



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

Facultad de Artes

Magíster en Artes

RandomHead

Composición Electrónica Aleatoria

Por

Sean Moscoso Samilla

Memoria de obra presentada a la Facultad de Artes

de la Pontificia Universidad Católica de Chile

para optar al grado académico de Magíster en Artes, Mención Música

Profesor Guía: Alejandro Guarello Finlay

Diciembre, 2017

Santiago de Chile

©2017, Sean Moscoso Samilla

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica que acredita al trabajo y a su autor.

Tabla de contenidos

I. Resumen	5
II. Objetivos	6
III. Descripción del proyecto	7
IV. Antecedentes e Ideas Preliminares	8
4.1 Discurso Musical.....	8
4.2 Aleatoriedad	10
4.3 Música Electrónica.....	11
4.3.1 Emancipación del sonido en la música.	11
4.3.2 Música Concreta.....	12
4.3.3 Música Electrónica.....	13
4.3.4 Música por computador	15
V. Proceso de creación de la obra.....	17
5.1 Análisis de la cita	19
5.1.1 Frecuencias.....	19
5.1.2 Duraciones.....	19
5.1.3 Naturaleza musical	20
5.1.4 El uso de citas en una obra.....	20
5.2 Forma	22
5.2.1 Caracterización de los paneles o secciones.....	23
5.3 Programación del software.....	34
5.3.1 Transformación de la cita.....	34
5.3.2 Caracterización de los paneles desde el código	39
VI. A modo de conclusión	44
VII. Bibliografía.....	46
VIII. Anexos.....	48

Profesor Guía: Alejandro Guarello Finlay

El presente documento describe un proyecto de creación musical enfocado en el uso de procedimientos de la tradición compositiva escrita y de la creación contemporánea electrónica. El resultado del proyecto es una obra musical cuya organización formal general es fija, entiéndase esto por el comportamiento de los sonidos y no por la determinación de las alturas y sus duraciones. Cada panel de esta estructura es generado internamente de manera aleatoria, por lo que, respetando el comportamiento musical general de la sección, los detalles específicos son generados al azar en cada una de las versiones posibles de la obra. Este azar es determinado inicialmente al obtener como variable el momento en que es activado el código de la obra en el software-intérprete de la pieza musical.



Página 5

II. Objetivos

General

- Crear música utilizando estrategias compositivas que permitan mantener el pensamiento musical como si este fuese escrito de manera tradicional, pero aplicando además técnicas contemporáneas de creación musical, tanto como lenguaje como recursos electrónicos.

Específicos

- Componer música mediante el uso de una cita musical y sus posibles variaciones
- Establecer un comportamiento estructural (utilizando el recurso de paneles) de los sonidos a través del tiempo
- Producir música aplicando procedimientos compositivos contemporáneos:
aleatoriedad, electrónica, autogeneración.
- Valerse de un recurso tecnológico (Sonic pi) para la composición de una obra musical.

III. Descripción del proyecto

El proyecto *RandomHead* tiene como eje central el desarrollar un trabajo compositivo que integre procedimientos creativos de la tradición musical escrita occidental como operaciones relacionadas a la composición musical contemporánea.

A través de la música por ordenador se ha buscado encontrar un punto de contacto, un espacio de actividad donde ambas prácticas concurren simultáneamente, desarrollando así un lenguaje musical que integre una macro forma, utilizando una cita musical, aleatoriedad y los recursos de la música electrónica.

A partir de 1960, en el mundo de la música contemporánea académica, la forma abierta y la forma móvil fueron las principalmente usadas por la electrónica y la música aleatoria (Scholes, 1981: 605-606). En este proyecto se utilizarán procedimientos como la aleatoriedad y el uso del computador para la generación de la obra, sin embargo, se establecerá una estructura que describa los comportamientos de los sonidos, a la manera de la música escrita tradicional, y estas características permanecerán sin importar el nivel de aleatoriedad exhibida en cada sección. El uso de una cita musical también corresponde a un procedimiento relacionado con la tradición musical escrita.

“Hoy en día, el componer música implica optar conscientemente por algún tipo de comportamiento y algún modo de articulación formal. Ante esta instancia, el compositor actual enfrenta problemas de elegir entre todas las posibilidades conocidas y por conocer” (Guarello, 2006: 28)

De esta forma, se pretende lograr poner en relación ciertas ideas y prácticas musicales que no necesariamente han sido desarrolladas simultáneamente con regularidad. (Becerra-Schmidt. 1958:48-75)

IV. Antecedentes e Ideas Preliminares

4.1 Discurso Musical

“es evidente que el concepto de forma es un concepto derivado del mundo de las realidades visuales y plásticas, y aplicado por analogía en el terreno acústico al mundo de las ideas sonoras, donde, en rigor, no hay formas concretas, en el sentido plástico de este concepto, sino un constante fluir, un movimiento ininterrumpido”

(Riemann, 1929:17)

Cuando se piensa acerca del discurso musical tradicionalmente se asocia a una cuestión formal, para ser preciso, esquemas formales, los cuales han sido consolidados en términos históricos hasta el siglo XIX y más (forma ternaria, binaria, sonata, etc.). Los esquemas formales resumen y sintetizan asuntos relacionados a la distribución en el tiempo de partes o secciones relacionadas, es decir con repeticiones de partes y referencias a secciones iniciales en los últimos momentos de la pieza o movimiento. En el discurso musical encontramos también procedimientos composicionales como la fuga o las variaciones, pero que tradicionalmente también recurren a sistemas de recapitulación o re exposición cumpliendo con la naturaleza tradicional de formas cerradas. Sin embargo, el discurso musical es mucho más que todo lo anterior y se refiere a la

disposición de los sonidos y de los objetos formales en el tiempo estableciendo una relación potencialmente perceptible (debiera ser reconocida por el auditor)

Estos elementos del discurso musical se relacionan temporalmente estableciendo a través de la memoria del oyente fenómenos como el contraste, la mutación, la repetición, la repetición variada, la cita, la referencia, etc.

Desde la aparición de la forma abierta (1960 aprox.) el discurso musical puede asumir un comportamiento análogo a una trayectoria en donde los momentos sonoros van sucediendo sin necesariamente cumplir con esquemas o comportamientos tradicionales, los momentos se suceden, pero no necesariamente deben transitar por secciones obligatorias o ser re expuestos.

“Es natural pensar que cualquier cosa que hagamos presupone algún tipo de orden, pero en el caso del orden temporal esta percepción es esencialmente un orden de cambio y transitoriedad; pues el significado esencial del tiempo es que todo es mutable y pasajero” (Bohm, 1988).

Por lo tanto, deberíamos entender el discurso musical como una sucesión de sonidos musicales, en donde el orden, la combinación y su disposición configuran un discurso. Una sintaxis, una manera acerca de cómo se disponen los sonidos en el tiempo, como se articula el lenguaje; como discurso musical va adquiriendo distintos comportamientos retomando así la idea expuesta en la cita Riemann (1929)

4.2 Aleatoriedad

Para Werner Meyer-Eppler en Eimert (1959) los procesos aleatorios (de *alea* = dado) son procesos cuyo desarrollo general es previsible, pero que en lo particular dependen de la casualidad. Desde la matemática pueden ser estudiados desde la estadística. Musicalmente la aleatoriedad ha sido usada para procesos de alta complejidad (Xenakis), los cuales determinan los macro acontecimientos o su apariencia global, pero no los micro acontecimientos que resultan aleatorios porque ellos no afectan la percepción del resultado.

La composición algorítmica con elementos aleatorios es aquella que consideramos que incluye azar, sin embargo, el concepto de azar es un concepto que usamos “cuando no presentan a nuestra percepción ningún tipo de regularidad ni muestra alguna determinación, y cuando no conocemos la causa que las ha generado” (Supper,2004:92) por lo tanto es un concepto que describe nuestro desconocimiento de las causas de los eventos.

Generalmente los conceptos algoritmo y composición algorítmica se asocian al uso del computador, sin embargo, procesos algorítmicos han sido usados por otros músicos siglos antes de la aparición de la computación, Mozart en 1777 compone *Musikalisches Würfelspiel* (K.294). Un juego de dados mediante el cual se podía componer un vals de forma aleatoria. Mediante el resultado al azar de los dados se seleccionan fragmentos musicales, que, dispuestos de forma secuencial, componen una obra completa. (Sirvent, 2014:18)

De lo que se trata es de la libertad de elección entre un número mayor o menor de posibilidades dadas. Estas pueden ser elegidas por un intérprete o un computador. (Dibelius,2004:268) Incluso es posible distinguir ordenes aleatorios de otros ordenes aleatorios, lo que implica que tienen algún orden implícito. (Bohn, 1988:146)

Para Tomás Marco (2002) toda música es aleatoria en esencia, ya que el compositor no puede controlar todos los elementos que ocurren en la ejecución musical, dado que cada interpretación nunca es exacta a la anterior. Sin embargo, el resultado global siempre es el mismo. Los compositores históricamente pasaron por alto este grado de aleatoriedad dado que estas no hacían peligrar sustancialmente la identidad de la pieza.

A partir de la música aleatoria propiamente tal los compositores comenzaron a entregar diversos grados de libertad al interprete (disposición de las partes, repeticiones, versiones, instrumentación, etc). Podemos ver ejemplos en *Serenata para un satélite* de Bruno Maderna (1969) y en Karlheinz Stockhausen con su *Klavierstück XI* (1956)

4.3 Música Electrónica

4.3.1 Emancipación del sonido en la música.

En la música tradicional los sonidos utilizados fueron en su gran mayoría de frecuencia determinada y constante. Solo las inclusiones de algunos instrumentos de percusión en la orquesta incorporaron sonidos de frecuencia indeterminada (ruido) este último aspecto que fue creciente en la práctica composicional del romanticismo en adelante. Tiene uno de sus momentos más relevantes en el futurismo, un movimiento modernista de vanguardia artística (1909) Que buscaba romper con la tradición y el arte del pasado sin asumir los rasgos de alguna cultura previa, logrando crear un arte musical nuevo, una renovación en los principios y en las técnicas artísticas, que aún tienen consecuencias en la actualidad. Luigi Russolo (1885-1947) artista y compositor, lideró el movimiento que, en lo musical, se caracterizó por concebir la música creada preferentemente con ruidos. Russolo combinó los instrumentos tradicionales con nuevos

instrumentos. Como el *Intonarumori*. Una máquina construida utilizando cajas y dispositivos con el objetivo de producir ruidos y experimentar con los timbres que ya la industrialización ofrecía al mundo acústico. Esta búsqueda por la emancipación del sonido da pie a un conjunto de nuevos instrumentos que exploraran nuevas dimensiones del sonido desde lo eléctrico. Una de estas primeras “máquinas cantoras” fue el *Theremin* creado por el profesor de física Leon Theremin en 1920, que posee dos generadores de alta frecuencia alimentados por electricidad. Uno controla las alturas y el otro, la intensidad del sonido. Para este instrumento Edgar Varèse compuso en 1934 *Equatorial* para dúo de Theremin; dos años después le agrego otros instrumentos. En 1928 Maurice Martenot también aplicando el principio del oscilador usado antes en el Theremin y en el Esferòfono de Mager, crea las Ondas Martenot. En este caso las “Ondas Martenot” contaban con un teclado de piano que servía de referencia respecto al lugar donde se producía cada tono, y además, un sistema que posibilitaba el *glissando* de alturas. Ya entre 1946 y 1948 el compositor Olivier Messiaen incluyó las Ondas Martenot en su sinfonía *Turangalila*.

4.3.2 Música Concreta

Fue en 1948, en París en torno al *Club d'Essai de la Radiotélévision Française* (RTF) que operó como la planta experimental de la radiodifusión de Francia, cuando el compositor Pierre Schaeffer experimentó con las nuevas tecnologías de grabación y reproducción del sonido. En las obras concretas se utilizó la mezcla, la grabación, la reproducción y manipulación de sonidos grabados, para organizar un discurso musical uniendo los sonidos musicales y concretos. Dentro de los ejemplos de sonidos concretos se consideran los cantos de pájaros, sonidos de trenes y otros, más tarde conocidos como objetos sonoros. En el inicio de este periodo los tocadiscos estaban en auge, y la única técnica de manipulación sonora existente consistía en fabricar surcos cerrados para crear ostinatos (loop), invertir

el giro del motor y cambiar su velocidad de reproducción. Schaeffer denominó esta música *musique concrète* por tres motivos: porque el material sonoro en el que trabajó existía de una manera concreta, es decir, fue grabado con un micrófono antes de que se iniciara realmente la realización de cada pieza; porque los sonidos grabados rehúyen su fijación escrita, es decir, porque no resultaría posible escribir una partitura que correspondiera satisfactoriamente a los objetos sonoros utilizados; y , por último, porque el material sonoro existe <de manera concreta> en el soporte magnético o surco mecánico gracias a su grabación microfónica (Supper, 2004:25-26)

Schaeffer distingue entre el cuerpo sonoro (*corps sonore*), el que produce el sonido, y el objeto sonoro (*objet sonore*), es decir, el sonido producido en sí. Una obra de música concreta debe concebirse de tal manera que el oyente no establezca ninguna relación con la fuente que generó el sonido. La audición de música concreta a través de altoparlante fue denominada por Schaeffer como *acousmatique*. “no existe ningún instrumento musical con el que se pueda interpretar música concreta. Esa es la principal dificultad. Sin embargo, es posible imaginar una maquina gigantesca de tipo cibernético, capaz de producir millones de combinaciones. Pero todavía no hemos llegado a tanto” (Schaeffer en Prieberg, 1961:95)

4.3.3 Música Electrónica

Su génesis se remonta a 1951, en la Radio de Colonia (NWDR, luego WDR en 1955), donde Hebert Eimert, director artístico del estudio, crea el primer estudio de música electrónica junto a sus colaboradores Fritz Enkel, ingeniero, Robert Bayer, compositor y Werner Meyer-Eppler, docente de la catedra de fonética y telecomunicación en la Universidad de Bonn. Si en la Música concreta se opera sobre el sonido pregrabado, postproducción podríamos llamarle, la electrónica

trabaja creando el sonido propiamente tal, utilizando a este como materia prima. En palabras de Eimert “Con la música electrónica es posible sin duda crear, como por arte de magia, los más interesantes montajes de sonidos”. (Eimert,1959:13) El tono sinusoidal se establece como el punto de partida, creado sintéticamente mediante osciladores eléctricos y electromecánicos. El sonido es “construido” por la suma de varios tonos sinusoidales o el filtrado de ruido blanco, el cual contiene la totalidad de las frecuencias auditivas, entre otros procedimientos de síntesis de sonido. Por lo tanto, los principales recursos para una composición electrónica eran generadores de ondas, filtros, transposiciones al modificar la velocidad, estereofonía, reverberación, entre otros. (Supper, 2004:31)

Las primeras composiciones puramente electrónicas fueron realizadas por Hebert Eimer y Robert Bayer en torno a 1952, a quienes se sumó el compositor Karlheinz Stockhausen al incorporarse al estudio de Colonia. En 1953, en el marco de la *Neuen Musikfest* (festival de la nueva música) se presentaron obras electrónicas en los estudios de la Radio de Colonia. En 1956, replicando la experiencia alemana, la Asociación de Radiodifusión Italiana (RAI) instaló su propio estudio para música electrónica en Milán, de donde se destacan los primeros trabajos de Luciano Berio y Bruno Maderna. También Japón construyó un estudio para música electrónica en la Radiodifusora de Tokyo (NHK) donde su jefe a cargo fue Toshiro Mayuzumi, quien ya había compuesto música concreta e incursionaba con la música electrónica hacia 1955. Cabe destacar que el posterior desarrollo y alcance de la tecnología aplicada al sonido y a la música, permitió incluso, adentrarse “microscópicamente” dentro del espectro sonoro generando las bases de la música espectral (Gérard Grisey, Tristan Murail y Hughes Dufourt) que propone a la música instrumental tradicional nuevos mundos sonoros.

Resulta relevante aclarar que el termino música electrónica actualmente es utilizado indistintamente para referirse a la música electroacústica, electrónica, acusmática, concreta llegando a un punto tan genérico que hoy incluye la electrónica en vivo, la *Tape Music*, música por ordenador e incluso la música bailable electrónica, *Rave*. (Supper,2004:169)

4.3.4 Música por computador

En general el término “música por computador” es usado genéricamente para referirse a dos tipos de uso del ordenador para crear música. Uno de ellos es realizar cálculos de un sonido electroacústico (timbre y síntesis de sonido) y el otro, para la producción de una “partitura”. *“Los procesos de realización de la música con ordenador pueden denominarse <composición algorítmica> y las obras incluidas en esta categoría, por tanto <composiciones algorítmicas>”*.(Supper,2004:34).

El computador, entendido como los circuitos que tiene un ordenador en su interior (Hardware) la cual ejecuta de manera rápida una secuencia de instrucciones (programas) y guarda y recupera información (datos). Para que este hardware funcione correctamente es necesario ejecutar un programa (software) base, un sistema operativo, además este sirve para que el usuario cree u opere otros programas con funciones específicas. (Cadiz,2008;73), para que el computador pueda leer la información sonora es necesario digitalizar el audio:

“Una señal continua o analógica es una que posee valores en cada instante de tiempo. En contraste, una señal discreta o digital solo posee valores en ciertos instantes de tiempo, distanciados a intervalos regulares. Una señal discreta no está definida para valores intermedios de tiempo. Una señal continua es convertida en una señal digital mediante un proceso denominado

digitalización, el cual a su vez es un proceso que depende de dos operaciones: muestreo (sampling en inglés), que corresponde a tomar muestras de la señal en el tiempo y cuantización, que consiste en aproximar valores continuos de la amplitud a un grupo de valores manejables en el computador.”
(Cadiz,2008:63)

Dentro de estos programas específicos para realizar música en un computador podemos distinguir los siguientes grupos: los programas para composición y notación musical (Finale, Sibelius, MuseScore, etc.), los programas para hacer síntesis directa (Csound, SuperCollider,etc.), los programas para hacer grabación y modificación de audio digital (ProTools, Reaper,etc) y los programas para controlar audio digital en tiempo real (Max/msp, pure data, Sonic-pi) (Cadiz,2008;82), Para este trabajo he utilizado un software de este último grupo: Sonic-Pi, el cual es multiplataforma y de licencia libre.

V. Proceso de creación de la obra

Mi proyecto de ingreso al Magister estaba centrado en el trabajo multidisciplinario, para ello mi investigación anterior al programa, había sido principalmente dedicada al auto aprendizaje de programación, y mi anteproyecto de tesis proponía desde un comienzo el trabajo con lenguaje de programación.

Durante mi primer año del programa académico trabajé problemáticas sobre probabilidades (estocástica) y combinaciones aleatorias. Con el fin de abordar estos aspectos utilicé el software libre Processing¹ el cual está diseñado principalmente para desarrollos visuales. Sin embargo, mis obras de primer año fueron únicamente sonoras, pese a ser programadas y realizadas con ayuda de Processing.

Mientras estaba terminando mi segunda obra, mi investigación evolucionó centrándose sólo en las características del comportamiento sonoro y su control en el tiempo. A partir de este cambio el software Processing me presentaba muchas limitaciones técnicas, fundamentalmente en su velocidad de procesamiento debido a sus características específicas diseñadas principalmente para aspectos visuales. Rápidamente busque otras opciones para trabajar, la herramienta (software) que antes ofrecía soluciones ahora se había convertido en un problema más. Resultó, así como entre varios softwares (pure data, processing, tidal, max/msp) que fui evaluando e investigando fue que llegué al utilizado en esta obra. El software Sonic Pi, el cual utiliza el lenguaje *Ruby*, funciona en Windows, MacOS, Raspbian y Linux.

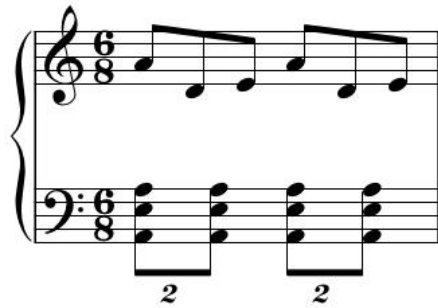
¹ Lenguaje de programación y entorno de desarrollo diseñado para la creación de proyectos multimedia e interactivos. Creado por Ben Fry y Casey Reas el 2001.

RandomHead está concebida a partir de una cita musical como elemento generador de relaciones interválicas, estéticas y procedimentales. Esta cita corresponde al primer compás de la parte de piano de la canción *Daydreaming*² de la banda inglesa *Radiohead*. La elección de esta cita gira entorno a la casualidad ya que mientras buscaba ideas sonoras e investigaba diversos materiales para la realización de esta obra, mi trabajo se centró en reflexiones relacionadas con cuestiones de mi identidad y pertenencia buscando un modo de hacer música en la cual me sintiera directamente involucrado. Por esos días uno de mis ex alumnos me envió esta canción de *Radiohead* junto a sus saludos y aclarándome que esta música le recordaba todo lo que yo le había enseñado en clases, cuestión que resultó emocionalmente importante para mí. En esta canción pude reconocer muchas de mis propias prácticas musicales y sin duda la carrera artística de esta banda experimental inglesa ha resultado relevante para mis propias experimentaciones. Así el uso de esta música opera a nivel personal, como una declaración de principios estéticos. Esta vinculación voluntaria hacia esta banda funciona como una afinidad ideológica, un árbol genealógico al cual personalmente me siento adscrito, este texto que habla de otro texto, esta música que habla de otra música. Todo lo anterior quedó expresado en el uso del primer compás del acompañamiento de piano de esta canción.

² Segundo sencillo del Álbum *A Moon shaped Pool*, publicado en mayo del 2016. Video Clip de la canción en <https://www.youtube.com/watch?v=TTAU7ILDZYU>

5.1 Análisis de la cita

5.1.1 Frecuencias



En un primer nivel la melodía presenta los siguientes intervalos: quinta justa descendente (la-re) y segunda mayor (re-mi) y al repetir el giro (*porrectus*) una cuarta justa (mi-la). En un segundo nivel el

acorde se compone de la superposición de intervalos de quinta justa (la-mi) y cuarta justa (mi-la`) y de octava (la-la`). Las alturas aparecen en un registro medio del piano.

De los intervalos que componen la cita (segunda mayor; cuarta justa; quinta justa y octava) se extrajeron los valores numéricos son utilizados en la construcción de la obra, ellos son: 2-4-5-8

En este motivo es posible reconocer la reiteración del intervalo de cuarta, por lo que el material adquiere características plagales en sus relaciones interválicas.

5.1.2 Duraciones

La cita consta de una superposición de tres corcheas regulares y dos corcheas irregulares sobre un pulso de negra con punto a velocidad de 46 bpm, éstas se repiten hasta completar el compás binario de subdivisión ternaria de 6/8. La melodía presenta una sucesión constante de corcheas en ausencia de silencios, mientras el acorde mantiene una constante en dosillos. Este ritmo tres contra dos, melodía y armonía, coinciden en el primer y cuarto tiempo del compás.

Las relaciones de números obtenidos al analizar el compás en su totalidad son: 2 (dosillo); 3(tresillos); 4 (dos dosillos) y 6 (dos tresillos)

5.1.3 Naturaleza musical

El material musical de la cita cumple una función secundaria, es un simple acompañamiento de piano, sobre el cual luego se sobreponen melodías con sus respectivos textos y diversos efectos sonoros y que fácilmente pasa a ser desapercibida debido a su naturaleza estática de *ostinato* rítmico.

5.1.4 El uso de citas en una obra

El citar una obra o parte de ella es un mecanismo que permite introducir en la propia creación material que le pertenece al propio autor o a otro, material que puede o no poseer una carga asignada en la cultura y es usada por el autor debido que para él si tiene un significado cultural. En el caso de este trabajo, la citación responde a una necesidad comunicativa y de intertextualidad³: la de vincularse con otro autor ideológicamente. (Campaña, 2008)⁴

Una de las acepciones de citar, en latín *citāre*, es “poner en movimiento” y es la que yo utilice preferentemente para esta obra. La inclusión del material de la banda Radiohead significa poner en movimiento la influencia cultural de la banda y su propia música en el uso de sus relaciones rítmico-interválicas como números en movimiento en constante actualización y combinación. La cita es un punto de

³ La **intertextualidad** es la relación que un texto (oral o escrito) mantiene con otros textos (orales o escritos), ya sean contemporáneos o anteriores; el conjunto de textos con los que se vincula explícita o implícitamente un texto constituye un tipo especial de contexto, que influye tanto en la producción como en la comprensión del discurso.

⁴ Campaña, Claudia. El arte de la cita. Velázquez en la obra de Bru y Cienfuegos. Santiago: Quebecor World, 2008.

partida, una toma de posición ética y estética. La problemática abordada en esta obra trata más del comportamiento del material sonoro que de su naturaleza formal.

La cita en esta obra se aplica y opera además en distintos niveles, aprovechando otras acepciones de la palabra citar⁵:

- **Excitar:** *Poner a uno en movimiento para (ex) salir de su estado, cuestión que pretende funcionar en lo personal como artista, pero también a quien escucha esta obra en sus potenciales versiones.*

Interpretación en la obra: *Posibilitar que lo que está adentro salga en forma de música.*

- **Incitar:** *Estimular a alguien que entre (in-) en algo, además de generar mediante la música la posibilidad de conectar la experiencia sonora de los posibles auditores con mi particular mundo estético.*

Interpretación en la obra :*Quienes estén afuera puedan ser invitados a entrar en otro mediante la música*

- **Solicitar:** *Estimular a alguien para estar a solas (sollus citare = citar a solas).*

Interpretación en la obra: *Es por esto por lo que la obra opera en un ámbito íntimo.*

- **Resucitar:** *volver a la vida, re- (reiteración), su- (abajo) y citare.*

Interpretación en la obra *Volver a la vida, reencontrarse con una cuestión vital que es la creación.*

⁵ Diccionario Etimológico Español en línea <http://etimologias.dechile.net/?citar> [revisado 1/5/2017]

5.2 Forma

“En primer lugar, debemos aceptar que todo aquello que tiene un límite. es decir que es finito, necesariamente poseerá una forma determinada por sus propios límites, lo cual permite separar el fenómeno de su contexto. Si analizamos someramente nuestro límite corporal nos encontraremos con que, aparentemente, todos los seres humanos poseemos la misma forma. Pero, inmediatamente surgen diferencias.” (Guarello. 2006:28)

Pese a ser una obra altamente aleatoria e indeterminada en sus detalles específicos, esta ha sido concebida de manera análoga a la de una obra tradicionalmente escrita. Esto se logra a través de la "fijación" de una macro forma obtenida mediante la caracterización de cada una de sus secciones (paneles).

Todas las versiones que resulten al gatillar el proceso de generación de la obra seguirán un mismo orden o secuencia de secciones (momentos homogéneos o “lógicos”) donde se manifestarán, en términos generales, los mismos comportamientos asociados a ellas, con posibles variaciones respecto de su duración (contracción y dilatación temporal) e “ilimitadas” variaciones paramétricas.

Esta macro forma es entendida como el orden establecido de secciones (de uno a diez) con ciertas características generales que siempre permanecen a pesar del nivel de aleatoriedad que experimenten sus micro elementos que operan en la generación de cada sección.

5.2.1 Caracterización de los paneles o secciones

Panel 1

Descripción

Alturas: Sonidos agudos – medios – graves del piano

Duración: Sonidos Breves y semi breves sin reverberación

Intensidad: uso de sonidos de 0% hasta 100% de intensidad

Paneo: uso de sonidos en toda la gama entre canal izquierdo y canal derecho

Timbre: Sonido de piano (eléctrico)

Direccionalidad:

La música recorre desde una sub información de la cita musical, debido a que esta aparece parcialmente, hasta una manifestación completa de ella. El ritmo 3:2 de la cita surge inicialmente sin las alturas originales de la cita, para gradualmente ser recuperadas hacia el final del panel.

Panel 2

Descripción

Alturas: Sonidos medios del piano

Duración: Sonidos semi breves sin reverberación

Intensidad: uso de sonidos de 0% hasta 100% de intensidad

Paneo: uso de sonidos en toda la gama entre canal izquierdo y canal derecho

Timbre: Sonido de piano (eléctrico)

Direccionalidad:

Panel donde se reitera constantemente el ritmo 3:2 de la cita original y variantes percibiéndose como un *ostinato* rítmico, que aumenta su densidad sonora respecto a la exhibida en el panel 1. Las alturas van desde las originales de la cita hacia transposiciones en diferentes campos dentro del rango medio,

Panel 3

Descripción

Alturas: Sonidos Medios del piano

Duración: Sonidos semi breves con y sin reverberación

Intensidad: uso de sonidos de 0% hasta 200% de intensidad

Paneo: uso de sonidos en toda la gama entre canal izquierdo y canal derecho

Timbre: Sonido de piano (eléctrico)

Direccionalidad:

Densidad creciente con predominio gradual de los ritmos irregulares que poco a poco van cubriendo el ritmo original 3:2 haciéndolo progresivamente irreconocible pese a que sus alturas aun permiten identificarla.

Además, en el transcurso del panel van apareciendo notas aisladas con reverberación cambiante distintas a las de la cita. En esta sección aumenta la densidad sonora debido a la superposición creciente de diversas alturas, duraciones y reverberaciones.

Panel 4

Descripción

Alturas: Sonidos medios del piano

Duración: Sonidos semi largos y largos con reverberación

Intensidad: uso de sonidos de 0% hasta 100% de intensidad

Paneo: uso de sonidos en toda la gama entre canal izquierdo y canal derecho

Timbre: Sonido de piano (eléctrico)

Direccionalidad:

Panel donde la densidad es decreciente, se inicia gradualmente la rarificación de los sonidos los cuales tienen un predominio de larga reverberación. El ritmo 3:2 de la cita comienza progresivamente a dispersarse, distanciándose los ataques unos de otros, sin embargo, las alturas muy cercanas a las notas de la cita aun permiten reconocerla.

Panel 5

Descripción

Alturas: Sonidos Agudos – Medios – Graves del piano

Duración: Sonidos semi breves, semi largos y largos con y sin reverberación

Intensidad: uso de sonidos de 0% hasta 100% de intensidad

Paneo: uso de sonidos en toda la gama entre canal izquierdo y canal derecho

Timbre: Sonido de piano (eléctrico)

Direccionalidad:

Los sonidos van desde el uso posible de todo el registro del piano hacia una escasez de sonidos medios para dejar paso a sonidos extremos (graves y agudos) hacia el final del panel. Sonidos de breve duración y reverberación variable contribuyen a aumentar la densidad sonora respecto de la sección anterior (panel 4)

Panel 6

Descripción

Alturas: Sonidos Medios Graves y Graves del piano

Duración: Sonidos semi largos y largos con distintas reverberaciones.

Intensidad: uso de sonidos de 0% hasta 100% de intensidad

Paneo: uso de sonidos en toda la gama entre canal izquierdo y canal derecho

Timbre: Sonido de piano (eléctrico)

Direccionalidad:

En este panel los sonidos comienzan a distanciarse temporalmente, esto abre la posibilidad de silencios por primera vez en la obra debido a la esta rarefacción de los eventos sonoros.

Panel 7

Descripción

Alturas: Sonidos Agudos - Medios - Graves del piano

Duración: Sonidos semi largos y largos con distintas reverberaciones.

Intensidad: uso de sonidos de 0% hasta 100% de intensidad

Paneo: uso de sonidos en toda la gama entre canal izquierdo y canal derecho

Timbre: Sonido de piano (eléctrico), metálicos (piano preparado) y muteados y semi muteados

Envolvente: utilización tanto de sonidos con ataques alargados como sonidos percusivos de ataque rápido

Direccionalidad:

En este Panel la densidad va en aumento por superposición de los sonidos de distintas reverberaciones, similar al panel 5. Los sonidos pasan de una envolvente tradicional de piano (usado en todos los paneles anteriores) a sonidos cuyos ataques son alargados y menos percusivos. En esta sección los timbres pasan del piano tradicional, usado en todos los paneles anteriores, a sonidos metálicos evocando a un piano preparado y a sonidos similares al piano con sordina.

Panel 8

Descripción

Alturas: Sonidos Agudos – Medio Agudos del piano

Duración: Sonidos breves y semi breves con distintas reverberaciones.

Intensidad: uso de sonidos de 0% hasta 100% de intensidad

Paneo: uso de sonidos en toda la gama entre canal izquierdo y canal derecho

Timbre: Sonido de piano (eléctrico), metálicos (piano preparado) y muteados y semi muteados

Envolvente: utilización tanto de sonidos con ataques alargados como sonidos percusivos de ataque rápido

Direccionalidad:

El panel se inicia con una disminución en la densidad debido a la separación temporal de los sonidos, pero en menor medida que el panel 6, cambiando gradualmente hacia un aumento en la densidad debido a la mayor ocurrencia de sonidos. El timbre inicial de piano tradicional se modifica en el tiempo hacia el uso solo de sonidos metálicos y muteados de mayor intensidad.

Panel 9

Descripción

Alturas: Sonidos Agudos – Medios - Graves del piano

Duración: Sonidos semi breves, breves y muy breves con distintas reverberaciones.

Intensidad: uso de sonidos de 0% hasta 200% de intensidad

Paneo: uso de sonidos en toda la gama entre canal izquierdo y canal derecho

Timbre: Sonido de piano (eléctrico), metálicos (piano preparado) y muteados y semi muteados

Envolvente: utilización tanto de sonidos con ataques alargados como sonidos percusivos de ataque rápido a los que se suman otros con envolvente invertida

Direccionalidad:

Panel que se inicia con alta densidad de sonidos con todos los elementos sonoros usados anteriormente (alturas, timbres, envolventes, etc.) y pasa gradualmente hacia la mitad de la sección a una rarificación que posibilita la aparición silencios y que, hacia el final, inicia un progresivo avance hacia una densidad creciente.

.

Panel 10

Descripción

Alturas: Sonidos Agudos – Medios - Graves del piano

Duración: Sonidos breves y muy breves con distintas reverberaciones.

Intensidad: uso de sonidos de 0% hasta 300% de intensidad

Timbre: Sonido de piano (eléctrico), muteados y semi muteados

Envolvente: utilización tanto de sonidos con ataques alargados como sonidos percusivos de ataque rápido

Direccionalidad:

Panel que se inicia con un predominio de sonidos graves y asordinados sumando progresivamente sonidos agudos para luego poco a poco pasar a sonidos sin sordina y muy agudos, los cuales alcanzan la mayor densidad de la pieza con la superposición de todos los registros del piano, pero con predominio de los agudos. El final es sorpresivo suspendiendo toda la actividad quedando solo la reverberación final de los sonidos

Este azar es determinado inicialmente al obtener como variable el momento en que es activado el código de la obra en el software-intérprete de la pieza musical.

Muchas de las variables en la obra son aleatorias y su generación depende del momento en que es activado el código en el software, sin embargo, también pueden recuperarse versiones antes realizadas. El código de la obra permite conocer cual versión está siendo interpretada, es así como también puede ejecutarse una versión en específica.

Para ejemplificar y comparar temporalmente he tomado tres versiones cuyas semillas aleatorias han sido elegidas arbitrariamente: estas son la versión 0 – 2017 y 50000.

Semilla 0 duración total 12:44

Panel 1 (inicio) 0:00 hasta 0:45

Panel 2 0:45 hasta 2:22

Panel 3 2:22 hasta 3:16

Panel 4 3:16 hasta 5:11

Panel 5 5:11 hasta 6:00

Panel 6 6:00 hasta 7:08

Panel 7 7:08 hasta 7:48

Panel 8 7:48 hasta 9:11

Panel 9 9:11 hasta 11:50

Panel 10 11:50 hasta 12:44 (final)

Semilla 2017 duración total 12:24

Panel 1 (inicio) 0:00 hasta 0:47

Panel 2 0:47 hasta 2:21

Panel 3 2:21 hasta 3:11

Panel 4 3:11 hasta 5:13

Panel 5 5:13 hasta 5:58

Panel 6 5:58 hasta 7:05

Panel 7 7:05 hasta 7:42

Panel 8 7:42 hasta 8:48

Panel 9 8:48 hasta 11:29

Panel 10 11:29 hasta 12:24 (final)

Semilla 500.000 duración total 12:45

Panel 1 (inicio) 0:00 hasta 0:48

Panel 2 0:48 hasta 2:24

Panel 3 2:24 hasta 3:13

Panel 4 3:13 hasta 5:11

Panel 5 5:11 hasta 6:03

Panel 6 6:03 hasta 7:07

Panel 7 7:07 hasta 7:44

Panel 8 7:44 hasta 9:08

Panel 9 9:08 hasta 11:49

Panel 10 11:49 hasta 12:45 (final)

%	versión 0	versión 2017	versión 500000
panel 1	5,87	6,31	6,27
panel 2	10,73	12,63	12,54
panel 3	7,06	6,72	6,4
panel 4	15,05	16,39	15,42
panel 5	6,41	6,04	6,79
panel 6	8,9	9	8,36
panel 7	5,23	4,97	4,83
panel 8	10,86	8,87	10,98
panel 9	20,81	21,63	21,04
panel 10	7,06	7,39	7,32

Cuadro comparativo de duración de los paneles expresado en el porcentaje de la duración total de la obra

5.3 Programación del software

5.3.1 Transformación de la cita

“Esta 'manipulación' se efectuaba fundamentalmente a través de la variación ornamental o de elaboración o por la reelaboración de la información. La necesidad de manipulación de la información sonora obedece a una necesidad de la percepción humana” (Guarello,2006.:13)

A continuación, explicaré los rudimentos de cómo, mediante el código de programación (*Ruby*⁶) para el software Sonic.pi, se utilizaron para la creación de la obra *RandomHead*.

- **Tocar una nota**

Para tocar una nota basta con escribir `play` seguido de un número, por ejemplo 60, y el software reproduce un do central.

<code>play 60</code>	toca la 60ª tecla del piano, o sea el do central
por lo tanto	
<code>play 61</code>	es do sostenido
y	
<code>play 59</code>	es si natural

⁶ Ruby es un lenguaje de programación creado por el programador japonés Yukihiro "Matz" Matsumoto en 1995.

Octava	Note Numbers											
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
-1	0	1	2	3	4	5	6	7	8	9	10	11
0	12	13	14	15	16	17	18	19	20	21	22	23
1	24	25	26	27	28	29	30	31	32	33	34	35
2	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59
4	60	61	62	63	64	65	66	67	68	69	70	71
5	72	73	74	75	76	77	78	79	80	81	82	83
6	84	85	86	87	88	89	90	91	92	93	94	95
7	96	97	98	99	100	101	102	103	104	105	106	107
8	108	109	110	111	112	113	114	115	116	117	118	119
9	120	121	122	123	124	125	126	127				

- **Determinar su duración**

Para elegir cuanto tiempo habrá entre cada sonido es necesario escribir `sleep` seguido de un número, ese número determina cuantos pulsos debe esperar antes de seguir con el resto del código.

Si la velocidad de la obra es de 60 bpm

```
play 60
sleep 1
play 64
sleep 1
```

`sleep 1` significa pasar un pulso entre el sonido do(60) y el mi(64), de esta manera el sonido do es una negra y también lo es el mi.

Por lo tanto:

`sleep 0,5` es para una corchea y `sleep 2` para una blanca

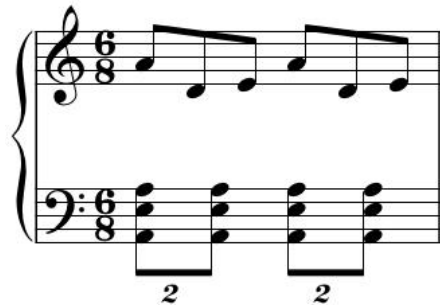
Sí quisiera tocar un acorde solo basta con no colocar `sleep` entre los `play` para que todas las alturas se ejecuten simultáneamente

```
play 60
```

```
play 64  
play 67
```

al darle al ejecutar este código sonara un acorde de do mayor.

- **Escribir la cita**



Primero escribiremos la mano derecha. Rítmicamente son tres corcheas en un pulso, por lo tanto, el código para definir la duración será `sleep 0.33` para que cada corchea valga un tercio de la duración del pulso. Y las alturas serán la (69) – re (62) y mi (64)

```
play 69  
sleep 0.33  
play 62  
sleep 0.33  
play 64  
sleep 0.33
```

Ahora escribiremos la mano izquierda. Rítmicamente son dos corcheas en un pulso, por lo tanto, el código para definir la duración será `sleep 0.5` para que cada corchea valdrá la mitad de la duración del pulso. Y la alturas serán la superposición de las alturas la (45) , mi (52) y la (57)

```
play 45  
play 52
```

```
play 57
sleep 0.5
play 45
play 52
play 57
sleep 0.5
```

sí escribimos estos dos códigos resultará que tocará primero la mano derecha y a continuación la mano izquierda, necesitamos que se ejecuten simultáneamente.

Para esto necesitamos usaremos `in_thread` que nos permite superponer secciones de código todo lo que este escrito entre `in_thread do` y `end` será ejecutado al mismo tiempo que el resto del código.

```
in_thread do
  play 69
  sleep 0.3
  play 62
  sleep 0.3
  play 64
  sleep 0.3
end
play 45
play 52
play 57
sleep 0.5
play 45
play 52
play 57
sleep 0.5
```

aquí tendríamos ambas manos tocando simultáneamente, por lo tanto, ya tenemos en código la cita.

A continuación, explicare un par de manera de cómo puede ser variadas y manipulada la cita desde el código

- **Transportación**

Ya que las alturas están expresadas en números, solo basta con que a todos los números sume o reste una cantidad determinada para transportar todo el motivo completo o parte de él. Por ejemplo:

Si sumo a cada número siete números estaría transportando una quinta justa ascendente. Así cuatro números menos seria transportar una tercera mayor descendente o si sumo doce números estaría subiendo toda una octava.

- **Omitir partes probabilísticamente**

También es posible que desee que una línea del código suene solo dentro de una probabilidad. Para esto solo debo agregar la línea de código `if one_in(2)`

Para que la línea de código tenga una probabilidad de ser ejecutada entres dos, o sea un 50%. Ejemplo

```
play 60 if one_in(2)
```

Esta nota do tiene un 50% de probabilidades de sonar.

```
play 62 if one_in(10)
```

Esta nota re tiene solo un 10% de probabilidades de sonar.

Por lo tanto, de esta forma la cita puede ser ejecutada parcialmente y cada vez que sea leída por el código sonara de distintas maneras sub informando a medida que se cumpla o no la probabilidad de cada línea del código que haya planteado de esta forma.

5.3.2 Caracterización de los paneles desde el código

A continuación explicare los rudimentos del código de programación usado en *RandomHead* de manera de clarificar el proceso de caracterización de los paneles.

- **Aleatoriedad**

Lo primero es aclarar que en el software Sonic Pi no existe la aleatoriedad real, en realidad es una pseudo-aleatoriedad, cuando pedimos un numero al azar Sonic Pi entrega un número “aleatorio” de una secuencia determinada con anterioridad, es decir, entregará una secuencia de manera repetitiva, lo cual es muy útil para asegurar que la música creada al azar suene siempre idéntica en cualquier otra computadora en que se aplique el código. Para obtener una verdadera aleatoriedad es necesario cambiar la “semilla” generadora de esta secuencia pseudo aleatoria. La solución para esta composición fue asociar el reloj interno del computador a una semilla en específico, por lo tanto, cada vez que se ejecute el código esta tomara en cuenta el milisegundo en que fue ejecutado y seleccionará una secuencia aleatoria para cada versión. Lo que nos permite garantizar la aleatoriedad, pero también nos permite saber que versión está ocurriendo y la posibilidad de ejecutar la misma versión solo pidiéndolo en el código de la obra para volver a escuchar la misma versión, garantizando la existencia de ambas posibilidades.

Dentro del código hay varias expresiones que pueden usarse para conseguir resultados aleatorios a continuación algunos usados en la obra:

Si deseo conseguir un numero entre 0 y 1 y sus decimales, usado por ejemplo para expresar porcentajes, puede usarse `rand`

```
rand 0.75006  
rand 0.13476  
rand 0.59802
```

sí quiero obtener un con decimales al azar entre un mínimo y un máximo, puede usarse `rrand(minimo,maximo)`

```
rrand(50,100) 70.92895  
rrand(50,100) 58.54187  
rrand(50,100) 79.79125
```

sí quiero obtener un numero entero al azar entre un mínimo y un máximo, puede usarse `rrand_i(minimo,maximo)`

```
rrand_i(20,70) 64  
rrand_i(20,70) 38  
rrand_i(20,70) 52
```

- **Paneo**

Si quiero que un sonido suene solo por alguno de los parlantes en stereo, la expresión Pan. Siendo -1 todo (100%) a la izquierda, 0 ambos y 1 todo (100%) a la derecha. Ejemplo:

```
play 60, pan: -1      nota do 100% izquierda  
play 60, pan: 1       nota do 100% derecha  
play 60, rrand(-1,1)  nota do 0.5298 (75% derecha, 25% izquierda) aleatorio
```


- **Volumen**

Si quiero modificar el volumen de un sonido debe usarse la expresión `amp`. Siendo `amp: 1` el 100% del volumen, `amp: 0` no sonido y `amp: 2` el 200% del volumen. Ejemplo

<code>play 60, amp: 0.5</code>	nota do al 50% de intensidad
<code>play 60, amp: 0,1</code>	nota do al 10% de intensidad
<code>play 60, amp: rand</code>	nota do al 6% de intensidad - aleatorio

- **Alturas**

Si quiero modificar aleatoriamente las alturas musicales, pero quiero mantenerlas dentro de un registro (agudo, medio o grave). Una forma es la siguiente.

<code>play rrand_i(12,45)</code>	38	aleatorio
----------------------------------	----	-----------

de esta forma puedo mantener el rango aleatorio entre do 0 y la 2

<code>play rrand_i(84,108)</code>	94	aleatorio
-----------------------------------	----	-----------

de esta forma puedo mantener el rango aleatorio entre do 6 y do 8

<code>play rrand_i(0,127)</code>	64	aleatorio
----------------------------------	----	-----------

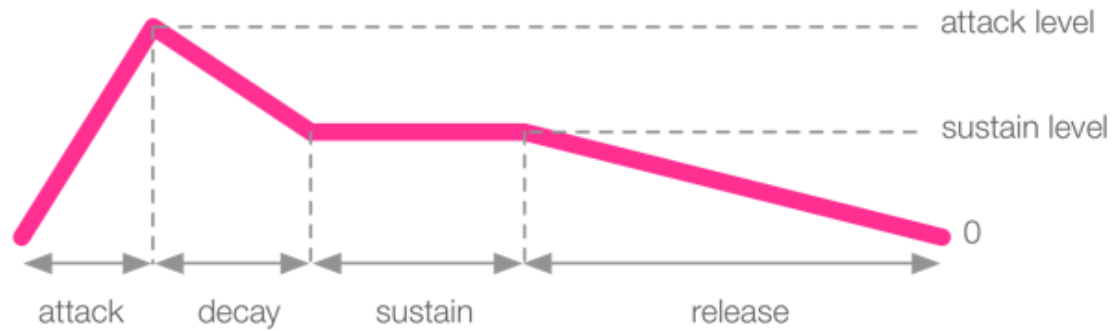
de esta forma puedo elegir el rango completo del piano

Dureza del toque de la tecla (Hardness of keypress)

Si quiero simular la fuerza con la que se ejecuta las teclas `hard: .` proporciona control de esta variable.

<code>play 60, hard 0.5</code>	sonido “normal”
<code>play 60, hard 1</code>	sonido metálico
<code>play 60, hard 0</code>	sonido ensordinado

- **Envolventes**



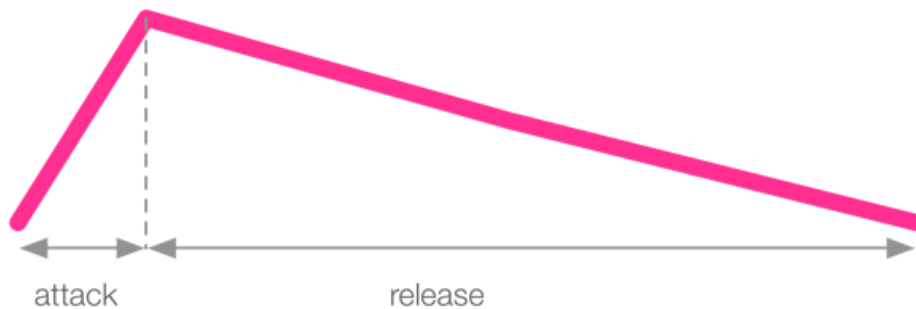
cada una de estas variables puede ser determinada con los siguientes comandos:

```
attack: 0.5
decay: 0.5
sustain: 0.5
release: 0.5
```

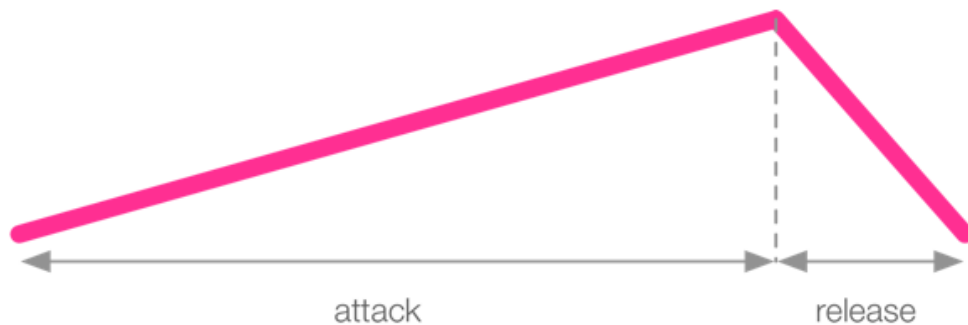
habrá que tener atención que la duración de este sonido será la suma de todas estas variables en la envolvente, para el ejemplo anterior este durara 2 segundos en bpm 60

ejemplos de otros algunas envolventes posibles:

```
play 60, attack: 0.7, release: 4
```



play 60, attack: 4, release: 0.7



play 60, attack: 0.5, release: 0.5



VI. A modo de conclusión

Para finalizar este texto explico ciertas reflexiones que fueron tomando forma en el proceso de creación y posterior análisis de la obra. Material que guio el proceso, pero no encontró un lugar fijo en la memoria de obra para ser exhibida.

Una analogía que ayuda a entender con mayor facilidad el proceso y la creación de esta obra es pensar en la relación que tiene el genotipo del ser humano con sus rasgos particulares. El algoritmo de esta obra es fijo, pero desde él se generan distintos resultados conservando las características específicas de cada panel, por lo tanto, manteniendo el discurso musical general. Pues de esta misma manera un genotipo humano generará siempre una estructura humana (dos manos, dos pies, cabeza, ojos, etc.) sin embargo, este humano puede ser más alto o más bajo, más gordo o más flaco, de pelo rizado o pelo liso y diversos rasgos faciales, color de piel, etc. Estos cambios no varían nuestra percepción de que estamos frente a un ser humano, por diverso que sea en sus características particulares (fenotipo). Por lo tanto, nuestro genotipo sería la programación, la cual siempre al ser ejecutada generará “la pieza” en cuestión manteniendo las 10 secciones y sus características generales, respetando el discurso musical, obteniendo “infinitas” versiones suficientemente distintas entre ellas (fenotipo).

Entonces, la obra no es solo el resultado de cada versión sino también el proceso. Mientras no se ejecute el algoritmo, todas las obras potenciales pueden ser consideradas “La obra” en cuestión. Podríamos pensar como un campo de posibilidades, claramente no ilimitadas, pero altamente indeterminadas por su aleatoriedad interna. Solo al ejecutarse el código en el software, es cuando se “precipita” una de estas posibilidades y podemos escuchar una de las versiones

en concreto. Por lo tanto, parece útil pensar en el principio de “incertidumbre de Heisenberg”⁷ a modo de analogía libre, ya que, mientras la obra no se ejecute podemos considerar todas las versiones como “la obra”, pero si un tercero gatilla o activa el software, éste generará una versión en concreto, por lo que en ese momento, y solo en ese momento, dejara de tener versiones en potencia para fijarse exclusivamente en esa versión, la que se realizara en forma audio.

⁷ Resulta imposible medir simultáneamente, y con precisión absoluta, el valor de la posición y la cantidad de movimiento de una partícula

VII. Bibliografía

- Becerra-Schmidt, G. (1958). *Crisis de la enseñanza de la composición en Occidente*. II. Ritmo. Revista Musical Chilena, 12(59), pp. 48-75.
- Bohm, D., Peat, D., & Peat, F. D. (1988). *Ciencia, orden y creatividad: las raíces creativas de la ciencia y la vida*. Editorial Kairós.
- Bosseur, J. Y. (1992). *Vocabulaire de la musique contemporaine*. Minerve.
- Cádiz, R.(2010) *Introducción a la Música Computacional*, Fondo de Desarrollo de la Docencia, Vicerrectoría Académica, Pontificia Universidad Católica de Chile.
- Campaña, C., Bru, R., Browne, G. C., & Velázquez, D. (2008). *El arte de la cita: Velázquez en la obra de Bru y Cienfuegos*. Museo Nacional de Bellas Artes.
- Scholes, P. A. (1981). *Diccionario Oxford de la música*. Ed. Arte y literature.
- Dibelius, U. (2004). *La música contemporánea a partir de 1945* (Vol. 15). Ediciones AKAL
- Eimert. Herbert(1959) *¿Qué es la música electronica?*. Editorial Nueva Vision. Buenos Aires
- Gómez Urdinola, L. F., & Carvajal Ramírez, P. F. (2014). *Composición y producción de 8 temas de música electrónica a partir de sonidos concretos, técnicas de manipulación y síntesis sonora* (Bachelor's thesis, Facultad de Artes).

Guarello, F. (2006). *Composición musical: consideraciones generales*. Revista Resonancias n° 18 IMUC pp 11-33. Santiago de Chile

Marco, Tomás (2002) *Pensamiento musical y siglo XX*. Fundación Autor. Madrid

Prieberg, Fred(1961) *Música de la era Técnica*. Editorial Universitaria de Buenos Aires

Reck Miranda, E. (1999). *Música y nuevas tecnologías. Perspectivas para el siglo XXI*. L, ANGELOT, Barcelona.

Riemann, H., & y Maneja, A. R. (1929). *Composición musical*. Editorial Labor. España

Sirvent, Laura (2014) *Estudio e implementación de métodos composición algorítmica con propósitos explorativos* (Tesis de Pregrado) Universidad de Alicante. España

Supper, M. (2004). *Música electrónica y música con ordenador: historia, estética, métodos, sistemas*. Alianza Editorial.

VIII.Anexos

A continuación, y a modo de posible consulta dejo el código de la obra completo y tal como fue creado.

Sonic Pi versión 2.11.1 Lenguaje de programación Ruby

```
1.use_random_seed Time.new.usec
2.print Time.new.usec
3.#use_random_seed 50000
4.vol1 = 0.3
5.alt1 = 69
6.vol2 = 0.3
7.alt2 = 69
8.suma2 = 0.0
9.suma1 = 0.0
10.a = rrand_i(1,5)
11.define :voz1a do
12.with_fx :reverb , room: 0.8 do
13.use_synth :piano
14.use_bpm 45
15.if one_in(10)
16.play_pattern_timed [alt1+suma1],[0.33], amp: rand
17.play_pattern_timed [alt1-7+suma1],[0.33], amp: rand
18.play_pattern_timed [alt1-5+suma1],[0.33], amp: rand
19.suma1= suma1 + 1
20.else
21.if one_in(10)
22.play_pattern_timed [alt1],[0.33], amp: rand if one_in(2)
23.play_pattern_timed [alt1-7],[0.33], amp: rand if one_in(2)
24.play_pattern_timed [alt1-5],[0.33], amp: rand if one_in(2)
25.else
26.sleep 0.7 if one_in(10)
```



```

27.play_pattern_timed [alt1],[0.33], amp: rand
28.play_pattern_timed [alt1-7],[0.33], amp: rand
29.play_pattern_timed [alt1-5],[0.33], amp: rand
30.end
31.end
32.end
33.end
34.define :voz2a do
35.th_fx :reverb , room: 0.8 do
36.use_synth :piano
37.use_bpm 45
38.if one_in(5)
39.sleep 0.5 if one_in(30)
40.play_chord [alt2-28, alt2-21], amp: rand
41.sleep 0.5
42.play_chord [alt2-28, alt2-21], amp: rand
43.sleep 0.5
44.else
45.if one_in(5)
46.play_chord [alt2-24+suma2, alt2-17+suma2], amp: rand
47.sleep 0.5
48.play_chord [alt2-24+suma2, alt2-17+suma2], amp: rand
49.sleep 0.5
50.suma2= suma2 + 1
51.else
52.sleep 0.3 if one_in(10)
53.play_chord [alt2-24, alt2-17], amp: rand if one_in(2)
54.sleep 0.5
55.play_chord [alt2-24, alt2-17], amp: rand if one_in(2)
56.sleep 0.5
57.end
58.end
59.end
60.end
61.define :voz1b do

```

```

62.with_fx :reverb , room: 0.8 do
63.use_synth :piano
64.use_bpm 50
65.play_choose([60,61,62,63,64,65,66,67,68,69,70,71,72]), amp:
    rrand(0,3) if one_in(2)
66.sleep 0.33
67.play_choose([60,61,62,63,64,65,66,67,68,69,70,71,72]), amp:
    rrand(0,3) if one_in(2)
68.sleep 0.33
69.play_choose([60,61,62,63,64,65,66,67,68,69,70,71,72]), amp:
    rrand(0,3) if one_in(2)
70.sleep 0.33
71.end
72.end
73.define :voz2b do
74.with_fx :reverb , room: 0.8 do
75.use_synth :piano
76.use_bpm 50
77.play_chord [60,61,62,63,64,65,66,67,68,69,70,71,72], amp:
    rand if one_in(2)
78.sleep 0.5
79.play_chord [60,61,62,63,64,65,66,67,68,69,70,71,72], amp:
    rand if one_in(2)
80.sleep 0.5
81.end
82.end
83.define :voz1c do
84.with_fx :reverb , room: 1 do
85.use_synth :piano
86.use_bpm 10
87.if one_in(10)
88.play_pattern_timed [alt1+suma1],[0.33], amp: rand, release: 2
89.play_pattern_timed [alt1-7+suma1],[0.33], amp: rand, release:

```

```

90.play_pattern_timed [alt1-5+suma1],[0.33], amp: rand, release:
  2
91.suma1= suma1 + 1
92.else
93.if one_in(10)
94.play_pattern_timed [alt1],[0.33], amp: rand , release: 2 if
  one_in(2)
95.play_pattern_timed [alt1-7],[0.33], amp: rand, release: 2 if
  one_in(2)
96.play_pattern_timed [alt1-5],[0.33], amp: rand, release: 2 if
  one_in(2)
97.else
98.sleep 0.7 if one_in(10)
99.play_pattern_timed [alt1],[0.33], amp: rand, release: 2
100.play_pattern_timed [alt1-7],[0.33], amp: rand, release: 2
101.play_pattern_timed [alt1-5],[0.33], amp: rand, release: 2
102.end
103.end
104.end
105.end
106.define :voz2c do
107.with_fx :reverb , room: 1 do |tempo|
108.use_synth :piano
109.use_bpm 10
110.if one_in(5)
111.sleep 0.5 if one_in(30)
112.play_chord [alt2-28, alt2-21], amp: rand, release: 2
113.sleep 0.5
114.play_chord [alt2-28, alt2-21], amp: rand, release: 2
115.sleep 0.5
116.else
117.if one_in(5)
118.play_chord [alt2-24+suma2, alt2-17+suma2], amp: rand,
  release: 2
119.sleep 0.5

```

```
120.play_chord [alt2-24+suma2, alt2-17+suma2], amp: rand,  
    release: 2  
121.sleep 0.5  
122.suma2= suma2 + 1  
123.else  
124.sleep 0.3 if one_in(10)  
125.play_chord [alt2-24, alt2-17], amp: rand, release: 2  
126.sleep 0.5  
127.play_chord [alt2-24, alt2-17], amp: rand, release: 2  
128.sleep 0.5  
129.end  
130.end  
131.end  
132.end  
133.define :master1 do  
134.in_thread do  
135.voz1a  
136.end  
137.voz2a  
138.end  
139.define :master2 do  
140.in_thread do  
141.voz1b  
142.end  
143.voz2b  
144.end  
145.define :master3 do  
146.in_thread do  
147.voz1c  
148.end  
149.voz2c  
150.end  
151.define :a1 do |x, min, max|  
152.with_fx :reverb , room: 1 do  
153.use_synth :piano
```

```

154.use_bpm x
155.a= choose([2,4,9])
156.sleep rand if one_in(2)
157.play choose([a+(12*(rrand_i(min,max) ))]), amp: rand , pan:
    rrand(-1,1)#, hard: rand, attack: rand , decay: rand ,sustain:
    rand ,release: rand #if one_in(2)
158.#sleep rand if one_in(2)
159.sleep 0.33 #if one_in(2)
160.play choose([a+(12*(rrand_i(min,max) ))]), amp: rand , pan:
    rrand(-1,1)#, hard: rand , attack: rand, decay: rand ,sustain:
    rand, release: rand #if one_in(2)
161.#sleep 0.33 if one_in(2)
162.sleep rand #if one_in(2)
163.play choose([a+(12*(rrand_i(min,max) ))]), amp: rand , pan:
    rrand(-1,1)#, hard: rand , attack: rand, decay: rand ,sustain:
    rand, release: rand #if one_in(2)
164.#sleep 0.33 if one_in(2)
165.sleep rand #if one_in(2)
166.end
167.end
168.define :b1 do |x, min, max|
169.with_fx :reverb , room: 1 do
170.use_synth :piano
171.use_bpm x
172.a= choose([9,16,21])
173.sleep rand if one_in(2)

174.play_chord [9+(12*(rrand_i(min,max)
    )),16+(12*(rrand_i(min,max) )),21+(12*(rrand_i(min,max)
    ))], amp: rand , pan: rrand(-1,1)#, hard: rand, attack: rand ,
    decay: rand ,sustain: rand ,release: rand #if one_in(2)
175.#sleep rand if one_in(2)
176.sleep 0.5 #if one_in(2)
177.play_chord [9+(12*(rrand_i(min,max)
    )),16+(12*(rrand_i(min,max) )),21+(12*(rrand_i(min,max)

```

```

    )], amp: rand , pan: rrand(-1,1)#, hard: rand, attack: rand ,
    decay: rand ,sustain: rand ,release: rand #if one_in(2)
178.sleep 0.5 #if one_in(2)
179.sleep rand #if one_in(2)
180.end
181.end
182.define :a2 do |x, min, max|
183.with_fx :reverb , room: 1 do
184.use_synth :piano
185.use_bpm x
186.a= choose([1,2,3,4,5,6,7,8,9,10,11])
187.sleep rand if one_in(2)
188.play choose([a+(12*(rrand_i(min,max) ))]), amp: rand , pan:
    rrand(-1,1), hard: rand, attack: rand , decay: rand ,sustain:
    rand ,release: rand #if one_in(2)
189.#sleep rand if one_in(2)
190.sleep 0.33 #if one_in(2)
191.play choose([a+(12*(rrand_i(min,max) ))]), amp: rand , pan:
    rrand(-1,1), hard: rand , attack: rand, decay: rand ,sustain:
    rand, release: rand #if one_in(2)
192.#sleep 0.33 if one_in(2)
193.sleep rand #if one_in(2)
194.play choose([a+(12*(rrand_i(min,max) ))]), amp: rand , pan:
    rrand(-1,1), hard: rand , attack: rand, decay: rand ,sustain:
    rand, release: rand #if one_in(2)
195.#sleep 0.33 if one_in(2)
196.sleep rand #if one_in(2)
197.end
198.end
199.define :b2 do |x, min, max|
200.with_fx :reverb , room: 1 do
201.use_synth :piano
202.use_bpm x
203.a= choose([1,2,3,4,5,6,7,8,9,10,11])
204.sleep rand if one_in(2)

```

```

205.play_chord [9+(12*(rrand_i(min,max)
    )),16+(12*(rrand_i(min,max) )),21+(12*(rrand_i(min,max)
    ))], amp: rand , pan: rrand(-1,1), hard: rand, attack: rand ,
    decay: rand ,sustain: rand ,release: rand #if one_in(2)
206.#sleep rand if one_in(2)
207.sleep 0.5 #if one_in(2)
208.play_chord [9+(12*(rrand_i(min,max)
    )),16+(12*(rrand_i(min,max) )),21+(12*(rrand_i(min,max)
    ))], amp: rand , pan: rrand(-1,1), hard: rand, attack: rand ,
    decay: rand ,sustain: rand ,release: rand #if one_in(2)
209.sleep 0.5 #if one_in(2)
210.sleep rand #if one_in(2)
211.end
212.end
213.define :a3 do |x, min, max|
214.with_fx :reverb , room: 0.6 do
215.use_synth :piano
216.use_bpm x
217.a= choose([1,2,3,4,5,6,7,8,9,10,11])
218.sleep rand if one_in(2)
219.play choose([a+(12*(rrand_i(min,max) ))]), amp: rand , pan:
    rrand(-1,1), hard: rand, attack: 0.05 , release: 0.05 #if
    one_in(2)
220.#sleep rand if one_in(2)
221.sleep 0.33 #if one_in(2)
222.play choose([a+(12*(rrand_i(min,max) ))]), amp: rand , pan:
    rrand(-1,1), hard: rand, attack: 0.05 , release: 0.05 #if
    one_in(2)
223.#sleep 0.33 if one_in(2)
224.sleep rand #if one_in(2)
225.play choose([a+(12*(rrand_i(min,max) ))]), amp: rand , pan:
    rrand(-1,1), hard: rand, attack: 0.05 , release: 0.05 #if
    one_in(2)
226.#sleep 0.33 if one_in(2)
227.sleep rand #if one_in(2)

```

```

228.end
229.end
230.define :b3 do |x, min, max|
231.with_fx :reverb , room: 0.6 do
232.use_synth :piano
233.use_bpm x
234.a= choose([1,2,3,4,5,6,7,8,9,10,11])
235.sleep rand if one_in(2)
236.play_chord [9+(12*(rrand_i(min,max)
    )),16+(12*(rrand_i(min,max) )),21+(12*(rrand_i(min,max)
    ))], amp: rand , pan: rrand(-1,1), hard: rand, attack: 0.05
    ,release: 0.05 #if one_in(2)
237.#sleep rand if one_in(2)
238.sleep 0.5 #if one_in(2)
239.play_chord [9+(12*(rrand_i(min,max)
    )),16+(12*(rrand_i(min,max) )),21+(12*(rrand_i(min,max)
    ))], amp: rand , pan: rrand(-1,1), hard: rand, attack: 0.05
    ,release: 0.05 #if one_in(2)
240.sleep 0.5 #if one_in(2)
241.sleep rand #if one_in(2)
242.end
243.end
244.define :a4 do |x|
245.with_fx :reverb , room: 0.8 do
246.use_synth :piano
247.use_bpm x
248.af= rrand_i(15,100)
249.sleep rand if one_in(2)
250.play_pattern_timed [af],[0.33], amp: rrand(0,2) ,pan: rrand(-
    1,1) if one_in(2)
251.play_pattern_timed [af-7],[0.33], amp: rrand(0,2) ,pan:
    rrand(-1,1) if one_in(2)
252.play_pattern_timed [af-5],[0.33], amp: rrand(0,2) , pan:
    rrand(-1,1) if one_in(2)
253.end

```



```

254.end
255.define :b4 do |x|
256.with_fx :reverb , room: 0.8 do
257.use_synth :piano
258.use_bpm x
259.ar= rrand_i(15,100)
260.sleep rand if one_in(2)
261.play_chord [ar-12, ar-5, ar],amp: rrand(0,2) ,pan: rrand(-
    1,1)  if one_in(2)
262.sleep 0.5
263.play_chord [ar-12, ar-5, ar],amp: rrand(0,2) ,pan: rrand(-
    1,1)  if one_in(2)
264.end
265.end
266.define :antes1 do |x, min, max|
267.with_fx :reverb , room: 1 do
268.use_synth :piano
269.use_bpm x
270.play rrand_i(min,max),pan: rrand(-1,1),hard:0 ,amp:
    rrand(0,2)
271.sleep rand
272.play rrand_i(min,max),pan: rrand(-1,1),hard:0 ,amp:
    rrand(0,2)
273.sleep rand
274.play rrand_i(min,max),pan: rrand(-1,1),hard:0 ,amp:
    rrand(0,2)
275.sleep rand
276.play rrand_i(min,max),pan: rrand(-1,1),hard:0 ,amp:
    rrand(0,2)
277.sleep rand
278.end
279.end

280.define :antes2 do |x, z|
281.with_fx :reverb , room: 0.8 do

```

```

282.use_synth :piano
283.use_bpm x
284.in_thread do
285.play rrand_i(1,127) , amp: rand ,pan: rrand(-1,1) if one_in(z)
286.sleep 0.33
287.play rrand_i(1,127), amp: rand ,pan: rrand(-1,1) if one_in(z)
288.sleep 0.33
289.play rrand_i(1,127) , amp: rand ,pan: rrand(-1,1) if one_in(z)
290.sleep 0.33
291.end
292.play_chord [rrand_i(1,127), rrand_i(1,127),rrand_i(1,127)],
    amp: rand , pan: rrand(-1,1) if one_in(z)
293.sleep 0.5
294.play_chord [rrand_i(1,127), rrand_i(1,127),rrand_i(1,127)],
    amp: rand , pan: rrand(-1,1) if one_in(z)
295.sleep 0.5
296.end
297.end
298.define :antes3 do |x, z|
299.with_fx :reverb , room: 0.8 do
300.use_synth :piano
301.use_bpm x
302.in_thread do
303.play 69 , amp: rand ,pan: rrand(-1,1) if one_in(z)
304.sleep 0.33
305.play 62 , amp: rand ,pan: rrand(-1,1) if one_in(z)
306.sleep 0.33
307.play 64 , amp: rand ,pan: rrand(-1,1) if one_in(z)
308.sleep 0.33
309.end
310.play_chord [45,52,57], amp: rand , pan: rrand(-1,1) if
    one_in(z)
311.sleep 0.5
312.play_chord [45,52,57], amp: rand , pan: rrand(-1,1) if
    one_in(z)

```

```

313.sleep 0.5
314.end
315.end
316.#####
    #####
317.#####
    #####
318.10.times do
319.antes2 80, 5
320.end
321.10.times do
322.antes2 70, 4
323.end
324.5.times do
325.antes2 60, 3
326.end
327.5.times do
328.antes2 50, 2
329.end
330.10.times do
331.antes2 45, 1
332.end
333.6.times do
334.antes3 45, 2
335.end
336.6.times do
337.antes3 45, 1
338.end
339.#####
    #####
340.60.times do
341.master1
342.end
343.20.times do
344.in_thread do

```

```
345.master2
346.end
347.master1
348.end
349.20.times do
350.in_thread do
351.master3
352.end
353.master2
354.end
355.20.times do
356.in_thread do
357.master1
358.end
359.master3
360.end
361.20.times do
362.in_thread do
363.a1 45, 4, 6
364.end
365.b1 45 ,0, 9
366.end
367.10.times do
368.in_thread do
369.a1 15, 0, 4
370.end
371.b1 15 ,0, 4
372.end
373.20.times do
374.in_thread do
375.a2 45, 0, 9
376.end
377.b2 60 ,0, 9
378.end
379.20.times do
```

```
380.in_thread do
381.b2 80, 7, 9
382.end
383.b2 60 ,7, 9
384.end
385.10.times do
386.in_thread do
387.a2 80, 8, 9
388.end
389.a2 60 ,8, 9
390.end
391.10.times do
392.in_thread do
393.a2 80, 7, 9
394.end
395.a2 60 ,7, 9
396.end
397.10.times do
398.in_thread do
399.a2 80, 6, 9
400.end
401.a2 60 ,6, 9
402.end
403.10.times do
404.in_thread do
405.b2 180, 0, 9
406.end
407.a2 180 ,0, 9
408.end
409.10.times do #antes de esto
410.in_thread do
411.a2 144, 7, 9
412.end
413.in_thread do
414.a3 10, 4, 6
```

```
415.end
416.a2 144 ,7, 9
417.end
418.40.times do
419.in_thread do
420.a3 288 , 0, 9
421.end
422.in_thread do
423.b4 15 if one_in(2)
424.end
425.b3 144 , 0, 9
426.end
427.20.times do
428.in_thread do
429.a3 144 , 2, 9
430.end
431.in_thread do
432.b3 144 , 2, 9
433.end
434.in_thread do
435.a4 10
436.end
437.b4 10
438.end
439.20.times do
440.in_thread do
441.a4 45
442.sleep rand
443.a4 144 if one_in(2)
444.end
445.sleep rand
446.if one_in(2)
447.b4 45
448.else
449.b4 144
```

```
450.end
451.end
452.sleep rrand_i(1,10)
453.3.times do
454.antes1 100, 1,127
455.end
456.10.times do
457.in_thread do
458.antes1 100, 1,40
459.end
460.antes1 100, 1, 40
461.end
462.10.times do
463.in_thread do
464.antes1 140,40,80
465.end
466.in_thread do
467.antes1 140,40,80
468.end
469.antes1 140,40,80
470.end
471.10.times do
472.in_thread do
473.antes1 200,80,127
474.end
475.in_thread do
476.antes1 200,80,127
477.end
478.in_thread do
479.antes1 200,80,127
480.end
481.in_thread do
482.antes1 200,80,127
483.end
484.antes1 200,80,127
```

```
485.end
486.50.times do
487.in_thread do
488.antes1 300,1,127
489.end
490.in_thread do
491.antes1 300,1,127
492.end
493.in_thread do
494.antes1 300,1,127
495.end
496.in_thread do
497.antes1 300,1,127
498.end
499.antes1 300,1,127
500.end
501.sleep 10
```