
40 // mapeo de máquina de estado finito a diccionario, se convierte en stream.
41 ~mapC = ~pfsmC.asStream.collect({ |i| ~diccionarioC[i]});
42 )
43
44 // valores individuales, crea problema con subdivisiones, aún no se puede utilizar
45 ~mapC.next;
46

```

Figura 3.2 Diccionario y máquina de estados para generar valores de duración.

Estas unidades representan respectivamente:

♩3:8 (0.0769...)      0.083578433782412

0.125 ♩      0.13523266394826

5:4 ♩ (0.2)      0.20138698377608

<sup>39</sup> Si se desean probar los códigos, se pueden descargar de manera gratuita en la siguiente página: <https://jorgeeduardojonas.wordpress.com/>

3:2 ♪ (0.333...)	0.35404327790707
0.5 ♪	0.5728537070818
1.0 ♪	0.92689620223069
1.5 ♪	1.4997486427839
2.4 ♪ + ♪ 5:4	2.4266427957296
4.0 ∞	3.9263881227028


Para la máquina de estados, como mencioné anteriormente, se plantearon probabilidades de transición fuertemente inclinadas a un discurso deseado.

Por lo tanto la máquina vinculada a los valores de duración tiene una fuerte tendencia de ir del valor mas corto hacia los valores largos, y una vez estando en los valores mas largos regresar a estados anteriores, como una especie de gradiente en *loop*, una máquina (3 o 5 por los instrumentos) atorada que se reinicia constantemente retomando el comportamiento.<sup>40</sup>

Además del diccionario y la máquina de estados, están vinculados a una función que encapsula los valores de duración dentro de un ritmo de una misma subdivisión (escogiendo una entre varias posibles maneras de completar el pulso), para no generar ritmos demasiados complejos o que confundan la computadora y provoquen la caída del sistema. Por ejemplo que no se sucedan los valores 13:8 ♪ y

---

<sup>40</sup> Aunque ciertamente hay muy poca probabilidad de que la máquina avance hasta el valor más largo 4.0.

5:4  dentro de la misma subdivisión. Finalmente, el contenido rítmico que se encuentra dentro de un pulso comprendido dentro de una colección o una instancia de la clase *Array*, es enviado un mensaje *.scramble*, que reordena el contenido de manera aleatoria, obteniendo aún más variabilidad.

```

48 // Ejemplo 2
49 // p. 79
50 // Switch para completar pulsos
51 (
52   ~ritmosC = Pfunc{switch( ~mapC.next, // si el valor es 2.4, se completa para no generar valores demasiado complicados, etc.
53     [4.0],      {[4.0]},
54     [2.4],      {[2.4] ++ [[0.2,0.2,0.2],[0.4,0.2].scramble].choose}, // provisional agregar ligadura
55     [1.5],      {[1.5] ++ [[0.5], [1.0, 0.5], [1.5]].choose},
56     [1.0],      {[1.0]},
57     [0.5],      {[0.5] ++ [[0.5],[0.25,0.25]].choose},
58     [0.333],    {[0.333] ++ [[0.333,0.333], [0.666], [0.16667,0.16667,0.16667], [0.333,0.16667,0.16667]].choose},
59     [0.2],      {[0.2] ++ [[0.2,0.2,0.2,0.2], [0.8], [0.2,0.2,0.4]].choose},
60     [1/8],      {if( 0.5.coin, {[1,[1,1,1,1,1,1,1]].convertRhythm.flat},
61                  {[1,[3,2,1,1,1].scramble].convertRhythm.flat})},
62     [1/13],     {if(0.5.coin,{[1/13 ! 13].flat}, {[1/13,2/13,2/13,1/13,1/13,1/13,1/13,3/13].scramble})}
63   );
64
65 );
66
67 ~ritmosConvC = ~ritmosC.asStream;
68 )
69 // genera Array que abarca rítmica de un pulso completo
70 ~ritmosConvC.next;
71 (
72 // para acceder al ritmo completo.
73 ~rCSwitch = Pfunc{switch(g = ~ritmosConvC.next; g.postIn.size,
74   1, {g[0]},
75   2, {Pseq([g[0],g[1]],inf).asStream.nextN(2)},
76   3, {Pseq([g[0],g[1],g[2]],inf).asStream.nextN(3)},
77   4, {Pseq([g[0],g[1],g[2],g[3]],inf).asStream.nextN(4)},
78   5, {Pseq([g[0],g[1],g[2],g[3],g[4]],inf).asStream.nextN(5)},
79   6, {Pseq([g[0],g[1],g[2],g[3],g[4],g[5]],inf).asStream.nextN(6)},
80   7, {Pseq([g[0],g[1],g[2],g[3],g[4],g[5],g[6]],inf).asStream.nextN(7)},
81   8, {Pseq([g[0],g[1],g[2],g[3],g[4],g[5],g[6],g[7]],inf).asStream.nextN(8)},
82   9, {Pseq([g[0],g[1],g[2],g[3],g[4],g[5],g[6],g[7],g[8]],inf).asStream.nextN(9)},
83  10, {Pseq([g[0],g[1],g[2],g[3],g[4],g[5],g[6],g[7],g[8],g[9]],inf).asStream.nextN(10)},
84  11, {Pseq([g[0],g[1],g[2],g[3],g[4],g[5],g[6],g[7],g[8],g[9],g[10]],inf).asStream.nextN(11)},
85  12, {Pseq([g[0],g[1],g[2],g[3],g[4],g[5],g[6],g[7],g[8],g[9],g[10],g[11]],inf).asStream.nextN(12)},
86  13, {Pseq([g[0],g[1],g[2],g[3],g[4],g[5],g[6],g[7],g[8],g[9],g[10],g[11],g[12]],inf).asStream.nextN(13)}
87 ).flatten;
88
89 ~durCorto = ~rCSwitch.asStream;
90
91 )
92
93 [~durCorto, ~ritmosConvC, ~mapC].do(_.reset);
94 // para generar valores individuales que serán pasados a Fomus
95 ~durCorto.next;
96

```

Figura 3.3. Vinculación de máquina de estados con función que completa los valores de los pulsos con duraciones de la misma subdivisión.

Entonces por ejemplo si se selecciona el valor 2.4, que en este caso sería una blanca ligada a una corchea de cinquillo de semicorcheas, se completará el pulso con las tres semicorcheas de cinquillo restantes o con una semicorchea y una corchea de



cinquillo de semicorchea en cualquier orden. La misma lógica subyace con los valores de duración restantes.

Para el parámetro de las alturas, igualmente están vinculados un diccionario y una máquina de estados. El diccionario de las alturas consta, partiendo del símbolo  $\backslash a$  con un conjunto que se compone de tres clases alturas<sup>41</sup>  $\{0, 6, B\}$ ,  $\backslash b$  con cuatro clases altura  $\{0, 5, 6, B\}$ , cada letra del diccionario acumulando una altura llegando hasta el símbolo  $\backslash j$  constituido del agregado cromático. Dentro de la manera en la que están compuestos los conjuntos de alturas existe un alto grado de auto similitud en sus formas normales o formas más compactas:

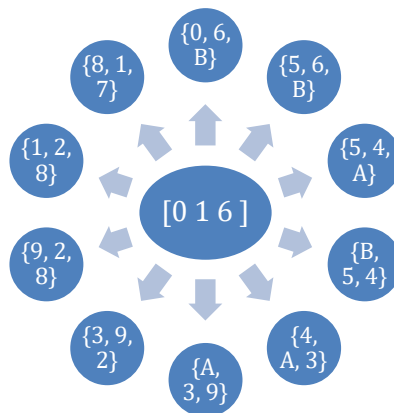


Figura 3.4 Conjuntos de alturas utilizadas en la obra.

Esto se puede asemejar un poco a los procedimientos seriales Weberianos, sin embargo, en este caso no se concibe como una serie dado que el tratamiento es distinto, la manera en que se van seleccionando las notas por medio de una función aleatoria del conjunto, como entidades sonoras sin orden, que ya ligado con la

<sup>41</sup> Refiero al lector a la literatura tradicional de teoría de música atonal anglosajona (Forte, 1973), (Morris, 1987) (Lewin, 1987), ya que un repaso de las bases teóricas excederían las dimensiones y propósitos de este trabajo.

máquina de estados propone un discurso acumulativo y expansivo. A pesar de que esta entidad sonora es percibida más que nada dentro de la estructura del código dentro del alfabeto de la máquina de estados, la misma naturaleza de la máquina hace que retroceda una vez que se pierde en el proceso acumulativo, forzando a que retorne a su “identidad” original, como un filtro que es aplicado a una imagen distorsionándola hasta tornarla irreconocible.

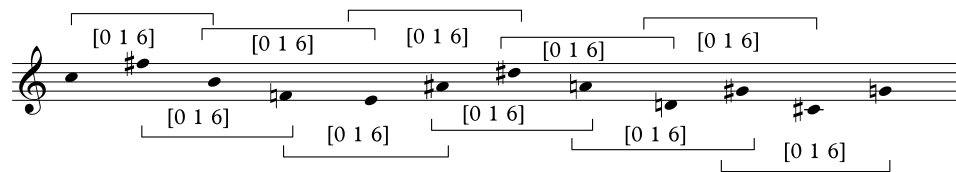


Figura 3.5 Conjuntos de alturas con propiedades de similitud utilizado en la obra.

Por último, se tiene nuevamente otro diccionario y otra máquina de estados para el registro o índice acústico en el que será seleccionada la altura, dentro de esta parte también se vincula la relación entre sonido y silencio, planteando distintos escenarios en los que el predominio de alguno de los dos elementos sonido o silencio es intercalado, y las alturas son restringidas a cierto(s) índice(s) acústico(s) momentáneamente. Por ejemplo en el comienzo hay una fuerte probabilidad (80 %) de que se escoja un silencio, y si se escoge una altura será dentro de la octava central, posteriormente se mantiene una fuerte probabilidad de silencio pero incrementa la posibilidad de sonido y abriéndose este a la octava central y una octava por encima (índices acústicos 4 y 5 respectivamente).

```

336 {
337 ~p fsmI = P fsm([
338 [0], // Estado inicial, 0.
339
340 \a, [0,0,0,0,1], // 20% de probabilidad de agregar notas, en estados del 0 al 8; 80% de probabilidad de permanecer con el
341 \b, [1,1,1,1,2], // mismo número de notas.
342 \c, [2,2,2,2,3],
343 \d, [3,3,3,3,4],
344 \e, [4,4,4,4,5],
345 \f, [5,5,5,5,6],
346 \g, [6,6,6,6,7],
347 \h, [7,7,7,7,8],
348 \i, [8,8,8,8,9],
349 \j, [9,9,9,9,1] // En el estado 9, hay un 80% de probabilidad de permanecer en el agregado, y un 20% de probabilidad de
350 // reiniciar el proceso.
351 ]);
352 };
353
354 ~p fsmI.asStream;
355 // Diccionario para notas
356 {
357 ~diccionarioI = Dictionary[
358 \a -> [60,66,71], // comienza con 3 notas
359 \b -> [60,66,71,65],
360 \c -> [60,66,71,65,64],
361 \d -> [60,66,71,65,64,70],
362 \e -> [60,66,71,65,64,70,63],
363 \f -> [60,66,71,65,64,70,63,69],
364 \g -> [60,66,71,65,64,70,63,69,62],
365 \h -> [60,66,71,65,64,70,63,69,62,68],
366 \i -> [60,66,71,65,64,70,63,69,62,68,61],
367 \j -> [60,66,71,65,64,70,63,69,62,68,61,67] // termina con agregado, con la posibilidad de reiniciar hacia 3 notas.
368 ];
369
370 ~mapeoI = ~p fsmI.asStream.collect({ |i| ~diccionarioI[i]});
371
372 {
373 ~nIswitch = Pfunc{switch( a = ~mapeoI.next; a.postln.size,
374 3, {Prand([a[0],a[1],a[2]],inf).asStream.nextN(3)},
375 4, {Prand([a[0],a[1],a[2],a[3]],inf).asStream.nextN(4)},
376 5, {Prand([a[0],a[1],a[2],a[3],a[4]],inf).asStream.nextN(5)},
377 6, {Prand([a[0],a[1],a[2],a[3],a[4],a[5]],inf).asStream.nextN(6)},
378 7, {Prand([a[0],a[1],a[2],a[3],a[4],a[5],a[6]],inf).asStream.nextN(7)},
379 8, {Prand([a[0],a[1],a[2],a[3],a[4],a[5],a[6],a[7]],inf).asStream.nextN(8)},
380 9, {Prand([a[0],a[1],a[2],a[3],a[4],a[5],a[6],a[7],a[8]],inf).asStream.nextN(9)},
381 10, {Prand([a[0],a[1],a[2],a[3],a[4],a[5],a[6],a[7],a[8],a[9]],inf).asStream.nextN(10)},
382 11, {Prand([a[0],a[1],a[2],a[3],a[4],a[5],a[6],a[7],a[8],a[9],a[10]],inf).asStream.nextN(11)},
383 12, {Prand([a[0],a[1],a[2],a[3],a[4],a[5],a[6],a[7],a[8],a[9],a[10],a[11]],inf).asStream.nextN(12)}
384 }.flatten;
385 };
386
387 ~notasI = ~nIswitch.asStream;
388 };

```

Figura 3.6 Diccionario, máquina de estados y función para seleccionar alturas.

Conforme va avanzando el autómata se va invirtiendo el predominio de sonido sobre silencio a la vez que va planteando un discurso con cambios de registros que se abren, se cierran, se crean trinchas de registros, finalizando con ataques solamente en las notas más graves o más agudas del instrumento y aquellas que comprende el rango central, y de nuevo, un predominio de silencio sobre sonido. En resumen, recopilando todos los datos y procesos expuestos tenemos una pieza con una direccionalidad generalizada, que va de valores cortos a valores más largos, al mismo tiempo se incrementa el número de alturas, hay una direccionalidad de registro y a la vez una modulación de dominación de silencio sobre sonido que se

revierte y posteriormente retorna a un estado similar, todos estos procediendo de manera automatizada y relativamente independientes pero con tendencias deseadas.

```

393 // autómeta para octavas
394 {
395 {
396 ~p fsmOctI = P fsm([
397 [0], // Estado inicial 0.
398
399 \a, [0,0,0,1], // 75% de probabilidad para permanecer en el mismo estado, 25% para avanzar.
400 \b, [1,1,1,2],
401 \c, [2,2,2,3],
402 \d, [3,3,3,4],
403 \e, [4,4,4,5],
404 \f, [5,5,5,6],
405 \g, [6,6,6,7],
406 \h, [7,7,7,8],
407 \i, [8,8,8,9],
408 \j, [9,9,9,10],
409 \k, [10,10,10,11],
410 \l, [11,11,11,12],
411 \m, [12,12,12,13],
412 \n, [13] // Estado final, aquí permanece.
413 ]);
414 };
415
416 ~p fsmOctI.asStream;
417
418 // Diccionario para octavas
419 {
420 {
421 {
422 {
423 // Tesitura para piano (se modifica para clarinete y para flauta)
424 |
425 ~d iccionarioOctI = Dictionary[
426
427 \a -> [0,\rest ! 4].flat, // Dominan silencios, octava central.
428 \b -> [0,12,\rest ! 4].flat, // Octava central más una octava superior.
429 \c -> [-12,0,12,\rest ! 6].flat, // Se agrega octava inferior.
430 \d -> [0,\rest ! 2].flat, // Empieza balance silencio-sonido, octava central.
431 \e -> [-24,0,24,\rest ! 3].flat, // Octava central mas octava inferior*2, superior*2.
432 \f -> [-12,0,12,\rest ! 2].flat, // Domina sonido, se cierra trinche octava central + inferior + superior.
433 \g -> [-36,-24,-12,0,12,24,36,\rest ! 3].flat, // Rango completo de 7 octavas sin trinche.
434 \h -> [0,0,\rest], // Octava central. Sigue dominando sonido. 2:1.
435 \i -> [-48,-36,-24,-12,0,12,24,36,48,\rest ! 3].flat, // Empieza filtraje hacia silencio.
436 \j -> [-48,-36,-24,-12,0,12,24,36,48,\rest ! 5].flat,
437 \k -> [-48,-36,-24,-12,0,12,24,36,48,\rest ! 8].flat,
438 \l -> [-48,-36,-24,0,24,36,48,\rest ! 13].flat, // Se empieza a formar "trinche" de rango.
439 \m -> [-48,-36,0,36,48,\rest ! 13].flat,
440 \n -> [-48,0,48,\rest ! 13].flat // Octava central más extremo agudo y grave, sonido-silencio 3:13.
441 ];
442
443 });
444
445 ~m apeoOctI = ~p fsmOctI.asStream.collect({ |i| ~d iccionarioOctI[i]});
446 //~mapeoOctI.next;
447

```

Figura 3.7 Diccionario y máquina de estados para registro de alturas.

Posteriormente los valores son pasados por un Pfunc, de manera similar a la expuesta con las alturas y duraciones. En este caso escogerá de manera aleatoria un silencio o un operador que subirá, bajará o mantendrá el índice acústico de la altura seleccionada.

Finalmente se empaqueta todo como eventos dentro de un Pbind y es pasada como una rutina dentro de la clase Fomus2 en el cual se especificará el número de eventos

que quiere que se calcule, dentro del segundo argumento. Si se ejecuta la instancia de Fomus2 con el método `.ly` se podrá abrir el archivo en PDF, y si se ejecuta la instancia de Fomus2 con el método `.xml` se podrá abrir con un editor de partitura como Sibelius o Finale.

El lector se puede referir al Apéndice donde podrá encontrar el código en su forma completa.

A continuación se puede apreciar el desenvolvimiento de la parte del piano después de ejecutar el código y exportarlo tal cual en formato MIDI al *piano roll* del software *Audacity*, es una variante de la versión original.

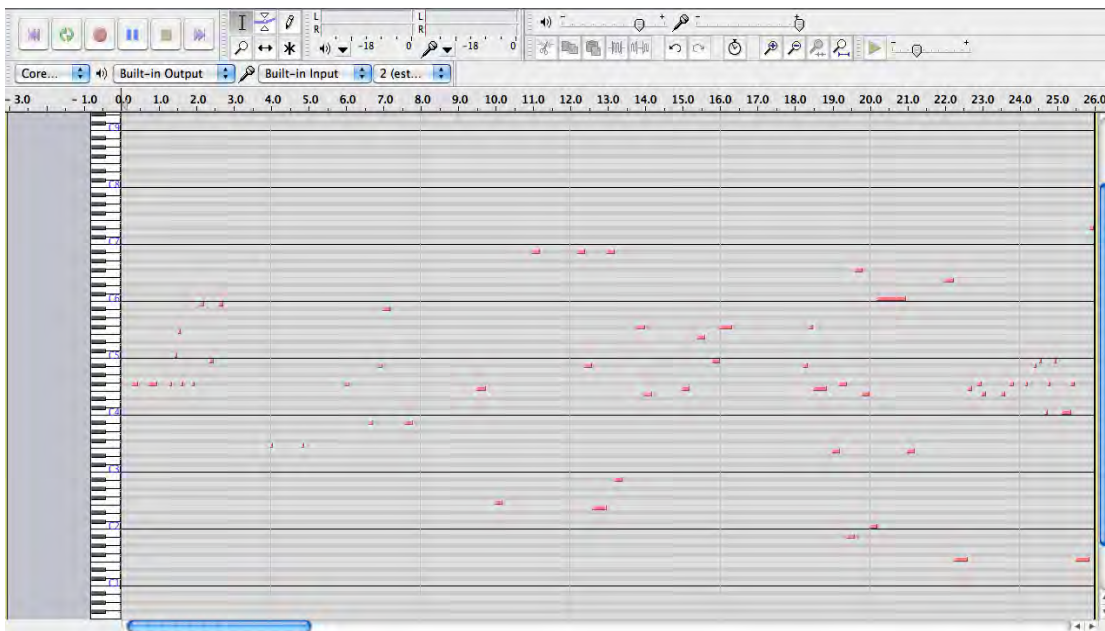


Figura 3.8 Piano roll 1.

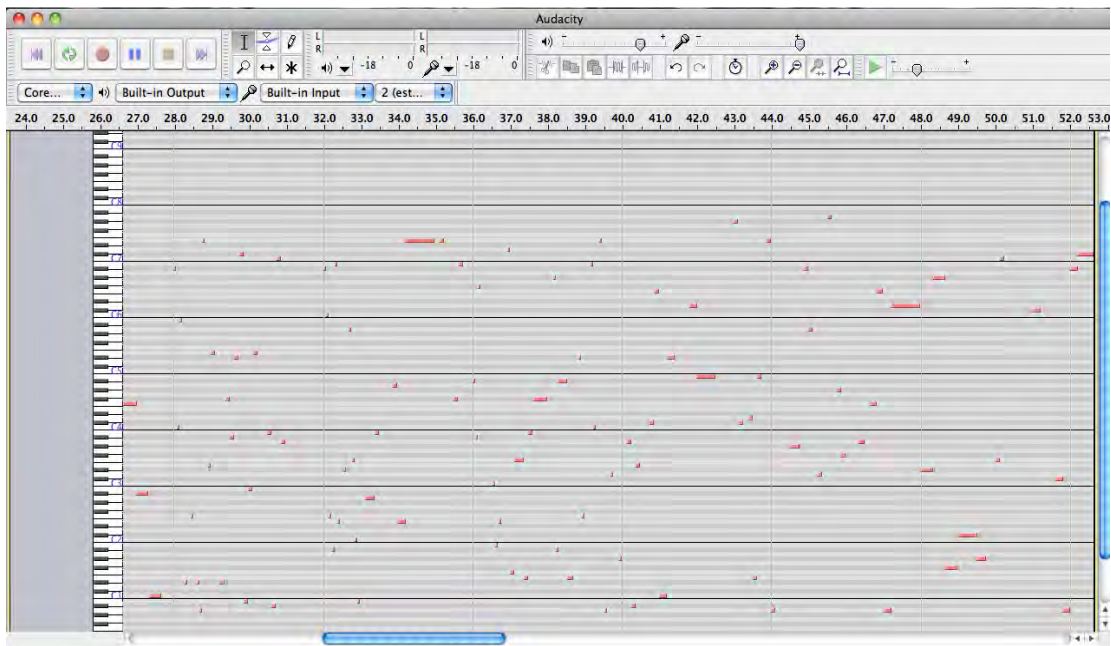


Figura 3.9 Piano roll 2.

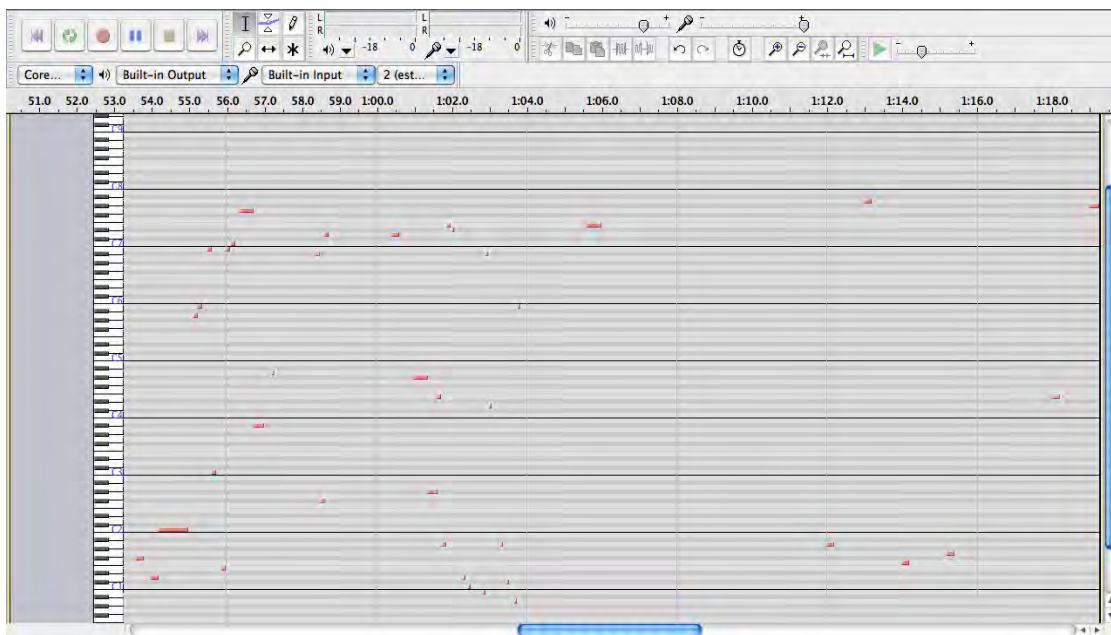


Figura 3.10 Piano roll 3.

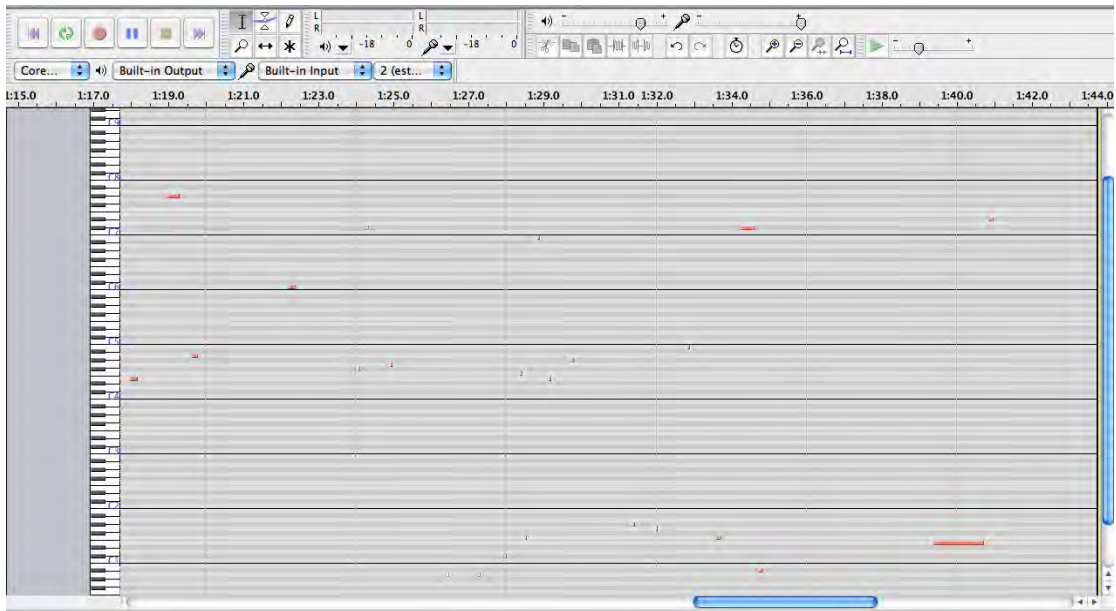


Figura 3.11 Piano roll 4.

Los ejemplos de código precedentes fueron extraídos de la parte del piano de la obra *IV*, todos los instrumentos tienen una estructura muy similar, solamente variando en su registro correspondiente. La obra *VI* tiene una estructura esencialmente idéntica<sup>42</sup>, con ligeros cambios de valores, siendo ambas obras como reflejos distorsionados entre sí mismas.

### 3.1.2 ANÁLISIS RETROSPECTIVO DE LAS OBRAS *IV* Y *VI*

Ya habiendo expuesto la implementación que sirve para las obras *IV* y *VI*, en este apartado haré un breve análisis de cómo considero que influyó sobre mi proceso creativo mi primera aproximación a la *CAO* en el momento en el que estaba aprendiendo a programar en SuperCollider. Sin embargo, antes de entrar en detalles, me parece oportuno revisar brevemente la terminología del modelo

<sup>42</sup> Anexo en el apéndice el código fuente completo de ambas obras.

tripartito utilizado en la semiología musical de Jean Jacques Nattiez, con la cual me auxiliaré en secciones subsiguientes.

Como se mencionó en apartados precedentes, el modelo tripartito que adopta Nattiez, influenciado por Jean Molino (Molino, 1975), distingue tres niveles de distintos dentro de un hecho musical, constando de los niveles *poiético*, *neutral o inmanente* y *estésico*. correspondiendo *grosso modo* a la creación, al trazo o resultado y a los procesos de recepción, lo cual no es pasivo si no un proceso de *percepción y construcción de significado* activa.

El nivel poiético se ocupa principalmente de todos los aspectos relacionados con la creación de una obra musical, abarcando el proceso creativo, “a *process of creation that may be described or reconstituted*” (Nattiez, 1990: 12) más adelante retorna a la definición de lo poiético por Etienne Gilson (1963) de quien Molino tomó el término, repasando lo que *poiético* significaba para Gilson, quien lo dividía en tres elementos: “(1) deliberations on what must be done to produce the object; (2) operations upon external materials; (3) the production of the work.” (Nattiez, 1990: 13).

Molino extendió esta definición, incluyendo el estudio de técnicas y reglas de generación, análisis de estrategias de producción “which, from evidence and clues left by the author, or from characteristics of the work itself, serve to furnish a model for the production of the work”. Por otro lado también incluye *las intenciones del autor* “who in the plastic arts or in literature often wants to communicate or express



something about the work”, y finalmente, el *significado expresivo* consciente o inconsciente que puede encontrarse en el trabajo. (Molino, 1984; citado en Ídem.)

El nivel estésico que abarca los *procesos* de recepción y *construcción* de significado(s) de la obra. El término era un neologismo de Paul Valéry (Valéry, 1945), que abarca “Enjoying, contemplating or reading a work, musical performance, as well as scientific and analytical approaches to music” (Ibíd., 12) y consiste en un complejo proceso de recepción, que reconstruye el mensaje. (Ibíd., 17)

Finalmente el nivel neutro consta del *trazo*, el resultado físico de los procesos poiéticos, la partitura, una grabación, “the symbolic form is embodied physically and materially in the form of a *trace* accessible to the five senses.” (Ibíd., 12) Más adelante añade que “Between the poietic process and the esthetic process there exists a *material trace*, not in itself bearer of an immediately decipherable meaning, but without which meaning(s) could not exist”. (Ibíd., 15)

Examinando estos tres niveles, Nattiez afirma que el analista podría aprehender el hecho musical en su totalidad: “The *essence* of a work is at once its genesis, its organization, and the way it is perceived.” (Nattiez, 1990: ix)

Ya habiendo expuesto un breve repaso de lo que comprende cada uno de los elementos del modelo tripartito, seguiré con el análisis retrospectivo de las obras *IV* y *VI*.

FOMUS<sup>43</sup> es un software libre escrito por David Psenicka que automatiza muchas de las tareas involucradas en la notación musical convencional, permite al usuario separar los datos en parámetros musicales como alturas, duraciones y de manera un poco más limitada (sobre todo en grafía de música contemporánea) articulaciones o modificaciones tímbricas. Esto *puede* predisponer a componer pensando principalmente en los parámetros de altura y duración, en detrimento del discurso tímbrico.

*Estésicamente* se podría decir que la obra realmente no es afectada, el formato de concierto es tradicional<sup>44</sup>, el escucha no se ve sujeto a mayor cambio dentro de la experiencia de la obra, también, para la ejecución<sup>45</sup> no representa alguna transformación debido al uso del ordenador, la partitura sigue siendo la misma, por lo tanto puede pasar desapercibido que se empleó un lenguaje de programación para su concepción.

Dentro del nivel *neutro* aparentemente tampoco hay cambio en el sentido de que la partitura sigue siendo una partitura, y es más, con una notación tradicional y sin

---

<sup>43</sup> <http://fomus.sourceforge.net>

<sup>44</sup> Hablo de formato de concierto pensando en como puede ser percibida y experimentada la obra, es decir, desde un punto de vista estésico.

<sup>45</sup> Aquí podemos llegar a una pregunta fundamental ¿Dónde termina el nivel poiético y donde comienza el nivel estésico? "If we conceive of the work as an entity comprised of relations that are fixed by the score, the graphic sign (the score) *is* the work, and the esthetic process begins at the instant the performer *interprets* the work, in both senses of the word: (a) the performer performs the work, (b) the performer makes a personal selection of interpretants, from the moment of the first reading of the work." (Nattiez, 1990: 72) Por otro lado, se puede adoptar la postura en la cual la obra no se ha producido completamente, o está inacabada hasta que haya sido ejecutada, por lo tanto el proceso poiético se extiende hasta la ejecución, como el último paso del nivel poiético y el primero del estésico.

necesidad de página de indicaciones de ejecución. Sin embargo, visto de cerca, considero que el nivel neutro sí queda trastocado ya que el código ejecutable al ser un texto analizable se inscribe al nivel inmanente, con todo y sus propiedades dinámicas.

Considero que esto es sumamente importante en un nivel de análisis porque permite reconstruir algunos de los procedimientos que hizo el compositor, e inferir decisiones de composición provocadas por el mismo, además, permite que la persona que lo esté analizando (suponiendo que sea versado en algún lenguaje de programación y compositor) se *apropie* de algunas de las estructuras para crear obras completamente distintas, más sobre este importante punto en la sección de conclusiones generales al final del capítulo.

Finalmente me parece importante hacer notar que Nattiez describe los polos poiéticos y estésicos como procesos, mientras el polo neutro lo describe como una estructura, "Poietic and esthetic phenomena are *processes*; immanent configurations are organized into *structures* or *quasi-structures*." (Nattiez, 1990: 15) Con la inclusión del código al corpus del nivel inmanente introduce sus propiedades performativas adquiriendo este polo dinamismo e interacciones distintas. Se podría argumentar que quien analiza la música puede tocar la obra en un instrumento o escuchar alguna ejecución y que aquí también subyacería una interactividad, y en efecto sí que lo existe, sin embargo es de naturaleza fundamentalmente distinta porque al ejecutar el código se está *ejecutando* la estructura, se impulsa la máquina con una interacción dentro de este polo que considero sin precedente.

Considero que dentro del nivel poético es más evidente la manera en que se inscribe el software al hecho musical, algunos puntos que creo importantes y que experimenté al componer estas obras son los siguientes.

Antes de escribir el código tenía en mente algunas de las ideas de la forma general de la pieza, como quería que se cerraran, expandieran y formaran trinchas de registros los instrumentos, así también relaciones de transformaciones en forma de gradientes entre otras cuestiones. Al empezar a escribir el código y encontrarme con múltiples problemas de programación, fueron modificándose algunos de los conceptos generales, es decir, a causa de mi poca experiencia en el momento que empecé a programar estas obras, resultaban bichos que generaban comportamientos y resultados inesperados que no me desagradaron y que decidí últimamente incluir dentro de la obra. Un ejemplo de esto es la función que mencioné que completa valores rítmicos.

Por otro lado también generaba una interacción de prueba y error, al cambiar unas cuantas variables podrían darse transformaciones muy distintas, podía probar por ejemplo, con distintas longitudes de secciones, transformaciones más rápidas o más lentas y revisarlas inmediatamente, aquí aunque no hubiese algún problema con la escritura del código, la automatización me permitía ver casi en tiempo real como podía crear discursos variables teniendo una visión macroscópica del desarrollo del algoritmo en términos musicales, para luego “a mano”<sup>46</sup> entrar en detalles

---

<sup>46</sup> Además de que era necesario corregir algunos detalles de la versión que arroja el programa, por ejemplo agrupaciones rítmicas, notas que sobrepasaban el registro del instrumento en turno.

microscópicas. Este principio de automatización y variabilidad me generaban intersecciones completamente inesperadas entre los instrumentos que intensificaba o realizaba “a mano” dentro del editor de partitura.

También encontré que las mismas estructuras algorítmicas las utilicé de manera modular, las mismas construcciones que hice con los mapeos de diccionarios a máquinas de estados, fueron empleados en las variables de los parámetros musicales que estaba tomando en cuenta, el principio de modularidad también se estaba inscribiendo en mi proceso creativo, y más allá la misma estructura del código iba presentando propiedades de auto-similitud en distintos niveles de resolución.

Además pronto me encontré con el problema de que varios pasajes resultantes me gustaban, creaba una pequeña base de datos con las opciones y finalmente escogía uno o de plano unía ambas y desechaba el resto, esto se asemeja un poco a prácticas de copy paste y *collage*. Sentía la impresión de tener fragmentos *materiales* de frases e intersecciones que podía estar desplazando con comandos de copy paste ya en la edición de la partitura dentro de Sibelius. Por lo tanto no solamente el código sino el software de edición afectaba el proceso creativo.

Por otro lado, estéticamente se le dio una fuerte prioridad a las variables de altura y duración, ya que son los que privilegia Fomus y los que son fácilmente representables y mapeados a notación musical, que también favorece estos aspectos. También sentí que con el paso extra de abstracción que por momentos puede llegar a establecer la programación, era fácil perder un poco de control sobre

nociones de practicidad e idiosincrasia para la ejecución, resultando en esta ocasión en una partitura bastante difícil de ejecutar.

Recapitulando, a primera vista se podía llegar a la conclusión de que el hecho musical quedaba inmutado en este caso, que el uso del ordenador había quedado desapercibido, sin embargo, un análisis más cercano nos permite apreciar que realmente se queda inscrito de distintas maneras dentro del nivel inmanente y considero que sobre todo dentro del nivel poiético.

### 3.2 ANAMNESIS Y 27 DE A 40 GRADOS

La siguiente familia de obras comprende *Anamnesis*<sup>47</sup> para violín solo y *27 de a 40 grados*<sup>48</sup> para quinteto de cuerdas (dos violines, viola, violoncelo y contrabajo). En estas obras también se utilizan algoritmos ligados a la probabilidad y toma de decisiones automatizada. Además se exploran los movimientos brownianos<sup>49</sup> para controlar el comportamiento de movimiento de altura que a su vez son delimitadas por “fronteras” fluctuantes, sistemas de Lindenmayer<sup>50</sup> (sistema-L) estocásticos

---

<sup>47</sup> Obra que resultó ganadora del pedido internacional de obras en el XI Festival Internacional de Música Nueva Monterrey 2014 estrenada por el violinista Juanmanuel Flores el domingo 16 de noviembre de ese año.

<sup>48</sup> Estrenada el viernes 21 de noviembre del 2014 por el ensamble Liminar en el marco del XI Festival Internacional de Música Nueva de Monterrey.

<sup>49</sup> Los movimientos brownianos son movimientos o desplazamientos de moléculas que se encuentran dentro de un estado líquido o gaseoso, que resultan de la colisión con las demás moléculas presentes. Xenakis comenzó a utilizar los movimientos brownianos o *random walks* en su música instrumental y electroacústica, siendo obras donde empleó esta técnica las siguientes: Mikka (1971), Cendrées (1973), N'Shima (1975), Theraps (1975-6), Jonchaies (1977), Ikhoor (1978), entre otros, para una lista completa referirse a Solomos en: Iannis Xenakis (Mercuès: PO Editions, 1996)

<sup>50</sup> Los Sistemas L o sistemas de Lindenmayer son una gramática formal que simulan el crecimiento de hojas y ramificaciones en plantas.

para generar una gramática formal de articulaciones, así como también se utilizaron una serie de *gradientes* o *tendency masks* que van *modulando* parámetros o probabilidades que en su totalidad va estableciendo el discurso musical global resultante.

A raíz de la fuerte inclinación que sentí al trabajar las dos obras anteriores a pensar en alturas y duraciones en detrimento del timbre u otro desenvolvimiento del sonido, como se verá más adelante, en esta pieza se tomó la decisión de tener un acercamiento un tanto distinto. Buscando por ello establecer un mayor balance entre las prioridades de atributos o propiedades del sonido y su desarrollo dentro de la obra, así como un contacto más directo con el instrumento, sobre todo en *Anamnesis*.

Al igual que en las obras *IV* y *VI* existe una relación homóloga entre *Anamnesis* y *27 de a 40 grados*, al partir ambas piezas de esencialmente los mismos algoritmos y formas muy parecidas se verá que tienen discursos muy afines que son sujetos a *escala* entre otras diferencias que se abordarán en el presente apartado.

Finalmente, en la obra *Anamnesis* planteé una intervención sonora a partir del mismo material de la pieza, el cual puede ser grabado *in situ*<sup>51</sup> dentro del concierto para posteriormente ser procesado a partir de automatizaciones que controlan distintos parámetros de síntesis de sonido, aplicados a las grabaciones de la obra recién ejecutada. El programa que escribí para la parte de la intervención toma

---

<sup>51</sup> Para una futura implementación, ya que en la actual se manipulan fragmentos de la grabación de la ejecución.

decisiones basados en atributos acústicos y psicoacústicos de los archivos de audio de la obra acústica; puede ser ejecutado y dejar que corra solo, o también se puede manipular desde SuperCollider interviniendo las automatizaciones para que tenga una direccionalidad deseada, combinando improvisación y *robots* musicales. Además se utilizaron distintas versiones del mismo programa, de manera independiente sin la obra acústica y con otros archivos de audio como fuente para participar en distintas sesiones de improvisación en vivo en el 2013 y 2014. Esto pone en evidencia los principios de modularidad y variabilidad que abordaré después de exponer los detalles de implementación.

### 3.3.1 DETALLES DE IMPLEMENTACIÓN

Cabe mencionar que no concebí la pieza como las piezas *IV* y *VI* en el sentido de que toda la estructura de la obra se encuentra dentro del código fuente, sino que fue pensada por secciones que se trabajaron de manera distinta, con diferentes códigos o incluso algunas pequeñas secciones que no involucran la computadora, no obstante, todo englobando dentro del concepto general de la obra. Por lo tanto a continuación expondré los detalles de implementación de las secciones que sí incluyen programación dentro de *Anamnesis* y de *27 de a 40 grados*, y posteriormente lo uniré con los fragmentos que no, la manera en que están vinculadas y porqué se tomó la decisión de crear fragmentos dentro del ordenador y otros no, así como las particularidades de cada una ya que no son idénticas, aunque si parten de la misma raíz.



Como se mencionó al inicio de esta sección, ambas piezas controlan el comportamiento de altura a partir de movimientos brownianos, el rango dentro del cual se mueve está limitado dentro de dos barreras fluctuantes, una superior y otra inferior que se van modulando de acuerdo a la sección en que se encuentre, por lo tanto, estas barreras marcan fuertemente la direccionalidad del flujo de las alturas, que se puede ver en el siguiente *plot* dentro de la figura 3.11.

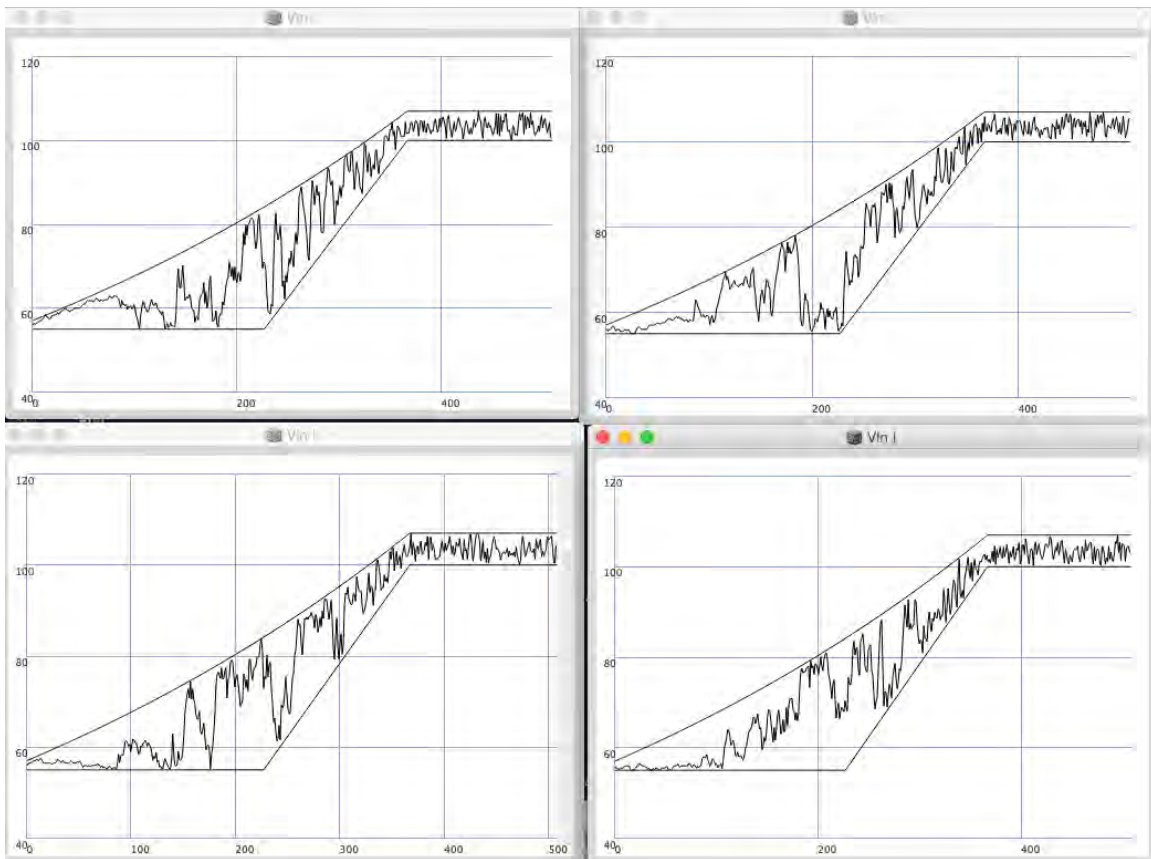


Figura 3.12 Múltiples opciones de la parte del Violín I en una sección de 27 de a 40 grados, barreras que delimitan rango del movimiento browniano, el cual a su vez es modulado en tamaño de pasos. Eje horizontal representa tiempo en una grilla de 0.125 de segundo, y eje vertical alturas en números MIDI

Cada vez que se ejecuta el código para alguno de los instrumentos, se generan las mismas barreras para el mismo número de valores (dentro del registro respectivo de cada instrumento), con un movimiento browniano que se desenvuelve de manera

distinta dentro de los rangos establecidos. Dicho movimiento browniano a su vez esta siendo sujeto a una modulación dentro del argumento del tamaño de paso, que musicalmente se mapea como pasos de intervalo, entonces hay momentos en los que podrá moverse por intervalos más amplios o dentro de un rango más estrecho. También esta presente el caso en el cual existe un rango de paso común o predominante dentro de un determinado tiempo, en donde se insertan detonantes con baja probabilidad de ocurrir, que se traducen en cambios interválicos bruscos y momentáneos.

Con todo lo anterior tomo la decisión de mantener un comportamiento errático de los movimientos aleatorios, pero que son controlados con una direccionalidad marcada por las barreras de rango fijadas, y también por medio de la modulación del tamaño de intervalo, de nuevo buscando establecer un balance entre la dicotomía determinista / estocástica de acuerdo a las necesidades planteadas.

Los momentos en que se van generando las modulaciones de probabilidades y otros cambios que definen el comportamiento musical dentro de la obra, están divididas en distintas secciones o fases. Hay un numero total de eventos arbitrariamente fijado en 368 a la cual se le calcularon distintas proporciones áureas y proporciones áureas entre valores adyacentes que son las fronteras donde comienzan o culminan procesos de transformación, estas secciones son:

I 87            II 141            III 227            IV 368

I' 120            II' 194            III' 314            IV' 509

Lo que se acaba de plantear sobre los movimientos brownianos, sus barreras de registro, el tamaño del paso del movimiento, entre otros elementos que expondré a continuación, están ubicados dentro de estas fronteras de números de eventos que tienen una estructura seudofractal, dentro de cada uno de estas fronteras se están llevando a cabo modulaciones de parámetros, en las que suben o bajan ciertas probabilidades a manera de gradiente o de oleajes.

Por lo tanto las barreras de registro cambian de la siguiente manera: el límite del movimiento inferior se encuentra dividido en tres segmentos:

1. desde el inicio hasta llegar a la sección III (227 valores) en donde permanece en el registro extremo grave del instrumento;
2. después de la sección III hasta llegar a la IV (368 valores) va subiendo hasta empujar el límite inferior de la barrera hasta el registro extremo agudo del instrumento;
3. aquí permanece hasta el final de los eventos. La barrera superior solamente tiene dos segmentos, desde el inicio hasta llegar a IV (368 valores);
4. posteriormente en IV' llega a un punto máximo fijo, que junto con la barrera inferior acorralan los movimientos melódicos en un espacio reducido en registro extremo agudo. Estas barreras se pueden apreciar en la figura 9.

Al mismo tiempo el movimiento browniano está siendo alterado de múltiples maneras, por un lado como ya se mencionó, el tamaño del paso está siendo modificado, de acuerdo a la sección en la que se encuentra, en la sección I sus

desplazamientos son principalmente por medio de movimientos de cuarto de tono, mientras va pasando por las siguientes secciones va incrementando este movimiento, ya en la sección IV el tamaño del desplazamiento es seleccionado por medio de una función aleatoria de distribución uniforme de un arreglo de valores que puede llegar a ser mayor de una octava.

```

147 // ALTURAS
148 // Limites de movimiento browniano
149 ~hi = Pgeom.fromEndpoints(57.0, 107.0, ~numEvTot).asStream.all;
150 ~segLo = Pseq(Pseries.fromEndpoints(55.0,100.0,~numEvTot - ~aurFase[2]).asStream.all, inf).asStream;
151
152 ~lo = Routine{
153   (~numEvTot + ~mes).do{|i|
154     case
155       {i < ~aurFase[2]} {55.0.yield} //permanece en limite inferior hasta llegar a ~aurFase2 valor 227 donde culmina cresta 3
156
157       {(i >= ~aurFase[2]).and( i < ~numEvTot)}
158       {~segLo.next.yield}
159
160       {(i >= ~numEvTot).and(i <= (~numEvTot + ~mes)) }
161       {100.0.yield}
162     }
163   };
164
165
166 ~lo = ~lo.all;
167
168 // Meseta
169 ~hiMes = 107.0 ! ~mes;
170
171 ~hi = ~hi ++ ~hiMes;
172
173 // Modulación de argumento de paso
174 ~stepMod = Routine{
175   (~numEvTot + ~mes).do{ |i|
176     case
177       {i < ~aurFase[0]} {0.5.yield}
178
179       {(i >= ~aurFase[0]).and(i < ~aurFase[1])} {rrand(1.0,3.0).yield}
180
181       {(i >= ~aurFase[1]).and(i < ~aurFase[2])} {rrand(3.0, 8.0).yield}
182
183       {i >= ~aurFase[2]} {rrand(3.0, 13.0).yield}
184     }
185   };
186
187 ~stepMod = ~stepMod.all;
188
189 ~vlnIBrwnI = Pbrown(Pseq(~lo, inf), Pseq(~hi, inf), Pseq(~stepMod,inf), inf).asStream.round(0.5);
190

```

Figure 3.13 Barreras de registro y modulación de tamaño de desplazamiento de movimiento browniano.

También se agrega otra capa al comportamiento de alturas al incluir una máquina de estados ligado a un diccionario (de manera análoga a las estructuras de las obras IV y VI) con posibles desplazamientos distintos que puede o no llegar a afectar el comportamiento típico que está estableciendo el movimiento en determinado momento, dependiendo de la sección de la obra en la que se encuentra. Por ejemplo,

que en la sección I donde el comportamiento típico establece desplazamientos de cuartos de tono, de pronto aparezca uno de 10 semitonos, es decir, como detonante de comportamientos atípicos o marcados.

Esto se controla por medio de una modulación de pesos de probabilidad, aplicando *tendency masks* a las probabilidades, se crea una rutina dentro de la cual dependiendo de la sección en la que se encuentre, irá incrementando o disminuyendo la probabilidad de que ocurra o no el evento, entonces conforme avance una sección será más probable que se vayan detonando comportamientos distintos, para después ir restableciéndose al originalmente planteado, creando “olas” de gradientes que van y vienen.

Finalmente para las alturas, existe otra estructura que está estrechamente vinculada, lo cual se calculó fuera del código que genera la partitura y que se fue agregando ‘manualmente’, se trata de gradientes de presión de la altura pisada sobre la cuerda, siendo una serie de transformaciones, igualmente a manera de “olas”, de ahogar cuerda con poca resonancia, presión de armónico y presión ordinaria.

```

69 // Relajación entre cresta 3 y 4
70
71 {
72 ~numero = 30;
73
74 ~crece = Pseq(Pseries.fromEndpoints(0.35, 0.8, ~numero).asStream.all).asStream;
75 ~decrece0 = Pseq(Pseries.fromEndpoints(0.1, 0.05, ~numero).asStream.all).asStream;
76 ~decrece1 = Pseq(Pseries.fromEndpoints(0.55, 0.15, ~numero).asStream.all).asStream;
77
78 // ~crece.next
79
80 ~val = Routine { arg x, y, z;
81   ~numero.do{
82     x = ~crece.next;
83     y = ~decrece0.next;
84     z = ~decrece1.next;
85     Pwrand([0, 1, 2], [y, z, x]).normalizeSum().asStream.next.yield;
86   }
87 };
88 };
89
90 3.do{~val.nextN(10).postln};
91
92 )

```

Figure 3.14 Gradiente de cambios de presión en cuerda.

Por ejemplo en el código proporcionado en la figura 3.13. se puede apreciar como se están transformando los pesos de las probabilidades para escoger uno de los valores del arreglo [0, 1, 2] que representan presión ahogada, de armónico y ordinario respectivamente. Así las probabilidades de que aparezca 0 disminuyen de 0.1 a 0.05, las probabilidades de que se escoja 1 disminuyen de 0.55 a 0.15, y finalmente la probabilidad de que se escoja el 2 va a incrementar de 0.35 a 0.8. Todas estas modulaciones de probabilidad son calculadas paralelamente, y en este caso, generando 30 valores, que a su vez serían mapeados a las alturas que están sujetas a una grilla de 0.125. Es importante aclarar que esto no quiere decir que todos los valores corresponderán a una fusa, por ejemplo si se da el caso que se repita una altura 3 veces, corresponderá a  $0.125 + 0.125 + 0.125 = 0.375$ , una semicorchea con puntillo.

Por lo tanto, recapitulando el ejemplo: en un inicio había mayor probabilidad de que hubiese presión de armónico (representado con 1) y menor probabilidad de que hubiese una presión ordinaria sobre la altura pisada (representado con 2), lo cual se

va invirtiendo mientras que la presión ahogada de nula o poca resonancia va desapareciendo (representado con 0), buscando con todo esto obtener un comportamiento tímbrico complejo que está siendo sujeto a una transformación íntegra deseada, y que consiguientemente arrojaría el siguiente resultado:

[ 1, 1, 1, 2, 2, 0, 2, 2, 1, 2 ] [ 2, 1, 2, 1, 2, 1, 2, 2, 2, 2 ] [ 0, 2, 1, 2, 1, 2, 2, 2, 2, 1 ]

Además de las estructuras para los desplazamientos de alturas hasta aquí expuestos, existen otros procesos para las duraciones y las articulaciones que ya en su totalidad todas estas construcciones las concibo de como una analogía de capas de acetatos (pero *dinámicos* en el sentido de que el código del cual surgen es dinámico) para conformar el desenvolvimiento sonoro resultante, ya desde aquí se puede apreciar una mayor preocupación de transformaciones tímbricas en comparación con las primeras dos obras.

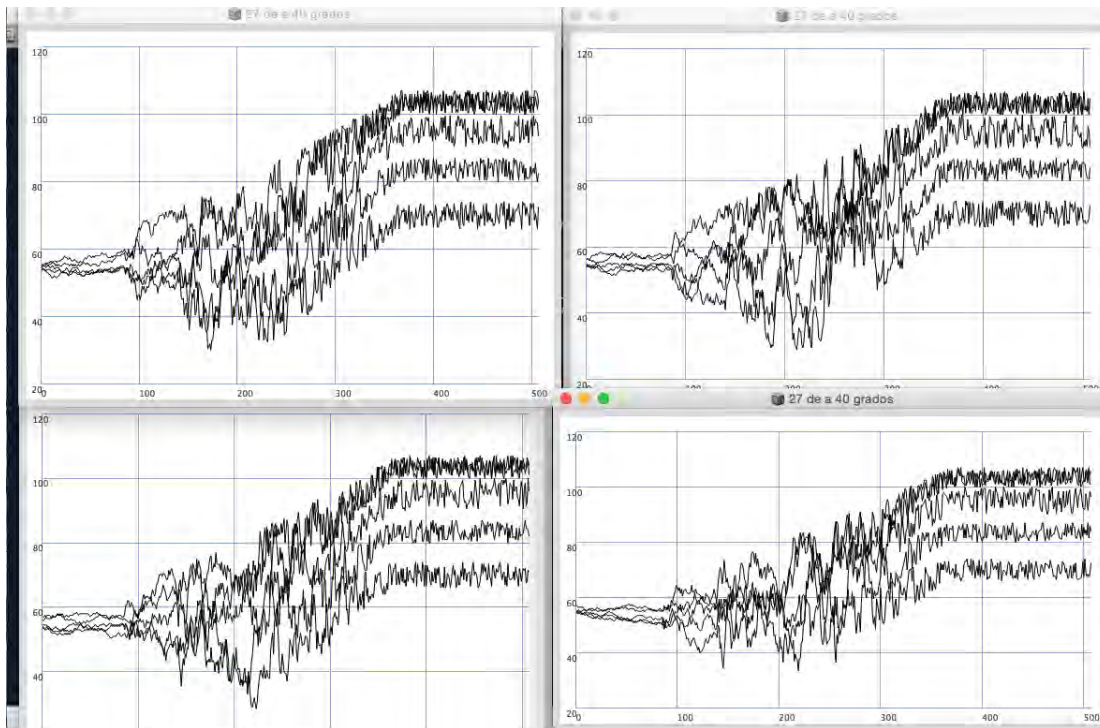


Figura 3.15 Cuatro posibilidades del quinteto de cuerdas, sección de movimiento browniano, eje 'x' tiempo, eje 'y' altura.

Los valores de duración están acomodados en una 'grilla' o *cuantización* de valores de 0.125, o sea de una fusa. Las probabilidades de que se escoja determinada duración están siendo sujetas a una transformación continua yendo hacia figuras de duración cada vez más cortas, desde 2.0 hasta 0.125, cambiando de acuerdo a la sección en la que se encuentre.

Entonces se puede apreciar en el siguiente extracto de código como antes de llegar a la sección I las probabilidades establecen una tendencia a escoger valores cortos, mientras entre la sección II hay mayor probabilidad de escoger un valor intermedio, y finalmente, después de llegar a la sección III se tendrá una mayor probabilidad de escoger la duración mínima 0.125.



```

255 ~contArreglo = Pseq(~contador.nextN(~numEvTot + ~mes),1).asStream.all;
256
257 ~conActualNumber = 0;
258
259 ~durVal = Routine{
260   (~numEvTot + ~mes).do{|i,y|
261     if((~psn[i] == 0).or(~contActual = ~contArreglo[i]) == "no")
262
263       // Si no hay evento las duraciones pueden ser
264       // Varía de acuerdo a la fase en la que se encuentra
265       {
266         case
267         {~conActualNumber < ~aurFase[0]}
268
269         {
270           Pwrand((0.125, 0.25 .. 2.0),
271             Pseries.fromEndpoints(0.1,1.0,16).asStream.all.normalizeSum,
272             inf).asStream.next.yield
273         }
274
275         {(~conActualNumber >= ~aurFase[0]).and(~conActualNumber < ~aurFase[1])}
276
277         {
278           Pwrand((0.125, 0.25 .. 2.0),
279             {
280               Pseries(0.1, 1.0, 8).asStream.all ++
281               Pseries(1.0, 0.1, 8).asStream.all
282             }.normalizeSum,
283             inf).asStream.next.yield
284         }
285
286         {(~conActualNumber >= ~aurFase[1]).and(~conActualNumber < ~aurFase[2])}
287
288         {
289           Pwrand((0.125, 0.25 .. 2.0),
290             Pexprand(0.001, 1.0, 16).asStream.all.reverse.normalizeSum,
291             inf).asStream.next.yield
292         }
293
294         {~conActualNumber >= ~aurFase[2]}
295
296         {
297           Pwrand((0.125, 0.25 .. 2.0),
298             Pexprand(0.001, 1.0, 16).asStream.all.reverse.normalizeSum
299             ).asStream.next.yield
300         }
301       }
302
303       // Si hay evento las duracion será de 0.125
304       {
305         ~conActualNumber = ~contActual;
306         0.125.yield}
307     }
308   }
309 };

```

Figura 3.16 Estructura de gradientes para duraciones.

Para las articulaciones, como se mencionó anteriormente, se utilizan una serie de sistemas Lindenmayer estocásticas que a su vez están ligadas, al igual que con algunos elementos de las alturas, a gradientes o *tendency masks* de probabilidades en las que se establece que haya mayor o menor probabilidad de que ocurra alguna articulación derivada de una de las gramáticas de sistemas-L, dependiendo de la sección en la que se encuentre en determinado momento. Cabe mencionar que estas

articulaciones derivaron de una serie de improvisaciones y exploraciones que realicé en el violín y que posteriormente fueron concebidos dentro de la estructura del código, igualmente se formalizaron los desplazamientos de glissandi que estaban presentes en las improvisaciones con los movimientos brownianos (evidentemente influenciado por Xenakis) a partir de los cuales se planteó el discurso hasta aquí expuesto.

Un sistema-L involucra tres componentes principales:

un *alfabeto*, que constará del conjunto de los caracteres que podrán ser incluidos, un *axioma* que describe el estado inicial del sistema, y las *reglas de generación* que son aplicadas al axioma de manera recursiva, las cuales generan nuevas instancias una y otra vez. Un sistema L incluye un antecesor y un predecesor, como una regla de correspondencia, por ejemplo si se tiene la regla de que  $A \rightarrow C$ , cada vez que se encuentre A generará una C.

A continuación incluyo un ejemplo sencillo que se encuentra dentro del libro *Nature of Programming* (Shifmann, 2012)

Alfabeto: A B

Axioma: A

Reglas:  $(A \rightarrow AB)$   $(B \rightarrow A)$

Generando el ejemplo de la figura 17.

Dentro de un sistema-L estocástico aplica lo mismo que en el ejemplo precedente, pero con la inclusión de probabilidades dentro de sus reglas de generación, por ejemplo  $A \rightarrow (B \text{ ó } C)$ , entonces A podrá ser substituido por B o C teniendo en este caso cada una un 50% de probabilidad.

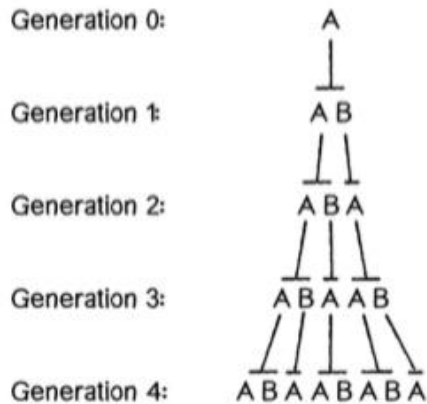


Figura 3.17 Cuatro generaciones dentro del sistema-L. (Shifmann, 2012: 383)

Existen cuatro sistemas-L (todos estocásticos) en la obra, el primero se utiliza hasta la sección I, el segundo hasta la sección II, el tercero en la sección III y el último se utiliza hasta llegar a una “posición compacta” que cubriré más adelante y cuyo código fue escrito fuera del programa principal de la pieza.

Entonces, por ejemplo, en el primer sistema-L estocástico las reglas de generación están ligadas a un diccionario que contiene un acento moderado, acento ordinario, sforzato, un cambio de posición brusco de arco y un jeté, con una particular tendencia de escoger una articulación según en la que se encuentre. Los siguientes sistemas de Lindenmayer tienen estructuras similares y van agregando nuevas articulaciones refrescando el material, a la vez que se establecen distintas

probabilidades, siguiendo la misma idea general de “olas” de mayor a menor probabilidad de que haya o no un evento o articulación.

Ya con lo expuesto hasta aquí he cubierto *grosso modo* los detalles de implementación que se encuentran dentro del código fuente principal tanto de *27 de a 40 grados* como de *Anamnesis*, ya que ambas tienen estructuras esencialmente idénticas en esta sección que cubre gran parte de las obras, con la notable diferencia de escala de uno a cinco instrumentos.

Un arreglo de valores del sistema-L 1 podría arrojar el siguiente resultado:

[ [sfz], [>], [x^ j], [x^ j], [>], [x^ j], [sfz], [x^ j], [>], [>], [-], [sfz], [x^ c], [>], [x^ c], [x^ j], [-], [x^ c], [>], [x^ c], [-], [x^ c], [sfz], [>], [x^ j], [-], [x^ c], [x^ c], [x^ j], [-], [>], [x^ c], [-], [sfz], [>], [x^ j], [>], [x^ c], [x^ c], [sfz] ]

Siendo ‘-’ tenuto, ‘sfz’ sforzato, ‘>’ una acentuación, ‘x^ j’ articulación jeté, y ‘x^ c’ un cambio repentino de posición de arco, representadas de esta manera debido a la sintaxis del software FOMUS con el cual se comunicará SuperCollider para generar la partitura. Cabe destacar que el resultado arrojado por el programa está lejos de ser una versión final, en contraste con lo que sucedía en *IV* y *VI*, ya que las “capas de acetatos”, a diferencia del trabajo parecido al *collage* en las primeras dos obras, lo considero como una etapa o capa del trabajo, son los andamios sobre los cuales se puede trabajar de diversas maneras, en la figura 19 se puede apreciar como es el resultado del programa, y lo que sería el mismo fragmento ya en la versión final de *Anamnesis*.



Figura 3.18 Ramificaciones estocásticas que pueden ser representadas con L-Systems estocásticos (Shifmann, 2012)

Se puede notar que el resultado que se da al ejecutar al código es el material relativamente “crudo” sobre el cual se podrá trabajar y agregarán otros procesos por encima, al igual se toma la libertad necesaria como para modificar algunas secciones alterando el resultado original en aras de un discurso sonoro deseado a mayor detalle.

The image displays a musical score for a violin part, consisting of five staves. The first staff is labeled 'Violin' and the subsequent four are labeled 'Vln.'. The music is written in a 4/4 time signature. The first staff contains a series of eighth and sixteenth notes with various accidentals. The second staff starts at measure 4 and includes a 'c' dynamic marking. The third staff starts at measure 7 and includes a 'j' dynamic marking. The fourth staff starts at measure 10. The fifth staff starts at measure 13 and includes 'j' and 'sfz' dynamic markings. The score is presented as raw material generated by a program.

Figura 3.19 Material “crudo” arrojado por el programa, para la obra Anamnesis.

Hay otras secciones que no están incluidas en el corpus general del código pero que siguen básicamente el mismo concepto de gradientes, principalmente, por ejemplo en *27 de a 40 grados* a partir de aproximadamente el compás 87 o en *Anamnesis* en la tercera página comienza una transformación de alturas definidas hacia notas aleatorias hacia lo más agudo posible culminando en una posición compacta. Esto consiste en ahogar las cuerdas del instrumento cerca del puente donde estarán en un espacio reducido la mano derecha, arco y el puente, teniendo como resultado alturas extremadamente agudas y un timbre particular deseado, se empujó el continuo desarrollo de los movimientos brownianos hasta el límite del instrumento.

```

315 ~sisEle1 = LSys(
316     // axioma
317     "a()",
318     // reglas
319     [
320         "a()" ->
321         {
322             [
323                 "a()", "A()", "b()", "A()c()", "b()A()", "j()"
324             ].wchoose([1,3,3,2,1,3].normalizeSum)
325         },
326         "A()" ->
327         {
328             [
329                 "a()", "A()", "c()b()", "j()", "c()j()"
330             ].wchoose([1,1,2,2,3].normalizeSum)
331         },
332         "b()" ->
333         {
334             [
335                 "c()", "A()", "j()"
336             ].choose
337         },
338         "c()" ->
339         {
340             [
341                 "a()", "A()", "b()", "a()b()", "j()"
342             ].choose
343         },
344         "j()" ->
345         {
346             [
347                 "a()", "A()", "b()", "c()", "j()"
348             ].choose
349         }
350     ]
351 );
352 ~arregloSl1 = [];
353 ~sisEle1_Dict = Dictionary[
354     \a -> "[-]", // tenuto, cr a acento moderado, modificar notación
355     \A -> "[>]", // acento ordinario
356     \b -> "[sfz]", // mayor acentuación
357     \c -> "[x^ c]", // cambio de posición de arco, modificar manualmente
358     \j -> "[x^ j]", // jeté sin acentuar
359 ];

```

Figura 3.20 Sistema-L 1 con pesos de probabilidades.

A partir de este punto ambas obras toman una dirección un tanto distinta, gran parte de las secciones subsiguientes fueron compuestas de manera más libre, sin embargo algunos elementos del discurso siguen siendo creados auxiliado de fragmentos de código. Como por ejemplo en la obra de *Anamnesis* se sigue utilizando un sistema-L para la generación de distintas articulaciones pero ya en la posición compacta, también sobre esto se genera un gradiente para la presión de arco que en

esta sección está representada por pasos discretos entre presión mínima, presión ordinaria y máxima presión de arco.

De manera similar en *27 de a 40 grados* siguen existiendo algunos elementos del discurso que fueron creados auxiliados de fragmentos de código. Un ejemplo podría ser alrededor del compás 108, donde se implementa una estructura semejante a la recién expuesta, o también alrededor del compás 120 se maneja una distribución de poisson<sup>52</sup> para calcular cambios de presión de arco discretos, contrastando con las secciones anteriores donde los cambios eran continuos, así como también acentuaciones esporádicas en cada uno de los instrumentos.

En la figura 3.19. se puede apreciar un tratamiento de gradiente o modulación de probabilidades (*tendency mask*) casi idéntica a la que utilicé para controlar las presiones de las alturas pisadas en el instrumento, con *variables* y *mapeo* distintos, aplicado para establecer un discurso de cambio de textura, generado por cambio de presiones de arco, dentro del cual 0 representa una presión de arco mínima, 1 presión ordinaria y 2 máxima presión de arco u *overpressure*.

---

<sup>52</sup> En entrevista con Varga, Xenakis explica la distribución de Poisson de la siguiente manera anecdótica: "There was a German general at the beginning of the century called Bortkewitch who had the idea of examining whether Poisson's law was true of a phenomenon independent of mathematics. He examined how many soldiers in peace-time were dying of a kick from a horse. Horses are nice animals, as you know, and soldiers are nice too, in peace, but horses kick from time to time. And people die of it. Things like that occur rarely, of course, which corresponds to the conditions of Poisson's law of probabilities, which applies to 'rare events' without causality. The law is true of music as well: that recognition was my contribution." (Varga, 1996: 79)





En este particular fragmento de código se está partiendo de un pasaje con presión ordinaria de arco, modulando la textura hacia uno de *flautato*, a la vez que se va introduciendo la presión máxima de arco, pero con menor probabilidad de ser escogida con respecto al *flautato*.

```

144 // Gradiente disipación resolución 0.25
145 // 0 -> Flautato
146 // 1 -> Ordinario
147 // 2 -> Over Pressure
148
149 {
150   ~numero = 21*4;
151
152   ~crece = Pseq(Pseries.fromEndpoints(0, 0.8, ~numero).asStream.all).asStream;
153   ~disminuye0 = Pseq(Pseries.fromEndpoints(1.0, 0.05, ~numero).asStream.all).asStream;
154   ~disminuye1 = Pseq(Pseries.fromEndpoints(0, 0.15, ~numero).asStream.all).asStream;
155
156   // ~crece.next
157
158   ~val = Routine { arg x, y, z;
159     ~numero.do{
160       x = ~crece.next; // crece flautato
161       y = ~disminuye0.next; // disminuye overpressure
162       z = ~disminuye1.next; // ordinario crece poco
163       Pwrand([0, 2, 1], [x, y, z].normalizeSum).asStream.next.yield;
164     }
165   };
166 };
167
168 21.do{~val.nextN(4).postln};
169
170 )
171

```

Figura 3.22 Gradientes de presión de arco.

	Cresta 1	Cresta 2	Cresta 3	Cresta 4 + Mes
Vln I	22 - 23	30 - 31	49 - 50	67
Vln II	25	33-34	56	71
Vla	20 - 21	30 - 32	52 - 53	69
Vc	22	35	55	72
Cb	22	36	54	70

Figura 3.23 Compases de mayor actividad.

### 3.2.2 INTERVENCIÓN DE ANAMNESIS

A continuación un breve espacio para abordar los detalles de implementación en la intervención de *Anamnesis*.

El código que escribí para controlar los parámetros de síntesis que se efectúa sobre los diversos *buffers* provenientes de la parte acústica de la obra, en esta caso del violín, consta de las siguientes partes y características principales.

Los buffers escogidos son cargados a la memoria, (se escogen fragmentos libremente de distintas secciones de la obra) para luego ser analizados por la clase *SCMIRAudioFile*<sup>53</sup> (que representa SuperCollider Music Information Retrieval ) escrito por Nick Collins e implementada como una librería en SC, en el cual es posible analizar aspectos acústicos y psicoacústicos del audio cargado.

Por ejemplo se pueden extraer datos partiendo de los UGens de *Machine Listening*<sup>54</sup> implementados en SC como el de MFCC Mel Frequency Cepstral Coefficients (Coeficientes Cepstrales en las Frecuencias de Mel) que son coeficientes para la representación del habla basados en la percepción auditiva humana, Chromagram que permite el análisis de alturas partiendo de determinada división de la octava, Onsets que permite la detección de ataques de un sonido, Loudness análisis de intensidad de sonido a partir de las unidades psicoacústicas de sones, Tartini un *pitch follower* que es un detector de alturas, entre otros.

---

<sup>53</sup> Descargable de manera gratuita en: <http://www.cogs.susx.ac.uk/users/nc81/code.html> .

<sup>54</sup> Para una discusión detallada sobre los UGens el lector se puede referir al capítulo 15 de *The SuperCollider Book* (2011: 439)

Después de cargar los buffers y ser analizados e instanciados los datos dentro de una variable, se agrupan los datos dentro de tres arreglos: uno que será utilizado para almacenar la información del análisis que arrojó los valores de intensidad más altos, otro para los valores de intensidad mediana y finalmente otro donde son guardados los valores mínimos, siendo estos tres arreglos para cada uno de los archivos escogidos. Posteriormente los datos serán mapeados a posiciones dentro de los archivos de audio que presentan las características del análisis, por decir, dentro de *frames 'x'* a *frames 'y'* los valores de sones corresponden a las secciones del archivo donde se cumplen los requisitos de evaluación (máxima intensidad, mínima intensidad e intensidad baja), por lo que se estará moviendo entre distintos segmentos del archivo.

Antes de abordar la manera en que se manejaron los datos del análisis efectuado sobre los archivos de audio, me gustaría abordar brevemente la técnica de síntesis implementada a partir del cual se diseñó un sintetizador, basado en las técnicas de programación y diseño de sintetizadores expuesto por Alberto de Campo en un capítulo dedicado al microsonido en SuperCollider. (Wilson; Cottle y Collins; 2011: 463-504) A continuación detallaré los algoritmos empleados para controlar las automatizaciones que modularán los parámetros de síntesis.

El Sintetizador digital implementado en SuperCollider es una instancia sencilla del *Glisson Synthesis*<sup>55</sup>, una clase de síntesis granular en el cual cada gránulo es

---

<sup>55</sup> Una técnica de síntesis granular que implicaba pequeños *glissandi* de los gránulos, expuesto por Xenakis (1992: 54-55) y que posteriormente sería implementado por Jones y Parks en 1988, luego por Curtis Roads en 1998.

abordado como un vector formado por frecuencia, tiempo y amplitud. La característica sobresaliente de esta técnica es la de provocar pequeñas trayectorias de glissandi independientes en cada uno de los gránulos que conforman una nube de partículas de sonido.

Esta característica del *Glisson Synthesis* permite “patrones de magnetización” de las nubes de partículas granulares, que son una combinación de varios parámetros que determinan la dirección de los desplazamientos de los glissandi dentro de una nube. En primera instancia, los glissandi pueden tener movimientos amplios, (consistiendo de un amplio rango de frecuencia) o puede consistir de desplazamientos dentro de un registro corto. En segundo lugar, pueden ser clasificados de acuerdo a la dirección del movimiento del desplazamiento del vector, pueden ser unidireccionales, todos los gránulos ascendiendo o descendiendo; o bidireccionales, aleatoriamente hacia arriba o hacia abajo. En tercer lugar, pueden ser divergentes, parten de un centro de frecuencia común; o convergentes partiendo de frecuencias divergentes que convergen en un mismo centro de frecuencia. El centro de frecuencia puede cambiar a través del tiempo. (Roads, 2001: 122)

En la siguiente imagen se puede apreciar que el sintetizador implementado en SuperCollider:

```

34
35
36 (
37 SynthDef(\glissonCloud2, { |out=0, freq1=440, freq2=660, sustain=0.05, level=0.1, amp=0.5, pan=30, envwide=0.5, filepos, buf|
38
39   var sloptime = (1 - envwide) * 0.5;
40
41   var sound = PlayBuf.ar(1, buf, XLine.ar(freq1, freq2, sustain), 1, filepos);
42
43   // Envolventes de gránulos-----
44
45   var env = EnvGen.ar(
46     Env([0, level, level, 0], [sloptime, envwide, sloptime], \sin),
47     timeScale: sustain,
48     doneAction: 2) * AmpComp.ir(freq1/freq2); // Con compensación psicoacústica de amplitud ligado a freq
49
50
51   OffsetOut.ar(out, PanAz.ar(2, (sound * env)*amp, pan)); // Cuadra
52
53   }, \ir !8
54 ).add;
55

```

Figura 3.24 Implementación de sintetizador de Glisson Synthesis

Como se puede apreciar, el sintetizador realmente es sencillo, consiste básicamente de:

- un UGen que reproducirá el Buffer (de acuerdo a los procesos del sistema que especifiquen donde debe de encontrarse el *puntero* );
- una envolvente sencilla con una ventana de forma senoide;
- un desplazamiento de la frecuencia de cada gránulo donde se llevará a cabo el glissando, de freq1 a freq2 por un determinado tiempo *sustain*;
- finalmente un paneador PanAz, en la imagen se puede apreciar que esta en formato estéreo, pero cambiando la variable se puede presentar a un mayor número de canales.

Regresando al funcionamiento del mecanismo que controla los argumentos del sintetizador recién expuesto, para desplazarse dentro de los límites de los *frames* de los archivos de audio se crearon distintos modos de transición representadas en la siguiente figura.

Por lo tanto la media de la distribución gaussiana empleada está representada por un valor derivado de los arreglos que se poblaron con los datos del análisis,

```
// =====
// ----- Transiciones //////////////////////////////////////
// =====

~trans = Dictionary[

  \gaussMin  -> Pgauss(~arIndMin[~arIndMin.size.rand], 30, inf).abs,
  \gaussMed  -> Pgauss(~arIndMed[~arIndMed.size.rand], 30, inf).abs,
  \gaussMax  -> Pgauss(~arIndMax[~arIndMax.size.rand], 30, inf).abs,
  \envIndValMin -> Prout({ loop { ~arIndMin[~arIndMin.size.rand].yield }}),
  \envIndValMed -> Prout({ loop { ~arIndMed[~arIndMed.size.rand].yield }}),
  \envIndValMax -> Prout({ loop { ~arIndMax[~arIndMax.size.rand].yield }})

];

~transFSM = P fsm([
  #[0],

  \gaussMin, #[0, 0, 4, 5],
  \gaussMed, #[1, 1, 2, 4, 5],
  \gaussMax, #[2, 2, 0, 3, 4],
  \envIndValMin, #[3, 3, 2, 5],
  \envIndValMed, #[4, 4, 0, 2],
  \envIndValMax, #[5, 5, 0, 2, 3],

  nil, nil

], inf).asStream;
```

Figura 3.25 Diccionario de transiciones ligadas a una máquina de estados.

esto a su vez siendo controlado por una máquina de estados que determina las rutas a seguir. Por ejemplo, si se encuentra en `\gaussMin` que corresponde a una distribución gaussiana<sup>56</sup> que se mueve alrededor de los valores de los frames (con una desviación arbitraria de 30 frames) donde se encuentren amplitudes menores, tiene un 50% de probabilidad de permanecer ahí, 25% de probabilidad de avanzar a `\envIndValMed` que esta ligada a una función donde se escogerán aleatoriamente

<sup>56</sup> Una distribución normal o gaussiana se refiere a una distribución de valores que se agrupan alrededor de una media, también llamada distribución Laplace. (Ambos matemáticos, Carl Friedrich Gauss y Pierre-Simon Laplace estaban definiendo la distribución en el siglo XIX). (Shifmann, 2012: 11) En la figura 22 se pueden apreciar tres distribuciones gaussianas cuya media está siendo modulada por movimientos brownianos (que se verá más adelante), programado en SuperCollider y Processing.

frames dentro de los valores de intensidad mediana, y finalmente un 25% de probabilidad de avanzar a `\envIndValMax` que representa lo mismo pero para los valores de intensidad máxima.

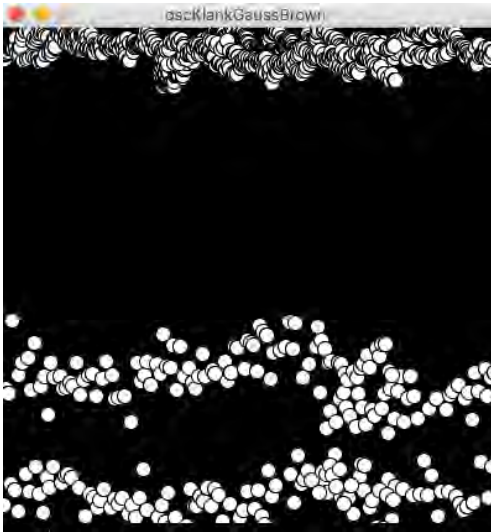


Figura 3.26 Distribución uniforme con media modulada.

Las interpolaciones entre los valores recién expuestos dentro de un arreglo determinado depende de una colección de formas de transición definido anteriormente en el código. Consiste de siete opciones distintas de transición dentro de las cuales se puede escoger entre: extremadamente corta, corta, media, relativamente lenta, lenta, muy lenta y extremadamente lenta. Estas son tomadas como tiempos dentro de una envolvente que finalmente será leída por una Pseq para alimentar el sintetizador con los valores correspondientes para su argumento de posición de puntero en el archivo.

Para el argumento de *sustain* dentro del sintetizador implementado dentro de la envolvente, se tienen cinco posibilidades distintas que corresponderán a la manera en que se va a modular el tamaño del gránulo, hinchándose y contrayéndose cada



gránulo según una función senoide, un patrón PClipNoise que brincará del valor mínimo al valor máximo sin interpolación, otra opción sería una distribución exponencial favoreciendo los valores cortos, un patrón PLFPulse que imita una forma de onda cuadrada, en el cual son modulados sus parámetros, y finalmente una PLFSaw que imita una diente de sierra.

```

//-----
// Rutina para tener tiempos dinámicos

// Transiciones dentro de siete rangos distintos
// extremadamente corta 1 - 30
// corta                30 - 80
// media                80 - 150
// algo lenta           150 - 250
// lenta                250 - 450
// muy lenta            450 - 850
// lo más lento         850 - 1500

~transInter = Dictionary[
  \exCorta  -> Pexprand(0.001, 30, inf),
  \corta    -> Pexprand(30, 80, inf),
  \media    -> Prand((80..150), inf),
  \media2   -> Prand((150..250), inf),
  \lenta    -> Prand((250..450), inf),
  \muylenta -> Phprand(450, 850, inf),
  \x lenta  -> Phprand(850, 1500, inf)
];

// Especificar probabilidades
// valores actuales arbitrarios
~tempDin = P fsm([
  #[0],
  \exCorta,  #[1, 6],
  \corta,    #[1, 0, 2, 3],
  \media,    #[0, 2, 3],
  \media2,   #[5, 1, 6],
  \lenta,    #[0, 6],
  \muylenta, #[0, 1, 2, 3, 4, 5, 6],
  \xlenta,   #[0, 5],
  nil, nil
], inf).asStream;

```

Figura 3.27 Distintas posibilidades de transiciones.

Todas estas opciones son reunidas en una base de datos dentro de un diccionario, mapeado nuevamente a una máquina de estados que dictará las probabilidades de transición de estados.

Posteriormente programé la sección que controla las posibilidades de desplazamiento de frecuencias y de magnetización, pudiendo la frecuencia del cual

partirá el glissando permanecer estático, cambiar de acuerdo a una distribución uniforme o por desplazamientos brownianos cortos, medianos y amplios.

```

~sustain2 = Dictionary[
  \sinOsc -> Prout({ var sin = PSinOsc(1000, 0, 1.99, 2, inf).asStream;
                    loop{ sin.next.yield }}).asStream,
  \clip   -> Prout({ var clip = PClipNoise(0.99, 1.0, inf).asStream;
                    loop { clip.next.yield }}).asStream,
  \exp    -> Prout({ var exp = Pexprand(0.001, 0.5, inf).asStream;
                    loop{ exp.next.yield }}).asStream,
  \lfp    -> Prout({ var pulse;
                    pulse = PLFPulse(50, 0,
                                     Pstutter(20,|
                                     Pseq([[0.1, 0.2 .. 1.0), (1.0, 0.9 .. 0.1)].flat,
                                     inf)), 1.0, 0.1,
                                     inf).asStream;
                    loop{ pulse.next.yield}}).asStream,
  \lfs    -> Prout({ var saw = PLFSaw(40, 0, 0.49, 0.5, inf).asStream;
                    loop { saw.next.yield}}).asStream
];

~susIndepeFSM2 =
P fsm([
  #[0],
  \sinOsc, #[0, 0, 1, 2, 3],
  \clip,  #[1, 2, 3],
  \exp,   #[4, 3, 2],
  \lfp,   #[0, 4],
  \lfs,   #[3, 4],
  nil, nil
]).asStream;

~susRout2 = Prout({ var stutter = Pstutter( 300, Prout({ loop{ ~susIndepeFSM2.next.yield}})).asStream;
                  loop { ~sustain2[stutter.next].next.yield}}).asStream;

```

Figura 3.28 Opciones de modulación de tamaño de gránulo.

Después procedí a establecer las opciones de trayectorias de paneo, teniendo hasta 14 opciones distintas, dispuestas de manera similar a la de los otros argumentos, como un diccionario ligado a una máquina de estados. La base de datos del diccionario de posibles desplazamientos de paneo, dentro de cualquier número de canales (tomando en cuenta que se modificaría el argumento de número de canales)

consiste básicamente en desplazamientos controlados por movimientos brownianos con pasos cortos, medianos y amplios, ruido blanco consistiendo de una distribución uniforme, desplazamientos unidireccionales de distintas velocidades que se asemejan al movimiento de un aspersor, movimientos de vaivén similares a los de una onda sinusoidal, saltos sin interpolación que se concentran en los valores extremos, entre otros, algunas de estas opciones se muestran en la figura 28 y se puede referir a todas las opciones en el código fuente disponible en el apéndice.

```
// =====
// Divergente
// Movimiento divergente , con frecuencia eje móvil
// ligado a freq2
// Función para modular freq2
// Función para modular paso de movimiento browniano de freq1

~freqModDict = Dictionary[
    \estatico      -> Pstutter(300, Prand((0.1,0.11..2.0),inf),inf).asStream,
    \rand          -> Prand((0.1,0.11..3.0),inf).asStream,
    \brownianoCorto -> Prout({ var brwnC = Pbrown(0.1, 3.0, 0.05,inf).asStream;
                          loop{ brwnC.next.yield}}).asStream,
    \brownianoMediano -> Prout({ var brwnM = Pbrown(0.1, 3.0, 0.125, inf).asStream;
                          loop{ brwnM.next.yield}}).asStream,
    \brownianoAmplio -> Prout({ var brwnA = Pbrown(0.1, 3.0,0.25,inf).asStream;
                          loop{ brwnA.next.yield}}).asStream,
];

~freqFSM = P fsm([
    #[0],
    \estatico,      #[0, 1, 2],
    \rand,          #[1, 2, 3, 4],
    \brownianoCorto, #[0, 3],
    \brownianoMediano, #[0, 3, 4],
    \brownianoAmplio, #[0, 1],
    nil, nil
    ]
).asStream;

~freqRout = Prout({ var freq = Pstutter( 200, Prout({ loop{ ~freqFSM.next.yield}})).asStream;
                  loop { ~freqModDict[freq.next].next.yield}}).asStream;
```

Figura 3.29 Posibilidades de modular freq1.

Finalmente todo es empaquetado dentro de una estructura de Pdef y Pbind para controlarlo en vivo, empujando el mecanismo hacia zonas deseadas o dejándolo correr libremente.

Hasta este punto he expuesto *grosso modo* detalles del funcionamiento del sistema que estará controlando los parámetros de Glisson Synthesis de los archivos de audio de la ejecución de *Anamnesis*, en el siguiente apartado me gustaría abordar esta familia de obras a partir de un análisis retrospectivo ubicándolo dentro del hecho musical.

```

\brwnR: Prout({
var brownR = Pbrown(-1.0, 1.0, 0.5, inf).asStream;
var rep = Prand((50..100),inf).asStream;
var cuentaMax = rep.next;

(cuentaMax+1).do{ arg x;
  if( x < cuentaMax, {brownR.next.yield;}, { nil.yield; } )}}).asStream,

\white: Prout({
var white = Pwhite(-1.0, 1.0, inf).asStream;
var rep = Prand((50..100),inf).asStream;
var cuentaMax = rep.next;

(cuentaMax+1).do{ arg x;
  if( x < cuentaMax, {white.next.yield;}, { nil.yield; } )}}).asStream,

\seqL1: Prout({ // unidireccional a la Saw
var seqL = Pseq( Array.series(100, -1.0, 0.01) ++ Array.series(101, 0, 0.01), inf).asStream;
var rep = Prand((50..100),inf).asStream;
var cuentaMax = rep.next;

(cuentaMax+1).do{ arg x;
  if( x < cuentaMax, {seqL.next.yield;}, { nil.yield; } )}}).asStream,

\seqM1: Prout({ // unidireccional
  var seqM = Pseq(Pseries.fromEndpoints(-1.0, 1.0, 50).asStream.all, inf).asStream;
var rep = Prand((50..100),inf).asStream;
var cuentaMax = rep.next;

(cuentaMax+1).do{ arg x;
  if( x < cuentaMax, {seqM.next.yield;}, { nil.yield; } )}}).asStream,

\seqR1: Prout({ // unidireccional
  var seqR = Pseq(Pseries.fromEndpoints(-1.0,1.0,15).asStream.all, inf).asStream;
var rep = Prand((50..100),inf).asStream;
var cuentaMax = rep.next;

(cuentaMax+1).do{ arg x;
  if( x < cuentaMax, {seqR.next.yield;}, { nil.yield; } )}}).asStream,

\seqL2: Prout({ // Bidireccional a la SinOsc

```

Figura 3.30 Distintas opciones de desplazamientos de sonido, paneo.

### 3.2.3 ANÁLISIS RETROSPECTIVO DE ANAMNESIS Y 27 DE A 40 GRADOS

Dentro de las estructuras que construí para estas obras, las imaginé siendo superpuestas a manera de acetatos para crear un sistema que describe un continuo serpenteo espasmódico, que a su vez crea ramificaciones con olas de distintas densidades que se van arrastrando ascendentemente, a manera de una continua evaporación visible (audible) con mayor o menor fuerza o violencia.

Lo que expuse en las obras *IV* y *VI* en cuanto a los tres polos de la tripartición es igualmente aplicable para estas dos obras en cuanto a lo que involucraron las automatizaciones, las estructuras modulares y lo demás expuesto por lo que me enfocaré en las diferencias primordiales que percibí.

La diferencia primordial en el polo poiético considero que yace en la decisión de tener mayores consideraciones para el desenvolvimiento tímbrico de las obras, ya que me percaté de la excesiva primacía del pensamiento de altura y duración en detrimento del desarrollo tímbrico. Con esto en mente formulé la gramática de articulaciones, que como mencioné fueron derivadas de improvisaciones que realicé en el violín. Además creé estructuras modulares que utilicé en distintas instancias, en ejecuciones distintas del código, es decir, toda la obra no esta englobada dentro de un mismo código, fue pensada por secciones dentro de las cuales me permití acercarme o alejarme dependiendo de las necesidades expresivas en turno.

Finalmente dentro del polo poiético considero que al estar componiendo dentro de un ambiente de programación que puede fungir para procesamiento de señales y síntesis de sonido y no únicamente como una poderosa herramienta de CAO (no siendo exclusivas entre sí), tomé la decisión de establecer un discurso análogo entre

la parte acústica de *Anamnesis* y su contraparte intervenida, es decir el mismo lenguaje de programación con sus posibilidades me invitaban a ir más allá y a explorar otras posibilidades. Además de utilizarlo como una herramienta de improvisación.

Dentro del polo estésico considero que para *27 de a 40 Grados*<sup>57</sup> sucede lo mismo que en las obras anteriores en el sentido de que queda intacto, pero en la ejecución de *Anamnesis* con su contraparte intervenida creo que si pueden ocurrir cambios.

En *Anamnesis* una ejecución tradicional—análoga—es almacenada, accedida y manipulada en formato digital, a partir de su representación numérica. Aquí se produce un cambio de *formato* y por lo tanto opino que esto posibilita un cambio en la manera en que se pueden plantear estrategias de composición (nivel poiético), y la manera en que se puede experimentar la obra (estésico). Aquí formato se puede entender de diversas maneras, forma parte del polo neutro siendo un formato dentro de una computadora, es decir el cambio de formato de análogo a digital. Puede ser abordada desde el polo poiético, siendo su misma existencia dentro del ordenador catalizador para su manipulación a partir de diversas estrategias creativas. Finalmente, puede también abordarse desde el polo estésico, si se toma en cuenta por otro lado, el formato en el cual será experimentada la obra.

Considérese la experiencia del escucha en un formato de concierto tradicional en el cual tiene al ejecutante al frente, el escucha generalmente está sentado en una

---

<sup>57</sup> Por razones de practicidad se tomó la decisión de no incluir una contraparte intervenida para *27 de a 40 Grados*.

posición fija, y en el cual probablemente se quede respetando las respectivas convenciones sociales en turno, no se pondrá de pie a caminar alrededor del ejecutante, ni (posiblemente) llegará y partirá cuando guste sino que percibirá la obra como un discurso temporal sintagmático.<sup>58</sup>

Por otro lado, en un formato de intervención o de instalación sonora es común que el escucha la explore espacialmente, camine alrededor del recinto, se recueste, arribe en medio del desarrollo un discurso sonoro, siendo este estático o con una configuración estructural lineal, y se aleje cuando guste. Considero que hay diferencias fundamentales en la manera en que se permite explorar las obras en cuanto a su temporalidad.

Ya en el capítulo dos mencioné la noción de paradigma y sintagma, y la manera en que lo adopta Manovich para describir narrativas en objetos de la nueva media:

Narratology, the branch of modern literary theory devoted to the theory of narrative, distinguishes between narration and description. Narration is those parts of the narrative that move the plot forward; description is those parts that do not. Examples of description are passages that describe the landscape, or a city, or a character's apartment. In short, to use the language of the information age, description passages present the reader with descriptive information. As its name itself implies, narratology paid most attention to narration and hardly any to description. But in the information age, narration and description have changed roles. If traditional cultures provided people with well-defined narratives (myths, religion) and little 'standalone' information, today we have too much information and too few narratives that can tie it all together. For better or worse, information access has become a key activity of the computer age. (Manovich, 2001: 216-217)

---

<sup>58</sup> Me gustaría hacer hincapié en que no es mi intención ni pretensión la de saber qué es lo que experimentará el escucha hablo partiendo de experiencia propia.

Es posible que al pasar de experimentar la ejecución de una obra en formato de concierto tradicional, al de una intervención o instalación sonora, cambien modos de experiencia y de formas de escucha de los asistentes, desde luego que percibimos de manera sintagmática, escuchamos en sucesión, sin embargo al presentarse ante propuestas que generalmente presentan discursos más estáticos y contemplativos, y la manera en que se permite que se explore la obra, quizá invite a enfocarse en la “descripción” de las texturas sonoras más que a las correlaciones de forma y desarrollo.

Me parece importante constatar que desde luego que una obra ejecutada de manera tradicional también puede invitar a entrar a este estado mental o manera de percepción, bastaría en pensar en estéticas como las de Morton Feldman, en las que personalmente me puedo recostar en mi asiento y perder la noción del tiempo dentro de obras como *Violin and String Quartet* (1985).

Sin embargo lo que me parece fundamental constatar, es *la posibilidad de la transmutación ontológica de la obra*, una ejecución tradicional—análoga—es almacenada, accedida y manipulada en formato digital, a partir de su representación numérica (esto ya posibilita diversas estrategias poiéticas), en el cual es posible una mutación de expresión artística, involucrando a su vez *potenciales* cambios en los tres niveles que se están tomando en cuenta para la forma simbólica en turno.



### 3.3 AUTOPO(I)ÉTICA

La última obra que compuse para este trabajo, titulada *Autopo(i)ética*<sup>59</sup> para cuarteto de cuerdas involucra la generación de partitura en tiempo real de manera automatizada a partir de toma de decisiones, en la cual se escogerán distintos caminos posibles de “escenas sonoras”. Cada uno de los ejecutantes tiene una laptop en el escenario donde se ejecutará el código iniciando el programa y la parte de cada músico será generada apareciendo en cada una de sus respectivas pantallas, lo cual será leído a primera vista por el cuarteto.

La versión que se presenta por el momento tiene opciones limitadas en cuanto a caminos a escoger para las escenas sonoras, sin embargo se concibe como un proyecto que irá creciendo paulatinamente y al cual se le irán agregando más y más opciones con distintos tipos de discursos, ya sea una pieza de unos siete minutos o una serie de miniaturas distintas, con mayor o menor grado de estructuras deterministas, estocásticas, jerárquicas o rizomáticas.

#### 3.3.1 DETALLES DE IMPLEMENTACIÓN EN AUTOPO(I)ÉTICA

El programa se escribió dentro del sistema operativo Mac OSX 10.6.8 y ha sido probado en distintas versiones de Mac OS hasta el 10.11.1, así como también en la

---

<sup>59</sup> Estrenada en el Festival Internacional Cervantino, 2014 por el Cuarteto de Cuerdas Cromano, con apoyo del CMMAS (Centro Mexicano para la Música y las Artes Sonoras) y CONARTE (Consejo para la Cultura y las Artes de Nuevo León).

distribución Linux de Ubuntu 14.04 LTS<sup>60</sup>. El software que se utilizó en su implementación fue:

- SuperCollider
- FOMUS
- LilyPond
- SKIM
- Evince (dentro de LINUX)

A continuación expondré la estructura general de la primera versión de la pieza la cual tiene un planteamiento marcadamente determinista en cuanto a la forma general y con mayor o menor grado de indeterminismo dentro de las ‘escenas sonoras’ y *de su ejecución*, no obstante, también se presentarán las posibilidades que tiene el código de expansión para generar un mayor número de opciones a desarrollarse en un futuro.

La forma en que fue escrita la pieza es similar al de las obras anteriores, separando distintos parámetros musicales dentro de patrones en SuperCollider que ya en su totalidad conforman un discurso sonoro unificado. Una diferencia fundamental fue concebir las estructuras de las escenas de una manera aún más modular facilitando su interconexión y también la implementación de nuevos algoritmos que generan distintas escenas y diferentes opciones de caminos, como una serie de nodos que

---

<sup>60</sup> Para la implementación dentro de Ubuntu se tuvieron que hacer algunas modificaciones dentro del código de SuperCollider, como también dentro de la clase Fomus2, se puede encontrar más información sobre esto en el Apéndice y sus versiones de descarga correspondientes en la liga proporcionada.

tienen la posibilidad de conectarse con otros nodos específicos, además de la necesidad de expandir la clase *Fomus* a *Fomus2* al requerir mayor funcionalidades.

Para las alturas de la presente versión de la pieza, siempre se comienza con una colección o *Array* de tres notas [60, 60.5, 61]<sup>61</sup> que corresponden a do, do cuarto de tono sostenido y do sostenido respectivamente, dentro del índice acústico 4<sup>62</sup>. Estas alturas estarán ligadas a otra estructura de articulaciones (que será planteado más adelante) con la cual se establecerá que serán siempre al inicio en forma de armónicos artificiales en los cuatro instrumentos, dando las alturas reales dos octavas por encima dentro de este espacio reducido de tres cuartos de tono, un estrecho *cluster* fluctuante de cuartos de tono.

Las notas serán escogidas en órdenes aleatorios, el orden en que se ejecutan será distinta cada vez que se ejecuta. Se irán agregando alturas de tres en tres (cada nota apareciendo a la vez), partiendo de la misma disposición de tres cuartos de tono conforme vayan avanzando las escenas, hasta llegar a un conjunto conglomerado que comprende una gama de cuartos de tono dentro de un espacio de un intervalo 6.

---

<sup>61</sup> Quizá se pueda expresar utilizando notación del *pitch class sets* {0, 0.5, 1} aplicando un módulo 24 en lugar de un módulo 12.

<sup>62</sup> También se elaboró un esquema en el que una computadora escoge una colección de tres alturas, las cuales son comunicadas a las tres computadoras restantes mediante una red WIFI, no se implementó en su estreno debido a problemas de conexión, sin embargo se mantiene como una posibilidad, además, también será muy útil para una futura (pronta) implementación de algoritmos genéticos, y estrenada a unos meses del tiempo en que escribo estas líneas, por el Cuarteto de Cuerdas Cromano a raíz de una residencia artística con los mismos.

Se puede apreciar una tendencia acumulativa similar a las que utilicé dentro de las obras *IV* y *VI*, aunque en contextos muy distintos.

Además de este conjunto de cuartos de tono, se tiene otro sistema que define un operador de suma o resta que afectará la altura, con la finalidad de además de tener la conglomeración de alturas, agregar distintas probabilidades de ataques de alturas fuera de ella, a manera de detonantes que suceden similarmente en *Anamnesis* y en *27 de a 40 Grados*, y el formación de “trinchas” de registro similares a la que se encuentran en las obras *IV* y *VI*.

Para las duraciones se plantea un sistema igualmente afín al de las obras *27 de a 40 grados* y *Anamnesis*, en el que hay una grilla con un valor mínimo de 0.125 y distintos conjuntos con tendencias a escoger valores más cercanos a la unidad mínima, valores centrales o valores de mayor duración, dependiendo del tipo de movimiento rítmico y/o direccionalidad que se quiera establecer en determinada escena o en la interconexión de diversas escenas.

Por otro lado, para las articulaciones se hicieron escenas con probabilidades fijas y otras que presentan gradientes como las que expuse en los apartados anteriores, un ejemplo de las que tienen probabilidades con pesos fijos sería el de la escena inicial.

```

183
184 ~artA = Pseq([Pseq(["artharm-sounding][fp][x^ glissandoSempre"], 2),
185   Pwrand(["artharm-sounding", "artharm-sounding]>", "artharm-sounding][x^ ric...]",
186     "artharm-sounding][fp]"), [1, 0.5, 0.4, 0.3].normalizeSum, ~numeros[0] - 2)],
187   ~numeros[0]).asStream;
188

```

Figura 3.31 Articulaciones iniciales.

Esta escena tiene elementos deterministas como el forte piano con armónico artificial al inicio y la indicación de que habrá glissandi entre todas las alturas, también involucra toma de decisiones con probabilidades con distintos pesos entre un conjunto de articulaciones y dinámicas.

Un ejemplo de la parte de articulaciones de una escena que involucre modificaciones de probabilidades se puede apreciar en la siguiente figura, en donde se están modificando las probabilidades continuamente para favorecer ataques y articulaciones más secos o cortos, generalmente después de este tipo de escenas que involucran transformaciones continuas, les seguirán escenas 'meseta' dentro de las cuales se mantienen fijas las probabilidades a las que se llegaron.

En la figura se puede apreciar una modulación de pesos de probabilidades o una *tendency mask* estableciendo un gradiente de cambio de articulaciones de manera similar a la expuesta en *Anamnesis y 27 de a 40 Grados*.

Entonces para ejemplificar como se puede concebir una escena en su totalidad a partir de las estructuras recién expuestas y como se conectaría con otra presento un breve ejemplo del funcionamiento general del programa.

```

339 ~crece6 = Pseq(Pseries.fromEndpoints(0.0, 1.0, ~numeros[6] - 2).asStream.all).asStream, // staccatissimo
340 ~creceG2 = Pseq(Pseries.fromEndpoints(0.0, 1.0, ~numeros[6] - 2).asStream.all).asStream, // staccatissimo armónico
341 ~creceG3 = Pseq(Pseries.fromEndpoints(0.0, 0.9, ~numeros[6] - 2).asStream.all).asStream, // battuto ricochet
342 ~decreceG = Pseq(Pseries.fromEndpoints(0.6, 0.01, ~numeros[6] - 2).asStream.all).asStream, // ord
343 ~decreceG2 = Pseq(Pseries.fromEndpoints(0.6, 0.01, ~numeros[6] - 2).asStream.all).asStream, // armónico artificial ord
344 ~decreceG3 = Pseq(Pseries.fromEndpoints(0.6, 0.2, ~numeros[6] - 2).asStream.all).asStream, // armónico sfz
345 ~decreceG4 = Pseq(Pseries.fromEndpoints(0.5, 0.1, ~numeros[6] - 2).asStream.all).asStream, // armónico trem sfz
346
347 // ~crece.next
348
349
350 ~gradienteG = Prout({ arg crece, crece2, crece3, decrece, decrece2, decrece3, decrece4;
351
352   (~numeros[6] - 2).do{
353
354     crece = ~creceG.next;
355     crece2 = ~creceG2.next;
356     crece3 = ~creceG3.next;
357     decrece = ~decreceG.next;
358     decrece2 = ~decreceG2.next;
359     decrece3 = ~decreceG3.next;
360     decrece4 = ~decreceG4.next;
361
362     Pwrand(["[]", "[artharm-sounding]!", "[x^ batt...]", "[sfz]", " ", "[x^ ric...]", "[artharm-sounding]", "[artharm-
363 sounding][x^ ric...]", "[artharm-sounding][sfz]", "[artharm-sounding][trem][sfz]",
364   [ crece, // staccatissimo sube 1
365     crece2, // staccatissimo con armónico sube 1
366     crece3, // battuto ricochet se incorpora y sube 0.9
367     0.6, // sfz se mantiene 0.6
368     decrece, // sonido ord baja 0.01
369     0.6, // ricochet se mantiene 0.6
370     decrece2, // armónico artificial ord baja 0.01
371     0.5, // armónico artificial ricochet se mantiene
372     decrece3, // armónico artificial sfz baja 0.2
373     decrece4 // armónico trem sfz baja 0.1
374     // 0.5 // matices se mantienen pero con staccatissimo
375   ].normalizeSum, ~numeros[6] - 2).asStream.next.yield;
376   }
377
378 }).asStream,
379
380 ~artGGrad = Pseq([Pseq(["[x^ accelYRallAPlacer]", 2), ~gradienteG], ~numeros[6]).asStream,
381

```

Figura 3.32 Escena con gradientes internos

Existen dos arreglos para la duración general de las escenas y para el número de valores que necesita generar FOMUS para la partitura, como por el momento la forma general es aún limitada, no he visto la necesidad de plantear una solución más sofisticada, pero en un futuro probablemente se tenga que implementar una función que sume las duraciones e instancie el valor a la variable de duración que será utilizado en el `.wait` de la rutina general.

En cuanto a las duraciones, se propone una solución similar a la de la familia anterior de obras, en las que se establece una grilla cuantizada en valores mínimos de 0.125, en esta ocasión por las razones prácticas de lectura a primera vista que se requiere. Son divididas en varias instancias de patrones Prand (distribución

uniforme) y Pexprand (distribución exponencial) para representar tendencias de escoger valores cortos o valores largos, y finalmente empaquetados dentro de un arreglo general de duraciones.

Posteriormente se irán agregando las alturas, de la manera expresada al inicio de esta sección instanciados dentro de patrones. Entonces por ejemplo, si partimos de la primera escena con la que podría comenzar la obra, se presenta una estructura determinista que elige un patrón que escoge aleatoriamente entre valores que comprenden 1.0 .. 3.0 dentro de la grilla previamente mencionada. Forzosamente (por el momento) comenzará con valores relativamente largos e intercambiando entre alturas establecidas dentro del conjunto de cuartos de tono, los cuatro instrumentos están haciendo esto.

Después se le agrega un operador de índice acústico, estableciendo el registro en el que serán ejecutadas las notas. Algunas de las escenas contienen gradientes de probabilidades en los que se van alejando de algún índice acústico hacia una determinada región, o estableciendo disoluciones de sonidos a texturas puntillistas, en este caso creciendo la probabilidad de que se escoja un silencio.

Las probabilidades van fluctuando de escena en escena (dentro de las que contienen modulaciones o *tendency masks* de funciones de probabilidad) y de acuerdo al camino que se escogió, podrán presentarse opciones que yo consideré viables para dichas transformaciones.

Para las articulaciones implementé distintos patrones que serán escogidos de acuerdo a la selección de escenas establecida, por el momento necesariamente están

ligadas a la escena que se escogió, aunque aquí también están presentes el manejo de gradientes. Acto seguido vinculé las posibilidades combinatorias permitidas dentro de un *Event* utilizado como un *namespace* de *Pbinds* que engloban todos los parámetros y sus transformaciones.

Finalmente todas las escenas son agrupadas dentro de un *Pdict* como una base de datos de posibles discursos sonoros, para posteriormente ser incorporados dentro de la rutina final que vincula *SuperCollider* con *Fomus* para generar la partitura automáticamente.

Al evaluar el código aparece una sencilla interfaz gráfica que consiste de un botón para comenzar o detener el proceso, los cuatro instrumentistas ejecutan el código presionando el botón al mismo tiempo. En una versión en la que estén conectadas por red las computadoras, solamente sería necesario que lo ejecute uno solo, o incluso desde un ordenador en cabina.

### 3.3.2 ANÁLISIS RETROSPECTIVO *AUTOPO(1)ÉTICA*

En esta obra se pueden encontrar similitudes y marcadas diferencias con respecto a las primeras obras planteadas, las cuales abordaré en seguida.





Figure 3.33 Estreno de *Autopo(i)ética* llevado a cabo por el cuarteto de cuerdas Cromano en el marco del Festival Internacional Cervantino, dentro del concierto de Prácticas de Vuelo del CMMAS (Centro Mexicano para la Música y las Artes Sonoras)

Dentro del proceso creativo experimenté gran parte de lo ya he expuesto en las dos familias de obras precedentes, en cuanto a la retroalimentación inmediata gracias a la automatización, modularidad con estructuras de código similares y demás. Sin embargo una de las diferencias que encuentro significativas en la construcción del sistema de esta obra es que permanece abierta, su expansibilidad es permanente (esto también abarca el polo inmanente) sobre esto Manovich dice:

[...] “*new media*” is new because new properties (i.e., new software techniques) can always be easily added to it. Put differently, in industrial (i.e., mass-produced) media technologies, “hardware” and “software” were one and the same thing [...] In its very structure computational media is “*avant garde*” since it is constantly being extended and thus redefined. (Manovich, 2013: 92-93)

Por ejemplo, en este momento me encuentro trabajando sobre una generación de una base de datos más flexible para esta obra a partir de algoritmos genéticos. Ya conseguí que funcionara esto con arreglos de caracteres alfanuméricos. Proponiendo una “meta”, que puede ser determinista o indeterminista y que musicalmente se podría traducir a un gesto, un acorde, una textura o lo que las necesidades expresivas requieran, se parte de una entrada completamente distorsionada (una población) o incluso valores aleatorios que no tienen nada que ver con la meta (como se verá en el siguiente ejemplo), se van produciendo nuevas generaciones con operaciones de mutación y *mating pools* cada vez más aptas para la “supervivencia”, se va acercando paulatinamente a la meta, o a una meta que incluya variaciones que giran alrededor de una norma.

Entonces con los caracteres alfanuméricos propongo un texto meta como por ejemplo: “ya merito termino mi tesis” y el programa va avanzando de:

m2'oU<iW[>3Ap-epQ?GgT>)\*i}

m\_'oePaWe>:%:p\$bQkGK|>)\*i7

\_'oePaWe>:%:p\$bQkGK|>)\*i7

m2'oePaWe>3:p\$b5kGK|>?\*)}

hacia:

Yo meriio ter ino mi tesis

Yo merito ter ino mi tesis

Yo morito ter ino mi teais

Yo moritott r ino mi tesis

Yo morito t r ino mi tesis



planteado anteriormente, y presentar otra posibilidad para el concierto de la próxima noche. Entonces con la camioneta en movimiento me puse a programar un par de escenas adicionales que fueron implementadas y presentadas al día siguiente en el concierto. Además, se debe de tomar en cuenta, que es altamente probable la necesidad de hacer modificaciones de código al estar actualizando sistema operativo, quizá cambien algunas dependencias o compatibilidades volviendo algunas funcionalidades obsoletas, estas modificaciones de código *pueden* resultar en cambios estéticos.

Por otro lado, dentro del nivel estético podríamos regresar a la pregunta que hace Nattiez, ¿dónde comienzan y terminan cada una de los niveles del modelo tripartito del hecho musical? (Nattiez, 1990: 72) Si tomamos en cuenta la experiencia de lectura de obra por un músico como parte poética—en el sentido de que se está recreando la obra— asimismo como parte del nivel estético—está escuchando la obra, hasta cierto punto también es escucha— al estar leyendo su parte, (como todo lo que involucra en *Autopo(i)ética*) y las texturas resultantes con el resto de los músicos, el nivel estético sí es afectado; por otro lado quizá no tenga tiempo de analizar y apreciarlo del todo como una experiencia sintagmática, sino como una intensa lectura de datos con eventuales sorpresas al nivel sintagmático, como por ejemplo cambios abruptos del ensamble.

Quizá para el espectador no sea muy sorprendente ver las computadoras en el escenario, no considero que realmente se afecte el polo estético en ese sentido, más que posiblemente generar una expectativa y curiosidad sobre la razón, pudiendo ser

esto abordado con pláticas introductorias (que no son poco comunes) al concierto para contextualizar sobre lo que sucederá (aunque ciertamente esta plática no forma parte del objeto bajo análisis que es la obra). Otra estrategia podría ser la de proyectar la partitura (no las partes) en el escenario, en el que también se vería el programa y el *post window* de SuperCollider o una ventana de la Terminal haciendo los cálculos y procesos.

Por otro lado, el software vuelve a inscribirse en el nivel inmanente en el sentido de que la partitura misma esta siendo generada en tiempo real, ¿cuál es el trazo analizable? ¿Una ejecución particular (de un sistema en constante crecimiento y expansión)? ¿Imágenes de la partitura resultante? (siendo esta borrada con cada *refresh* del programa) ¿El texto dinámico ejecutable mismo? ¿Cuál es la ontología de una obra de esta naturaleza? Como se puede ver surgen una serie de interrogantes a raíz de la *softwarización* en el quehacer musical.

## CAPÍTULO 4

### ANÁLISIS DE ANALOGIQUE A OBRA PRE-INFORMÁTICA DE XENAKIS: HACIA UNA PRE-COMPOSICIÓN ASISTIDA POR ORDENADOR

*“Todos somos hijos de Xenakis”*

—Manuel Rocha Iturbide<sup>63</sup>

En este capítulo me gustaría abordar el análisis que realicé para la obra *Analogique A* de Xenakis, logrando abarcar los siguientes puntos:

- Descripción teórica básica y uso de los modelos de Markov dentro de la música y de otras artes, notoriamente en fechas cercanas a la composición de la obra.
- Exposición del mecanismo tal cual aparece en Formalized Music, así como su implementación en SuperCollider.
- Comparación de resultados obtenidos con su implementación, la partitura, Formalized Music y con el análisis de Di Scipio (2005).
- ¿Qué se obtiene al *softwarizar* la formalización expuesta por Xenakis? ¿Qué beneficios puede tener utilizar un programa SuperCollider como herramienta

---

<sup>63</sup> Manuel Rocha Iturbide (1963) [www.artesonoro.net](http://www.artesonoro.net) , frase dicha cuando fungió como comentarista invitado de mi ponencia en el VII Coloquio de alumnos de posgrado de la entonces ENM ahora Facultad de Música de la UNAM que se llevó a cabo en la Ciudad de México entre los días 13 y 15 de junio del 2012.

de investigación musicológica? ¿Cuál es su potencial para fines de creación musical para otros compositores?

- Abordar *Analogique A* explorando las propiedades que tiene en común con los objetos de nueva media, en cuales discrepa y por qué considero que es una pre-composición asistida por ordenador.
- Comentarios finales y conclusiones.

#### 4.1 MODELOS DE MARKOV

Dentro de la teoría de la probabilidad se le conoce como proceso estocástico a un concepto matemático que ayuda a describir una sucesión de variables aleatorias que se desarrollan en función de otra variable, generalmente temporal ( $t$ ). Una cadena de Markov<sup>64</sup> es un tipo de proceso estocástico donde la probabilidad de que ocurra un evento depende del evento que le precede y generalmente son representados por medio de gráficas de transición o matrices de probabilidades de transición. Si más de un evento es utilizado para el cálculo de las probabilidades de transición entonces estaremos hablando de una cadena de Markov de orden  $n$ , donde la variable  $n$  indica el número de eventos que son tomados en cuenta.

Henry F. Olson, un ingeniero estadounidense dentro de *RCA Records* (1901 – 1982 ) es considerado como el primero en aplicar procesos de Markov para la generación de estructuras musicales alrededor de los 50s, analizando melodías “populares” de Stephen Foster<sup>65</sup> para después producir modelos de Markov de primer y segundo

---

<sup>64</sup> Desarrollado por el matemático ruso Andrei Markov (1856 - 1922) a principios del siglo XX.

<sup>65</sup> Stephen Foster (1826 - 1864) famoso cantautor estadounidense de su época.

orden con respecto a alturas y ritmos implementados en un sintetizador. (Nierhaus, 2009: 71)

El uso de distintos tipos de modelos de Markov dentro de la música es común en contextos de análisis musical como en el trabajo de Wei Chei y Barry Vercoe<sup>66</sup>, igualmente en el caso de imitación estilística como en la armonización de corales al estilo de Bach por Moray Allan a partir de un canto dado. Dan Ponsford, Geraint Wiggins y Chris Mellish utilizan modelos de Markov para producir progresiones de acordes de acuerdo a la práctica de formas de danza en Francia del siglo XVII por compositores como Couperin, Lully y Marais, por mencionar algunos. (Ibíd., 76)

En el ámbito de la composición, una de las primeras implementaciones se encuentra en el cuarto movimiento de la ya mencionada *Illiad Suite* de Lejaren Hiller e Isaacson, obra que generalmente es considerada como la primera composición generada por una computadora, siendo estrenada en agosto de 1956. Aunque en varias fuentes se le atribuye a Xenakis<sup>67</sup> (y el mismo compositor lo considera así) haber sido el primer compositor en utilizar la computadora para crear “música original”<sup>68</sup>.

---

<sup>66</sup> Chei y Vercoe utilizaron modelos de Markov ocultos o HMM (por sus siglas en inglés) para distinguir entre distintos estilos musicales donde se analizó un corpus de música *folk* de Austria, Alemania e Irlanda.

<sup>67</sup> Es interesante notar que Xenakis y Hiller tuvieron correspondencia a principios de los 60s, años en los que el compositor de ascendencia griega trabaja sobre su serie ST: “Letter from Iannis Xenakis, April 10, 1962 in which he asks Hiller to send a copy of his book, *Experimental Music*, with its description of *Illiad Suite*.” (Bewley, 2004) Consultado en: <http://library.buffalo.edu/music/exhibits/hillerehibitssummary.pdf>



Los primeros contactos de Xenakis con la computadora le fueron facilitados mediante un ingeniero de la IBM François Genuys con el cual entabló contacto en 1961, y ya en 1962 se le permitió tener acceso a unas horas de trabajo con una IBM 7090. Hay que recordar que era sumamente difícil tener acceso a una computadora en estos años y más aún para fines musicales. Producto de este contacto fue el comienzo de la familia de obras ST, dentro de las cuales retoma y amplía algunos conceptos de obras anteriores como *Achorripsis* (1956-7) y *Pithoprakta* (1955-56). Makis Solomos afirma refiriéndose a la serie ST de Xenakis que resultó de este contacto: "l'ordinateur est introduit, pratiquement pour la première fois, dans la musique, à des fins de création" y en pie de página menciona a Hiller de la siguiente manera: "Lejaren Hiller s'était déjà servi de l'ordinateur, mais en proposant une œuvre, l'*Illiac Suite* (1956), qui n'innove pas musicalement, puisqu'elle est fondée sur une codification du contrepoint dans le style de Palestrina." (Solomos, 1996)<sup>69</sup>



Figura 4.1 Lejaren Hiller junto a la computadora Illiac. <http://www.mediaartnet.org/works/illiac-suite/>

---

<sup>69</sup> Consultado en versión electrónica en: <http://www.univ-montp3.fr/~solomos/Chapitr1.html>.

Es curioso notar que, aparentemente sin conocerse, los modelos de Markov también habían sido aplicados en análisis de literatura y en la generación de poesía experimental alrededor del mismo tiempo en que Xenakis estaba concibiendo *Analogique A et B*, por Theo Lutz<sup>70</sup> (1932 - 2010) y Max Bense<sup>71</sup> (1910 - 1990) quienes en 1959 colaboraron en la *Technische Hochschule Stuttgart* (de nuevo una institución académica) para procesar frases de la novela “El Castillo” de Franz Kafka con un programa que implementaba cadenas de Markov. Al resultado le llamaron “*Die Stochastischen Texte*”. Por lo tanto a mediados y finales de los 50s, y sin conocerse entre sí, en Illinois nacía la primera obra compuesta por computadora, en Stuttgart nacían los textos estocásticos y en Paris la música estocástica, teniendo como uno de los tantos puntos de intersección los modelos de Markov.



Figura 4.2 IBM 7090 mismo modelo en el que Xenakis tuvo sus primeros contactos.

---

<sup>70</sup> Theo Lutz en ese entonces alumno de filosofía, matemáticas e informática. Se puede encontrar una implementación de este programa: [http://auer.netzliteratur.net/0\\_lutz/lutz\\_original.html](http://auer.netzliteratur.net/0_lutz/lutz_original.html) adaptado a PHP (https://secure.php.net/) por Johannes Auer.

<sup>71</sup> Max Bense filósofo, escritor y publicista alemán.

Posiblemente los escritos de Claude Shannon<sup>72</sup> (1916 - 2001) fueron punto de partida para las diversas implementaciones hechas en estas épocas. Además de las mencionadas, poco después en los 60s el manifiesto de *Oulipo* llamaba a “reinvigorate the old poetic collage form of the cento [...] by a few considerations taken from Markov’s chain theory” (Primer manifiesto Oulipo citado en Cramer, 2005: 74) su artículo de 1948 *A Mathematical Theory of Communication* coinciden fechas muy cercanas a estas implementaciones y se sabe que Xenakis estaba al tanto de sus escritos y teorías, mencionándolo en diversas entrevistas (Varga, 1996) (Matossian, 2005) y escritos (Xenakis, 1992).

#### 4.2 TEORIZACIÓN DE MÚSICA ESTOCÁSTICA MARKOVIANA EN FORMALIZED MUSIC

En su libro titulado *Formalized Music* Xenakis dedica dos capítulos enteros para detallar y teorizar sobre la implementación de los modelos de Markov en su pieza *Analogique A et B*, la cual originalmente fue concebida como dos piezas independientes construidas esencialmente con los mismos procedimientos, siendo estas *Analogique A* (1958) para 9 instrumentos de cuerda (3 violines, 3 cellos y 3 contrabajos) y *Analogique B* (1959) para cinta electromagnética. Esta última en principio fue compuesta en formato mono en los estudios Gravesaner de Scherchen, posteriormente siendo proyectada a través de ocho altoparlantes, ambas obras *A* y *B* fusionadas para formar la versión final.

A continuación y en el resto de esta sección detallaré los conceptos del mecanismo construido por Xenakis para la obra.

---

<sup>72</sup> Claude Shannon fue un notable matemático e ingeniero electrónico estadounidense considerado como el “padre de la teoría de la información”.

Para esta pieza el compositor toma en consideración tres parámetros musicales sobre los cuales operará su “mecanismo compositivo”, siendo estos altura, dinámica y densidad representados por las variables  $f$ ,  $g$  y  $d$  respectivamente. Estas variables son consideradas al inicio como partes de un solo gránulo o cuanto sonoro (exceptuando densidad al menos que densidad equivalga a 1), al estar abordando la naturaleza del sonido a partir de una concepción corpuscular o cuántica. “All sound is an integration of grains, of elementary sonic particles, of sonic quanta. Each of these grains has a threefold nature: duration, frequency and intensity. All sound, even all continuous sonic variation is conceived as an assemblage of a large number of elementary grains adequately disposed in time.” Atribuyendo esta idea como una aproximación a las nociones introducidas por Dennis Gabor<sup>73</sup> en la teoría de la información. (Xenakis, 1992: 43, 373)

En el año en que Xenakis escribió *Analogique A*, una especie de música granular instrumental, ya había compuesto la miniatura *Concret PH* en los estudios del GRM, esta obra es generalmente reconocida como el primer acercamiento a la síntesis granular, y una joya de la música electroacústica temprana, en la cual se puede escuchar los crujidos de una tiza quemándose. “Together with the tape work *Concret PH* (1958), realized with a rough technique of granular *transformation* (not *synthesis*) of sound, *Analogique B* represents the first music ever made leaning on a corpuscular view of the physical world”. (Di Scipio, 2005: 2)

---

<sup>73</sup> Aunque en años posteriores se lo atribuiría a Einstein. (Di Scipio, 2001)

Xenakis, que estaba profundamente influenciado por los sonidos de la naturaleza, y tenía una pronunciada visión espacial del sonido por su misma profesión y experiencia como ingeniero civil, luego arquitecto en el despacho de Le Corbusier donde laboró por 12 años, puso en términos fantásticos las imágenes que le evocaban este acercamiento cuántico al sonido:

A complex sound may be imagined as a multi-colored firework in which each point of light appears and instantaneously disappears against a black sky. But in this firework there would be such a quantity of pints of light organized in such a way that their rapid and teeming succession would create forms and spirals, slowly unfolding, or conversely, brief explosions setting the whole sky aflame. A line of light would be created by a sufficiently large multitude of points appearing and disappearing instantaneously. (Ibíd., 43-44)

Unos años después podría materializar estas ideas al comenzar a trabajar sobre sus *Diatopes* en los 60s.

Regresando a las variables mencionadas  $f$ ,  $g$  y  $d$ , toma en cuenta una variable  $\Delta t$  como una especie de cuantización y sampleo constante de estos gránulos, por lo cual en esta instancia no será modificada. Aunque en técnicas granulares y de microsonido posteriores la duración (tamaño) del gránulo y su frecuencia (distancia de un gránulo a otro) sí son tomados en consideración.

Después de establecer las bases de una concepción granular del sonido y las variables que se están tomando en cuenta, Xenakis procede a describir una definición preliminar de *screen*, que podría ser traducida como “reja”, “grilla” o “pantalla”, términos que utilizaré de manera intercambiable. Estas rejas consisten en disposiciones de los gránulos en un tiempo  $\Delta t$  dado sobre una cuadrícula en la

cual están expresadas grupos de gránulos (densidad) con distintas alturas (frecuencia) e intensidad. Cada una de estas cuadrículas con sus “nubes” o agrupaciones de gránulos específicos, pueden ser dispuestas sucesivamente una después de otra para crear un discurso sónico<sup>74</sup>.

Por lo tanto, una reja o *screen* puede ser descrita como un conjunto de “nubes” que a su vez está compuesto de gránulos elementales (o su ausencia, pensando en un conjunto vacío) con sus variables respectivas  $f$  y  $g$ . Estos gránulos de sonido crean una densidad ( $d$ ) que Xenakis describe que está distribuido de manera *ergódica*: “The ergodic principle states that the capricious effect of an operation that depends on chance is regularized more and more as the operation is repeated. Here it is understood that a very large succession of screens is being considered.” (Ibíd., 56)

Ya teniendo definidas las variables de los gránulos y de la construcción de rejillas a partir de los mismos, Xenakis procede a construir un mecanismo a partir del cual podrá elaborar un proceso estocástico, o como el lo define “an analogue of a stochastic process” (Ibíd., 81) que servirá tanto para la producción de entidades sonoras como para sus transformaciones a través del tiempo.

La altura dentro de *Analogique A* es manejada con el sistema de temperamento igual, mientras que en *Analogique B* es trabajada por frecuencias lo cual es distinto porque es manejado de manera en que estas frecuencias pueden caer fuera del temperamento igual, es decir, de nuestra gama cromática dividida en semitonos.

---

<sup>74</sup> Más adelante, Xenakis imagina cada uno de estos gránulos con vectores (Ibíd., 54-56), estableciendo las bases del *Glisson Synthesis* que sería desarrollado en tiempos posteriores.

Divide el rango de alturas en seis regiones distintas, las cuales a su vez están agrupadas en dos arreglos que denomina  $f_0$  y  $f_1$ .

		$(f_0)$	$(f_1)$
1	=	E <sub>0</sub> ... E <sub>1</sub>	I
2	=	E <sub>1</sub> ... D <sub>2</sub>	II
3	=	D <sub>2</sub> ... D <sub>b3</sub>	I, II, III
4	=	D <sub>b3</sub> ... C <sub>4</sub>	IV
5	=	C <sub>4</sub> ... B <sub>4</sub>	III
6	=	C <sub>4</sub> ... A <sub>5</sub>	IV

$(f_0)$  [1, 2, 5, 6]

$(f_1)$  [3, 4]

(A<sub>3</sub> = 440 Hz<sup>75</sup>)

Se puede apreciar que por un lado  $f_0$  y  $f_1$  se complementan abarcando el rango completo de alturas sobre el cual se trabajará, y por otro que  $f_0$  cubre los registros extremos y  $f_1$  el registro medio. Di Scipio hace la observación de que no se especifican claramente las fronteras entre las regiones: "Xenakis did not seem to have clarified whether the two extremes in a particular set or region do or do not belong to the set. For example, does D<sub>2</sub> belong to region II or III? Does it belong to both?" Al entablar el análisis, pude observar que prácticamente pertenece a ambos, y que en repetidas ocasiones se invaden ligeramente los registros.

---

<sup>75</sup> Xenakis utiliza el sistema franco-belga para la notación del índice acústico en el cual el "La central" o "la 440" corresponde al índice acústico 3.

Xenakis estaba al tanto de las investigaciones de psicoacústica llevadas a cabo por Harvey Fletcher y Wilden A. Munson en la década de los 30s<sup>76</sup>, a partir de las cuales dedujeron una gráfica que expresa las curvas isofónicas. En esta gráfica se pueden apreciar las curvas calculadas para la relación existente entre la percepción de frecuencia e intensidad, para que estos sonidos sean percibidos con la misma intensidad. Dichas pruebas fueron llevadas a cabo con ondas sinusoidales, quizá siendo una influencia para que en *Analogique B* decidiera crear gránulos elementales a partir de sinusoidales.

Xenakis hace mención de estos estudios: "The ear is more sensitive at the center of the audible area. At the extremities it perceives less amplitude and fewer melodic intervals."(1992: 47-49) Posiblemente sea esta la razón por la cual, como se verá más adelante, que la definición de sus rejas o están agrupadas la alturas en los extremos, o están agrupadas en la región central, además de disposiciones que discrepan de las que teoriza que también pudieron haber sido influenciadas por estas razones en una decisión final para hacer modificaciones en la partitura, esto será clarificado en su sección correspondiente.

Retornando a las variables de las alturas, el mecanismo dentro del cual operan estos arreglos de alturas escoge uno de los dos arreglos, selecciona una de las regiones que se encuentra dentro del arreglo  $f_0$  ó  $f_1$  y finalmente elige una altura específica dentro de la región.

---

<sup>76</sup> Robison y Dadson hicieron una revisión en 1956, apenas un par de años antes de *Analogique A*.



La manera en que se escoge uno de los dos arreglos sigue determinadas reglas en que se entretajan los demás parámetros, mientras la elección de región y altura se lleva a cabo con una función de distribución uniforme, cada una de las alturas tiene la misma probabilidad de ser seleccionada dentro de un rango establecido.

Para las dinámicas se consideran tres valores, que a su vez están ligadas a las regiones de alturas descritas, e igualmente agrupados en dos arreglos distintos.

$(g_0)$	$pp$	$f$	$ff$
	II, III	I	IV

$(g_1)$	$pp$	$f$	$ff$
	I, IV	II, III	

Podemos notar que en el arreglo  $g_0$  se toman en cuenta los tres elementos mientras que en  $g_1$  se descarta el tercer elemento  $ff$ . Y que tanto las alturas como las frecuencias están ligadas por medio de números romanos, por decir  $f_0g_0$  consiste en alturas dentro de las regiones extremas 1, 2, 5 y 6 ligadas a  $f$ ,  $pp$ ,  $pp$ ,  $ff$  respectivamente<sup>77</sup>.

---

<sup>77</sup> Es importante hacer mención que en (Xenakis, 1992) liga los parámetros de esta manera por medio de números romanos, mientras en (Di Scipio, 2005) expone el número de regiones en las alturas del 1 – 6, como I – VI, lo cual al estar revisando y comparando análisis puede resultar un poco confuso.

Por densidad Xenakis se refiere a la cantidad de eventos (notas) que suceden dentro de una unidad de tiempo ( $\Delta t$ ), los valores que se toman en cuenta siguen una disposición logarítmica<sup>78</sup>:

( $d_0$ )	1 evento / $\Delta t$	3 eventos / $\Delta t$	9 eventos / $\Delta t$
	II, III	I	IV
( $d_1$ )	1 evento / $\Delta t$	3 eventos / $\Delta t$	9 eventos / $\Delta t$
	IV	II, III	I

El valor de  $\Delta t$  equivale a la mitad de un compás de la obra, con  $\text{♩} = 50$  lo cual equivaldría a 1.2 segundos. Se puede observar que los números romanos que ligan los variables de alturas e intensidades también están dentro del parámetro de densidades, lo cual termina por enlazar los tres dominios que se están tomando en cuenta, esto lo veremos un poco más adelante cuando se detalle el funcionamiento del mecanismo.

Las duraciones de las notas no son consideradas como variables dentro del mecanismo, en la partitura se puede apreciar que tres instrumentos (vln1, vc1 y cb1) siempre tocan tresillos contra la mitad de duración de compás ( $\Delta t$ ), tres

---

<sup>78</sup> Di Scipio hace la observación de que esta disposición : “appears to be consistent with the Fechner law, in a perceptual domain (“density”) that at the time had not been yet investigated (and probably not even recognized as such) by psychoacousticians.” (Ibíd., 9)

instrumentos valores de semicorchea ( $vln2$ ,  $vc2$  y  $cb2$ ) y los tres restantes cinquillos ( $vln3$ ,  $vc3$  y  $cb3$ ). Es sobre estas figuraciones rítmicas que se distribuyen las distintas densidades posibles en una especie de cuadrícula discreta cuantizada por  $\Delta t$  que sería el valor de un pulso dentro del compás de  $2/2$ ; por lo tanto por cada  $\Delta t$  habrán  $5 + 3 + 4$  por cada tres instrumentos de posibilidades.

#### 4.2.1 DESCRIPCIÓN DEL MECANISMO MTPZ.

Xenakis estableció probabilidades a las cuales llamó protocolo, para conectar los parámetros discutidos. Estos son representados por medio de matrices de probabilidades de transición MTP (por sus siglas en inglés). Por ejemplo<sup>79</sup>:

$\downarrow$	$f_1$	$f_2$	$f_3$	$f_4$	...
$g_1$	0.5	0	0.2	0	...
$g_2$	0	0.3	0.3	1	
$g_3$	0.5	0.7	0.5	0	

La forma de interpretar la matriz es la siguiente, para cada valor de altura  $f_i$  hay uno o varios valores correspondientes de intensidad  $g_i$  definidos por la probabilidad indicada. Por ejemplo, la frecuencia  $f_1$  tiene un 50% de probabilidad de correspondencia con las intensidades  $g_1$  y  $g_3$  mientras que con  $g_2$  tiene una probabilidad nula.

---

<sup>79</sup> Ejemplo extraído de (Xenakis, 1992: 45).

Las matrices de probabilidades de transición que son utilizadas para controlar los parámetros de altura, intensidad y densidad dentro de *Analogique A* funcionan de la misma manera, son matrices que representan una cadena de Markov de primer orden, es decir, la probabilidad de transición de un estado a otro depende del estado en que se encuentra, es por ello una estructura que se desenvuelve en el tiempo. El compositor se refiere a esto diciendo que “we can inject memory into the stochastic method as well — and here we have arrived at the Markov Chain.” (Varga, 1996: 82)

Por lo tanto para cada una de las tres variables mencionadas, se utilizan los siguientes protocolos para la selección del arreglo correspondiente.

Para las alturas,

	$\alpha$			$\beta$	
$\downarrow$	$f_0$	$f_1$	$\downarrow$	$f_0$	$f_1$
$f_0$	0.2	0.8	$f_0$	0.85	0.4
$f_1$	0.8	0.2	$f_1$	0.15	0.6

intensidades,

	$\gamma$			$\varepsilon$	
$\downarrow$	$g_0$	$g_1$	$\downarrow$	$g_0$	$g_1$
$g_0$	0.2	0.8	$g_0$	0.85	0.4
$g_1$	0.8	0.2	$g_1$	0.15	0.6

y finalmente densidades:

	$\lambda$			$\mu$	
$\downarrow$	$d_0$	$d_1$	$\downarrow$	$d_0$	$d_1$
$d_0$	0.2	0.8	$d_0$	0.85	0.4
$d_1$	0.8	0.2	$d_1$	0.15	0.6

Entonces, para ejemplificar, si nos encontráramos en la TPM  $\alpha$  en el arreglo  $f_0$  habría 20% de probabilidad de permanecer en el mismo arreglo y un 80% de probabilidad de cambiar a  $f_1$ .

Aplicando la ley de probabilidad compuesta entre las dos matrices, veamos la probabilidad de que  $f_i$  dentro de un tiempo  $t$  seguirá siendo  $f_i$  al momento de aplicar  $t + \Delta t$  o cambie a  $f_j$ .

$$p(f_0 \rightarrow f_0) = (0.2 + 0.85)/2 = 0.525$$

$$p(f_0 \rightarrow f_1) = (0.8 + 0.15)/2 = 0.475$$

$$p(f_1 \rightarrow f_0) = (0.8 + 0.4)/2 = 0.6$$

$$p(f_1 \rightarrow f_1) = (0.2 + 0.6)/2 = 0.4$$

Como podemos ver hay una ligera tendencia hacia  $f_0$ , lo que se traduce auditivamente a que hay un favorecimiento a las alturas de registros extremos y a su vez tiene su repercusión en el entretejido con los parámetros restantes como veremos más adelante.

Posteriormente Xenakis determinó unas reglas de acoplamiento para conectar todas las matrices:

$f_0$	$f_1$	$d_0$	$d_1$	$g_0$	$g_1$	$g_0$	$g_1$	$f_0$	$f_1$	$d_0$	$d_1$
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
$\lambda$	$\mu$	$\alpha$	$\beta$	$\lambda$	$\mu$	$\beta$	$\alpha$	$\gamma$	$\varepsilon$	$\gamma$	$\varepsilon$

La manera en que operaría sería por ejemplo, si se escoge la altura del arreglo  $f_0$  entonces la variable de intensidad será calculada con la matriz  $\lambda$  y el valor de la densidad con la matriz  $\gamma$ ; si al contrario la altura es seleccionada del arreglo  $f_1$  la densidad será seleccionada de la matriz  $\mu$  y la intensidad de la matriz  $\varepsilon$ . Si están las dos posibilidades, por decir que una variable haga mapeo con  $f_0$  y otra con  $f_1$ , entonces se seleccionará aleatoriamente el conjunto, por medio de una distribución uniforme (en este caso 50% y 50% de probabilidad). (Ibíd., 83-84)

El número total de combinaciones posibles es de ocho, correspondiendo a las ocho rejillas:

$A = f_0 g_0 d_0$	$B = f_0 g_0 d_1$	$C = f_0 g_1 d_0$	$D = f_0 g_1 d_1$
$E = f_1 g_0 d_0$	$F = f_1 g_0 d_1$	$G = f_1 g_1 d_0$	$H = f_1 g_1 d_1$

Estas combinaciones de parámetros son la versión final de la anterior definición preliminar de *screens*, sus protocolos y reglas internas desenvuelven el discurso de la obra *Analogique A*.

Veamos ahora como se conectan estas rejillas y como se construye el mecanismo base sobre el cual se puede apreciar al menos idealmente la distribución estadística de la pieza en su totalidad. Supongamos que estamos en la rejilla  $f_0 g_1 d_1$  en determinado momento  $\Delta t$  y queremos calcular la probabilidad de que se convierta en la rejilla  $f_1 g_1 d_0$  en el momento de  $t + \Delta t$ .

Reja en el momento $t$	$f_0$	$g_1$	$d_1$
Matrices de probabilidades por categoría de parámetro	$\alpha$	$\varepsilon$	$\mu$
	$\beta$	$\gamma$	$\lambda$
Reja en el momento $t + \Delta t$	$f_1$	$g_1$	$d_0$
Valores de probabilidades tomadas de la MTP que corresponden a los parámetros de acoplamiento	0.8	0.6	0.4
	0.15	0.2	0.8
Probabilidades compuestas	0.475	0.4	0.6

(Xenakis, 1992: 88)

De la tabla anterior podemos calcular la probabilidad compuesta para eventos independientes:  $0.475 \cdot 0.4 \cdot 0.6 = 0.114$ , siendo esta la probabilidad de que la rejilla  $f_0$

$g_1 d_1$  se transforme en  $f_1 g_1 d_0$ . Si extendemos el mismo procedimiento a las ocho rejas en cuestión podremos construir la matriz de probabilidades de transición general la cual refleja las probabilidades de transformación sobre la que evoluciona la obra. (En el apéndice y en los ejemplos descargables se encuentra el código para calcular esto dentro de SuperCollider)

Es notoria cierta simetría dentro de las ocho rejas, la complementariedad entre los arreglos  $f$  que ya se había mencionado es evidentemente reflejada en las rejas; A, B, C y D abarca alturas dentro de los registros extremos mientras E, F, G y H cubre la región intermedia.

Por otro lado se puede apreciar que las rejas A, B, E y F tienen una mayor tendencia a ser  $pp$  que  $f$  o  $ff$  revisando las probabilidades de los componentes, pero generalizando realmente tiene probabilidades similares de tener dinámica suave ( $pp$ ) o de mayor intensidad ( $f$  o  $ff$ ), mientras que en las rejas restantes jamás serán  $ff$ .

Dentro de las densidades hay una distribución más uniforme de sus componentes, las rejas A, C, E y G tienen un máximo de 14 eventos /  $\Delta t$ , mientras B, D, F y H presentan un máximo de 16 eventos /  $\Delta t$ , aunque la regla general ya en práctica es que varíe ligeramente alrededor de el número promedio de densidad.

Recapitulando el funcionamiento del mecanismo en *Analogique A*, Xenakis generó secciones (que Xenakis llamó perturbaciones o zonas de equilibrio y que abordaré en seguida) donde en cada una opera el mecanismo reiteradamente seleccionando la reja de acuerdo a sus probabilidades. Cada vez que se escoge una nueva reja se



seleccionan los arreglos por medio de las matrices correspondientes de acuerdo a las reglas de correspondencia o *coupling*, después se elige un elemento de cada arreglo por medio de una función aleatoria (distribución uniforme) y finalmente se mapean los valores a la notación musical. En este último paso no se estableció ningún proceso formalizado, fue dejado a libertad de Xenakis.

Todo el proceso descrito se repite cada  $\Delta t$ , es decir cada medio compás siendo el tempo  $\text{♩} = 50$ , lo que se traduce a cada 1.2 segundos dentro de una sección, 97 veces a lo largo de la obra entera.

Ya habiendo hasta aquí expuesto la manera en que funciona el mecanismo construido por Xenakis en la siguiente sección me gustaría ahondar un poco más en detalles que conciernen la forma general, la construcción de las rejillas, diferencias que encontré entre la manera en que teorizó minuciosamente y los cambios que hizo ya en la práctica dentro de la partitura, así como exponer modificaciones que hizo de manera reiterada a 4 de las 8 rejillas, nunca presentándolas de la manera expuesta teóricamente y las razones por las cuales considero que hizo esto.



Screen A ( $f_0 g_0 d_0$ )

#									$\frac{9}{IV}$
f	$\frac{3}{I}$								
pp		$\frac{1}{II}$					$\frac{1}{III}$		
	$E_0$	$E_1$	$D_2$	$D\flat_3$	$C_4$	$B_4$	$A_5$		

Screen B ( $f_0 g_0 d_1$ )

#									$\frac{1}{IV}$
f	$\frac{9}{I}$								
pp		$\frac{3}{II}$					$\frac{3}{III}$		
	$E_0$	$E_1$	$D_2$	$D\flat_3$	$C_4$	$B_4$	$A_5$		

Screen C ( $f_0 g_1 d_0$ )

#									
f		$\frac{1}{II}$					$\frac{1}{III}$		
pp	$\frac{3}{I}$								$\frac{9}{IV}$
	$E_0$	$E_1$	$D_2$	$D\flat_3$	$C_4$	$B_4$	$A_5$		

Screen D ( $f_0 g_1 d_1$ )

#									
f		$\frac{3}{II}$					$\frac{3}{III}$		
pp	$\frac{9}{I}$								$\frac{1}{IV}$
	$E_0$	$E_1$	$D_2$	$D\flat_3$	$C_4$	$B_4$	$A_5$		

Screen E ( $f_1 g_0 d_0$ )

#				$\frac{9}{IV}$					
f			$\frac{3}{I}$						
pp			$\frac{1+1}{II+III}$						
	$E_0$	$E_1$	$D_2$	$D\flat_3$	$C_4$	$B_4$	$A_5$		

Screen F ( $f_1 g_0 d_1$ )

#				$\frac{1}{IV}$					
f			$\frac{9}{I}$						
pp			$\frac{3+3}{II+III}$						
	$E_0$	$E_1$	$D_2$	$D\flat_3$	$C_4$	$B_4$	$A_5$		

Screen G ( $f_1 g_1 d_0$ )

#									
f			$\frac{1+1}{II+III}$						
pp			$\frac{3}{I}$	$\frac{9}{IV}$					
	$E_0$	$E_1$	$D_2$	$D\flat_3$	$C_4$	$B_4$	$A_5$		

Screen H ( $f_1 g_1 d_1$ )

#									
f			$\frac{3+3}{II+III}$						
pp			$\frac{9}{I}$	$\frac{1}{IV}$					
	$E_0$	$E_1$	$D_2$	$D\flat_3$	$C_4$	$B_4$	$A_5$		

Figura 4.3 Ocho rejas expuestas en (Xenakis, 1992).

### 4.3 FASES ESTACIONARIAS Y PERTURBACIONES

*Analogique A* consiste de 10 secciones musicales<sup>80</sup> de alrededor de 30  $\Delta t$  o 15 compases cada una, dentro de las cuales aplica el mecanismo recién descrito respetando los pesos de sus probabilidades de transición (ver figura anterior) en donde dice que está en equilibrio “E” (en el sentido de equilibrio termodinámico Boltzmanniano) o una serie de perturbaciones P de mayor o menor fuerza sobre el mecanismo con las cuales son deformadas las probabilidades de transición y se aleja en mayor o menor medida de la zona de equilibrio o del estado estacionario.

Antes de abordar la construcción y aplicación del mecanismo, Xenakis elabora sobre el concepto de orden, que tendrá peso significativo en la decisión del estado estacionario del MTPZ y las perturbaciones aplicadas sobre él. Comienza con un ejemplo de masas de personas<sup>81</sup> para explicar el concepto de orden y desorden:

A crowd of 500, 000 persons is assembled in a town square. If we examine the group displacement of this crowd we can prove that it does not budge. However, each individual moves his limbs, his head, his eyes, and displaces his center of gravity by a few centimeters in every direction. If the displacements of the centers of gravity were very large the crowd would break up with yells of terror because of the multiple collisions between the individuals. The statistical values of these displacements normally lie between very narrow limits which vary with the density of the crowd, from the point of view of of these values as they affect immobility, the disorder is weak. (Xenakis, 1992: 61)

---

<sup>80</sup> Se puede observar una gráfica en la cual se expone la forma general del conjunto de *Analogique A* et B en (Harley, 2004: 22)

<sup>81</sup> Que es recurrente en su discurso ya que experimentó las crudas represiones de las manifestaciones en los 40s de la resistencia Griega en la segunda guerra mundial, Makis Solomos incluso llama *Metastasis* como un poema sinfónico que describe una de estas anécdotas.

Más adelante menciona que el concepto de orden y desorden puede ser aplicado a un gran número de fenómenos (antes de este ejemplo había elaborado uno sobre el movimiento de las moléculas en gases) y que estos fenómenos o conjuntos contienen una *variedad* de elementos a partir de los cuales se puede estimar un valor cuantitativo de orden o desorden. Entonces los distintos elementos que conforman un cierto fenómeno los denomina como su *variedad* y el grado de orden o desorden como su *entropía*, tomando estos conceptos de “theory of communications, which itself borrows from the second law of thermodynamics (Boltzmann’s theorem  $H$ ).” (Ibíd., 61)

Posteriormente prosigue a calcular la entropía expresada en bits (en apéndice se puede encontrar la fórmula adaptada a SuperCollider para calcular la entropía de los protocolos del mecanismo), si se encuentra en orden absoluto la entropía será cero, y si hay máxima incertidumbre la entropía será 1.

Aplica este concepto a las variables en los nubes de gránulos formulados y propone que se pueden clasificar de acuerdo a una expresión simbólica en la cual agrupa dos parámetros,  $\Delta F$  y  $\Delta G$ , cada una puede tener mayor o menor grado de desorden de manera independiente, además de establecer posibles trayectorias en las que se va modulando el grado de orden o desorden. (Ibíd., 65)

Este mismo concepto será aplicado al nivel de las pantallas: “We shall neglect the entropies of the three variables at the grain level, for what matters is the macroscopic mechanism at the screen level. The fundamental questions posed by

these mechanisms are, 'Where does the transformation summarized by an (MTP) go?'" (Ibíd., 84)

Acto seguido propone dejar 100 mecanismos partir de una variable particular y dejarlas desenvolverse de acuerdo a las probabilidades expresadas, cada una de estos 100 mecanismos será un estado en el cual paulatinamente se llegará a un estado de equilibrio. "Is there a general tendency for the states of the hundred mechanisms, and if so, what is it?" (Ibíd.) Parte de la siguiente matriz para ejemplificar (que es utilizada dentro de la obra):

↓	X	Y
X	0.2	0.8
Y	0.8	0.2

<u>Estado</u>	<u>Mecanismos X</u>	<u>Mecanismos Y</u>
0	100	0
1	20	80
2	68	32
3	39	61
4	57	43
5	46	54
6	52	48
7	49	51
8	50	50
9	50	50
⋮	⋮	⋮

Podemos observar oscilaciones que tienen una tendencia general hacia un estado estacionario hacia el octavo estado, pero que ya se va definiendo desde el cuarto estado. (Ibíd., 85) En el apéndice se puede encontrar una implementación en SuperCollider de la fórmula que describe subsecuentemente para calcular el estado estacionario, que consiste en la multiplicación de la matriz por un vector, en este caso:

$$\begin{vmatrix} 0.2 & 0.8 \\ 0.8 & 0.2 \end{vmatrix} \times \begin{vmatrix} 100 \\ 0 \end{vmatrix} = \begin{vmatrix} 20 \\ 80 \end{vmatrix}$$

El siguiente estado será:

$$\begin{vmatrix} 0.2 & 0.8 \\ 0.8 & 0.2 \end{vmatrix} \times \begin{vmatrix} 20 \\ 80 \end{vmatrix} = \begin{vmatrix} 68 \\ 32 \end{vmatrix}$$

y así sucesivamente hasta llegar al estado estacionario.

Este mismo principio lo aplica sobre la matriz de probabilidades de transición calculada y expuesta anteriormente, expresada como una multiplicación entre un vector y la MTPZ (lo cual también está implementado en SuperCollider), entonces si tomamos en cuenta que la matriz que representa la MTP = Z, ubicado en la página 157 y que V es el vector sobre el cual se efectuará el producto, obtendríamos la siguiente fórmula:

$$V' = ZV$$

(Ibíd., 90 - 92)

Y aplicándolo por ejemplo sobre  $V_C^0$  obtendríamos el siguiente proceso:

$\Delta t = 0$ 

$$V_C^0 = \begin{vmatrix} 0 \\ 0 \\ 100 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{vmatrix}$$

 $\Delta t = 1$ 

$$V_C' = \begin{vmatrix} 8.4 \\ 7.6 \\ 2.1 \\ 1.9 \\ 33.6 \\ 30.4 \\ 8.4 \\ 7.6 \end{vmatrix}$$

 $\Delta t = 2$ 

$$V_C'' = \begin{vmatrix} 19.328 \\ 12.655 \\ 12.958 \\ 10.948 \\ 16.011 \\ 12.057 \\ 7.767 \\ 8.278 \end{vmatrix}$$

 $\Delta t = 3$ 

$$V_C' = \begin{vmatrix} 17.146 \\ 10.907 \\ 13.503 \\ 13.297 \\ 14.163 \\ 11.854 \\ 8.102 \\ 11.03 \end{vmatrix}$$

 $\Delta t = 4$ 

$$V_C' = \begin{vmatrix} 17.021 \\ 11.021 \\ 13.265 \\ 12.986 \\ 14.541 \\ 12.138 \\ 8.241 \\ 10.789 \end{vmatrix}$$

Podemos ver que se llega al estado estacionario la cuarta fase, en el momento  $\Delta t = 4$ .

Xenakis le dedica un espacio considerable a la exposición de los conceptos de orden, desorden y entropía en estos capítulos, lo cual pone en evidencia la importancia de estas nociones en la obra, expresando que se presenta o manifiesta el mecanismo en su estado estacionario para *afirmar* una *identidad* del sistema, al cual después de ser afirmado es *negado* con mayor o menor fuerza mediante perturbaciones P:



[...] at first we argue positively by proposing and offering as evidence the existence itself; and then we confirm it negatively by opposing it with perturbatory states [...] The bi-pole of being a thing and not being this thing creates the whole [...] a dual dialectics is thus at the basis of this compositional attitude, a dialectics that sets the pace to be followed [...] an experiment establishes a body of data, a web which it disentangles from the magma of objective reality with the help of negations and transformations imposed on this body [which is] homothetic with that of the experimental sciences . (Xenakis, 1992: 94-95)

Por lo tanto, para establecer esta negación que paradójicamente reafirma la existencia del sistema, siendo esto para Xenakis en principio audiblemente perceptible<sup>82</sup>, se presentan 100 mecanismos Z simultáneamente, los cuales son comparados y enumerados, después se procede de nuevo al siguiente estado de cada uno de estos 100 mecanismos y se vuelven a enumerar, y así procederá de acuerdo a las probabilidades establecidas, estando en estado estacionario o en perturbación. Por lo tanto estos 100 estados dentro de cada  $\Delta t$  serán dispuestos de manera sintagmática para establecer los encadenamientos de cada una de las rejas.

Por último se presenta la forma general de cada uno de las fases de perturbación o estacionarias, que fueron escogidas libremente dentro de la obra (Ibíd., 96):

$$E \longrightarrow P_A^0 \longrightarrow P'_A \longrightarrow E \longrightarrow P'_C \longrightarrow P_C^0 \longrightarrow P_B^0 \longrightarrow P'_B \longrightarrow E \longrightarrow P'_A$$


---

<sup>82</sup> "To propose the entity or its modification in our particular case of musical composition is to give a human observer the means to perceive the two propositions and to compare them. Then the antitheses, entity and modification, are repeated enough times for the entity to be identified" (Ibíd., 95) Es curioso que a pesar de esta constatación, la afirmación de la identidad de los estados estacionarios E solamente ocurren en 3 de las 10 partes de la obra, siendo las 7 restantes perturbaciones sobre el mecanismo.

#### 4.4 ANÁLISIS POR FASES ESTACIONARIAS Y PERTURBACIONES

A continuación expondré los datos encontrados después de un análisis de reja por reja<sup>83</sup> dentro de las fases estacionarias y perturbaciones, en el apéndice se pueden encontrar todos los datos de las rejillas desglosadas donde se puede hacer una comparación, asimismo al ejecutar el código en SuperCollider se da una versión que se apega estrictamente a las teorizaciones expuestas, dando un importante marco de referencia para hacer la comparación, siendo el programa una importante herramienta musicológica además de fungir como potencial herramienta creativa para obras originales.

##### 4.4.1 FASE ESTACIONARIA E

Dentro de la fase estacionaria E al inicio de la obra, y dentro de las otras dos fases estacionarias, encontré que para las rejillas A, B, C y D a pesar de muy raras veces ser expuestas de forma completa, giran alrededor de variaciones similares, en las perturbaciones P encontré que sí hay modificaciones más fuertes que aparecen de manera reiterada y que abordaré más adelante. Las rejillas E, F, G y H sí fueron transformadas eliminando parte de su integración.

Para la rejilla A por ejemplo dentro de los primeros compases, encontré que fue presentada en 5 ocasiones y que en la región IV la densidad en promedio son 7 alturas en vez de 9; en la región III siempre hay inconsistencias. Algunos ejemplos de estas inconsistencias son: aparece ligeramente fuera de rango, es omitido en un

---

<sup>83</sup> El lector se puede referir a la figura 4.3 donde se encuentran las rejillas propuestas por Xenakis para compararlas con los resultados que expondré a continuación.

par de ocasiones o con dinámicas que no le corresponden; II es omitido en 2 ocasiones y 3 veces completo y I es presentado completo 2 veces y en 3 ocasiones tiene una densidad 4 en vez de 3.

A partir de esto se puede deducir que Xenakis le dio mayor importancia a los registros extremos que comprenden los números romanos IV y I de esta reja, manteniéndolos más estables que su conformación interna I y II. Esto quizá por considerarlo como características más sobresalientes de esta reja en particular y por sus consideraciones psicoacústicas mencionadas anteriormente, en el que a partir de estar en contacto con investigaciones de curvas de percepción isófona afirma que el oído tiende a ser más sensible hacia los registros intermedios, tal vez tomó esta decisión para no opacar la característica sobresaliente de esta reja, aunque al estar ambas en dinámicas *ff* y *f* IV y I respectivamente no sería el caso. Cabe resaltar que en esta fase estacionaria no se presenta completa una sola vez la reja en su totalidad.

A continuación presento 4 instancias de la rejas A calculadas en la implementación que desarrollé en SuperCollider<sup>84</sup>, se puede comparar con instancias de la reja A en la fase estacionaria E en los siguientes compases en la partitura: compás 2 primer tiempo ( $\Delta 3$ ), compás 5 primer tiempo ( $\Delta 9$ ), compás 10 segundo tiempo ( $\Delta 20$ ), compás 11 segundo tiempo ( $\Delta 22$ ).

---

<sup>84</sup> Aún falta implementar una función que corrija la agrupación de silencios para Lilypond, generando un archivo .xml por medio de Fomus se puede abrir el archivo en Sibelius y corregir manualmente, además de definir los cambios de claves.

The image shows a page of musical notation for a string quartet. It features four systems of staves. The first system has a dynamic marking of *ff*. The second system has a measure number '10' in a box. The third system has a dynamic marking of *pp*. The fourth system has dynamic markings of *f*, *pp*, and *f*. The notation includes various rhythmic values, accidentals, and phrasing slurs.

Figura 4.4 Instancias de pantalla A en Analogique A. (Editions Salabert, 1968)

The image shows a page of musical notation for a string quartet, specifically focusing on four instances of a screen A. The staves are labeled Vln1, Vln2, Vln3, Vc1, Vc2, Vc3, Cb1, Cb2, and Cb3. The notation includes dynamic markings such as *ff*, *pp*, and *f*, and articulation markings like *Sva-*. The music is in 2/2 time and features complex rhythmic patterns with triplets and quintuplets.

Figura 4.5 Cuatro instancias de screen A generadas por implementación.

Dentro de la reja B, que aparece en tres ocasiones en la primera fase estacionaria, podemos encontrar diferencias en las variaciones similares a las expuestas en A.

Aquí IV siempre aparece completo constando de una sola altura *ff* en registro extremo agudo, III siempre aparece con una densidad de 2 en vez de 3, similarmente II siempre aparece incompleto con una densidad de 1 o 2 en vez de 3, y finalmente I que generalmente contiene una densidad de 8 o 6 en vez de 9. Aquí podemos ver que a pesar de que si existen variantes estas son ligeras, y al igual que en A no aparece de forma completa en ninguna de sus instancias. Se pueden encontrar ejemplos de la reja B en los compases 1 ( $\Delta 1$ ), compás 2 segundo tiempo ( $\Delta 4$ ), compás 4 primer tiempo ( $\Delta 7$ ).

Figura 4.6 Tres instancias de pantalla B en Analogique A. (Editions Salabert, 1968)

Figura 4.7 Cuatro rejas B generadas con implementación.

En las instancias presentadas de la reja C que aparece cuatro veces en la primera fase estacionaria, encontré que IV aparece completo en la mayoría de las ocasiones, solamente una vez presenta una densidad de 10 en vez de 9. III aparece dos veces con dinámica *pp* en vez de *f* y dos veces de la manera establecida, II es omitido en dos ocasiones y presentado dos veces. Finalmente I siempre aparece completo. Nuevamente vemos una predilección y mayor cuidado con los registros extremos, en esta ocasión quizá más enfatizado por estar en dinámica *pp* tomando en cuenta las consideraciones psicoacústicas. Se pueden encontrar ejemplos de la reja C dentro de la partitura en los compases 3 segundo tiempo ( $\Delta 3$ ), 6 segundo tiempo ( $\Delta 12$ ), 8 segundo tiempo ( $\Delta 16$ ).

This figure shows three instances of screen C in analog mode. The score is arranged in three columns. The first column contains three staves with dynamics *pp* and *ppp*. The second column contains three staves with dynamics *ff* and *f*. The third column contains three staves with dynamics *f* and *f*. The music features various rhythmic patterns and melodic lines across different instruments.

Figura 4.8 Tres instancias de pantalla C en Analogique. (Salabert, 1968)

This figure shows four instances of screen C generated with implementation. The score is arranged in two systems of staves. The first system includes Violin 1 (Vln1), Violin 2 (Vln2), Violin 3 (Vln3), Violin 1 (Vc1), Violin 2 (Vc2), and Violin 3 (Vc3). The second system includes Cello 1 (Cb1), Cello 2 (Cb2), and Cello 3 (Cb3). The music features various rhythmic patterns and melodic lines across different instruments. Dynamics include *pp*, *f*, and *Sva-*. The score is in 2/2 time and includes various articulations and phrasing marks.

Figure 2.9 Cuatro instancias de reja C generada con implementación.

En la reja D, que es escogida en 5 ocasiones dentro de la misma fase estacionaria del inicio, encontré que IV aparece solamente una vez en *ff* en vez de *pp*. III presenta mayor variación al contar con una densidad de 2 en vez de 3 en tres ocasiones, una omisión y una densidad de 1 en vez de 3. Para II variaciones de densidad de 1 en vez de 3 dos veces, 2 en vez de 3 tres veces y por último I presenta variaciones de densidad de entre 7 - 10. Se pueden encontrar ejemplos en los compases 5 segundo tiempo ( $\Delta 10$ ), 8 primer tiempo ( $\Delta 15$ ), 11 primer tiempo ( $\Delta 11$ ), 12 primer tiempo ( $\Delta 23$ ).

The image displays four systems of musical notation, each representing a different instance of 'reja D' in the piece 'Analogique A'. Each system is composed of multiple staves, likely representing different instruments or voices. The notation includes various rhythmic values, accidentals, and dynamic markings such as *f*, *pp*, and *ff*. The first system on the left is marked with *pp* and *f*. The second system is marked with *f* and *pp*. The third system is marked with *pp* and *f*. The fourth system is marked with *pp* and *f*. The notation is complex, with many notes and rests, and some staves have a '1' written above them, possibly indicating a first ending or a specific measure.

Figura 4.10 Instancias de reja D en Analogique A.(Editions Salabert, 1968)



The image displays a musical score for nine instruments: Violin 1 (Vln1), Violin 2 (Vln2), Violin 3 (Vln3), Viola 1 (Vc1), Viola 2 (Vc2), Viola 3 (Vc3), Cello 1 (Cb1), Cello 2 (Cb2), and Cello 3 (Cb3). The score is written in 2/2 time and consists of two measures. The first measure shows a series of eighth notes in Vln1, and various rhythmic patterns in the other instruments, including triplets and quintuplets. The second measure continues these patterns, with some instruments playing sustained notes or rests. Dynamics such as *pp* (pianissimo) and *f* (forte) are indicated throughout the score.

Figura 4.10 Cuatro instancias de reja D generada con implementación.

Con estas observaciones podemos afirmar que las rejadas A, B, C y D fueron tratadas de manera muy similar, al tener mayor peso y cuidado la construcción de sus registros extremos que se encuentran ligado a IV y I, mientras que II y III fueron omitidos o presentados con menor densidad de manera reiterada. Cabe mencionar que dentro de la primera fase estacionaria no se presentan completa ninguna de las rejadas, aunque considero que si tiene cuidado en presentar las características fundamentales de estas, haciendo eco de la afirmación de Xenakis:

Microsounds and elementary grains have no importance on the scale which we have chosen. Only groups of grains and the characteristics of these groups have any meaning. Naturally in very particular cases, the single grain will be reestablished in all its glory. (Xenakis, 1992: 50)

Unas de las inconsistencias más fuertes con respecto a las rejas propuestas, tanto dentro de las fases estacionarias como dentro de las perturbaciones, están en las rejas E, F, G y H. Son cambiadas reiteradamente más allá de las variaciones recién expuestas en las que, a pesar de existir dichas incongruencias, siguen conservando su identidad. Por esta razón propongo que Xenakis realmente se basó en cuatro rejas distintas, o más bien con las cuatro rejas que propone E .. H pero omitiendo la mitad de sus números romanos, que en efecto les da una identidad notoriamente distinta y que expondré a continuación.

La reja E en la primera fase estacionaria es presentada cuatro veces, igualmente con variaciones como en las rejas anteriores pero basadas en una estructura distinta a la expuesta por el compositor. Dentro del análisis encontré que IV presenta una ligera variedad de densidad entre 7 - 8 en vez de 9 en la región 4 de alturas, que corresponde al rango intermedio superior y con una dinámica *ff*; III que debería de aparecer junto con II, cada una con una densidad de 1 compartiendo el mismo registro y en *pp*, son omitidas en *toda* la obra, mientras que I presenta una ligera variación de densidad 2 en vez de 3 o aparece completo. Con estos datos y tomando en cuenta que lo hace a lo largo de toda la obra puedo corroborar la hipótesis de que la reja E fue modificada omitiendo los números romanos II y III. Posiblemente esto se deba a que Xenakis pensó que estas alturas serían opacadas, ya que se encuentran dentro del mismo registro y en una dinámica *pp* con una densidad de 1 + 1, y comparándolo con I que tiene una densidad mayor, de 3 y en dinámica *f*, y IV que está en una región ligeramente más aguda con una densidad de 9 y dinámica *ff*. Por

lo tanto la identidad de E es establecida como una conglomeración de alturas dentro de un registro intermedio con dinámicas *f* y *ff* eliminando los ataques *pp*.

Reja E ( $f_1 g_0 d_0$ ) propuesta por Xenakis:

<i>ff</i>				9 IV		
<i>f</i>			3 I			
<i>pp</i>			1 + 1 II + III			
	1 ( $4_1 - 4_0$ )	2 ( $2_2 - 4_1$ )	3 ( $1_3 - 2_2$ )	4 ( $0_4 - 1_3$ )	5 ( $11_4 - 0_4$ )	6 ( $9_5 - 11_4$ )

Reja E propuesta a raíz de análisis:

<i>ff</i>				9 IV		
<i>f</i>			3 I			
<i>pp</i>						
	1 ( $4_1 - 4_0$ )	2 ( $2_2 - 4_1$ )	3 ( $1_3 - 2_2$ )	4 ( $0_4 - 1_3$ )	5 ( $11_4 - 0_4$ )	6 ( $9_5 - 11_4$ )

En la figura 4.12 se pueden apreciar cuatro instancias de la reja E tal y como fueron teorizadas en (Xenakis, 1992), y se puede referir a la partitura en los compases: 3 primer tiempo ( $\Delta 5$ ), compás 7 primer tiempo ( $\Delta 13$ ), compás 10 primer tiempo ( $\Delta 19$ ) para hacer las comparaciones.

The image shows a musical score for 'Analogique A'. It consists of several staves. The first staff has a bracket with the letter 'E' underneath it, indicating a specific instance. Below it, there are more staves with various musical notations, including notes, rests, and dynamics like *ff* and *f*. The score is arranged in a multi-measure format.

Figura 4.11 Cuatro instancias de reja E en Analogique A. (Editions Salabert, 1968)

The image shows a musical score for 'Formalized Music'. It features nine staves for different instruments: Vln1, Vln2, Vln3, Ve1, Ve2, Ve3, Cb1, Cb2, and Cb3. The score is in 2/2 time. Each staff contains musical notation with notes, rests, and dynamics. Four instances of the letter 'E' are marked with a bracket across the staves, indicating specific occurrences in the music. Dynamics range from *ff* to *pp*.

Figura 4.12 Cuatro instancias de reja E, tal y como aparecen en Formalized Music.

Similarmente la reja F es presentada incompleta, en la primera fase estacionaria aparece en tres ocasiones con las siguientes características: IV aparece con una densidad de 3 en vez de 1 y ligeramente fuera de registro dos veces (en las otras fases aparece constantemente con densidad de 1, razón por la cual no modifiqué esta variable en la nueva reja propuesta), nuevamente III y II son omitidos y I con ligeras variaciones. En esta ocasión se omite una densidad de 6 compuestas de II + III teniendo ambas densidades de 3 y con una dinámica *pp*, considero que la omisión podría parcialmente obedecer las mismas razones que en la reja anterior.

Reja F ( $f_1$   $g_0$   $d_1$ ) propuesta por Xenakis:

<i>ff</i>				1 IV		
<i>f</i>			9 I			
<i>pp</i>			3 + 3 II + III			
	1 ( $4_1 - 4_0$ )	2 ( $2_2 - 4_1$ )	3 ( $1_3 - 2_2$ )	4 ( $0_4 - 1_3$ )	5 ( $11_4 - 0_4$ )	6 ( $9_5 - 11_4$ )

Reja F propuesta a raíz de análisis:

<i>ff</i>				3 IV		
<i>f</i>			9 I			
<i>pp</i>						
	1 ( $4_1 - 4_0$ )	2 ( $2_2 - 4_1$ )	3 ( $1_3 - 2_2$ )	4 ( $0_4 - 1_3$ )	5 ( $11_4 - 0_4$ )	6 ( $9_5 - 11_4$ )

En la partitura se pueden encontrar rejas F en los siguientes compases: 9 segundo tiempo ( $\Delta 18$ ), 13 primer y segundo tiempo ( $\Delta 25$  y  $\Delta 26$ ). A continuación se pueden observar estos ejemplos y cuatro ejemplos de la reja F generadas con la implementación, en la que se encuentran las discrepancias recién mencionadas.

The image displays two sets of musical notation. The left set consists of five staves, with the bottom three containing musical notation and dynamic markings such as 'ff'. The right set consists of six staves, with the top two containing musical notation and dynamic markings like 'ff', and the bottom three containing musical notation and dynamic markings like '(f)'. The notation includes various note values, rests, and dynamic markings.

Figura 4.13 Ejemplos de reja F en Analogique A. (Editions Salabert, 1968)

The musical score for Figure 4.14 is a string quartet in 2/2 time with a key signature of one sharp (F#). It consists of nine staves: Violin I (Vln1), Violin II (Vln2), Violin III (Vln3), Viola I (Vc1), Viola II (Vc2), Viola III (Vc3), Cello I (Cb1), Cello II (Cb2), and Cello III (Cb3). The score is divided into two systems. The first system contains the first four instances of the F fret, and the second system contains the remaining five. Dynamics include *ff*, *f*, and *pp*. The notation features various rhythmic patterns, including eighth and sixteenth notes, and rests, with some notes marked with articulation marks. Fingering numbers (3 and 5) are indicated for several notes.

Figure 4.14 Cuatro instancias de la reja F creadas con implementación.

Las rejas G que son presentadas también omiten parte de su construcción teorizada, apareciendo en tres ocasiones dentro de la primera fase estacionaria con las siguientes características: IV pequeñas variantes análogas a las ya expuestas con una densidad que gira alrededor del promedio de 9 en la región intermedia aguda y con dinámica *pp*, III y II son nuevamente omitidos debiendo de aparecer ambas con densidades de 1 y dinámica *f*, y por último I presenta en una ocasión una pequeña variación, por lo que giran alrededor de una densidad de 3 en el rango intermedio inferior y con una dinámica *pp*. Se puede constatar que son nuevamente omitidas los números romanos II y III y lo que es aún más importante que se invierte lo que sucede en las dos rejas precedentes E y F; en el sentido de que la porción que es

eliminada es ahora la que contiene dinámica *f* y se mantienen solamente los ataques con dinámicas *pp*.

Con esto se está estableciendo una correspondencia complementaria de esta reja con respecto a E, y como expondré en seguida, sucede lo mismo entre F y H. Esto refuerza aún más la noción de Di Scipio (2005) de que hay una lógica binaria subyacente en el pensamiento del compositor en esta obra. Quizá por razones intuitivas y técnicas basadas en el contacto de Xenakis con estudios psicoacústicos lo llevaron a efectuar las modificaciones en las dos rejas anteriores, y posteriormente estableció esta correspondencia, además de imprimirle una identidad aún más fuerte a cada una de estas, por ser opuestos.

Reja G ( $f_1 g_1 d_0$ ) propuesta por Xenakis:

<i>ff</i>						
<i>f</i>			1 + 1 II + III			
<i>pp</i>			3 I	9 IV		
	1 ( $4_1 - 4_0$ )	2 ( $2_2 - 4_1$ )	3 ( $1_3 - 2_2$ )	4 ( $0_4 - 1_3$ )	5 ( $11_4 - 0_4$ )	6 ( $9_5 - 11_4$ )

Reja G propuesta a raíz de análisis:

<i>ff</i>						
<i>f</i>						
<i>pp</i>			3 I	9 IV		
	1 ( $4_1 - 4_0$ )	2 ( $2_2 - 4_1$ )	3 ( $1_3 - 2_2$ )	4 ( $0_4 - 1_3$ )	5 ( $11_4 - 0_4$ )	6 ( $9_5 - 11_4$ )



Se pueden encontrar instancias de la reja G en los compases 1 segundo tiempo ( $\Delta 2$ ), 6 primer tiempo ( $\Delta 11$ ), 7 segundo tiempo ( $\Delta 14$ ), la figura 4.16 corresponde nuevamente a cuatro instancias generadas por la implementación, apegadas a como fueron propuestas por Xenakis.

Figura 4.15 Rejas G en Analogique A. (Editions Salabert, 1968)

The musical score consists of nine staves, grouped into three sections of three staves each. The top section (Violins) includes Vln1, Vln2, and Vln3. The middle section (Violas) includes Vc1, Vc2, and Vc3. The bottom section (Cellos) includes Cb1, Cb2, and Cb3. The score is written in 2/2 time with a key signature of one sharp (F#). Dynamics include *pp* (pianissimo) and *f* (forte). Fingerings (3 and 5) and slurs are used to indicate specific techniques and phrasing.

Figura 4.16 Cuatro instancias de reja G generadas por implementación.

Finalmente dentro de la octava y última reja H, que aparece 3 veces dentro de la primera fase estacionaria, se presentan las siguientes propiedades: IV abarca el registro intermedio superior con dinámica *f*, en esta primera fase estacionaria siempre aparece con densidad mayor de 1 que es la establecida; apareciendo con densidades 4, 2 y 2, teniendo un promedio de 2.6, más del doble de lo que debería ser. En la segunda fase estacionaria aparece una sola vez y con densidad 1; y en la tercera fase estacionaria 5 veces con una densidad promedio de 1.2 por lo que mantiene un promedio ligeramente por encima de 1 (1.6) en esta variable y no es modificada.

Los números romanos III y II vuelven a ser omitidos debiendo de abarcar una densidad de 3 cada una, 6 en total. I con dinámica  $f$  comparte el mismo registro que II y III pero con 9 de densidad y por lo general aparece completo o con pequeñas variantes. Con esta información se puede confirmar que H es una especie de inverso de F en cuanto a dinámicas (E y F siempre son  $ff$  y G y H siempre  $pp$ ), y al mismo tiempo es una especie de inverso de G en cuanto a densidades (G con 9 en el registro intermedio superior y 1 en el inferior, H con el inverso de esto). Se pueden encontrar ejemplos de la reja H en los compases: 4 segundo tiempo ( $\Delta 8$ ), 9 primer tiempo ( $\Delta 17$ ),

Reja H ( $f_1 g_1 d_1$ ) propuesta por Xenakis:

$ff$						
$f$			3 + 3 II + III			
$pp$			9 I	1 IV		
	1 ( $4_1 - 4_0$ )	2 ( $2_2 - 4_1$ )	3 ( $1_3 - 2_2$ )	4 ( $0_4 - 1_3$ )	5 ( $11_4 - 0_4$ )	6 ( $9_5 - 11_4$ )

Reja H propuesta a raíz de análisis:

$ff$						
$f$						
$pp$			9 I	1 IV		
	1 ( $4_1 - 4_0$ )	2 ( $2_2 - 4_1$ )	3 ( $1_3 - 2_2$ )	4 ( $0_4 - 1_3$ )	5 ( $11_4 - 0_4$ )	6 ( $9_5 - 11_4$ )

Figure 4.17 shows two columns of musical staves. The left column contains ten staves of music, with dynamics markings *pp* and *pp*<sub>1</sub>. The right column contains ten staves of music, with a *pp* marking. The notation includes various rhythmic values and articulations.

Figura 4.17 Ejemplos de reja H en Analogique A. (Editions Salabert, 1968)

Figure 4.18 is a musical score for a string quartet and three cellos. The score is in 2/2 time and consists of nine staves. The staves are labeled Vln1, Vln2, Vln3, Vc1, Vc2, Vc3, Cb1, Cb2, and Cb3. The music features various dynamics, including *pp* and *f*, and includes triplets and quintuplets.

Figura 4.18 Cuatro instancias de reja H generadas con implementación.

Dentro del resto de las fases estacionarias se observa un comportamiento muy similar al recién expuesto por lo que me parecería redundante exponer los resultados<sup>85</sup>, quizá valga la pena notar algunas excepciones, como por ejemplo el empleo de silencios al comienzo de la segunda fase estacionaria, en el compás 48 donde debería de ir  $\Delta 3$  y de nuevo en el compás 50 donde debería de ir  $\Delta 5$ . El silencio aparece nuevamente en el compás 54 con la diferencia que se escapa una altura  $7_0$  en el contrabajo 2 de la reja D precedente, y en el compás 61 sucede algo similar ahora en el violoncelo 3 se “retarda” una altura  $6_2$  proveniente de la reja G anterior.

Aunque el silencio no aparece implementado en su mecanismo, quizá sea otra manera de negarlo, en la primera parte de su teorización expone la posibilidad de una reja vacía y otra en la cual solamente aparece un solo gránulo. Por otro lado aparece silencio un compás después de una pausa total abrupta de tres segundos en el compás 47. Estos silencios solamente vuelven a ocurrir más adelante dentro de la perturbación  $P'_c$ , en la tercera fase estacionaria en el compás 137 y una última ocasión en el la perturbación  $P'_A$  en el compás 146.

Finalmente en cuanto al timbre, parece que Xenakis quisiera vigorizar la afirmación de identidad de las fases estacionarias al ser las únicas que presentan los nueve instrumentos arco ordinario, ya que en las perturbaciones se hacen cambios tímbricos con distintas combinaciones entre pizzicatos, el empleo de sordina, *frappé*, col legno, sul ponticello.

---

<sup>85</sup> El lector puede referirse al apéndice para consultar las tablas del análisis.

Las tres fases estacionarias cumplen satisfactoriamente las probabilidades establecidas, (se puede calcular en implementación de SuperCollider)

Fase estacionaria E compases 1 - 15

A	B	C	D	E	F	G	H
5	3	4	5	4	3	3	3
16.66%	10%	13.33%	16.66%	13.33%	10%	10%	10%

Fase estacionaria E compases 47 - 64

A	B	C	D	E	F	G	H
5	5	3	6	4	4	3	1
16.12 %	16.12 %	9.67 %	19.35 %	12.9 %	12.9 %	9.67 %	3.22 %

Fase estacionaria E<sup>86</sup> compases 125 - 140

A	B	C	D	E	F	G	H
6	2	3	5	6	2	2	5
19.35 %	6.45 %	9.67 %	16.12 %	19.35 %	6.45 %	6.45 %	16.12 %

Total promedio de las tres fases estacionarias. 92  $\Delta t$  en total

A	B	C	D	E	F	G	H
16	10	10	16	14	9	8	9
17.39%	10.87 %	10.87 %	17.39 %	15.22 %	9.78 %	8.7 %	9.78 %

Siendo las probabilidades de la fase estacionaria (Xenakis, 1992: 92):

---

<sup>86</sup> Aparentemente en la tercera fase estacionaria, en el compás 138 violoncelo 2 existe una errata de edición al no cambiar la clave de sol a clave de fa en el cambio de  $\Delta 26$  y  $\Delta 27$  (rejas A y H respectivamente), ya que de otra manera en H no corresponderían las alturas, y por otro lado en el siguiente compás vuelve a aparecer la clave de sol.

A	B	C	D	E	F	G	H
17.11	11.069	13.792	12.942	14.558	12.11	8.238	10.716

#### 4.4.2 FASE DE PERTURBACIÓN $P_A^0$ COMPASES 16 - 30

En la obra aparecen tres fases de perturbación  $P_x^0$  que es la perturbación más fuerte que se le puede aplicar al mecanismo (Ibíd., 92), ya que consiste en la reiteración de la misma reja. Corresponden a  $P_A^0$ ,  $P_C^0$  y  $P_B^0$  que posteriormente serán presentados en un estado más cercano a la zona de equilibrio<sup>87</sup>,  $P'_A$ ,  $P'_C$  y  $P'_B$ .

Entonces en la disposición de fases  $E \rightarrow P_A^0 \rightarrow P'_A$  se está afirmando el mecanismo, después colocándolo inmediatamente en una de sus negaciones más fuertes y en la siguiente perturbación lo acerca aunque ligeramente<sup>88</sup>.

A continuación expondré un análisis similar al de la fase estacionaria, detallando los datos excepcionales encontrados, ya que mucho de lo expuesto sobre las rejas en las fases estacionarias es aplicable para toda la obra, como las variantes de densidad que giran alrededor de un promedio, y las rejas modificadas de E, F, G y H.

La perturbación  $P_A^0$  consiste entonces en la repetición de la reja A a lo largo de toda la fase. Se presenta la reja A con pequeñas variaciones como las expuestas anteriormente, en las que la densidad puede ser ligeramente mayor o menor, también permanece el hecho de darle más importancia a los registros extremos, que

---

<sup>87</sup> La fase estacionaria es conseguida en el cuarto estado, “we notice that by the fourth stage we have almost attained the stationary state. Consequently the mechanism we have built shows a very rapid abatement of the oscillations, and a very great convergence towards final stability, the goal (stochos) [...]” (Xenakis, 1992: 92)

<sup>88</sup> El lector se puede referir al final de la sección 4.3 para apreciar el esquema de la forma general.

notamos como un aspecto característico de esta reja en particular, registro extremo agudo con máxima densidad y *ff*. Por lo que es de práctica común en esta fase que se omitan III, registro inmediato inferior de IV, ó II registro inmediato superior de I, solamente se presentan juntos en el compás 25. Incluso en una ocasión omite las dos, quedándose solamente con los registros extremos (Compás 20  $\Delta$ 9). Solamente en una ocasión la densidad en IV es presentada notoriamente escasa, justo después del ejemplo recién mencionado, en el compás 20  $\Delta$ 10.

En el compás 24  $\Delta$ 18 se omite por completo I y en el  $\Delta$ 17 hay un exceso de alturas en esta misma región, ¿quizá como una especie de compensación?

Los instrumentos que tocarán la mayoría de las notas de IV que tiene mayor densidad y en dinámica *ff* llevan sordina, mientras que las alturas que siguen en densidad dentro de IV serán casi siempre tocadas por los contrabajos con pizzicatos. Por último las alturas dentro de III y II serán generalmente tocadas por los violoncelos con arco ordinario, estas que tienen el mismo timbre que en la fase estacionaria son regularmente omitidas, quizá por esta razón, buscando reforzar su negación.

#### 4.4.3 PERTURBACIÓN $P'_A$

Esta perturbación aparece dos veces a lo largo de la pieza, la primera corresponde a los compases 32 al 46 y la segunda al final de la pieza entre los compases 141 al 155. Dentro de ambas realmente no existen muchas excepciones en cuanto a la construcción de las rejas, salvo en los compases 42 y 146 en el cual se “retardan” alturas de rejas anteriores.



Lo que me parece importante en cuanto a la elección (siendo esta decisión libre) de esta perturbación por parte de Xenakis, es que se favorecen las rejillas D y H, D tiene densidad 9 en *pp* dentro de región extrema grave y H es una de las rejillas modificadas que consta solamente de la dinámica *pp* abarcando el registro medio.

Por lo tanto si lo comparamos con la fase inmediatamente anterior  $P_A^0$  donde la densidad mayor es concentrada en el registro extremo agudo y en dinámica *ff*, se puede evidenciar el deseo de Xenakis de establecer un marcado contraste entre estas dos secciones. Por otro lado, con esto se refuerza aun más la noción de Di Scipio de la lógica binaria dentro de la concepción de esta obra.

Tímbicamente ambas presentan pizzicatos en violines, los cellos presentan sordina y los bajos sordina con posición de arco *sul ponticello*. Aquí también se puede evidenciar el deseo de establecer una correspondencia por oposición, al invertir el uso de pizzicatos y sordina.

Existen algunas diferencias con respecto a la relación del número de rejillas utilizadas y las probabilidades expresadas en la perturbación que se pueden apreciar en las siguientes tablas, no obstante, considero que las características fundamentales que consisten en una supremacía de las rejillas H y D se cumple.

Probabilidades del mecanismo con la perturbación  $P'_A$

A	B	C	D	E	F	G	H
2.1	8.4	8.4	33.6	1.9	7.6	7.6	30.4

$P'_A$  conteo de rejas entre compases 32 - 46

A	B	C	D	E	F	G	H
1	1	1	10	1	3	2	11
3.33%	3.33%	3.33%	33.33%	3.33%	10%	6.66%	36.66%

$P'_A$  conteo de rejas entre compases 141 - 155

Porcentaje sobre un total de 28 (por  $\Delta 23$  que solamente presenta dos alturas, que pueden formar parte de la reja anterior y por  $\Delta 12$  que es silencio)<sup>89</sup>

(Total de 28)

A	B	C	D	E	F	G	H
0	4	2	12	0	3	1	6
0%	14.285%	7.142%	42.857%	0%	10.714%	3.571%	21.428%

Si se cuenta  $\Delta 23$  como parte de H (total de 29)

A	B	C	D	E	F	G	H
0	4	2	12	0	3	1	7
0%	13.793%	6.896%	41.379%	0%	10.344%	3.448%	24.137%

#### 4.4.4 FASE $P'_C$ COMPASES 65 -80

Después de  $P'_A$  retorna a la fase estacionaria que ya fue expuesta, posteriormente apareciendo la perturbación  $P'_C$ .

En lo que llevo hasta aquí expuesto de la forma global, se puede apreciar que primero se afirma la *identidad* del mecanismo E, acto seguido rompe abruptamente con esta identidad pasando a una de las perturbaciones más fuertes  $P'_A$ , luego se acerca a la zona estacionaria mediante  $P'_A$  para finalmente regresar a E. Con  $P'_C$

---

<sup>89</sup> Dos maneras distintas de realizar el conteo de rejas en esta fase, presentan una ligerísima variación.

parece ser que se aleja “por pasos” (distinto a lo que sucede en el primer contraste de E con  $P_A^0$ ) para acto seguido llegar a una perturbación más fuerte  $P_C^0$ .

En esta perturbación hay pocas excepciones en cuanto a la construcción de rejas, curiosamente caen en tres ocasiones sobre la reja B, siendo la última la más notoria en el compás 78  $\Delta$  27 en donde omite las regiones III y IV, siendo IV característica fundamental. Las demás son similares a las ya expuestas y consideraría redundante exponerlas para mejor pasar a otros aspectos de esta sección.

Dentro de las probabilidades de esta perturbación, se favorece la tendencia de que se escojan las rejas E y F, ambas formando parte de las rejas que fueron modificadas, consistiendo solamente de ataques *f* y *ff* en el registro medio.

Tímbicamente es contrastado con la fase estacionaria con pizzicatos en los violines, cellos serán arco ordinario igual que en E, y aparecen en los contrabajos *frappé* col legno, por lo tanto tendiendo a ser percusivo en los extremos.

Probabilidades del mecanismo con la perturbación  $P_C'$

A	B	C	D	E	F	G	H
8.4	7.6	2.1	1.9	33.6	30.4	8.4	7.6

$P_C'$  conteo de rejas entre compases 65 - 80

A	B	C	D	E	F	G	H
2	3	0	0	9	12	0	3
6.896%	10.344%	0%	0%	31.034%	41.379%	0%	10.344%

#### 4.4.5 FASE $P_C^0$ COMPASES 80 – 95

Después de alejarse de la fase estacionaria mediante  $P_C'$  se aleja aún más con una perturbación  $P_C^0$ , que consiste de la reiteración de las rejas C. Esta reja tiene la mayor parte de su densidad en el registro agudo con dinámica *pp*. Aquí podemos encontrar un par de correspondencias: por un lado es la “inversión dinámica” de la primera perturbación del tipo  $P_x^0$  que ocurre en la pieza, correspondiendo al de la reja A.

Por otro lado está haciendo un contraste con respecto a la fase inmediata precedente, ya que la perturbación anterior favorecía las rejas compuestas solamente con ataques *f* o *ff* en los registros intermedios, y en la reja C se tiene una mayor densidad en los registros extremos, sobre todo en el agudo, ambos con dinámicas *pp*.

En cuanto a la construcción de las rejas, se hacen omisiones notables del contenido de la reja repetidamente. Generalmente consisten en una disminución de densidad en la región IV en valores que oscilan entre 4 y 6, cuando sucede esto es común que se omita el resto de la reja, por lo tanto sigue predominando esta característica aún y en estas instancias.

Es notoria la frecuencia con la cual hace estas modificaciones, 8 veces presenta la reja incompleta solamente incluyendo la porción aguda, lo cual corresponde a un 26.66% de la totalidad de la perturbación. Se pueden ver ejemplos de esto desde el inicio en los compases 80, 81, 83, 85, 86, entre otros. El otro tipo de variante que hace en esta perturbación ocurre mucho menos, con un total de 3 ocasiones,

consiste en nuevamente disminuir la densidad establecida pero con la porción inferior de alturas presente. Quizá haga esto para generar puntos de descanso y no sobre saturar, ya que en la fase precedente había mucha actividad con dinámicas *f* y *ff* en la región intermedia, por otro lado para generar una ligera variabilidad en la que la cual sin embargo, la característica fundamental de la reja no es omitida.

En esta ocasión los violines y cellos aparecen con sordina, y los contrabajos con pizzicato, inversión de lo que ocurría en la fase anterior, en donde los violines eran quienes tenían pizzicato.

#### 4.4.6 FASE $P_B^0$ COMPASES 95 – 109

La última perturbación fuerte que aparece en la obra corresponde a  $P_B^0$ , siendo la máxima perturbación de tipo  $P_x^0$  alargada. En esta perturbación ocurre algo similar que en la precedente. Hay un número considerable de excepciones parecidas entre sí, que consisten principalmente en (nuevamente) una disminución de densidad en la porción que debería de tener una mayor aglomeración de elementos. Sin embargo sigue predominando por encima de los otros y globalmente se percibe la característica principal de la reja B que consiste en poseer una mayor densidad con ataques *f* en el registro grave. El número de veces que ocurren reducciones notorias es de 10, correspondiendo a un 33.33% del total de  $\Delta t$ .

Además es notoria una reja de solamente dos elementos, una altura de la región IV (en vez de 9) y una altura en la región I en el compás 99, después de la cual ocurren *ff* en la región IV en los contrabajos lo cual no es característica de la reja B. Es difícil especular la razón por la cual habrá hecho estas modificaciones, más que a

decisiones compositivas intuitivas como la expuesta en la perturbación anterior, tener mayor variedad con la economía del material en cuestión, sin perder las características distintivas de B, a la vez que se generan puntos de descanso para el oyente.

Por otro lado vuelve a existir una oposición entre fases, ya que en la fase precedente se tiene la característica de predominar los agudos en *pp* mientras que en esta predominan los graves con dinámica *f*.

#### 4.4.7 FASE $P'_B$ COMPASES 110-124

En la última perturbación que abordaré (Ya se expuso el  $P'_A$  del final de la obra y la fase estacionaria que sucede después de  $P'_B$ ) que precede la llegada de la fase estacionaria, se tiene una tendencia a escoger las rejas A y C, correspondiendo nuevamente a dos rejas complementarias entre sí. Tienen estructuras similares pero sus dinámicas son invertidas. Además las rejas a las que se les está dando predilección son las que habían sido utilizadas para las perturbaciones del tipo  $P^0$  (además de  $P^0_B$  la fase inmediata anterior a esta) por lo que es altamente probable que esta sea la razón por la que escogió esta perturbación particular.

Dentro de esta fase ocurren algunas excepciones notables, por ejemplo, aparece una dinámica *p* a lo largo de la fase, que no había sido considerada como parte de las rejas. Por otro lado Xenakis aparenta haber hecho operación de unión entre dos rejas, notablemente en el compás 112  $\Delta 5$ , uniendo C y E dentro de un mismo  $\Delta t$ .

(C	$f_0$	$g_1$	$d_0$ )
E	$f_1$	$g_0$	$d_0$ )

Region Frec.	Frecuencias	Densidad	Dinámica
6 (9 <sub>5</sub> - 11 <sub>4</sub> )	(4 <sub>5</sub> 6 <sub>5</sub> 3 <sub>5</sub> 0 <sub>5</sub> 7 <sub>5</sub> 8 <sub>5</sub> 2 <sub>5</sub> 9 <sub>5</sub> 1 <sub>5</sub> )	(9)	(pp)
5 (11 <sub>4</sub> - 0 <sub>4</sub> )	(6 <sub>4</sub> )	(1)	(f)
4 (0 <sub>4</sub> - 1 <sub>3</sub> )	5 <sub>3</sub> 8 <sub>3</sub> 2 <sub>3</sub> 4 <sub>3</sub>	4	ff
3 (1 <sub>3</sub> - 2 <sub>2</sub> )			
2 (2 <sub>2</sub> - 4 <sub>1</sub> )	5 <sub>2</sub> 3 <sub>2</sub> 4 <sub>2</sub>	3	f
1 (4 <sub>1</sub> - 4 <sub>0</sub> )	(7 <sub>0</sub> 5 <sub>0</sub> 6 <sub>0</sub> )	(3)	(pp)

Finalmente por primera vez en la obra todos los instrumentos están tocando con técnica *frappé col legno*, quizá buscando crear mayor tensión previa de la llegada de la última fase estacionaria<sup>90</sup>, dentro de la cual todos tocarán arco *ordinario*.

Probabilidades para  $P'_B$ :

A	B	C	D	E	F	G	H
35.7	8.9	32.3	8.1	6.3	1.6	5.7	1.4

Como se incluyeron dos rejillas en un mismo  $\Delta t$ , se calcularon las probabilidades para  $31 \Delta t$

A	B	C	D	E	F	G	H
12	1	10	3	2	0	2	1
38.709%	3.22%	32.258%	9.677%	6.451%	0%	6.451%	3.22%

$30\Delta t$

A	B	C	D	E	F	G	H
12	1	10	3	2	0	2	1
40%	3.33%	33.33%	10%	6.66%	0%	6.66%	3.33%

<sup>90</sup> Me parece curioso poder hacer una analogía tonal con este mecanismo, parece como si la llegada de la fase estacionaria fuese la llegada a la tónica, y nos encontraríamos en una dominante (Perturbación).

## 4.5 FORMA GENERAL

Hasta este punto he expuesto todas las peculiaridades de las rejillas empleadas, la proposición y comprobación de la hipótesis de que Xenakis utiliza 4 rejillas alternativas entre E, F, G y H que emplea debido posiblemente a razones psicoacústicas y lo que considero más importante, procesos dialécticos entre formalización e intuición dentro del proceso creativo. Además considero haber expuesto algunas relaciones binarias o complementarias, sustentando con datos empíricos la hipótesis de Di Scipio de existir una lógica binaria subyacente y también por otro lado elucidar con estos mismos datos la dialéctica entre formalización e intuición presentes en esta obra, a la que Di Scipio se refiere como una de las características más entrañables de esta pieza:

This music is not the audible trace of a thoroughly formalistic approach, but the result of a clash between different domains of rationality. It entailed not only a substantial body of theoretical premises, but also important manual, non-formalized adjustments and arbitrary choices [...] a work expressive of a lively and intricate dialectic between formalization and intuition. (Di Scipio, 2005: 2-3)

En esta sección procederé a resumir la forma general de la obra con los datos recién expuestos en el apartado precedente .

$$E \longrightarrow P_A^0 \longrightarrow P_A' \longrightarrow E \longrightarrow P_C' \longrightarrow P_C^0 \longrightarrow P_B^0 \longrightarrow P_B' \longrightarrow E \longrightarrow P_A'$$

Xenakis afirma que la decisión de las fases estacionarias y las perturbaciones fueron de libre elección, a continuación expondré las razones por las que considero que escogió estas rejillas.



Comienza con la fase estacionaria, como era de esperarse para afirmar la *identidad* del mecanismo, y como se comprobó en efecto emplea las probabilidades expuestas. En cuanto a la construcción de las rejas A, B, C y D ocurren variantes que giran alrededor de una media, no obstante siempre conservando sus principales características.

Posteriormente escoge la perturbación  $P_A^0$ , siendo del tipo  $P_x^0$  que es la perturbación más fuerte del sistema, ya que es la reiteración de una sola reja negando todas las probabilidades del mecanismo, como constata en su deseo de negar una entidad para afirmar su existencia.

La perturbación consiste de rejas A, la cual tiene como característica fundamental una mayor densidad en el registro extremo agudo en *ff* y en segundo lugar una densidad menor en el extremo grave con dinámica *f*. A raíz del análisis pude observar que son estas dos propiedades de las rejas a las que les da prioridad, ya que omite constantemente las regiones intermedias, correspondiendo a los números romanos I y II. Es peculiar el empleo de sordina en el rango donde pide *ff*, ¿quizá pensando que opacarían demasiado los pizzicatos de los contrabajos de mucha menor densidad? Además, el registro que presenta el timbre de la fase estacionaria es la menos empleada, por lo que es evidente que se trata de una negación redundante del mecanismo.

Después presenta la perturbación  $P'_A$ , una perturbación de menor fuerza acercándose ligeramente a las probabilidades del mecanismo, hay que recordar que el estado estacionario es conseguido entre las terceras y cuartas fases de la

perturbaciones. Por lo que primero niega bruscamente la fase estacionaria para acto seguido acercarse un poco a él antes de su arribo. Las probabilidades de la perturbación en turno favorecen ampliamente las rejas D y H, siendo H una de las rejas que fueron modificadas excluyendo la porción II y III, lo cual se traduce a que solamente consta de dinámicas *pp* dentro de un registro medio, mientras que la reja D tiene mayor densidad en el registro grave y con dinámica *pp*.

Aquí se puede observar algo importante, por un lado la modificación hecha a las rejas H (que las dotan de otra identidad) parece ser que ocurrió a raíz de un deseo de expresar una oposición entre esta fase y la precedente, está presente nuevamente una lógica binaria en el sentido de que mientras en A hay mayor densidad en agudos con *ff*, en las rejas D favorece ampliamente el registro grave en *pp* y H consta solamente de alturas con dinámica *pp* en el registro medio. D aparece 33.33% del tiempo y H 36.66%, entre ambas formando casi un 70% de las rejas utilizadas. Finalmente, parece ocurrir un intercambio tímbrico entre los registros extremos de ambas perturbaciones, los pizzicatos que estaban presentes en los graves (de menor densidad) pasan a los agudos (de menor densidad), mientras que el empleo de la sordina también ocurre como intercambio, pero ahora con posición de arco Sul Pont.

Posteriormente regresa a la fase estacionaria E, retornando a la identidad previamente antagonizada, apareciendo de nuevo los arco ordinario, que al parecer también forma parte de la identidad de la fase E, ya que en sus tres instancias es empleada la misma fórmula tímbrica.

Después de haber arribado a la fase estacionaria, se vuelve a alejar pero ahora de manera menos “violenta” que en un inicio (al menos teóricamente), por medio de la perturbación  $P'_C$ . Esta perturbación refuerza la hipótesis de la modificación deliberada de la reja para fines de establecer una lógica binaria de oposición, dado que favorece ampliamente las rejas E y F ocurriendo entre ambas un total de 72.413% del total de las rejas empleadas, siendo estas rejas un fuerte contraste dinámico entre la perturbación  $P'_A$  que precedía la fase estacionaria y como veremos con la perturbación que le sigue.

Más adelante sucede la perturbación  $P_C^0$  con esto alejándose aún más de la fase estacionaria. La reiteración de la reja C significa que se enfocará principalmente en la región aguda con dinámicas *pp* luego en segundo lugar de densidad en la región grave igualmente en *pp*. Además, coloca la sordina en los violines y cellos mientras que los contrabajos pasan a hacer pizzicatos, atenuando aún más la intensidad de esta fase y oponiéndose a los frappé col legno de los contrabajos. Finalmente, disminuye reiteradamente las densidades correspondientes.

Posteriormente alarga una perturbación fuerte con  $P_B^0$ , ocurriendo nuevamente una correspondencia de oposición, ya que se invierten las densidades, en esta ocasión la mayor densidad pasa a los bajos y con dinámica *f*, lo cual intensifica volviendo al arco, por otro lado vuelve a suceder un intercambio de *pizzicati* en los registros extremos.

Después de esta perturbación comienza a “acercarse” a la fase estacionaria con  $P'_B$  dentro del cual son ampliamente favorecidas las rejas A y C, ocurriendo entre ambas

un 68% del total de rejas. Aquí suceden varias cosas interesantes, por un lado se está favoreciendo las rejas que habían aparecido con las perturbaciones más fuertes, exceptuando B que es la fase inmediata precedente. Quizá con esto buscando reafirmar una jerarquía de rejas, a la vez que la oposición entre ambas, ya que son dinámicamente opuestas, las dos se enfocan principalmente en el registro agudo pero A con *ff* y B con *pp*, lo mismo sucede en la región grave con *f* y *pp* respectivamente, estableciendo una oposición dentro de una misma fase por primera vez. Por otro lado, aparecen por primera vez todos los instrumentos con *frappé col legno*, creando quizá mayor tensión previa a la llegada de la fase estacionaria, donde podría parecer que busca “reposo”. Finalmente, en algunas  $\Delta t$  Xenakis parece hacer operaciones de combinación de rejas, tal vez buscando distorsionar aún más su identidad, o crear “mayor conflicto interno” dentro de esta fase, antes de arribar a la fase estacionaria.

La fase estacionaria aparece por última vez después de la fase precedente que evidencia conflictos internos. Regresan todos los arcos y se expresa la *identidad* del mecanismo sin mayores irregularidades.

Posteriormente vuelve a alejarse por segunda ocasión con la perturbación  $P'_A$ , manifestándose de manera similar a la que ya se había presentado, incluyendo la combinatoria tímbrica, presentándose como una última negación de la identidad del mecanismo, que como afirma Xenakis, paradójicamente es a través de la negación que se reafirma la existencia de una entidad.

#### 4.6. HACIA UNA PRE-COMPOSICIÓN ASISTIDA POR ORDENADOR EN ANALOGIQUE A

Ya habiendo expuesto hasta aquí un análisis detallado de la formalización del mecanismo y de su aplicación, pasaré a abordar la obra a partir de los conceptos elaborados en el capítulo 2 de este trabajo y del contexto en el que fue compuesta, buscando vincular una obra pre-informática con conceptos de los nuevos medios y de los estudios de software, con lo que espero poder establecer y articular puntos de contacto entre características de la media tradicional y la nueva media, más específicamente, cómo y por qué considero que esta obra se inserta como una Pre Composición Asistida por Ordenador.

Los años que preceden las primeras exploraciones y esfuerzos compositivos de Xenakis comprenden una época sumamente turbulenta en la vida del compositor. Apenas unos años antes de sus primeros esfuerzos compositivos, en diciembre de 1944, formando parte de la resistencia griega contra la ocupación británica de su país había quedado desfigurado perdiendo el ojo izquierdo a raíz de una esquirla de un tanque británico, apenas sobreviviendo al incidente. Xenakis formó parte de esta resistencia a partir de 1941, un año después de que fue aceptado en el politécnico de Atenas para iniciar sus estudios de ingeniería civil y mismo año en que entraría el ejército italiano a ocupar la ciudad. Como parte de la resistencia griega participó en numerosas protestas y demostraciones contra la ocupación italiana, de los nazis y finalmente la ocupación británica. Desde luego que todas estas experiencias lo marcarían hondamente, las protestas serían reiteradamente mencionadas como ejemplos de masas, para la presente obra utiliza esta analogía para abordar los

conceptos de orden, desorden y entropía, que como vimos, es de primordial importancia para la concepción de esta obra. (para cita completa ver sección 4.2)

A raíz de sus actividades como activista dentro de la resistencia y en medio de la fuerte inestabilidad política en Grecia que atravesaba una guerra civil, es perseguido y forzado a huir de su país, con muchas dificultades y prácticamente sin dinero llega a París en 1947 , posteriormente es condenado a muerte *in absentia* juzgado como terrorista por la administración política del país por el que había luchado.

Este año en que llega a París, con la ayuda de unos antiguos amigos del politécnico consigue trabajo en el despacho del famoso arquitecto Le Corbusier, en el cual en un inicio se desempeñaría como ingeniero calculando resistencia de materiales. El contacto con Le Corbusier definitivamente tuvo una fuerte impresión sobre Xenakis, quienes a pesar de que tuvieron una difícil relación, que eventualmente llevara a una querrela entre ambos, se tenían un respeto mutuo, reconociendo el compositor que: “[Le Corbusier] opened my eyes, he was the one that taught me to think in architectural terms, and he directed my attention towards contemporary architecture.” (Varga, 1996: 21)

Dentro de los primeros esfuerzos compositivos de Xenakis entre los años de los 1947 y 1952, tiempo en el que asistía a cursos de composición, trabajaba en el despacho de Le Corbusier y componía en su tiempo libre, se podían encontrar marcadas influencias Bartokianas, una especie de música “modal” en la que buscaba constatar y reafirmar sus raíces griegas, sobre lo cual en entrevista menciona:

In my loneliness and isolation I tried to hang on to something—after all, my old life and new circumstances, my old image of the world and the new experiences, all these were in conflict. I wanted to find out who I really was. In that process, traditional Greek folk music appeared to be a safe point, and I was also interested in Greek church music, as well Romania and other nations [...] (Ibíd., 26)

Más adelante afirma un profundo sentimiento de culpa y tristeza por haber dejado el país por el cual había peleado y casi perdido la vida, sintiendo que tenía una deuda con sus amigos y compatriotas que habían sido encarcelados, asesinados o que se encontraba prófugos. “I felt that I had a mission, I had to do something important to regain the right to live” (Ibíd., 47) Por lo que se propuso una filosofía en busca de una “universalidad”:

[one can search universality] not through religion, not through emotions or tradition, but through the sciences [...] but even with that, one can get nowhere without general ideas, points of departure. Scientific thought is only a means with which to realize my ideas, which are not of scientific origin. These ideas are born of intuition, some kind of vision. None of this was clear for me then but I worked instinctively in this direction (Ídem.)

Con esto refiriéndose a los años de 1953-54, años en los que compondría lo que es considerado como su primer opus maduro y en la que encontraría su voz con su seminal obra *Metastasis*, dejando atrás sus trabajos tempranos más tradicionales.

Es de suma importancia en estos años el contacto con el compositor francés Olivier Messiaen, asistiendo a sus talleres de composición entre 1950 y 1953 ya que es una de las tantas fuerzas que lo empuja en esta dirección al impulsarlo a dejar los estudios musicales tradicionales de conservatorio y aprovechar su formación y visión como arquitecto para buscar su propia voz:

[Xenakis] est certainement l'un des hommes les plus extraordinaires que je connaisse. On a beaucoup parlé de notre première rencontre, et du fait que je lui avais conseillé de renoncer aux études musicales classiques. Ma position était peut-être folle pour un professeur du Conservatoire, mais le personnage que j'avais devant moi était un héros, ne ressemblant à aucun autre, et je n'ai fait que mon devoir. La suite a confirmé ce que ce premier mouvement m'avait fait pressentir (Messiaen citado en Solomos, 1996)

No es casualidad que apenas un año después de esto naciera *Metastasis* y con ello la voz inconfundible de Xenakis, además de que el compositor recordaría la anécdota repetidamente en diversas fuentes.

Es con esta actitud y en ese espíritu en el que comienza a componer *Analogique A*, en búsqueda de la universalidad mencionada, a través de las ciencias y un pensamiento científico guiado por la intuición.

A pesar de también ser abordadas *Achorriopsis (1956-57)* y aspectos de *Pithoprakta (1955-56)* dentro de “música estocástica libre” en *Formalized Music*; implementando por ejemplo la distribución de Poisson para calcular la probabilidad de que ocurra una densidad particular dentro de una nube de sonidos (Xenakis, 1992: 12), la distribución de Gauss para calcular la velocidad de *glissandi* en *Pithoprakta*, razonamiento que atribuye ser derivado de la teoría cinética de los gases de Boltzmann y Maxwell, (Ibíd., 15), a la vez que explora una pregunta fundamental para él en esos años, “¿Cuál es el mínimo de restricciones lógicas necesarias para la construcción de un proceso musical?” (Ibíd., 16, traducción propia). En *Analogique A et B* dedica un espacio considerablemente mayor para elaborar sobre los conceptos y formalización de una “música estocástica markoviana” basada en el mecanismo y



los procesos y conceptos expuestos en los apartados precedentes; cadenas de Markov, función de probabilidad de distribución uniforme, conceptos de orden y desorden, entropía, y el abarcar el sonido a partir de su concepción cuántica. Por lo que Xenakis comienza a consolidar esta búsqueda de una universalidad a través de las ciencias y el razonamiento científico, de manera exhaustiva y evocando su necesidad de una “*originalité absolue*”.

Esta obra generalmente es concebida como una de las menos logradas dentro del catálogo de Xenakis, Di Scipio hace mención de esto afirmando que :

[the piece] is often regarded as a very problematic composition. Some would say it is one of the least successful works ever composed by Xenakis. That is usually explained with the strong emphasis he put on theoretical and technical details, summing up to an overload of formalistic premises whose result seems to be musically rather poor. That opinion is also reinforced by a misunderstanding of Xenakis own dissatisfaction with the results he could achieve – which had primarily to do with the technological limitations he experienced in the realization. In short, for many *Analogique A et B* is little more than an unsatisfactory experiment. (Di Scipio, 2005: 2)

Además, refiriéndose a constataciones de Matossian, Solomos afirma que el resultado no es convincente (1996). El mismo Hermann Scherchen, a quien Xenakis se refería como “su padre confesor” : “It must be noted, on this point, that Hermann Scherchen disliked this work [...] Harley suggests [...] that Xenakis in 1959 wrote *Syrmos*, dedicated to Scherchen in an attempt to create a more convincing music based on the same grounds as *Analogique A*. Harley adds that *Syrmos* ‘is much more engaging to listen to’ ” (Ídem.)

Sin embargo como mencioné al inicio del apartado anterior, concuerdo con Di Scipio en que esta obra realmente es una joya dentro de la literatura de la música contemporánea. Al ser un gran ejemplo de una auténtica lucha entre distintas inteligencias, que dan testimonio de la dialéctica entre formalización e intuición que subyace de manera intensificada en el trabajo de todo creador (y prácticamente dentro de cualquier experiencia humana).

Por otro lado, en esta época es muy importante su contacto con las indagaciones en la *musique concrète* hechas por Pierre Schaeffer y otros en los estudios de la entonces GRMC (ahora GRM *Groupe de Recherches Musicales*) donde fue admitido en 1955 y donde permanecería hasta 1962. Sobre esta experiencia afirma lo siguiente:

Another fruitful encounter for me: *musique concrete*. Contrary to serial music that repulsed me<sup>91</sup> — its aesthetics (that I cannot grasp), its futile exasperation at romantic music (which I actually love), its constructions (which I find terribly limited) — *musique concrete* immediately and profoundly attracted me. Thanks to this, I was able to anticipate a new world of musical possibilities that I was quick to exploit. (Kannach, 2008: xviii)

A raíz de su contacto con la música concreta y de las postulaciones teóricas de Dennis Gabor en la teoría de la información sobre una concepción cuántica del sonido, en estos años compone *Concret PH*, que como mencioné anteriormente es generalmente reconocida como el primer acercamiento a la síntesis granular, en los que se pueden escuchar los crujidos de un tiza ardiendo. Ya con esto en *Analogique A*, una especie de música granular instrumental (en *Analogique B* ya propiamente

---

<sup>91</sup> Xenakis ya había expuesto en “*La crise de la musique sérielle*” su antagonismo con respecto a los ideales y estéticas de la música serial, que en esos años era dominante. Publicado en la revista *Gravesaner Blätter* en 1955.

granular) se establecen intersecciones entre la fascinación de Xenakis por la naturaleza y del mundo sonoro que se encuentra en él, como también de la formalización que considera que lo acerca hacia una universalidad a través de las ciencias.

Partiendo de este breve repaso que permite adentrarnos en el contexto en el que Xenakis comenzaba a trabajar la obra y tomando en cuenta todo lo hasta aquí expuesto, es más fácil encontrar puntos de contacto entre los principios de la nueva media expuestos en el capítulo 2 y los métodos y conceptualizaciones del compositor, ya que hay fuertes puntos de contacto entre las *ciencias* computacionales y los métodos y teorías vinculadas a las construcciones de Xenakis, que se pueden elucidar auxiliados de los conceptos de Manovich.

Por ejemplo, se pueden encontrar muchas instancias y analogías con el primer principio o tendencia de los objetos de la nueva media que es la reducción a datos numéricos dentro de Analogique A. La concepción misma del gránulo o cuanto sonoro (como ya se mencionó influenciado por Gabor), dio lugar a su manipulación como unidades sonoras mínimas sujetas a las variables de frecuencia ' $f$ ' e intensidad ' $g$ ', y posteriormente una nube de gránulos a la variable de densidad. A su vez, estos gránulos fueron distribuidos dentro de una grilla rítmica cuantizada que consta también de un valor mínimo  $\Delta t$ , siendo en este caso las subdivisiones ya mencionadas. Además, estas variables que conforman el gránulo, fueron controladas por distintas probabilidades a través de un complejo mecanismo que pueden ser expresadas por matrices de probabilidades de transición, que son *necesariamente*

reducible a datos numéricos. Ya está siendo sujeto a manipulación matemática y puede ser (y es) descrito formalmente. Esta es una importante diferencia con respecto a los ejemplos expuestos en el capítulo 2, no se trata únicamente de una reducción a unidades discretas con las que se harán combinatorias y procesos de mapeo, sino que desde su génesis es reducible a datos numéricos y es sujeto a procesos matemáticos.

Por otro lado también presenta analogías con el segundo principio de los objetos de la nueva media expuesto, la modularidad. Un ejemplo evidente son las variables (claramente aquí también está presente el principio de variabilidad) que forman el gránulo sonoro, que a su vez pueden formar una cierta configuración específica de nube de gránulos, que forma una de las rejillas o pantallas. Dentro de estas mismas pantallas están interconectadas matrices de probabilidades de transición, donde están incluidas estructuras de las variables mencionadas. Todo esto es posibilitado gracias al primer principio que vimos, que permite esta manipulación. También se presenta modularidad en la homologación de la estructura y el mecanismo utilizado en *Analogique A* hacia *Analogique B*, con una modificación de sus variables (principio de variabilidad) pasa de una música instrumental granular markoviana a una música granular con ondas sinusoidales markoviana.

Además, Xenakis vislumbra emplear procedimientos similares aplicados a la espacialización multicanal, aunque solamente lo menciona. Aquí se puede observar el principio de modularidad en otro nivel, que se asemeja bastante a procedimientos usuales dentro de la CAO o de ambientes de programación dirigidos a artistas; se

puede imaginar la analogía de estar aplicando el mismo método (en el sentido de método de una clase) *.perturbacion(a,0)* para distintos fines y con distintos procedimientos de mapeo y escalabilidad, para síntesis, espacialización multicanal, composición algorítmica, o para lo que pueda ser de utilidad. Mas aún, se podría decir que utiliza distribuciones de probabilidad como si fueran métodos, extensiones, o implementaciones; por un lado él no escribió ni desarrolló las teorías ni fórmulas, y las personas que las desarrollaron (probablemente) no pensaron en su aplicación musical. En otras palabras, toma herramientas de diversos campos científicos, y las aplica en su mundo sonoro o *universo cerrado* que es la partitura, de manera similar todo esto es posible y muchas veces casi la norma en la CAO, aquí volvemos al punto de *bricolage* en los métodos de Xenakis (Solomos, 2005) y prácticas de *bricolage* en la programación por parte de artistas (McLean y Wiggins, 2010). (ver página 59)

En cuanto al tercer principio que es la automatización, ya en el capítulo 2 expuse algunos de los pensamientos de Xenakis sobre lo que la automatización representa en la creación musical (ver página 58) ya después de haber trabajado varias obras en su serie ST, que se realizaron con la ayuda de un ordenador, (Xenakis, 1992: 144) sin embargo, desde antes se puede apreciar su deseo de seguir desarrollando sus ideas en una computadora, en el preámbulo del capítulo III antes de abordar *Analogique A et B* menciona:

“richer and more complex stochastic mechanisms are highly interesting to construct and to put in work, but in view of the considerable volume of calculation which they

necessitate it would be useless to undertake them by hand, but *very desirable to program them for the computer*". (Ibíd., 79, mi énfasis)

La reducción a datos numéricos y la modularidad presentes en la estructuración de esta obra permite plantear una pseudoautomatización, todo esta puesta en escena para crear una automatización pero aún no se presenta una interacción con un "software performance [...] constructed by software in real time" (Manovich, 2013: 33), es decir, sigue siendo un "documento o texto estático" (Ibíd., 34). Lo que me parece muy importante, es que ya desde un principio se anhelaba y se vislumbraba implementarlo dentro de un lenguaje de programación, y ya desde su creación y su teorización es creada de tal manera en la que esto es facilitado, no se trata de potenciales conversiones y mapeos sino que la reducción a datos numéricos y su manipulación algorítmica ya está hecha y desarrollada.

La variabilidad que es el cuarto principio que Manovich le adjudica a la nueva media es evidente, desde la utilización de variables para la conformación y manipulación de gránulos o eventos sonoros, la utilización de una distribución de probabilidad uniforme para escoger determinada matriz de probabilidades de transiciones, las mismas matrices de probabilidades de transición, el empleo de métodos estocásticos ya expone la presencia de la variabilidad en esta obra. Ligado con la modularidad del método empleado, también varía el objeto sobre el cual actúa, como recién mencioné, sobre síntesis, instrumentos acústicos, o espacialización. Más aún, podemos encontrar otra forma de variabilidad en su manejo del concepto de entropía y de la afirmación de una *identidad* en la fase estacionaria del mecanismo,

siendo esta mismísima identidad una constitución variable que desde su formación involucra una serie de procesos estocásticos.

Los procesos de mapeo en esta obra ya involucran una transcodificación inherente, de gránulo a instrumento o a sinusoidal, de los datos numéricos que representan intensidad o frecuencia, que a su vez representan una nota en la partitura, de una pantalla o reja que constituye una configuración específica de una nube de gránulos, a un tiempo dentro de la partitura para nueve instrumentos de cuerda. Por otro lado me parece importante resaltar la transcodificación de conceptos en el imaginario de Xenakis, por ejemplo con las analogías del de orden o desorden en la naturaleza, en manifestaciones sociales, formalizadas a partir de todos los procedimientos empleados en la construcción del mecanismo y finalmente materializados en la obra.

Finalmente, como se expuso en el capítulo 2 aborda un algoritmo de fases fundamentales de una obra musical, en la que se pueden encontrar similitudes y analogías con los procesos de composición o artísticos que involucran un ordenador, expresados en (McLean y Wiggins, 2010). (ver página 61)

Es por estas razones que considero que en la música estocástica de Xenakis se encuentran puntos de contacto con principios de la nueva media más fuertes, ya desde su misma concepción, teorización, abordaje y procesos posee similitudes con los objetos de la nueva media y CAO, adentrándose a lo que se puede considerar como una *pre composición asistida por ordenador* .

## CAPÍTULO 5

### COMENTARIOS FINALES Y FUTUROS PROYECTOS

*“Un livre ne commence ni ne finit : tout au plus fait-il semblant.”*

— Mallarmé

A lo largo de este trabajo he abordado la práctica de la composición asistida por ordenador en mi quehacer musical a partir de un marco de los estudios de software inserto en el modelo semiológico musical tripartito. Esto con la intención de poder aprehender cómo es que se adhería el medio utilizado ( e.g. el ordenador, el lenguaje de programación) en las obras que compuse para este trabajo de investigación (*IV, VI, Anamnesis, 27 de a 40 grados y Autopo(i)ética*), algunas cuestiones de su lógica subyacente y cómo considero que es articulada en alguno o los tres polos de la tripartición, auxiliado de conceptos propuestos en los estudios de software y de la nueva media. Buscando con esto, entre otras cosas, propiciar una reflexión crítica sobre la práctica de la CAO que estaba empleando y explorando, a través de una investigación artística. (López Cano y San Cristóbal, 2014)

Encontré que efectivamente sí pueden ser adheridos al hecho musical elementos, características y procesos inherentes a la nueva media, aunque quizá en un inicio no sea tan evidente, ya auxiliado de los marcos teóricos y llevando a cabo el proceso de creación, fue posible sacarlos a luz generando una mayor conciencia sobre el rol que toma el medio en el cual desarrollo mi trabajo y cómo es trastocada por el mismo. Además, en un futuro buscar explotar a profundidad las características que ofrece,



siendo estas características inherentes del ordenador, y a la vez siendo este un producto de la sociedad que le es contemporánea, de la cual formo parte.

Por otro lado propongo ejemplos concretos en el lenguaje de programación SuperCollider que son disponibles para descargar, donde se pueden encontrar algunas propuestas de soluciones a problemáticas técnicas de programación dentro de la CAO. Algunos ejemplos de esto pueden ser la creación y control de gradientes de probabilidades o máscaras de tendencia para establecer determinada idea musical, aplicación de máquinas de estados para controlar distintos atributos musicales, utilización de sistemas – L estocásticos para controlar una base de datos de articulaciones, funciones de probabilidad que determinan detonantes de eventos a manera de gradiente, la extensión de la clase Fomus<sup>92</sup> que incluye la posibilidad de generar partituras de ensambles, incluir articulaciones, la posibilidad de generar una obra en tiempo real como en el caso de *Autopo(i)ética* y la manera en que resolví esto, distintas maneras de automatizar diversos parámetros de síntesis para la parte de la intervención en la obra *Anamnesis*, entre otras.

Posteriormente hice un análisis detallado de la obra *Analogique A* del compositor *Iannis Xenakis* y desarrollé una implementación de su obra en el lenguaje de programación SuperCollider con el cual se puede hacer una versión estrictamente apegada a la teorización de la obra en su libro *Formalized Music*. Dentro del análisis encontré la manera en la que el compositor ya en praxis modificó detalles de su sistema con fines estéticos, proponiendo la hipótesis de que Xenakis había utilizado

---

<sup>92</sup> Originalmente creada por Bernardo Barros, y que amplié hacia Fomus2 con algunas extensiones de funcionalidad.

distintas pantallas en algunas secciones o que había hecho modificaciones significativas sobre algunas de estas. Posteriormente comprobé con datos empíricos que en efecto borró secciones en 4 de las 8 pantallas propuestas y argumenté las razones por las cuales hizo esto. Además con esto pude ahondar y extender la noción de una lógica binaria subyacente en la obra que propone Di Scipio en (Di Scipio, 2005).

Ya con la implementación que se escribió se puede ver que sin estas modificaciones no se hubieran logrado las oposiciones y contrastes que propuse en el capítulo 4 que son de fundamental importancia en la obra para el compositor y para la comprensión de la misma. Además, se posibilita un desarrollo posterior de la implementación, quedando potencialmente como una herramienta de creación de obras originales, lo cual después se puede buscar dejarlo lo suficientemente flexible para aplicar este mecanismo en un sinnúmero de maneras, como por ejemplo en un sistema multicanal (el cual vislumbraba Xenakis pero no desarrolla), dejando la posibilidad de ofrecer cualquier número de rejillas con cualquier tipo de distribución y funciones estocásticas dentro de esta. En fin, dejar una clase lista para ser abordada y apropiada por cualquiera y en cualquier contexto, partiendo de una de las obras que considero primordiales en el pensamiento Xenakiano y en la literatura de la música del siglo XX, al ser un brillante ejemplo de un auténtico choque de fuerzas creadoras humanas, entre formalización e intuición.

## 5.1 FUTUROS PROYECTOS

- Expandir la obra *Autopo(i)ética*, implementando algoritmos genéticos, dinamizando la base de datos generando nuevas formas posibles.
- Expandir la intervención de *Anamnesis* para que tenga mayor capacidad independiente, abarcando más elementos psicoacústicos para la toma de decisiones y la generación de distintos posibles discursos automatizados.
- Empaquetar la implementación de la obra *Analogique A* con las características recién expuestas, además de tener propiedades analíticas, agregando métodos para que arroje el resultado de la entropía de determinado mecanismo, de su fase estacionaria, entre otros.
- Desarrollar el potencial creativo de una clase que empaquete la implementación de *Analogique A*, desarrollar un sistema multicanal derivado que menciona pero no desarrolla Xenakis.

# APÉNDICE I

## PARTITURAS OBRAS ORIGINALES

- I. IV Miniatura para piano, flauta y clarinete bajo en Bb.
- II. VI Miniatura para piano, trombón, fagot y viola.
- III. Anamnesis para violín solo.
- IV. 27 de a 40 Grados para quinteto de cuerdas.
- V. Autopo(i)ética para cuarteto de cuerdas. (en apéndice de código fuente)

Jorge E. Gómez Elizondo

*IV*

(2011)

Miniatura para Flauta, Clarinete en Bb y Piano.

IV

Jorge E. Gomez Elizondo  
2011

Flute  $\text{♩} = 60$   
Bass Clarinet in Bb *mf*  
Piano *mf*

This system contains the musical notation for the first three staves. The Flute part is in the treble clef with a tempo marking of quarter note = 60. The Bass Clarinet part is in the bass clef. The Piano part is in the bass clef. The music features complex rhythmic patterns with many sixteenth and thirty-second notes, often beamed together. There are several dynamic markings including *mf* and *f*. The system is flanked by double bar lines with repeat dots.

Fl. *mf*  
B. Cl. *mp*  
Pano *f*

This system contains the musical notation for the next three staves. The Flute part is in the treble clef. The Bass Clarinet part is in the bass clef. The Piano part is in the bass clef. The music continues with complex rhythmic patterns and dynamic markings such as *mf*, *mp*, and *f*. The system is flanked by double bar lines with repeat dots.

Musical score for measures 1-12. The score is written for Flute I (Fl.), B. Clarinet (B. Cl.), and Piano (Pno.). The Flute I part begins with a dynamic marking of *mf* and includes a first ending bracket labeled [Z]. The B. Clarinet part features a complex rhythmic pattern with many sixteenth notes and rests, marked with *f* and *mp*. The Piano part provides a steady accompaniment with a mix of eighth and sixteenth notes, marked with *mf*. The key signature has one sharp (F#) and the time signature is 4/4.



Musical score for measures 13-24. The score continues for Flute I (Fl.), B. Clarinet (B. Cl.), and Piano (Pno.). The Flute I part has a dynamic marking of *f* and includes a first ending bracket labeled [Z]. The B. Clarinet part continues with its intricate rhythmic texture, marked with *f* and *mp*. The Piano part maintains its accompaniment, marked with *mf*. The key signature and time signature remain the same as in the previous system.



Musical score for measures 20-29. The score is written for Flute (Fl.), B. Clarinet (B. Cl.), and Piano (Pno.).

- Flute (Fl.):** Measures 20-21 are marked *pp*. Measures 22-29 are marked *mp*. The flute plays a melodic line with various articulations and dynamics.
- B. Clarinet (B. Cl.):** Measures 20-21 are marked *ff*. Measures 22-29 are marked *p*. The clarinet plays a melodic line with various articulations and dynamics.
- Piano (Pno.):** Measures 20-21 are marked *ff*. Measures 22-29 are marked *mp*. The piano accompaniment features complex rhythmic patterns and chords.

Measure numbers 20, 21, 22, 23, 24, 25, 26, 27, 28, and 29 are indicated at the beginning of their respective staves.

Musical score for measures 15-24. The score is written for Flute (Fl.), B. Clarinet (B. Cl.), and Piano (Pno.).

- Flute (Fl.):** Measures 15-24 are marked *pp*. The flute plays a melodic line with various articulations and dynamics.
- B. Clarinet (B. Cl.):** Measures 15-24 are marked *pp*. The clarinet plays a melodic line with various articulations and dynamics.
- Piano (Pno.):** Measures 15-24 are marked *pp*. The piano accompaniment features complex rhythmic patterns and chords.

Measure numbers 15, 16, 17, 18, 19, 20, 21, 22, 23, and 24 are indicated at the beginning of their respective staves.



30

Pno. *f* *dim*

B. Cl. *f* *mp subito*

Fl. *pp* *p subito*

31

Pno. *ritard.*

B. Cl. *ritard.*

Fl. *ritard.*

Jorge E. Gómez Elizondo

*VI*

(2012)

Miniatura para Fagot, Trombón tenor Piano y Viola.

# VI

Jorge E. Gomez Elizondo  
2012

The musical score is for a section titled "VI". It features six staves: Bassoon, Tenor Trombone, Piano, Viola, Trombone, and Violin. The score is written in 3/4 time with a key signature of one sharp (F#). The tempo is marked as  $J = 60$ . The score includes various dynamics such as *mf*, *f*, *sf*, *ff*, *mp*, and *pp*. There are also performance instructions like "sull pont." and "pizz.". The score is divided into measures with bar numbers 1183, 1184, 1185, 1186, 1187, 1188, 1189, 1190, 1191, 1192, 1193, 1194, 1195, 1196, 1197, 1198, 1199, 1200, 1201, 1202, 1203, 1204, 1205, 1206, 1207, 1208, 1209, 1210, 1211, 1212, 1213, 1214, 1215, 1216, 1217, 1218, 1219, 1220, 1221, 1222, 1223, 1224, 1225, 1226, 1227, 1228, 1229, 1230, 1231, 1232, 1233, 1234, 1235, 1236, 1237, 1238, 1239, 1240, 1241, 1242, 1243, 1244, 1245, 1246, 1247, 1248, 1249, 1250, 1251, 1252, 1253, 1254, 1255, 1256, 1257, 1258, 1259, 1260, 1261, 1262, 1263, 1264, 1265, 1266, 1267, 1268, 1269, 1270, 1271, 1272, 1273, 1274, 1275, 1276, 1277, 1278, 1279, 1280, 1281, 1282, 1283, 1284, 1285, 1286, 1287, 1288, 1289, 1290, 1291, 1292, 1293, 1294, 1295, 1296, 1297, 1298, 1299, 1300, 1301, 1302, 1303, 1304, 1305, 1306, 1307, 1308, 1309, 1310, 1311, 1312, 1313, 1314, 1315, 1316, 1317, 1318, 1319, 1320, 1321, 1322, 1323, 1324, 1325, 1326, 1327, 1328, 1329, 1330, 1331, 1332, 1333, 1334, 1335, 1336, 1337, 1338, 1339, 1340, 1341, 1342, 1343, 1344, 1345, 1346, 1347, 1348, 1349, 1350, 1351, 1352, 1353, 1354, 1355, 1356, 1357, 1358, 1359, 1360, 1361, 1362, 1363, 1364, 1365, 1366, 1367, 1368, 1369, 1370, 1371, 1372, 1373, 1374, 1375, 1376, 1377, 1378, 1379, 1380, 1381, 1382, 1383, 1384, 1385, 1386, 1387, 1388, 1389, 1390, 1391, 1392, 1393, 1394, 1395, 1396, 1397, 1398, 1399, 1400, 1401, 1402, 1403, 1404, 1405, 1406, 1407, 1408, 1409, 1410, 1411, 1412, 1413, 1414, 1415, 1416, 1417, 1418, 1419, 1420, 1421, 1422, 1423, 1424, 1425, 1426, 1427, 1428, 1429, 1430, 1431, 1432, 1433, 1434, 1435, 1436, 1437, 1438, 1439, 1440, 1441, 1442, 1443, 1444, 1445, 1446, 1447, 1448, 1449, 1450, 1451, 1452, 1453, 1454, 1455, 1456, 1457, 1458, 1459, 1460, 1461, 1462, 1463, 1464, 1465, 1466, 1467, 1468, 1469, 1470, 1471, 1472, 1473, 1474, 1475, 1476, 1477, 1478, 1479, 1480, 1481, 1482, 1483, 1484, 1485, 1486, 1487, 1488, 1489, 1490, 1491, 1492, 1493, 1494, 1495, 1496, 1497, 1498, 1499, 1500, 1501, 1502, 1503, 1504, 1505, 1506, 1507, 1508, 1509, 1510, 1511, 1512, 1513, 1514, 1515, 1516, 1517, 1518, 1519, 1520, 1521, 1522, 1523, 1524, 1525, 1526, 1527, 1528, 1529, 1530, 1531, 1532, 1533, 1534, 1535, 1536, 1537, 1538, 1539, 1540, 1541, 1542, 1543, 1544, 1545, 1546, 1547, 1548, 1549, 1550, 1551, 1552, 1553, 1554, 1555, 1556, 1557, 1558, 1559, 1560, 1561, 1562, 1563, 1564, 1565, 1566, 1567, 1568, 1569, 1570, 1571, 1572, 1573, 1574, 1575, 1576, 1577, 1578, 1579, 1580, 1581, 1582, 1583, 1584, 1585, 1586, 1587, 1588, 1589, 1590, 1591, 1592, 1593, 1594, 1595, 1596, 1597, 1598, 1599, 1600, 1601, 1602, 1603, 1604, 1605, 1606, 1607, 1608, 1609, 1610, 1611, 1612, 1613, 1614, 1615, 1616, 1617, 1618, 1619, 1620, 1621, 1622, 1623, 1624, 1625, 1626, 1627, 1628, 1629, 1630, 1631, 1632, 1633, 1634, 1635, 1636, 1637, 1638, 1639, 1640, 1641, 1642, 1643, 1644, 1645, 1646, 1647, 1648, 1649, 1650, 1651, 1652, 1653, 1654, 1655, 1656, 1657, 1658, 1659, 1660, 1661, 1662, 1663, 1664, 1665, 1666, 1667, 1668, 1669, 1670, 1671, 1672, 1673, 1674, 1675, 1676, 1677, 1678, 1679, 1680, 1681, 1682, 1683, 1684, 1685, 1686, 1687, 1688, 1689, 1690, 1691, 1692, 1693, 1694, 1695, 1696, 1697, 1698, 1699, 1700, 1701, 1702, 1703, 1704, 1705, 1706, 1707, 1708, 1709, 1710, 1711, 1712, 1713, 1714, 1715, 1716, 1717, 1718, 1719, 1720, 1721, 1722, 1723, 1724, 1725, 1726, 1727, 1728, 1729, 1730, 1731, 1732, 1733, 1734, 1735, 1736, 1737, 1738, 1739, 1740, 1741, 1742, 1743, 1744, 1745, 1746, 1747, 1748, 1749, 1750, 1751, 1752, 1753, 1754, 1755, 1756, 1757, 1758, 1759, 1760, 1761, 1762, 1763, 1764, 1765, 1766, 1767, 1768, 1769, 1770, 1771, 1772, 1773, 1774, 1775, 1776, 1777, 1778, 1779, 1780, 1781, 1782, 1783, 1784, 1785, 1786, 1787, 1788, 1789, 1790, 1791, 1792, 1793, 1794, 1795, 1796, 1797, 1798, 1799, 1800, 1801, 1802, 1803, 1804, 1805, 1806, 1807, 1808, 1809, 1810, 1811, 1812, 1813, 1814, 1815, 1816, 1817, 1818, 1819, 1820, 1821, 1822, 1823, 1824, 1825, 1826, 1827, 1828, 1829, 1830, 1831, 1832, 1833, 1834, 1835, 1836, 1837, 1838, 1839, 1840, 1841, 1842, 1843, 1844, 1845, 1846, 1847, 1848, 1849, 1850, 1851, 1852, 1853, 1854, 1855, 1856, 1857, 1858, 1859, 1860, 1861, 1862, 1863, 1864, 1865, 1866, 1867, 1868, 1869, 1870, 1871, 1872, 1873, 1874, 1875, 1876, 1877, 1878, 1879, 1880, 1881, 1882, 1883, 1884, 1885, 1886, 1887, 1888, 1889, 1890, 1891, 1892, 1893, 1894, 1895, 1896, 1897, 1898, 1899, 1900, 1901, 1902, 1903, 1904, 1905, 1906, 1907, 1908, 1909, 1910, 1911, 1912, 1913, 1914, 1915, 1916, 1917, 1918, 1919, 1920, 1921, 1922, 1923, 1924, 1925, 1926, 1927, 1928, 1929, 1930, 1931, 1932, 1933, 1934, 1935, 1936, 1937, 1938, 1939, 1940, 1941, 1942, 1943, 1944, 1945, 1946, 1947, 1948, 1949, 1950, 1951, 1952, 1953, 1954, 1955, 1956, 1957, 1958, 1959, 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, 1971, 1972, 1973, 1974, 1975, 1976, 1977, 1978, 1979, 1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012.

This page contains a musical score for Trombones (Tbn.), Pianos (Pno.), and Violins (Vln.). The score is divided into two systems, labeled 1 and 2. Each system includes staves for the Trombone section (Tbn.), the Piano section (Pno.), and the Violin section (Vln.). The notation is complex, featuring various musical symbols such as notes, rests, and dynamic markings. The dynamic markings include *f* (forte), *pp* (pianissimo), *mf* (mezzo-forte), and *ff* (fortissimo). The score is written in a key signature of one sharp (F#) and a time signature of 2/4. The Trombone part is in the bass clef, the Piano part is in the treble clef, and the Violin part is in the treble clef. The score is densely packed with musical notation, including many accidentals and dynamic markings.

This page contains a musical score for Trombones (Bbn.), Percussion (Perc.), and Violins (Vln.). The score is divided into two systems, separated by a double bar line. The top system includes measures 219-222, and the bottom system includes measures 223-226. The Trombone part features complex rhythmic patterns with various dynamics such as *mp*, *f*, *ff*, and *pp*. The Percussion part includes a section marked "cymbal" and features a prominent *mf* dynamic. The Violin part is marked *mf* and includes a section marked *pp*. The score is written in a key signature of one sharp (F#) and a 4/4 time signature. The notation includes stems, beams, and various articulation marks.

4

The image shows a page of a musical score for four instruments: Bassoon (Bsn.), Trombone (Tbn.), Piano (Pno.), and Viola (Via.). The score is written in a single system with four staves. The key signature is one flat (B-flat major or D minor). The time signature is 4/4. The score includes various musical notations such as notes, rests, dynamics (mf, mp, f), and articulation marks. The Bassoon part starts with a *mf* dynamic and features several slurs and accents. The Trombone part also starts with a *mf* dynamic and includes slurs and accents. The Piano part is marked *mf* and features a complex texture with many notes and slurs. The Viola part is marked *mp* and includes slurs and accents. The score is numbered 4 at the top left.

Jorge E. Gómez Elizondo

## Anamnesis

Violín Solo  
(2013-14)

*a Juanmanuel Flores*

## Notas de Ejecución

- ♯ 1/4 de tono sostenido
- ♭ 1/4 de tono bemol
- ♯ 3/4 tono sostenido
- ♭ 3/4 tono bemol

No se pide la afinación exacta sino alturas ligeramente arriba o debajo del temperamento.

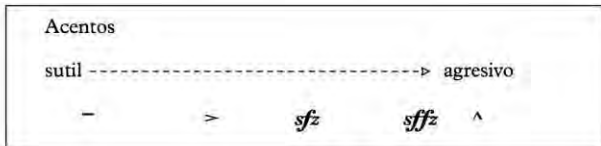
Las alteraciones solamente afectan a la nota a la que preceden y a las repeticiones inmediatas.

× Presión media en pisada, ahogando la cuerda buscando una sonoridad con poca o nula resonancia. Si aparece sobre una nota al aire, la cuerda será ahogada ligeramente después de la cejilla.

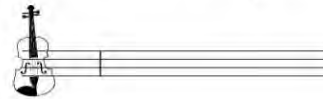
◆ Presión de armónico, aún y cuando no se encuentra en un nodo de armónico.

✕ trill    ◆ trill "Trinos" de presión de pisada. (ahogado - armónico, ordinario - armónico).

▲ Altura extremadamente aguda, lo más agudo posible en el instrumento con cuerda pisada.



Sistema de presión de arco



Sistema de posición de arco



Cada línea representa una cuerda.

Es común que en esta obra aparezcan tres sistemas a la vez, indicando presión de arco, posición de arco y el sistema tradicional para las alturas.

Posición compacta, por ejemplo al final de la página 3, se refiere a una posición en la que la mano izquierda, el arco y el puente están colindando, (la mano izquierda está casi en una posición sul pont, ahogando las cuatro cuerdas).

arco lento, granulando el sonido con presión



Arcada lentísima, exagerando granulación

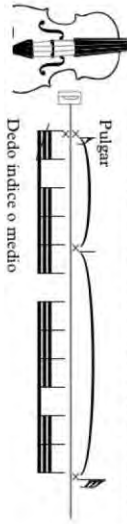


Ricochet, ricochets subsecuentes son ligados.



Trémolo de arco.





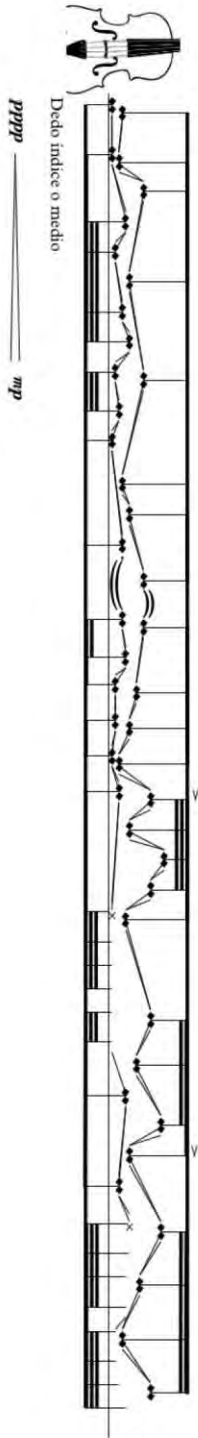
Mano izquierda hace un puente por encima del arco, cerca del puente, ahogado y desahogado la cuerda con el dedo índice o medio.

Bend <math>\sphericalangle</math> Aplastar la cuerda con dedo índice estrinándose sobre el arco (sin ahogar con pulgar), buscando cambiar la afinación con un timbre semi-ahogado y sucia.



Deslizándose con libertad siguiendo el contorno, pulgar, arco e índice o medio.

Pulgar





Violin and Oboe musical score with various performance markings and dynamics. The score is divided into five systems, each starting with a double bar line. The first system includes a *S.P.E.* marking and dynamics of *sfz*, *f*, *mp*, and *mf*. The second system features a *rit.* marking, *meno mosso*, and dynamics of *sfz*, *f*, *mp*, *mf*, and *p*. The third system includes a *S.P.E.* marking, *legno*, *crise*, and dynamics of *sfz*, *f*, *mp*, *mf*, and *p*. The fourth system has a *S.P.E.* marking, *accell.*, and dynamics of *f*, *sfz*, *p*, *mp*, *mf*, and *ff*. The fifth system includes a *rit.* marking, *meno mosso*, and dynamics of *sfz*, *mf*, *ff*, *mp*, *f*, and *pp*. The Oboe part is indicated by *Ond.* markings.

Violin (Vln.) and Oboe (Oboe) musical score. The score is divided into systems, with measures 80-85 and 86-90 clearly marked. Dynamics include *mf*, *p*, *f*, *sfz*, and *ff*. Performance instructions include *accel.*, *S.R. → S.P.E. Allargando liberamente.*, and *Avanzando hacia Posición compacta.* The score concludes with a *fff* dynamic marking.

Avanzando hacia Posición compacta.

S.R. → S.P.E. Allargando liberamente.

Finalis (Finis) Muz. Instrukc. 3-4<sup>o</sup> sero y parte esta otodada. Mani Instrukc. Programo 250000.



Vln.

meno flautato

11'

12'

flautato

pppp

90°

alla punta

pizzicato

Vln.

13'

14'

15'

accel.

bend

bend

bend

5x con bend cada vez más amplio

bend

W

bajo barrido con crine lo más agitado posible

bend oscilando libremente

mp

p

Imprevedibile simile, cada vez más agitado con mayor número de rasgados y dobles approx. 15''

siempre cruz.

Vln.

pp

Doblando arco hasta sobrepasar cepillo del V

cruzando cepillo instantaneamente, inclinando similit. al flautato, normal.

normal

Jorge E. Gómez Elizondo

*27 de a 40 Grados*  
(2013)

Para quinteto de cuerdas/  
For string quintet

## Notas de Ejecución

- ♯ 1/4 tono sostenido
- ♭ 1/4 tono bemol La altura exacta no es deseada, más bien afinación ligeramente más alta o baja.
- ♯ 3/4 tono sostenido Las alteraciones solamente afectan a la nota indicada o repeticiones inmediatas de la misma.
- ♭ 3/4 tono bemol
- × Ahogar cuerda con mano izquierda, si se indica sobre cuerda al aire se ahogará ligeramente después de la cejilla.

◆ ◊ Presión de armónico.

✕ trm ◊ trm Trinos de presión de mano izquierda.

▲ Registro extremo agudo.

Acentos

sútil -----> muy agresivo

— > *sfz* *sffz* ^

⌘ Ricochet, ricochets encadenados pueden ser ligados en un solo ataque.

⌘ Trémolo de arco.

\*Shorter values will have a tie instead of a —

continuo

Extremadamente poca presión  
Presión ordinaria  
Presión Extrema

discreto

Sistema de presión de arco.

arcoada sobre el tira cuerda.

tasto - tasto extremo  
ordinario

sul pont - sul pont extremo  
detrás del puente,  
inicio de tira cuerda.

Sistema de posición de arco.

IV

I

Detrás del puente (excepto en posición compacta), cada línea de la tablatura representa una cuerda.

La posición compacta (e.g. compás 104) se refiere a la posición donde la mano izquierda, el arco y el puente están colindando, (la mano izquierda esta prácticamente en una posición sul pont tocando las cuatro cuerdas).

Libremente detrás de puente (e.g. Cb. compás 117) Moviendo el arco ad libitum detrás del puente y sobre el tira cuerdas.

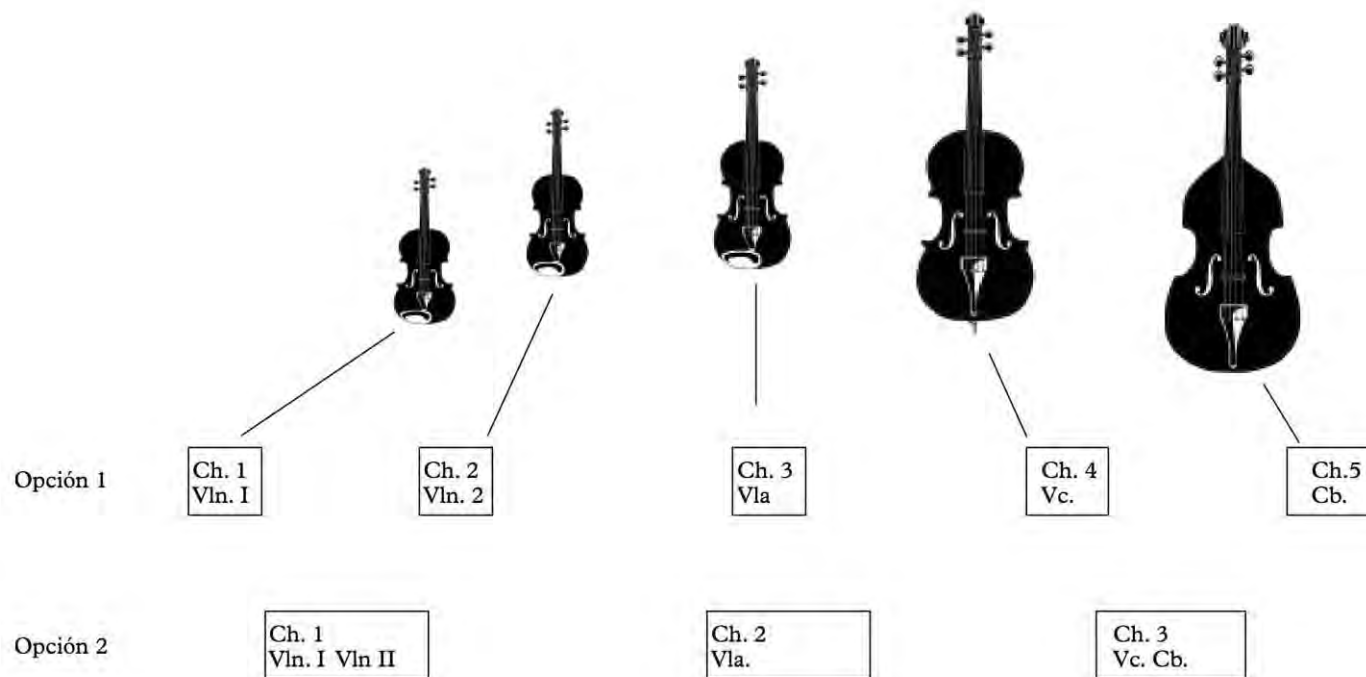
Accacchatura arpeggios, las alturas indicadas pueden ser vistas como guías, lo que es más importante es el gesto de arpeggio rápido.



## Indicaciones de amplificación

La amplificación es altamente deseable, con la excepción de ser ejecutada en un recinto pequeño.

Los instrumentos preferentemente serán pasados a canales independientes, con una espacialización que respete el acomodo original en el escenario.



27 de a 40 Grados

Violin I  $\text{♩} = 60$   $\frac{4}{4}$   $\frac{3}{4}$   $\frac{4}{4}$   $\frac{3}{4}$   $\frac{4}{4}$   $\frac{2}{4}$   $\frac{3}{4}$   $\frac{4}{4}$   $\frac{3}{4}$   $\frac{4}{4}$   $\frac{5}{4}$   $\frac{4}{4}$   $\text{♩} = 72$

Violin II

Viola

Violoncello

Contrabajo

Fuga E. Gomez Borrero  
1997-2011



This page contains the musical score for four string instruments: Violin I (Vln. I), Violin II (Vln. II), Viola (Vla.), and Cello (Cb.). The score is written in a standard musical notation with five staves per instrument. The music is characterized by a complex, rhythmic texture with frequent sixteenth and thirty-second notes. Dynamic markings such as *pp*, *f*, *mf*, and *ff* are used throughout. Performance instructions like *orditura* and *rit.* are present. The score is divided into measures by vertical bar lines, and some sections are marked with repeat signs. The overall style is that of a classical or romantic-era orchestral score.



This page of a musical score contains measures 97 through 100 for four string parts: Violin I, Violin II, Viola, and Cello/Double Bass. The score is written in standard musical notation with various dynamics and performance markings.

- Violin I (Vln. I):** Starts with a dynamic of *ff* and a marking of *ritardando per il tasto*. It features a melodic line with several slurs and accents.
- Violin II (Vln. II):** Starts with a dynamic of *f* and a marking of *ritardando*. It has a melodic line with slurs and accents, including a *fz* marking.
- Viola (Vla.):** Starts with a dynamic of *f* and a marking of *ritardando*. It features a melodic line with slurs and accents, including a *fz* marking.
- Cello/Double Bass (Cb.):** Starts with a dynamic of *f* and a marking of *ritardando*. It has a melodic line with slurs and accents, including a *fz* marking.

Additional markings include *ritardando* at the end of the section and *ritardando* at the beginning of the section.

This page contains a musical score for a string ensemble, including Violins I, Violins II, Viola, and Cello/Double Bass. The score is written in a single system with five staves. The top staff is for Violin I, the second for Violin II, the third for Viola, and the bottom two for Cello and Double Bass. The music is in a 2/4 time signature and features a variety of dynamics including *pp*, *f*, *ff*, *mf*, and *ppp*. There are several dynamic markings such as *simile ordinario*, *ritardando*, and *dim.* throughout the piece. The score includes various musical notations such as slurs, accents, and phrasing slurs. A rehearsal mark '15' is located at the end of the system.

This page contains a musical score for measures 121 through 127. The instruments are Violin I (Vln. I), Violin II (Vln. II), Viola (Vla.), and Cello/Double Bass (Cb.). The score is written in a single system with five staves. The key signature has one sharp (F#) and the time signature is 4/4. The music features complex rhythmic patterns, including sixteenth and thirty-second notes, and rests. Dynamic markings such as *f*, *mf*, *mp*, *pp*, *sfz*, and *rit.* are used throughout. Performance instructions like *rit.* and *rit. poco* are present. A double bar line with repeat dots is located at the end of measure 127. The page number 247 is printed in the top right corner.



This page contains a musical score for Violins I and Violins II. The score is written in two systems, each with two staves. The first system is labeled 'Viol. I' and the second 'Viol. II'. The music is in a common time signature and features a variety of dynamic markings including *f* (forte), *sf* (sforzando), *mf* (mezzo-forte), and *ppm* (pianissimo). Performance instructions such as *arco* (arco) and *pizzicato* (pizzicato) are used to indicate playing techniques. The score includes complex rhythmic patterns, including sixteenth and thirty-second notes, and rests. The notation is detailed, with many slurs and accents. The page number '248' is located at the top left.

This musical score consists of five staves, labeled I through V. The notation includes various musical symbols such as notes, rests, and dynamic markings. The dynamics range from *pp* (pianissimo) to *ff* (fortissimo). There are also markings for *f* (forte) and *sf* (sforzando). The score is written in a standard musical notation with a treble clef and a key signature of one flat. The staves are connected by a brace on the left side. The first staff (I) starts with a *d* marking. The second staff (II) has a *quasi piano* marking. The third staff (III) has a *quasi piano* marking. The fourth staff (IV) has a *quasi piano* marking. The fifth staff (V) has a *pp* marking. The score is written in a standard musical notation with a treble clef and a key signature of one flat.

Musical score for measures 80-85, featuring five staves: Cello (C), Violin I (Vn. I), Violin II (Vn. II), Viola I (Va. I), and Viola II (Va. II). The score includes dynamic markings such as *mf*, *f*, *ff*, *sfz*, and *rit.*, and performance instructions like *ritardando* and *rit.*. Measure numbers 80, 81, 82, 83, 84, and 85 are indicated at the beginning of their respective staves.

Musical score for measures 86-91, featuring five staves: Cello (C), Violin I (Vn. I), Violin II (Vn. II), Viola I (Va. I), and Viola II (Va. II). The score includes dynamic markings such as *mf*, *f*, *ff*, *sfz*, and *rit.*, and performance instructions like *ritardando* and *rit.*. Measure numbers 86, 87, 88, 89, 90, and 91 are indicated at the beginning of their respective staves.

This page contains the musical score for measures 11 through 15 of a piece, featuring Violin I and Violin II parts. The score is written on five systems of staves. The Violin I part is on the top staff of each system, and the Violin II part is on the bottom staff. The music is in a key with one flat (B-flat major or D minor) and a 4/4 time signature. The tempo is marked *Andante*. The dynamics range from *pp* (pianissimo) to *ff* (fortissimo). The score includes various musical notations such as slurs, accents, and dynamic markings. A rehearsal mark is present at the beginning of measure 11. The page number 11 is located at the bottom right corner.

This page contains a musical score for three instruments: Violin I (Viol. I), Violin II (Viol. II), and Viola (Viola). The score is written in a standard musical notation style with five-line staves. The instruments are arranged vertically from top to bottom: Viol. I, Viola, and Viol. II. The score includes various musical notations such as notes, rests, and dynamic markings. The dynamics used are *p* (piano), *mp* (mezzo-piano), *mf* (mezzo-forte), and *f* (forte). Performance instructions include "and point" and "and point - ext. and point sempre". The score is divided into measures by vertical bar lines, and there are repeat signs and first/second endings indicated. The page number "252" is located at the top left.

This page contains a musical score for five instruments: Violin I (Vln. I), Violin II (Vln. II), Viola, Violoncello (Vcllo), and Contrabasso (Cb.). The score is written in a standard musical notation with five staves. The Vln. I and Vln. II parts are marked with *mf* (mezzo-forte). The Viola part is marked with *mf* and includes a section marked *rit. per. sus. - rit. per.* (ritardando per se stesso - ritardando per se stesso). The Vcllo and Cb. parts are marked with *mf*. The score includes various musical notations such as notes, rests, slurs, and dynamic markings. A rehearsal mark '93' is located at the beginning of the Vln. I staff.

This page contains a musical score for four string instruments: Violin I (Vln. I), Violin II (Vln. II), Viola (Vcl.), and Cello/Double Bass (Cb.). The score is written in a standard musical notation with five-line staves. The instruments are arranged vertically from top to bottom: Vln. I, Vln. II, Vcl., and Cb. The score includes various musical notations such as notes, rests, and dynamic markings. The dynamic markings *sf* (sforzando) and *ord.* (ordinario) are used throughout the score. The page number 254 is located at the top left. The instrument labels are placed to the left of their respective staves. The score is divided into two systems, with the first system starting at measure 58 and the second system starting at measure 61.

**Viol. I**

*p* *mf* *f* *sf* *sfz*

Constantemente representado a  
 Siempre en registro agudo extremo.

**Viol. II**

*p* *mf* *f* *sf* *sfz*

Constantemente representado a  
 Siempre en registro agudo extremo.

**Viola**

*p* *mf* *f* *sf* *sfz*

Constantemente representado a  
 Siempre en registro agudo extremo.

**Cb.**

*p* *mf* *f* *sf* *sfz*

Constantemente representado a  
 Siempre en registro agudo extremo.

*mf* *sfz*

2 3 4 5 3 4 2

*mf* *sfz*

2 3 4 5 3 4 2









# APÉNDICE II

## CÓDIGO FUENTE FOMUS2 Y OBRAS

Todos los códigos aquí proporcionados, así como código adicional donde se detallan sus distintas secciones y funcionalidades, además de actualizaciones o modificaciones, y clases necesarias se pueden descargar en <https://jorgeeduardojonas.wordpress.com/>

### Contenido

- Código fuente Fomus2
- Breve ejemplo Fomus2
- Código fuente IV (parte de piano)
- Código fuente Anamnesis
- Código fuente Intervención de Anamnesis
- Código fuente Autopo(i)ética

## CÓDIGO FUENTE FOMUS2<sup>93</sup>

(Versión MacOSX)

```
// SuperFomus — SuperCollider bindings to Fomus Music Notation
// Copyright (C) 2011 Bernardo Barros
//
// This program is free software: you can redistribute it and/or modify
// it under the terms of the GNU General Public License as published by
// the Free Software Foundation, either version 3 of the License, or
// (at your option) any later version.
//
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License
// along with this program. If not, see <http://www.gnu.org/licenses/>.
Fomus2 {
var <eventList;
var <>layoutDef = "small-ensemble";
var <>beat = "1/4";
var <>timeSig = "(4 4)";
var <>multi = 1; // por default solamente un instrumento
var <>fileName = "~/Desktop/SuperFomus";
var <>pdfViewer = "/Applications/Preview.app/Contents/MacOS/Preview" ;
// var <>pdfViewer = "/Applications/Skim.app/Contents/MacOS/Skim" ;
var <>tuplets = 14;
var <>instrument = "default";
var <>instrumentName = "piano"; // valor default, si no tiene output .ly
var <>instPartId = "apart";
```

---

<sup>93</sup> Esta clase fue originalmente escrita por Bernardo Barros, se extendió a Fomus2 agregando algunas funcionalidades, como poder agregar articulaciones y poder hacer una partitura para ensamble, para obtener más ejemplos y material adicional se puede visitar: <https://jorgeeduardojonas.wordpress.com/>

```

var <>abbrev;
var <>minTupDur = 1; // para no producir tuplets que excedan 1
var <>qt = true;
var <>lilyPath = "/Applications/LilyPond.app/Contents/Resources/bin/lilypond";
var <>lilyViewPath = "/usr/bin/okular";
var <>ledgersDown = 3;
var <>ledgersUp = 3;
// var <>octsChangeScore = 2; //mientras sea valor mas alto, menos probabilidad de
utilizar oct
// var <>octsInstaffScore = 10; // para que no ponga señales de 8a o 15a a notas dentro
del pentagrama
// var <noteAccs = "note-accs = ("s" = 1, "n" = 0, "f" = -1)";
// var <noteMicro = "note-microtones = ("-qs" = 1/2, "-qf" = -1/2, "-tqs" = 3/4, "-tqf" = -
3/4)";
var <>octsUp = 2; // permite 8a y 15a, 1 solo 8a, 0 ninguno
var <>octsDown = 2; // permite 8a y 15a, 1 solo 8a, 0 ninguno

*new { arg noteList, n;
^super.new.init(noteList, n);
}

init { arg thisStuff=[], n=1;

eventList = Array.new;
// se le agrega FuncStream para que funcione con Pcollects
(thisStuff.size > 0 or: thisStuff.class == Routine).if({
this.put(thisStuff, n)
});

(thisStuff.size > 0 or: thisStuff.class == FuncStream).if({
this.put(thisStuff, n)
});

}

put { arg stuffIn, n=1;

```

```

case
{stuffIn.class == Event}
{ eventList = eventList.add(stuffIn)}

{stuffIn.class == Array}
{
stuffIn.do { |thisEvent|
if( thisEvent.class == Event,
{ eventList = eventList.add(thisEvent)},
{ "At least one element of the Array is not an Event".error}
);
}
}

{stuffIn.class == Routine}
{
stuffIn.nextN(n,()).do { |thisEvent|
eventList = eventList.add(thisEvent)
};
}

// ya con esto acepta FuncStreams al ponerlo .collect a un Pbind
{stuffIn.class == FuncStream}
{
stuffIn.nextN(n,()).do { |thisEvent|
eventList = eventList.add(thisEvent)
};
}

{(stuffIn.class == Event or: stuffIn.class == Array).not}
{ "You must provide an Event, a Stream or an Array of Events".error};

}

asString {
var out = "";

```

```

eventList.do { arg i;
out = out ++ i.asFomusString ++ "\n"
};
^out;
}

qtString {
this.qt.if(
{"quartertunes = yes"},
{"quartertunes = no"})
}
header {

^(
this.qtString ++ "\n" ++
"layout = " ++ this.layoutDef ++ "\n" ++
"beat: " ++ this.beat ++ "\n" ++
"timesig: " ++ this.timeSig ++ "\n" ++
"lily-exe-path = " ++ this.lilyPath ++ "\n" ++
"lily-view-exe-path = " ++ this.lilyViewPath ++ "\n" ++
"note-accs = (\s\ = 1, \n\ = 0, \f\ = -1)" ++ "\n" ++ // baja probabilidad de error en
module lilyout
"note-microtones = (\qs\ = 1/2, \-qf\ = -1/2, \-tqs\ = 3/4, \-tqf\ = -3/4)" ++ "\n" ++
// para tuplets
"tuplets = " ++ this.tuplets ++ "\n"
"min-tupletdur = " ++ this.minTupDur ++ "\n" ++
// "ledgers-up = " ++ this.ledgersUp ++ "\n" ++
// "ledgers-down = " ++ this.ledgersDown ++ "\n" ++
// "octs-change-score = " ++ this.octsChangeScore ++ "\n" ++
// "octs-instaff-score = " ++ this.octsInstaffScore ++ "\n" ++
"octs-up = " ++ this.octsUp ++ "\n" ++
"octs-down = " ++ this.octsDown ++ "\n" ++

// Función para partituras de instrumentos multiples

```



```

if(
this.multi != 1,

{
var partDef = [], datos, primeraParte;

primeraParte = ["part " ++ this.instPartId.at(0) ++ "\n"];
this.multi.do{ arg x;

partDef = partDef ++[

"part <id: " ++ this.instPartId.at(x) ++ " , " ++
" name = " ++ this.instrumentName.at(x).asCompileString
++ " inst = " ++ this.instrument.at(x) ++ ">" ++ "\n"]
};
datos = Pseq([ partDef[0..(partDef.size - 1)], primeraParte[0]], 1).asStream.all
.asString.replace("[", "").replace("]", "").replace(", ", "");

},

{

"part <id: " ++ this.instPartId ++ " , " ++
" name = " ++ this.instrumentName.asCompileString ++ " inst = " ++ this.instrument ++
">" ++ "\n" ++
"part " ++ this.instPartId ++ "\n"
}
)

).asString;
}

write {
var file;
file = File(this.fileName.standardizePath ++ ".fms", "w");
file.write(this.header ++ "\n");
file.write(this.asString);
}

```

```

file.close;
}

ly {
this.write;
(
"fomus " ++ this.fileName.standardizePath ++ ".fms" ++
" -o " ++ this.fileName.standardizePath ++ ".ly"
).runInTerminal;
}

midi {
this.write;

(
"fomus " ++ this.fileName.standardizePath ++ ".fms" ++
" -o " ++ this.fileName.standardizePath ++ ".mid"
).runInTerminal;
}
xml {
this.write;
(
"fomus " ++ this.fileName.standardizePath ++ ".fms" ++
" -o " ++ this.fileName.standardizePath ++ ".xml"
).runInTerminal;
}
show {
(this.lilyViewPath ++ " " ++ this.fileName.standardizePath ++ ".pdf").runInTerminal;
}

}

+ Event {

asFomusString {
var outString = "", instrument = "";

```

```

// instrument = this.instrument ++ "aquí estoy".println;

this.keys.includes(\instrument).if({
  outString = "part " ++ this[\instrument].asString
});

this.keys.includes(\part).if({
  outString = "part " ++ this[\part].asString
});

this.keys.includes(\voice).if({
  outString = "voice " ++ this[\voice].asString
});

// se mueve de lugar el mark para que esté después de pitch en
// el archivo fms y se puedan agregar articulaciones

// Se extiende funcionalidad para agregar apoyaturas
this.keys.includes(\grace).if({
  outString = outString ++ " grace " ++ this[\grace].asString ++ " ; "
});

this.keys.includes(\time).if({

  outString = outString ++ " time " ++ this[\time].asString

});

this.keys.includes(\time).not.if({
  outString = outString ++ " time + ";
});

this.keys.includes(\dur).or(this.keys.includes(\duration)).if({
  outString = outString ++ " duration " ++ this[\dur].asString
});

this.keys.includes(\midinote).if({

```

```

if( this[\midinote].class == Array,
{ // if this is a chord:
this[\midinote].do({arg thisNote;
outString = outString ++ " pitch " ++ thisNote.asString + "; \n"
});

// para que los acordes también puedan tener marks
this.keys.includes(\mark).if({
outString = outString ++ this[\mark].asString
});

outString = outString ++ ",";

},{

// pequeña modificación para que se pueda utilizar Rest tanto en
// streams de midinote como de duración
if((this[\midinote].isNumber == false).or(this[\isRest] == true),
{ outString = outString ++ " rest "},
{ outString = outString ++ " pitch " ++ this[\midinote].asString; }
);
/*if((this[\midinote].isNumber).and(this[\isRest] == false),
{outString = outString ++ " pitch " ++ this[\midinote].asString;},
{outString = outString ++ " rest "}
);*/

// Se cambió a este lugar Mark
// Para las articulaciones
this.keys.includes(\mark).if({
outString = outString ++ this[\mark].asString
});

outString = outString ++ ",";
}
)
});

```

```

this.keys.includes(\note).if({

if( this[\note].class == Array,
{ // if this is a chord:
this[\note].do({arg thisNote;
outString = outString ++ " pitch " ++ (thisNote + 60).asString + "; \n"
})
},{
if(this[\note].isNumber,{outString = outString ++ " pitch " ++ (this[\note] + 60).asString;},
{outString = outString ++ " rest "}
);

/*
// Se cambió a este lugar Mark
// Para las articulaciones
this.keys.includes(\mark).if({
outString = outString ++ this[\mark].asString
});*/
outString = outString ++ ",";
}
)
});
^outString
}
}

```

## BREVE EJEMPLO FOMUS2

Agregar clase a la carpeta que arroje en la ventana post

```
Platform.systemExtensionDir;
```

```
// Ejemplo para ensambles
```

```
(
```

```
~p1 = Pbind(
```

```
\instrument, "violin",
```

```
\part, "vln1",
```

```
\time, Pseq([0, Pseq("+", inf)], inf), // Esta parte es para empezar nuevo instrumento
```

```
\midinote, Pbrown(60, 80, 0.5, inf),
```

```
\dur, Prand((0.125, 0.25..1.5), inf)
```

```
).asStream;
```

```
~p2 = Pbind(
```

```
\instrument, "violin",
```

```
\part, "vln2",
```

```
\time, Pseq([0, Pseq("+", inf)], inf), // 0 indica cambio de instrumento
```

```
\midinote, Pseq([64, 63], inf),
```

```
\dur, Prand((0.125, 0.25..1.5), inf)
```

```
).asStream;
```

```
~p3 = Pbind(
```

```
\instrument, "violin",
```

```
\part, "vln3",
```

```
\time, Pseq([0, Pseq("+", inf)], inf),
```

```
\midinote, Pbrown(60, 80, 0.5, inf),
```

```
\dur, Prand((0.125, 0.25..1.5), inf)
```

```
).asStream;
```

```
~p4 = Pbind(
```

```
\instrument, "cello",
```

```
\part, "vc1",
```

```
\time, Pseq([0, Pseq("+", inf)], inf),
```

```
\midinote, Pseq([64, 63], inf),
```

```
\dur, Prand((0.125, 0.25..1.5), inf)
).asStream;
```

```
~p5 = Pbind(
\instrument, "cello",
\part, "vc2",
\time, Pseq([0, Pseq("+", inf)], inf),
\midinote, Pbrown(60, 80, 0.5, inf),
\dur, Prand((0.125, 0.25..1.5), inf)
).asStream;
```

```
~p6 = Pbind(
\instrument, "cello",
\part, "vc3",
\time, Pseq([0, Pseq("+", inf)], inf),
\midinote, Pseq([64, 63], inf),
\dur, Prand((0.125, 0.25..1.5), inf)
).asStream;
```

```
~p7 = Pbind(
\instrument, "contrabass",
\part, "cb1",
\time, Pseq([0, Pseq("+", inf)], inf),
\midinote, Pbrown(30, 50, 0.5, inf),
\dur, Prand((0.125, 0.25..1.5), inf)
).asStream;
```

```
~p8 = Pbind(
\instrument, "contrabass",
\part, "cb2",
\time, Pseq([0, Pseq("+", inf)], inf),
\midinote, Pseq([30, 50], inf),
\dur, Prand((0.125, 0.25..1.5), inf)
).asStream;
```

```
~p9 = Pbind(
\instrument, "contrabass",
\part, "cb3",
```

```

\time, Pseq([0, Pseq("+", inf)], inf),
\midinote, Pseq([30, 50], inf),
\dur, Prand((0.125, 0.25..1.5), inf)
).asStream;
~p1EvArr = Array.fill(20, { ~p1.next() });
~p2EvArr = Array.fill(20, { ~p2.next() });
~p3EvArr = Array.fill(20, { ~p3.next() });
~p4EvArr = Array.fill(20, { ~p4.next() });
~p5EvArr = Array.fill(20, { ~p5.next() });
~p6EvArr = Array.fill(20, { ~p6.next() });
~p7EvArr = Array.fill(20, { ~p7.next() });
~p8EvArr = Array.fill(20, { ~p8.next() });
~p9EvArr = Array.fill(20, { ~p9.next() });

~p1Seq = Pseq(~p1EvArr, 1).asStream;
~p2Seq = Pseq(~p2EvArr, 1).asStream;
~p3Seq = Pseq(~p3EvArr, 1).asStream;
~p4Seq = Pseq(~p4EvArr, 1).asStream;
~p5Seq = Pseq(~p5EvArr, 1).asStream;
~p6Seq = Pseq(~p6EvArr, 1).asStream;
~p7Seq = Pseq(~p7EvArr, 1).asStream;
~p8Seq = Pseq(~p8EvArr, 1).asStream;
~p9Seq = Pseq(~p9EvArr, 1).asStream;

~binds = Pseq([
~p1Seq,
~p2Seq,
~p3Seq,
~p4Seq,
~p5Seq,
~p6Seq,
~p7Seq,
~p8Seq,
~p9Seq

```



```

],1).asStream;
)

(
// 20 * 9
x = Fomus2(~binds, 180).multi_(9) // multi: numero de instrumentos
.instPartId_([
  "vln1", "vln2", "vln3",
  "vc1", "vc2", "vc3",
  "cb1", "cb2", "cb3"
]) // arreglo de id de instrumentos
.instrumentName_([
  "vln I", "vln II", "vln III",
  "vc I", "vc II", "vc III",
  "cb I", "cb II", "cb III"
]) // arreglo de nombre de instrumentos
.instrument_([
  "violin", "violin", "violin",
  "cello", "cello", "cello",
  "contrabass", "contrabass", "contrabass"
]) // arreglo de instrumento de instDef en fomus
.layoutDef_("small-ensemble")
.beat_("1/2")
.timeSig_("(2 2)");
)
x.ly
x.xml;

```

## CÓDIGO FUENTE IV (PIANO)

```

// Seleccionar Todo
// Registro para piano

(
//*****

// ===== Generador de ritmos con tendencia a valores Cortos =====

// Valores rítmicos derivados de secciones aureas de idea original de 114", 1'54".
// En la contraparte de esta pieza si se manejarán valores rítmicos derivados
// de la duración total de 128", 2'04". ya que el resto el duración del resto de las piezas
parte
// de 128"; otra posibilidad es calcular valores apartir de 114"...
// La máquina de estado finito tiene la tendencia de ir hacia valores largos, pero siempre
regresando hacia
// valores más cortos, haciendo curvas o saltos.

(
(
~p fsmC = P fsm([
[0], // 0 estado inicial

\j, [0,1], // 50% de probabilidad de quedarse en el estado 0 que mapea a 0.0769 en
diccionario y 50% de irse a 0.125.
\k, [0,1,1,2,3], // 20 % quedarse en 0.125, 40% de irse a 0.2, y 20 para irse a 0.333 ó
0.5.
\l, [1,2,2,2,3,4], // 16.6% regresar a 0.125, 50% de quedarse en 0.2, 16.6% de irse a
0.333 ó 0.5.
\m, [0,1,2,3,4], // 20% de probabilidad de irse a los estados del 0 al 4,(0.0769, 0.125,
0.2, 0.333, 0.5, 1.0).
\n, [1,2,4,5], // 25% de probabilidad de quedarse en 1.0, regresar a los estado 1 o 2, o
avanzar a 1.5.
\o, [1,2,3,4,5,6], // igual probabilidad de irse a los estados 1 al 6. 16.6% de seguir
avanzando.

```

```

\p, [7,6,5,4,2], // 20% de seguir avanzando.
\q, [8,7,6,5,4,3,2],// 0.1428% de probabilidad de avanzar a último estado.
\r, [7,6,5,4,3,2,1],// si llega retrocede
nil, nil
]);
);

//

~pfsmC.asStream;

// diccionario fijo tendencia a valores Cortos (inversión del diccionario largo)

/*
duraciones para 128":
128/1.618033
= 79.108
79.108/1.618033
= 48.8914
48.8914/1.618033
= 30.2165
30.2165/1.618033
= 18.6748
18.6748/1.618033
= 11.5416
11.5416/1.618033
= 7.1331
7.1331/1.618033
= 4.4085
4.4085/1.618033
= 2.7246
2.7246/1.618033
= 1.6838
1.6838/1.618033
= 1.0406
1.0406/1.618033

```

```

= 0.64312
0.64312/1.618033
= 0.3974
0.3974/1.618033
= 0.2456
0.2456/1.618033
= 0.1517
0.1517/1.618033
= 0.0937
0.0937/1.618033
*/
(
~diccionarioC = Dictionary[
\j -> [1/13],
\k -> [1/8],
\l -> [0.2],
\m -> [0.333],
\n -> [0.5],
\o -> [1.0],
\p -> [1.5],
\q -> [2.4],
\r -> [4.0]
]);

// mapeo de máquina de estado finito a diccionario, se convierte en stream.
~mapC = ~pfsmC.asStream.collect({ |i| ~diccionarioC[i]});

// Switch para completar pulsos

(
~ritmosC = Pfunc{switch( ~mapC.next, // si el valor es 2.4, se completa para no generar
valores demasiado complicados, etc.

[4.0], {[4.0]},
[2.4], {[2.4] ++ [[0.2,0.2,0.2],[0.4,0.2].scramble].choose}, // provisional agregar ligadura
[1.5], {[1.5] ++ [[0.5], [1.0, 0.5], [1.5]].choose},

```

```

[1.0], {[1.0]},
[0.5], {[0.5] ++ [[0.5],[0.25,0.25]].choose},
[0.333], {[0.333] ++ [[0.333,0.333], [0.666], [0.16667,0.16667,0.16667,0.16667],
[0.333,0.16667,0.16667]].choose},
[0.2], {[0.2] ++ [[0.2,0.2,0.2,0.2], [0.8], [0.2,0.2,0.4]].choose},
[1/8], {if( 0.5.coin, {[1,[1,1,1,1,1,1,1,1]].convertRhythm.flat},
{[1,[3,2,1,1,1].scramble].convertRhythm.flat}}},
[1/13], {if(0.5.coin,{[1/13 ! 13].flat},
{[1/13,2/13,2/13,1/13,1/13,1/13,1/13,1/13,3/13].scramble})}

});
);

~ritmosConvC = ~ritmosC.asStream;

/*~ritmosConvC.next
~ritmosConvC.nextN(30).flat;*/

(
// para acceder al ritmo completo.
~rCSwitch = Pfunc{switch(g = ~ritmosConvC.next; g.postIn.size,
1, {g[0]},
2, {Pseq([g[0],g[1]],inf).asStream.nextN(2)},
3, {Pseq([g[0],g[1],g[2]],inf).asStream.nextN(3)},
4, {Pseq([g[0],g[1],g[2],g[3]],inf).asStream.nextN(4)},
5, {Pseq([g[0],g[1],g[2],g[3],g[4]],inf).asStream.nextN(5)},
6, {Pseq([g[0],g[1],g[2],g[3],g[4],g[5]],inf).asStream.nextN(6)},
7, {Pseq([g[0],g[1],g[2],g[3],g[4],g[5],g[6]],inf).asStream.nextN(7)},
8, {Pseq([g[0],g[1],g[2],g[3],g[4],g[5],g[6],g[7]],inf).asStream.nextN(8)},
9, {Pseq([g[0],g[1],g[2],g[3],g[4],g[5],g[6],g[7],g[8]],inf).asStream.nextN(9)},
10, {Pseq([g[0],g[1],g[2],g[3],g[4],g[5],g[6],g[7],g[8],g[9]],inf).asStream.nextN(10)},
11, {Pseq([g[0],g[1],g[2],g[3],g[4],g[5],g[6],g[7],g[8],g[9],g[10]],inf).asStream.nextN(11)},
12,
{Pseq([g[0],g[1],g[2],g[3],g[4],g[5],g[6],g[7],g[8],g[9],g[10],g[11]],inf).asStream.nextN(12)},
13,

```

```

{Pseq([g[0],g[1],g[2],g[3],g[4],g[5],g[6],g[7],g[8],g[9],g[10],g[11],g[12]],inf).asStream.nextN
(13)}

)}.flatten;
);

/*
=====
=====
variable global para generador de ritmo con tendencia a valores Cortos ~durCorto
=====
===== */

~durCorto = ~rCSwitch.asStream;
);

//*****
/*~durCorto.next

(
z = Pbind(

\midinote,Prand([60,65,66,71,30,30,30,30,30],inf) + “,”, // no se agregaba punto y coma
\dur, ~durCorto
);
)
// con Fomus
x = Fomus(z.asStream,30);
x.ly
x.midi
x.xml
*/
// ===== Pieza I = 1'54" =====
//*****

```

```

(
(
~pfsml = P fsm([

[0], // Estado inicial, 0.

\ a, [0,0,0,0,1], // 20% de probabilidad de agregar notas, en estados del 0 al 8; 80% de
probabilidad de permanecer con el
\ b, [1,1,1,1,2], // mismo número de notas.
\ c, [2,2,2,2,3],
\ d, [3,3,3,3,4],
\ e, [4,4,4,4,5],
\ f, [5,5,5,5,6],
\ g, [6,6,6,6,7],
\ h, [7,7,7,7,8],
\ i, [8,8,8,8,9],
\ j, [9,9,9,9,1] // En el estado 9, hay un 80% de probabilidad de permanecer en el
agregado, y un 20% de probabilidad de
]); // reiniciar el proceso.
);

~pfsml.asStream;
// Diccionario para notas
(
~diccionariol = Dictionary[
\ a -> [60,66,71], // comienza con 3 notas
\ b -> [60,66,71,65],
\ c -> [60,66,71,65,64],
\ d -> [60,66,71,65,64,70],
\ e -> [60,66,71,65,64,70,63],
\ f -> [60,66,71,65,64,70,63,69],
\ g -> [60,66,71,65,64,70,63,69,62],
\ h -> [60,66,71,65,64,70,63,69,62,68],
\ i -> [60,66,71,65,64,70,63,69,62,68,61],
\ j -> [60,66,71,65,64,70,63,69,62,68,61,67] // termina con agregado, con la posibilidad

```

de reiniciar hacia 3 notas.

```
]);
```

```
~mapeol = ~pfsml.asStream.collect({ |i| ~diccionariol[i]});
```

```
(
```

```
~nlswitch = Pfunc{switch( a = ~mapeol.next; a.postln.size,
```

```
3, {Prand([a[0],a[1],a[2]],inf).asStream.nextN(3)},
```

```
4, {Prand([a[0],a[1],a[2],a[3]],inf).asStream.nextN(4)},
```

```
5, {Prand([a[0],a[1],a[2],a[3],a[4]],inf).asStream.nextN(5)},
```

```
6, {Prand([a[0],a[1],a[2],a[3],a[4],a[5]],inf).asStream.nextN(6)},
```

```
7, {Prand([a[0],a[1],a[2],a[3],a[4],a[5],a[6]],inf).asStream.nextN(7)},
```

```
8, {Prand([a[0],a[1],a[2],a[3],a[4],a[5],a[6],a[7]],inf).asStream.nextN(8)},
```

```
9, {Prand([a[0],a[1],a[2],a[3],a[4],a[5],a[6],a[7],a[8]],inf).asStream.nextN(9)},
```

```
10, {Prand([a[0],a[1],a[2],a[3],a[4],a[5],a[6],a[7],a[8],a[9]],inf).asStream.nextN(10)},
```

```
11, {Prand([a[0],a[1],a[2],a[3],a[4],a[5],a[6],a[7],a[8],a[9],a[10]],inf).asStream.nextN(11)},
```

```
12,
```

```
{Prand([a[0],a[1],a[2],a[3],a[4],a[5],a[6],a[7],a[8],a[9],a[10],a[11]],inf).asStream.nextN(12)}
```

```
}).flatten;
```

```
);
```

```
~notasl = ~nlswitch.asStream;
```

```
);
```

```
// ~notasl.next
```

```
// *****
```

```
// autómata para octavas
```

```
(
```

```
(
```

```
~pfsmOctl = Pfsm([
```

```
[0], // Estado inicial 0.
```

```
\a, [0,0,0,1], // 75% de probabilidad para permanecer en el mismo estado, 25% para  
avanzar.
```

```
\b, [1,1,1,2],
```

```
\c, [2,2,2,3],
```



```

\d, [3,3,3,4],
\e, [4,4,4,5],
\f, [5,5,5,6],
\g, [6,6,6,7],
\h, [7,7,7,8],
\i, [8,8,8,9],
\j, [9,9,9,10],
\k, [10,10,10,11],
\l, [11,11,11,12],
\m, [12,12,12,13],
\n, [13] // Estado final, aquí permanece.
]);
);

~pfsmOctl.asStream;
// Diccionario para octavas

(
(

// Tesitura para piano (se modifica para clarinete y para flauta)
~diccionarioOctl = Dictionary[

\la -> [0,\rest ! 4].flat, // Dominan silencios, octava central.
\lb -> [0,12, \rest ! 4].flat, // Octava central más una octava superior.
\lc -> [-12,0,12, \rest ! 6].flat, // Se agrega octava inferior.
\ld -> [0,\rest ! 2].flat, // Empieza balance silencio-sonido, octava central.
\le -> [-24,0,24,\rest ! 3].flat, // Octava central mas octava inferior*2, superior*2.
\lf -> [-12,0,12, \rest ! 2].flat, // Domina sonido, se cierra trinche octava central + inferior +
superior.
\lg -> [-36,-24,-12,0,12,24,36, \rest ! 3 ].flat, // Rango completo de 7 octavas sin trinche.
\lh -> [0,0,\rest], // Octava central. Sigue dominando sonido. 2:1.
\li -> [-48,-36,-24,-12,0,12,24,36,48,\rest ! 3].flat, // Empieza filtraje hacia silencio.
\lj -> [-48,-36,-24,-12,0,12,24,36,48,\rest ! 5].flat,
\lk -> [-48,-36,-24,-12,0,12,24,36,48,\rest ! 8].flat,
\ll -> [-48,-36,-24,0,24,36,48,\rest ! 13].flat, // Se empieza a formar "trinche" de rango.

```

```

\m -> [-48,-36,0,36,48,\rest ! 13].flat,
\n -> [-48,0,48,\rest ! 13].flat // Octava central más extremo agudo y grave, sonido-
silencio 3:13.

]);

~mapeoOctl = ~p fsmOctl.asStream.collect({ |i| ~diccionarioOctl[i]});
//~mapeoOctl.next;
(
~chooseOctlswitch = Pfunc{ switch ( b = ~mapeoOctl.next; b.postIn.size,
3, {Pxrnd([b[0],b[1],b[2]],inf).asStream.nextN(3)},
5, {Pxrnd([b[0],b[1],b[2],b[3],b[4]],inf).asStream.nextN(5)},
6, {Pxrnd([b[0],b[1],b[2],b[3],b[4],b[5]],inf).asStream.nextN(6)},
9, {Pxrnd([b[0],b[1],b[2],b[3],b[4],b[5],b[6],b[7],b[8]],inf).asStream.nextN(9)},
10, {Pxrnd([b[0],b[1],b[2],b[3],b[4],b[5],b[6],b[7],b[8],b[9]],inf).asStream.nextN(10)},
12,
{Pxrnd([b[0],b[1],b[2],b[3],b[4],b[5],b[6],b[7],b[8],b[9],b[10],b[11]],inf).asStream.nextN(12
)},
14,
{Pxrnd([b[0],b[1],b[2],b[3],b[4],b[5],b[6],b[7],b[8],b[9],b[10],b[11],b[12],b[13]],inf).asStrea
m.nextN(14)},
16, {Pxrnd([b[0],b[1],b[2],b[3],b[4],b[5],b[6],b[7],b[8],b[9],b[10],
b[11],b[12],b[13],b[14],b[15]],inf).asStream.nextN(16)},
17, {Pxrnd([b[0],b[1],b[2],b[3],b[4],b[5],b[6],b[7],b[8],b[9],b[10],
b[11],b[12],b[13],b[14],b[15],b[16]],inf).asStream.nextN(17)},
18, {Pxrnd([b[0],b[1],b[2],b[3],b[4],b[5],b[6],b[7],b[8],b[9],b[10],
b[11],b[12],b[13],b[14],b[15],b[16],b[17]],inf).asStream.nextN(18)},
20, {Pxrnd([b[0],b[1],b[2],b[3],b[4],b[5],b[6],b[7],b[8],b[9],
b[10],b[11],b[12],b[13],b[14],b[15],b[16],b[17],b[18],b[19]],inf).asStream.nextN(20)}

)}.flatten;
);

~chooseOctl = ~chooseOctlswitch.asStream;
)
);

```

```
// ~chooseOctl.next

// *****

//~chooseOctl.next;

//seleccionar todo
)

(
p = Pbind(
  \midinote, (~notasI + ~chooseOctl).max(21).min(108) , // max y min para limitar al
  registro del piano
  \midinote, (~notasI + ~chooseOctl) ,
  \dur, ~durCorto
);
)

// con Fomus
x = Fomus2(p.asStream,400).instrument_("piano").octsUp_(0).octsDown_(0);
x.ly
x.midi
x.xml
```

## CÓDIGO FUENTE ANAMNESIS

Agregar clase Pgolden a la carpeta que arroje en la ventana post

Platform.systemExtensionDir;

```
Pgolden : Pattern { // golden mean
var <>goldenOf = 0, <>upDown = 0, <>length = 1;
*new{ arg goldenOf = 0, upDown = 0, length = 1;
^super.newCopyArgs(goldenOf, upDown, length)
}
storeArgs{^[goldenOf, upDown, length]}

embedInStream{ arg inval;
var cur = goldenOf.asStream;
var len = length.value(inval);
var updown = upDown.value(inval);

len.value(inval).do{
if(upDown == 0) {
cur = cur.next(inval) / 1.618033;
inval = cur.yield;
}{
cur = cur.next(inval) * 1.618033;
inval = cur.yield;
}
};
^inval;
}
}
```

```

// =====
// Anamnesis para violín Solo =====
// Jorge Eduardo Gómez Elizondo =====
// primera sección
(

// Esperar un momento a que en la ventana post se imprima a Routine
// Aun existe un bicho en el que puede arrojar un error al querer
// correr con Fomus2,si esto sucede volver a ejecutar la rutina
~media = (0.09, 0.1 .. 1.0).collect{|i| i ! 5;}.flat;
~media.size;
~mediaArreglo = Pseq(~media, inf).asStream;
// "Meseta" donde la probabilidad permanece estable y las alturas en el registro agudo
~mes = Pgolden(~media.size, 0, 6).asStream.nextN(6).round(1).reverse;
~mes = ~mes[4];

// Secciones Aureas de longitud total (sin meseta)
~aurFase = Pgolden(~media.size, 0, 3).asStream.nextN(3).round(1).reverse;
// Modulación de probabilidades
~cresta1 = Pseq(Pseries.fromEndpoints(0.0, 0.75,
~aurFase[0])).asStream.all, 1).asStream;

~cero1 = Pseq(
Pseries.fromEndpoints(0.75, 0.01, ((~aurFase[1] -
~aurFase[0])/1.618033).round(1)).asStream.all,
1).asStream;

~cresta2 = Pseq(Pseries.fromEndpoints(0.01, 0.8,
(~aurFase[1] - ~aurFase[0]) - ((~aurFase[1] -
~aurFase[0])/1.618033).round(1)).asStream.all,
1).asStream;

~cero2 = Pseq(
Pseries.fromEndpoints(0.8, 0.1, ((~aurFase[2] -
~aurFase[1])/1.618033).round(1)).asStream.all,
1).asStream;

```

```
~cresta3 = Pseq(Pseries.fromEndpoints(0.1, 0.9, (~media.size –
~aurFase[2])).asStream.all, 1).asStream;
```

```
~cero3 = Pseq(
Pseries.fromEndpoints(0.9, 0.25, ((~media.size –
~aurFase[2])/1.618033).round(1)).asStream.all,
1).asStream;
```

```
~cresta4 = Pseq(
Pseries.fromEndpoints(0.25, 1.0,
~media.size – (~aurFase[2] + ((~media.size –
~aurFase[2])/1.618033).round(1))).asStream.all,
1).asStream;
```

```
~psn = Routine{ |x|
```

```
(~media.size + ~mes).do{ |i|
```

```
x = case
```

```
// primera cresta
```

```
{i < ~aurFase[0]}
```

```
{~cresta1.next}
```

```
// sección aurea entre fase indice 0 y 1, baja a cero
```

```
{(i >= ~aurFase[0]).and(i < (~aurFase[0] + ((~aurFase[1] –
~aurFase[0])/1.618033).round(1)))}
```

```
{~cero1.next}
```

```
// segunda cresta culmina en aur 1
```

```
{(i >= (~aurFase[0] + ((~aurFase[1] – ~aurFase[0])/1.618033).round(1))).and
```

```
(i < ~aurFase[1])}
```

```
{~cresta2.next}
```

```
// baja a cero en aurea de aur 1 y aur2
```

```
{(i >= ~aurFase[1]).and
```

```
(i < (~aurFase[1] + ((~aurFase[2] – ~aurFase[1])/1.618033).round(1)))}
```

```
{~cero2.next}
```

```

// tercera cresta culmina en aur2
{(i >= (~aurFase[1] + ((~aurFase[2] - ~aurFase[1])/1.618033).round(1))).and
(i < ~aurFase[2])}
{~cresta3.next}

{(i >= ~aurFase[2]).and
(i < (~aurFase[2] + ((~media.size - ~aurFase[2])/1.618033).round(1)))}
{~cero3.next}

{(i >= (~aurFase[2] + ((~media.size - ~aurFase[2])/1.618033).round(1))).and
(i < ~media.size)}
{~cresta4.next}

{(i >= ~media.size).and(i <= (~media.size + ~mes))} {0.85};

// Probabilidades pesadas
// en la rutina solamente se pone yield en esta parte porque
// es el valor que será arrojado
Pwrand([0,1],[1-x,x].normalizeSum,inf).asStream.next.yield;
}
}.all;
~numEventos = ~psn.collect{|i| if(i > 0) {i} }.reject(_ == nil).size;
// ALTURAS _____
// Limites de movimiento browniano
~hi = Pgeom.fromEndpoints(57.0, 107.0, ~media.size).asStream.all;
~segLo = Pseq(Pseries.fromEndpoints(55.0,100.0,~media.size -
~aurFase[2]).asStream.all, inf).asStream;

~lo = Routine{
(~media.size + ~mes).do{|i|
case
{i < ~aurFase[2]} {55.0.yield}

{(i >= ~aurFase[2]).and(i < ~media.size)}
{~segLo.next.yield}

```

```

{(i >= ~media.size).and(i <= (~media.size + ~mes)) }
{100.0.yield}
}
};
~lo = ~lo.all;

// Meseta
~hiMes = 107.0 ! ~mes;

~hi = ~hi ++ ~hiMes;

~stepMod = Routine{
(~media.size + ~mes).do{ |i|
case
{i < ~aurFase[0]} {0.5.yield}

{(i >= ~aurFase[0]).and(i < ~aurFase[1])} {rrand(1.0,3.0).yield}

{(i >= ~aurFase[1]).and(i < ~aurFase[2])} {rrand(3.0, 8.0).yield}

{i >= ~aurFase[2]} {rrand(3.0, 13.0).yield}
}
};

~stepMod = ~stepMod.all;

~vlnBrwnl = Pbrown(Pseq(~lo, inf), Pseq(~hi, inf), Pseq(~stepMod,inf),
inf).asStream.round(0.5);
// [~hi, ~lo, ~vlnBrwnl.nextN((~media.size + ~mes)-1)].plot.superpose_(true);

// Glissandos cuando hay evento

~glissExpAsc = Pfsm([

#[0],

\|a, #[0, 0, 1], // 66.66% 0 33.33% 1
\|b, #[0, 0, 1, 1, 1, 2, 2], // 28.5% 0 42.8% 1 28.5% 2

```



```

\c, #[0, 1, 2, 2, 2, 3, 3], // 14.2% 0 14.2% 1 42.8% 2 28.5% 3
\d, #[1, 2, 3, 3, 3, 4, 4, 5], // 12.5% 1 12.5% 2 37.5% 3 25.0% 4 12.5% 5
\e, #[2, 3, 3, 4, 4, 4, 5, 5], // 12.5% 2 25.0% 3 37.5% 4 25.0% 5

nil,
]).asStream;
~glissExpDictionary = Dictionary[
\ a -> Pwrand([0.0,0.5,1.0,1.5, 2.0], [0.5, 0.5, 1.0, 1.25, 1.5].normalizeSum, inf).asStream,
\ b -> Prand([2.5, 3.0, 3.5, 4.0, 4.5, 5.0], inf).asStream,
\ c -> Prand([5.0,5.5, 6.0, 7.0, 7.5, 8.0], inf).asStream,
\ d -> Prand([8.5,9.0,9.5,10.0,10.5,11.0, 11.5], inf).asStream,
\ e -> Prand([12.0,13.0,14.0,15.0,16.0,17.0], inf).asStream,
];

~glissDetonAsc = ~glissExpAsc.collect{|i| if( i != nil)

{~valorActual = ~glissExpDictionary[i].next;

case
// en pares para que el valor del glissando descendente no sea mayor al ascendente
{~valorActual <= 2.0}
{[Pwrand([~valorActual,~valorActual.neg],[0.8, 0.2],inf).asStream.next,
~valorActual.rand.round(0.5)]}

{(~valorActual > 2.0).and(~valorActual <= 5.0)}
{[Pwrand([~valorActual,~valorActual.neg],[0.8,
0.2],inf).asStream.next,~valorActual.rand.round(0.5)]}

{(~valorActual > 5.0).and(~valorActual <= 8.0)}
{[Pwrand([~valorActual,~valorActual.neg],[0.8,
0.2],inf).asStream.next,~valorActual.rand.round(0.5)]}

{(~valorActual > 8.0).and(~valorActual <= 11.5)}
{[Pwrand([~valorActual,~valorActual.neg],[0.8,
0.2],inf).asStream.next,~valorActual.rand.round(0.5)]}

```

```

{(~valorActual > 11.5).and(~valorActual <= 17.0)}
{[Pwrand([~valorActual,~valorActual.neg],[0.8,
0.2],inf).asStream.next,~valorActual.rand.round(0.5)]}

}

{~glissExpAsc.reset}
};
// Las Alturas Ici Monsieur
~expOrBrown = Routine{ |x|
(~media.size + ~mes).do{|i|
if(~psn[i] == 0)

{~vlnBrwnl.next.yield}

{ ~brwnNow = ~vlnBrwnl.next;
~ind = ~glissDetonAsc.next;
(~brwnNow + ~ind[0]);
(~brwnNow + ~ind[1]).yield;
}

}
};

// [~hi, ~lo, ~expOrBrown.nextN((~media.size + ~mes)-1)].plot.superpose_(true);
// DURACIONES_____
~a = Pseq((1,2 ..~numEventos),1).asStream;

~contador = Routine{ |x|
(~media.size + ~mes).do{ |i|
if(~psn[i] != 0) {~a.next.yield;} {"no".yield}

}
};

~contArreglo = Pseq(~contador.nextN(~media.size + ~mes),1).asStream.all;

```

```

~conActualNumber = 0;

~durVal = Routine{
  (~media.size + ~mes).do{|i,y|
    if((~psn[i] == 0).or((~contActual = ~contArreglo[i]) == "no"))

// Si no hay evento las duraciones pueden ser

  {
  case
  {~conActualNumber < ~aurFase[0]}

  {
  Pwrand((0.125, 0.25 .. 2.0),
  Pseries.fromEndpoints(0.1,1.0,16).asStream.all.normalizeSum, // mayor probabilidad
  valores largos
  inf).asStream.next.yield
  }

  {(~conActualNumber >= ~aurFase[0]).and(~conActualNumber < ~aurFase[1])}

  {
  Pwrand((0.125, 0.25 .. 2.0), // mayor probabilidad valores intermedios, palíndromo de
  probabilidades
  (
  Pseries(0.1, 1.0, 8).asStream.all ++
  Pseries(1.0, 0.1, 8).asStream.all
  ).normalizeSum,
  inf).asStream.next.yield
  }

  {(~conActualNumber >= ~aurFase[1]).and(~conActualNumber < ~aurFase[2])}
  {
  Pwrand((0.125, 0.25 .. 2.0),
  Pexprand(0.001, 1.0, 16).asStream.all.reverse.normalizeSum,
  inf).asStream.next.yield
  }
}

```

```

{~conActualNumber >= ~aurFase[2]}
{
Pwrand((0.125, 0.25 .. 2.0),
Pexprand(0.001, 1.0, 16).asStream.all.reverse.normalizeSum // Mayor probabilidad de
tener valores cortos
).asStream.next.yield
}

}

// Si hay evento las duracion será de 0.125

{
~conActualNumber = ~contActual;
0.125.yield}

}
};
~sisEle1 = LSys(

// axioma
"a()",

// reglas
[

"a()" ->
{
[
"a()", "A()", "b()", "A()c()", "b()A()", "j()"
].wchoose([1,3,3,2,1,3].normalizeSum)
},

"A()" ->
{
[
"a()", "A()", "c()b()", "j()", "c()j()"

```

```
.wchoose([1,1,2,2,3].normalizeSum)
},
```

```
"b()" ->
{
[
"c()", "A()", "j()"
].choose
},
```

```
"c()" ->
{
[
"a()", "A()", "b()", "a()b()", "j()"
].choose
},
```

```
"j()" ->
{
[
"a()", "A()", "b()", "c()", "j()"
].choose
}
]
);
```

```
~arregloSL1 = [];
```

```
~sisEle1_Dict = Dictionary[
```

```
\a -> "[-]", // tenuto, cr a acento moderado, modificar notación
```

```
\A -> "[>]", // acento ordinario
```

```
\b -> "[sfz]", // mayor acentuación
```

```
\c -> "[x^ c]", // cambio de posición de arco, modificar manualmente
```

```
\j -> "[x^ j]", // jeté sin acentuar
```

```
];
```

```

// para llenar un arreglo lo suficientemente largo para cubrir la primera fase
30.do{ |i|

~sisEle1.applyRules;
~arregloSL1 = ~arregloSL1 ++
~sisEle1.giveParsedString.flat.reject(_isNumber);

};

~seqSL1 = Pseq(~arregloSL1,1).asStream;

~sL1 = ~seqSL1.collect{|x| ~sisEle1_Dict[x]}.asStream;
// a = ~sL1.nextN(~numEventos);
// a.includes(_==nil)
// a.size
// Fase 2
// acentos moderados ordinarios agresivos
// cambios de presión de arco (sin llegar a máxima presión)
// armónico artificial
// opuesto posición de arco
// cuerdas doble

// C armónico artificial
// d cuerdas doble
//

~sisEle2 = LSys(
// axioma
"a()",

// reglas de producción
[

"a()" ->
{
[
"A()", "A(C)", "b()", "b(C)", "j()"

```

```
].choose
```

```
},
```

```
“A()” ->
```

```
{
```

```
[
```

```
“b()”, “b()C()”, “C()”, “d()”, “f()”, “F()”, “g()”, “j()”, “J()”, “k()”, “K()”
```

```
].choose
```

```
},
```

```
“b()” ->
```

```
{
```

```
[
```

```
“a()”, “C()”, “d()”, “f()”, “F()”, “g()”, “d()g()”, “G()”, “h()”, “j()”, “J()”, “k()”, “K()”
```

```
].choose
```

```
},
```

```
“C()” ->
```

```
{
```

```
[
```

```
“b()”, “f()”, “F()”, “G()”, “g()”, “h()”, “e()”, “E()”, “j()”, “J()”
```

```
].choose
```

```
},
```

```
“d()” ->
```

```
{
```

```
[
```

```
“d()”, “f()”, “g()”, “h()”, “e()”, “E()”, “j()”, “J()”, “k()”, “K()”
```

```
].choose
```

```
},
```

```
“e()” ->
```

```
{
```

```
[
```

```
“e()E()”, “E()e()”, “b()”, “B()”, “h()”, “J()”, “K()”
```

```
].choose
```

```
},
```

```
"E()" ->
{
[
"e()E()", "E()e()", "b()", "B()", "h()", "J()", "K()"
].choose
},
```

```
"f()" ->
{
[
"A()", "b()", "C()", "d()", "F()"
].choose
},
```

```
"F()" ->
{
[
"A()", "b()", "C()", "d()", "f()", "g()", "G()", "h()"
].choose
},
```

```
"g()" ->
{
[
"A()", "b()", "g()", "d()", "h()", "J()", "K()"
].choose
},
```

```
"G()" ->
{
[
"A()", "b()", "F()", "C()"
].choose
},
```



```

“h()” ->
{
[
“A()”, “b()”, “d()”, “g()”, “J()”, “K()”
].choose
},

```

```

“j()” ->
{
[
“A()”, “b()”, “C()”, “d()”, “e()”, “E()”, “j()”, “J()”, “k()”, “K()”
].choose
},

```

```

“J()” ->
{
[
“b()”, “d()”, “e()”, “G()”, “J()”, “J()J()”, “J()K()”, “k()”, “K()”
].choose
},

```

```

“k()” ->
{
[
“b()”, “d()”, “e()”, “G()”, “J()”, “J()K()”, “k()”, “K()”
].choose
},

```

```

“K()” ->
{
[
“b()”, “d()”, “e()”, “G()”, “J()”, “J()J()”, “J()K()”, “k()”, “K()”, “K()K()”
].choose
}
]
);

```

```

~sisEle2_Dict = Dictionary[
\la -> "[~]", // tenuto, cr a acento moderado, modificar notación
\A -> "[>]", // acento ordinario
\b -> "[sfz]", // mayor acentuación
\B -> "[sffz]", // acentuación agresiva
\c -> "[x^ c]", // cambio de posición de arco, modificar manualmente
\C -> "[artharm-sounding]", // armónico artificial
\d -> "[x^ d]", // cuerdas doble
\e -> "[x^ A3]", // arpeggio triple (baja probabilidad)
\E -> "[x^ A4]", // arpeggio cuádruple (baja probabilidad)
\f -> "[>][x^ c]", // ajustar acentuación manualmente
\F -> "[>][artharm-sounding]",
\g -> "[>][x^ d]",
\G -> "[>][x^ c][artharm-sounding]",
\h -> "[>][x^ c+d]",
\j -> "[x^ j]", // jeté sin acentuación
\J -> "[>][x^ j]", // jeté con acentuación
\k -> "[>][x^ j+c]", // jeté con acentuación y cambio de posición de arco
\K -> "[>][x^ j+d]" // jeté con acentuación sobre cuerdas dobles
];
~arregloSL2 = [];
// para llenar un arreglo lo suficientemente largo para cubrir la primera fase
80.do{ |i|

~sisEle2.applyRules;
~arregloSL2 = ~arregloSL2 ++
~sisEle2.giveParsedString.flat.reject(_isNumber);

};
~seqSL2 = Pseq(~arregloSL2, 1).asStream;

~sL2 = ~seqSL2.collect{|x| ~sisEle2_Dict[x]}.asStream;

/*a = ~sL2.nextN(100)
a.includes(_=="nil")*/
~sisEle3 = LSys(

```

```
// axioma
```

```
"a()",
```

```
// reglas de producción
```

```
[
```

```
"a()" ->
```

```
{
```

```
[
```

```
"A()", "A()C()", "b()", "b()C()", "B()", "j()"
```

```
].choose
```

```
},
```

```
"A()" ->
```

```
{
```

```
[
```

```
"b()", "b()C()", "B()", "C()", "d()", "f()", "F()", "g()", "j()", "J()"
```

```
].choose
```

```
},
```

```
"b()" ->
```

```
{
```

```
[
```

```
"a()", "B()", "C()", "d()", "f()", "F()", "g()", "d()g()", "G()", "h()", "J()", "K()"
```

```
].choose
```

```
},
```

```
"B()" ->
```

```
{
```

```
[
```

```
"b()", "B()", "C()", "e()", "E()", "g()", "h()", "J()", "K()"
```

```
].choose
```

```
},
```

```
"C()" ->
```

```
{
```

```
[
```

```

“b()”, “f()”, “F()”, “G()”, “g()”, “h()”, “j()”
].choose
},

```

```

“d()” ->
{
[
“d()”, “f()”, “g()”, “h()”, “e()”, “E()”, “J()”, “K()”
].choose
},

```

```

“e()” ->
{
[
“e()e()e()”, “e()E()”, “E()e()”, “b()”, “B()”, “h()”, “j()”, “J()”, “k()”, “K()”
].choose
},

```

```

“E()” ->
{
[
“e()E()”, “E()e()”, “E()E()E()”, “b()”, “B()”, “h()”, “j()”, “J()”, “k()”, “K()”
].choose
},

```

```

“f()” ->
{
[
“A()”, “b()”, “C()”, “d()”, “F()”, “e()”, “E()”
].choose
},

```

```

“F()” ->
{
[
“A()”, “b()”, “C()”, “d()”, “f()”, “g()”, “G()”, “h()”, “e()”, “E()”

```

```
].choose
```

```
},
```

```
“g()” ->
```

```
{
```

```
[
```

```
“A()”, “b()”, “g()”, “d()”, “h()”, “e()”, “E()”, “J()”, “K()”
```

```
].choose
```

```
},
```

```
“G()” ->
```

```
{
```

```
[
```

```
“A()”, “b()”, “F()”, “C()”, “e()”, “E()”
```

```
].choose
```

```
},
```

```
“h()” ->
```

```
{
```

```
[
```

```
“A()”, “b()”, “d()”, “g()”, “e()”, “E()”, “j()”, “J()”, “K()”, “k()”
```

```
].choose
```

```
},
```

```
“j()” ->
```

```
{
```

```
[
```

```
“A()”, “b()”, “B()”, “C()”, “d()”, “e()”, “E()”, “j()”, “J()”, “k()”, “K()”
```

```
].choose
```

```
},
```

```
“J()” ->
```

```
{
```

```
[
```

```
“b()”, “B()”, “d()”, “e()”, “G()”, “J()”, “J()J()”, “J()K()”, “k()”, “K()”
```

```
].choose
},
```

```
"k()" ->
{
[
"b()", "B()", "d()", "e()", "G()", "J()", "J()K()", "k()", "K()"
].choose
},
```

```
"K()" ->
{
[
"b()", "B()B()", "B()", "d()", "e()", "G()", "J()", "J()J()", "J()K()", "k()", "K()", "K()K()"
].choose
}
]
);
```

```
~sisEle3_Dict = Dictionary[
\|a -> "[-]", // tenuto, cr a acento moderado, modificar notación
\|A -> "[>]", // acento ordinario
\|b -> "[sfz]", // mayor acentuación
\|B -> "[sffz]", // acentuación agresiva
\|c -> "[x^ c]", // cambio de posición de arco, modificar manualmente
\|C -> "[artharm-sounding]", // armónico artificial
\|d -> "[x^ d]", // cuerdas doble
\|e -> "[x^ A3]", // arpeggio triple
\|E -> "[x^ A4]", // arpeggio cuádruple
\|f -> "[>][x^ c]", // ajustar acentuación manualmente
\|F -> "[>][artharm-sounding]",
\|g -> "[>][x^ d]",
\|G -> "[>][x^ c][artharm-sounding]",
\|h -> "[>][x^ c+d]",
\|j -> "[x^ j]", // jeté sin acentuación
\|J -> "[>][x^ j]", // jeté con acentuación
```

```

\k -> "[>][x^ j+c]", // jeté con acentuación y cambio de posición de arco
\K -> "[>][x^ j+d]" // jeté con acentuación sobre cuerdas dobles
];
~arregloSL3 = [];
// para llenar un arreglo lo suficientemente largo para cubrir la tercera fase
25.do{ |i|

~sisEle3.applyRules;
~arregloSL3 = ~arregloSL3 ++
~sisEle3.giveParsedString.flat.reject(_isNumber);

};

~seqSL3 = Pseq(~arregloSL3,1).asStream;

~sL3 = ~seqSL3.collect{|x| ~sisEle3_Dict[x]}.asStream;

~mark = Routine{
(~media.size + ~mes).do{|i|

case
{~psn[i] == 0}
{" ".yield}

{(~psn[i] > 0).and(i < ~aurFase[0])}
{ ~sL1.next.yield}

{(~psn[i] > 0).and(i >= ~aurFase[0]).and(i < ~aurFase[1])}
{~sL2.next.yield}

{(~psn[i] > 0).and(i >= ~aurFase[1])}
{~sL3.next.yield}

}
};

// Fomus

```

```
~vln = Pbind(  
  \midinote, ~expOrBrown,  
  \dur, ~durVal,  
  \mark, ~mark  
).asStream;  
  
// Espera a que finalice de procesar la rutina  
  
)  
// ~vln.next()  
  
x = Fomus2(~vln, (~media.size + ~mes)).instrumentName_("violin");  
x.xml;  
x.ly;
```



## CÓDIGO FUENTE ANAMNESIS INTERVENCIÓN

Se necesita descargar e instalar la extensión SCMIR de Nick Collins en la siguiente página: <http://composerprogrammer.com/code.html>

```
//=====
// Intervención Anamnesis=====
//=====
(

// Cargar Buffers

// cambiar archivo
~b1 = Buffer.read(s,
"/Users/JETGE/Documents/misSonidos/vlnBoost/ricochetArmVol.wav");
~a1 = SCMIRAudioFile(
"/Users/JETGE/Documents/misSonidos/vlnBoost/ricochetArmVol.wav", [Loudness]);
~a1.extractFeatures(); // hay que darle una fracción de segundo para que analice el
archivo

~b2 = Buffer.read(s, "/Users/JETGE/Documents/misSonidos/samples/chimes.wav");
~a2 = SCMIRAudioFile("/Users/JETGE/Documents/misSonidos/samples/chimes.wav",
[Loudness]);
~a2.extractFeatures();
~b3 =
Buffer.read(s, "/Users/JETGE/Documents/misSonidos/samples/velociRescoVol.wav" );
~a3 =
SCMIRAudioFile("/Users/JETGE/Documents/misSonidos/samples/velociRescoVol.wav",
[Loudness]);
~a3.extractFeatures();

// "/Users/JETGE/Documents/misSonidos/instaLeiChon/JuanGelman.wav"

/*
"/Users/JETGE/Documents/misSonidos/samples/velociRescoVol.wav"
```

```

~a1.save("/Users/JETGE/Desktop/vln.scmirZ");
~a1.load("/Users/JETGE/Desktop/vln.scmirZ");

~a2.save("/Users/JETGE/Desktop/chimes.scmirZ");
~a2.load("/Users/JETGE/Desktop/chimes.scmirZ");*/

)

(
SynthDef(\gllissonCloud2, { |out=0, freq1=440, freq2=660, sustain=0.05, level=0.1,
amp=0.5,pan=30, envwide=0.5, filepos, buf|

var slopetime = (1 - envwide) * 0.5;

var sound = PlayBuf.ar(1, buf, XLine.ar(freq1, freq2, sustain), 1, filepos);

// Envoltentes de
gránulos=====

var env = EnvGen.ar(
Env([0, level, level, 0], [slopetime, envwide, slopetime], \sin),
timeScale: sustain,
doneAction: 2) * AmpComp.ir(freq1/freq2); // Con compensación psicoacústica de
amplitud ligado a freq
OffsetOut.ar(out, PanAz.ar(2, (sound * env)*amp, pan)); // Cuadra

}, \ir !8
).add;
~arIndMax = [];

~a1.featuredata.collect({|x| if(x > 0.7) {
~arIndMax = ~arIndMax ++ ~a1.featuredata.indicesOfEqual(x)}});
~arIndMed = [];

~a1.featuredata.collect({|x| if( (x > 0.25) && (x < 0.4))
{~arIndMed = ~arIndMed ++ ~a1.featuredata.indicesOfEqual(x)}});
~arIndMin = [];

```

```

~a1.featuredata.collect({|x| if((x < 0.25) && (x > 0.1)) {
~arIndMin = ~arIndMin ++ ~a1.featuredata.indicesOfEqual(x)}});
~arIndMax2 = [];

~a2.featuredata.collect({|x| if(x > 0.7) {
~arIndMax2 = ~arIndMax2 ++ ~a2.featuredata.indicesOfEqual(x)}});
~arIndMed2 = [];

~a2.featuredata.collect({ |x| if( (x > 0.25) && (x < 0.4))
{~arIndMed2 = ~arIndMed2 ++ ~a2.featuredata.indicesOfEqual(x)}});
~arIndMin2 = [];

~a2.featuredata.collect({|x| if((x < 0.25) && (x > 0.1)) {
~arIndMin2 = ~arIndMin2 ++ ~a2.featuredata.indicesOfEqual(x)}});
~arIndMax3 = [];

~a3.featuredata.collect({|x| if(x > 0.7) {
~arIndMax3 = ~arIndMax3 ++ ~a3.featuredata.indicesOfEqual(x)}});
~arIndMed3 = [];

~a3.featuredata.collect({ |x| if( (x > 0.25) && (x < 0.4))
{~arIndMed3 = ~arIndMed3 ++ ~a3.featuredata.indicesOfEqual(x)}});
~arIndMin3 = [];

~a3.featuredata.collect({|x| if((x < 0.25) && (x > 0.1)) {
~arIndMin3 = ~arIndMin3 ++ ~a3.featuredata.indicesOfEqual(x)}});

//=====
// Rutina para tener tiempos dinámicos

// Transiciones dentro de siete rangos distintos
// extremadamente corta 1 – 30
// corta 30 – 80
// media 80 – 150
// algo lenta 150 – 250
// lenta 250 450
// muy lenta 450 850

```

```

// lo más lento 850 1500
~transInter = Dictionary[

\exCorta -> Pexprand(0.001, 30,inf),
\corta -> Pexprand(30, 80, inf),
\media -> Prand((80..150), inf),
\media2 -> Prand((150..250), inf),
\lenta -> Prand((250..450), inf),
\muylenta -> Phprand(450, 850, inf),
\xlenta -> Phprand(850, 1500, inf)
];
// Especificar probabilidades
// valores actuales arbitrarios

~tempDin = Pfsm([
#[0],

\exCorta, #[1, 6],
\corta, #[1, 0, 2, 3],
\media, #[0, 2, 3],
\media2, #[5, 1, 6],
\lenta, #[0, 6],
\muylenta, #[0, 1, 2, 3, 4, 5, 6],
\xlenta, #[0, 5],

nil, nil

],inf).asStream;
// =====
// ===== Transiciones //////////////////////////////////////
// =====

~trans = Dictionary[

\gaussMin -> Pgauss(~arIndMin[~arIndMin.size.rand], 30, inf).abs,
\gaussMed -> Pgauss(~arIndMed[~arIndMed.size.rand], 30, inf).abs,
\gaussMax -> Pgauss(~arIndMax[~arIndMax.size.rand], 30, inf).abs,

```

```

\envIndValMin -> Prout({ loop { ~arIndMin[~arIndMin.size.rand].yield }}),
\envIndValMed -> Prout({ loop { ~arIndMed[~arIndMed.size.rand].yield }}),
\envIndValMax -> Prout({ loop { ~arIndMax[~arIndMax.size.rand].yield }})

];
~transFSM = P fsm([
#[0],

\gaussMin, #[0, 0, 4, 5],
\gaussMed, #[1, 1, 2, 4, 5],
\gaussMax, #[2, 2, 0, 3, 4],
\envIndValMin, #[3, 3, 2, 5],
\envIndValMed, #[4, 4, 0, 2],
\envIndValMax, #[5, 5, 0, 2, 3],

nil, nil

], inf).asStream;
// =====
// =====

~lazee =
Pn(
Plazy({ var transiciones, tiempos;

transiciones = (~trans[~transFSM.next]).asStream.nextN(7);
tiempos = (~transInter[~tempDin.next]).asStream.nextN(6);

Pseq(
Env(transiciones, tiempos,
).asSignal(tiempos.sum),
1)
}
), inf).asStream;
/*~sustain = Dictionary[

```

```
\sinOsc -> Prout({ var sin = PSinOsc(1000, 0, 1.99 , 2, inf).asStream;
loop{ sin.next.yield }}).asStream,
```

```
\clip -> Prout({ var clip = PClipNoise(0.99,1.0,inf).asStream;
loop { clip.next.yield }}).asStream,
```

```
\exp -> Prout({ var exp = Pexprand(0.001, 0.5, inf).asStream;
loop{ exp.next.yield }}).asStream,
```

```
\lfp -> Prout({ var pulse;
pulse = PLFPulse(50, 0,
Pstutter(20,
Pseq([(0.1, 0.2 .. 1.0), (1.0,0.9 .. 0.1)].flat,
inf)), 1.0, 0.1,
inf).asStream;
loop{ pulse.next.yield}}).asStream,
```

```
\lfs -> Prout({ var saw = PLFSaw(40, 0, 0.49, 0.5, inf).asStream;
loop { saw.next.yield}}).asStream
];*/
```

```
~sustain = Dictionary[
```

```
\sinOsc -> Prout({ var sin = Pseq(
Pseries.fromEndpoints(3.0, 0.355, 1000).asStream.all
++
Pseries.fromEndpoints(0.35, 3.0, 1000).asStream.all
,
inf).asStream;
loop{ sin.next.yield }}).asStream,
```

```
\clip -> Prout({ var clip = Pbeta(0.35, 3.0, 0.1, 0.1, inf).asStream;
loop{ clip.next.yield }}).asStream,
```

```
\exp -> Prout({ var exp = Pexprand(0.35, 3.0, inf).asStream;
loop{ exp.next.yield }}).asStream,
```

```

/* \lfp -> Pn(
Plazy({
Prout({
var stutNum = Pseq((1..40),inf).asStream;
var stooter = Pstutter(stutNum, Pseq([0.35, 2.0], inf)).asStream;

loop{ stooter.next.yield}}).asStream;

}),
inf).asStream,*/

\lfp ->Prout({ var saw = Pseq(Pseries.fromEndpoints(0.35, 3.0,
1000).asStream.all,inf).asStream;
loop { saw.next.yield }
}).asStream,
//
\lfs -> Prout({ var saw = Pseq(Pseries.fromEndpoints(3.0, 0.35,
1000).asStream.all,inf).asStream;
loop { saw.next.yield }
}).asStream

];
~susIndepeFSM =

P fsm([
#[0],

\sinOsc, #[0, 0, 1, 2, 3],
\clip, #[1, 2, 3],
\exp, #[4, 3, 2],
\lfp, #[0, 4],
\lfs, #[3, 4],
nil, nil

]).asStream;
// Hacer de acuerdo al symbol escogido
~susRout = Prout({ var stutter = Pstutter( 300, Prout({ loop{

```

```

~susIndepeFSM.next.yield}}).asStream;
loop { ~sustain[stutter.next].next.yield}}).asStream;

// =====
// =====
// =====

//=====
// Rutina para tener tiempos dinámicos 2

// Transiciones dentro de siete rangos distintos
// extremadamente corta 1 – 30
// corta 30 – 80
// media 80 – 150
// algo lenta 150 – 250
// lenta 250 450
// muy lenta 450 850
// lo más lento 850 1500

~transInter2 = Dictionary[

\exCorta -> Pexprand(0.0001, 30,inf),
\corta -> Pexprand(30, 80, inf),
\media -> Prand((80..150), inf),
\media2 -> Prand((150..250), inf),
\lenta -> Prand((250..450), inf),
\muylenta -> Phprand(450, 850, inf),
\lexlenta -> Phprand(850, 1500, inf)
];
// Especificar probabilidades de acuerdo a una idea
// valores actuales arbitrarios

~tempDin2 = P fsm([
#[0],

\exCorta, #[1, 6],
\corta, #[1, 0, 2, 3],

```



```

\media, #[0, 2, 3],
\media2, #[5, 1, 6],
\lenta, #[0, 6],
\muylenta, #[0, 1, 2, 3, 4, 5, 6],
\exlenta, #[0, 5],

nil, nil

],inf).asStream;
// =====
// ===== Transiciones //////////////////////////////////////
// =====

~trans2 = Dictionary[

\gaussMin -> Pgauss(~arIndMin2[~arIndMin2.size.rand], 30, inf).abs,
\gaussMed -> Pgauss(~arIndMed2[~arIndMed2.size.rand], 30, inf).abs,
\gaussMax -> Pgauss(~arIndMax2[~arIndMax2.size.rand], 30, inf).abs,
\envIndValMin -> Prout({ loop { ~arIndMin2[~arIndMin2.size.rand].yield } }),
\envIndValMed -> Prout({ loop { ~arIndMed2[~arIndMed2.size.rand].yield } }),
\envIndValMax -> Prout({ loop { ~arIndMax2[~arIndMax2.size.rand].yield } })

];
~transFSM2 = P fsm([
#[0],

\gaussMin, #[0, 0, 4, 5],
\gaussMed, #[1, 1, 2, 4, 5],
\gaussMax, #[2, 2, 0, 3, 4],
\envIndValMin, #[3, 3, 2, 5],
\envIndValMed, #[4, 4, 0, 2],
\envIndValMax, #[5, 5, 0, 2, 3],

nil, nil

```

```

], inf).asStream;
// =====
// =====

~lazee2 =
Pn(
Plazy({ var transiciones, tiempos;

transiciones = (~trans2[~transFSM2.next]).asStream.nextN(7);
tiempos = (~transInter2[~tempDin2.next]).asStream.nextN(6);

Pseq(
Env(transiciones, tiempos,
).asSignal(tiempos.sum),
1)
}
), inf).asStream;
~sustain2 = Dictionary[

\sinOsc -> Prout({ var sin = PSinOsc(1000, 0, 1.99 , 2, inf).asStream;
loop{ sin.next.yield }}).asStream,

\clip -> Prout({ var clip = PClipNoise(0.99,1.0,inf).asStream;
loop { clip.next.yield }}).asStream,

\exp -> Prout({ var exp = Pexprand(0.001, 0.5, inf).asStream;
loop{ exp.next.yield }}).asStream,

\lfp -> Prout({ var pulse;
pulse = PLFPulse(50, 0,
Pstutter(20,
Pseq([(0.1, 0.2 .. 1.0), (1.0,0.9 .. 0.1)].flat,
inf)), 1.0, 0.1,
inf).asStream;
loop{ pulse.next.yield}}).asStream,

```

```

\lfs -> Prout({ var saw = PLFSaw(40, 0, 0.49, 0.5, inf).asStream;
loop { saw.next.yield}}).asStream
];
~susIndepeFSM2 =

```

```

P fsm([
#[0],

```

```

\sinOsc, #[0, 0, 1, 2, 3],
\clip, #[1, 2, 3],
\exp, #[4, 3, 2],
\lfp, #[0, 4],
\lfs, #[3, 4],
nil, nil

```

```

]).asStream;

```

```

~susRout2 = Prout({ var stutter = Pstutter( 300, Prout({ loop{
~susIndepeFSM2.next.yield}})).asStream;
loop { ~sustain2[stutter.next].next.yield}}).asStream;

```

```

//=====
//=====
//=====

```

```

// Rutina para tener tiempos dinámicos 3

```

```

// Transiciones dentro de siete rangos distintos

```

```

// extremadamente corta 1 – 30

```

```

// corta 30 – 80

```

```

// media 80 – 150

```

```

// algo lenta 150 – 250

```

```

// lenta 250 450

```

```

// muy lenta 450 850

```

```

// lo más lento 850 1500

```

```

~transInter3 = Dictionary[

```

```

\exCorta -> Pexprand(0.0001, 30,inf),
\corta -> Pexprand(30, 80, inf),
\media -> Prand((80..150), inf),
\media2 -> Prand((150..250), inf),
\lenta -> Prand((250..450), inf),
\muylenta -> Phprand(450, 850, inf),
\exlenta -> Phprand(850, 1500, inf)
];
// Especificar probabilidades de acuerdo a una idea
// valores actuales arbitrarios

~tempDin3 = Pfsm([
#[0],

\exCorta, #[1, 6],
\corta, #[1, 0, 2, 3],
\media, #[0, 2, 3],
\media2, #[5, 1, 6],
\lenta, #[0, 6],
\muylenta, #[0, 1, 2, 3, 4, 5, 6],
\exlenta, #[0, 5],

nil, nil

],inf).asStream;
~lazee3 =
Pn(
Plazy({ var transiciones, tiempos;

transiciones = (~trans3[~transFSM3.next]).asStream.nextN(7);
tiempos = (~transInter3[~tempDin3.next]).asStream.nextN(6);

Pseq(
Env(transiciones, tiempos,
).asSignal(tiempos.sum),
1)
}

```

```

), inf).asStream;
// =====
// ===== Transiciones //////////////////////////////////////
// =====

~trans3 = Dictionary[

\gaussMin -> Pgauss(~arIndMin2[~arIndMin2.size.rand], 30, inf).abs,
\gaussMed -> Pgauss(~arIndMed2[~arIndMed2.size.rand], 30, inf).abs,
\gaussMax -> Pgauss(~arIndMax2[~arIndMax2.size.rand], 30, inf).abs,
\envIndValMin -> Prout({ loop { ~arIndMin2[~arIndMin2.size.rand].yield } }),
\envIndValMed -> Prout({ loop { ~arIndMed2[~arIndMed2.size.rand].yield } }),
\envIndValMax -> Prout({ loop { ~arIndMax2[~arIndMax2.size.rand].yield } })

];
~transFSM3 = P fsm([
#[0],

\gaussMin, #[0, 0, 4, 5],
\gaussMed, #[1, 1, 2, 4, 5],
\gaussMax, #[2, 2, 0, 3, 4],
\envIndValMin, #[3, 3, 2, 5],
\envIndValMed, #[4, 4, 0, 2],
\envIndValMax, #[5, 5, 0, 2, 3],

nil, nil

], inf).asStream;
// =====
// =====
//////////////////////////////////// Movimiento de Frecuencia //////////////////////////////////////
//////////////////////////////////// Convergencia Divergencia Estático //////////////////////////////////////
// =====
// =====

// =====
//////////////////////////////////// Divergente //////////////////////////////////////

```

```

// Movimiento divergente , con frecuencia eje móvil
// ligado a freq2

// Función para modular freq2
// Función para modular paso de movimiento browniano de freq1

~freqModDict = Dictionary[

\estatico -> Pstutter(300, Prand((0.1,0.11..2.0),inf),inf).asStream,

\rand -> Prand((0.1,0.11..3.0),inf).asStream,

\brownianoCorto -> Prout({ var brwnC = Pbrown(0.1, 3.0, 0.05,inf).asStream;
loop{ brwnC.next.yield}}).asStream,

\brownianoMediano -> Prout({ var brwnM = Pbrown(0.1, 3.0, 0.125, inf).asStream;
loop{ brwnM.next.yield}}).asStream,

\brownianoAmplio -> Prout({ var brwnA = Pbrown(0.1, 3.0,0.25,inf).asStream;
loop{ brwnA.next.yield}}).asStream,

];
~freqFSM = P fsm([
#[0],

\estatico, #[0, 1, 2],
\rand, #[1, 2, 3, 4],
\brownianoCorto, #[0, 3],
\brownianoMediano, #[0, 3, 4],
\brownianoAmplio, #[0, 1,],
nil, nil

]
).asStream;
~freqRout = Prout({ var freq = Pstutter( 200, Prout({ loop{
~freqFSM.next.yield}})).asStream;
loop { ~freqModDict[freq.next].next.yield}}).asStream;

```

```

// Freq 2 =====

~freqModDict2 = Dictionary[

\estatico -> Pstutter(300, Prand((0.1,0.11..2.0),inf),inf).asStream,

\rand -> Prand((0.1,0.11..3.0),inf).asStream,

\brownianoCorto -> Prout({ var brwnC = Pbrown(0.1, 3.0, 0.05,inf).asStream;
loop{ brwnC.next.yield}}).asStream,

\brownianoMediano -> Prout({ var brwnM = Pbrown(0.1, 3.0, 0.125, inf).asStream;
loop{ brwnM.next.yield}}).asStream,

\brownianoAmplio -> Prout({ var brwnA = Pbrown(0.1, 3.0,0.25,inf).asStream;
loop{ brwnA.next.yield}}).asStream,
// \brwnCortoDir ->
// \brwnMedDir -> BROWNIANOS DIRIGIDOS AQUÍ
// \brwnAmpDir ->
// \brwnCortoDir2 ->
// \brwnMedDir2 ->
// \brwnAmpDir2 ->
// \brwnCortoDir3 ->
// \brwnMedDir3 ->
// \brwnAmpDir3 ->

];
~freqFSM2 = P fsm([
#[0],

\estatico, #[0, 1, 2],
\rand, #[1, 2, 3, 4],
\brownianoCorto, #[0, 3],
\brownianoMediano, #[0, 3, 4],
\brownianoAmplio, #[0, 1,],
nil, nil

```

```

]
).asStream;
~freqRout2 = Prout({ var freq = Pstutter( 200, Prout({ loop{
~freqFSM2.next.yield}})).asStream;
loop { ~freqModDict2[freq.next].next.yield}}).asStream;

// =====
// ===== CONVERGENTE =====
//=====

/*Pdef(\glissCloudProbandoConv,
Pbind(
\instrument, \glissonCloud,
\out, 0,
\pan, Prand((-1.0, -0.999 .. 1.0), inf),
\level, Prand((0.01,0.02..0.2),inf),
// \freq1, Prout({ loop{~freqRout.next.yield}}),
\freq2, Prout({loop{~freqRout.next.yield}}),
\freq1, Prand([(Pkey(\freq2)-0.1).abs,Pkey(\freq2)+0.1],inf),
\sustain, Prout({ loop{~susRout.next.yield}}),
\dur, Pdefn(\gcdur1, Pkey(\sustain)/Prout({loop{~durDivRout.next.yield}})), // Ejecutar
~durDivRout abajo
// \dur, Pkey(\sustain)/~muyL.next,
\filepos, ~lazee.next * 1024
))

Pdef(\glissCloudProbandoConv).stop
Pdef(\glissCloudProbandoConv).play*/

// =====
// ===== Unidireccional =====
//=====

// Ascendente

// Pdef(\glissCloudProbandoUniAsc,
// Pbind(

```



```

// \instrument, \glissonCloud,
// \out, 0,
// \pan, Prand((-1.0, -0.999 .. 1.0), inf),
// \level, Prand((0.01,0.02..0.2),inf),
// \freq1, Prout({ loop{~freqRout.next.yield}}),
// \freq2, Pkey(\freq1) + 0.35, // hacer el factor por
// \sustain, Prout({ loop{~susRout.next.yield}}),
// \dur, Pdefn(\gcdur1, Pkey(\sustain)/Prout({loop{~durDivRout.next.yield}})), // Ejecutar
~durDivRout abajo
// // \dur, Pkey(\sustain)/~muyL.next,
// \filepos, ~lazee.next * 1024
// ))
//
//
//
// Pdef(\glissCloudProbandoUniAsc).play
//
// // Descendente
//
// Pdef(\glissCloudProbandoUniDesc,
// Pbind(
// \instrument, \glissonCloud,
// \out, 0,
// \pan, Prand((-1.0, -0.999 .. 1.0), inf),
// \level, Prand((0.01,0.02..0.2),inf),
// \freq1, Prout({ loop{~freqRout.next.yield}}),
// \freq2, (Pkey(\freq1) - 0.35).abs, // hacer el factor por
// \sustain, Prout({ loop{~susRout.next.yield}}),
// \dur, Pdefn(\gcdur1, Pkey(\sustain)/Prout({loop{~durDivRout.next.yield}})), // Ejecutar
~durDivRout abajo
// // \dur, Pkey(\sustain)/~muyL.next,
// \filepos, ~lazee.next * 1024
// ))
//
//

```

```

//
// Pdef(\glissCloudProbandoUniDesc).play

// =====
// //////////// DIVISIÓN DE SUSTAIN EN DUR ////////////
// =====

// Ajustar valores de numero de repeticiones
// de Pbrown dentro de una función

~durSusDiv = (

\muyL: Prout({

var muyL = Pbrown(1, 4, 0.125, inf).asStream;
var rep = Prand((1..4),inf).asStream;
var cuentaMax = rep.next;

(cuentaMax+1).do{ arg x;
if( x < cuentaMax, {muyL.next.yield;}, { nil.yield; });

}).asStream,
\lento: Prout({

var lento = Pbrown(4, 7, 0.125, inf).asStream;
var rep = Prand((3..5),inf).asStream;
var cuentaMax = rep.next;

(cuentaMax+1).do{ arg x;
if(x < cuentaMax, {lento.next.yield;}, { nil.yield; });

}).asStream,

\medLen: Prout({

var medLen = Pbrown(10, 25, 0.125, inf).asStream;
var rep = Prand((3..5),inf).asStream;
var cuentaMax = rep.next;

```

```
(cuentaMax+1).do{ arg x;
if( x < cuentaMax, {medLen.next.yield;}, { nil.yield;});
```

```
}).asStream,
```

```
\medMed: Prout({
```

```
var medMed = Pbrown(10, 13, 0.125, inf).asStream;
```

```
var rep = Prand((10..25),inf).asStream;
```

```
var cuentaMax = rep.next;
```

```
(cuentaMax+1).do{ arg x;
```

```
if( x < cuentaMax, {medMed.next.yield;}, { nil.yield;});
```

```
}).asStream,
```

```
\pocoR: Prout({
```

```
var pocoR = Pbrown(12, 17, 0.125, inf).asStream;
```

```
var rep = Prand((20..50),inf).asStream;
```

```
var cuentaMax = rep.next;
```

```
(cuentaMax+1).do{ arg x;
```

```
if( x < cuentaMax, {pocoR.next.yield;}, { nil.yield;});
```

```
}).asStream,
```

```
\rapido: Prout({
```

```
var rapido = Pbrown(17, 22, 0.125, inf).asStream;
```

```
var rep = Prand((30..100),inf).asStream;
```

```
var cuentaMax = rep.next;
```

```
(cuentaMax+1).do{ arg x;
```

```
if( x < cuentaMax, {rapido.next.yield;}, { nil.yield;});
```

```
}).asStream,
```

```

\muyR: Prout({

var muyR = Pbrown(20, 25, 0.125, inf).asStream;
var rep = Prand((30..100),inf).asStream;
var cuentaMax = rep.next;

(cuentaMax+1).do{ arg x;
if( x < cuentaMax, {muyR.next.yield;}, { nil.yield;});

}).asStream;
);
~durSusDivRout = (

\ml: Prout({

inf.do{ var hey = ~durSusDiv.muyL.next;
if( hey.isNumber, {hey.yield}, {~durSusDiv.muyL.reset.yield; nil.yield}})}).asStream,

\l: Prout({

inf.do{ var hey = ~durSusDiv.lento.next;
if( hey.isNumber, {hey.yield}, {~durSusDiv.lento.reset.yield; nil.yield}})}).asStream,

\medl: Prout({

inf.do{ var hey = ~durSusDiv.medLen.next;
if( hey.isNumber, {hey.yield}, {~durSusDiv.medLen.reset.yield; nil.yield}})}).asStream,

\mm: Prout({

inf.do{ var hey = ~durSusDiv.medMed.next;
if( hey.isNumber, {hey.yield}, {~durSusDiv.medMed.reset.yield; nil.yield}})}).asStream,

\pr: Prout({

inf.do{ var hey = ~durSusDiv.pocoR.next;
if( hey.isNumber, {hey.yield}, {~durSusDiv.pocoR.reset.yield; nil.yield}})}).asStream,

```

```

\ra: Prout({

inf.do{ var hey = ~durSusDiv.rapido.next;
if( hey.isNumber, {hey.yield}, {~durSusDiv.rapido.reset.yield; nil.yield})}}).asStream,

\mra: Prout({

inf.do{ var hey = ~durSusDiv.muyR.next;
if( hey.isNumber, {hey.yield}, {~durSusDiv.muyR.reset.yield; nil.yield})}}).asStream,

);

~durSusFSM = P fsm([
#[2],

\ml, #[2, 6],
\l, #[5, 3],
\medl, #[1, 6],
\mm, #[2, 3, 4],
\pr, #[0, 6],
\ra, #[6, 5, 4, 1],
\mra, #[6, 5, 4, 3, 2, 1, 0],
nil, nil
]

).asStream;
~durDivRout = Prout({
var durSus = Prout({
inf.do{~durSusDivRout[~durSusFSM.next].embedInStream}}).asStream;

loop{

durSus.select(_isNumber).next.yield;
// durSus.next.yield;

};

```

```

}).asStream;

~dicionPaneo = (

\brwnL: Prout({

var brownL = Pbrown(-1.0, 1.0, 0.05, inf).asStream;
var rep = Prand((50..100),inf).asStream;
var cuentaMax = rep.next;

(cuentaMax+1).do{ arg x;
if( x < cuentaMax, {brownL.next.yield;}, { nil.yield; } )}}).asStream,

\brwnM: Prout({

var brownM = Pbrown(-1.0, 1.0, 0.2, inf).asStream;
var rep = Prand((50..100),inf).asStream;
var cuentaMax = rep.next;

(cuentaMax+1).do{ arg x;
if( x < cuentaMax, {brownM.next.yield;}, { nil.yield; } )}}).asStream,

\brwnR: Prout({

var brownR = Pbrown(-1.0, 1.0, 0.5, inf).asStream;
var rep = Prand((50..100),inf).asStream;
var cuentaMax = rep.next;

(cuentaMax+1).do{ arg x;
if( x < cuentaMax, {brownR.next.yield;}, { nil.yield; } )}}).asStream,

\white: Prout({

var white = Pwhite(-1.0, 1.0, inf).asStream;
var rep = Prand((50..100),inf).asStream;
var cuentaMax = rep.next;

```

```
(cuentaMax+1).do{ arg x;
if( x < cuentaMax, {white.next.yield;}, { nil.yield; } )}}).asStream,
```

```
\seqL1: Prout({ // unidireccional a la Saw
```

```
var seqL = Pseq( Array.series(100, -1.0, 0.01) ++ Array.series(101, 0, 0.01),
inf).asStream;
var rep = Prand((50..100),inf).asStream;
var cuentaMax = rep.next;
```

```
(cuentaMax+1).do{ arg x;
if( x < cuentaMax, {seqL.next.yield;}, { nil.yield; } )}}).asStream,
```

```
\seqM1: Prout({ // unidireccional
```

```
var seqM = Pseq(Pseries.fromEndpoints(-1.0, 1.0, 50).asStream.all , inf).asStream;
var rep = Prand((50..100),inf).asStream;
var cuentaMax = rep.next;
```

```
(cuentaMax+1).do{ arg x;
if( x < cuentaMax, {seqM.next.yield;}, { nil.yield; } )}}).asStream,
```

```
\seqR1: Prout({ // unidireccional
```

```
var seqR = Pseq(Pseries.fromEndpoints(-1.0,1.0,15).asStream.all, inf).asStream;
var rep = Prand((50..100),inf).asStream;
var cuentaMax = rep.next;
```

```
(cuentaMax+1).do{ arg x;
if( x < cuentaMax, {seqR.next.yield;}, { nil.yield; } )}}).asStream,
```

```
\seqL2: Prout({ // Bidireccional a la SinOsc
```

```
var seqL = Pseq(
Pseries.fromEndpoints(-1.0, 1.0, 201).asStream.all ++
Pseries.fromEndpoints(1.0,-1.0,201).asStream.all,
inf).asStream;
```

```
var rep = Prand((50..100),inf).asStream;
var cuentaMax = rep.next;
```

```
(cuentaMax+1).do{ arg x;
if( x < cuentaMax, {seqL.next.yield;}, { nil.yield; } )}}.asStream,
```

```
\seqM2: Prout({ // Bidireccional a la SinOsc
```

```
var seqL = Pseq(Pseries.fromEndpoints(-1.0, 1.0, 50).asStream.all ++
Pseries.fromEndpoints(1.0, -1.0, 50).asStream.all,inf);
var rep = Prand((50..100),inf).asStream;
var cuentaMax = rep.next;
```

```
(cuentaMax+1).do{ arg x;
if( x < cuentaMax, {seqL.next.yield;}, { nil.yield; } )}}.asStream,
```

```
\seqR2: Prout({ // Bidireccional a la SinOsc
```

```
var seqL = Pseq(Array.series(10, -1.0, 0.1) ++ Array.series(10, 0, 0.1), inf).asStream;
var rep = Prand((50..100),inf).asStream;
var cuentaMax = rep.next;
```

```
(cuentaMax+1).do{ arg x;
if( x < cuentaMax, {seqL.next.yield;}, { nil.yield; } )}}.asStream,
```

```
\betaExt: Prout({ // Distribución Beta Valores extremos
```

```
var betaExt = Pbeta(-1.0, 1.0, 0.1, 0.1, inf).asStream;
var rep = Prand((50..100),inf).asStream;
var cuentaMax = rep.next;
```

```
(cuentaMax+1).do{ arg x;
if( x < cuentaMax, {betaExt.next.yield;}, { nil.yield; } )}}.asStream,
```

```
\betaUni: Prout({ // Distribución Beta , uniforme
```



```

var betaUni = Pbeta(-1.0, 1.0, 1.0, 1.0, inf).asStream;
var rep = Prand((50..100),inf).asStream;
var cuentaMax = rep.next;

(cuentaMax+1).do{ arg x;
if( x < cuentaMax, {betaUni.next.yield;}, { nil.yield; } )}}).asStream,

\betaGauss: Prout({ // Distribución Beta , quasi-Gauss

var betaGauss = Pbeta(-1.0, 1.0, 3.0, 3.0, inf).asStream;
var rep = Prand((50..100),inf).asStream;
var cuentaMax = rep.next;

(cuentaMax+1).do{ arg x;
if( x < cuentaMax, {betaGauss.next.yield;}, { nil.yield; } )}}).asStream,

\seqRand: Prout({

var randNum = rrand(5, 30);
var seqRand = Pseq(Array.rand(randNum, -1.0, 1.0),inf).asStream;
var rep = Prand((50..100),inf).asStream;
var cuentaMax = rep.next;

(cuentaMax+1).do{ arg x;
if( x < cuentaMax, {seqRand.next.yield;}, { nil.yield; } )}}).asStream,
);
~diccionPaneoRout = (

\brl: Prout({ // browniano lento

inf.do{ var hey = ~diccionPaneo.brwnL.next;
if( hey.isNumber, {hey.yield}, {~diccionPaneo.brwnL.reset.yield; nil.yield} )}}).asStream,

\brm: Prout({ // browniano mediano

inf.do{ var hey = ~diccionPaneo.brwnM.next;
if( hey.isNumber, {hey.yield}, {~diccionPaneo.brwnM.reset.yield; nil.yield} )}}).asStream,

```

\brr: Prout({ // browniano rápido

```
inf.do{ var hey = ~diccionPaneo.brwnR.next;
if( hey.isNumber, {hey.yield}, {~diccionPaneo.brwnR.reset.yield; nil.yield}})).asStream,
```

\sl1: Prout({ // seq lento unidireccional

```
inf.do{ var hey = ~diccionPaneo.seqL1.next;
if( hey.isNumber, {hey.yield}, {~diccionPaneo.seqL1.reset.yield; nil.yield}})).asStream,
```

\sm1: Prout({ // seq medio unidireccional

```
inf.do{ var hey = ~diccionPaneo.seqM1.next;
if( hey.isNumber, {hey.yield}, {~diccionPaneo.seqM1.reset.yield; nil.yield}})).asStream,
```

\sr1: Prout({ // seq rapido unidireccional

```
inf.do{ var hey = ~diccionPaneo.seqR1.next;
if( hey.isNumber, {hey.yield}, {~diccionPaneo.seqR1.reset.yield; nil.yield}})).asStream,
```

\sl2: Prout({ // seq lento bidireccional

```
inf.do{ var hey = ~diccionPaneo.seqL2.next;
if( hey.isNumber, {hey.yield}, {~diccionPaneo.seqL2.reset.yield; nil.yield}})).asStream,
```

\sm2: Prout({ // seq medio bidireccional

```
inf.do{ var hey = ~diccionPaneo.seqM2.next;
if( hey.isNumber, {hey.yield}, {~diccionPaneo.seqM2.reset.yield; nil.yield}})).asStream,
```

\sr2: Prout({ // seq rapido bidireccional

```
inf.do{ var hey = ~diccionPaneo.seqR2.next;
if( hey.isNumber, {hey.yield}, {~diccionPaneo.seqR2.reset.yield; nil.yield}})).asStream,
```

\btx: Prout({ // distribución beta, extremos

```
inf.do{ var hey = ~dicionPaneo.betaExt.next;
if( hey.isNumber, {hey.yield}, {~dicionPaneo.betaExt.reset.yield; nil.yield}})}).asStream,
```

```
\btu: Prout({ // distribución beta, uniforme
```

```
inf.do{ var hey = ~dicionPaneo.betaUni.next;
if( hey.isNumber, {hey.yield}, {~dicionPaneo.betaUni.reset.yield; nil.yield}})}).asStream,
```

```
\btg: Prout({ // distribución beta, quasi-gauss
```

```
inf.do{ var hey = ~dicionPaneo.betaGauss.next;
if( hey.isNumber, {hey.yield}, {~dicionPaneo.betaGauss.reset.yield;
nil.yield}})}).asStream,
```

```
\srd: Prout({ // distribución beta, quasi-gauss
```

```
inf.do{ var hey = ~dicionPaneo.seqRand.next;
if( hey.isNumber, {hey.yield}, {~dicionPaneo.seqRand.reset.yield;
nil.yield}})}).asStream,
```

```
);
```

```
// 13 opciones en total
```

```
~paneofsm = P fsm([
#[0],
```

```
\brl, #[0,1,2,3],
```

```
\brm, #[2,3,4,5],
```

```
\brr, #[5,6,7,8],
```

```
\sl1, #[11,10,9,8],
```

```
\sm1, #[10,9,8,7],
```

```
\sr1, #[0,2,4,6],
```

```
\sl2, #[8,10],
```

```
\sm2, #[0,1,2,3,4,12,5],
```

```
\sr2, #[6,7,8,9,12],
```

```
\btx, #[11,10,9],
```

```

\btu, #[0,1,2,3],
\btg, #[5,6,7,8,12],
\srd, #[12, 11, 10, 9],

nil, nil

]
).asStream;
// ~paneoFSM.next

~paneoRout = Prout({

var paneo = Prout({ inf.do{
~diccionPaneoRout[~paneoFSM.next].embedInStream}}).asStream;

loop{

paneo.select(_isNumber).next.yield;
}

}).asStream;
)
(

// probando
Pdef(\glissCloudProbandoConv,
Pbind(
\instrument, \glissonCloud2,
\pan, Prout({loop{~paneoRout.next.yield}}),
\level, Prand((0.01,0.02..0.2),inf),
\freq2, Prout({loop{~freqRout.next.yield}}),
\freq1, Prand([Pkey(\freq2)-0.1,Pkey(\freq2)+0.1],inf),
\sustain, Prout({ loop{~susRout.next.yield}}),
\dur, Pdefn(\gcdur1, Pkey(\sustain)/Prout({loop{(~durDivRout.next).yield}})),
\filepos, ~lazee.next * 1024
));

```

```

Pdef(\glissCloudProbandoConv).trace.play
)

// Algunos ejemplos =====

(
~t1bind1 = Pbind(
\instrument, \glissonCloud2,
\buf, ~b2,
\level, 0.3,
\amp, 0.1,
\freq1, 0.15,
\freq2, Prand([0.14, 0.16],inf),
\sustain, Prout({loop{~susRout.next.yield}}),
\pan, Prout({loop{~paneoRout.next.yield}}),
\filepos, Prout({loop{(~lazee.next * 1024).yield}}),
\dur, 0.05
);

Pdef(\cloudText1).fadeTime = 10; // FADE TIME

Pdef(\cloudText1, ~t1bind1);
Pdef(\cloudText1).play;
Pdef(\cloudText1).isPlaying;
)

Pdef(\cloudText1).stop

// ~gc3 = PdefGui();
// ~gc3.object = Pdef(\cloudText1)

~t1bind2 = Pbind(
\instrument, \glissonCloud2,
\buf, ~b3,
\level, 0.4,

```

```

\amp, 0.05,
\freq1, 2.25,
\freq2, Prand([2.24, 2.26],inf),
\sustain, Prout({ loop{~susRout.next.yield}}),
\pan, Prout({loop{~paneoRout.next.yield}}),
\filepos, Prout({loop{(~lazee.next * 1024).yield}}),
\dur, 0.05
);

```

```

Pdef(\cloudText2).play;
Pdef(\cloudText2).fadeTime = 10;
Pdef(\cloudText2).isPlaying;
Pdef(\cloudText2, ~t1bind2);

```

```

Pdef(\cloudText2).stop;

```

```

(
Pdef(\glissCloudProbandoConv,
Pbind(
\instrument, \glissonCloud2,
\out, 0,
// \buf, Pdefn(\buffer1, ~b2),
\buf, ~b2,
// \pan, Pdefn(\pan2, Prout({loop{~paneoRout.next.yield}})),
\pan, Prout({loop{~paneoRout.next.yield}}),
\level, Prand((0.01,0.02..0.3),inf),
\amp, 0.1,
// \freq2, Pdef(\freq2Conv,Prout({loop{~freqRout.next.yield}})),
\freq2, Prout({loop{~freqRout.next.yield}}),
\freq1, Prand([Pkey(\freq2)-0.1,Pkey(\freq2)+0.1],inf).abs,
\sustain, Prout({ loop{~susRout.next.yield}}),
\dur, Pkey(\sustain)/Prout({loop{~durDivRout.next.yield}}),
\filepos, ~lazee.next * 1024
));

```

```
Pdef(\glissCloudProbandoConv).play  
)
```

```
Pdef(\glissCloudProbandoConv, ~pautoBind).play
```

```
Pdef(\glissCloudProbandoConv, ~pautoBind)
```

```
Pdef(\glissCloudProbandoConv).fadeTime = 10;
```

```
Pdef(\glissCloudProbandoConv).stop;
```

## CÓDIGO FUENTE AUTOPO(I)ÉTICA<sup>94</sup>

```
// =====
// ==== Autopo(i)ética, Estampida de Acusmínimos; Homenaje a Efraín Huerta ====
// ==== Jorge Eduardo Gómez Elizondo 2014 – ... =====
// =====
(

// Numero de valores que se necesita para completar la escena en turno
// Aquí será necesario escribir una función para escoger según sea la escena
~numeros = [24, 27, 35, 60, 65, 72, 72, 72, 72, 20, 35, 35, 2];
// Espera esta en segundos, duración de cada escena, se pondrá en el wait de la rutina
~espera = [40, 25, 25, 15, 15, 15, 15, 15, 20, 15, 15, 15];

// Numero de repeticiones para que pueda seguir a siguiente
// estado el Pdict, (duración de cada escena)

// Duraciones de valores
~duraciones = [

~l = Prand((1,1.125..3.0), ~numeros[0]).asStream,
~lb = Prand((0.5,0.625..2.5), ~numeros[1]).asStream,
~lc = Prand((0.25,0.375..2.0), ~numeros[2]).asStream,
~ld = Pexprand(0.125, 1.5, ~numeros[3]).asStream.round(0.125),
~le = Pexprand(0.125, 0.75, ~numeros[4]).asStream.round(0.125),
~lf = Pexprand(0.125, 0.5, ~numeros[5]).asStream.round(0.125),
~lg = Prand([0.125,0.25], ~numeros[6]).asStream,
~lgGrad = Prand([0.125,0.25], ~numeros[6]).asStream,
~lgMes = Prand([0.125,0.25], ~numeros[6]).asStream,
~lhGrad = Prand([0.125,0.25], ~numeros[6]).asStream,
~lh = Prand([0.125,0.25], ~numeros[7]).asStream,

// disuelve a textura puntillista
```

<sup>94</sup> Aquí solamente la parte del violín, para ver más ejemplos y poder descargar todo el cuarteto: <https://jorgeeduardojonas.wordpress.com/>



```

~IPuntGrad1 = Pexprand(0.125, 0.75, ~numeros[4]).asStream.round(0.125),
~IPuntMes1 = Pexprand(0.125, 1.0, ~numeros[4]).asStream.round(0.125),
~IPuntGrad2 = Pexprand(0.125, 1.25, ~numeros[4]).asStream.round(0.125),
~IPuntMes1 = Pexprand(0.125, 1.5, ~numeros[4]).asStream.round(0.125),
~IPuntGrad3 = Pexprand(0.125, 1.5, ~numeros[4]).asStream.round(0.125),
~IPuntMes2 = Pexprand(0.125, 1.5, ~numeros[4]).asStream.round(0.125),
~li = Prand([0.125,0.25], ~numeros[8]).asStream,
~liB = Prand([0.125,0.25], ~numeros[8]).asStream,
~li2 = Prand([0.125,0.25], ~numeros[8]).asStream,
~lj = Prand((0.5,0.75..3.0), ~numeros[9]).asStream,
~lk = Prand([0.25,0.5], ~numeros[10]).asStream,
~ll = Prand([0.25,0.5], ~numeros[11]).asStream,
~lm = Pseq([4.0],inf)
];

// Conjunto de Escenas 2
~alturasInicioReducido = (0,0.5..4.0);
~alturasInicioAmplio = (0,0.5..11.5);

~mas = [0.5, 1, 6.5, 8.5, 11.5];
~menos = [1.5,2, 2.5, 7];

~formaNormal = { |x| // los valores dentro del arreglo mas o menos

var conjunto, conjRot, distancia, normal, normalInvertido, compara, cual = [];

conjunto = x.sort.removeDups;
// removeDups por si hay más de una forma normal, aún no soluciona más de un
intervalo más amplio
distancia = conjunto.differentiate.put(0, conjunto[conjunto.size - 1] - (conjunto[0] +
12)).abs.removeDups;
/* if(distancia.indicesOfEqual(distancia.max).size > 1) {

}

{ "es menor";*/
conjRot = conjunto.rotate(distancia.indicesOfEqual(distancia.max).unbubble.neg); //

```

```

guardar esto en variable?
normal = (conjRot - conjRot[0]).mod(12);
normalInvertido = (conjRot.reverse - conjRot.reverse[0]).neg.mod(12).abs;
compara = normal +++ normalInvertido;
cual = compara.select({ |x| compara[x][0] != compara[x][1]}).flat;
if( cual.size == 0) {normal} { if(cual[0] < cual[1]) {normal} {normalInvertido}};

// compara.size.do{ |x| if(compara[x][0] != compara[x][1]) { (cual = cual ++
compara[x]).postln} { normal.postln}};

// if( cual[0] < cual[1]) {normal;} {normalInvertido;};

};
~masNormal = ~formaNormal.value(~mas);
~menosNormal = ~formaNormal.value(~menos);

~fnRed = ~formaNormal.value(~alturasInicioReducido);
~fnAmp = ~formaNormal.value(~alturasInicioAmplio);
~notas = [
~notasMasAlni = [60, 60.5, 61],
~notasA = Prand(~notasMasAlni, ~numeros[0]).asStream, // primeras tres notas inicio
espacio reducido
// Segunda Escena

// ~inicioB = ~fnRed.slide(3).clump(3).choose;
// ~notasRemoveB = Array.fill(3, { |x| ~fnRed.remove(~inicioB[x])});
~notasMasB = [60, 60.5, 61, 63, 63.5, 64],

// ~notasMasB = (~notasMasB ++ (~notasRemoveB) + 60);
// ~notasMasB = (~notasMasB ++ ~notasMasAlni).sort;
// ~notasMasB = [60, 60.5, 61, 63, 63.5, 64];

// se agregan tres notas mas
// una por una
~notasB = Pseq([
Prand(~notasMasB.keep(4), 8),
Prand(~notasMasB.keep(5), 8),

```

```

Prand(~notasMasB, 8)
], ~numeros[1]).asStream,

// ~notasB.next
// Tercera Escena
// ~inicioC = ~fnRed.slide(3).clump(3).choose;
// ~notasRemoveC = Array.fill(3, { |x| ~fnRed.remove(~inicioC[x])});
// ~notasMasC = [];
// ~notasMasC = (~notasMasC ++ (~notasRemoveC + 60));
// ~notasMasC = (~notasMasC ++ ~notasMasB).sort;
~notasMasC = [60, 60.5, 61, 61.5, 62, 62.5, 63, 63.5, 64],
// se agregan tres notas mas
// una por una
~notasC = Pseq([
Prand(~notasMasC.keep(7), 8),
Prand(~notasMasC.keep(8), 8),
Prand(~notasMasC, 8)
], ~numeros[2]).asStream,
// se mantiene en un espacio de cuartos de tono 0,0.5 .. 4 central
~notasD = Prand(~notasMasC, ~numeros[3]).asStream,
~notasE = Prand(~notasMasC, ~numeros[4]).asStream,
~notasF = Prand(~notasMasC, ~numeros[5]).asStream,
~notasMasG = ~notasMasC ++ [64.5, 65, 65.5, 66],
~notasG = Prand( ~notasMasG, ~numeros[6]).asStream, // se expande a mitad de
octava

~octavasGGrad = Pwrand([-8, 0, 8], [0.15, 0.7, 0.15], ~numeros[6] + 1).asStream,
~notasGGrad = Pfunc({ (Prand( ~notasMasG, ~numeros[6] + 1).asStream.next +
~octavasGGrad.next).max(55.5)}).asStream,
~octavasGMes = Pwrand([-10, 0, 12], [0.15, 0.5, 0.35], ~numeros[6] + 1).asStream;
~notasGMes = Pfunc({ (Prand( ~notasMasG, ~numeros[6] + 1).asStream.next +
~octavasGMes.next).max(55.5)}).asStream,

// Disuelve en textura puntillista

~octavasPuntGrad1 = Pwrand([-8, 0, 10], [0.15, 0.7, 0.15], ~numeros[4] + 1).asStream,

```

```

// incrementa probabilidad de que se escoja silencio
~disminuyePG1 = Pseq(Pseries.fromEndpoints(1.0, 0.6, ~numeros[4] –
1).asStream.all).asStream,
~incrementaPG1 = Pseq(Pseries.fromEndpoints(0.0, 0.4, ~numeros[4] –
1).asStream.all).asStream,

~notasPuntGrad1 = Prout({ arg incrementa, disminuye;

(~numeros[4] – 2).do{

disminuye = ~disminuyePG1.next;
incrementa = ~incrementaPG1.next;

Pfunc{(Pwrand([Prand(~notasMasG, 1),\r], [disminuye, incrementa].normalizeSum,
~numeros[4] + 1).asStream.next
+ ~octavasPuntGrad1.next).max(55.5)}.asStream.next.yield;

};
}).asStream,
~octavasPuntGrad2 = Pwrand([-8, 0, 10], [0.15, 0.7, 0.15], ~numeros[4] + 1).asStream,
~disminuyePG2 = Pseq(Pseries.fromEndpoints(0.6, 0.4, ~numeros[4] –
1).asStream.all).asStream,
~incrementaPG2 = Pseq(Pseries.fromEndpoints(0.4, 0.6, ~numeros[4] –
1).asStream.all).asStream,

~notasPuntGrad2 = Prout({ arg incrementa, disminuye;

(~numeros[4] – 2).do{

disminuye = ~disminuyePG2.next;
incrementa = ~incrementaPG2.next;

Pfunc{(Pwrand([Prand(~notasMasG, 1),\r], [disminuye, incrementa].normalizeSum,
~numeros[4] + 1).asStream.next
+ ~octavasPuntGrad2.next).max(55.5)}.asStream.next.yield;

```

```

};
}).asStream,

~octavasPuntMes1 = Pwrand([-8, 0, 10], [0.15, 0.7, 0.15], ~numeros[4] + 1).asStream,
~notasPuntMes1 = Pfunc{{{Pwrand([Prand(~notasMasG, 1), \r], [0.4, 0.6], ~numeros[4] +
1).asStream.next +
~octavasPuntMes1.next).max(55)}}).asStream,
~octavasPuntGrad3 = Pwrand([-8, 0, 10], [0.15, 0.7, 0.15], ~numeros[4] + 1).asStream,
~disminuyePG3 = Pseq(Pseries.fromEndpoints(0.4, 0.15, ~numeros[4] -
1).asStream.all).asStream,
~incrementaPG3 = Pseq(Pseries.fromEndpoints(0.6, 0.85, ~numeros[4] -
1).asStream.all).asStream,

~notasPuntGrad3 = Prout({ arg incrementa, disminuye;

(~numeros[4] - 2).do{

disminuye = ~disminuyePG3.next;
incrementa = ~incrementaPG3.next;

Pfunc{(Pwrand([Prand(~notasMasG, 1),\r], [disminuye, incrementa].normalizeSum,
~numeros[4] + 1).asStream.next
+ ~octavasPuntGrad3.next).max(55.5)}}.asStream.next.yield;

};
}).asStream,

~octavasPuntMes2 = Pwrand([-8, 0, 10], [0.15, 0.7, 0.15], ~numeros[4] + 1).asStream,
~notasPuntMes2 = Pfunc{{{Pwrand([Prand(~notasMasG, 1), \r], [0.15, 0.85],
~numeros[4] + 1).asStream.next +
~octavasPuntMes2.next).max(55)}}).asStream,

~octavasHGrad = Pwrand([-8, 0, 10], [0.15,0.7,0.15], ~numeros[6] + 1).asStream,
~notasHGrad = Pfunc{{{Prand(~notasMasG, ~numeros[6] + 1).asStream.next +
~octavasHGrad.next).max(55.5)}}).asStream,
~octavasH = Pwrand([-5, 0, 5], [0.15,0.7,0.15], ~numeros[6] + 1).asStream,
~notasH = Pfunc{{{Prand(~notasMasG, ~numeros[7] + 1).asStream.next +
~octavasH.next).round(0.5).max(55.5)}}).asStream,

```

```

~octavasI = Pwrand([-7, 0, 7], [0.2,0.55,0.25], ~numeros[8] + 1).asStream,
~notasIArreglo = Pfunc({ Array.fill(12, { ~notasMasG.choose +
~octavasI.next}).asSet.asArray.sort }).asStream,
~notasI = Pseq(~notasIArreglo.next, ~numeros[8] + 1).asStream,

~octavasIB = Pwrand([-7, 0, 12], [0.2,0.5,0.3], ~numeros[8] + 1).asStream,
~notasIArregloB = Pfunc({ Array.fill(12, { ~notasMasG.choose +
~octavasIB.next}).asSet.asArray.sort }).asStream,
~notasIB = Pseq(~notasIArregloB.next, ~numeros[8] + 1).asStream,

~octavasI2 = Pwrand([-7, 0, 15], [0.2,0.4,0.4], ~numeros[8] + 1).asStream,
~notasIArreglo2 = Pfunc({ Array.fill(12, { ~notasMasG.choose +
~octavasI2.next}).asSet.asArray.sort }).asStream,
~notasI2 = Pseq(~notasIArreglo2.next, ~numeros[8] + 1).asStream,

// ~notasI = Pfunc({(Prand(~notasMasG, ~numeros[8] + 1).asStream.next +
~octavasI.next).max(55.5)}).asStream;

~notasJ = Prand([[60, 72], [60.5, 72], [60, 71.5]], ~numeros[9]).asStream,

~notasK = Prand([[60,72],[60.5, 72], [60, 71.5], \r, \r, \r, \r, \r], ~numeros[10]).asStream,

~notasL = Prand([[60,72],[60.5, 72], [60, 71.5], \r, \r, \r, \r, \r, \r],
~numeros[11]).asStream,
~notasM = Pseq([[60, 72], [60,72],[72,84]], inf).asStream

];
// =====
// =====
// =====
// =====

~articulaciones = [

~artA = Pseq([Pseq(["[artharm-sounding][fp][x^ glissandoSempre]", 2),
Pwrand(["[artharm-sounding]", "[artharm-sounding][>]", "[artharm-sounding][x^ ric...]",
"[artharm-sounding][fp]"),[1, 0.5, 0.4, 0.3].normalizeSum, ~numeros[0] - 2)],
~numeros[0]).asStream,

```

```

~matices = Pseq(["[artharm-sounding][<1..]", "[artharm-sounding][.>]", "[artharm-
sounding][.>1]"), 1),
~maticesStac = Pseq(["[artharm-sounding][!][<1..]", "[artharm-sounding][!][.>]", "[artharm-
sounding][!][.>1]"), 1),

~artB = Pwrand(["[artharm-sounding]", "[artharm-sounding][x^ ric...]", "[artharm-
sounding][>]",
"[artharm-sounding][fp]", ~matices ], [1, 0.5, 0.5, 0.5, 0.3].normalizeSum,
~numeros[1]).asStream,
~artC = Pseq([Pseq(["[artharm-sounding][x_ pocoCresc]"), 2),
Pwrand(["[artharm-sounding]", "[artharm-sounding][x^ ric...]", "[artharm-sounding][>]",
"[artharm-sounding][fp]", ~matices, "[artharm-sounding][trem]" ], [0.9, 0.5, 0.5, 0.6, 0.5,
0.5].normalizeSum, ~numeros[2] - 2)], ~numeros[2]).asStream,
~artD = Pseq([Pseq(["[artharm-sounding][x_ sempreCresc]"), 2),
Pwrand(["[sfzp]", "[artharm-sounding]", "[artharm-sounding][x^ ric...]", "[artharm-
sounding][>]",
"[artharm-sounding][sfzp]", ~matices, "[artharm-sounding][trem]", "[artharm-
sounding][sfzp][trem]" ],
[0.2, 0.7, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5].normalizeSum, ~numeros[3] - 2)],
~numeros[3]).asStream,
~artE = Pseq([Pseq(["[artharm-sounding][ff][x^ accel]"), 2),
Pwrand(["[sfz]", "[artharm-sounding]", "[artharm-sounding][x^ ric...]", "[artharm-
sounding][>]",
"[artharm-sounding][sfz]", "[artharm-sounding][trem][sfz]" ], [0.3, 0.4, 0.5, 0.5, 0.5,
0.5].normalizeSum,
~numeros[4] - 2)], ~numeros[4]).asStream,
~artF = Pseq([Pseq(["[artharm-sounding][x^ sulpont]"), 2),
Pwrand(["[sfz]", "[artharm-sounding]", "[artharm-sounding][x^ ric...]", "[artharm-
sounding][>]",
"[artharm-sounding][sfz]", "[artharm-sounding][trem][sfz]", [ 0.5, 0.5, 0.5, 0.5, 0.5,
0.5].normalizeSum, ~numeros[5] - 2)
], ~numeros[5]).asStream,
~artG = Pseq([Pseq(["[x^ accelYRallAPlacer]"), 2),
Pwrand(["[sfz]", " ", "[x^ ric...]", "[artharm-sounding]", "[artharm-sounding][x^ ric...]",
"[artharm-sounding][>]",
"[artharm-sounding][sfz]", "[artharm-sounding][trem][sfz]", ~matices],

```

```

[ 0.6, 0.6,0.6, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5].normalizeSum, ~numeros[6] - 2)
], ~numeros[6]).asStream,

// Para las duraciones generadas fuera de las
// articulaciones derivadas del sistema L
// cuidar que las sumas del valor de partida
// y el valor destino sumen cada una 1.

~creceG = Pseq(Pseries.fromEndpoints(0.0, 1.0, ~numeros[6] -
2).asStream.all).asStream, // staccatissimo
~creceG2 = Pseq(Pseries.fromEndpoints(0.0, 1.0, ~numeros[6] -
2).asStream.all).asStream, // staccatissimo armónico
~creceG3 = Pseq(Pseries.fromEndpoints(0.0, 0.9, ~numeros[6] -
2).asStream.all).asStream, // battuto ricochet
~dececeG = Pseq(Pseries.fromEndpoints(0.6, 0.01, ~numeros[6] -
2).asStream.all).asStream, // ord
~dececeG2 = Pseq(Pseries.fromEndpoints(0.6, 0.01, ~numeros[6] -
2).asStream.all).asStream, // armónico artificial ord
~dececeG3 = Pseq(Pseries.fromEndpoints(0.6, 0.2, ~numeros[6] -
2).asStream.all).asStream, // armónico sfz
~dececeG4 = Pseq(Pseries.fromEndpoints(0.5, 0.1, ~numeros[6] -
2).asStream.all).asStream, // armónico trem sfz

// ~crece.next
~gradienteG = Prout({ arg crece, crece2, crece3, decece, decece2, decece3,
decece4;

(~numeros[6] - 2).do{

crece = ~creceG.next;
crece2 = ~creceG2.next;
crece3 = ~creceG3.next;
decece = ~dececeG.next;
decece2 = ~dececeG2.next;
decece3 = ~dececeG3.next;
decece4 = ~dececeG4.next;

```



```

Pwrand(["[!]", "[artharm-sounding][!]", "[x^ batt...]", "[sfz]", " ", "[x^ ric...]", "[artharm-
sounding]", "[artharm-sounding][x^ ric...]", "[artharm-sounding][sfz]", "[artharm-
sounding][trem][sfz]"),
[ crece, // staccatissimo sube 1
crece2, // staccatissimo con armónico sube 1
crece3, // battuto ricochet se incorpora y sube 0.9
0.6, // sfz se mantiene 0.6
decrece, // sonido ord baja 0.01
0.6, // ricochet se mantiene 0.6
decrece2, // armónico artificial ord baja 0.01
0.5, // armónico artificial ricochet se mantiene
decrece3, // armónico artificial sfz baja 0.2
decrece4 // armónico trem sfz baja 0.1
// 0.5 // matices se mantienen pero con staccatissimo
].normalizeSum, ~numeros[6] - 2).asStream.next.yield;
}

}).asStream,

~artGGrad = Pseq([Pseq(["[x^ accelYRallAPlacer]", 2), ~gradienteG],
~numeros[6]).asStream,

~artGMes = Pwrand(["[!]", "[artharm-sounding][!]", "[x^ batt...]", "[sfz]", " ", "[x^ ric...]",
"[artharm-sounding]", "[artharm-sounding][x^ ric...]", "[artharm-sounding][sfz]", "[artharm-
sounding][trem][sfz]", ~maticesStac],

[ 1, // Staccatissimo ord.
1, // Staccatissimo armónico artificial
0.9, // col legno battuto ricochet
0.4, // sfz ord.
0.01, // sonido ord.
0.6, // ricochet ord.
0.01, // armónico artificial
0.5, // armónico artificial con ricochet
0.2, // armónico artificial con sfz
0.1, // armónico artificial trémolo y sfz

```

0.4 // matices sobre notas con staccatissimo y armónico artificial

].normalizeSum, ~numeros[6]

).asStream,

~artPuntGrad1 = Pwrand(["[!]", "[artharm-sounding][!]", "x^ batt...", "[sfz]", " ", "x^ ric..."],  
 "[artharm-sounding]", "[artharm-sounding][x^ ric...]", "[artharm-sounding][sfz]", "[artharm-  
 sounding][trem][sfz]", ~maticesStac],

[ 1, // Staccatissimo ord.

1, // Staccatissimo armónico artificial

0.9, // col legno battuto ricochet

0.4, // sfz ord.

0.01, // sonido ord.

0.6, // ricochet ord.

0.01, // armónico artificial

0.5, // armónico artificial con ricochet

0.2, // armónico artificial con sfz

0.1, // armónico artificial trémolo y sfz

0.4 // matices sobre notas con staccatissimo y armónico artificial

].normalizeSum, ~numeros[4]).asStream,

~artPuntGrad2 = Pwrand(["[!]", "[artharm-sounding][!]", "x^ batt...", "[sfz]", " ", "x^ ric..."],  
 "[artharm-sounding]", "[artharm-sounding][x^ ric...]", "[artharm-sounding][sfz]", "[artharm-  
 sounding][trem][sfz]", ~maticesStac],

[ 1, // Staccatissimo ord.

1, // Staccatissimo armónico artificial

0.9, // col legno battuto ricochet

0.3, // sfz ord.

0.01, // sonido ord.

0.5, // ricochet ord.

0.01, // armónico artificial

0.5, // armónico artificial con ricochet

0.2, // armónico artificial con sfz

0.1, // armónico artificial trémolo y sfz

0.4 // matices sobre notas con staccatissimo y armónico artificial

```
].normalizeSum, ~numeros[4]).asStream,
~artPuntMes1 = Pwrand(["[!]", "[artharm-sounding][!]", "x^ batt...", "[sfz]", " ", "x^ ric..."],
"[artharm-sounding]", "[artharm-sounding][x^ ric...]", "[artharm-sounding][sfz]", "[artharm-
sounding][trem][sfz]", ~maticesStac],
```

```
[ 1, // Staccatíssimo ord.
1, // Staccatíssimo armónico artificial
0.9, // col legno battuto ricochet
0.3, // sfz ord.
0.01, // sonido ord.
0.5, // ricochet ord.
0.01, // armónico artificial
0.5, // armónico artificial con ricochet
0.2, // armónico artificial con sfz
0.1, // armónico artificial trémolo y sfz
0.4 // matices sobre notas con staccatíssimo y armónico artificial
```

```
].normalizeSum, ~numeros[4]).asStream,
```

```
~artPuntGrad3 = Pseq([ Pseq(["x_ moltoDiminuendo"], 2),
Pwrand(["[!]", "[artharm-sounding][!]", "x^ batt...", "[sfz]", " ", "x^ ric..."], "[artharm-
sounding]", "[artharm-sounding][x^ ric...]", "[artharm-sounding][sfz]", "[artharm-
sounding][trem][sfz]", ~maticesStac],
```

```
[ 1, // Staccatíssimo ord.
1, // Staccatíssimo armónico artificial
0.9, // col legno battuto ricochet
0.3, // sfz ord.
0.01, // sonido ord.
0.5, // ricochet ord.
0.01, // armónico artificial
0.5, // armónico artificial con ricochet
0.2, // armónico artificial con sfz
0.1, // armónico artificial trémolo y sfz
0.4 // matices sobre notas con staccatíssimo y armónico artificial
```

```
].normalizeSum, ~numeros[4] – 2), ~numeros[4]).asStream,
~artPuntMes2 = Pseq([Pseq(["pp"],2),
Pwrand(["[!]", "[artharm-sounding][!]", "x^ batt...", "[sfz]", " ", "x^ ric..."], "[artharm-
sounding]", "[artharm-sounding][x^ ric...]", "[artharm-sounding][sfz]", "[artharm-
sounding][trem][sfz]", ~maticesStac],
```

```
[ 1, // Staccatissimo ord.
1, // Staccatissimo armónico artificial
0.9, // col legno battuto ricochet
0.3, // sfz ord.
0.01, // sonido ord.
0.5, // ricochet ord.
0.01, // armónico artificial
0.5, // armónico artificial con ricochet
0.2, // armónico artificial con sfz
0.1, // armónico artificial trémolo y sfz
0.4 // matices sobre notas con staccatissimo y armónico artificial
```

```
].normalizeSum, ~numeros[4] – 2), ~numeros[4]).asStream,

~decreceH = Pseq(Pseries.fromEndpoints(1.0, 0.2, ~numeros[6] –
2).asStream.all).asStream, // staccatissimo
~decreceH2 = Pseq(Pseries.fromEndpoints(1.0, 0.2, ~numeros[6] –
2).asStream.all).asStream, // staccatissimo armónico
~decreceH3 = Pseq(Pseries.fromEndpoints(0.9, 0.1, ~numeros[6] –
2).asStream.all).asStream, // battuto
~creceH = Pseq(Pseries.fromEndpoints(0.01, 0.6, ~numeros[6] –
2).asStream.all).asStream, // ord
~creceH2 = Pseq(Pseries.fromEndpoints(0.01, 0.5, ~numeros[6] –
2).asStream.all).asStream, // armónico
~creceH3 = Pseq(Pseries.fromEndpoints(0.01, 0.5, ~numeros[6] –
2).asStream.all).asStream, // armónico sfz
~creceH4 = Pseq(Pseries.fromEndpoints(0.01, 0.5, ~numeros[6] –
2).asStream.all).asStream, // armónico trem sfz
```

```
~gradienteH = Prout({ arg crece, crece2, crece3, crece4, decrece, decrece2, decrece3,
decrece4;
```

```
(~numeros[6] - 2).do{
```

```
decrece = ~decreceH.next;
decrece2 = ~decreceH2.next;
decrece3 = ~decreceH3.next;
crece = ~creceH.next;
crece2 = ~creceH2.next;
crece3 = ~creceH3.next;
crece4 = ~creceH4.next;
```

```
Pwrand(["[!]", "[artharm-sounding][!]", "[x^ batt...]", "[sfz]", " ", "[x^ ric...]", "[artharm-
sounding]", "[artharm-sounding][x^ ric...]", "[artharm-sounding][sfz]", "[artharm-
sounding][trem][sfz]"),
```

```
[ decrece, // staccatissimo baja 0.2
decrece2, // staccatissimo con armónico baja 0.2
decrece3, // battuto ricochet baja 0.1
0.6, // sfz se mantiene 0.6
crece, // sonido ord crece 0.6
0.6, // ricochet se mantiene 0.6
crece2, // armónico artificial ord sube 0.5
0.5, // armónico artificial ricochet se mantiene
crece3, // armónico artificial sfz sube 0.5
crece4 // armónico trem sfz sube 0.5
// 0.5 // matices se mantienen pero con staccatissimo
].normalizeSum, ~numeros[6] - 2).asStream.next.yield;
}
```

```
}).asStream,
```

```
~artHGrad = Pseq([Pseq(["[x^ accelYRallAPlacer]"], 2), ~gradienteH],
~numeros[6]).asStream,
~artH = Pseq([Pseq(["[x^ moltoAccel]"], 2),
Pwrand(["[!]", "[artharm-sounding][!]", "[x^ batt...]", "[sfz]", " ", "[x^ ric...]",
```

```

["trem"], "artharm-sounding", "artharm-sounding[x^ ric...]", "artharm-sounding[>]",
"artharm-sounding[sfz]", "artharm-sounding[trem][sfz]", ~matices,
[0.3, 0.3, 0.2, 0.6, 0.7,0.6,0.7, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5].normalizeSum, ~numeros[7] –
2)
], ~numeros[7]).asStream,
~artI = Pseq([Pseq(["x_ moltoDiminuendo"], "x_ moltoDiminuendo", "[x^
pocoRitardando] ", 1),
Pwrand(["[!][sfz]", "[!]", "[fp]", " ", "[x^ ric...]", "[trem]", "artharm-sounding"],
"artharm-sounding[x^ ric...]", "artharm-sounding[>]",
"artharm-sounding][fp]", "artharm-sounding][trem][fp]"),
[ 0.5, 0.5, 0.6, 0.7,0.6,0.7, 0.5, 0.5, 0.5, 0.5, 0.5].normalizeSum, ~numeros[8] – 1)
], ~numeros[8]).asStream,
~artIB = Pseq([Pseq(["pp"], "pp"], 1),
Pwrand(["[!][sfz]", "[!]", "[fp]", " ", "[x^ ric...]", "[trem]", "artharm-sounding"],
"artharm-sounding[x^ ric...]", "artharm-sounding[>]",
"artharm-sounding][fp]", "artharm-sounding][trem][fp]"),
[ 0.4, 0.4, 0.5, 0.7,0.6,0.7, 0.5, 0.5, 0.5, 0.5, 0.5].normalizeSum, ~numeros[8] – 1)
], ~numeros[8]).asStream,
~artI2 = Pseq([Pseq(["x_ moltoCrescendo"], "x_ moltoCrescendo", "[x^ moltoAccel] ", 1),
Pwrand(["[!][sfz]", "[!]", "[sfz]", " ", "[x^ ric...]", "[trem]", "artharm-sounding"],
"artharm-sounding[x^ ric...]", "artharm-sounding[>]",
"artharm-sounding][sfz]", "artharm-sounding][trem][sfz]"),
[ 0.7, 0.7, 0.6, 0.7,0.6,0.7, 0.5, 0.5, 0.5, 0.5, 0.5].normalizeSum, ~numeros[8] – 1)
], ~numeros[8]).asStream,
~artJ = Pseq([Pseq(["x^ overpressure_____
_____ord"], "x^ overpressure_____
_____ord"], "x_ diminuendo" "[x^ rallentando]"), 1),
Pwrand([ "[sfz]", " ", "[x^ ric...]", "[trem]"),
[ 0.1,0.6,0.1,0.1].normalizeSum, ~numeros[9] – 1)
], ~numeros[9]).asStream,
// los demás instrumentos cambian de acorde con cada ataque
// randomizar el instrumento que marca los cambios —
~artK = Pseq([Pseq(["x^ agresivo[!]", "x^ agresivo[!]", "x marcandoCambios"], 1),
Pseq(["[!]", ~numeros[10] – 1)
], ~numeros[10]).asStream,
~artL = Pseq([Pseq(["x^ diminuendoPocoaPoco"], "x^ disminuendoPocoaPoco"], "[x

```

```

marcandoCambios"], 1),
Pseq(["[!]", ~numeros[11] - 1)
], ~numeros[11]).asStream,

```

```

~artM = Pseq(["[x^ morendo][fp]", "[x^ morendo][fp]", "[x
IncorporaAGlissEnsamble][ferm]"], inf).asStream,

```

```

~artFin = Pseq(["x^ ¡FIN!"], inf).asStream

```

```

];

```

```

~eventos = (

```

```

dud: Pbind( \dur, 4,
\midinote, \r
).asStream,

```

```

// z: Pbind(\dur, ~inicio, \midinote, \r, \state, \z),

```

```

a: Pbind(\dur, ~l,
\mark, ~artA,
\midinote, ~notasA,
\state, \a).asStream.collect({ |ev|

```

```

Pbind(
\dur, ev[\dur],
\mark, ev[\mark],
\midinote, if(ev[\mark].contains("[artharm-sounding]"), ev[\midinote] + 24, ev[\midinote]),
\state, ev[\state]).asStream.next()

```

```

}),

```

```

b: Pbind(\dur, ~lb,
\mark, ~artB,
\midinote, ~notasB,
\state, \b).asStream.collect({ |ev|

```

```

Pbind(
\dur, ev[\dur],

```

```

\mark, ev[\mark],
\midinote, if(ev[\mark].contains("[artharm-sounding]"), ev[\midinote] + 24, ev[\midinote]),
\state, ev[\state]).asStream.next()

```

```

}),

```

```

c: Pbind(\dur, ~lc,
\mark, ~artC,
\midinote, ~notasC,
\state, \c).asStream.collect({ |ev|

```

```

Pbind(
\dur, ev[\dur],
\mark, ev[\mark],
\midinote, if(ev[\mark].contains("[artharm-sounding]"), ev[\midinote] + 24, ev[\midinote]),
\state, ev[\state]).asStream.next()

```

```

}),

```

```

d: Pbind(\dur, ~ld,
\mark, ~artD,
\midinote, ~notasD,
\state, \d).asStream.collect({ |ev|

```

```

Pbind(
\dur, ev[\dur],
\mark, ev[\mark],
\midinote, if(ev[\mark].contains("[artharm-sounding]"), ev[\midinote] + 24, ev[\midinote]),
\state, ev[\state]).asStream.next()

```

```

}),

```

```

e: Pbind(\dur, ~le,
\mark, ~artE,
\midinote, ~notasE,
\state, \e).asStream.collect({ |ev|

```



```
Pbind(
  \dur, ev[\dur],
  \mark, ev[\mark],
  \midinote, if(ev[\mark].contains("[artharm-sounding]"), ev[\midinote] + 24, ev[\midinote]),
  \state, ev[\state]).asStream.next()

}),
```

```
f: Pbind(\dur, ~lf,
  \mark, ~artF,
  \midinote, ~notasF,
  \state, \f).asStream.collect({ |ev|
```

```
Pbind(
  \dur, ev[\dur],
  \mark, ev[\mark],
  \midinote, if(ev[\mark].contains("[artharm-sounding]"), ev[\midinote] + 24, ev[\midinote]),
  \state, ev[\state]).asStream.next()

}),
```

```
g: Pbind(\dur, ~lg,
  \mark, ~artG,
  \midinote, ~notasG,
  \state, \g).asStream.collect({ |ev|
```

```
Pbind(
  \dur, ev[\dur],
  \mark, ev[\mark],
  \midinote, if(ev[\mark].contains("[artharm-sounding]"), ev[\midinote] + 24, ev[\midinote]),
  \state, ev[\state]).asStream.next()

}),
```

```
gGrad: Pbind(\dur, ~lgGrad,
  \mark, ~artGGrad,
```

```

\midinote, ~notasGGrad,
\state, \gGrad).asStream.collect({ |ev|

Pbind(
\dur, ev[\dur],
\mark, ev[\mark],
\midinote, if(ev[\mark].contains("[artharm-sounding]"), ev[\midinote] + 24, ev[\midinote]),
\state, ev[\state]).asStream.next(())

}),

gMes: Pbind(\dur, ~lgMes,
\mark, ~artGMes,
\midinote, ~notasGMes,
\state, \gMes).asStream.collect({ |ev|

Pbind(
\dur, ev[\dur],
\mark, ev[\mark],
\midinote, if(ev[\mark].contains("[artharm-sounding]"), ev[\midinote] + 24, ev[\midinote]),
\state, ev[\state]).asStream.next(())

}),

puntGrad1: Pbind(\dur, ~lPuntGrad1,
\mark, ~artPuntGrad1,
\midinote, ~notasPuntGrad1,
\state, \puntGrad1).asStream.collect({ |ev|

Pbind(
\dur, ev[\dur],
/*\mark, if(ev[\dur] >= 0.5,

if(ev[\midinote] == \r, " ",

["x^ batt..."], "x^ ric..."], "[artharm-sounding]", "[artharm-sounding][x^ ric...]",
"[artharm-sounding][trem][sfz]"
.wchoose([0.9, 0.6, 0.01, 0.5, 0.1].normalizeSum)),

```

```

if(ev[\midinote] == \r, " ",
["[!]", "[artharm-sounding][!]", "[sfz]", "[artharm-sounding][sfz]", ~maticesStac, " "]
.wchoose([1.0, 1.0, 0.4, 0.4, 0.01].normalizeSum)),*/

\mark, if(ev[\midinote] == \r, " ", ev[\mark]),
\midinote, if(ev[\mark].contains("[artharm-sounding]"), ev[\midinote] + 24, ev[\midinote]),
\state, ev[\state]).asStream.next(())

}),
puntGrad2: Pbind(\dur, ~IPuntGrad2,
\mark, ~artPuntGrad2,
\midinote, ~notasPuntGrad2,
\state, \puntGrad2).asStream.collect({ |ev|

Pbind(
\dur, ev[\dur],
\mark, if(ev[\midinote] == \r, " ", ev[\mark]),
\midinote, if(ev[\mark].contains("[artharm-sounding]"), ev[\midinote] + 24, ev[\midinote]),
\state, ev[\state]).asStream.next(())

}),

puntMes1: Pbind(\dur, ~IPuntMes1,
\mark, ~artPuntMes1,
\midinote, ~notasPuntMes1,
\state, \puntMes1).asStream.collect({ |ev|

Pbind(
\dur, ev[\dur],
\mark, if(ev[\midinote] == \r, " ", ev[\mark]),
\midinote, if(ev[\mark].contains("[artharm-sounding]"), ev[\midinote] + 24, ev[\midinote]),
\state, ev[\state]).asStream.next(())

}),

puntGrad3: Pbind(\dur, ~IPuntGrad3,
\mark, ~artPuntGrad3,

```

```

\midinote, ~notasPuntGrad3,
\state, \puntGrad3).asStream.collect({ |ev|

Pbind(
\dur, ev[\dur],
// \mark, if(ev[\midinote] == \r, " ", ev[\mark]),
\mark, if(ev[\midinote] == \r, if((ev[\mark] == "[x_ moltoDiminuendo]").or(ev[\mark] ==
"[pp]"), ev[\mark], " "), ev[\mark]),
\midinote, if(ev[\mark].contains("[artharm-sounding]"), ev[\midinote] + 24, ev[\midinote]),
\state, ev[\state]).asStream.next(())

}),

puntMes2: Pbind(\dur, ~IPuntMes2,
\mark, ~artPuntMes2,
\midinote, ~notasPuntMes2,
\state, \puntMes2).asStream.collect({ |ev|

Pbind(
\dur, ev[\dur],
// \mark, if(ev[\midinote] == \r, " ", ev[\mark]),
\mark, if(ev[\midinote] == \r, if((ev[\mark] == "[x_ moltoDiminuendo]").or(ev[\mark] ==
"[pp]"), ev[\mark], " "), ev[\mark]),
\midinote, if(ev[\mark].contains("[artharm-sounding]"), ev[\midinote] + 24, ev[\midinote]),
\state, ev[\state]).asStream.next(())

}),

hGrad: Pbind(\dur, ~lhGrad,
\mark, ~artHGrad,
\midinote, ~notasHGrad,
\state, \hGrad).asStream.collect({ |ev|

Pbind(
\dur, ev[\dur],
\mark, ev[\mark],
\midinote, if(ev[\mark].contains("[artharm-sounding]"), ev[\midinote] + 24, ev[\midinote]),
\state, ev[\state]).asStream.next(())

```

```
}),
```

```
h: Pbind(\dur, ~lh,
\mark, ~artH,
\midinote, ~notasH,
\state, \h).asStream.collect({ |ev|
```

```
Pbind(
\dur, ev[\dur],
\mark, ev[\mark],
\midinote, if(ev[\mark].contains("[artharm-sounding]"), ev[\midinote] + 24, ev[\midinote]),
\state, ev[\state]).asStream.next(())
```

```
}),
```

```
i: Pbind(\dur, ~li,
\mark, ~artI,
\midinote, ~notasI.max(55.5),
\state, \i).asStream.collect({ |ev|
```

```
Pbind(
\dur, ev[\dur],
\mark, ev[\mark],
\midinote, if(ev[\mark].contains("[artharm-sounding]"), ev[\midinote] + 24, ev[\midinote]),
\state, ev[\state]).asStream.next(())
```

```
}),
```

```
iB: Pbind(\dur, ~liB,
\mark, ~artIB,
\midinote, ~notasIB.max(55.5),
\state, \iB).asStream.collect({ |ev|
```

```
Pbind(
\dur, ev[\dur],
\mark, ev[\mark],
```

```
\midinote, if(ev[\mark].contains("[artharm-sounding]"), ev[\midinote] + 24, ev[\midinote]),
\state, ev[\state]).asStream.next(()
```

```
}),
```

```
i2: Pbind(\dur, ~li2,
\mark, ~artI2,
\midinote, ~notasI2.max(55.5),
\state, \i2).asStream.collect({ |ev|
```

```
Pbind(
\dur, ev[\dur],
\mark, ev[\mark],
\midinote, if(ev[\mark].contains("[artharm-sounding]"), ev[\midinote] + 24, ev[\midinote]),
\state, ev[\state]).asStream.next(()
```

```
}),
```

```
j: Pbind(\dur, ~lj,
\mark, ~artJ,
\midinote, ~notasJ,
\state, \j).asStream.collect({ |ev|
```

```
Pbind(
\dur, if(ev[\mark] != " ", 0.5, ev[\dur]),
\mark, ev[\mark],
\midinote, ev[\midinote],
\state, ev[\state]).asStream.next(()
```

```
}),
```

```
k: Pbind(\dur, ~lk,
\mark, ~artK,
\midinote, ~notasK,
\state, \k).asStream.collect({ |ev|
```

```

Pbind(
  \dur, ev[\dur],
  \mark, if(ev[\midinote] == \r, if((ev[\mark] == "[x^ agresivo][!]").or(ev[\mark] == "[x
  marcandoCambios]"), ev[\mark], " "), ev[\mark]),
  \midinote, ev[\midinote],
  \state, ev[\state]).asStream.next()

}),

```

```

l: Pbind(\dur, ~ll,
  \mark, ~artL,
  \midinote, ~notasL,
  \state, \l).asStream.collect({ |ev|

```

```

Pbind(
  \dur, ev[\dur],
  \mark, if(ev[\midinote] == \r, if((ev[\mark] == "[x^ disminuendoPocooPocoo]").or(ev[\mark]
  == "[x marcandoCambios]"), ev[\mark], " "), ev[\mark]),
  \midinote, ev[\midinote],
  \state, ev[\state]).asStream.next()

}),

```

```

m: Pbind(\dur, ~lm,
  \mark, ~artM,
  \midinote, ~notasM,
  \state, \m).asStream.collect({ |ev|

```

```

Pbind(
  \dur, ev[\dur],
  \mark, ev[\mark],
  \midinote, ev[\midinote],
  \state, ev[\state]).asStream.next()

}),

```

```

fin: Pbind(\dur, 4,
  \mark, "[x^ FIN]",

```

```

\midinote, \r,
\state, \fin).asStream;
);

//
// // ~posibilidades = [
// // Pseq([\a, \b, \c, \d, \e],inf),
// // Pseq([\b, \a, \d, \c, \e],inf),
// // ].choose;
//
// ~vln = Pdict(~eventos, Pseq([\a, \a2, \a3, \a4, \b, \c, \d, \e],inf));
// ~vln2 = Pdict(~eventosVln2, Pseq([\a, \a2, \a3, \a4, \b, \c, \d, \e],inf)).asStream;

// ~vln.next()
// ~vln2.next()

// Original, Cervantino
// ~vln = Pdict(~eventos, Pseq([\a, \b, \c, \d, \e, \f, \g, \gGrad, \gMes, \hGrad, \h, \i, \iB,
\i2, \j, \k, \l, \m], inf)).asStream;
// de gMes salta a overpressure de j versión más corta
// ~vln = Pdict(~eventos, Pseq([\a, \b, \c, \d, \e, \f, \g, \gGrad, \gMes, \j, \k, \l, \m],
inf)).asStream;

// Se disuelve en textura puntillista con dinámicas suaves predominantes, acentuaciones
esporádicas

// ~vln = Pdict(~eventos, Pseq([\a, \b, \c, \d, \e, \f, \g, \gGrad, \gMes], inf)).asStream;

~vln = Pdict(~eventos, Pseq([\a, \b, \c, \d, \e, \f, \g, \gGrad, \gMes, \puntGrad1,
\puntGrad2, \puntMes1, \puntGrad3, \puntMes2, \fin], inf)).asStream;
~rutina = Routine{
loop{

var vln;

```



```

"osascript -e 'tell application \"Skim\" to activate\"'.unixCmd;
~vln.next().collect({ |estado|

```

```

case

```

```

{ estado == \dud } {~num = 1; ~wait = 4;}
{ estado == \a } {~num = ~numeros[0] - 1; ~wait = ~espera[0];}
{ estado == \b } {~num = ~numeros[1] - 1; ~wait = ~espera[1];}
{ estado == \c } {~num = ~numeros[2] - 1; ~wait = ~espera[2];}
{ estado == \d } {~num = ~numeros[3] - 1; ~wait = ~espera[3];}
{ estado == \e } {~num = ~numeros[4] - 1; ~wait = ~espera[4];}
{ estado == \f } {~num = ~numeros[5] - 1; ~wait = ~espera[5];}
{ estado == \g } {~num = ~numeros[6] - 1; ~wait = ~espera[6];}
{ estado == \gGrad } {~num = ~numeros[6] - 1; ~wait = ~espera[6];}
{ estado == \gMes } {~num = ~numeros[6] - 1; ~wait = ~espera[6];}
{ estado == \puntGrad1 } {~num = ~numeros[4] - 3; ~wait = ~espera[6];}
{ estado == \puntGrad2 } {~num = ~numeros[4] - 3; ~wait = ~espera[6];}
{ estado == \puntMes1 } {~num = ~numeros[4] - 3; ~wait = ~espera[6];}
{ estado == \puntGrad3 } {~num = ~numeros[4] - 3; ~wait = ~espera[6];}
{ estado == \puntMes2 } {~num = ~numeros[4] - 3; ~wait = ~espera[6];}
{ estado == \hGrad } {~num = ~numeros[6] - 1; ~wait = ~espera[6];}
{ estado == \h } {~num = ~numeros[7] - 1; ~wait = ~espera[7];}
{ estado == \i } {~num = ~numeros[8] - 1; ~wait = ~espera[8];}
{ estado == \iB } {~num = ~numeros[8] - 1; ~wait = ~espera[8];}
{ estado == \i2 } {~num = ~numeros[8] - 1; ~wait = ~espera[8];}
{ estado == \j } {~num = ~numeros[9] - 1; ~wait = ~espera[9];}
{ estado == \k } {~num = ~numeros[10] - 1; ~wait = ~espera[10];}
{ estado == \l } {~num = ~numeros[11] - 1; ~wait = ~espera[11];}
{ estado == \m } {~num = 5}
{ estado == \fin } {~num = 1; ~wait = 10};
estado;
});

```

```

vln = Fomus2(~vln, ~num).instrumentName_(“violin”).instrument_(“violin”);
vln.fileName_(“~/Desktop/elViolincillo”);
vln.octsUp_(0);

```

```

vln.ly;
~wait.wait;

}
};
w = Window.new("Autopo(i)ética Estampida de Acusminimos", Rect(0, 0, 240, 80));
~boton = Button(w, Rect(10, 10, 200, 50));

~boton.states_(
["Iniciar", Color.black, Color.cyan],
["Corriendo", Color.black, Color.red],
])
.action_({ arg bot;

case
{ bot.value == 0 } {
fork{

1.do{
~rutina.stop;
~rutina.reset;
~vln.reset;
~duraciones.do(_reset);
~notas.do(_reset);
~articulaciones.do(_reset);
"finito! ***** ".postln;

~path = "/Users/JETGE/Documents/Archivos_SC
/3.6/Maestria/autoTiempoREal/Autopo(i)éticaEstampidaDeAcusmínimos/autopo(i)etica.rtf";
~file = File(~path, "r");
~text = ~file.readAllString;
~file.close;
~text.interpret;
CmdPeriod.run; // por si acaso
0.yield;

```

```
};  
};  
}  
{ bot.value == 1 } {  
  
fork{  
1.do{  
"Espera... ***** ".postln;  
5.0.wait;  
~rutina.play;  
}  
}  
  
};  
  
});  
  
w.alwaysOnTop_(true);  
w.front;  
  
);
```

# APÉNDICE III

## DATOS ANÁLISIS POR FASES

Fase Estacionaria E, compás 1-15<sup>95</sup>

Compás 1  $\Delta 1$

B	$f_0$	$g_0$	$d_1$	
Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )	7 <sub>5</sub>	1	<i>ff</i>	IV
5 (11 <sub>4</sub> - 0 <sub>4</sub> )	4 <sub>4</sub> 2 <sub>4</sub>	2	<i>pp</i>	III
4 (0 <sub>4</sub> - 1 <sub>3</sub> )				
3 (1 <sub>3</sub> - 2 <sub>2</sub> )				
2 (2 <sub>2</sub> - 4 <sub>1</sub> )	2 <sub>2</sub>	1	<i>pp</i>	II
1 (4 <sub>1</sub> - 4 <sub>0</sub> )	3 <sub>1</sub> 1 <sub>1</sub> 10 <sub>0</sub> 0 <sub>1</sub> 5 <sub>0</sub> 8 <sub>0</sub> 6 <sub>0</sub> 9 <sub>0</sub>	8	<i>f</i>	I

Compás 1  $\Delta 2$

G	$f_0$	$g_1$	$d_0$	
Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )				
5 (11 <sub>4</sub> - 0 <sub>4</sub> )				
4 (0 <sub>4</sub> - 1 <sub>3</sub> )	11 <sub>3</sub> 8 <sub>3</sub> 1 <sub>3</sub> 6 <sub>3</sub> 8 <sub>3</sub> 2 <sub>3</sub> 4 <sub>3</sub>	7	<i>pp</i>	IV
3 (1 <sub>3</sub> - 2 <sub>2</sub> )	10 <sub>2</sub> 6 <sub>2</sub> 3 <sub>2</sub>	3	<i>pp</i>	I + II + III
2 (2 <sub>2</sub> - 4 <sub>1</sub> )				
1 (4 <sub>1</sub> - 4 <sub>0</sub> )				

<sup>95</sup> Aquí se incluye solamente la fase estacionaria E, para descargar las tablas de la partituras completa, que pueden ser utilizadas para un posterior análisis y para consulta: <https://jorgeeduardojonas.wordpress.com/>

## Compás 2 Δ3

A  $f_0$   $g_0$   $d_0$ 

Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )	4 <sub>5</sub> 0 <sub>5</sub> 5 <sub>5</sub> (11 <sub>4</sub> ) 9 <sub>5</sub> 8 <sub>5</sub> 6 <sub>5</sub> 2 <sub>5</sub>	8	<i>ff</i> ( <i>pp</i> )	IV
5 (11 <sub>4</sub> - 0 <sub>4</sub> )				
4 (0 <sub>4</sub> - 1 <sub>3</sub> )	7 <sub>3</sub>	1	<i>pp</i>	(III)
3 (1 <sub>3</sub> - 2 <sub>2</sub> )				
2 (2 <sub>2</sub> - 4 <sub>1</sub> )				
1 (4 <sub>1</sub> - 4 <sub>0</sub> )	4 <sub>1</sub> 7 <sub>0</sub> 10 <sub>0</sub>	3	<i>f</i>	

\*11<sub>4</sub> puede ser considerado como dentro de la región 5

## Compás 2 Δ4

B  $f_0$   $g_0$   $d_1$ 

Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )	7 <sub>5</sub>	1	<i>ff</i>	IV
5 (11 <sub>4</sub> - 0 <sub>4</sub> )	7 <sub>4</sub> 3 <sub>4</sub>	2	<i>pp</i>	III
4 (0 <sub>4</sub> - 1 <sub>3</sub> )				
3 (1 <sub>3</sub> - 2 <sub>2</sub> )				
2 (2 <sub>2</sub> - 4 <sub>1</sub> )	9 <sub>1</sub>	1	<i>pp</i>	II
1 (4 <sub>1</sub> - 4 <sub>0</sub> )	2 <sub>1</sub> 2 <sub>1</sub> 1 <sub>1</sub> 9 <sub>0</sub> 8 <sub>0</sub> 2 <sub>1</sub> 5 <sub>0</sub> 6 <sub>0</sub>	8	<i>f</i>	I

## Compás 3 Δ5

E  $f_1$   $g_0$   $d_0$ 

Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )				
5 (11 <sub>4</sub> - 0 <sub>4</sub> )				
4 (0 <sub>4</sub> - 1 <sub>3</sub> )	8 <sub>3</sub> 6 <sub>3</sub> 6 <sub>3</sub> 0 <sub>3</sub> 9 <sub>3</sub> 11 <sub>3</sub> 2 <sub>3</sub>	7	<i>ff</i>	IV
3 (1 <sub>3</sub> - 2 <sub>2</sub> )	8 <sub>3</sub> 2 <sub>2</sub>	2	<i>f</i>	I + II + III
2 (2 <sub>2</sub> - 4 <sub>1</sub> )				
1 (4 <sub>1</sub> - 4 <sub>0</sub> )				

## Compás 3 Δ6

C	$f_0$	$g_1$	$d_0$	
Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )	4 <sub>5</sub> 6 <sub>5</sub> 8 <sub>5</sub> 2 <sub>5</sub> 3 <sub>5</sub> 11 <sub>4</sub> 1 <sub>5</sub> 5 <sub>5</sub> 7 <sub>5</sub>	9	<i>pp</i>	IV
5 (11 <sub>4</sub> - 0 <sub>4</sub> )	(0 <sub>4</sub> ) 5 <sub>4</sub>	2	( <i>pp</i> ) <i>ff</i>	III
4 (0 <sub>4</sub> - 1 <sub>3</sub> )				
3 (1 <sub>3</sub> - 2 <sub>2</sub> )				
2 (2 <sub>2</sub> - 4 <sub>1</sub> )				
1 (4 <sub>1</sub> - 4 <sub>0</sub> )	0 <sub>1</sub> 5 <sub>0</sub> 4 <sub>0</sub>	3	<i>pp</i>	I

Compás 4 Δ7

B	$f_0$	$g_0$	$d_1$	
Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )	9 <sub>5</sub>	1	<i>ff</i>	IV
5 (11 <sub>4</sub> - 0 <sub>4</sub> )	2 <sub>4</sub> 0 <sub>4</sub>	2	<i>pp</i>	III
4 (0 <sub>4</sub> - 1 <sub>3</sub> )				
3 (1 <sub>3</sub> - 2 <sub>2</sub> )				
2 (2 <sub>2</sub> - 4 <sub>1</sub> )	11 <sub>1</sub>	1	<i>pp</i>	II
1 (4 <sub>1</sub> - 4 <sub>0</sub> )	2 <sub>1</sub> 0 <sub>1</sub> 4 <sub>1</sub> 9 <sub>0</sub> 6 <sub>0</sub> 7 <sub>0</sub>	6	<i>f</i>	I

Compás 4 Δ8

H?	$f_1$	$g_1$	$d_1$	
Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )				
5 (11 <sub>4</sub> - 0 <sub>4</sub> )				
4 (0 <sub>4</sub> - 1 <sub>3</sub> )	8 <sub>3</sub> 4 <sub>3</sub> 7 <sub>3</sub> 1 <sub>3</sub>	4	<i>pp</i>	IV
3 (1 <sub>3</sub> - 2 <sub>2</sub> )	7 <sub>2</sub> 9 <sub>2</sub> 11 <sub>2</sub> 10 <sub>2</sub> 5 <sub>2</sub> 6 <sub>2</sub> 1 <sub>2</sub> 8 <sub>2</sub> 1 <sub>3</sub>	9	<i>pp</i>	I (II + III)
2 (2 <sub>2</sub> - 4 <sub>1</sub> )				
1 (4 <sub>1</sub> - 4 <sub>0</sub> )				

Compás 5 Δ9

A  $f_0$   $g_0$   $d_0$

Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )	6 <sub>5</sub> 2 <sub>5</sub> 5 <sub>5</sub> 8 <sub>5</sub> 9 <sub>5</sub> 4 <sub>5</sub> 0 <sub>5</sub>	7	<i>ff</i>	IV
5 (11 <sub>4</sub> - 0 <sub>4</sub> )				III
4 (0 <sub>4</sub> - 1 <sub>3</sub> )				
3 (1 <sub>3</sub> - 2 <sub>2</sub> )				
2 (2 <sub>2</sub> - 4 <sub>1</sub> )	4 <sub>1</sub>	1	<i>pp</i>	II
1 (4 <sub>1</sub> - 4 <sub>0</sub> )	8 <sub>0</sub> 6 <sub>0</sub> 7 <sub>0</sub> 8 <sub>0</sub>	4	<i>f</i>	I

Compás 5  $\Delta 10$

D  $f_0$   $g_1$   $d_1$

Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )	(7 <sub>5</sub> )	1	( <i>ff</i> )	IV
5 (11 <sub>4</sub> - 0 <sub>4</sub> )	8 <sub>4</sub> 4 <sub>4</sub>	2	<i>f</i>	III
4 (0 <sub>4</sub> - 1 <sub>3</sub> )				
3 (1 <sub>3</sub> - 2 <sub>2</sub> )				
2 (2 <sub>2</sub> - 4 <sub>1</sub> )	0 <sub>2</sub>	1	<i>f</i>	II
1 (4 <sub>1</sub> - 4 <sub>0</sub> )	3 <sub>1</sub> 1 <sub>1</sub> 2 <sub>1</sub> 0 <sub>1</sub> 10 <sub>0</sub> 9 <sub>0</sub> 8 <sub>0</sub> 5 <sub>0</sub> 11 <sub>0</sub> 7 <sub>0</sub>	10	<i>pp</i>	I

Compás 6  $\Delta 11$

G  $f_1$   $g_1$   $d_0$

Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )				
5 (11 <sub>4</sub> - 0 <sub>4</sub> )				
4 (0 <sub>4</sub> - 1 <sub>3</sub> )	6 <sub>3</sub> 9 <sub>3</sub> 3 <sub>3</sub> 4 <sub>3</sub> 0 <sub>4</sub> 8 <sub>3</sub> 11 <sub>3</sub> 1 <sub>3</sub> 5 <sub>3</sub> 2 <sub>3</sub>	10	<i>pp</i>	IV
3 (1 <sub>3</sub> - 2 <sub>2</sub> )	4 <sub>2</sub> 5 <sub>2</sub> 2 <sub>2</sub>	3	<i>pp</i>	I + (II + III)
2 (2 <sub>2</sub> - 4 <sub>1</sub> )				
1 (4 <sub>1</sub> - 4 <sub>0</sub> )				

Compás 6  $\Delta 12$ 

C	$f_0$	$g_1$	$d_0$	
Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )	6 <sub>5</sub> 9 <sub>5</sub> 6 <sub>5</sub> 0 <sub>5</sub> 7 <sub>5</sub> 11 <sub>4</sub> 2 <sub>5</sub> 4 <sub>5</sub> 1 <sub>5</sub>	9	<i>pp</i>	IV
5 (11 <sub>4</sub> - 0 <sub>4</sub> )	10 <sub>4</sub>	1	( <i>pp</i> )	III
4 (0 <sub>4</sub> - 1 <sub>3</sub> )				
3 (1 <sub>3</sub> - 2 <sub>2</sub> )				
2 (2 <sub>2</sub> - 4 <sub>1</sub> )	10 <sub>1</sub>	1	<i>f</i>	II
1 (4 <sub>1</sub> - 4 <sub>0</sub> )	8 <sub>0</sub> 5 <sub>0</sub> 11 <sub>0</sub>	3	<i>pp</i>	I

Compás 7  $\Delta 13$ 

E	$f_1$	$g_0$	$d_0$	
Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )				
5 (11 <sub>4</sub> - 0 <sub>4</sub> )				
4 (0 <sub>4</sub> - 1 <sub>3</sub> )	7 <sub>3</sub> 11 <sub>3</sub> 5 <sub>3</sub> 10 <sub>3</sub> 0 <sub>3</sub> 9 <sub>3</sub> 2 <sub>3</sub> 6 <sub>3</sub>	8	<i>ff</i>	IV
3 (1 <sub>3</sub> - 2 <sub>2</sub> )	1 <sub>2</sub> 2 <sub>2</sub> 7 <sub>2</sub>	3	<i>f</i>	I (II + III)
2 (2 <sub>2</sub> - 4 <sub>1</sub> )				
1 (4 <sub>1</sub> - 4 <sub>0</sub> )				

Compás 7  $\Delta 14$ 

G	$f_1$	$g_1$	$d_0$	
Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )				
5 (11 <sub>4</sub> - 0 <sub>4</sub> )				
4 (0 <sub>4</sub> - 1 <sub>3</sub> )	(1 <sub>4</sub> ) 4 <sub>3</sub> 1 <sub>3</sub> 3 <sub>3</sub> 0 <sub>4</sub> 11 <sub>3</sub> 7 <sub>3</sub> 8 <sub>3</sub> 10 <sub>3</sub>	9	<i>pp</i>	IV
3 (1 <sub>3</sub> - 2 <sub>2</sub> )	8 <sub>2</sub> 0 <sub>3</sub>	2	<i>pp</i>	I (II + III)
2 (2 <sub>2</sub> - 4 <sub>1</sub> )				
1 (4 <sub>1</sub> - 4 <sub>0</sub> )				

Compás 8  $\Delta 15$



D	$f_0$	$g_1$	$d_1$	
Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )	8 <sub>5</sub>	1	<i>pp</i>	IV
5 (11 <sub>4</sub> - 0 <sub>4</sub> )				III
4 (0 <sub>4</sub> - 1 <sub>3</sub> )				
3 (1 <sub>3</sub> - 2 <sub>2</sub> )	3 <sub>2</sub>	1	<i>f</i>	
2 (2 <sub>2</sub> - 4 <sub>1</sub> )	(8 <sub>1</sub> ) 9 <sub>1</sub>	2	( <i>pp</i> ) <i>f</i>	II
1 (4 <sub>1</sub> - 4 <sub>0</sub> )	2 <sub>1</sub> 0 <sub>1</sub> 3 <sub>1</sub> 6 <sub>0</sub> 5 <sub>0</sub> 10 <sub>0</sub> 9 <sub>0</sub> 6 <sub>0</sub> 7 <sub>0</sub>	9	<i>pp</i>	I

## Compás 8 Δ16

C	$f_0$	$g_1$	$d_0$	
Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )	4 <sub>5</sub> 7 <sub>5</sub> 1 <sub>5</sub> 11 <sub>4</sub> 3 <sub>5</sub> 5 <sub>5</sub> 9 <sub>5</sub> 6 <sub>5</sub> 4 <sub>5</sub>	9	<i>pp</i>	IV
5 (11 <sub>4</sub> - 0 <sub>4</sub> )	4 <sub>4</sub>	1	<i>f</i>	III
4 (0 <sub>4</sub> - 1 <sub>3</sub> )				
3 (1 <sub>3</sub> - 2 <sub>2</sub> )				
2 (2 <sub>2</sub> - 4 <sub>1</sub> )				II
1 (4 <sub>1</sub> - 4 <sub>0</sub> )	10 <sub>0</sub> 11 <sub>0</sub> 1 <sub>1</sub>	3	<i>pp</i>	I

## Compás 9 Δ17

H?	$f_1$	$g_1$	$d_1$	
Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )				
5 (11 <sub>4</sub> - 0 <sub>4</sub> )				
4 (0 <sub>4</sub> - 1 <sub>3</sub> )	(1 <sub>4</sub> ) 0 <sub>4</sub>	2	<i>pp</i>	IV
3 (1 <sub>3</sub> - 2 <sub>2</sub> )	11 <sub>2</sub> 10 <sub>2</sub> 0 <sub>3</sub> 8 <sub>2</sub> 1 <sub>3</sub> 5 <sub>2</sub> 6 <sub>2</sub> 3 <sub>2</sub> 9 <sub>2</sub> (1 <sub>2</sub> )	10	<i>pp</i>	I (II + III)
2 (2 <sub>2</sub> - 4 <sub>1</sub> )				
1 (4 <sub>1</sub> - 4 <sub>0</sub> )				

## Compás 9 Δ18

F?	$f_1$	$g_0$	$d_1$	
Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )				
5 (11 <sub>4</sub> - 0 <sub>4</sub> )				
4 (0 <sub>4</sub> - 1 <sub>3</sub> )	(11 <sub>3</sub> ) 2 <sub>3</sub> [0 <sub>3</sub> ]	3	<i>(ff) f</i>	IV
3 (1 <sub>3</sub> - 2 <sub>2</sub> )	11 <sub>2</sub> 4 <sub>2</sub> 6 <sub>2</sub> 2 <sub>2</sub> 3 <sub>2</sub> 7 <sub>2</sub> 10 <sub>2</sub> 2 <sub>2</sub>	8	<i>f</i>	I (II + III)
2 (2 <sub>2</sub> - 4 <sub>1</sub> )				
1 (4 <sub>1</sub> - 4 <sub>0</sub> )				

Compás 10 Δ19

E	$f_1$	$g_0$	$d_0$	
Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )				
5 (11 <sub>4</sub> - 0 <sub>4</sub> )				
4 (0 <sub>4</sub> - 1 <sub>3</sub> )	0 <sub>4</sub> 4 <sub>3</sub> 5 <sub>3</sub> 7 <sub>3</sub> 10 <sub>3</sub> 9 <sub>3</sub> 11 <sub>3</sub> 8 <sub>3</sub>	8	<i>ff</i>	IV
3 (1 <sub>3</sub> - 2 <sub>2</sub> )	4 <sub>2</sub> 5 <sub>2</sub> 6 <sub>2</sub>	3	<i>f</i>	I (II + III)
2 (2 <sub>2</sub> - 4 <sub>1</sub> )				
1 (4 <sub>1</sub> - 4 <sub>0</sub> )				

Compás 10 Δ20

A	$f_0$	$g_0$	$d_0$	
Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )	4 <sub>5</sub> 6 <sub>5</sub> 0 <sub>5</sub> 11 <sub>4</sub> 1 <sub>5</sub> 8 <sub>5</sub> 9 <sub>5</sub> 2 <sub>5</sub> 7 <sub>5</sub> 5 <sub>5</sub>	10	<i>ff</i>	IV
5 (11 <sub>4</sub> - 0 <sub>4</sub> )				III
4 (0 <sub>4</sub> - 1 <sub>3</sub> )				
3 (1 <sub>3</sub> - 2 <sub>2</sub> )				
2 (2 <sub>2</sub> - 4 <sub>1</sub> )	0 <sub>2</sub>	1	<i>pp</i>	II
1 (4 <sub>1</sub> - 4 <sub>0</sub> )	8 <sub>0</sub> 6 <sub>0</sub> 11 <sub>0</sub> 4 <sub>1</sub>	4	<i>f</i>	I

Compás 11 Δ21

D	$f_0$	$g_1$	$d_1$	
Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )	8 <sub>5</sub>	1	<i>pp</i>	IV
5 (11 <sub>4</sub> - 0 <sub>4</sub> )	6 <sub>4</sub> 0 <sub>4</sub>	2	<i>f</i>	III
4 (0 <sub>4</sub> - 1 <sub>3</sub> )				
3 (1 <sub>3</sub> - 2 <sub>2</sub> )				
2 (2 <sub>2</sub> - 4 <sub>1</sub> )	1 <sub>2</sub> (5 <sub>1</sub> )	2	<i>f (pp)</i>	II
1 (4 <sub>1</sub> - 4 <sub>0</sub> )	0 <sub>1</sub> 2 <sub>1</sub> 4 <sub>1</sub> 8 <sub>0</sub> 5 <sub>0</sub> 7 <sub>0</sub> 8 <sub>0</sub> 10 <sub>0</sub>	8	<i>pp</i>	I

## Compás 11 Δ22

A	$f_0$	$g_0$	$d_0$	
Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )	6 <sub>5</sub> 7 <sub>5</sub> 2 <sub>5</sub> 1 <sub>5</sub> 5 <sub>5</sub> 4 <sub>5</sub> 3 <sub>5</sub> 9 <sub>5</sub> (10 <sub>4</sub> 9 <sub>4</sub> )	10	<i>ff</i>	IV
5 (11 <sub>4</sub> - 0 <sub>4</sub> )				III
4 (0 <sub>4</sub> - 1 <sub>3</sub> )				
3 (1 <sub>3</sub> - 2 <sub>2</sub> )				
2 (2 <sub>2</sub> - 4 <sub>1</sub> )	1 <sub>2</sub>	1	<i>pp</i>	II
1 (4 <sub>1</sub> - 4 <sub>0</sub> )	1 <sub>1</sub> 4 <sub>1</sub> 11 <sub>0</sub> 1 <sub>1</sub>	4	<i>f</i>	I

## Compás 12 Δ23

D	$f_0$	$g_1$	$d_1$	
Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )	6 <sub>5</sub>	1	<i>pp</i>	IV
5 (11 <sub>4</sub> - 0 <sub>4</sub> )	6 <sub>4</sub>	1	<i>f</i>	III
4 (0 <sub>4</sub> - 1 <sub>3</sub> )				
3 (1 <sub>3</sub> - 2 <sub>2</sub> )				
2 (2 <sub>2</sub> - 4 <sub>1</sub> )	1 <sub>2</sub> 1 <sub>2</sub>	2	<i>f</i>	II
1 (4 <sub>1</sub> - 4 <sub>0</sub> )	0 <sub>1</sub> 2 <sub>1</sub> 4 <sub>1</sub> 8 <sub>0</sub> 5 <sub>0</sub> 8 <sub>0</sub> (10 <sub>0</sub> )	7	<i>pp (f)</i>	I

## Compás 12 Δ24

C	$f_0$	$g_1$	$d_0$	
Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )	7 <sub>5</sub> 2 <sub>5</sub> 11 <sub>4</sub> 5 <sub>5</sub> 1 <sub>5</sub> 9 <sub>5</sub> 8 <sub>5</sub> 3 <sub>5</sub> 0 <sub>5</sub> 4 <sub>5</sub>	10	<i>pp</i>	IV
5 (11 <sub>4</sub> - 0 <sub>4</sub> )	1 <sub>4</sub>	1	<i>f</i>	III
4 (0 <sub>4</sub> - 1 <sub>3</sub> )				
3 (1 <sub>3</sub> - 2 <sub>2</sub> )				
2 (2 <sub>2</sub> - 4 <sub>1</sub> )				II
1 (4 <sub>1</sub> - 4 <sub>0</sub> )	1 <sub>1</sub> 7 <sub>0</sub> 4 <sub>0</sub>	3	<i>pp</i>	I

Compás 13 Δ25

F	$f_1$	$g_0$	$d_1$	
Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )				
5 (11 <sub>4</sub> - 0 <sub>4</sub> )				
4 (0 <sub>4</sub> - 1 <sub>3</sub> )	11 <sub>3</sub>	1	<i>ff</i>	IV
3 (1 <sub>3</sub> - 2 <sub>2</sub> )	2 <sub>2</sub> 9 <sub>2</sub> 7 <sub>2</sub> 1 <sub>2</sub> 8 <sub>2</sub> 11 <sub>2</sub> 4 <sub>2</sub> 5 <sub>2</sub> 3 <sub>2</sub>	9	<i>f</i>	I (II + III)
2 (2 <sub>2</sub> - 4 <sub>1</sub> )				
1 (4 <sub>1</sub> - 4 <sub>0</sub> )				

Compás 13 Δ26

F	$f_1$	$g_0$	$d_1$	
Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )				
5 (11 <sub>4</sub> - 0 <sub>4</sub> )	2 <sub>4</sub>	1	<i>ff</i>	
4 (0 <sub>4</sub> - 1 <sub>3</sub> )	1 <sub>3</sub> 2 <sub>3</sub> 3 <sub>3</sub>	3	<i>f</i>	IV
3 (1 <sub>3</sub> - 2 <sub>2</sub> )	0 <sub>3</sub> 9 <sub>2</sub> 8 <sub>2</sub> 6 <sub>2</sub> 10 <sub>2</sub> 11 <sub>2</sub> 3 <sub>2</sub>	7	<i>f</i>	I (II + III)
2 (2 <sub>2</sub> - 4 <sub>1</sub> )				
1 (4 <sub>1</sub> - 4 <sub>0</sub> )				

Compás 14 Δ27

E  $f_1$   $g_0$   $d_0$

Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )				
5 (11 <sub>4</sub> - 0 <sub>4</sub> )				
4 (0 <sub>4</sub> - 1 <sub>3</sub> )	8 <sub>3</sub> 1 <sub>4</sub> 1 <sub>3</sub> 2 <sub>3</sub> 0 <sub>4</sub> 10 <sub>3</sub> 5 <sub>3</sub>	7	<i>ff</i>	IV
3 (1 <sub>3</sub> - 2 <sub>2</sub> )	5 <sub>2</sub> 7 <sub>2</sub> 5 <sub>2</sub>	3	<i>f</i>	I (II + III)
2 (2 <sub>2</sub> - 4 <sub>1</sub> )				
1 (4 <sub>1</sub> - 4 <sub>0</sub> )				

Compás 14 Δ28

D  $f_0$   $g_1$   $d_1$

Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )	5 <sub>5</sub>	1	<i>ff</i>	IV
5 (11 <sub>4</sub> - 0 <sub>4</sub> )	9 <sub>4</sub> 0 <sub>5</sub>	2	<i>pp</i>	III
4 (0 <sub>4</sub> - 1 <sub>3</sub> )				
3 (1 <sub>3</sub> - 2 <sub>2</sub> )				
2 (2 <sub>2</sub> - 4 <sub>1</sub> )	10 <sub>1</sub>	1	<i>pp</i>	II
1 (4 <sub>1</sub> - 4 <sub>0</sub> )	9 <sub>0</sub> 11 <sub>0</sub> 4 <sub>1</sub> 5 <sub>0</sub> 7 <sub>0</sub> 0 <sub>9</sub> 8 <sub>0</sub> 2 <sub>1</sub>	8	<i>f</i>	I

Compás 15 Δ29

A  $f_0$   $g_0$   $d_0$

Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )	3 <sub>5</sub> 4 <sub>5</sub> (10 <sub>4</sub> ) 5 <sub>5</sub> 6 <sub>5</sub> 2 <sub>5</sub> 0 <sub>5</sub> 1 <sub>5</sub> 11 <sub>4</sub> 7 <sub>5</sub>	10	<i>ff</i>	IV
5 (11 <sub>4</sub> - 0 <sub>4</sub> )				III
4 (0 <sub>4</sub> - 1 <sub>3</sub> )				
3 (1 <sub>3</sub> - 2 <sub>2</sub> )	9 <sub>2</sub> 3 <sub>2</sub>	2	<i>pp</i>	
2 (2 <sub>2</sub> - 4 <sub>1</sub> )				II
1 (4 <sub>1</sub> - 4 <sub>0</sub> )	1 <sub>1</sub> 6 <sub>0</sub> 6 <sub>0</sub>	3	<i>f</i>	I

Compás 15 Δ30

H?  $f_0$   $g_0$   $d_0$

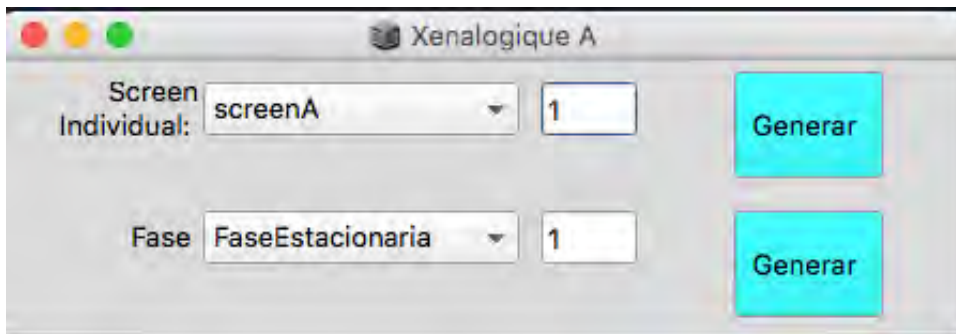
Region Frec.	Frecuencias	Densidad	Dinámica	
6 (9 <sub>5</sub> - 11 <sub>4</sub> )				
5 (11 <sub>4</sub> - 0 <sub>4</sub> )	1 <sub>4</sub>	1	<i>pp</i>	
4 (0 <sub>4</sub> - 1 <sub>3</sub> )	0 <sub>4</sub>	1	<i>pp</i>	IV
3 (1 <sub>3</sub> - 2 <sub>2</sub> )	10 <sub>2</sub> 1 <sub>3</sub> 5 <sub>2</sub> 6 <sub>2</sub> 8 <sub>2</sub> 11 <sub>2</sub> 0 <sub>3</sub> 7 <sub>2</sub> 4 <sub>2</sub>	9	<i>pp</i>	I (II + III)
2 (2 <sub>2</sub> - 4 <sub>1</sub> )				
1 (4 <sub>1</sub> - 4 <sub>0</sub> )				

A	B	C	D	E	F	G	H
5	3	4	5	4	3	3	3
16.66%	10%	13.33%	16.66%	13.33%	10%	10%	10%

# APÉNDICE IV

## CÓDIGO FUENTE XENALOGIQUE A

En la parte superior de la interfaz gráfica, se puede seleccionar el screen y el número de estos que se desee generar, y en la parte inferior se puede generar la fase estacionaria o alguna perturbación con el número de screens indicada. Se crean dos archivos, uno .xml que se puede abrir en Sibelius o Finale, y mediante Lilypond un PDF.



```
// =====
// ===== Iannis Xenakis =====
// Analogique A (1958)=====
//=====
// ===== Implementación: Jorge E. Gómez Elizondo =====
// =====

// (Xenakis, 1992: 82) MTP
/*

Notación:

| X Y
— |—————
X | 0.85 0.4
Y | 0.15 0.6
Representación de matriz en SC: [[ 0.85, 0.15], [ 0.4, 0.6 ]]
Columnas en cada arreglo
*/

// =====
// MTPF (of frecuencies) =====
// =====

(
~alpha = [[0.2, 0.8],[0.8, 0.2]]; // frecuencias
~gamma = [[0.2, 0.8],[0.8, 0.2]]; // intensidades
~lambda = [[0.2, 0.8],[0.8, 0.2]]; // densidades

~mtpAlpha = MarkovSet([
[f0, [f0, \f1], [0.2, 0.8]], // f0 siga siendo f0 20% de probabilidad, f0 -> f1 80% ...
[f1, [f0, \f1], [0.8, 0.2]],
]);
// 10.do { ~mtpAlpha.next(\f0).postln; };
~mtpAlpha.makeSeeds;
~mtpAlpha = ~mtpAlpha.asStream;
// 100.do{ ~mtpAlpha.next.postln; };

```



```

~beta = [[0.85, 0.15],[ 0.4, 0.6]]; // frecuencias
~epsilon = [[0.85, 0.15], [0.4, 0.6]]; // intensidades
~mu = [[0.85, 0.15],[0.4, 0.6]]; // densidades
~mtpBeta = MarkovSet([
[f0, [f0, f1], [0.85, 0.15]],
[f1, [f0, f1], [0.4, 0.6]],

]);
~mtpBeta.makeSeeds;
~mtpBeta = ~mtpBeta.asStream;

// =====
// MTPG (of intensities) =====
// =====

~mtpGamma = MarkovSet([
[g0, [g0, g1],[0.2, 0.8]],
[g1, [g0, g1],[0.8, 0.2]],
]);

~mtpGamma.makeSeeds;
~mtpGamma = ~mtpGamma.asStream;

~mtpEpsilon = MarkovSet([
[g0, [g0, g1],[0.85, 0.15]],
[g1, [g0, g1],[0.4, 0.6]],
]);

~mtpEpsilon.makeSeeds;
~mtpEpsilon = ~mtpEpsilon.asStream;

// =====
// MTPD (of densities) =====
// =====

~mtpLambda = MarkovSet([
[d0, [d0, d1],[0.2, 0.8]],
[d1, [d0, d1],[0.8, 0.2]],

```

```

]);
~mtpLambda.makeSeeds;
~mtpLambda = ~mtpLambda.asStream;

~mtpMu = MarkovSet([
[d0, [d0, \d1],[0.85, 0.15]],
[d1, [d0, \d1],[0.4, 0.6]],
]);

~mtpMu.makeSeeds;
~mtpMu = ~mtpMu.asStream;

// The transformations of the variables are indeterminate at the interior of each (MTP)
// (diagram processes),
// but on the other hand their (MTP) will be connected by means of a determined
// coupling of parameters
// (Xenakis, 1992: 83)

// f0 -> Lambda (d)
// f1 -> Mu (d)
// d0 -> Alpha (f)
// d1 -> Beta (f)
// g0 -> Lambda (d)
// g1 -> Mu (d)
// g0 -> Beta (f)
// g1 -> Alpha (f)
// f0 -> Gamma (g)
// f1 -> Epsilon (g)
// d0 -> Gamma (g)
// d1 -> Epsilon (g)

// Función para hacer el 'coupling' de las matrices
=====
// Agrupación de mtps dentro de un environment

~mtps = (

```

```
mtpf: (
alpha: ~mtpAlpha,
beta: ~mtpBeta,
),
```

```
mtpg: (
gamma: ~mtpGamma,
epsilon: ~mtpEpsilon,
),
```

```
mtpd: (
lambda: ~mtpLambda,
mu: ~mtpMu
)
```

```
);
```

```
/*
```

```
// como acceder, distintas notaciones
```

```
~mtps[mtpf][\alpha].next;
```

```
~mtps.mtpf.alpha.next;
```

```
~mtps.at(mtpf).at(\alpha).next;
```

```
// Se dejará una variable para escoger a partir de que colección de MTP se comenzará
```

```
// En la inicial se escoge aleatoriamente 1/2 de probabilidad para cada MTP
```

```
// Ya después dependerá de los 'couplings' que especificados en (Xenakis, 1992: 83)
```

```
// 50% de probabilidad para cada una de las MTPS
```

```
[~mtps.mtpf.choose.next, ~mtps.mtpg.choose.next, ~mtps.mtpd.choose.next];
```

```
// más adelante se hace el mapeo o coupling
```

```
*/
```

```
// Las 8 screens expuestas en (Xenakis, 1992: 101)
```

```
~screens = Dictionary[
```

```
\screenA -> [f0, \g0, \d0],
```

```
\screenB -> [f0, \g0, \d1],
```

```

\screenC -> [\f0, \g1, \d0],
\screenD -> [\f0, \g1, \d1],
\screenE -> [\f1, \g0, \d0],
\screenF -> [\f1, \g0, \d1],
\screenG -> [\f1, \g1, \d0],
\screenH -> [\f1, \g1, \d1]
];

// ~screens[\screenA]

// Diccionario con asociaciones de los couplings, pero que no actúa sobre ellos

~coupling = Dictionary[

\f0 -> [[ ~lambda, ~gamma ], [\d, \g] ], // (d) (g)
\f1 -> [[ ~mu, ~epsilon ], [\d, \g] ], // (d) (g)
\g0 -> [[ ~lambda, ~beta ], [\d, \f] ], // (d) (f)
\g1 -> [[ ~mu, ~alpha ], [\d, \f] ], // (d) (f)
\d0 -> [[ ~alpha, ~gamma ], [\f, \g]], // (f) (g)
\d1 -> [[ ~epsilon, ~beta ], [\g, \f]] // (g) (f)

];

// versión alternativa tiene una relación más directa entre variable
// afectada y su matriz correspondiente
~coupling2 = Dictionary[

\f0 -> ( d: ~lambda, g: ~gamma),
\f1 -> ( d: ~mu, g: ~epsilon),
\g0 -> ( d: ~lambda, f: ~beta),
\g1 -> ( d: ~mu, f: ~alpha),
\d0 -> ( f: ~alpha, g: ~gamma),
\d1 -> ( g: ~epsilon, f: ~beta)

];

// Determina la MTP con la que se escogerá el siguiente Screen
// basado en una entrada que especifica el Screen actual
~couplingAct = Dictionary[

```

// Si se escoge determinado Screen entonces el siguiente Screen  
// se escogerá de acuerdo a los mapeos de MTP establecidos

// Screen A f0g0d0

// f0 -> Lambda (d) Gamma (g)

// g0 -> Lambda (d) Beta (f)

// d0 -> Alpha (f) Gamma (g)

[f0, \g0, \d0] -> { [ ~mtps.mtpf.choose.next, ~mtps.mtpg.gamma.next,  
,~mtps.mtpd.lambda.next ] },

// Screen B f0g0d1

// f0 -> Lambda (d) Gamma (g)

// g0 -> Lambda (d) Beta (f)

// d1 -> Beta (f) Epsilon (g)

[f0, \g0, \d1] -> { [ ~mtps.mtpf.choose.next, ~mtps.mtpg.choose.next,  
~mtps.mtpd.lambda.next ] },

// Screen C f0g1d0

// f0 -> Lambda (d) Gamma (g)

// g1 -> Mu (d) Alpha (f)

// d0 -> Alpha (f) Gamma (g)

[f0, \g1, \d0] -> { [ ~mtps.mtpf.alpha.next, ~mtps.mtpg.gamma.next,  
~mtps.mtpd.choose.next ] },

// Screen D f0g1d1

// f0 -> Lambda (d) Gamma (g)

// g1 -> Mu (d) Alpha (f)

// d1 -> Beta (f) Epsilon (g)

[f0, \g1, \d1] -> { [ ~mtps.mtpf.choose.next, ~mtps.mtpg.choose.next,  
~mtps.mtpd.choose.next ] },

// Screen E f1g0d0

// f1 -> Mu (d) Epsilon (g)

```

// g0 -> Lambda (d) Beta (f)
// d0 -> Alpha (f) Gamma (g)

[f1, \g0, \d0] -> { [ ~mtps.mtpf.choose.next, ~mtps.mtpg.choose.next,
~mtps.mtpd.choose.next ] },

// Screen F f1g0d1
// f1 -> Mu (d) Epsilon (g)
// g0 -> Lambda (d) Beta (f)
// d1 -> Beta (f) Epsilon (g)

[f1, \g0, \d1] -> { [ ~mtps.mtpf.beta.next, ~mtps.mtpg.epsilon.next,
~mtps.mtpd.choose.next ] },

// Screen G f1g1d0
// f1 -> Mu (d) Epsilon (g)
// g1 -> Mu (d) Alpha (f)
// d0 -> Alpha (f) Gamma (g)

[f1, \g1, \d0] -> { [ ~mtps.mtpf.alpha.next, ~mtps.mtpg.choose.next, ~mtps.mtpd.mu.next
] },

// Screen H f1g1d1
// f1 -> Mu (d) Epsilon (g)
// g1 -> Mu (d) Alpha (f)
// d1 -> Beta (f) Epsilon (g)

[f1, \g1, \d1] -> { [~mtps.mtpf.choose.next, ~mtps.mtpg.epsilon.next,
~mtps.mtpd.mu.next] }
];

~rutinaEstacionaria = Routine({

var valorInicial, valorActual;

~screenInicial = ~screens[
~scrArr =

```

```

[screenA,\screenB,\screenC,\screenD,\screenE,\screenF,\screenG,\screenH].wchoose(
[17.11, 11.069, 13.792, 12.942, 14.558, 12.11, 8.238, 10.716].normalizeSum));
~screenInicial.yield; // primero dar el valor inicial
valorInicial = ~couplingAct[~screenInicial].value;
valorActual = valorInicial;
~observaScreen = valorActual;

inf.do{
valorActual.yield;
valorActual = ~couplingAct[valorActual].value;

};
});
// Pesos de probabilidades del tipo P1, fueron calculadas más abajo con función
// para llegar a fase estacionaria

~p1A = [ 2.1, 8.4, 8.4, 33.6, 1.9, 7.6, 7.6, 30.4 ];

~p1B = [ 35.7, 8.9, 32.3, 8.1, 6.3, 1.6, 5.7, 1.4 ];

~p1C = [ 8.4, 7.6, 2.1, 1.9, 33.6, 30.4, 8.4, 7.6 ];

~p1D = [ 18.9, 12.6, 12.6, 8.4, 17.1, 11.4, 11.4, 7.6 ];

~p1E = [ 16.5, 15, 15, 13.5, 11, 10, 10, 9 ];

~p1F = [ 20.4, 13.6, 3.6, 2.4, 30.6, 20.4, 5.4, 3.6 ];

~p1G = [ 40.8, 7.2, 27.2, 4.8, 10.2, 1.8, 6.8, 1.2 ];

~p1H = [ 9.6, 14.4, 14.4, 21.6, 6.4, 9.6, 9.6, 14.4 ];

~rutinaPerturbada = Routine({ arg perturbacion;

var valorInicial, valorActual;

valorInicial = ~screens[
~scrArr =

```

```

[screenA,\screenB,\screenC,\screenD,\screenE,\screenF,\screenG,\screenH].wchoose(
perturbacion.normalizeSum)];
valorInicial.yield;
valorActual = valorInicial;
~observaScreen = valorActual;

inf.do{
valorActual.yield;
valorActual = ~screens[
~scrArr =
[screenA,\screenB,\screenC,\screenD,\screenE,\screenF,\screenG,\screenH].wchoose(
perturbacion.normalizeSum)];

};
});

// (Xenakis, 1992: 84) "Where does the transformation summarized by an (MTP) go?
What is its destiny?"

// Let us consider the (MTP)

// | X Y
// —|—————
// X | 0.2 0.8
// Y | 0.8 0.2
// and suppose one hundred mechanisms identified by the law of this single (MTP). We
shall allow them all to set out from
// X and evolve freely. The preceding question then becomes, "Is there a general
tendency for the states of the hundred
// mechanisms, and if so, what is it?"

// (Xenakis, 1992: 85)

// If this is to be applied to the 100 mechanisms X as above, we shall have:
// Stage Mechanisms X Mechanisms Y
// —————
// 0 100 0

```



```
// 1 20 80
// 2 68 32
// 3 39 61
// 4 57 43
// 5 46 54
// 6 52 48
// 7 49 51
// 8 50 50
// 9 50 50
```

```
// (Xenakis, 1992: 86)
```

```
/*
```

```
=====
```

```
Multiplicación de Matrices 101wanowan =====
```

```
=====
```

Notación normal

```
A = [ 1, 0 ]
    [ 2, 1 ]
```

```
B = [ 5, 4 ]
    [-5, 1 ]
```

Sintaxis de arreglos en SC

```
a = [[1, 0], [2, 1]];
b = [[5, 4], [-5, 1]];
```

```
b * a = [[13, 4], [-3, 1]]
```

```
[ 5, 4 ] [ 1, 0 ] [ 13, 4 ]
[-5, 1 ] [ 2, 1 ] [-3, 1 ]
(5 * 1) + (4 * 2) = 13
(-5 * 1) + (1 * 2) = -3
```

```

(5 * 0) + (4 * 1) = 4
(-5 * 0) + (1 * 1) = 1
*/
// Dentro de una función ...

~equilibrium = { |stagesNum, a, b, c, d|

var mtp, veces, nuevo;

mtp = [[a, b], [c, d]];
veces = [100, 0];

stagesNum.do{

if( veces.rank > 1,
{
veces = veces[0] * mtp;
veces = (veces[0] + veces[1]).round(1);
veces.postln;
},

{
veces = veces * mtp;
veces = (veces[0] + veces[1]).round(1);
veces.postln;

}
);
};
};

// Los argumentos son:
// (numero de etapas, n11, n12, n21, n22) rows y columns respectivamente como
subíndice
// ~equilibrium.value(10, 0.2, 0.8, 0.8, 0.2)
// ~equilibrium.value(20, 0.2, 0.8, 0.8, 0.2)
// Si se corre 100 veces se obtiene el stationary stage

```

```

// MTP representada como una máquina de estados
/*(
a = Pfsm([
#[0,1],
0, [0!2, 1!8].flat,
1, [0!8, 1!2].flat,
],inf).asStream;
)

c = a.nextN(100); // Corre 100 veces
c.takeThese({|x| (x == 0)}); // se queda solamente con los resultados que fueron "X"
c.size; // ¿Cuántos fueron?
(100 - c.size).abs; // ¿Cuántos fueron "Y"?
// Para el MTP sigma  $\sigma$ 

(
b = Pfsm([
#[0],
0, Pwrand([0,1],[0.85,0.15],inf).asStream.nextN(100),
1, Pwrand([0,1],[0.4,0.6],inf).asStream.nextN(100),
],inf).asStream;
)

c = b.nextN(100)
c.takeThese({|x| (x == 0)});
c.size
(100 - c.size).abs;
// Se evaluó 9 veces y se saca la media aritmética
(25+17+16+28+24+32+31+48+33)/9 // 1 ==> 28.22222
(75+83+84+72+76+68+69+52+67)/9 // 0 ==> 71.7777777777778
~equilibrium.value(10, 0.85, 0.15, 0.4, 0.6)
*/

// =====
// Entropy =====

```

```

// =====
// (Xenakis, 1992: 86)

// Now that we know how to calculate the stationary probabilities of a Markov chain
// we can easily calculate its mean entropy.

// Este es el bueno, faltaba hacer la conversión a bits que no viene explícito
// en (Xenakis, 1992: 86)
/*a = (log(0.85) * -0.85) + (log(0.15) * -0.15);
a = (a * 1.442695040889).round(1e-3) // 1/log(2) ¿para la conversión a bits?

b = (log(0.4) * -0.4) + (log(0.6) * -0.6);
b = (b * 1.442695040889).round(1e-3);

// 73 y 27 del equilibrium stage
((a * 0.73) + (b * 0.27)).round(1e-3); // resultado 0.707 bits
*/

// Para la primera MTP (Xenakis, 1992: 82)

~meanEntropy = { |a, b, c, d, estacionario1, estacionario2|

var uno, dos;

uno = ((log(a) * a) + (log(b) * b)) * -1;
dos = ((log(c) * c) + (log(d) * d)) * -1;

uno = (uno * (1/log(2))).round(1e-3);
dos = (dos * (1/log(2))).round(1e-3);

((uno * estacionario1) + (dos * estacionario2)).round(1e-3);

};
// ~meanEntropy.value(0.85, 0.15, 0.4, 0.6, 0.73, 0.27); // 0.707 bits
// ~meanEntropy.value(0.8, 0.2, 0.2, 0.8, 0.5, 0.5); // 0.722 bits
/*
// MTPZ (Xenakis, 1992: 89)

```

// dA

$$0.021 + 0.357 + 0.084 + 0.189 + 0.165 + 0.204 + 0.408 + 0.096 // = 1.524$$

// dB

$$0.084 + 0.089 + 0.076 + 0.126 + 0.150 + 0.136 + 0.072 + 0.144 // = 0.877$$

// dC

$$0.084 + 0.323 + 0.021 + 0.126 + 0.150 + 0.036 + 0.272 + 0.144 // = 1.156$$

// dD

$$0.336 + 0.081 + 0.019 + 0.084 + 0.135 + 0.024 + 0.048 + 0.216 // = 0.943$$

// dE

$$0.019 + 0.063 + 0.336 + 0.171 + 0.110 + 0.306 + 0.102 + 0.064 // = 1.171$$

// dF

$$0.076 + 0.016 + 0.304 + 0.114 + 0.100 + 0.204 + 0.018 + 0.096 // = 0.928$$

// dG

$$0.076 + 0.057 + 0.084 + 0.114 + 0.100 + 0.054 + 0.068 + 0.096 // = 0.649$$

// dH

$$0.304 + 0.014 + 0.076 + 0.076 + 0.090 + 0.036 + 0.012 + 0.144 // = 0.752$$

$$[0.021, 0.084, 0.084, 0.336, 0.019, 0.076, 0.076, 0.304].sum // = 1$$

$$[0.357, 0.089, 0.323, 0.081, 0.063, 0.016, 0.057, 0.014].sum // = 1$$

$$[1.524, 0.877, 1.156, 0.943, 1.171, 0.928, 0.649, 0.752].sum/8 // = 1$$

\*/

// =====

// Calcular Matriz de Probabilidades automáticamente =====

// =====

// =====

// con coupling2 =====

// ¿solución alternativa?

```

~mtpz = { arg screen1, screen2;

// var f, g, d;

// meterlo dentro de una rutina para que se borren con
// cada iteración y puedan ser rellenados

~f = [];
~g = [];
~d = [];
case
// cuatro posibilidades de transformación para f según sean las screens
// f0 -> f0
{(~screens[screen1][0] == \f0).and(~screens[screen2][0] == \f0)}
{
~f = ~f.add(~coupling2[~screens[screen1][1]][\f][0][0]); // g calcula para f ...
~f = ~f.add(~coupling2[~screens[screen1][2]][\f][0][0]); // d calcula para f ...
}

// f0 -> f1
{(~screens[screen1][0] == \f0).and(~screens[screen2][0] == \f1)}
{
~f = ~f.add(~coupling2[~screens[screen1][1]][\f][0][1]);
~f = ~f.add(~coupling2[~screens[screen1][2]][\f][0][1]);
}

// f1 -> f1
{(~screens[screen1][0] == \f1).and(~screens[screen2][0] == \f1)}
{
~f = ~f.add(~coupling2[~screens[screen1][1]][\f][1][1]);
~f = ~f.add(~coupling2[~screens[screen1][2]][\f][1][1]);
}

// f1 -> f0
{(~screens[screen1][0] == \f1).and(~screens[screen2][0] == \f0)}
{
~f = ~f.add(~coupling2[~screens[screen1][1]][\f][1][0]);
}

```

```

~f = ~f.add(~coupling2[~screens[screen1][2]][\f][1][0]);
};

case
// cuatro posibilidades de transformación para g según sea el caso
// g0 -> g0
{(~screens[screen1][1] == \g0).and(~screens[screen2][1] == \g0)}
{
~g = ~g.add(~coupling2[~screens[screen1][0]][\g][0][0]); // f calcula para g ...
~g = ~g.add(~coupling2[~screens[screen1][2]][\g][0][0]); // d calcula para g ...
}

// g0 -> g1
{(~screens[screen1][1] == \g0).and(~screens[screen2][1] == \g1)}
{
~g = ~g.add(~coupling2[~screens[screen1][0]][\g][0][1]);
~g = ~g.add(~coupling2[~screens[screen1][2]][\g][0][1]);
}

// g1 -> g1
{(~screens[screen1][1] == \g1).and(~screens[screen2][1] == \g1)}
{
~g = ~g.add(~coupling2[~screens[screen1][0]][\g][1][1]);
~g = ~g.add(~coupling2[~screens[screen1][2]][\g][1][1]);
}

// g1 -> g0
{(~screens[screen1][1] == \g1).and(~screens[screen2][1] == \g0)}
{
~g = ~g.add(~coupling2[~screens[screen1][0]][\g][1][0]);
~g = ~g.add(~coupling2[~screens[screen1][2]][\g][1][0]);
};

case
// cuatro posibilidades de transformación para d según sea el caso
// d0 -> d0

```

```

{(~screens[screen1][2] == \d0).and(~screens[screen2][2] == \d0)}
{
~d = ~d.add(~coupling2[~screens[screen1][0]][\d][0][0]); // f calcula para d ...
~d = ~d.add(~coupling2[~screens[screen1][1]][\d][0][0]); // g calcula para d ...
}

// d0 -> d1
{(~screens[screen1][2] == \d0).and(~screens[screen2][2] == \d1)}
{
~d = ~d.add(~coupling2[~screens[screen1][0]][\d][0][1]);
~d = ~d.add(~coupling2[~screens[screen1][1]][\d][0][1]);
}

// d1 -> d1
{(~screens[screen1][2] == \d1).and(~screens[screen2][2] == \d1)}
{
~d = ~d.add(~coupling2[~screens[screen1][0]][\d][1][1]);
~d = ~d.add(~coupling2[~screens[screen1][1]][\d][1][1]);
}

// d1 -> d0
{(~screens[screen1][2] == \d1).and(~screens[screen2][2] == \d0)}
{
~d = ~d.add(~coupling2[~screens[screen1][0]][\d][1][0]);
~d = ~d.add(~coupling2[~screens[screen1][1]][\d][1][0]);
};

// compound probabilities
~fcomp = (~f[0] + ~f[1])/2;
~gcomp = (~g[0] + ~g[1])/2;
~dcomp = (~d[0] + ~d[1])/2;

// compound probabilities for independent events

~total = ~fcomp * ~gcomp * ~dcomp;
};
// Arreglo de símbolos de keys de screens para iterar

```



```

~scrArr = [\screenA,\screenB,\screenC,\screenD,\screenE,\screenF,\screenG,\screenH];

~screens.size.do{ arg x; ~screens[~scrArr[x]].postln; };
~screens.size.do{ arg x; ~scrArr[x].postln; };
// probabilidad de que una reja (primer argumento) vaya a otra (segundo argumento)
~mtpz.value(\screenA, \screenA);
~mtpz.value(\screenA, \screenB);

// lo mismo pero ya con el arreglo para iterar
~mtpz.value(\screenA, ~scrArr[0]);
~mtpz.value(\screenA, ~scrArr[1]);
// Partiendo de ScreenA todas las probabilidades
~screens.size.do{ arg x; ~mtpz.value(\screenA, ~scrArr[x]).postln; };

// Estructura de arreglo que va a contener la mtpz completa
Array.fill(8, { arg x; Array.fill(8, { arg y; [x, y]})});

// Listo!
~mtpzYa = Array.fill(8, { arg x; Array.fill(8, { arg y; ~mtpz.value(~scrArr[x],
~scrArr[y]).round(1e-3)}})});

~mtpzYa[0]; // A -> A, B, C ... H
~mtpzYa[1]; // B -> A, B, C ... H
~mtpzYa[2]; // C -> A, B, C ... H
~mtpzYa[3]; // D -> A, B, C ... H
~mtpzYa[4]; // E -> A, B, C ... H
~mtpzYa[5]; // F -> A, B, C ... H
~mtpzYa[6]; // G -> A, B, C ... H
~mtpzYa[7]; // H -> A, B, C ... H

// Todo de una vez
8.do{ arg x; ~mtpzYa[x].postln; };

// Perturbaciones y stationary states de la matriz
// Does the matrix have a region stability? (Xenakis, 1992: 89)
// The dA screens at the moment t will becom d'A screens at the moment t + Δt, and this
number will

```

```

// be equal to the sum of all the screens that will be produced by the remaining
mechanisms, in
// accordance with the corresponding probabilities (Xenakis, 1992: 89)
~mtpzYa[0][0]; // A -> A
~mtpzYa[1][0]; // B -> A
~mtpzYa[2][0]; // C -> A
~dA = Array.fill(~mtpzYa[0].size, { arg x; ~mtpzYa[x][0]});
~dA.sum;
~dB = Array.fill(~mtpzYa[0].size, { arg x; ~mtpzYa[x][1]});
~dB.sum;
~dC = Array.fill(~mtpzYa[0].size, { arg x; ~mtpzYa[x][2]});
~dC.sum;
~dD = Array.fill(~mtpzYa[0].size, { arg x; ~mtpzYa[x][3]});
~dD.sum;
~dE = Array.fill(~mtpzYa[0].size, { arg x; ~mtpzYa[x][4]});
~dE.sum;
~dF = Array.fill(~mtpzYa[0].size, { arg x; ~mtpzYa[x][5]});
~dF.sum;
~dG = Array.fill(~mtpzYa[0].size, { arg x; ~mtpzYa[x][6]});
~dG.sum;
~dH = Array.fill(~mtpzYa[0].size, { arg x; ~mtpzYa[x][7]});
~dH.sum;

/*
// At the stationary state the frequency of the screens A, B, C, ..., H will remain constant
// and the eight preceding equations will become:
// (d'A = dA, d'B = dB, d'C = dc, ... d'H = dH)

*/

// Xenakis utilizó el segundo método:
// This method is very laborious [el primero], for the chance of error is very high ...

// "... consists in making all 100 mechanisms Z set out from a single screen and letting
them
// evolve by themselves. After several more or less long oscillations, the stationary state,

```

```

// if it exists, will be attained and the proportions of the screens will remain invariable”
// (Xenakis, 1992: 90)
// A linear operator, the matrix of transition probabilities Z. Consequently system (e1) can
be
// summarized in a matrix equation:
// (e4)  $V' = ZV$ .

// Since we have broken down continuity into a discontinuous succession of thickness in
time  $\Delta t$ ,
// the equation (e4) will be applied to each stage  $\Delta t$ 
//(Xenakis, 1992: 91)
// Perturbación partiendo de H
~z0H = [0, 0, 0, 0, 0, 0, 0, 100];

// ~mecanismo Z está en archivo mtpZ.scd
// 8.do{ arg x; ~mecanismoZ[x].postln }; // [0.021, 0.357 ... ] A -> A, B -> A, C -> A ...
8.do{ arg x; ~mtpzYa[x].postln }; // [0.021, 0.084 ... ] A -> A, A -> B, A -> C ...

// convertir ~mtpzYa a estructura de ~mecanismoZ, invirtiendo rows y columns con
método flop
8.do{ arg x; (~z0H*~mtpzYa.flop[x]).sum.postln; };
(~z0H*~mtpzYa.flop[0]).sum;
(~z0H*~mtpzYa.flop[1]).sum;
(~z0H*~mtpzYa.flop[2]).sum;

Array.fill(8, { |x| (~z0H*~mtpzYa.flop[x]).sum });

~z0H;
~zIH = Array.fill(8, { |x| (~z0H*~mtpzYa.flop[x]).sum });

~zIIH = Array.fill(8, { |x| (~zIH*~mtpzYa.flop[x]).sum });

~zIIIH = Array.fill(8, { |x| (~zIIH*~mtpzYa.flop[x]).sum.round(1e-3) });

~zIVH = Array.fill(8, { |x| (~zIIIH*~mtpzYa.flop[x]).sum.round(1e-3) });

~zVH = Array.fill(8, { |x| (~zIVH*~mtpzYa.flop[x]).sum.round(1e-3) });

```

```

~zVIH = Array.fill(8, {|x| (~zVH*~mtpzYa.flop[x]).sum.round(1e-3)});

// Iterado, 100 mecanismos en Δt
// Partiendo de screen H f1g1d1,Perturbación H

~z0H = [0,0,0,0,0,0,0,100];
"Etapa Numero: 0".postln; ~z0H.postln;
8.do{|x|
" Etapa Numero: ".post; (x+1).postln;
~z0H = Array.fill(8, {|x| (~z0H*~mtpzYa.flop[x]).sum.round(1e-3)}).postln;
};

// Let us now calculate the state of the 100 mechanisms Z after the first stage
// with the maximal perturbations P applied
// PA
~z0A = [100, 0, 0, 0, 0, 0, 0, 0];
" Etapa Numero: 0".postln; ~z0A.postln;
8.do{|x|
" Etapa Numero: ".post; (x+1).postln;
~z0A = Array.fill(8, {|x| (~z0A*~mtpzYa.flop[x]).sum.round(1e-3)}).postln;
};
// PB
~z0B = [0, 100, 0, 0, 0, 0, 0, 0];
" Etapa Numero: 0".postln; ~z0B.postln;
8.do{|x|
" Etapa Numero: ".post; (x+1).postln;
~z0B = Array.fill(8, {|x| (~z0B*~mtpzYa.flop[x]).sum.round(1e-3)}).postln;
};
// PC
~z0C = [0, 0, 100, 0, 0, 0, 0, 0];
" Etapa Numero: 0".postln; ~z0C.postln;
8.do{|x|
" Etapa Numero: ".post; (x+1).postln;
~z0C = Array.fill(8, {|x| (~z0C*~mtpzYa.flop[x]).sum.round(1e-3)}).postln;
};
// PD

```

```

~z0D = [0, 0, 0, 100, 0, 0, 0, 0];
" Etapa Numero: 0".postln; ~z0D.postln;
8.do{|x|
" Etapa Numero: ".post; (x+1).postln;
~z0D = Array.fill(8, {|x| (~z0D*~mtpzYa.flop[x]).sum.round(1e-3)}).postln;
};
// PE
~z0E = [0, 0, 0, 0, 100, 0, 0, 0];
" Etapa Numero: 0".postln; ~z0E.postln;
8.do{|x|
" Etapa Numero: ".post; (x+1).postln;
~z0E = Array.fill(8, {|x| (~z0E*~mtpzYa.flop[x]).sum.round(1e-3)}).postln;
};
// PF
~z0F = [0, 0, 0, 0, 0, 100, 0, 0];
" Etapa Numero: 0".postln; ~z0F.postln;
8.do{|x|
" Etapa Numero: ".post; (x+2).postln;
~z0F = Array.fill(8, {|x| (~z0F*~mtpzYa.flop[x]).sum.round(1e-3)}).postln;
};

// PG
~z0G = [0, 0, 0, 0, 0, 0, 100, 0];
" Etapa Numero: 0".postln; ~z0G.postln;
8.do{|x|
" Etapa Numero: ".post; (x+1).postln;
~z0G = Array.fill(8, {|x| (~z0G*~mtpzYa.flop[x]).sum.round(1e-3)}).postln;
};

// PH
~z0H = [0, 0, 0, 0, 0, 0, 0, 100];
" Etapa Numero: 0".postln; ~z0H.post;
8.do{|x|
" Etapa Numero: ".post; (x+1).postln;
~z0H = Array.fill(8, {|x| (~z0H*~mtpzYa.flop[x]).sum.round(1e-3)}).postln;
};

```

```
// =====
// Construcción de rejas =====
// =====
```

```
// (Xenakis, 1992: 98-99)
// Regiones de frecuencias
// Seis regiones de frecuencias
```

```
// dentro de cada una de estas regiones se escoge la altura de acuerdo
// a una distribución normal
```

```
// Toma el tercer índice acústico como el central (A3 = 440 Hz) p. 98
```

```
/*
```

Regiones:

```
1: E0 -> E1 (28 – 40) mi grave contrabajo
2: E1 -> D2 (40 – 49)
3: D2 -> Db3 (49 – 61)
4: Db3 -> C4 (61 – 72)
5: C4 -> B4 (72 – 83)
6: B4 -> A5 (83 – 93)
```

```
*/
```

```
// regiones
```

```
~fr = (
```

```
r1: Prand((28..40), inf).asStream,
```

```
r2: Prand((40..49), inf).asStream,
```

```
r3: Prand((49..61), inf).asStream,
```

```
r4: Prand((61..72), inf).asStream,
```

```
r5: Prand((72..83), inf).asStream,
```

```
r6: Prand((83..93), inf).asStream
```

```
);
```

```
// ~fr.r1.value;
```

```
// ~fr[r1].value;
```

```

// agrupadas en f0 y f1
~fReg = (

// f0 regiones 1 – I, 2 – II,
// 5 – III, 6 – IV
f0: (

[ ~fr.r1, ~fr.r2, ~fr.r5, ~fr.r6];

),

// f1 regiones 3 (I, II, III)
// 4 (IV)
f1: (

[ ~fr.r3, ~fr.r4];

)
);

/*
~fReg[f0][0].next; // 1
~fReg[f0][1].next; // 2
~fReg[f0][2].next; // 5
~fReg[f0][3].next; // 6

~fReg[f1][0].next; // 3
~fReg[f1][1].next; // 4
*/

// Regiones de intensidades
~gReg = (

g0: [“[pp]”, “[f]”, “[ff]”],

g1: ([“[pp]”, “[f]”])

```

```

);
// ~gReg[g0]
// ~gReg[g0][0]

// Regiones de densidades
// 1, 3, 9
// Agrupado con los números romanos

~dReg = (
d0: (
i: 3,
ii: 1,
iii: 1,
iv: 9),

d1: (

i: 9,
ii: 3,
iii: 3,
iv: 1 )
);
// ~dReg[d0][i]
~rejaRut = {

arg fase;

var actual;

~frecuencias = ();
~intensidades = ();
~densidades = ();

actual = fase.postln;

~observaScreen = actual;

```



case

{actual == [f0, g0, d0]} { // Screen A

~frecuencias = (

i: ~fReg[f0][0].nextN(~dReg[d0][i]), // región de frecuencias 1 densidad veces

ii: ~fReg[f0][1].nextN(~dReg[d0][ii]), // región de frecuencias 2

iii: ~fReg[f0][2].nextN(~dReg[d0][iii]), // región de frecuencias 5

iv: ~fReg[f0][3].nextN(~dReg[d0][iv]) // región de frecuencias 6

);

~intensidades = (

i: ~gReg[g0][1],

ii: ~gReg[g0][0],

iii: ~gReg[g0][0],

iv: ~gReg[g0][2]

);

// densidades ya está ligado a frecuencias

}

{actual == [f0, g0, d1]} { // Screen B

~frecuencias = (

i: ~fReg[f0][0].nextN(~dReg[d1][i]), // región de frecuencias 1

ii: ~fReg[f0][1].nextN(~dReg[d1][ii]), // región de frecuencias 2

iii: ~fReg[f0][2].nextN(~dReg[d1][iii]), // región de frecuencias 5

iv: ~fReg[f0][3].nextN(~dReg[d1][iv]) // región de frecuencias 6

);

~intensidades = (

i: ~gReg[g0][1],

ii: ~gReg[g0][0],

iii: ~gReg[g0][0],

iv: ~gReg[g0][2]

```
);
}
```

```
{actual == [f0, g1, d0]} { // Screen C
```

```
~frecuencias = (
```

```
i: ~fReg[f0][0].nextN(~dReg[d0][i]), // región de frecuencias 1
ii: ~fReg[f0][1].nextN(~dReg[d0][ii]), // región de frecuencias 2
iii: ~fReg[f0][2].nextN(~dReg[d0][iii]), // región de frecuencias 5
iv: ~fReg[f0][3].nextN(~dReg[d0][iv]) // región de frecuencias 6
);
```

```
~intensidades = (
```

```
i: ~gReg[g1][0],
ii: ~gReg[g1][1],
iii: ~gReg[g1][1],
iv: ~gReg[g1][0]
```

```
);
```

```
}
```

```
{actual == [f0, g1, d1]} { // Screen D
```

```
~frecuencias = (
```

```
i: ~fReg[f0][0].nextN(~dReg[d1][i]), // región de frecuencias 1
ii: ~fReg[f0][1].nextN(~dReg[d1][ii]), // región de frecuencias 2
iii: ~fReg[f0][2].nextN(~dReg[d1][iii]), // región de frecuencias 5
iv: ~fReg[f0][3].nextN(~dReg[d1][iv]) // región de frecuencias 6
);
```

```
~intensidades = (
```

```
i: ~gReg[g1][0],
ii: ~gReg[g1][1],
iii: ~gReg[g1][1],
iv: ~gReg[g1][0]
```

);

}

{actual == [f1, g0, d0]} { // Screen E

~frecuencias = (

i: ~fReg[f1][0].nextN(~dReg[d0][i]), // región de frecuencias 3

ii: ~fReg[f1][0].nextN(~dReg[d0][ii]), // región de frecuencias 3

iii: ~fReg[f1][0].nextN(~dReg[d0][iii]), // región de frecuencias 3

iv: ~fReg[f1][1].nextN(~dReg[d0][iv]) // región de frecuencias 4

);

~intensidades = (

i: ~gReg[g0][1],

ii: ~gReg[g0][0],

iii: ~gReg[g0][0],

iv: ~gReg[g0][2]

);

}

{actual == [f1, g0, d1]} { // Screen F

~frecuencias = (

i: ~fReg[f1][0].nextN(~dReg[d1][i]), // región de frecuencias 3

ii: ~fReg[f1][0].nextN(~dReg[d1][ii]), // región de frecuencias 3

iii: ~fReg[f1][0].nextN(~dReg[d1][iii]), // región de frecuencias 3

iv: ~fReg[f1][1].nextN(~dReg[d1][iv]) // región de frecuencias 4

);

~intensidades = (

i: ~gReg[g0][1],

ii: ~gReg[g0][0],

iii: ~gReg[g0][0],

iv: ~gReg[g0][2]

```
);
}
```

```
{actual == [f1, g1, d0]} { // Screen G
```

```
~frecuencias = (
i: ~fReg[f1][0].nextN(~dReg[d0][i]), // región de frecuencias 3
ii: ~fReg[f1][0].nextN(~dReg[d0][ii]), // región de frecuencias 3
iii: ~fReg[f1][0].nextN(~dReg[d0][iii]), // región de frecuencias 3
iv: ~fReg[f1][1].nextN(~dReg[d0][iv]) // región de frecuencias 4
);
```

```
~intensidades = (
i: ~gReg[g1][0],
ii: ~gReg[g1][1],
iii: ~gReg[g1][1],
iv: ~gReg[g1][0]
);
}
```

```
{actual == [f1, g1, d1]} { // Screen H
```

```
~frecuencias = (
i: ~fReg[f1][0].nextN(~dReg[d1][i]), // región de frecuencias 3
ii: ~fReg[f1][0].nextN(~dReg[d1][ii]), // región de frecuencias 3
iii: ~fReg[f1][0].nextN(~dReg[d1][iii]), // región de frecuencias 3
iv: ~fReg[f1][1].nextN(~dReg[d1][iv]) // región de frecuencias 4
);
```

```
~intensidades = (
i: ~gReg[g1][0],
ii: ~gReg[g1][1],
iii: ~gReg[g1][1],
iv: ~gReg[g1][0]
);
```

```
};
```

```

};

// ===== screens individuales =====
// =====

// esto es para construir una sola reja, para generarla desde la GUI

~rejaRutIndividual = {

arg actual;

~frecuencias = ();
~intensidades = ();
~densidades = ();
~observaScreen = actual;

case

{actual == [f0, \g0, \d0]} { // Screen A

~frecuencias = (
i: ~fReg[f0][0].nextN(~dReg[d0][i]), // región de frecuencias 1 densidad veces
ii: ~fReg[f0][1].nextN(~dReg[d0][ii]), // región de frecuencias 2
iii: ~fReg[f0][2].nextN(~dReg[d0][iii]), // región de frecuencias 5
iv: ~fReg[f0][3].nextN(~dReg[d0][iv]) // región de frecuencias 6
);

~intensidades = (
i: ~gReg[g0][1],
ii: ~gReg[g0][0],
iii: ~gReg[g0][0],
iv: ~gReg[g0][2]
);

// densidades ya está ligado a frecuencias
}

```

```
{actual == [f0, g0, d1]} { // Screen B
```

```
~frecuencias = (
i: ~fReg[f0][0].nextN(~dReg[d1][i]), // región de frecuencias 1
ii: ~fReg[f0][1].nextN(~dReg[d1][ii]), // región de frecuencias 2
iii: ~fReg[f0][2].nextN(~dReg[d1][iii]), // región de frecuencias 5
iv: ~fReg[f0][3].nextN(~dReg[d1][iv]) // región de frecuencias 6
);
```

```
~intensidades = (
i: ~gReg[g0][1],
ii: ~gReg[g0][0],
iii: ~gReg[g0][0],
iv: ~gReg[g0][2]

);
}
```

```
{actual == [f0, g1, d0]} { // Screen C
```

```
~frecuencias = (
i: ~fReg[f0][0].nextN(~dReg[d0][i]), // región de frecuencias 1
ii: ~fReg[f0][1].nextN(~dReg[d0][ii]), // región de frecuencias 2
iii: ~fReg[f0][2].nextN(~dReg[d0][iii]), // región de frecuencias 5
iv: ~fReg[f0][3].nextN(~dReg[d0][iv]) // región de frecuencias 6
);
```

```
~intensidades = (
i: ~gReg[g1][0],
ii: ~gReg[g1][1],
iii: ~gReg[g1][1],
iv: ~gReg[g1][0]

);
}
```

```
{actual == [f0, g1, d1]} { // Screen D
```

```
~frecuencias = (
i: ~fReg[f0][0].nextN(~dReg[d1][i]), // región de frecuencias 1
ii: ~fReg[f0][1].nextN(~dReg[d1][ii]), // región de frecuencias 2
iii: ~fReg[f0][2].nextN(~dReg[d1][iii]), // región de frecuencias 5
iv: ~fReg[f0][3].nextN(~dReg[d1][iv]) // región de frecuencias 6
);
```

```
~intensidades = (
i: ~gReg[g1][0],
ii: ~gReg[g1][1],
iii: ~gReg[g1][1],
iv: ~gReg[g1][0]
```

```
);
```

```
}
```

```
{actual == [f1, g0, d0]} { // Screen E
```

```
~frecuencias = (
i: ~fReg[f1][0].nextN(~dReg[d0][i]), // región de frecuencias 3
ii: ~fReg[f1][0].nextN(~dReg[d0][ii]), // región de frecuencias 3
iii: ~fReg[f1][0].nextN(~dReg[d0][iii]), // región de frecuencias 3
iv: ~fReg[f1][1].nextN(~dReg[d0][iv]) // región de frecuencias 4
);
```

```
~intensidades = (
i: ~gReg[g0][1],
ii: ~gReg[g0][0],
iii: ~gReg[g0][0],
iv: ~gReg[g0][2]
```

```
);
```

```
}
```

```
{actual == [f1, \g0, \d1]} { // Screen F
```

```
~frecuencias = (
i: ~fReg[f1][0].nextN(~dReg[d1][i]), // región de frecuencias 3
ii: ~fReg[f1][0].nextN(~dReg[d1][ii]), // región de frecuencias 3
iii: ~fReg[f1][0].nextN(~dReg[d1][iii]), // región de frecuencias 3
iv: ~fReg[f1][1].nextN(~dReg[d1][iv]) // región de frecuencias 4
);
```

```
~intensidades = (
i: ~gReg[g0][1],
ii: ~gReg[g0][0],
iii: ~gReg[g0][0],
iv: ~gReg[g0][2]
);
}
```

```
{actual == [f1, \g1, \d0]} { // Screen G
```

```
~frecuencias = (
i: ~fReg[f1][0].nextN(~dReg[d0][i]), // región de frecuencias 3
ii: ~fReg[f1][0].nextN(~dReg[d0][ii]), // región de frecuencias 3
iii: ~fReg[f1][0].nextN(~dReg[d0][iii]), // región de frecuencias 3
iv: ~fReg[f1][1].nextN(~dReg[d0][iv]) // región de frecuencias 4
);
```

```
~intensidades = (
i: ~gReg[g1][0],
ii: ~gReg[g1][1],
iii: ~gReg[g1][1],
iv: ~gReg[g1][0]
);
}
```

```
{actual == [f1, \g1, \d1]} { // Screen H
```



```

~frecuencias = (
i: ~fReg[\f1][0].nextN(~dReg[\d1][\i]), // región de frecuencias 3
ii: ~fReg[\f1][0].nextN(~dReg[\d1][\ii]), // región de frecuencias 3
iii: ~fReg[\f1][0].nextN(~dReg[\d1][\iii]), // región de frecuencias 3
iv: ~fReg[\f1][1].nextN(~dReg[\d1][\iv]) // región de frecuencias 4
);

~intensidades = (
i: ~gReg[\g1][0],
ii: ~gReg[\g1][1],
iii: ~gReg[\g1][1],
iv: ~gReg[\g1][0]
);

};

};

// ~rejaRutIndividual.value([ \f0, \g0, \d0 ] );

// ~frecuencias
// primero repartir densidad entre los tres violines

// en los instrumentos 1 el máximo puede ser 4 por  $\Delta t$ 
// en los instrumentos 2 el máximo puede ser 3 por  $\Delta t$ 
// en los instrumentos 3 el máximo puede ser 5 por  $\Delta t$ 

// Observador que dice en que reja se encuentra (implementado dentro de función
~rejaRut)
~observaScreen;
~romanos = ['i', 'ii', 'iii', 'iv'];

/*
Agrupar dentro de una función de una sola condicional:
si es f0 o si es f1
después una función dentro, que pueble los arreglos de frecuencias

```

```

*/
// función global para poblar cualquier screen

~funPoblar = { arg freq;

// Estas se declaran dentro de la función para que sean vaciadas
// cada vez que se le mande un .value a la función

~vln1Alturas = []; ~vln2Alturas = []; ~vln3Alturas = [];
~vc1Alturas = []; ~vc2Alturas = []; ~vc3Alturas = [];
~cb1Alturas = []; ~cb2Alturas = []; ~cb3Alturas = [];

~vln1Dinamicas = []; ~vln2Dinamicas = []; ~vln3Dinamicas = [];
~vc1Dinamicas = []; ~vc2Dinamicas = []; ~vc3Dinamicas = [];
~cb1Dinamicas = []; ~cb2Dinamicas = []; ~cb3Dinamicas = [];

case
// Para poblar rejas A .. D todos los f0's
{ freq[0] == \f0 } {
// entonces los candidatos son:
{ arg x;

// en l siempre serán contrabajos de densidad variable
// E0 – E1
// densidad será 3 o 9
// ScreenA fogodo i = 3
// ScreenC fog1do i = 3
if(~frecuencias['i'].size == 3) {

// Si es 3 entonces es A o C

~cb1Alturas.add(~frecuencias['i'].take(~frecuencias['i'].choose));
~cb2Alturas.add(~frecuencias['i'].take(~frecuencias['i'].choose));
~cb3Alturas.add(~frecuencias['i'].take(~frecuencias['i'].choose));

```

```

~cb1Alturas = (~cb1Alturas ++ (\r ! 3));
~cb2Alturas = (~cb2Alturas ++ (\r ! 2));
~cb3Alturas = (~cb3Alturas ++ (\r ! 4));

~cb1Alturas = ~cb1Alturas.scramble;
~cb2Alturas = ~cb2Alturas.scramble;
~cb3Alturas = ~cb3Alturas.scramble;

// las dinámicas de más serán eliminadas en condicional en Pbind.collect
~cb1Dinamicas.add(~intensidades['i']);
~cb2Dinamicas.add(~intensidades['i']);
~cb3Dinamicas.add(~intensidades['i']);

}
// si no será 9 en i, en los graves
// Screen B
// screen D

{ {
var densidad;
// distribuir 9
densidad = [
[1, 3, 5], [2, 3, 4], [2, 2, 5], [3, 2, 4], [3, 3, 3],[4, 2, 3], [4, 1, 4]
].choose;

densidad[0].do{
~cb1Alturas = ~cb1Alturas.add(~frecuencias['i'].take(~frecuencias['i'].choose))
};
densidad[1].do{
~cb2Alturas = ~cb2Alturas.add(~frecuencias['i'].take(~frecuencias['i'].choose))
};
densidad[2].do{
~cb3Alturas = ~cb3Alturas.add(~frecuencias['i'].take(~frecuencias['i'].choose))
};

```

```

~cb1Alturas = (~cb1Alturas ++ (\r ! (4 - densidad[0]]));
~cb2Alturas = (~cb2Alturas ++ (\r ! (3 - densidad[1]]));
~cb3Alturas = (~cb3Alturas ++ (\r ! (5 - densidad[2]]));

~cb1Alturas = ~cb1Alturas.scramble;
~cb2Alturas = ~cb2Alturas.scramble;
~cb3Alturas = ~cb3Alturas.scramble;

~cb1Dinamicas.add(~intensidades['i']);
~cb2Dinamicas.add(~intensidades['i']);
~cb3Dinamicas.add(~intensidades['i']);
}.value;
};
// en ll vc
// densidad será 1 o 3
// Screens A y C
if(~frecuencias['ii'].size == 1) { { var cual;

cual = [1, 2, 3].choose;

case
{cual == 1} {
// agrupar los silencios en el Pbind con un collect
~vc1Alturas = (~frecuencias['ii'] ++ (\r ! 3)).scramble;
~vc2Alturas = \r ! 3;
~vc3Alturas = \r ! 5;
~vc1Dinamicas.add(~intensidades['ii']);
~vc2Dinamicas.add(~intensidades['ii']);
~vc3Dinamicas.add(~intensidades['ii']);
}

{cual == 2} {
~vc1Alturas = \r ! 4;
~vc2Alturas = (~frecuencias['ii'] ++ (\r ! 2)).scramble;
~vc3Alturas = \r ! 5;

```

```

~vc1Dinamicas.add(~intensidades['ii']);
~vc2Dinamicas.add(~intensidades['ii']);
~vc3Dinamicas.add(~intensidades['ii']);
}
{cual == 3} {
~vc1Alturas = \r ! 4;
~vc2Alturas = \r ! 3;
~vc3Alturas = (~frecuencias['ii'] ++ (\r ! 4)).scramble;

~vc1Dinamicas.add(~intensidades['ii']);
~vc2Dinamicas.add(~intensidades['ii']);
~vc3Dinamicas.add(~intensidades['ii']);
}

}.value;
}

// si no será 3
// Screen B
// Screen D

{
{

~vc1Alturas.add(~frecuencias['ii'].take(~frecuencias['ii'].choose));
~vc2Alturas.add(~frecuencias['ii'].take(~frecuencias['ii'].choose));
~vc3Alturas.add(~frecuencias['ii'].take(~frecuencias['ii'].choose));

~vc1Alturas = (~vc1Alturas ++ (\r ! 3));
~vc2Alturas = (~vc2Alturas ++ (\r ! 2));
~vc3Alturas = (~vc3Alturas ++ (\r ! 4));

~vc1Alturas = ~vc1Alturas.scramble;
~vc2Alturas = ~vc2Alturas.scramble;
~vc3Alturas = ~vc3Alturas.scramble;
~vc1Dinamicas.add(~intensidades['ii']);
~vc2Dinamicas.add(~intensidades['ii']);

```

```

~vc3Dinamicas.add(~intensidades['ii']);
}.value;
};

// en III vc
// densidad será 1 o 3
// si densidad es 1, quiere decir que en ii hay 1, un vc ya tiene una nota
// entonces ...
// Si es un 1 A y C
if(~frecuencias['iii'].size == 1) { { var cual, instArray;

// encuentra cual tiene lugar vacío (silencio) para agregar altura

instArray = [~vc1Alturas.includes(\r), ~vc2Alturas.includes(\r), ~vc3Alturas.includes(\r)];
cual = instArray.detectIndex({ arg x; x == true });
case
{cual == 0} { { var indice;

indice = ~vc1Alturas.detectIndex({ |x| x == \r });
~vc1Alturas.put(indice,
~frecuencias['iii'].take(~frecuencias['iii'].choose)); }.value;

~vc1Dinamicas.add(~intensidades['iii']);

}

{cual == 1} { {var indice;

indice = ~vc2Alturas.detectIndex({ |x| x == \r });
~vc2Alturas.put(indice,
~frecuencias['iii'].take(~frecuencias['iii'].choose)); }.value;

~vc2Dinamicas.add(~intensidades['iii']);

}

{cual == 2} { { var indice;

```

```

indice = ~vc3Alturas.detectIndex({ |x| x == \r });
~vc3Alturas.put(indice,
~frecuencias['iii'].take(~frecuencias['iii'].choose)); }.value;

~vc3Dinamicas.add(~intensidades['iii']);

};

}.value;
}

// si no será 3
// y los cellos ya tienen notas ...
// entonces
// Screen B
// Screen D
{ { var cual, instArray, vc1Indice, vc2Indice, vc3Indice;

/*
// encuentra cual tiene lugar vacío (silencio) para agregar altura
// puede que uno no tenga suficientes lugares vacíos
// entonces ...

instArray = [
~vc1Alturas.occurrencesOf(\r),
~vc2Alturas.occurrencesOf(\r),
~vc3Alturas.occurrencesOf(\r)
];

cual = instArray.detectIndex({ arg x; x >= 3 });
*/
// ~frecuencias['iii'] = ~frecuencias['iii'].flatten
// ~frecuencias['iii'].add([77, 82, 84])
// ~vc1Alturas = [\r, \r, 41, \r]
vc1Indice = ~vc1Alturas.selectIndex({ |x| x == \r}).choose;
~vc1Alturas.put(vc1Indice, ~frecuencias['iii'].take(~frecuencias['iii'].choose));
/*

```

solucion que pone todas las alturas restants dentro de un solo instrumento  
implementado dentro de un case

```

~vc1Alturas = ~vc1Alturas ++ ~frecuencias['iii'];
~vc1Alturas = ~vc1Alturas.reject({ |x| x == \r });
~frecuencias['iii'] = [] ;*/
~vc1Dinamicas.add(~intensidades['iii']);

vc2Indice = ~vc2Alturas.selectIndex({ |x| x == \r }).choose;
~vc2Alturas.put(vc2Indice, ~frecuencias['iii'].take(~frecuencias['iii'].choose));
~vc2Dinamicas.add(~intensidades['iii']);
vc3Indice = ~vc3Alturas.selectIndex({ |x| x == \r }).choose;
~vc3Alturas.put(vc3Indice, ~frecuencias['iii'].take(~frecuencias['iii'].choose));
~vc3Dinamicas.add(~intensidades['iii']);
}.value;
};

```

// en IV violines

// densidad será 1 o 9

```
if(~frecuencias['iv'].size == 1) { { var cual;
```

```
cual = [1, 2, 3].choose;
```

case

```
{cual == 1} {
```

// agrupar los silencios en el Pbind con un collect

```
~vln1Alturas = (~frecuencias['iv'] ++ (\r ! 3)).flat.scramble;
```

```
~vln2Alturas = \r ! 3;
```

```
~vln3Alturas = \r ! 5;
```

```
~vln1Dinamicas.add(~intensidades['iv']);
```

```
~vln2Dinamicas.add(~intensidades['iv']);
```

```
~vln3Dinamicas.add(~intensidades['iv']);
```

```
}
```

```
{cual == 2} {
```

```
~vln1Alturas = \r ! 4;
```



```

~vln2Alturas = (~frecuencias['iv'] ++ (\r ! 2)).flat.scramble;
~vln3Alturas = \r ! 5;

~vln1Dinamicas.add(~intensidades['iv']);
~vln2Dinamicas.add(~intensidades['iv']);
~vln3Dinamicas.add(~intensidades['iv']);
}
{cual == 3} {
~vln1Alturas = \r ! 4;
~vln2Alturas = \r ! 3;
~vln3Alturas = (~frecuencias['iv'] ++ (\r ! 4)).flat.scramble;

~vln1Dinamicas.add(~intensidades['iv']);
~vln2Dinamicas.add(~intensidades['iv']);
~vln3Dinamicas.add(~intensidades['iv']);
}

}.value; }

// si no será 9
{ { var densidad;

// distribuir 9
densidad = [
[1, 3, 5], [2, 3, 4], [2, 2, 5], [3, 2, 4], [3, 3, 3],[4, 2, 3], [4, 1, 4]
].choose;

densidad[0].do{
~vln1Alturas = ~vln1Alturas.add(~frecuencias['iv'].take(~frecuencias['iv'].choose))
};
densidad[1].do{
~vln2Alturas = ~vln2Alturas.add(~frecuencias['iv'].take(~frecuencias['iv'].choose))
};
densidad[2].do{
~vln3Alturas = ~vln3Alturas.add(~frecuencias['iv'].take(~frecuencias['iv'].choose))
};

```

```

~vln1Alturas = (~vln1Alturas ++ (\r ! (4 - densidad[0]]));
~vln2Alturas = (~vln2Alturas ++ (\r ! (3 - densidad[1]]));
~vln3Alturas = (~vln3Alturas ++ (\r ! (5 - densidad[2]]));

~vln1Alturas = ~vln1Alturas.scramble;
~vln2Alturas = ~vln2Alturas.scramble;
~vln3Alturas = ~vln3Alturas.scramble;

~vln1Dinamicas.add(~intensidades['iv']);
~vln2Dinamicas.add(~intensidades['iv']);
~vln3Dinamicas.add(~intensidades['iv']);
}.value;
};
}.value;
}

// si es igual a f1 =====
// si es igual a f1 =====
// si es igual a f1 =====
{ freq[0] == \f1 } {

{ arg x;

// para los i 3 o 9
// i ii y iii comparten el mismo registro
if(~frecuencias['i'].size == 3 ) { {
// Esto corresponde a las rejillas
// E
// G

// si i es 3, en ii y iii sera 1 y 1
// poblar todos los instrumentos para f1g0d0 y f1g1d0

~vc1Alturas.add(~frecuencias['i'].take(~frecuencias['i'].choose));
~vc2Alturas.add(~frecuencias['i'].take(~frecuencias['i'].choose));
~vc3Alturas.add(~frecuencias['i'].take(~frecuencias['i'].choose));

```

```

~vc1Alturas = (~vc1Alturas ++ (\r ! 3));
~vc2Alturas = (~vc2Alturas ++ (\r ! 2));
~vc3Alturas = (~vc3Alturas ++ (\r ! 4));

~vc1Alturas = ~vc1Alturas.scramble;
~vc2Alturas = ~vc2Alturas.scramble;
~vc3Alturas = ~vc3Alturas.scramble;

~vc1Dinamicas.add(~intensidades['i']);
~vc2Dinamicas.add(~intensidades['i']);
~vc3Dinamicas.add(~intensidades['i']);

// hasta aquí i ya está poblado por cellos
// repartir ii y iii en bajos, dejar la posibilidad de hacerlo al revés ... mayor variedad

{ var cual;

cual = [1, 2, 3].choose;

case
{cual == 1} {

~cb1Alturas = (~frecuencias['ii'] ++ (\r ! 3)).flat.scramble;
~cb2Alturas = \r ! 3;
~cb3Alturas = \r ! 5;

~cb1Dinamicas.add(~intensidades['ii']);
}

{cual == 2} {
~cb1Alturas = \r ! 4;
~cb2Alturas = (~frecuencias['ii'] ++ (\r ! 2)).flat.scramble;
~cb3Alturas = \r ! 5;

~cb2Dinamicas.add(~intensidades['ii']);
}

{cual == 3} {

```

```

~cb1Alturas = \r ! 4;
~cb2Alturas = \r ! 3;
~cb3Alturas = (~frecuencias['ii'] ++ (\r ! 4)).flat.scramble;

~cb3Dinamicas.add(~intensidades['ii']);
}
}.value;
{ var instArray, cual;

// encuentra cual tiene lugar vacío (silencio) para agregar altura para iii

instArray = [~cb1Alturas.includes(\r), ~cb2Alturas.includes(\r), ~cb3Alturas.includes(\r)];
cual = instArray.detectIndex({ arg x; x == true });
case
{cual == 0} { { var indice;

indice = ~cb1Alturas.detectIndex({ |x| x == \r });
~cb1Alturas.put(indice,
~frecuencias['iii'].take(~frecuencias['iii'].choose)); }.value;

~cb1Dinamicas.add(~intensidades['iii']);

}

{cual == 1} { {var indice;

indice = ~cb2Alturas.detectIndex({ |x| x == \r });
~cb2Alturas.put(indice,
~frecuencias['iii'].take(~frecuencias['iii'].choose)); }.value;

~cb2Dinamicas.add(~intensidades['iii']);

}

{cual == 2} { { var indice;

```

```

indice = ~cb3Alturas.detectIndex({ |x| x == \r });
~cb3Alturas.put(indice,
~frecuencias['iii'].take(~frecuencias['iii'].choose)); }.value;

~cb3Dinamicas.add(~intensidades['iii']);

};

}.value;

}.value;
}

// si i es 9, ii y iii son 3 y 3
// esto cubre f1g0d1 y f1g1d1, F y H respectivamente

{ {
// primero a poblar i con 9
{ var densidad;

// distribuir 9 dentro de las posibilidades de las grillas rítmicas
// temporalmente entre los cellos, después distribuir entre bajos también
densidad = [
[1, 3, 5], [2, 3, 4], [2, 2, 5], [3, 2, 4], [3, 3, 3],[4, 2, 3], [4, 1, 4]
].choose;

densidad[0].do{
~vc1Alturas = ~vc1Alturas.add(~frecuencias['i'].take(~frecuencias['i'].choose))
};
densidad[1].do{
~vc2Alturas = ~vc2Alturas.add(~frecuencias['i'].take(~frecuencias['i'].choose))
};
densidad[2].do{
~vc3Alturas = ~vc3Alturas.add(~frecuencias['i'].take(~frecuencias['i'].choose))
};
}
}

```

```

~vc1Alturas = (~vc1Alturas ++ (\r ! (4 - densidad[0]]));
~vc2Alturas = (~vc2Alturas ++ (\r ! (3 - densidad[1]]));
~vc3Alturas = (~vc3Alturas ++ (\r ! (5 - densidad[2]]));

~vc1Alturas = ~vc1Alturas.scramble;
~vc2Alturas = ~vc2Alturas.scramble;
~vc3Alturas = ~vc3Alturas.scramble;

~vc1Dinamicas.add(~intensidades['i']);
~vc2Dinamicas.add(~intensidades['i']);
~vc3Dinamicas.add(~intensidades['i']);
}.value;

{
// después ii con 3
~cb1Alturas.add(~frecuencias['ii'].take(~frecuencias['ii'].choose));
~cb2Alturas.add(~frecuencias['ii'].take(~frecuencias['ii'].choose));
~cb3Alturas.add(~frecuencias['ii'].take(~frecuencias['ii'].choose));

~cb1Alturas = (~cb1Alturas ++ (\r ! 3));
~cb2Alturas = (~cb2Alturas ++ (\r ! 2));
~cb3Alturas = (~cb3Alturas ++ (\r ! 4));

~cb1Alturas = ~cb1Alturas.scramble;
~cb2Alturas = ~cb2Alturas.scramble;
~cb3Alturas = ~cb3Alturas.scramble;

~cb1Dinamicas.add(~intensidades['ii']);
~cb2Dinamicas.add(~intensidades['ii']);
~cb3Dinamicas.add(~intensidades['ii']);
}.value;

// finalmente iii con 3 de acuerdo a los espacios que ii dejó disponibles
{ var cual, instArray, cb1Indice, cb2Indice, cb3Indice;

```

```

// encuentra cual tiene lugar vacío (silencio) para agregar altura
// puede que uno no tenga suficientes lugares vacíos
// entonces ...

/*
instArray = [
~cb1Alturas.occurrencesOf(\r),
~cb2Alturas.occurrencesOf(\r),
~cb3Alturas.occurrencesOf(\r)
];

cual = instArray.detectIndex({ arg x; x >= 3 });
*/
/*vc2Indice = ~vc2Alturas.selectIndex({ |x| x == \r }).choose;
~vc2Alturas.put(vc2Indice, ~frecuencias['iii'].take(~frecuencias['iii'].choose));
~vc2Dinamicas.add(~intensidades['iii']);*/
cb1Indice = ~cb1Alturas.selectIndex({ |x| x == \r }).choose;
~cb1Alturas.put(cb1Indice, ~frecuencias['iii'].take(~frecuencias['iii'].choose));
~cb1Dinamicas.add(~intensidades['iii']);
cb2Indice = ~cb2Alturas.selectIndex({ |x| x == \r }).choose;
~cb2Alturas.put(cb2Indice, ~frecuencias['iii'].take(~frecuencias['iii'].choose));
~cb2Dinamicas.add(~intensidades['iii']);
cb3Indice = ~cb3Alturas.selectIndex({ |x| x == \r }).choose;
~cb3Alturas.put(cb3Indice, ~frecuencias['iii'].take(~frecuencias['iii'].choose));
~cb3Dinamicas.add(~intensidades['iii']);

}.value;
}.value;
};
// para los iv
// vlns puede ser 9 o 1
// si es 1
if(~frecuencias['iv'].size == 1) { { var cual;

cual = [1, 2, 3].choose;

```

```

case
{cual == 1} {
// agrupar los silencios en el Pbind con un collect
~vln1Alturas = (~frecuencias['iv'] ++ (\r ! 3)).flat.scramble;
~vln2Alturas = \r ! 3;
~vln3Alturas = \r ! 5;

~vln1Dinamicas.add(~intensidades['iv']);
~vln2Dinamicas.add(~intensidades['iv']);
~vln3Dinamicas.add(~intensidades['iv']);
}

{cual == 2} {
~vln1Alturas = \r ! 4;
~vln2Alturas = (~frecuencias['iv'] ++ (\r ! 2)).flat.scramble;
~vln3Alturas = \r ! 5;

~vln1Dinamicas.add(~intensidades['iv']);
~vln2Dinamicas.add(~intensidades['iv']);
~vln3Dinamicas.add(~intensidades['iv']);
}

{cual == 3} {
~vln1Alturas = \r ! 4;
~vln2Alturas = \r ! 3;
~vln3Alturas = (~frecuencias['iv'] ++ (\r ! 4)).flat.scramble;

~vln1Dinamicas.add(~intensidades['iv']);
~vln2Dinamicas.add(~intensidades['iv']);
~vln3Dinamicas.add(~intensidades['iv']);
}

}.value;
}

// si no, es 9
{{ var densidad;

```



```

// distribuir 9
densidad = [
[1, 3, 5], [2, 3, 4], [2, 2, 5], [3, 2, 4], [3, 3, 3],[4, 2, 3], [4, 1, 4]
].choose;

densidad[0].do{
~vln1Alturas = ~vln1Alturas.add(~frecuencias['iv'].take(~frecuencias['iv'].choose))
};
densidad[1].do{
~vln2Alturas = ~vln2Alturas.add(~frecuencias['iv'].take(~frecuencias['iv'].choose))
};
densidad[2].do{
~vln3Alturas = ~vln3Alturas.add(~frecuencias['iv'].take(~frecuencias['iv'].choose))
};

~vln1Alturas = (~vln1Alturas ++ (\r ! (4 - densidad[0]]));
~vln2Alturas = (~vln2Alturas ++ (\r ! (3 - densidad[1]]));
~vln3Alturas = (~vln3Alturas ++ (\r ! (5 - densidad[2]]));

~vln1Alturas = ~vln1Alturas.scramble;
~vln2Alturas = ~vln2Alturas.scramble;
~vln3Alturas = ~vln3Alturas.scramble;

~vln1Dinamicas.add(~intensidades['iv']);
~vln2Dinamicas.add(~intensidades['iv']);
~vln3Dinamicas.add(~intensidades['iv']);

}.value;
}

}.value
};

// Dentro de la misma función funPoblar, vaciar los instrumentos para empezar nueva
reja
// Y dejar su contenido dentro de arreglos globales que contienen toda la información

```

```

~vln1AlturasGlobal = ~vln1AlturasGlobal ++ ~vln1Alturas;
~vln2AlturasGlobal = ~vln2AlturasGlobal ++ ~vln2Alturas;
~vln3AlturasGlobal = ~vln3AlturasGlobal ++ ~vln3Alturas;

~vc1AlturasGlobal = ~vc1AlturasGlobal ++ ~vc1Alturas.scramble;
~vc2AlturasGlobal = ~vc2AlturasGlobal ++ ~vc2Alturas;
~vc3AlturasGlobal = ~vc3AlturasGlobal ++ ~vc3Alturas;
~cb1AlturasGlobal = ~cb1AlturasGlobal ++ ~cb1Alturas;
~cb2AlturasGlobal = ~cb2AlturasGlobal ++ ~cb2Alturas;
~cb3AlturasGlobal = ~cb3AlturasGlobal ++ ~cb3Alturas;
~vln1DinamicasGlobal = ~vln1DinamicasGlobal ++ ~vln1Dinamicas;
~vln2DinamicasGlobal = ~vln2DinamicasGlobal ++ ~vln2Dinamicas;
~vln3DinamicasGlobal = ~vln3DinamicasGlobal ++ ~vln3Dinamicas;

~vc1DinamicasGlobal = ~vc1DinamicasGlobal ++ ~vc1Dinamicas;
~vc2DinamicasGlobal = ~vc2DinamicasGlobal ++ ~vc2Dinamicas;
~vc3DinamicasGlobal = ~vc3DinamicasGlobal ++ ~vc3Dinamicas;

~cb1DinamicasGlobal = ~cb1DinamicasGlobal ++ ~cb1Dinamicas;
~cb2DinamicasGlobal = ~cb2DinamicasGlobal ++ ~cb2Dinamicas;
~cb3DinamicasGlobal = ~cb3DinamicasGlobal ++ ~cb3Dinamicas;

};
// Para eliminar contenido y comenzar otro ciclo

~losGlobales = [
'vln1AlturasGlobal',
'vln2AlturasGlobal',
'vln3AlturasGlobal',
'vc1AlturasGlobal',
'vc2AlturasGlobal',
'vc3AlturasGlobal',
'cb1AlturasGlobal',
'cb2AlturasGlobal',
'cb3AlturasGlobal',
'vln1DinamicasGlobal',

```

```

'vln2DinamicasGlobal',
'vln3DinamicasGlobal',
'vc1DinamicasGlobal',
'vc2DinamicasGlobal',
'vc3DinamicasGlobal',
'cb1DinamicasGlobal',
'cb2DinamicasGlobal',
'cb3DinamicasGlobal'
];
~vaciar = ({
var env = currentEnvironment;

~losGlobales.do( { | item |
env.put( item, [] );
} );
};
);

~alles = {

var todo;

~restAgVln1 = [];
~artVln1 = [];

~vln1 = Pbind(

\cuentaDur, Pseq((0..3), inf),
\cuenta, Pseq([0, Pseries(0, 1, inf)]),
\instrument, "violin",
\part, "vln1",
\time, Pseq([0, Pseq("+", inf)], inf), // es necesario poner time para ensambles
\midinote, Pseq( ~vln1AlturasGlobal , inf),
\restAg, Pkey(\midinote),
\mark, Pstutter(4, Pseq(~vln1DinamicasGlobal, inf)), // Pstutter como un módulo
\dur, 0.25,
// \cuenta, if(Pkey(\dur) == 0.25, ~cuenta, Pstutter(2, ~cuenta)),

```

```

).asStream.collect({ |ev|

~artVln1 = ~artVln1 ++ ev[\mark];
// ~restAg = ~restAg ++ ev[\midinote];

Pbind(
  \instrument, ev[\instrument],
  \part, ev[\part],
  \time, ev[\time],
  \midinote, ev[\midinote],
  \mark, if((ev[\mark] == ~artVln1[ev[\cuenta]]).or(ev[\mark] == nil ).or(ev[\midinote] == \r),
  if((ev[\cuenta] == 0).and(ev[\midinote] != \r), ev[\mark], " "),
  ev[\mark]),
  // \mark, ev[\mark],
  \dur, ev[\dur]
  // \dur, if((ev[\midinote] == \r).and(~restAg[ev[\cuenta]] == \r), 0.5, ev[\dur])

).asStream.next(())

});

~restAgVln2 = [];
~artVln2 = [];

~vln2 = Pbind(

  \cuentaDur, Pseq((0..3), inf),
  \cuenta, Pseq([0, Pseries(0, 1, inf)]),
  \instrument, "violin",
  \part, "vln2",
  \time, Pseq([0, Pseq("+", inf)], inf), // es necesario poner time para ensambles
  \midinote, Pseq( ~vln2AlturasGlobal , inf),
  \restAg, Pkey(\midinote),
  \mark, Pstutter(3, Pseq(~vln2DinamicasGlobal, inf)), // Pstutter como un módulo
  \dur, 1/3

).asStream.collect({ |ev|

```

```

~artVln2 = ~artVln2 ++ ev[\mark];
// ~restAg = ~restAg ++ ev[\midinote];

Pbind(
  \instrument, ev[\instrument],
  \part, ev[\part],
  \time, ev[\time],
  \midinote, ev[\midinote],
  \mark, if((ev[\mark] == ~artVln2[ev[\cuenta]]).or(ev[\mark] == nil ).or(ev[\midinote] == \r),
  if((ev[\cuenta] == 0).and(ev[\midinote] != \r), ev[\mark], " "),
  ev[\mark]),
  // \mark, ev[\mark],
  \dur, ev[\dur]

).asStream.next(())

});
~restAgVln3 = [];
~artVln3 = [];

~vln3 = Pbind(

  \cuentaDur, Pseq((0..3), inf),
  \cuenta, Pseq([0, Pseries(0, 1, inf)]),
  \instrument, "violin",
  \part, "vln3",
  \time, Pseq([0, Pseq("+", inf)], inf), // es necesario poner time para ensambles
  \midinote, Pseq( ~vln3AlturasGlobal , inf),
  \restAg, Pkey(\midinote),
  \mark, Pstutter(5, Pseq(~vln3DinamicasGlobal, inf)), // Pstutter como un módulo
  \dur, 1/5

).asStream.collect({ |ev|

~artVln3 = ~artVln3 ++ ev[\mark];
// ~restAg = ~restAg ++ ev[\midinote];

```

```

Pbind(
  \instrument, ev[\instrument],
  \part, ev[\part],
  \time, ev[\time],
  \midinote, ev[\midinote],
  \mark, if((ev[\mark] == ~artVln3[ev[\cuenta]]).or(ev[\mark] == nil ).or(ev[\midinote] == \r),
  if((ev[\cuenta] == 0).and(ev[\midinote] != \r), ev[\mark], " "),
  ev[\mark]),
  // \mark, ev[\mark],
  \dur, ev[\dur]

).asStream.next(())

});
~artVc1 = [];

~vc1 = Pbind(
  \cuenta, Pseq([0, Pseries(0, 1, inf)]),
  \instrument, "cello",
  \part, "vc1",
  \time, Pseq([0, Pseq("+", inf)], inf),
  \midinote, Pseq( ~vc1AlturasGlobal ,inf),
  \mark, Pstutter(4, Pseq(~vc1DinamicasGlobal, inf)), // Pstutter como un módulo
  \dur, 0.25
).asStream.collect({ |ev|

~artVc1 = ~artVc1 ++ ev[\mark];

Pbind(

  \instrument, ev[\instrument],
  \part, ev[\part],
  \time, ev[\time],
  \midinote, ev[\midinote],
  \mark, if((ev[\mark] == ~artVc1[ev[\cuenta]]).or(ev[\mark] == nil ).or(ev[\midinote] == \r),
  if((ev[\cuenta] == 0).and(ev[\midinote] != \r), ev[\mark], " "),
  ev[\mark]),

```

```

// \mark, ev[\mark],
\dur, ev[\dur]
// \dur, if((ev[\midinote] == \r).and(~restAg[ev[\cuenta]] == \r), 0.5, ev[\dur])

).asStream.next(());
});
~artVc2 = [];

~vc2 = Pbind(
  \cuenta, Pseq([0, Pseries(0, 1, inf)]),
  \instrument, "cello",
  \part, "vc2",
  \time, Pseq([0, Pseq("+", inf)], inf),
  \midinote, Pseq( ~vc2AlturasGlobal ,inf),
  \mark, Pstutter(3, Pseq(~vc2DinamicasGlobal, inf)), // Pstutter como un módulo
  \dur, 1/3
).asStream.collect({ |ev|

~artVc2 = ~artVc2 ++ ev[\mark];

Pbind(

  \instrument, ev[\instrument],
  \part, ev[\part],
  \time, ev[\time],
  \midinote, ev[\midinote],
  \mark, if((ev[\mark] == ~artVc2[ev[\cuenta]]).or(ev[\mark] == nil ).or(ev[\midinote] == \r),
  if((ev[\cuenta] == 0).and(ev[\midinote] != \r), ev[\mark], " "),
  ev[\mark]),
  // \mark, ev[\mark],
  \dur, ev[\dur]
  // \dur, if((ev[\midinote] == \r).and(~restAg[ev[\cuenta]] == \r), 0.5, ev[\dur])

).asStream.next(());
});
~artVc3 = [];

```

```

~vc3 = Pbind(
  \cuenta, Pseq([0, Pseries(0, 1, inf)]),
  \instrument, "cello",
  \part, "vc3",
  \time, Pseq([0, Pseq("+", inf)], inf),
  \midinote, Pseq( ~vc3AlturasGlobal ,inf),
  \mark, Pstutter(5, Pseq(~vc3DinamicasGlobal, inf)), // Pstutter como un módulo
  \dur, 1/5
).asStream.collect({ |ev|

~artVc1 = ~artVc1 ++ ev[\mark];

Pbind(

  \instrument, ev[\instrument],
  \part, ev[\part],
  \time, ev[\time],
  \midinote, ev[\midinote],
  \mark, if((ev[\mark] == ~artVc1[ev[\cuenta]]).or(ev[\mark] == nil ).or(ev[\midinote] == \r),
  if((ev[\cuenta] == 0).and(ev[\midinote] != \r), ev[\mark], " "),
  ev[\mark]),
  // \mark, ev[\mark],
  \dur, ev[\dur]
  // \dur, if((ev[\midinote] == \r).and(~restAg[ev[\cuenta]] == \r), 0.5, ev[\dur])

).asStream.next(());
});
~artCb1 = [];

~cb1 = Pbind(
  \cuenta, Pseq([0, Pseries(0, 1, inf)]),
  \instrument, "contrabass",
  \part, "cb1",
  \time, Pseq([0, Pseq("+", inf)], inf),
  \midinote, Pseq( ~cb1AlturasGlobal ,inf),
  \mark, Pstutter(4, Pseq(~cb1DinamicasGlobal, inf)), // Pstutter como un módulo

```



```

\dur, 0.25
).asStream.collect({ |ev|

~artCb1 = ~artCb1 ++ ev[\mark];

Pbind(

\instrument, ev[\instrument],
\part, ev[\part],
\time, ev[\time],
\midinote, ev[\midinote],
\mark, if((ev[\mark] == ~artCb1[ev[\cuenta]]).or(ev[\mark] == nil ).or(ev[\midinote] == \r),
if((ev[\cuenta] == 0).and(ev[\midinote] != \r), ev[\mark], " "),
ev[\mark]),
// \mark, ev[\mark],
\dur, ev[\dur]
// \dur, if((ev[\midinote] == \r).and(~restAg[ev[\cuenta]] == \r), 0.5, ev[\dur])

).asStream.next(());
});
~artCb2 = [];

~cb2 = Pbind(
\cuenta, Pseq([0, Pseries(0, 1, inf)]),
\instrument, "contrabass",
\part, "cb2",
\time, Pseq([0, Pseq("+", inf)], inf),
\midinote, Pseq( ~cb2AlturasGlobal ,inf),
\mark, Pstutter(3, Pseq(~cb2DinamicasGlobal, inf)), // Pstutter como un módulo
\dur, 1/3
).asStream.collect({ |ev|

~artCb2 = ~artCb2 ++ ev[\mark];

Pbind(

```

```

\instrument, ev[\instrument],
\part, ev[\part],
\time, ev[\time],
\midinote, ev[\midinote],
\mark, if((ev[\mark] == ~artCb2[ev[\cuenta]]).or(ev[\mark] == nil ).or(ev[\midinote] == \r),
if((ev[\cuenta] == 0).and(ev[\midinote] != \r), ev[\mark], " "),
ev[\mark]),
// \mark, ev[\mark],
\dur, ev[\dur]
// \dur, if((ev[\midinote] == \r).and(~restAg[ev[\cuenta]] == \r), 0.5, ev[\dur])

).asStream.next(());
});
~artCb3 = [];

~cb3 = Pbind(
\cuenta, Pseq([0, Pseries(0, 1, inf)]),
\instrument, "contrabass",
\part, "cb3",
\time, Pseq([0, Pseq("+", inf)], inf),
\midinote, Pseq( ~cb3AlturasGlobal ,inf),
\mark, Pstutter(5, Pseq(~cb3DinamicasGlobal, inf)), // Pstutter como un módulo
\dur, 1/5
).asStream.collect({ |ev|

~artCb3 = ~artCb3 ++ ev[\mark];

Pbind(

\instrument, ev[\instrument],
\part, ev[\part],
\time, ev[\time],
\midinote, ev[\midinote],
\mark, if((ev[\mark] == ~artCb3[ev[\cuenta]]).or(ev[\mark] == nil ).or(ev[\midinote] == \r),
if((ev[\cuenta] == 0).and(ev[\midinote] != \r), ev[\mark], " "),
ev[\mark]),

```

```

// \mark, ev[\mark],
\dur, ev[\dur]
// \dur, if((ev[\midinote] == \r).and(~restAg[ev[\cuenta]] == \r), 0.5, ev[\dur])

).asStream.next(());
});
// Violines =====

~vln1EvArr = Array.fill( ~vln1AlturasGlobal.size, { ~vln1.next(() ) });
~vln2EvArr = Array.fill( ~vln2AlturasGlobal.size, { ~vln2.next(() ) });
~vln3EvArr = Array.fill( ~vln3AlturasGlobal.size, { ~vln3.next(() ) });

~vln1Seq = Pseq(~vln1EvArr, 1).asStream;
~vln2Seq = Pseq(~vln2EvArr, 1).asStream;
~vln3Seq = Pseq(~vln3EvArr, 1).asStream;

// Cellos =====

~vc1EvArr = Array.fill( ~vc1AlturasGlobal.size, { ~vc1.next(() ) });
~vc2EvArr = Array.fill( ~vc2AlturasGlobal.size, { ~vc2.next(() ) });
~vc3EvArr = Array.fill( ~vc3AlturasGlobal.size, { ~vc3.next(() ) });

~vc1Seq = Pseq(~vc1EvArr, 1).asStream;
~vc2Seq = Pseq(~vc2EvArr, 1).asStream;
~vc3Seq = Pseq(~vc3EvArr, 1).asStream;

// Contrabajos =====

~cb1EvArr = Array.fill( ~cb1AlturasGlobal.size, { ~cb1.next(() ) });
~cb2EvArr = Array.fill( ~cb2AlturasGlobal.size, { ~cb2.next(() ) });
~cb3EvArr = Array.fill( ~cb3AlturasGlobal.size, { ~cb3.next(() ) });

~cb1Seq = Pseq(~cb1EvArr, 1).asStream;
~cb2Seq = Pseq(~cb2EvArr, 1).asStream;
~cb3Seq = Pseq(~cb3EvArr, 1).asStream;

// Todos los binds empaquetados en un Pseq =====

```

```

~bindsTodos = Pseq([
~vln1Seq,
~vln2Seq,
~vln3Seq,
~vc1Seq,
~vc2Seq,
~vc3Seq,
~cb1Seq,
~cb2Seq,
~cb3Seq
],1).asStream;

```

```

~total = (
~vln1AlturasGlobal.size +
~vln2AlturasGlobal.size +
~vln3AlturasGlobal.size +
~vc1AlturasGlobal.size +
~vc2AlturasGlobal.size +
~vc3AlturasGlobal.size +
~cb1AlturasGlobal.size +
~cb2AlturasGlobal.size +
~cb3AlturasGlobal.size
);

```

```

todo = Fomus2(~bindsTodos, ~total).multi_(9) // multi: numero de instrumentos

```

```

.instPartId_([
~vln1", ~vln2", ~vln3",
~vc1", ~vc2", ~vc3",
~cb1", ~cb2", ~cb3"
]) // arreglo de id de instrumentos
.instrumentName_([
~Vln1", ~Vln2", ~Vln3",
~Vc1", ~Vc2", ~Vc3",
~Cb1", ~Cb2", ~Cb3"
]) // arreglo de nombre de instrumentos
.instrument_([

```

```

“violin”, “violin”, “violin”,
“cello”, “cello”, “cello”,
“contrabass”, “contrabass”, “contrabass”
]) // arreglo de instrumento de instDef en fomis
.layoutDef_ (“small-ensemble”)
.tuplets_ (“5”)
.beat_ (“1/2”)
.timeSig_ (“(2 2)");
todo.ly;
todo.xml;

};

)
// A comenzar el GUI!

(
var ventana, popup, popupFase, botonIndividual, cuantosIndividual, cuantosE, faseE,
botonE, opciones;

~vaciar.value;

opciones = [“Fase Estacionaria”, “P1A”, “P1B”, “P1C”, “P1D”, “P1E”, “P1F”, “P1G”,
“P1H”];

ventana = Window(“Xenologique A”, Rect(100, 100, 450, 130), scroll: true);
// .background_(Color.new255(255.0.rand, 255.0.rand, 255.0.rand, 255.0.rand));
// Cual reja individual se quiere generar
popup = EZPopUpMenu(ventana, Rect(10, 10, 230, 25), “Screen Individual: “,
[
\screnA -> {
“Screen A seleccionada”.postln;
~reja = \screnA;
},
\screnB -> {
“Screen B seleccionada”.postln;
~reja = \screnB;
}
]
)

```

```

},
\screenC -> {
"Screen C seleccionada".postln;
~reja = \screenC;
},
\screenD -> {
"Screen D seleccionada".postln;
~reja = \screenD;
},
\screenE -> {
"Screen E seleccionada".postln;
~reja = \screenE;
},
\screenF -> {
"Screen F seleccionada".postln;
~reja = \screenF;
},
\screenG -> {
"Screen G seleccionada".postln;
~reja = \screenG;
},
\screenH -> {
"Screen H seleccionada".postln;
~reja = \screenH;
}

], initVal: 0, initAction: true);

// Define cuantas rejass individuales x se quiere generar
cuantosIndividual = NumberBox(ventana, Rect(250, 10, 45, 25)).clipLo_(1)
.action_({ arg num;
~num = num.value;
("Generar " ++ num.value ++ " valores").postln;
});
cuantosIndividual.valueAction_(1);
// Estado 1 Genera

```

```

// Estado 2 Limpia para generar siguiente seleccionada
botonIndividual = Button(ventana, Rect(340, 5, 70, 50)).states_([
["Generar ", Color.black, Color.cyan],
["Generando ", Color.black, Color.red]
]).action_({ |boton|

case
{ boton.value == 0 } { ~vaciar.value; "Limpio!".postln;}
{ boton.value == 1 } {
fork{ ("Generando " ++ ~num ++ " " ++ ~reja ).postln;

~num.do{ ~rejaRutIndividual.value(~screens[~reja]);
~funPoblar.value(~observaScreen);

~alles.value;

4.wait;

{botonIndividual.valueAction_(0)}.defer;

}
}

});

popupFase = EZPopUpMenu(ventana, Rect(10, 70, 230, 25), "Fase ",
[
\FaseEstacionaria -> { arg sel;
"Fase Estacionaria seleccionada".postln;
~faseSel = sel.value;

},

\perturbacionA1 -> { arg sel;
"Perturbacion A1".postln;
~faseSel = sel.value;

},

```

```

\perturbacioB1 -> { arg sel;
"Perturbacion B1".postln;
~faseSel = sel.value;
},
\perturbacionC1 -> { arg sel;
"Perturbacion C1".postln;
~faseSel = sel.value;
},
\perturbacionD1 -> { arg sel;
"Perturbacion D1".postln;
~faseSel = sel.value;
},
\perturbacionE1 -> { arg sel;
"Perturbacion E1".postln;
~faseSel = sel.value;
},
\perturbacionF1 -> { arg sel;
"Perturbacion F1".postln;
~faseSel = sel.value;
},
\perturbacionG1 -> { arg sel;
"Perturbacion G1".postln;
~faseSel = sel.value;
},
\perturbacionH1 -> { arg sel;
"Perturbacion H1".postln;
~faseSel = sel.value;
}

], initVal: 0, initAction: true);

cuantosE = NumberBox(ventana, Rect(250, 70, 45, 25)).clipLo_(1)
.action_({ arg num;
~numE = num.value;
("Generar " ++ num.value ++ " valores de " ++ opciones[~faseSel]).postln;

```



```

});
cuantosE.valueAction_(1);

botonE = Button(ventana, Rect(340, 70, 70, 50)).states_([
["Generar ", Color.black, Color.cyan],
["Generando ", Color.black, Color.red]
]).action_({ |boton|

case
{ boton.value == 0 } {
~vaciar.value;
~rejaRut.reset;
~rutinaEstacionaria.reset;
~rutinaPerturbada.reset;
"Limpio!".postln;}

{ boton.value == 1 } {

case
{~faseSel == 0} { // Fase estacionaria

fork{ ("Generando " ++ ~numE ++ " " ++ "Fase Estacionaria" ).postln;

~numE.do{ ~rejaRut.value(~rutinaEstacionaria.value);
~funPoblar.value(~observaScreen);};

~alles.value;

4.wait;
{botonE.valueAction_(0)}.defer;
}
}

{~faseSel == 1 } { // Perturbación P1A

fork{ ("Generando " ++ ~numE ++ " " ++ "Perturbación P1A" ).postln;

```

```

~numE.do{ ~rejaRut.value(~rutinaPerturbada.value(~p1A));
~funPoblar.value(~observaScreen);};

~alles.value;

4.wait;
{botonE.valueAction_(0)}.defer;
};
}

{~faseSel == 2 } { // Perturbación P1B

fork{("Generando " ++ ~numE ++ " " ++ "Perturbación P1B" ).postln;

~numE.do{ ~rejaRut.value(~rutinaPerturbada.value(~p1B));
~funPoblar.value(~observaScreen);};

~alles.value;
4.wait;
{botonE.valueAction_(0)}.defer;
};
}

{~faseSel == 3 } { // Perturbación P1C

fork{("Generando " ++ ~numE ++ " " ++ "Perturbación P1C" ).postln;

~numE.do{ ~rejaRut.value(~rutinaPerturbada.value(~p1C));
~funPoblar.value(~observaScreen);};

~alles.value;
4.wait;
{botonE.valueAction_(0)}.defer;
};
}

{~faseSel == 4 } { // Perturbación P1D

```

```
fork{("Generando " ++ ~numE ++ " " ++ "Perturbación P1D" ).postln;
```

```
~numE.do{ ~rejaRut.value(~rutinaPerturbada.value(~p1D));
```

```
~funPoblar.value(~observaScreen);};
```

```
~alles.value;
```

```
4.wait;
```

```
{botonE.valueAction_(0)}.defer;
```

```
};
```

```
}
```

```
{~faseSel == 5 } { // Perturbación P1E
```

```
fork{("Generando " ++ ~numE ++ " " ++ "Perturbación P1E" ).postln;
```

```
~numE.do{ ~rejaRut.value(~rutinaPerturbada.value(~p1E));
```

```
~funPoblar.value(~observaScreen);};
```

```
~alles.value;
```

```
4.wait;
```

```
{botonE.valueAction_(0)}.defer;
```

```
}
```

```
}
```

```
{~faseSel == 6 } { // Perturbación P1F
```

```
fork{("Generando " ++ ~numE ++ " " ++ "Perturbación P1F" ).postln;
```

```
~numE.do{ ~rejaRut.value(~rutinaPerturbada.value(~p1F));
```

```
~funPoblar.value(~observaScreen);};
```

```
~alles.value;
```

```
4.wait;
```

```
{botonE.valueAction_(0)}.defer;
```

```
};
```

```
}
```

```

{~faseSel == 7 } { // Perturbación P1G

fork{("Generando " ++ ~numE ++ " " ++ "Perturbación P1G" ).postln;

~numE.do{ ~rejaRut.value(~rutinaPerturbada.value(~p1G));
~funPoblar.value(~observaScreen)};

~alles.value;
4.wait;
{botonE.valueAction_(0)}.defer;
};
}

{~faseSel == 8 } { // Perturbación P1H

fork{("Generando " ++ ~numE ++ " " ++ "Perturbación P1H" ).postln;

~numE.do{ ~rejaRut.value(~rutinaPerturbada.value(~p1H));
~funPoblar.value(~observaScreen)};

~alles.value;
4.wait;
{botonE.valueAction_(0)}.defer;
}
}

});
ventana.front;
)

```

# APÉNDICE V

## EJEMPLO DE FASE ESTACIONARIA E POR IMPLEMENTACIÓN

The musical score is arranged in three systems. The first system contains Violin 1 (Vln1), Violin 2 (Vln2), and Violin 3 (Vln3). The second system contains Viola 1 (Vc1), Viola 2 (Vc2), and Viola 3 (Vc3). The third system contains Cello 1 (Cb1), Cello 2 (Cb2), and Cello 3 (Cb3). The music is in 2/2 time and features various dynamic markings and articulations.

**Violin 1 (Vln1):** Treble clef, 2/2 time. Dynamics: *ff*. Features a trill in the first measure and an 8va trill in the third measure.

**Violin 2 (Vln2):** Treble clef, 2/2 time. Dynamics: *ff*, *pp*, *ff*. Features triplets in the first and second measures, and an 8va trill in the third measure.

**Violin 3 (Vln3):** Treble clef, 2/2 time. Dynamics: *ff*, *pp*, *ff*. Features quintuplets in the first and second measures, and an 8va trill in the third measure.

**Viola 1 (Vc1):** Alto clef, 2/2 time. Dynamics: *pp*, *f*, *pp*. Features a trill in the first measure and a trill in the third measure.

**Viola 2 (Vc2):** Alto clef, 2/2 time. Dynamics: *pp*, *f*, *pp*. Features triplets in the first and second measures, and a triplet in the third measure.

**Viola 3 (Vc3):** Alto clef, 2/2 time. Dynamics: *pp*, *f*, *pp*. Features quintuplets in the first and second measures, and a quintuplet in the third measure.

**Cello 1 (Cb1):** Bass clef, 2/2 time. Dynamics: *f*, *pp*, *f*. Features a trill in the first measure and a trill in the third measure.

**Cello 2 (Cb2):** Bass clef, 2/2 time. Dynamics: *f*. Features triplets in the first and second measures, and a triplet in the third measure.

**Cello 3 (Cb3):** Bass clef, 2/2 time. Dynamics: *f*. Features quintuplets in the first and second measures, and a quintuplet in the third measure.

2

The image displays a musical score for three systems, each consisting of three staves. The first system is in treble clef, while the second and third systems are in bass clef. The score is divided into four measures. The first measure includes a dynamic marking of *pp* and an *8va* instruction. The second measure features *pp* and *8va* markings. The third measure is marked *ff* and includes an *8va* instruction. The fourth measure is marked *pp*. The notation includes various rhythmic values, accidentals, and articulation marks. Fingerings are indicated by numbers 3 and 5. The score concludes with a double bar line.

The image displays a musical score for three systems, each consisting of three staves. The notation includes various dynamics such as *pp* (pianissimo), *ff* (fortissimo), and *f* (forte), along with articulations like slurs, accents, and fingerings (3 and 5). The first system is marked with *8va-* and includes a measure with a *pp* dynamic. The second system features a variety of dynamics including *f*, *pp*, and *f*. The third system also includes *pp*, *f*, and *pp* dynamics. The score is written in a key signature with one sharp (F#) and a time signature of 3/4. The notation is complex, with many slurs and articulations throughout.

4

12

The image displays a musical score for three systems, each consisting of three staves. The first system is in treble clef, while the second and third systems are in bass clef. The score is divided into four measures per system. The first system includes dynamic markings such as *ff*, *pp*, and *8va*. The second system includes *pp*, *f*, and *pp f*. The third system includes *f*, *pp*, and *f pp*. The notation includes various rhythmic values, accidentals, and articulation marks. Fingerings are indicated by numbers 3 and 5. Trills and triplets are also present. The score concludes with a double bar line.



## BIBLIOGRAFÍA

Antonopoulos, Antonios, *Silence in time continuum as a stochastic process in Iannis Xenakis's instrumental work*, Proceedings of the International Conference Beyond the Centers: Musical Avant-Gardes since 1950, Thessalonki, Grecia, 1-3 julio, 2010, <http://btc.web.auth.gr/>

Bolter, Jay David and Grusin, Richard; *Remediation: Understanding New Media*, MIT Press, 2000.

Bewley, John, *Lejaren A. Hiller: Computer Music Pioneer*, 2004, consultado en: <http://library.buffalo.edu/music/exhibits/hillerexhibitssummary.pdf>

Chandler, Daniel, *Semiotics: The Basics*, Routledge, New York, 2007.

Collins, Nick; Wilson, Scott; Cottle, David (eds.), McCartney, James (fwd.); *The SuperCollider Book*, 2011.

Cottle, David Michael, *Computer Music with Examples in SuperCollider*, 2005.

Cramer, Florian; Fuller, Mathew (ed.); *Words Made Flesh: Code, Culture, Imagination*, Media Design Research Piet Zwart Institute, Rotterdam, 2005.

Creeber, Glen and Royston, Martin (eds.), *Digital Culture: Understanding New Media*, Open University Press, 2008.

Da Silva, Patrício, *David Cope and Experiments in Musical Intelligence*, [www.spectrumpress.com](http://www.spectrumpress.com), 2003.

Deuze, Mark; *Participation, Remediation, Bricolage: Considering Principal Components of a Digital Culture*, The Information Society, 22:64-75, 2006.

Di Scipio, Agostino, *Formalization and Intuition in Analogique A et B (with some remarks on the historical-mathematical sources of Xenakis)*, Definitive Proceedings of the “International Symposium Iannis Xenakis”, Atenas, mayo 2005, pp. 95-108.

Dodge, Charles and Jerse, Thomas, *Computer Music*. Schirmer, 1997.

DuBois, Roger Luke, *Application of Generative String-Substitution Systems in Computer Music*, submitted in partial fulfillment of the requirements for the degree of Doctor of Musical Arts in the Graduate School of Arts and Sciences, Columbia University, 2003.

Emmerson, Simon, *Music, Electronic Media and Culture*, Ashgate, 2000.

Ferneyhough, Brian; Boros, James and Toop, Richard (eds.); Harvey, Jonathan (fwd.); *Brian Ferneyhough – Collected Writings*, harwood academic publishers, Amesterdam, 1995.

Findlen, Paula (ed.), *Athanasius Kircher: The Last Man Who Knew Everything*, Routledge, New York, 2004.

Forte, Allen, *The Structure of Atonal Music*, New Haven and London: Yale University Press, 1973.

Fuller, Mathew (ed.), *Software Studies: a Lexicon*, MIT Press, 2008.

Georgaki, Anastasia, *The Grain of Xenakis’ Technological Thought in The Computer Music Research of Our Days*, in Makis Solomos; Anastasia Georgaki; Giorgos Zervos; (eds.), *Definitive Proceedings of the “International Symposium Iannis Xenakis” Athens, May, 2005*.

Holtzman, S.R., *A Description of an Automatic Digital Sound Synthesis Instrument*, D.A.I. Research Report No. 59. Edinburgh: Department of Artificial Intelligence, 1978.

Hagan L. , Kerry, *Genetic Analysis of Analogique B*, EMS: Electroacoustic Music Studies Network – Montréal, 2005.

Harley, James, *Xenakis: His Life in Music*, Routledge, 2004.

Hoadley, Richard, *Calder's Violin: Real-Time Notation And Performance Through Musically Expressive Algorithms*, Digital Performance Laboratory, Anglia Ruskin University, Cambridge UK, 2012.

Hoffman, Peter, *Towards an "Automated Art": Algorithmic Processes in Xenakis' Compositions*, Contemporary Music Review, Vol. 21, Nos 2/3, pp. 121 – 131, Routledge, 2002.

\_\_\_\_\_, *Music Out of Nothing? A Rigorous Approach to Algorithmic Composition by Iannis Xenakis*, PhD thesis. Berlin, 2009.

Holmes, Thom, *Electronic and Experimental Music: Technology, Music and Culture*, Third Edition, Routledge, 2008.

Koenig, G. M, *The Use of Computer Programs in Creating Music*, In Music and Technology (Proceedings of the Stockholm Meeting organized by UNESCO). Paris: La Revue Musicale, 1971, pp. 93-115.

Lévi-Strauss, Claude, *El pensamiento salvaje*, Fondo de Cultura Económica, México D.F., 2012.

Lévy, Fabien, *Complexité grammatologique et complexité aperceptive en Musique: Étude esthétique et scientifique du décalage entre la pensée de l'écriture et la preception cognitive des processus musicaux sous l'angle des théories de l'information et de la complexité*, Thèse pour obtenir le grade de docteur de l'EHESS en musicologie, Ecole des Hautes Etudes en Sciences Sociales, décembre 2003.

\_\_\_\_\_, *Notre notation musicale est-elle une surdit  au monde?*, en *Ethique et significations, La fidelit  en art et en discours*, Bruylant publisher, series "Au coeur des textes - N  5", Paris, 2005.

\_\_\_\_\_, *When the computer enables freedom from the machine (On an outline of the work H r do-Ribotes)*, The OM Composer's Book. 1, Ircam/Delatour publ., pp. 125 - 131, Paris, 2006.

Licht, Alan, *Sound Art*, Rizzoli, 2007.

Ligeti, Gy rgy, *Pierre Boulez: Decision and Automatism in Structures Ia* , Die Reihe, 4, 1960 pp. 38-39

Louridas, Panagiotis; *Design as Bricolage: Anthropology Meets Design Thinking*, Design Studies Vol 20, No 6, pp 517-535, 1999.

Lloyd, Garret, *Musimathics* Vol. 1. MIT Press, 2006.

\_\_\_\_\_, *Musimathics* Vol. 2. MIT Press, 2008.

Luque, Sergio, *Stochastic Synthesis: Origins and Extensions*, <http://www.sergioluque.com>, 2006.

\_\_\_\_\_, *The Stochastic Synthesis of Iannis Xenakis*, Leonardo Music Journal, Vol.19, pp 77-84, 2009.

Maconie, Robin, *The Works of Karlheinz Stockhausen*, Oxford University Press, London, 1976.

Manovich, Lev, *Automation of Sight: from Photography to Computer Vision*, consultado en la p gina web: <http://www.manovich.net/articles.php>, 1997.

\_\_\_\_\_ , *Avant-garde as Software*, consultado en la página web:  
<http://www.manovich.net/articles.php> , 1999.

\_\_\_\_\_ , *The Language of New Media*, MIT Press, 2001.

\_\_\_\_\_ , *Post-media Aesthetics*, consultado en la página web:  
<http://www.manovich.net/articles.php> , 2001.

\_\_\_\_\_ , *Models of Authorship in New Media*, consultado en la página web:  
<http://www.manovich.net/articles.php> , 2002.

\_\_\_\_\_ , *New Media from Borges to HTML*, Introducción a *The New Media Reader*,  
Wardrip-Fruin, Noah; Montfort, Nick; (eds.); MIT Press, 2003.

\_\_\_\_\_ , *Don't Call it Art: Ars Electronica 2003*, consultado en la página web:  
<http://www.manovich.net/articles.php> , 2003.

\_\_\_\_\_ , *Understanding Hybrid Media*, publicado en Betti-Sue Hertz, ed.; *Animated  
Paintings*, San Diego: San Diego Museum of Art, 2007.

\_\_\_\_\_ , *Software Takes Command*, consultado en la página web:  
<http://www.softwarestudies.com/softbook>, November 20, 2008.

\_\_\_\_\_ , *Media After Software*, consultado en la página web:  
<http://www.manovich.net/articles.php> , 2012.

Manning, Peter, *Electronic and Computer Music*, Oxford University Press, 2004.

Manousakis, Stelios, *Musical L-Systems*, Master's Thesis – Sonology, The Royal Conservatory,  
The Hague, junio, 2006.

McLuhan, Marshall, *Understanding Media: The extensions of man*, MIT Press, 1994.

Messiaen, Olivier, *Technique de mon langage musical*, Alphonse Leduc, Paris, 1944.

Molino, Jean, *Musical fact and the semiology of music*, Music analysis, vol. 9, no. 2 (1990), pp. 105-156. Traducción al español de Juan Carlos Zamora

Montfort, Nick; Baudoin, Patsy; Bell, John; Bogost, Ian; Douglass, Jeremy; Marino, Mark C.; Mateas, Michael; Reas, Casey; Sample, Mark and Vawter, Noah; *10 PRINT CHR\$(205.5+RND(1)); : GOTO 10*, MIT Press, 2013.

Morales Manzanares, Roberto; *Expanding compositional archetypes in a composer assistant tool*, Memoirs of the Fourth International Seminar on Mathematical Music Theory, Huatulco, México, November 22-24, 2010, pp. 63 – 76.

Morgan, Robert, *Twentieth Century Music: a history of musical style in modern Europe and America*, Norton & Company, Inc., 1991.

Nierhaus, Gerhard, *Algorithmic Composition: Paradigms of Automated Music Generation*, Springer, 2009.

Ong, Walter, *Orality and Literacy: The Technologizing of the World*, Routledge, 1982.

Oswalt, Philipp, *Iannis Xenakis' Polytopes*, Contemporary Music Review, 2002, Vol. 21, Nos 2/3, pp. 35-44.

Padisson, Max; Deliège, Irène (eds.), *Contemporary Music: Theoretical and Philosophical Perspectives*, Ashgate Publishing Limited, England, 2010.

Pape, Gerard, *Iannis Xenakis and the "Real" of Musical Composition*, Computer Music Journal, 26:1, pp. 16-21, primavera 2002.

Przemyslaw, Prusinkiewicz and Lindenmayer, Aristid, *The Algorithmic Beauty of Plants*, Springer-Verlag, New York, 1990.

Puckette, Miller, *The Theory and Technique of Electronic Music*, World Scientific Publishing, 2007.

Reas, Casey and Fry, Ben, *Processing: A Programming Handbook for Visual Designers and Artists*, MIT Press, 2007.

Reck Miranda, Eduardo, *Composing Music With Computers*, Focal Press, 2001.

\_\_\_\_\_, *Computer Sound Design: Synthesis Techniques and Programming*, Focal Press, 2002.

Roads, Curtis, *The Computer Music Tutorial*, MIT Press 2000.

\_\_\_\_\_, *Microsound*, MIT Press, 2004.

Rocha Iturbide, Manuel, consultado en la página web:

<http://www.artesonoro.net/artesonoroglobal/ElArteSonoroEnMexico.pdf>

\_\_\_\_\_, consultado en la página web:

<http://www.artesonoro.net/artesonoroglobal/NuevoArteSonoroMexico.html?.pdf>

\_\_\_\_\_, consultado en la página web:

<http://www.artesonoro.net/artesonoroglobal/QueEsElArteSonoro.html>

\_\_\_\_\_, consultado en la página web:

<http://www.artesonoro.net/artesonoroglobal/ElarteSonoroHacia.html>

Shiffman, Daniel, *Learning Processing: A Beginners Guide to Programming Images, Animation and Interaction*, Morgan Kaufmann, 2008.

\_\_\_\_\_ , *The Nature of Code: Simulating Natural Systems With Processing*, versión electrónica <http://natureofcode.com/> , 2012.

Smith, Reginald, *The New Music: The Avant-Garde Since 1945*, Oxford University Press, 2nd Edition, 1987.

Solomos, Makis, *The Unity of Xenaki's Instrumental and Electroacoustic Music: The Case for "Brownian Movements"*, *Perspectives of New Music*, Vol 39. No. 1, Invierno 2001, pp. 244 – 254.

\_\_\_\_\_ , *Iannis Xenakis*, Mercuès, P. O. Editions, 1996, 176p. Nouvelle édition en ligne : 2004 : [www.univ-montp3.fr/~solomos/lintro.html](http://www.univ-montp3.fr/~solomos/lintro.html)

\_\_\_\_\_ , *The granular connection (Xenakis, Vaggione, Di Scipio...)*, Symposium *The Creative and Scientific Legacies of Iannis Xenakis International Symposium*, junio 2006.

\_\_\_\_\_ , *Cellular Automata in Xenakis' Music Theory and Practice*, En Makis Solomos, Anastasia Georgaki, Giorgos Zervos (eds.), *Definitive Proceedings of the International Symposium Iannis Xenakis*, Atenas, mayo 2005.

Sterken, Sven, *Towards a Space-Time Art: Iannis Xenakis's Polytopes*, *Perspectives of New Music*, Vol. 39, No.2 (Summer, 2001), pp. 262-273.

Supper, Martin, *A Few Remarks on Algorithmic Composition*, *Computer Music Journal*, Volume 25, Number 1, Spring 2001, pp. 48-53.



Squibbs, Ron, *Aspects of compositional realization in Xenakis's pre-stochastic and early stochastic music*, Proceedings of the Xenakis International Symposium Southbank Centre, London, 1-3 abril 2011 – <http://www.gold.ac.uk/ccmc/xenakis-international-symposium>

Vargas, Balint András, *Conversations with Xenakis*, faber and faber, London, 1996.

Vázquez, Hebert, *Fundamentos teóricos de la música atonal*, Universidad Nacional Autónoma de México, Fondo Nacional para la Cultura y las Artes, México, D.F., 2006.

Wiggins, Geraint; McLean Alex; *Bricolage Programming in the Creative Arts*, Centre for Cognition, Computation and Culture Department of Computing Goldsmiths, University of London, 2010.

Wishart, Trevor, *Audible Design: a Plain and Easy Introduction to Sound Composition*, Orpheus the Pantomime Ltd, 1994.

\_\_\_\_\_, *On sonic art*, Routledge, 1996.

Wang, Ge, *The Chuck Audio Programming Language: a Strongly Timed and On-The-Fly Environ/Mentality*, a dissertation presented to the Faculty of Princeton University in candidacy for the degree of Doctor of Philosophy, 2008.

Xenakis Iannis, Brown Roberta and Rahn John, *Xenakis on Xenakis*, Perspectives of New Music, Vol.25, No. 1/2, 25th Anniversary Issue (Winter – Summer, 1987), pp.16-63.

Xenakis, Iannis, *Achorripsis*, Bote & Bock, Berlin, 1958.

\_\_\_\_\_, *Analogique A*, Editions Salabert, 22 rue Chauchat, Paris 9º, 1958.

\_\_\_\_\_, *Formalized Music: Thought and Mathematics in Composition*, Pendragon Press, 1992.