



**Universidad  
de Valparaíso**  
CHILE

UNIVERSIDAD DE VALPARAÍSO  
FACULTAD DE HUMANIDADES  
INSTITUTO DE FILOSOFÍA  
CARRERA DE MÚSICA

CREACIÓN DE PIEZAS DE  
*COMPUTER MUSIC* EN  
SUPERCOLLIDER



Proyecto de título para optar al grado de  
Licenciado en arte, tecnología y gestión musical y  
al título profesional de Músico con mención en  
ejecución instrumental o canto

DIEGO ANDRÉS HENRÍQUEZ NEIRA

Profesor Guía: Pablo César Palacios Torres

Valparaíso, Chile

2022

# Dedicatoria y agradecimientos

*Este trabajo de investigación que permite la finalización de un largo proceso, donde me inicié formalmente en estudios musicales y luego técnicos, está dedicado principalmente a mi familia.*

También agradezco encarecidamente a:

De mi universo musical, a los profesores Paul Hernández, Ángela Vallejos y Carlos Araya, de quienes tengo muy buenos recuerdos en mi primera etapa de formación musical. Así también al profesor Pablo Palacios, que me ayudó en esta ocasión a ordenar toda la información que quería transmitir en este documento. Por último al profesor Andrés Rivera, de quien por primera vez escuché el término live coding.

De mi universo ingenieril, le agradezco a mis compañeros del laboratorio de computadores Labproce, el lugar donde pude profundizar muchos conocimientos y donde tuve mis primeras prácticas de computer music.

# Resumen

El presente trabajo tiene como finalidad principal crear música que se enmarque en el género de computer music usando la aplicación SuperCollider. Esto es logrado en primer lugar mediante un relato académico que circunscribe las categorías computer music y composición algorítmica. Este relato es devenido de una investigación en libros, artículos de revistas, información audiovisual. Luego, se incluye una descripción breve y general del software SuperCollider, ofreciendo terminología que ayudará a entender el proceso creativo. Finalmente, usando una autoetnografía, método que permite revelar la cultura de un creador, se documentan las diferentes prácticas creativas que se tuvieron durante el desarrollo del proyecto. Los resultados de las prácticas los hemos enmarcado dentro de lo que se conoce como live coding, un subgénero de computer music que se enfoca en la creación y alteración de algoritmos durante ejecuciones musicales en vivo, en este caso, mediante la herramienta y lenguaje de programación SuperCollider.

**Tags: computer music, composición algorítmica, supercollider, live coding, autoetnografía.**

## Índice

Introducción	4
<b>1. El computador como instrumento musical</b>	<b>7</b>
1.1 <i>Computer music</i> , una búsqueda interdisciplinaria	7
1.2 Preocupaciones estéticas en <i>computer music</i>	10
1.3 Max Mathews, Music-N y la unidad generadora	11
1.4 Computación, formalización, Xenakis y el uso de matemáticas en la música	14
1.5 Computación y composición	15
<b>2. Composición algorítmica y <i>computer music</i></b>	<b>17</b>
2.1 Definiciones, distinciones y antecedentes artísticos	17
2.2 Live Coding y música interactiva, nuevas estéticas de <i>computer music</i>	18
<b>3. SuperCollider: una plataforma para la composición algorítmica</b>	<b>21</b>
3.1 Aplicación SC, estructura y funcionamiento	22
3.2 Lenguaje de programación SC	25
<b>4. Creación de piezas musicales para computer music en SuperCollider</b>	<b>30</b>
4.1 Palabras preliminares	31
4.1.1 La labor creativa en los albores de la cuarta revolución industrial	31
4.1.2 De la irracionalidad, la racionalidad y la naturaleza humana	31
4.1.3 Codificación / como ejecución / como creación / como relación	32
4.2 Práctica creativa I: Registro de audiciones musicales	32
4.3 Práctica creativa II: Improvisaciones libres de live coding	40
4.3.1 Comentarios sobre la creación de sintetizadores y patrones para improvisaciones de live coding	41
4.3.2 Una reflexión cibernética para la creación de un algoritmo de improvisación en SC	52
Conclusión y discusión final	58
Referencias en orden alfabético	60
Índice de Figuras, Tablas y Ecuaciones	64
Anexo A	66
Anexo B	69
Anexo C	72

# Introducción

El trabajo que presentamos a continuación trata sobre la creación de piezas musicales usando la computadora como instrumento principal, y el software *SuperCollider* (también referido como SC). Aquí, el proceso creativo es acompañado por un relato que permite al lector aproximarse a la categoría *computer music*, la que durante el siglo pasado fue dando forma a una estética que fusionó el arte y la ciencia. En este tipo de esfuerzo académico, el lenguaje técnico de disciplinas extra musicales es inevitable, por esto ofrecemos algunas herramientas conceptuales, devenidas de investigadores clásicos y actuales, que permitan orientar y contextualizar a quién por primera vez se enfrenta a este término. Las creaciones que resultan de las prácticas que se describen aquí se nutren de todo el marco teórico investigado.

El presente trabajo de tesis no lo hemos de considerar una labor propiamente compositiva, ya que no es objetivo entrar en tal categoría con la profundidad que hubiese correspondido, en su lugar, hemos elegido hablar de creaciones, resultantes de prácticas creativas, y así finalmente equilibrar los demás ámbitos de investigación.

Se documenta las prácticas creativas mediante una autoetnografía, que la definiremos como “un estudio de la introspección individual en primera persona, que pretende arrojar luz sobre la cultura a la que pertenece el sujeto” (López-Cano & Opazo, 2014, p. 141). Este método nos permitirá dar cuenta de cómo se trabajó con SuperCollider en la preparación de improvisaciones de live coding. Se añade un resultado de creación, donde se utiliza una algoritmia inspirada desde la observación cotidiana de espacios donde las tecnologías de la computación son usadas para regular espacios públicos, específicamente el metro que recorre Limache a Valparaíso.

Aquí responderemos, desde una exploración académica, a la pregunta: ¿Cómo podemos comprender/problematizar el ejercicio de la creación mediante el software SuperCollider?

Entonces, el presente trabajo tiene como finalidad principal crear música que se enmarque en el género de *computer music* usando la aplicación SuperCollider. Un primer objetivo secundario será explicar el lugar que se ha elegido para el acto de creación, esto es, desde las categorías principales que caracterizan las piezas musicales resultantes: *computer music* y composición algorítmica. El último objetivo es realizar

una descripción general del software SuperCollider, elegido como plataforma para la creación y mostrar posibilidades de uso.

Las razones para realizar un trabajo de creación dentro de los márgenes escogidos son variadas. En primer lugar la exploración sonora a través de la computadora, lo cual ya configura una tarea compleja, sin embargo, aquí hemos dado algunas posibilidades de uso. Luego está la dimensión estética de la máquina misma, y para esto hemos generado observaciones de una selección de trabajos relacionados, teniendo en cuenta consideraciones históricas, técnicas y otras, que permitan luego generar resultados en la labor de creación. Otra razón es la oportunidad de observarse para reconocer el lugar cultural desde donde aparecerá una música.

Actualmente cada vez es más frecuente que músicos introduzcan en sus prácticas y/o flujos de trabajo alguna herramienta computarizada, ya sea como usuarios o productores de estas. También, algo que ha pasado durante las últimas décadas es que el género ya no es sólo tema de investigación académica, la aparición de comunidades dedicadas al live coding ha ido en aumento, en esto los medios de masificación de contenidos, tales como YouTube, Bandcamp, Spotify o Twitch, han servido para que estas comunidades o bien también individuos puedan mostrar sus trabajos.

Al momento de terminar este documento el contenido relacionado directamente con el tema de investigación y sus resultados se sigue incrementando. Eli Fieldsteel (Universidad de Illinois, EE.UU.) comparte el período de enseñanza 2022 de su curso sobre SuperCollider<sup>1</sup>, poco antes anuncia que publicará un libro sobre ello. Otros canales relacionados a SC son los de individuos como SynthDef<sup>2</sup>, James Harkins<sup>3</sup>, Sound Engraver<sup>4</sup>. El medio Eulerroom presenta en su canal el evento *i-Now - Pattern Club Live*<sup>5</sup>, organizado por DINA Venue (Inglaterra) y con la curaduría de Alex McClean. En Bandcamp, en la categoría de *computer music*, lanzamientos de discos como Ally Choir (Bride), tv music (samb\_rules) y Seven Sequences (Craig Pepples, David Hicks), nos dan muestra de lo vivo que se encuentra el género en este medio. Por último, notable es la comunidad de live coders en la red social de programadores

---

<sup>1</sup> Intro to Audio Coding in SuperCollider - MUS 499C Fall 2022: [https://www.youtube.com/playlist?list=PLPYzvS8A\\_rTYEba\\_4SDvRJylyjKaDNjn9](https://www.youtube.com/playlist?list=PLPYzvS8A_rTYEba_4SDvRJylyjKaDNjn9). Sitio oficial: <https://www.elifieldsteel.com/>

<sup>2</sup> [https://www.youtube.com/@synth\\_def](https://www.youtube.com/@synth_def)

<sup>3</sup> <https://www.youtube.com/@jamesharkins4272>. Harkins también tiene un material explicativo de patrones en SuperCollider: [http://doc.sccode.org/Tutorials/A-Practical-Guide/Pg\\_01\\_Introduction.html](http://doc.sccode.org/Tutorials/A-Practical-Guide/Pg_01_Introduction.html)

<sup>4</sup> <https://www.youtube.com/@SoundEngraver>, <https://soundengraver.com/>

<sup>5</sup> DINA i-now: Sheffield Pattern Live. <https://www.youtube.com/watch?v=5dixc2zwK8I>

GitHub, donde se encuentra el perfil del colectivo TOPLAP<sup>67</sup>, que reúne una cantidad importante de información acerca de software para sintetización.

En conclusión, el espacio creativo en torno a SC, sistemas similares, y las prácticas generadas de las personas y los colectivos tiene una efervescente actividad en la red, al punto que es difícil sacar una fotografía sobre los trabajos previos como estado del arte. Podemos deducir que realizar un estado del arte, como tradicionalmente se comprende, carece de relevancia al momento del volumen de información que se genera día a día.

---

<sup>6</sup> Temporal Organisation for the Pragmatics of Live Artistic Programming, sitio oficial: <https://toplap.org/>

<sup>7</sup> <https://github.com/toplap/awesome-livecoding>

# 1. El computador como instrumento musical

El uso de tecnologías en la música se ha discutido desde varias perspectivas. Marshall McLuhan (1964) se refería a estas como extensiones o separaciones de las facultades humanas, como cuando se piensa en la creación de instrumentos musicales. Douglas Keislar (2011) esquematiza los desarrollos en tecnología musical antes del computador, su análisis permite reunir conceptos para tener una perspectiva histórica y ofrece una forma de catalogar las relaciones entre instrumentos y músicos. A esto pueden sumarse las perspectivas de individuos que participaron activamente en los avances de las ciencias computacionales en sus inicios, como Alan Turing, Norbert Wiener y John von Neumann, claves en el inicio del movimiento cibernético. Por último, destaca la figura de Max V. Mathews, considerado el padre de *computer music*, creador del primer programa computacional musical y la unidad generadora, concepto que sigue vigente aún en aplicaciones como SuperCollider.

En este primer capítulo expondremos la conceptualidad que rodea a *computer music*. Primero desde las definiciones y distinciones de autores relacionados con la disciplina. Luego hablaremos brevemente de Max Mathews, figura clave de esta y su punto de partida, también de algunos de sus principales aportes. Finalmente, complementaremos y daremos contexto al instrumento elegido para crear música, desde un génesis tecnológico a la razón del uso de las matemáticas en música.

## 1.1 *Computer music*, una búsqueda interdisciplinaria

Los primeros trabajos musicales hechos para computadoras fueron hechos en los laboratorios Bell a manera de experimentos en 1957, el ingeniero Max Vernon Mathews usa una computadora de uso general de esos años para crear los primeros sonidos musicales y junto al psicólogo Newman Guttman crean la pieza *The Silver Scale*, dando inicio a *computer music*. En 1961 Mathews logra remediar las primeras dificultades encontradas, que se debían a equipos que no estaban diseñados para desempeñar tales actividades, y en 1962 varios miembros del equipo de ingenieros del laboratorio realizan un conjunto de composiciones reunidas en *Music From Mathematics* (Pierce, 1996). En el canal de Alex di Nunzio en youtube

(<https://www.youtube.com/c/alexdinunzio/channels>) se puede escuchar esta y otras piezas más de la historia del género.

En su investigación, Douglas Keislar (2011, pp. 11-12) distingue dos usos principales del término *computer music*:

(1) un género musical o categoría, análoga a la sinfonías, jazz combo, y similares, en cual la computadora juega una parte en la composición, performance, o realización sonora; (2) una disciplina técnica, análoga a computación gráfica (un campo de las ciencias de la computación), que abarca muchos aspectos del uso de computadores en aplicaciones relacionadas a la música.

Nick Collins (2010) se refiere al término primero de forma general, como “una música que involucra una computadora en cualquier etapa de su ciclo de vida” (p. 1), luego de forma estricta escribe que es “una música que es creada enteramente en una computadora, basada principalmente en tecnología computacional para la concepción artística, y no puede tener ninguna existencia independiente” (p. 2). Por último, añade también una tercera posición, aludiendo a F. Richard Moore, uno de los primeros músicos entrenados que trabajó junto a Max Mathews y John Pierce en los laboratorios Bell, haciendo un alcance más comprensivo del término dice que “es una búsqueda interdisciplinaria, en la intersección de las artes y las ciencias” (Collins, 2010, p. 3). Para el propio Pierce (1996, p. 50), *computer music* es “cualquier cosa que tenga que ver con la generación digital de sonidos musicales –comercial o académica”.

La Figura 1-1, es un esquema propuesto por Moore, donde se observan afuera disciplinas que han sido trabajadas por separado y hacia el interior se encuentran subdisciplinas de interés en *computer music*. En su obra expresa que “entender la relación entre las propiedades objetivas y subjetivas del sonido están en el corazón del *computer music*” (Moore, 1990, p. 24).



Figura 1-1 Contexto disciplinario de computer music (subdisciplinas esenciales en *italica*)(trad.).

(Fuente: Elements of computer music. (Moore, 1990, p. 24)

Este contexto puede incluso ser ampliado a las matemáticas, ya que se han usado varios recursos conceptuales y técnicos desde esta disciplina. Los primeros que usualmente se mencionan, son los que introduce Xenakis en su música estocástica, distribuciones Gaussianas y cadenas de Markov. También se han generado composiciones automáticas a partir de modelos de sistemas dinámicos no lineales, en particular algoritmos genéticos y sistemas caóticos (Supper, 2001; Bidblack, 1992).

Actualmente nos encontramos en un momento tecnológico en el que los dispositivos computarizados permiten realizar procedimientos de síntesis, networking, espacialización, procesamiento, conversión AD/DA, entre otras técnicas del repertorio de computer music en tiempo real y capaces de ser ejecutadas en equipos que han sido cada vez más asequibles para una importante cantidad de población. Los costos de fabricación han bajado constantemente por décadas, incluso sin grandes cambios a las partes básicas (Hennessy y Patterson, 2017). Una consecuencia de la asequibilidad de más y mejores computadores es el nacimiento de comunidades de software abierto. Una de estas es la de usuarios y desarrolladores de SuperCollider, aplicación creada por James McCartney, que se ha convertido en la encargada de mantener y actualizar las versiones del programa.

Todo esto ha causado que la cantidad de música hecha usando tecnología de computadores crezca considerablemente. Entonces cabe revisar cuáles han sido las preocupaciones y resultados a la hora de emprender la tarea de creación de piezas de *computer music*.

## 1.2 Preocupaciones estéticas en *computer music*

Desde temprano en *computer music*, Mathews tuvo un interés por la psicoacústica debido al potencial tímbrico de los computadores. Muchas obras compuestas en este género gravitan fuertemente en torno a la cuestión tímbrica, transcribiendo digitalmente timbres que podríamos llamar naturales, de voces e instrumentos convencionales, y también creando nuevos y exóticos sonidos. A la llegada de la síntesis análoga y digital, el instinto de los compositores requirió una nueva calibración (Mathews, 1974).

Algo que puede ocurrir en alguien que se compromete a la tarea de elaborar piezas de *computer music*, en un contexto de trabajo académico, es perderse en la investigación tecnológica (Croft, 2015), perdiendo de vista la meta. El debate proceso-producto puede dar cuenta de las “tensiones que hay entre la primacía del oído y la búsqueda de sistema” (Collins, 2010, p. 298), un equilibrio entre el interés por el resultado final de un trabajo musical y la obsesión con la máquina.

Collins (2011) realiza una pequeña revisión de músicas en las que la computadora juega un importante papel<sup>8</sup>, al respecto, observa (p. 273):

... la estereotípica música electrónica dance tiene un evidente sentido del ritmo para no hacer ambiguo el baile, mientras que la estructura rítmica en trabajos electroacústicos tiende a ser más experimental, a veces libre hasta el punto de ser amétricos (...). La elección de materia frecuencial se extiende desde escalas diatónicas en doce notas de temperamento igual para mucho del pop mainstream, a través de tonalidades alternativas (...), a barridos de glissandos y otras configuraciones más exóticas en espacio frecuencial que evoca a timbre, más que a tono estable. Los recursos tímbricos también varían desde instrumentaciones fijas de sintetizadores virtuales afinados y sonidos de percusiones sampleadas, hasta complejas transformaciones de sonido en tiempo y espacio son el canon central de la música.

El autor también se refiere también a los peligros de trabajar con estructuras muy amplias, estas puedan no tener el efecto que se desea debido a los períodos de concentración de los escuchas. También recomienda no descartar el uso de las formas clásicas, ya que pueden proveer “punto de acceso listos” (p. 296)<sup>9</sup> para los auditores, sin embargo “muchas interesantes formas musicales dejan que ejecutantes u observadores puedan elegir el curso de los eventos, incluyendo formas abiertas de trabajo, trabajos interactivos con compañeros musicales virtuales o videojuegos” (p.

<sup>8</sup> Ver Tabla 8.1 (Collins, 2011, p. 272)

<sup>9</sup> El texto original (Collins, 2011, p. 296) describe las formas musicales populares sobre-aprendidas como ‘ready-made access points for listeners’.

297). Un ejemplo que podría abarcar ambas perspectivas se da en la obra de Lejaren Hiller y Leonard Isaacson: *Illiad Suite for String Quartet* (1955-1956), la primera obra en musical escrita en una computadora digital<sup>10</sup>, donde los auditores “pueden no distinguir si la pieza fue generada por medio de composición algorítmica o tradicional” (Supper, 2001, p. 49).

El uso de los métodos estocásticos, mencionados anteriormente, y la mecánica estadística para el tratamiento de densidades sonoras en la música de Xenakis también puede considerarse la “fundación de una estética” (Collins, 2010, p. 305). Sin embargo, la relación entre entropía y estética es compleja, tanto como lo es la sensibilidad humana. Los patrones de tensión y relajación que excitan a un escucha usualmente son contrarios a la entropía (Xenakis, 1971).

Otra estética de *computer music* puede provenir de los cuatro criterios de la música electrónica<sup>11</sup>, expuestos por Stockhausen en sus charlas (Olano, 2013). Estos aspectos que el músico menciona pueden ser trabajados mediante código usando plataformas como SuperCollider con las diferentes unidades generadoras que disponen. Estos bloques funcionales y conceptuales son transversales en el *computer music* de hoy en día, por lo que es el tema del siguiente apartado en este capítulo.

### 1.3 Max Mathews, Music-N y la unidad generadora

En 1957, mientras trabajaba en los laboratorios de la entonces Bell Telephone en Murray Hill, Nueva Jersey (NJ), mejorando la calidad del sonido en sistemas de comunicación (Mathews, 1963), Max Vernon Mathews desarrolló el primer programa para computadora digital dedicado a la síntesis de sonidos musicales. El ingeniero eléctrico lo llamó *Music*. Las instrucciones eran compiladas en cinta magnética en Nueva York, donde estaba alojada la computadora arrendada por Bell, una IBM 704 (ver Figura 1-2), equipo hecho con tecnología de tubos de vacío. Esta fue la primera computadora con sistema de punto flotante, un sistema que amplía la capacidad de representar de forma binaria los dígitos y que fue aprovechado para desarrollar también los lenguajes FORTRAN y Lisp. Luego del compilado en cinta, esta era llevada a Bell en NJ, donde tenían un convertidor digital-análogo (DAC) de 12-bits, basado también en tubos de vacío. Cabe mencionar que el grupo de Mathews era el único capaz de realizar esta conversión en esos años (Roads, 1980), así es que podría también decirse que estos fueron pioneros en la computación acústica.

<sup>10</sup> Escrita pero no ejecutada en una computadora *Illiad*.

<sup>11</sup> Estructura del tiempo, la división del sonido, la composición espacial multicapa y la igualdad de sonido y ruido, o mejor, de tono y ruido.



Figura 1-2 Ordenador IBM 704

(Fuente: <https://www.historyofinformation.com/image.php?id=7000>)

Music 1 fue el primero de una familia de programas usados para investigar la síntesis de sonido, conocida como Music-N, que estableció un modelo para programas como CSound, Pure Data y SuperCollider. En Music 3 (1960) Mathews introduce el concepto de la unidad generadora, que se convirtió, según Curtis Roads, en “uno de los avances conceptuales más importantes que hicieron posible *computer music*” (Roads, 1980, p. 16). En la entrevista de 1980 para *Computer Music Journal*, Mathews describe por qué este se convirtió en un concepto necesario:

Quería dar al músico un gran poder y generalidad en hacer sonidos musicales, pero al mismo tiempo quería un programa lo más simple posible; quería que la complejidad del programa variase con la complejidad del deseo de los músicos. Si el músico quisiera hacer algo simple, él o ella no debería hacer tanto para lograrlo. Si el músico quisiera hacer algo muy elaborado estaba la opción de trabajar más en lo elaborado. La única opción que ví fue no hacer los instrumentos yo mismo –no imponer mi gusto e ideas sobre instrumentos a los músicos– pero en vez, hacer un grupo de bloques bastante universales y darles a los músicos ambas, la tarea y la libertad de ponerlos juntos en sus instrumentos.

En su influyente trabajo de 1963 para la revista *Science*, Mathews describe la abstracción detrás de este concepto (Mathews, 1963, p. 554):

La contribución central de Bell Telephone Laboratories al computer music es un programa que compute los muchos samples de una nota desde los pocos parámetros que la caracterizan ... El programa representa un compromiso

entre un procedimiento general, a través del cual cualquier sonido podría ser producido, pero que podría requerir una excesiva cantidad de trabajo de parte del compositor, y un procedimiento muy simple, que podría limitar demasiado el rango de sonidos musicales obtenibles. En orden de dar al compositor flexibilidad entre estos dos extremos, el programa está dividido en dos partes. En la primera parte el compositor especifica, en lenguaje de computadora, las características de un grupo de instrumentos. La unidad de programación que representa el instrumento (o “unidad de instrumento”) puede ser tan simple o tan compleja como desee. Después prepara una partitura que consista de una lista de notas a tocar en las unidades-instrumento que ha creado ... Los bloques interconectados de programa que hacen a una unidad-instrumento son llamados unidades generadoras.

Actualmente, se habla simplemente que esta abstracción corresponde a la orquesta y la partitura (Keislar, 2011). La Figura 1-3 representa mediante unidades-instrumento la sección de una programación. Los generadores computando desde parámetros de notas (izquierda) y que se alojan en memoria. Esto generará luego samples de las formas de onda de presión sonora.

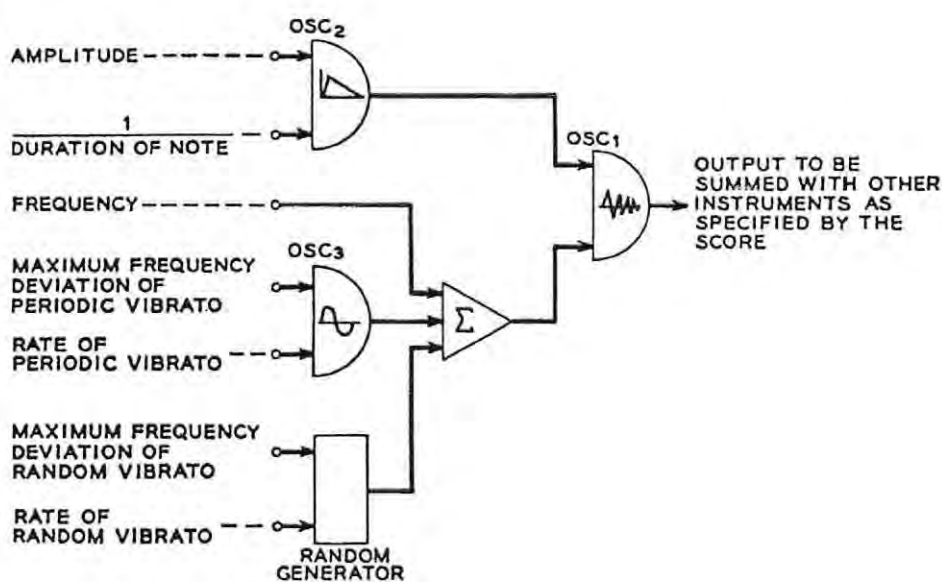


Figura 1-3 Diagrama esquemático de una típica unidad-instrumento en la orquesta de computadora.  
(Fuente: Mathews, 1963)

El siguiente apartado del capítulo tiene el propósito de complementar los contenidos anteriores y dar contexto al instrumento elegido para crear música, desde un génesis tecnológico a la razón del uso de las matemáticas en música.

## 1.4 Computación, formalización, Xenakis y el uso de matemáticas en la música

Durante el siglo XX, la construcción de máquinas capaces de automatizar cálculos tomó un gran interés. Francisco Varela (2005) relata algunos de los momentos históricos de las ciencias y tecnologías de la cognición (CTC). El movimiento cibernético, en su intención de crear una ciencia de la mente, anhelaba “expresar los procesos que subyacían a los fenómenos mentales en mecanismos explícitos y formalismos matemáticos” (p. 32). John Von Neumann, representó las neuronas de McCulloch-Pitts, usando tubos de vacío para configurar lo que hoy se conoce como arquitectura o modelo de computadoras Von Neumann: CPU, memoria y ALU, conectadas por bus (ver Figura 1-4).

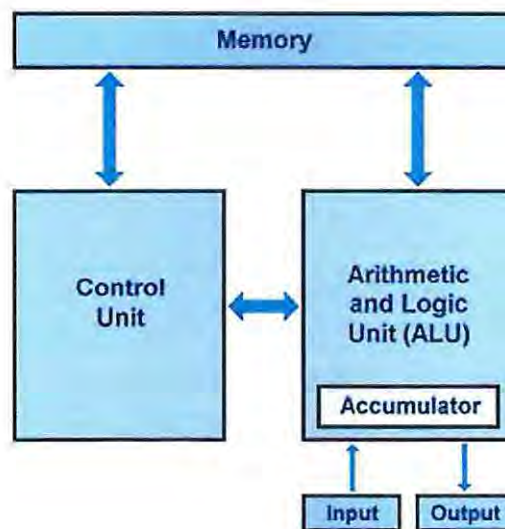


Figura 1-4 Arquitectura Von Neumann.

(Fuente: <https://nclab.com/john-von-neumann-and-modern-computer-architecture/>)

Según Mario Bunge, la filosofía formalista de las matemáticas corresponde a “la concepción según la cual la matemática es un sistema de signos o inscripciones, como los numerales, que no representan nada excepto, quizás, otros símbolos” (Bunge, 2005, p. 88). Esto es relevante en cuanto a los caminos que adoptaron algunos músicos en el siglo XX.

Iannis Xenakis (1963), quién llamó Música Formalizada a una de sus obras teóricas, en la cual expone sus principios compositivos y filosóficos. En esta obra, el compositor explica su enfoque compositivo estocástico que le sirvió como “marco lógico y método para resolver los conflictos y nudos que se encuentren” (p. 5). Respecto a esto, el compositor argumenta:

... Desde la antigüedad, los conceptos de posibilidad (tyche), desorden (ataxia), y desorganización, han sido considerados opuestos y negación de la razón (logos), orden (taxis), y organización (systasis). Es solo recientemente que el conocimiento ha podido penetrar la posibilidad y ha descubierto cómo separar sus grados –en otras palabras, racionalizarla progresivamente, sin lograr, sin embargo, una explicación definitiva y total del problema de la “posibilidad pura”<sup>12</sup> (p. 4).

... Es entonces no sorprendente que la presencia o ausencia del principio de causalidad<sup>13</sup>, primero en la filosofía y luego en las ciencias, puede influenciar la composición musical. Esto causa que se siga caminos que parecen divergentes, pero que, de hecho, se mezclen en teoría de probabilidad y finalmente en lógica polivalente<sup>14</sup>, que son especies de generalización y enriquecimiento del principio de causalidad. La explicación del mundo, y consecuentemente del fenómeno sónico que nos rodea o que puede ser creado, necesitado y aprovechado por la ampliación del principio de causalidad, la base de esta ampliación, es formada por la ley de los grandes números. Esta ley implica la evolución asintótica hacia un estado estable, hacia una especie de meta, de stochos, de donde viene el adjetivo “estocástica” (ídem).

Xennakis explica (p.8) cómo para 1954, a tres años de que Mathews desarrollara la primera versión de *Music*, la música devenida de los métodos compositivos de Arnold Schönberg, que podríamos también llamar tradicionales a ese tiempo, decaía porque ya no se podía lidiar con su propia complejidad. Las densidades polifónicas, tratadas de manera lineal, eran tan altas que se producía un “sinsentido auditivo e ideológico” (ídem). Es entonces que para escapar de la categoría lineal del pensamiento musical introduce la noción de probabilidad.

## 1.5 Computación y composición

F.R. Moore (1990), argumenta en su trabajo que “la principal razón para usar computadoras para mediar en el acto de composición es para aumentar las maneras en cómo las posibilidades musicales pueden ser exploradas” (p. 398). El esquema de la Figura 1-5 ayuda a visualizar esto último, donde las flechas indican el flujo de información. Este cuadro ha sido ocupado de manera arbitraria bajo lo que hemos considerado ha sido la participación de la computadora en la labor de práctica creativa.

---

<sup>12</sup> Aquí el autor ocupa la expresión “pure chance”.

<sup>13</sup> Xenakis usa una expresión de Platón para presentar este principio de la obra *Timeo*: “for it is impossible for anything, to come into being without cause” (Xenakis, 1963, p.1).

<sup>14</sup> La cual establece más de dos valores de verdad.

Como se puede observar, el esquema incluye tanto las rutas que usan tecnologías tradicionales como las electrónicas.

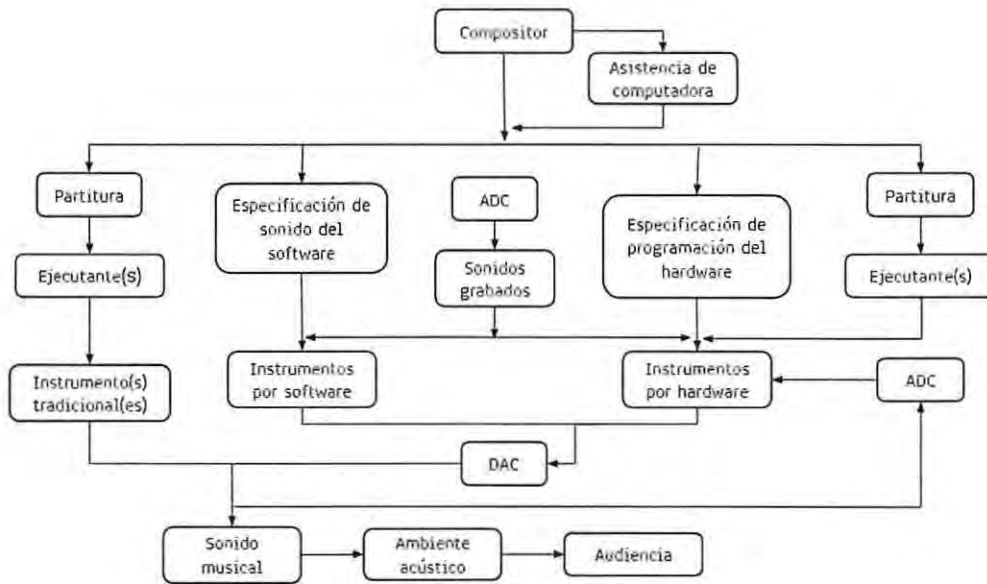


Figura 1-5 Composición mediada por computadora. (Trad.)  
 (Fuente: Elements of Computer Music, F.R. Moore, 1990, p. 399)

Actualmente, las maneras en que las computadoras participan en el trabajo creativo aún pueden ser explicadas teniendo en cuenta el esquema de Moore, incluso las estrategias que involucran algún tipo de automatización en alguna de las etapas, esto al indicar la posibilidad de la asistencia de la máquina.

El siguiente capítulo trata en breve el uso de la algoritmia en la música y algunos ejemplos de géneros musicales resultantes de su cruce con las herramientas de *computer music*.

## 2. Composición algorítmica y *computer music*

Cuando se elige trabajar dentro del paradigma de *computer music* también se acotan las estrategias de composición, más apropiadas a la naturaleza del “nuevo medio liberador –el electrónico–” (Varèse y Wen-chung, 1966, p. 15) como son las algorítmicas.

Pese a que la composición algorítmica predata a *computer music*, en este trabajo hemos priorizado los contextos que los relacionan directamente de una u otra forma. La algoritmia no computacional también está fuera del margen de esta investigación.

### 2.1 Definiciones, distinciones y antecedentes artísticos

Las definiciones y analogías del término algoritmo son bastante claras y pueden ser encontradas en varias fuentes. Aquí lo entenderemos en su forma más simple: una serie de instrucciones finitas. Sin embargo, composición algorítmica es un término bastante amplio, respecto a esto, Alex McLean y Roger Dean (2018, p. 4) afirman:

Esto es porque el rango de estructuras que pueden ser consideradas algorítmicas es extremadamente amplio, abarcando todos los programas de computadora (o según algunas definiciones, todos los programas de computadora que eventualmente terminan), y tal vez todas las partituras musicales.

El químico y músico, Lejaren Hiller (1981), realiza una revisión de sus obras y esquematiza algunas estrategias ocupadas en ellas. En su trabajo describe al computador como una “máquina de composición, con facilidades para algoritmos de composición, conversión análoga a digital de sonidos naturales, algoritmos de sintetización de sonidos, lenguajes de interpretación de partituras de alto nivel, y conversión digital a análoga de sonido” (Hiller, 1981, p. 8).

David Cope (Nierhaus (Ed.), 2015), afirma que todos los compositores usan algoritmos, a sabiendas o no, incluso si lo niegan. Con instrucciones que pueden ser “altamente restrictivas, vagas o en algún lugar entre estos extremos” (p. 405). Quién trabaja en plataformas computacionales no escapa a esta realidad y quizás es donde es más crítico aceptarla y entenderla, esto para tener todos los sentidos calibrados a la hora de construir algoritmos, parametrizar entradas a los programas y evaluar para obtener resultados. Cope llama a este individuo “el algoritmo más interesante de todos: el compositor humano” (p. 414).

Basándonos en lo afirmado por Cope podríamos decir que todos los compositores son compositores algorítmicos y que es algo inevitable. Pero no es propósito de este trabajo destrabar tal afirmación. Lo que sí es un hecho es que hay quiénes lo aceptan y trabajan en la exploración musical algorítmica con las herramientas de *computer music*. Dos de las categorías que han nacido de esta exploración son las que hablaremos a continuación y que se enmarcan en la utilización de tales herramientas en performances en vivo.

## 2.2 Live Coding y música interactiva, nuevas estéticas de *computer music*

Los sistemas en tiempo real permitieron la proliferación de prácticas de programación algorítmica en vivo en el contexto de *computer music*, lo que se conoce como live coding, “una práctica de improvisación donde la ejecución instrumental es basada en escribir/editar código en tiempo real para controlar procesos de generación de sonido” (Valle, 2016). Eduardo Ledesma (2015, p. 91) escribe:

Código es el mecanismo invisible y virtual que subyace al software, las instrucciones que lo ejecutan. En ciertos casos el código aflora a la superficie y se vuelve el objeto de la representación, un subgénero ubicado en la intersección entre la tecnología digital y la poesía experimental.

Esta definición retórica nos evoca a los procedimientos entre las capas de lenguajes de la computadora, en las que desde la escritura de instrucciones en lenguaje de alto nivel llega hasta las capas más bajas que son el ensamblador y los opcode, y por último el lenguaje máquina, grupo de cifras binarias traducidas a hexadecimales reunidos, el verdadero idioma del ente digital. Estos son soterrados por el software con el que el usuario interactúa gráficamente (GUI), en un estado de representación de sí mismo, y que, en ciertos casos—como en un error fatal de sistema por ejemplo— podemos ver el verdadero código con nuestros ojos. En esto, la unidad central de procesamiento (CPU) se convierte en un intérprete de ceros y unos, un sistema formal automático (AFS) (Haugeland, 1981) que permite a los programadores crear mundos conceptuales que obedecen leyes independientes a las plataformas donde se construyen (Sorensen et al, 2014).

Destacamos las palabras de un miembro de Toplap, en el documental on-the-fly, quién expresa: “Aquí el código puede funcionar de manera correcta o incorrecta, donde correcto es si es que te gusta del resultado e incorrecto si es que no.” (Canal Eulerroom, 2022, 12m58s). Toplap es una organización fundada en 2004, que promueve la práctica

del live coding así como las diversas plataformas que se distribuyen en la red social para programadores github.

En la práctica y performance de live coding colectivo, el código es proyectado para la vista del público mientras se va escribiendo/editando, pareciéndose a las fiestas electrónicas llamadas raves, por lo cual se ha denominado “algoraves” a estos encuentros. Estos llevan 10 años realizándose en diversos lugares del mundo y a través de sitios de stream por internet (ver Figura 2-1). A parte del código, en estas fiestas se suele hacer una combinación con visuales, que se pueden producir con las mismas plataformas donde se genera la música u otras que sean compatibles con el tipo de dato que genere la plataforma en que se esté generando el sonido.

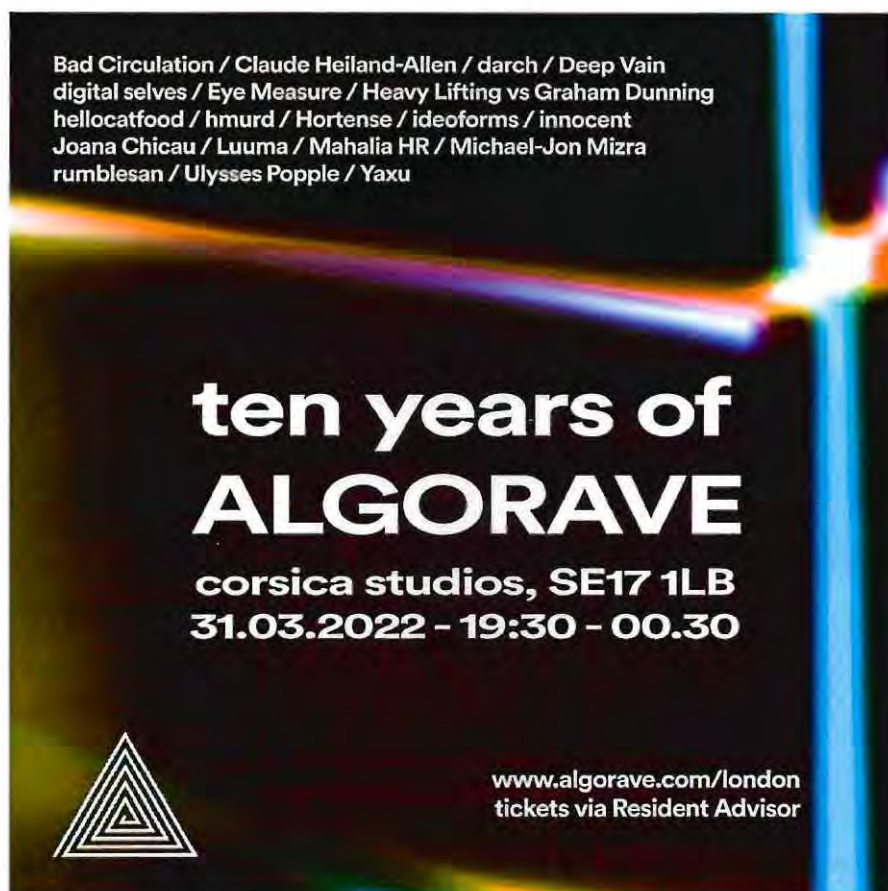


Figura 2-1 Poster décimo aniversario Algorave.  
(Fuente: <https://www.facebook.com/algoravers/photos>)

La música interactiva es considerada un subgénero de *computer music*, donde los algoritmos son diseñados para recibir entradas de sistemas orientados a la performance (Garnett, 2001). Dentro de este género se pueden encontrar trabajos donde “el ejecutante de alguna manera controle la electrónica o la electrónica afecte el sonido del ejecutante” (idem, p. 21).

El año 2021 se realizaron tres obras en el contexto de colaboración del centro de estudios musicales latinoamericanos (CEMLA) y la Universidad católica de

Valparaíso. Las obras fueron hechas por tres grupos de músicos nacionales (ver Figura 2-2) que fueron convocados para realizar piezas donde se ocupasen sensores de tela de fácil construcción en dicha Universidad (Pizarro et al, 2018). Los registros audiovisuales de estas obras están disponibles en el canal de CEMLA en el sitio web YouTube.



Figura 2-2 Imágenes desde registros fotográficos; desde arriba hacia abajo: “Strings” (Andres Rivera, Camilo Lillo); “Sillón Sonoro” (Georgia del Campo, José Candela); “Rueda de Canto” (Felipe Barrera, Oscar Santis).  
(Fuente: Registro fotográfico de Karla Seemann)

Dentro de lo que se puede considerar *computer music* hay otras expresiones, como la música de video juegos y las generadas con algoritmos generativos, que son de las más antiguas, sin embargo aquí no las profundizaremos.

En el siguiente capítulo se realiza una breve y general descripción de SuperCollider, la plataforma que se eligió para la creación de música de género *computer music* mediante codificación (coding), es decir, la descripción de algoritmos para la realización de tareas que realizará la computadora.

### 3. SuperCollider: una plataforma para la composición algorítmica

A través de los años se han desarrollado múltiples lenguajes para la práctica de *computer music*, como muestra la Figura 4-1. El constante desarrollo y mejoramiento de la tecnología computacional permitieron la sintetización y procesamiento en tiempo real a través de computadores, donde a través de una algoritmia, las diferentes piezas de hardware se interconectan en un flujo de datos digitales para producir un resultado sonoro. Es así que “el ingeniero de software, que desarrolla herramientas para la síntesis de sonido, puede ser considerado una abstracción del constructor tradicional de instrumentos. Un compositor que asume este rol, desintermedia a ese ingeniero” (Keislar, 2011, p. 25).

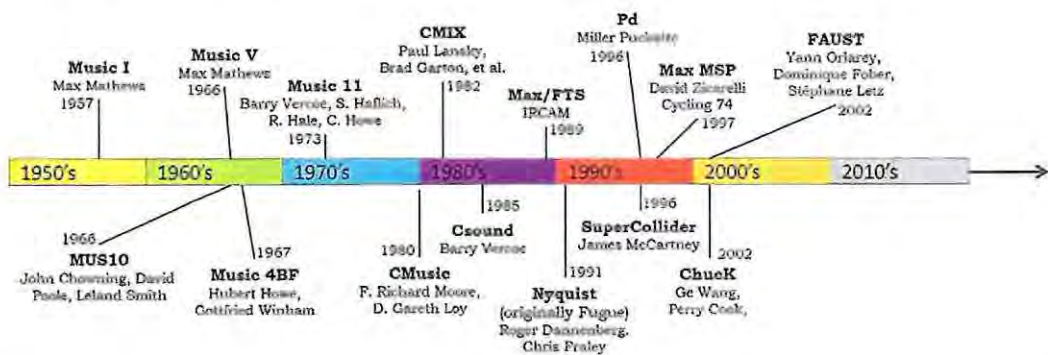


Figura 3-1 Línea de tiempo de lenguajes representativos e históricamente significativos para computer music.

(Fuente: Dannenberg, 2018, p. 1)

James McCartney (Wilson et al (Eds.), 2011, p. xi), creador de SuperCollider, explica:

La principal idea de SuperCollider es componer algorítmicamente objetos para crear procesos generadores de sonidos. Las Unidades Generadoras, un concepto inventado por Max Mathews para sus lenguajes Music N, son muy parecidos a objetos, y son naturalmente apropiados para un lenguaje basado en Smalltalk puramente orientado a objetos.

En el estilo orientado a objetos, la abstracción es uno de sus cuatro principios, Microsoft (2022) define a este principio como la “modelación de los atributos relevantes e interacciones de entidades como clases para definir una representación abstracta de un sistema”. En otras palabras, las clases son descripciones de ideas hechas código, luego podemos crear objetos que estén basados en estas. En el trabajo artístico, pero también en contextos científicos, usualmente los conceptos son inicialmente poco especificados, si es que no, mal definidos, y uno refina ideas iterativamente (Wilson et

al (Eds.), 2011). En SuperCollider, así como otros lenguajes de computadora modernos, existen clases y objetos que están predefinidos dentro del código fuente<sup>15</sup>, programados por los desarrolladores. Estos se compilan (cargan en memoria RAM) al iniciar la aplicación, sin embargo los usuarios pueden alterar o crear sus propias abstracciones.

### 3.1 Aplicación SC, estructura y funcionamiento

La Figura 4-2 muestra la actual estructura de la aplicación, usada en la documentación online oficial del software (<https://doc.sccode.org/>), donde se indican:

1. Un servidor de audio (*scsynth* o *supernova*), escrito en C++.
2. Un lenguaje de programación para audio, basado en SmallTalk.
3. Un intérprete para el lenguaje (*sclang*).
4. El programa intérprete como cliente para el servidor.
5. La aplicación, incluyendo los dos programas y funcionalidades mencionadas.

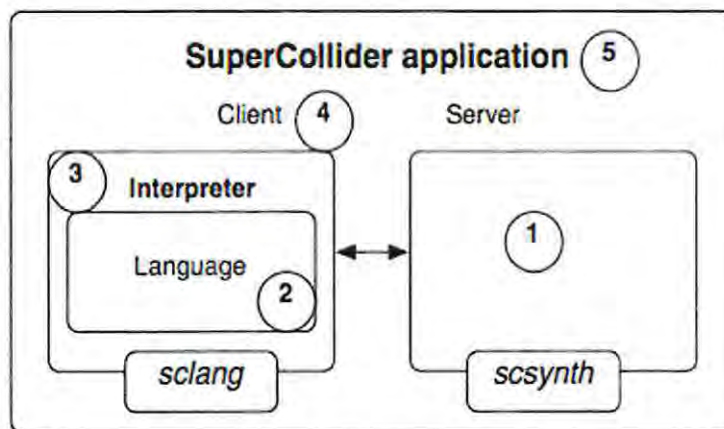


Figura 3-2 Estructura de la aplicación SuperCollider.  
(Fuente: <https://doc.sccode.org/Guides/ClientVsServer.html>)

SuperCollider es un software basado en texto, con una arquitectura cliente-servidor comunicada a través del protocolo de comunicación digital Open Sound Control (OSC). Este es “un protocolo abierto de eventos discretos, transporte independiente (no importa el nivel<sup>16</sup> de red del mecanismo que envía paquetes), basado en mensajes, desarrollado para comunicación entre computadores, sintetizadores de sonido, y otros equipos multimedia” (Henríquez, 2022, p. 25). Una forma de entender la relación de trabajo de la estructura de SC, es la de una cadena de diseño, creación y ejecución de un instrumento musical. Un ejemplo simple de esto es evaluar un oscilador sinusoidal

<sup>15</sup> Ver Tabla 1 (McCartney, 2002, p. 62).

<sup>16</sup> Capas del modelo abierto de sistema de interconexión, OSI.

usando `{SinOsc.ar}.play`, configurado de forma predeterminada<sup>17</sup>. Al ejecutar el código, este es interpretado hacia el servidor como un mensaje OSC, como se vé en la Figura 4-3.

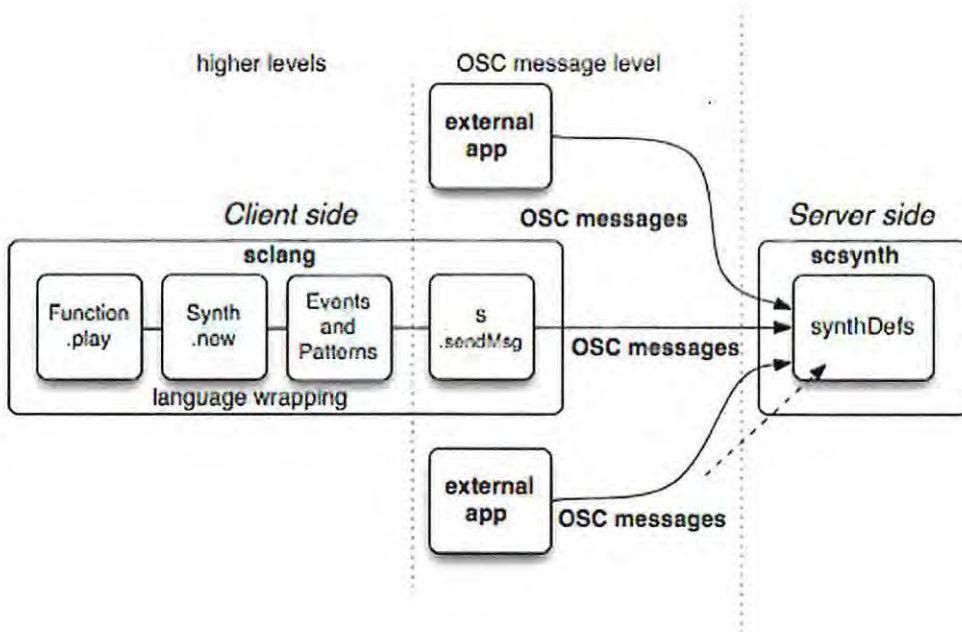


Figura 3-3 Slang como cliente de alto nivel.  
(Fuente: <https://doc.sccode.org/Guides/ClientVsServer.html>)

Una vez que el sintetizador está en el servidor, este se encargará de sintetizar y procesar el sonido, esto lo hace evaluando un árbol de nodos alojados dinámicamente. El método de clase “.ar” (audio rate) quiere decir que para el objeto SinOsc que está siendo usado, se computará un bloque completo de valores (64 samples por bloque) (Wilson et al, 2011). La Figura 4-4 muestra el árbol de nodos generado al ejecutar `{SinOsc.ar}.play`, donde el instrumento está representado por el nodo 1006.

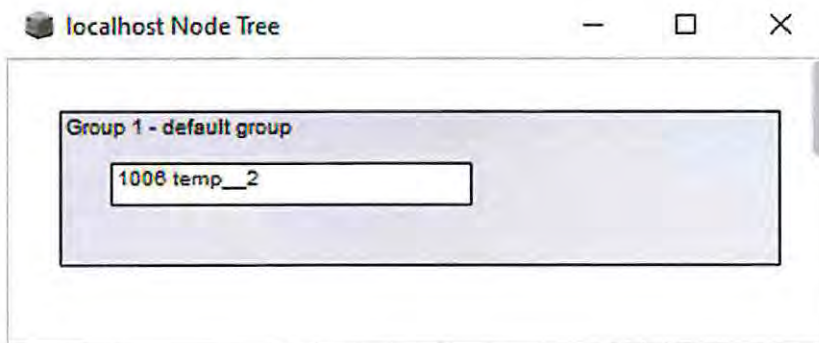


Figura 3-4 Árbol de nodos de `{SinOsc.ar}.play`.

El resultado sonoro puede observarse en la Figura 4-5, una oscilación de 440 Hertz solamente en el canal izquierdo. SuperCollider reconoce los dos canales de salida

<sup>17</sup> Frecuencia: 440 Hertz; sin desfase; factor de amplificación  $\times 1.0$  a la salida; sin valor agregado a la salida.

### 3. SuperCollider: Una plataforma para la composición algorítmica

básicos, 0 (izquierda) y 1 (derecha), esto si es que no se tiene conectado un sistema con más salidas. Este resultado se puede obtener fácilmente al aplicar el método `.scope(#canales)` directamente en el código, que abre un osciloscopio<sup>18</sup> integrado en SC (ver Figura 4-5a). También puede realizarse un ploteo en sistema cartesiano con el tiempo en el eje de abscisas y el número generado a la salida en el eje de las ordenadas usando el método `.plot(#segundos)` (Figura 4-5b).

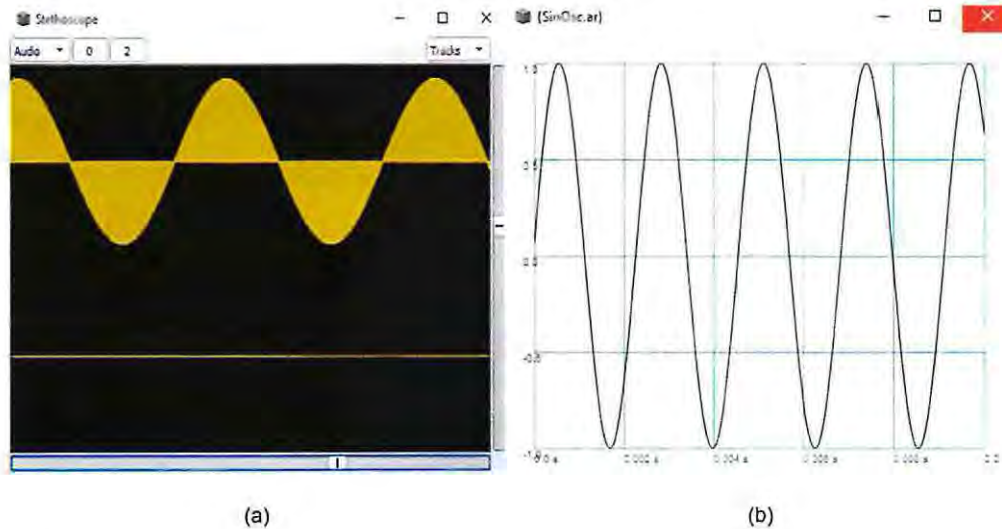


Figura 3-5 Resultados gráficos de `{SinOsc.ar}`: `.scope(2)` a la izquierda y `.plot(0.01)` a la derecha.

Por último, Andrea Valle (2016) usa un sexto elemento en el esquema de la Figura 4-1, conectado en doble sentido al intérprete, el IDE, el ambiente de desarrollo integrado. Este es “un editor donde el código puede ser escrito/editado y que le da código al intérprete para evaluación” (p. 5). La Figura 4-6 muestra este elemento, a la izquierda está el campo para escribir; en la parte superior derecha se encuentra la ventana de ayuda integrada; y en la parte inferior derecha se encuentra una terminal donde se pueden monitorear salidas o respuestas del sistema. Por último, debajo de la terminal, hay un monitor que permite ver: el estado del intérprete, el rendimiento del equipo, la cantidad de unidades generadoras y sintetizadores en funcionamiento, y el valor en decibeles del sonido ejecutado.

<sup>18</sup> El programa lo referirá como estetoscopio.



Figura 3-6 SuperCollider IDE v3.12.2

## 3.2 Lenguaje de programación SC

En este apartado veremos algunos aspectos del lenguaje SuperCollider, el uso de las unidades generadoras en este contexto, que sirven para plantear el diseño de sintetizadores y funciones que pueden servir al compositor humano.

SuperCollider es un lenguaje de programación interpretado que ha sido hecho pensando en la sintetización digital de sonido en tiempo real. Esto desde el lado del cliente, ya que por el lado del servidor hay una compilación previa, que ejecuta las instrucciones a más bajo nivel, usando el driver de audio del sistema operativo para transformar los números generados en oscilaciones del campo magnético de los altavoces que se estén ocupando (ver Figura 4-7), en Windows, WASAPI es el driver de baja latencia integrado que se ocupa desde las versiones Vista o superior.

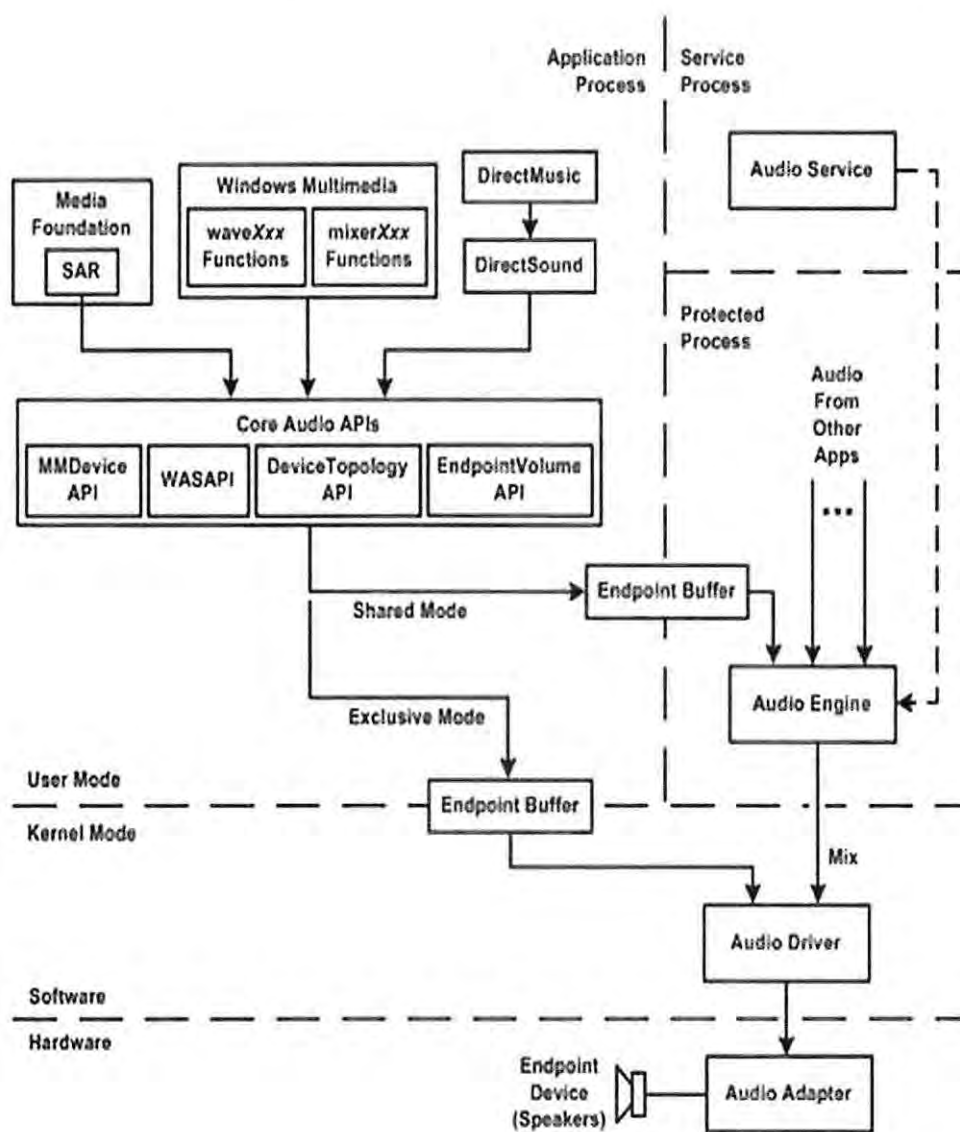


Figura 3-7 Diagrama de audio en Windows.  
 (Fuente: <https://www.thewelltemperedcomputer.com/KB/WASAPI.htm>)

En la ejecución del código y gracias a los desarrolladores de SC, el compositor humano se puede despreocupar en gran medida de lo expuesto y solamente “tocar” el instrumento musical computador. James McCartney (2002, p. 61) explica:

Un lenguaje de computadora presenta un modelo abstracto de computación que permite a uno escribir un programa sin preocuparse de detalles que no son relevantes al dominio del problema del programa. Mientras mayor sea el poder de abstracción del lenguaje, más podrá el programador enfocarse en el problema y menos en las restricciones y límites de las abstracciones de lenguaje y en el hardware del computador.

Algunos autores ofrecen opciones de representación gráfica de las unidades generadoras. La Figura 3-8 muestra la forma de diagramar de David Cottle (2005), donde el primer esquema corresponde a un oscilador con una frecuencia de 400 Hertz<sup>19</sup>.

<sup>19</sup> Esto es incorrecto, ya que el valor predefinido sería el expuesto en el apartado 4.1 (440 Hertz).

El segundo y el tercero representan dos osciladores interconectados, donde el superior controla la frecuencia del inferior. Para este caso, la diferencia entre ocupar “freq” y “freq:”, es que en el primero se está creando una variable local<sup>20</sup>, mientras que el segundo refiere a la variable predefinida, esto por que en SC se puede optar qué parámetro va a ser alterado y cual no. El caso {SinOsc.ar}.play, puede ser reescrito como {SinOsc.ar(mul: 0.5)}.play, y al ejecutarlo, la salida del SinOsc, ya conocida en 4.1, tendrá la mitad de la intensidad original.

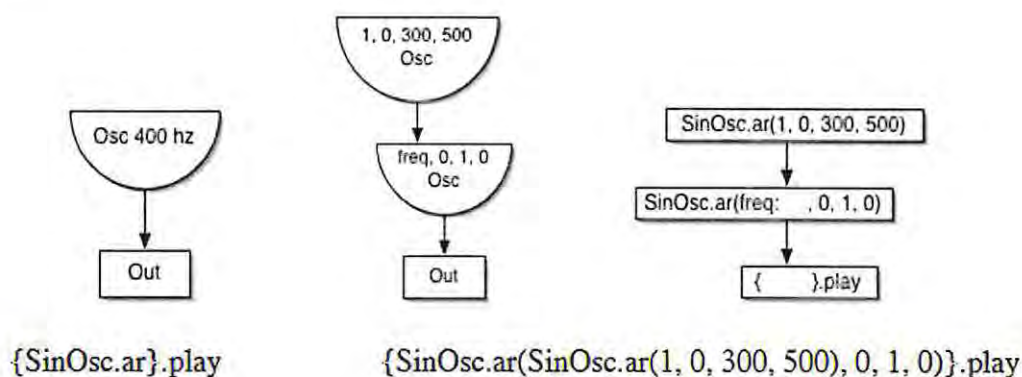


Figura 3-8 Diagramas de para la representación de sintetizadores.  
(Fuente: Cottle, 2005, p. 93)

En la Figura 3-9 (izquierda) se muestran dos versiones de código donde se define un sintetizador llamado `\pulseSine`, una versión por defecto y otra con argumentos personalizados, donde separa las variables de pulso y senoide; a la derecha se encuentra un grafo del sintetizador. El resultado sonoro es que la senoide con frecuencia 250 Hertz aparece y desaparece con una frecuencia de 5 Hz. Como el argumento `out` es 0, solo se escucha por el lado izquierdo. Por último, una razón por la cual el argumento `kfreq` se define explícitamente es que luego puede ser seteada desde el exterior de la definición del sintetizador usando patrones.

<sup>20</sup> Que tiene efecto en un ámbito local, en este caso la función definida entre los paréntesis del tipo llave: {}.

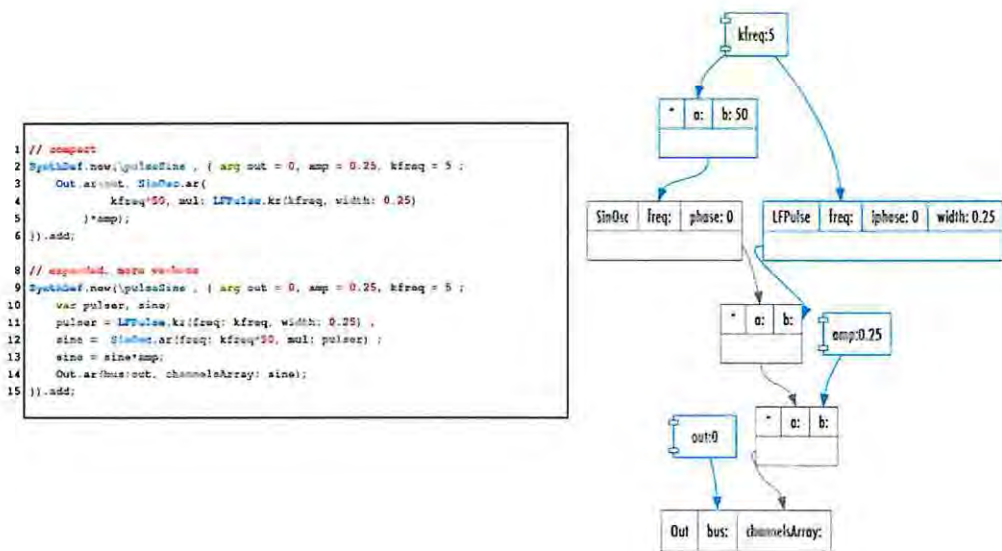


Figura 3-9 Códigos y grafo para el sintetizador pulseSine.  
(Fuente: Valle, 2016, pp. 142-143)

Los anexos al final de este documento corresponden a una colección de sintetizadores que se pueden evaluar para tocar patrones y una propuesta para configurar los patrones, si bien hay muchas formas de tocar con SuperCollider, acá hemos elegido crear un espacio proxy para la automatización de la creación de grupos, para que el árbol de nodos se genere de forma ordenada. También, para ayudar a la documentación de este informe se trabajó con lo que se denomina un *Quark*, una extensión para SC que puede ser descargado ya sea desde el manejador de paquetes integrado en el IDE de SC (preferentemente, ya que la instalación se hace de forma automática) o el repositorio de GitHub: <https://github.com/supercollider-quarks/Spectrogram>.

Nick Collins (2016) realiza una clasificación donde relaciona la terminología de ciencia de computadores, la sintaxis de SC y musical (ver Tabla 3-1). Esta propuesta de organización sirve como un insumo para la esquematización de conceptos más generales, y también nos da un primer vistazo al acotamiento de estrategias de composición. Aprovecharemos también esta oportunidad para mencionar una dificultad (y peligro) que se presenta pero que no se puede evitar: la traducción. En este documento, hemos dejado todas las palabras de las construcciones en su idioma original, ya que estas son propias del lenguaje SC, excepto en el término “instancias” de la Tabla 3-1, que no existe como estaba dispuesto en el esquema original. Este concepto corresponde a copias de objetos dentro del paradigma de programación.

Tabla 3-1 Correspondencias entre programación y música respecto a código SC.  
(Fuente: Collins, 2016, p. 10)

Ciencia de computadores	Construcción en SC	Posible uso musical
Iteración	Pseq, Routine, {}.fork, Dseq, Stepper, Select, do...	Secuencia de eventos, generación/manipulación de objetos musicales
Condicional	if, switch, Index...	Elección entre opciones musicales
OOP (Programación orientada a objetos)	SynthDef/Synth, Class/instancias	Granos dentro de una síntesis granular/ instrumentos de una orquesta virtual/ tipos de sonido en una pieza
Estado, variables	var, Integer, Float, Array...	Datos musicales, valor(es) de parámetros musicales
Generación de números aleatorios, probabilidad	Rand, rrand, choose...	Música estocástica, generatividad
Periféricos	MouseX, KeyState, SerialPort, Arduino...	Interfaces para música
GUI (Interfaz gráfica de usuario)	Windows, Slider...	Parches para ejecución

Actualmente SC ha sido usado exitosamente para presentaciones en vivo, diseño de GUIs, música orientada a baile, composiciones de música electroacústica, espacialización de sonido, interacción de audio y hardware, sonificación científica, aplicaciones basadas en multimedia o redes, performances de sistemas distribuidos, dentro de otras (Valle, 2016).

A continuación se presenta el último capítulo de este trabajo de investigación, que corresponde a la revisión de las prácticas creativas que se mantuvieron hasta la conclusión de este documento

## 4. Creación de piezas musicales para computer music en SuperCollider

En este capítulo nos enfocaremos en explicar los procesos creativos que se llevaron a cabo mediante las herramientas conceptuales vistas anteriormente. Para documentar la labor y práctica creativa que dé como resultado piezas musicales en los márgenes delimitados, esto es, computer music y composición algorítmica, se ha elegido trabajar basándonos en el esquema de bucle de interacción y retroalimentación entre práctica creativa y reflexión de Rubén López Cano y Úrsula San Cristóbal, que se muestra a continuación en la Fig. 4-1.



Figura 4-1 Bucle de interacción y retroalimentación entre práctica creativa y reflexión.  
(Fuente: López; Opazo, 2014, p. 169)

Este recurso metodológico autoetnográfico permite “convertir algunas actividades características de la práctica artística en tareas de investigación formalizadas” (idem, p. 168). Para esta investigación, se han tenido dos prácticas en que se ha utilizado: audiciones musicales e improvisaciones libres de live coding. Este acercamiento nos permite desarrollar un trabajo creativo sin entrar en la categoría de composición propiamente.

## 4.1 Palabras preliminares

Como pudo observarse en el primer capítulo, al hablar de *computer music* y explicar lo que es, resulta inevitable contextualizar desde el cruce arte-ciencia. En este apartado dedicaremos algunas reflexiones previas a la exposición de las prácticas de creación.

### 4.1.1 La labor creativa en los albores de la cuarta revolución industrial

Para ayudar a comprender los trabajos creativos en nuestra época, es necesario observar nuestro contexto actual, las actividades que realizamos cotidianamente y los artefactos con los que nos relacionamos. Los computadores han ido acelerando y generando nuevos procesos humanos desde su aparición, convirtiéndose en una constante de casi todas las culturas que han sobrevivido hasta este momento histórico. En este mismo tren de pensamiento, muchos de los paradigmas previos se han tenido que alterar o acomodar frente a este nuevo elemento.

A nivel geopolítico, en años recientes, muchos países se han dado cuenta de su dependencia de los sistemas computacionales. Los Estados Unidos de América, quiénes tuvieron durante el siglo XX un importante papel en este ámbito, pero que con los años delegan la mayor parte de la fabricación de semiconductores a países dispuestos a aprender sacrificando el costo medioambiental que eso conlleva, fueron perdiendo terreno técnico frente a China, teniendo como consecuencia directa que el principal proveedor de microchips de alta gama, la compañía manufacturera de semiconductores de Taiwán (TSMC), fue foco de creciente atención debido a un posible conflicto militar que podría poner en riesgo un equilibrio de poder obtenido en base a la tecnología. Este nuevo panorama lleva al país norteamericano a tomar la decisión de retomar el camino de la fabricación nuevamente<sup>21</sup>.

Reiteramos que la comprensión del artefacto computador requiere la observación de su contexto. Jean Claude Risset (2004), quién fue uno de los primeros músicos en sumarse a los laboratorio Bell, escribe: "Los experimentos de *computer music* pueden ayudar a demostrar que la computadora puede ser armoniosa, y no necesita ser deshumanizante – ya que con mucha frecuencia es usada como excusa para prácticas inhumanas" (p. 30).

### 4.1.2 De la irracionalidad, la racionalidad y la naturaleza humana

En la sección 1.4 se mencionó brevemente a John von Neumann, inventor de la primera arquitectura de computadoras digitales. Este matemático se caracterizó por ser un

---

<sup>21</sup> <https://www.forbes.com/sites/brianbushard/2022/07/28/chips-act-passes-house-approves-280-billion-bill-to-boost-microchip-production-and-counter-china/?sh=5805622bfda6>

individuo extremadamente lógico, a pesar de esto, rápidamente entendió y aceptó la conclusión del trabajo de Kurt Gödel en su teorema de incompletitud de las matemáticas<sup>22</sup>, que acabó con la idea de la universalidad de las matemáticas. En paralelo, Herbert Wiener, otro matemático, contemporáneo de von Neumann y quién acuñara el término cibernética (como se verá más adelante), vivió esta crisis a su propio modo. Wiener, que era altamente intuitivo, aceptó la imprecisión (matemática) como la naturaleza de las cosas (Heims, 1982, p. 156). En su autobiografía escribe:

La declaración de nuestra propia naturaleza y el intento de levantar un enclave de organización frente a la avasalladora tendencia de la naturaleza al desorden es una insolencia a los dioses y la necesidad de hierro que imponen. Aquí yace la tragedia, pero aquí yace la gloria también. (Wiener, 1956, p. 325)

Hemos tomado dos ejemplos de estereotipos científicos, con personalidades y estilos notablemente diferentes, pero que pueden ayudar en la comprensión de quién elegiría estar en este cruce arte-ciencia y ejercer la labor de creación. En esta elección se tuvo en mente el mostrar un pensamiento no propiamente artístico, pero que ayude en una comprensión holística de quién esté en otra arista del esquema de la Figura 1-1. Entonces, habrá quienes tendrán una orientación Neumanniana, favoreciendo el dualismo de la lógica y el empirismo formal, y otros que tiendan a la síntesis conceptual de perspectiva Wieneriana (Heims, 1982, p. 158).

#### 4.1.3 Codificación / como ejecución / como creación / como relación

Para finalizar este apartado, damos una reflexión acerca de uno resultados característicos de las prácticas creativas que se presentan a continuación, que es el aspecto del lenguaje de programación. Esta representación/descripción de algoritmos nos permite relacionarnos con la máquina. El acercamiento “improvisacional” con este agente computacional en las prácticas creativas permite otra observación, más íntima, en la que puede revelarse la “similitud ontológica de las formas vivas y mediatizadas (tecnológicas)” (Brown, 2016).

## 4.2 Práctica creativa I: Registro de audiciones musicales

Para dar cuenta del universo auditivo al que hemos acudido para dar una estética a las piezas musicales resultantes acorde a nuestros objetivos, se ha confeccionado una hoja de registro para la práctica de audiciones musicales, como puede apreciarse en la Figura

---

<sup>22</sup> Gutiérrez, C. “Teorema de Incompletitud de Gödel” (Versión para no iniciados): <https://users.dcc.uchile.cl/~cgutierr/otros/godel.pdf>

4-2. Estas fueron llenadas a mano, para poder tener libertad de tomar notas en cualquier momento y lugar en que se esté portando una hoja.

Ficha para audiciones musicales	
Momento de registro	
Nombre	
Autor	
Disco	
Medio de distribución	
Publicación	
Fuente	

Descripción:

Impresiones:

Figura 4-2 Hoja para registro de audiciones.

La primera parte contiene una ficha con los datos de lo escuchado, estos permiten identificar la obra. Luego, el campo de Descripción tiene como finalidad tomar alguna información dada por el medio de distribución, o el autor, que pudiese ser de relevancia subjetiva para complementar lo anterior. Relevante es, por ejemplo, cuando se ha mencionado que las piezas fueron compuestas y ejecutadas en SuperCollider, como en el caso del trabajo de Mads Kjiëlgaard y LuisaMei.

El campo de Impresiones sirve para registrar cualquier aspecto de lo escuchado que se desee destacar de forma libre, ya sean palabras, símbolos o lo que sea necesario para esto. También se puede anotar ideas que sean evocadas al realizar la audición.

Los resultados de esta práctica, dentro del bucle de la Figura 4-1, generan objetos físicos que son reflexionados a posteriori. A continuación exponemos otro formato de ficha, con datos relevantes tomados de las prácticas de audiciones. Estas prácticas son

sometidas a un juicio estético musical y también reflexiones, teniendo en cuenta parámetros propios de categorías asociadas a computer music que podamos propiamente ocupar para este propósito. Desde la disciplina musical: altura, timbre, intensidad y tiempo; u otros más complejos, tales como ritmo, melodía, armonía, textura y espacialidad. En cuanto a parámetros extramusicales, algunos aspectos de interés serían por ejemplo la programación, el software o el hardware usado, entre otros, incluso podrían mencionarse aspectos históricos, narrativos y conceptuales. Para esto, se rescatan anotaciones hechas en el campo de impresiones de las hojas de registro de audiciones. No todas las pistas de las obras/discos fueron incluidas en el juicio adjunto, se prefirieron las piezas que llamaban la atención en aspectos que pudiesen enriquecer de alguna manera los resultados de este trabajo.

Autor	Mads Kjeldgaard
Disco/Obra	Hold Time
Medio de distribución	Band Camp
Publicación	10/23/2020
Fuente	<a href="https://mads-kjeldgaard.bandcamp.com/album/hold-time">https://mads-kjeldgaard.bandcamp.com/album/hold-time</a>

Obra generada mediante SuperCollider. En la pista 1, ((())), llama la atención el desarrollo de los patrones polirrítmicos, en su combinación y el uso del control del tiempo, a veces afectando el cuerpo en ciertos momentos, provocando cierto nivel de ansiedad. Esto podría lograrse mediante la alteración del tiempo en SC, mediante el método *.tempo* del UGen TempoClock. Esto es acompañado con pads largos que generan un equilibrio con lo anterior. En cierto momento se siente lo que parece una imitación al palpar del corazón, en baja intensidad sonora. Ya al final de la pieza una melodía que parece desarmada de corto ataque y bajo sostenimiento.

Autor	Ben Peers
Disco/Obra	Patterns generated by recursion
Medio de distribución	Band Camp
Publicación	14/1/2022

Fuente	<a href="https://ellirecords.bandcamp.com/album/patterns-generated-by-recursion">https://ellirecords.bandcamp.com/album/patterns-generated-by-recursion</a>
--------	---

Disco hecho con el software Max/MSP, usando algoritmos generativos. Las texturas logradas con variedad de timbres y efectos, destacando la reverberancia. Llama la atención rítmicas generadas con beats aleatorios de baja frecuencia y juegos de paneos. También los cambios de tempo, esto puede ser logrado también en SC, mediante el método *.tempo* del UGen TempoClock.

Autor	Scott Wilson (SC), Xenia Pestova (Piano)
Disco/Obra	On the Impossibility of Reflection
Medio de distribución	Soundcloud, Youtube
Publicación	1/1/2011
Fuente	<a href="https://scottwilson.ca/music/">https://scottwilson.ca/music/</a> (versión estudio, Soundcloud) <a href="https://www.youtube.com/watch?v=d6iPWk_Upvo">https://www.youtube.com/watch?v=d6iPWk_Upvo</a> (ejecución en simposio de SuperCollider, 2012)

Esta es una pieza de piano de cola y SuperCollider, donde hay mayor presencia del piano y la computadora complementa con diversas sonoridades que se disparan durante la ejecución. En la versión de youtube se vé que la pianista está sola en el escenario, lo que puede indicar que es la pieza es de formato interactivo.

Autor	Ge Wang
Disco/Obra	Twilight (2013) for laptop orchestra
Medio de distribución	Youtube
Publicación	13/7/2014
Fuente	<a href="https://www.youtube.com/watch?v=chA-4GRcb-I&amp;t=24s">https://www.youtube.com/watch?v=chA-4GRcb-I&amp;t=24s</a>

#### 4. Creación de piezas musicales de computer music para SuperCollider

En este trabajo de música interactiva, llama la atención el uso del sistema de hardware hecho por Ge Wang, este permite la sincronización de quienes interactúan con el algoritmo, lo cual resulta en una performance gestual llamativa.

Autor	Kraftwerk
Disco/Obra	Computer World
Medio de distribución	Spotify App
Publicación	Remasterizado 2009 (original 1981)

El concepto de este trabajo tiene como foco los computadores. Destacamos la pieza Numbers, en la que con un fondo rítmico constante, con un timbre parecido a lo que se puede lograr con *Impulse.ar* en SC, se tocan voces contando números desde el uno al ocho en distintos idiomas. También, en el tema Computer Love, donde hace una referencia narrativa al desarrollo de una relación amorosa mediante el aparato, acompañado por varias melodías y una base rítmica con el sello de Karl Bartos.

Autor	Kraftwerk
Disco/Obra	Tour de France
Medio de distribución	Spotify App
Publicación	Remasterizado 2009 (original 1983)

En la versión remasterizada de 2009 escuchada, al iniciar la pieza Prologue, lo hace tocando una armonía con alto ataque y alto sostenimiento, suave delay, con un filtrado cortando hacia las frecuencias altas. El timbre de este sonido es bastante especial y se intentará replicarlo en SC. Este timbre de este sonido es muy llamativo, y es ocupado en esta versión remasterizada en todas las etapas del 1 al 3 del Tour de France.

Autor	William Fields
Disco/Obra	Performance de Live Coding en Algorithmic Art Assembly
Medio de distribución	Youtube

Publicación	10/3/2022
Fuente	<a href="https://www.youtube.com/watch?v=vkAcpToksrs">https://www.youtube.com/watch?v=vkAcpToksrs</a>

En esta ejecución se acompaña la música con visuales sincronizadas, al tener una duración de media hora se logra abarcar una amplia diversidad de patrones y sonidos, también cambios de tempo. Dentro de los patrones y sonidos generados aparecen los aproximados a un set de percusión, con gradualidad de artificialidad en sus timbres. Cuando es más artificial suelen aparecer granulaciones que distorsionan el sonido al ser ampliado generando un efecto robotizador.

Aproximadamente a los diecinueve y veintisiete minutos hay una pausa, esto también puede ser considerado como un recurso, aunque hay que tener en cuidado en su duración, para que la audiencia no pierda concentración en lo que se está experimentando, ya que bien puede considerarse como pausa o una finalización. En esta ejecución lo hace dos veces, con un cambio de patrones tanto en timbres y en visuales.

Autor	Varios
Disco/Obra	Music From Mathematics (1ra Ed, 2da Ed)
Medio de distribución	Youtube
Publicación	1960 (1ra Ed.), 1962 (2da Ed.)
Fuente	<a href="https://www.youtube.com/watch?v=6Ep6kXOha0U">https://www.youtube.com/watch?v=6Ep6kXOha0U</a> (1ra Ed.) <a href="https://www.youtube.com/watch?v=FgKKA9sdfs4">https://www.youtube.com/watch?v=FgKKA9sdfs4</a> (2da Ed.)

Estas compilaciones tienen las primeras piezas musicales hechas mediante computadoras, muchas de estas compuestas por miembros de los laboratorios Bell, como Max Mathews y John Pierce. La primera edición tiene un relato explicativo de cada pieza y también incluye un movimiento de la obra *Illiac Suite* para cuarteto de cuerdas de Lejaren Hiller, así como unas piezas para piano. En la segunda edición la gestualidad lograda mediante los medios electrónico-digital es notable, como se puede apreciar en la pieza de J. Pierce, *Molto Amoroso*.

Si bien actualmente se pueden obtener resultados más sofisticados de manera más simple, la importancia y valor de estas piezas de música corresponden a su lugar en la historia de computer music y a las ciencias de la computación e ingeniería.

Autor	Toplap Düsseldorf
Disco/Obra	_EXECUTE Vol.1 - A Compilation Of Contemporary Algorithmic Music
Medio de distribución	Bandcamp
Publicación	20/5/2022
Fuente	<a href="https://toplapduesseldorf.bandcamp.com/album/execute-vol-1-a-compilation-of-contemporary-algorithmic-music">https://toplapduesseldorf.bandcamp.com/album/execute-vol-1-a-compilation-of-contemporary-algorithmic-music</a>

En esta compilación hay música hecha con varios software para la composición algorítmica, entre ellos SC, aunque no hay información de la o las piezas en que este se usa. Destacamos un ejemplo, la pieza *Something is Bothering Me* de la artista portuguesa Earth to Abigail<sup>23</sup>, donde los sutiles sonidos generados, junto con los patrones utilizados, crean una textura equilibrada y clara. Este disco es un excelente compendio del *computer music* que se está manifestando actualmente.

Autor	John Chowning
Disco/Obra	Turenas · Stria · Phoné · Sabelithe (Music with Computers)
Medio de distribución	Youtube
Publicación	1988
Fuente	<a href="https://www.youtube.com/watch?v=mqIPWGwrpng">https://www.youtube.com/watch?v=mqIPWGwrpng</a> (Turenas) <a href="https://www.youtube.com/watch?v=ftU1v0bPRE4">https://www.youtube.com/watch?v=ftU1v0bPRE4</a> (Stria) <a href="https://www.youtube.com/watch?v=41PBrcIKbk8">https://www.youtube.com/watch?v=41PBrcIKbk8</a> (Phoné)

<sup>23</sup> <https://earthtoabigail.bandcamp.com>

#### 4. Creación de piezas musicales de computer music para SuperCollider

	<a href="https://www.youtube.com/watch?v=53FQCel ewDs">https://www.youtube.com/watch?v=53FQCel ewDs</a> (Sabelithe)
--	--

En este trabajo, llama la atención el uso de la espacialidad y los timbres logrados. Chowning es conocido como el desarrollador de los algoritmos para modulación de frecuencia, esto le dá un valor técnico e histórico a su música. Este músico también fue parte del equipo de Max Mathews en los laboratorios Bell, previo a que co-fundase el CCRMA, uno de los primeros centros para desarrollo de acústica computacional y *computer music*.

Autor	LuisaMei
Disco/Obra	ooo-Primi
Medio de distribución	Bandcamp
Publicación	25/11/2022
Fuente	<a href="https://luisamei.bandcamp.com/album/ooo-primi">https://luisamei.bandcamp.com/album/ooo-primi</a> <a href="https://www.youtube.com/watch?v=_Gnd1JidNIY">https://www.youtube.com/watch?v=_Gnd1JidNIY</a> (performance de live coding colaborativa)

Disco hecho usando SC, con sofisticados timbres granulados. En la performance colaborativa, que comparte la artista en el sitio de bandcamp, acompaña la música con visuales generadas con ayuda de SC por parte de su colaborador. En esta también se puede observar algo de la programación de la sintetización de sonido e imágenes.

Autor	Superpang (varios artistas)
Disco/Obra	Tour Mode
Medio de distribución	Bandcamp
Publicación	1/6/2021
Fuente	<a href="https://superpang.bandcamp.com/album/tour-mode">https://superpang.bandcamp.com/album/tour-mode</a>

Superpang<sup>24</sup> es un medio de distribución para músicos de *computer music* y música algorítmica. Aquí, en este disco, tomamos el ejemplo de la pieza *Palace of Swallow*, donde al principio, suena juego de paneos que llaman la atención, esto se puede ser replicado en cierta manera con SC usando el UGen *Pan.ar* y dejando una variable para controlar la posición tipo *\pan.kr*.

### 4.3 Práctica creativa II: Improvisaciones libres de live coding

Como se habló anteriormente, live coding es una ejecución en tiempo real de código, donde el algoritmo en que el computador está trabajando es alterado desde el interior, así como una larva parásita eclosiona dentro de su huésped (Collins, 2010). El diagrama de composición mediada por computadora para las prácticas de live coding de la Figura 4-3 deviene de la Figura 1-5. En este cuadro se muestra al compositor como punto de partida, quién trabaja con un software de sonido con ciertas características específicas. A través del software se crean instrumentos virtuales que alimentarán el convertidor de salida para generar sonidos musicales en un ambiente acústico y finalmente a los auditores. Para este trabajo el software elegido es SuperCollider y los instrumentos se construyen con las unidades generadoras.

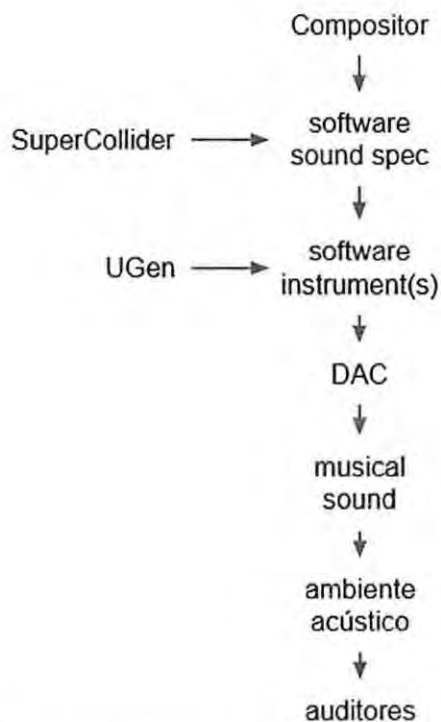


Figura 4-3 Diagrama de composición mediada por computadora para las prácticas de live coding.

Como una innovación al flujo de trabajo creativo, se añadió el uso de la plataforma GitHub para almacenar y generar un control de versiones de los archivos generados en SC. El

<sup>24</sup> <https://superpang.org>

procedimiento incluye la utilización del editor de código: *Visual Studio Code*, con el cual desde una terminal se puede transferir (push) desde el computador local hacia el perfil y descargar (pull) desde el perfil hasta el equipo local. Esto permite generar una carpeta de trabajo, pudiendo controlar versiones de los archivos, esto es, poder ver el historial de cambios en estos y poder restaurarlos de necesitarse.

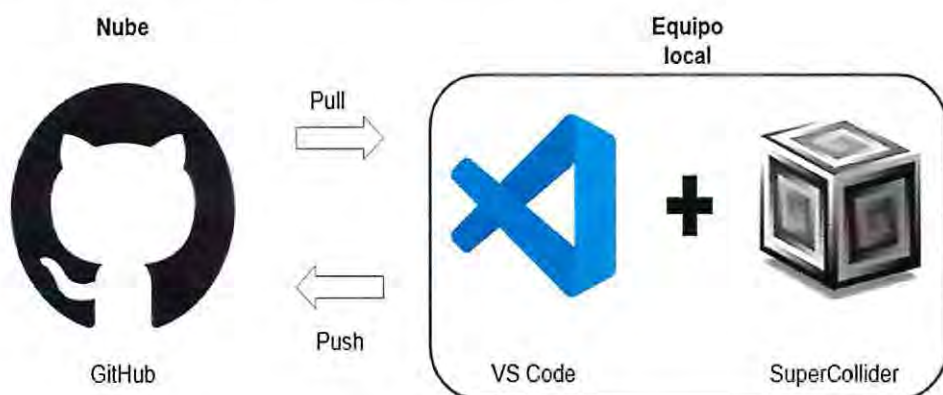


Figura 4-4 Flujo de carga/descarga de archivos entre GitHub y el computador local.

A continuación realizamos el análisis de una práctica con material generado a través del bucle de la Figura 4-1.

#### 4.3.1 Comentarios sobre la creación de sintetizadores y patrones para improvisaciones de live coding

En primer lugar, se trabajó con dos archivos complementarios, uno con el código para generar los sintetizadores diseñados y otro para generar patrones. El trabajar con dos archivos paralelamente tiene como beneficio el ordenar el flujo de trabajo. Dentro de lo visto entre live coders que muestran sus prácticas en diferentes canales de YouTube. Otra forma es activar un archivo dentro del script de patrones, pero no ha habido éxito a la fecha de entrega de este documento para activar los métodos recomendados<sup>25</sup>. Se puede también alterar los sintetizadores definidos mientras corre, sin embargo los instrumentos al ser estructuras de datos en la memoria volátil, esta alteración genera un *lag* cuyo tiempo dependerá de la máquina donde se esté corriendo SC y la complejidad del instrumento; a mayor poder de procesamiento menor *lag*. Ahora explicaremos los instrumentos ocupados en la práctica escogida para este apartado.

En primer lugar, al iniciar el servidor, se le ha configurado una grilla de tiempo con un tempo de 72/60, es decir, 1.2 segundos por beat. Para evitar que esta grilla se destruya al realizar un *hardreset* (combinación de teclas *Ctrl+*), se usa el método *.permanent*.

<sup>25</sup> A través de *.load* o *.loadRelative*. Ver: <https://www.youtube.com/watch?v=rnjiSc3blCc&t=453s>

También se ha creado un *ProxySpace* que automatizará los nodos y grupos. Los métodos *.push* y *.quant*, permiten trabajar dentro del espacio creado y asegurar que la grilla de tiempo se ajustará en él. Todo esto puede verse en la Tabla 4-1.

Tabla 4-1 Bóoteo y configuración inicial.

1	(
2	s.boot;
3	t = TempoClock.new(72/60).permanent_(true);
4	p = ProxySpace.new(s, clock: t);
5	p.push;
6	p.quant;
7	)

El instrumento de la Tabla 4-2 funciona principalmente en base al resonador *Resonz*, el cual es alimentado con el generador de ruido blanco *ClipNoise*, este generador tiene límites entre -1 y 1. Se ha preferido este a otros generadores similares como *WhiteNoise* o *LFDClipNoise*. La envoltura escogida es de tipo exponencial, hecha con *XLine*. Aunque los parámetro de verticales por defecto se han configurado para decaer desde 1 a 0.0001, estos se han dejado disponibles para ser alterados en el algoritmo de patrones a través de las variables *\start* y *\end*. Al definir *doneAction* con *Done.freeSelf*, el instrumento se liberará al terminar de tocar<sup>26</sup>. La duración se ha dejado con una variable con la misma etiqueta con la que se altera el patrón rítmico.

<sup>26</sup> Esto es importante, ya que de no ser liberado, el bloque creado se mantiene en memoria y puede provocar un colapso del sistema.

Tabla 4-2 Instrumento \resnoise.

```

1  (
2  SynthDef(\resnoise, {
3    var sig, env;
4    sig = ClipNoise.ar;
5    env = XLine.kr(\start.kr(1.0), \end.kr(0.0001), \dur.kr(1/8), doneAction:
6    Done.freeSelf);
7    sig = Resonz.ar(sig, \freq.kr(2000), env);
8    sig = Pan2.ar(sig, \pan.kr(0.0), \amp.kr(0.2));
9    Out.ar(0, sig)
10  }).add;
11  )

```

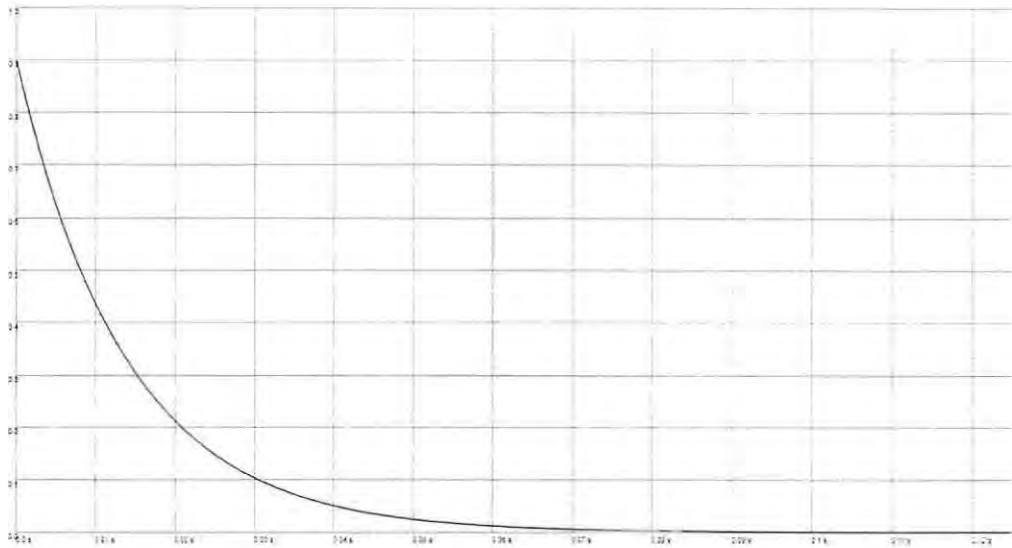


Figura 4-5 Forma de la envolvente de \resnoise.

La Tabla 4-3 muestra el patrón en que se prueba el instrumento \resnoise, etiquetado así por tener la intención de ser usado como un snare drum. Destacamos primero el uso de \note para traducir, de forma sencilla, frecuencias a tonos, el rango de frecuencias elegido para este instrumento ha sido medio-bajo. En cuanto al tiempo, en este ejemplo se ha dejado con golpes cortos. Sin embargo, al haber dejado la duración de la envoltura con la misma etiqueta que ocupa Pbind para el manejo de la duración hasta el próximo evento, el sonido dura todo el beat, los cuales se han dejado en fracciones de 1/4 y 1/8, esto es, negras y corcheas. El paneo se ha automatizado con un patrón aleatorio de distribución uniforme, para que vaya hasta la mitad en L y R. Los resultados en cuanto a enriquecimiento de frecuencia y su duración se pueden observar en el espectrograma de la Figura 4-6, generado por SC gracias a una extensión descargada, configurado para la ejecución del patrón con un rango de 0.0 KHz a 1.1 KHz. Al ejecutar la línea 12, se liberará la variable *\resnare* luego tres segundos, con la intensidad decayendo regularmente.

Tabla 4-3 Patrón para instrumento \resnoise.

1	(
2	~resnare = Pbind(
3	\instrument, \resnoise,
4	\start, Pseq([1.0],inf),
5	\end, Pwhite(0.00005,0.0001,inf),
6	\dur, Prand([1/8,1/4], inf),
7	\note, Pxrand([1,3,5,7,9],inf) - 20.0,
8	\pan, Pwhite(-0.5,0.5,inf),
9	\amp, Pxrand([0.09,0.2,0.5,0.7,0.9],inf)
10	)
11	)
12	~resnare.free(3);



Figura 4-6 Espectrograma del instrumento \resnoise con el patrón de la Tabla 4-4.

Otro instrumento es el de la Tabla 4-4, llamado \rlpfimp, por los UGen en que está basado, un filtro resonador pasa bajos y un generador de impulsos. Se han añadido los factores  $n$ ,  $d$  y  $f$  para ampliar o disminuir el rango de la señal computada. La frecuencia de los impulsos es definida en el algoritmo de los patrones de la Tabla 4-5. Al UGen *RLPF* ingresa la señal del generador de impulso, la frecuencia de corte calculada del bloque varía según la envolvente de Figura 4-7, y el recíproco del factor calidad,  $rq$ , que es definido como

Ecuación 4-1 Factor de calidad  $rq$ .

$$rq = \frac{\text{Ancho de banda}}{\text{Frecuencia de corte}} [-]$$

y se le asignan valores aleatorios con distribución uniforme en el algoritmo de patrones de la Tabla 4-5. Se debe tener cuidado con estos valores, ya que si este factor es cero,

implica que la frecuencia de corte es infinita y habrá problemas con el cómputo del bloque.

Tabla 4-4 Instrumento \rlpfimp.

```

1 (
2 SynthDef.new(\rlpfimp, {
3   arg n, d, f;
4   var sig, env;
5   env = EnvGen.kr(Env([8000,1000,50]), doneAction: Done.freeSelf) * n;
6   sig = Impulse.ar(\freq.kr(50) * d);
7   sig = RLPF.ar(sig, env * f, \rq.kr(0.05));
8   sig = Pan2.ar(sig, \pan.kr(0.0), \amp.kr(0.2));
9   Out.ar(0, sig)
10  }).add;
11 )

```

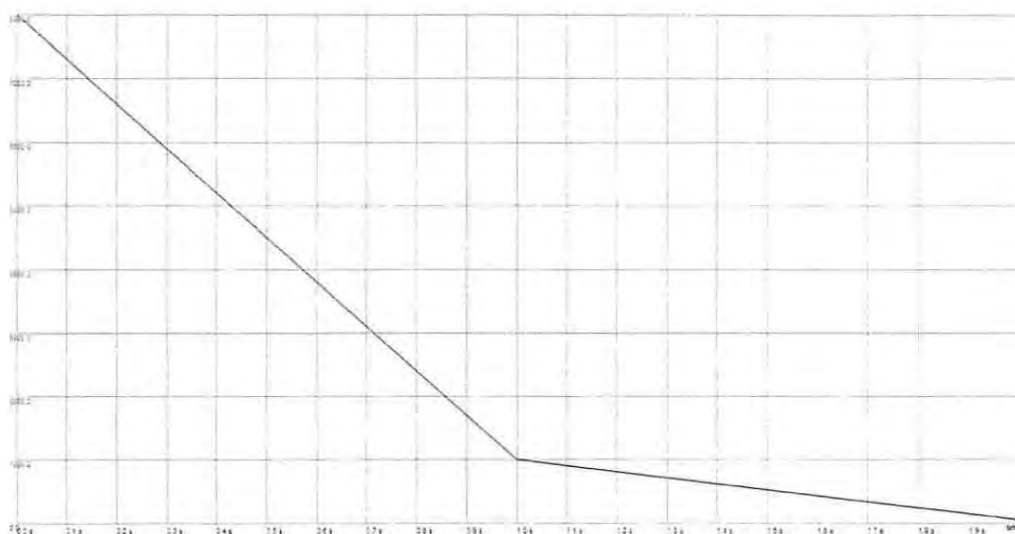


Figura 4-7 Forma de la envolvente de \resonoise.

Tabla 4-5 Patrón para instrumento \rlpfimp.

```

1 (
2 ~resfimp = Pbind(
3   \instrument, \rlpfimp,
4   \dur, Pxrang([1/2,1],inf),
5   \midinote, Pseq([20],inf) + [0.0,0.1] + 15,
6   \rq, Pwhite(0.01,0.5,inf),
7   \d, Pwhite(0.1,0.8,inf),
8   \f, Pwhite(0.3,0.5,inf),
9   \n, Pwhite(0.1,3.0,inf),
10  \pan, Pwhite(-1,1, inf) ,
11  \amp, Pwhite(0.2,0.5, inf)
12 )
13 )
14 ~resfimp.free(2)

```



Figura 4-8 Espectrograma del instrumento `\rlpfimp` con el patrón de la Tabla 4-6.

Una variante de interés de este patrón se muestra en la Tabla 4-6, la etiqueta `\midinote` permite traducir valores numéricos a notas midi<sup>27</sup>. Se han añadido dos sumandos, uno que permite añadir dos desfases a la señal y otro que añade al total de salida, este último es mejor para cambiar totalmente el rango de frecuencias de lo que toca el instrumento `\rlpfimp`. En este patrón se ha añadido también un factor al valor de `rq`, para ir alterando la salida de éste de forma porcentual, manteniendo el cuidado mencionado anteriormente. Para controlar con mayor rapidez la ganancia de salida, etiquetada como `\amp`, se agregó también un factor.

Tabla 4-6 Instrumento `\rlpfimp`.

```

1  (
2  ~resfimp = Pbind(
3    \instrument, \rlpfimp,
4    \dur, Pxrang([1],10),
5    \midinote, Pseq(#[28,32,48,64],inf) + [0.1,0.25] + 25,
6    \rq, Pwhite(0.001,0.5,inf) * 0.75,
7    \d, Pwhite(0.01,0.8,inf),
8    \f, Pwhite(0.03,0.5,inf),
9    \n, Pwhite(0.01,3.0,inf),
10   \pan, Pwhite(-1,1, inf) ,
11   \amp, Pwhite(0.2,0.5, inf) * 0.2
12  )
13  )
14  ~resfimp.free(2)

```

<sup>27</sup> Otras traducciones pueden ser consultadas en la documentación, ejecutando `Pbind.help` en el IDE de SC.



Figura 4-9 Espectrograma del instrumento `\rlpfimp` con el patrón de la Tabla 4-6.

El instrumento en la Tabla 4-7, denominado `\klang2`, es uno de los instrumentos que fueron evolucionando desde las primeras prácticas. *Klang* permite generar grupos de senoides de forma sencilla, aquí se generan en un rango exponencial de 400 a 2000 Hertz, 24 veces, usando el método `.dup`. Se agregó un factor para variar posteriormente el valor de la función, entre llaves, ya que de otra forma se repite el mismo grupo de frecuencias en cada ejecución del instrumento. La envolvente de este instrumento es de tipo senoidal (ver Figura 4-10), configurada en la línea 7 de la Tabla 4-8. Para el control de la duración de esta se ha asignado la variable *d*. En la línea 7 se realiza una modulación entre la señal de forma de onda y el grupo de oscilaciones generadas con *Klang*.

Tabla 4-7 Instrumento \klang2.

```

1 (
2 SynthDef.new(\klang2, {
3   arg n, d;
4   var sig, env;
5   sig = Klang.ar([ {exprand(400, 2000)}.dup(24) * n, nil, nil ], 1, 0);
6   env = EnvGen.kr(Env.sine(d), doneAction: Done.freeSelf);
7   sig = sig * env;
8   sig = Pan2.ar(sig, \pan.kr(0.0), \amp.kr(0.2));
9   Out.ar(0, sig)
10  }).add;
11 )

```

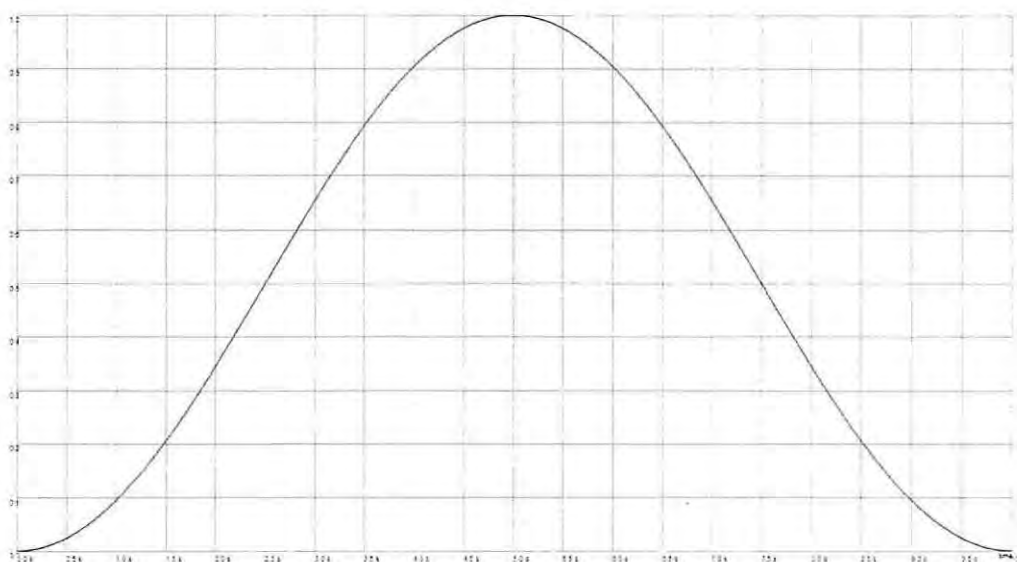


Figura 4-10 Forma de la envolvente de \klang2.

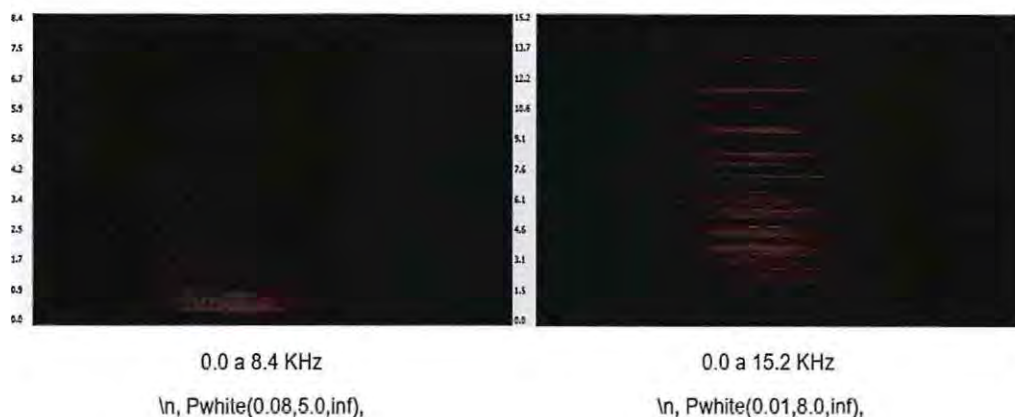
La ejecución de este instrumento se ha hecho de manera incidental, con duraciones a compás entero y cuatro repeticiones. Como se mencionó, se puede extender la sonoridad aumentando el valor de `\d` en el algoritmo de patrones de la Tabla 4-8. Otro factor que tiene un interesante resultado es la extensión o disminución del rango de frecuencias que se ejecutan, mediante la alteración de la variable `n`, como se puede observar en la Figura 4-11. El panning se ha definido con una distribución uniforme de valores aleatorios, desde -1 a 1, entonces habrán repeticiones que se ejecutarán completamente en L o R.

Tabla 4-8 Patrón para instrumento \klang2.

```

1 (
2 ~klang = Pbind(
3   \instrument, \klang2,
4   \dur, Pxrnd([1],4),
5   \n, Pwhite(0.08,5.0,inf),
6   \d, 10.0,
7   \pan, Pwhite(-1,1,inf),
8   \amp, Pwhite(0.01,0.07, inf) * 0.5
9 )
10 )
11 ~klang.free(2)

```

Figura 4-11 Espectrograma del instrumento \klang2 con el patrón de la Tabla 4-8 y diferentes valores de `\n`.

Otro instrumento creado es `\blips`, hecho a partir de un generador de impulsos oscilatorios de banda limitada, como puede verse en la Tabla 4-9. La documentación advierte que esta forma de onda puede generar problemas auditivos si se usa con altas intensidades por mucho tiempo. Su frecuencia y número de armónicos se controlan desde el algoritmo de patrones en la Tabla 4-10. La envolvente del instrumento es hecha con el método `.perc` que genera la forma de la Figura 4-12, donde su ataque y *release* se han dejado variables.

Tabla 4-9 Instrumento \blips.

```

1  (
2  SynthDef.new(\blips, {
3    var sig, env;
4    sig = Blip.ar(\freq.kr(200), \harm.kr(30));
5    env = EnvGen.kr(Env.perc(\atk.kr(0.001), \rel.kr(0.5)), doneAction:
6    Done.freeSelf);
7    sig = sig * env;
8    sig = Pan2.ar(sig, \pan.kr(0.0), \amp.kr(0.2));
9    Out.ar(0, sig)
10  }).add;
11  )

```

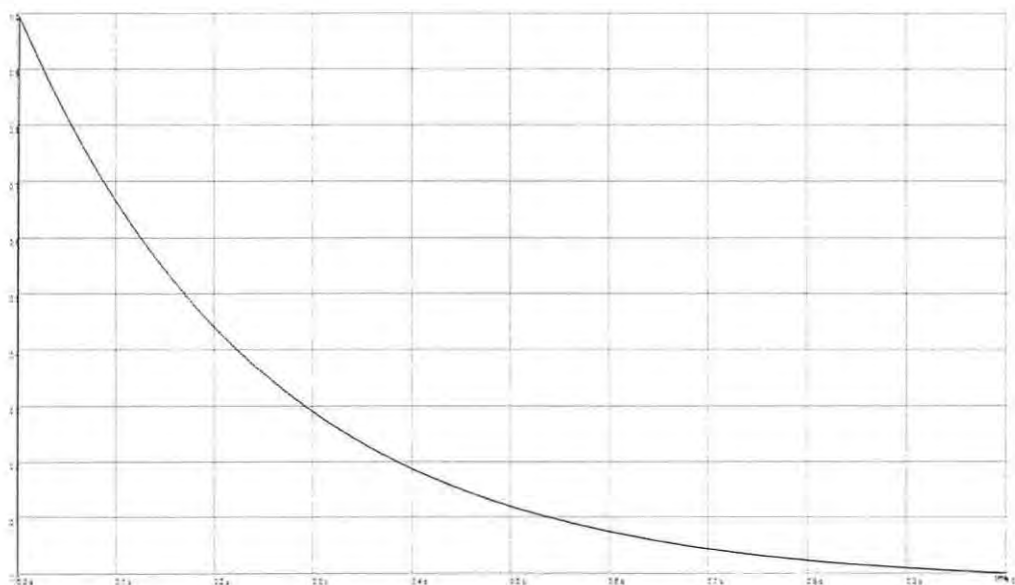


Figura 4-12 Forma de la envolvente de \blips.

Este patrón se ha usado con figuras rítmicas cortas, como corcheas (1/8) y negras (1/4). También se han usado notas midi para generar los diferentes pitch de frecuencias, con desfases y un sumando para alterar el resultado general en este aspecto. Al número de armónicos se le ha añadido un sumando aleatorio entre -2.0 y 6.0. El resultado espectral de la ejecución del instrumento se puede observar en la Figura 4-13.

Tabla 4-10 Patrón para instrumento \blips.

1	(
2	~blips = Pbind(
3	\instrument, \blips,
4	\dur, Prand([1/8],inf),
5	\midinote, Pxrnd(#[25,25,30,28], inf) + [-0.15,0.15] + 38,
6	\harm, Pxrnd([1,3,5,9,10,12,15], inf) + (-2.0..6.0).choose,
7	\atk, 0.065,
8	\rel, 0.1,
9	\curve, -8.0,
10	\pan, Pwhite(-0.4,0.4, inf),
11	\amp, Pwhite(0.01,0.5, inf) * 0.75
12	)
13	)
14	~blips.free(10)



Figura 4-13 Espectrograma del instrumento \blips con el patrón de la Tabla 4-10.

Los instrumentos y los patrones confeccionados en este apartado han sido ocupados de manera libre durante las prácticas, incrementando las técnicas que permitan un desenvolvimiento fluido en la codificación y alteración. En el Anexo A se encuentra una colección de instrumentos que se pueden usar con los patrones del Anexo B. A continuación mostramos otra experiencia, donde lo presentado aquí fue contextualizado en una actividad que involucra la observación de un entorno particular y que sirvió para inspirar a una práctica de creación de un algoritmo.

### 4.3.2 Una reflexión cibernética para la creación de un algoritmo de improvisación en SC

Esta creación fue en primera instancia una solicitud evaluada para la asignatura de Etnografía Musical, donde se escogió como objeto de estudio el metro que abarca, a la fecha en que se escribe este documento, desde la ciudad de Limache a Valparaíso. Lo que sigue es una transcripción directa de lo que fue la investigación, reflexión y método creativo.

*“Por veinte años he tenido que realizar viajes intercomunales entre Villa Alemana y Valparaíso, en este considerable tiempo humano fui testigo de cambios estructurales del espacio conocido primero como tren y luego metro. Por mencionar algunos, el cortador de boletos desapareció, siendo reemplazado por una pieza de tecnología para marcar la entrada a un espacio medianamente automatizado. Espacio con un paisaje sonoro propio, desde el beep de entrada, hasta el beep de salida, las indicaciones de las estaciones, entre otros.*

*En este primer avance, nos referiremos a este espacio como un ciberespacio de primer orden, definiremos y reflexionaremos por qué en la siguiente sección. En la obra que nos proponemos realizar, tomaremos el paisaje sonoro de este ciberespacio y lo reestructuramos, en un intento humano por mantener el control, ¿cómo?, generando otro ciberespacio de diferente orden, sintetizado, que engloba una parte del que no podemos manejar sino solamente transitar.”*

En la actualidad, el término ciberespacio tiene diversas definiciones, las discusiones donde este es utilizado abarcan dimensiones artísticas, legales, militares y científicas entre otras. Esto me interesa debido a las consecuencias implícitas que tienen ciertas definiciones, que expondremos aquí, para nuestras relaciones cotidianas.

A mediados del siglo XX, Norbert Wiener explica (Wiener, 1961) en su trabajo: *“Cibernética: O Control y Comunicación en el Animal y la Máquina”*, como fue que él y el mexicano Arturo Rosenblueth, debido a la poca unidad en la literatura concerniente a los problemas de la comunicación, control y mecánica estadística, ya sea en la máquina o en el tejido vivo, y la falta de terminología común, o incluso la falta de un nombre para el campo (intelectual) acuñaron a la naciente disciplina. La idea que los humanos pueden interfacear con máquinas y que ese sistema resultante puede proveer un ambiente alternativo para la interacción, provee una fundación para el concepto de ciberespacio (Ottis & Lorents, 2010). Dice Wiener: "Hemos decidido llamar a todo el

campo del control y teoría de la comunicación, sea en la máquina o el animal, por el nombre Cibernética, que lo formamos desde el griego κυβερνήτης o timonero”<sup>28</sup>.

En otra ocasión (Wiener, 1948), Wiener complementa diciendo:

"La palabra cibernética es tomada de la palabra 'kybernetes', o timonero. De la misma palabra Griega, a través de la corrupción latina 'gubernator' viene el término gobernador que ha sido usado por mucho tiempo para designar cierto tipo de mecanismo de control... El concepto básico en que ambos, Maxwell (Clerk) y los investigadores de la cibernética, quieren describir, a través de la elección de esta palabra es el del mecanismo de la retroalimentación, que es especialmente bien representado por el motor de dirección de un barco."

Curioso es que Wiener pasara sus últimas etapas se la pasaba predicando sobre los enormes peligros de las máquinas y la automatización, que él mismo había contribuido a desarrollar y que en su país, EEUU, se fue adoptando el término inteligencia artificial, lo que contribuyó al olvido del término cibernética, que se empezó a asociar con el comunismo (Rajsbaum & Morales, 2016).

La primera vez que se usó el término ciberespacio fue en el libro *Neuromancer* (Gibson, 1984), el autor describe:

"Ciberespacio. Una alucinación consensual experimentada diariamente por billones de operadores legítimos, en cada país, por niños a los que se les enseña conceptos matemáticos... Una representación gráfica de datos abstraída de bancos en cada computador en el sistema humano. Complejidad impensable."

Esta definición se enfoca en la percepción humana del nuevo ambiente, pero aún es relevante, ya que ilustra el potencial para desarrollar una verdadera experiencia inmersiva de ciberespacio. La segunda parte de la definición identifica la complejidad como una de las características principales del ciberespacio (Ottis & Lorents, 2010).

Si bien la definición de Gibson definición parece ser más cercana a lo que fue la inspiración del movimiento cibernético, el crear una ciencia de la mente (Varela, 1988), no es precisamente la que me interesa, es necesario entender estas visiones, ya que dan una noción de lo que me parece que ocurre, en cierto nivel (que no es el inmersivo total), en espacios donde los humanos coinciden físicamente, particularmente en el que he puesto mi atención para la obra que se propone, esto es, el metro que va desde Limache a Valparaíso. Donde, ya sea en cualquier estación en la que se decida abordar, es entrar en un espacio regulado a través de tecnologías. Antes de continuar con las

---

<sup>28</sup> En el artículo usa la palabra *steersman*.

reflexiones, me parece necesario seguir indagando en definiciones, ahora para saber qué entienden algunas instituciones por ciberespacio.

El ministerio de defensa Nacional de Chile, indica en su sitio:

"En el ámbito de la Defensa, hoy se considera al ciberespacio como un nuevo ambiente en el que se desenvuelven diversos conflictos de diversa naturaleza, nacionales e internacionales. Esto se traduce en la definición del ciberespacio como una dimensión diferente al espacio terrestre, aéreo y marítimo, que requiere contar con las políticas, planificaciones y capacidades que permitan ejercer los roles propios de la Defensa Nacional en este ámbito." (MDN, n.d.)

Esta definición es más cercana a la del departamento de defensa de los Estados Unidos de América, donde se considera que "este es un dominio global dentro del ambiente de la información, que consiste en una red interdependiente de infraestructura de tecnologías de la información, incluyendo la internet, redes de telecomunicación, sistemas de computadoras, y procesadores y controladores embebidos" (CJCS, 2018). Esta noción informática del término es la más usada por las instituciones, esto en los contextos de ciberseguridad. Sin embargo, el trabajo de Ottis & Lorents, cuyo resumen aparece en el sitio del Centro de Excelencia de Ciberdefensa Cooperativa (CCDCOE, n.d.) de la OTAN, ofrece la siguiente definición: "Ciberespacio es un conjunto tiempo-dependiente de información interconectado de sistemas y los usuarios humanos que interactúan con estos sistemas."

Los autores proponen esta definición ya que dicen que no hay una definición común de ciberespacio y que las usadas suelen ser vagas o les falta algún componente esencial (Ottis & Lorents, 2010). En su caso, ellos hacen referencia a los humanos que interactúan con los sistemas, que recuerdan un poco a la descripción de Gibson, una experiencia consensual experimentada por operadores legítimos.

Parte de la inspiración de la definición de Ottis & Lorents está en el trabajo de Lance Strate (1999), donde indica: "Como un concepto colectivo, el ciberespacio puede ser entonces definido como las diversas experiencias del espacio, asociadas con la computación y tecnologías relacionadas."<sup>29</sup>

El artículo de Strate tiene como propósito presentar un estudio y taxonomía de los ciberespacios (Strate, 1999, p. 383), estos los resume en tres niveles u órdenes:

- El primer nivel, concerniente con la ontología del ciberespacio. Se refiere a este como un ciberespacio de orden cero, equivalente al sótano o el lobby de un edificio.

---

<sup>29</sup> El texto original dice: "As a collective concept, cyberspace can then be defined as the diverse experiences of space associated with computing and related technologies."

4. Creación de piezas musicales de computer music para SuperCollider

- El siguiente nivel lo refiere como ciberespacio de primer orden. Se enfoca en los elementos básicos del ciberespacio, o sus bloques de construcción.

- El nivel final es el ciberespacio de segundo orden. Representa una síntesis de los elementos básicos.

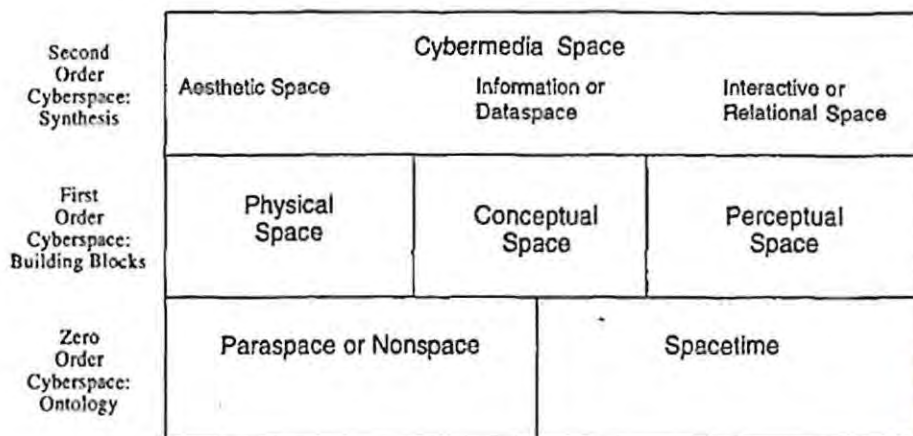


Figura 4-14 Los órdenes del ciberespacio.  
(Fuente: Strate, 1999, p. 384)

Desde las últimas definiciones podemos ya considerar el lugar de los humanos, teniendo experiencias en entornos de tecnologías específicas, interconectadas, tiempo-dependientes y asociadas a la computación. Podríamos considerar la visión de obras de ciencia-ficción, tipo matrix, blade runner, y otras, pero aún no llegamos ahí. Nuestra realidad local es oír el beep de entrada y salida al marcar nuestras tarjetas, el beep que marca cada estación, y las voces en los altavoces ordenando usar mascarilla y no comer en los vagones.

Este es fin de la transcripción, a continuación exponemos otra transcripción, correspondiente a la ficha técnica solicitada para la realización del proyecto de creación durante el año 2022:

Tabla 4-11 Ficha para inscripción de obra.

Unidad de Análisis (UA)	Metro de Valparaíso, viajes desde Villa Alemana a Valparaíso
¿Qué información sociofónica espero obtener con esta actividad?	Paisaje sonoro de lo que se denominará un ciberespacio de primer orden, esta denominación es explicada en el texto de avance para la primera etapa. Información sociofónica a la que están expuestas las personas que usan este medio de transporte.

¿Qué niveles de observación contempla en el acercamiento a la UA?	Debido a que este trabajo tiene como propósito realizar una obra, estará centrado en el primer nivel de observación, el "yo", relacionado con la subjetividad y la sensibilidad (Cambrón, A. (2010). <i>"Etnografía sonora. Reflexiones prácticas"</i> ).
Metodología	Se realizarán grabaciones en viajes cotidianos, activando las tomas desde que se ingresa a la estación de partida usando la tarjeta magnética y finalizando de forma similar en estación de destino. El recorrido cotidiano es desde la estación La Concepción hasta la estación puerto, y el regreso desde la estación Francia hasta La Concepción. Los horarios no son fijos y dependen de la necesidad de viajar. Se hará un mapeo del paisaje sonoro de los archivos de audio, creando categorías que nos permita caracterizar momentos que nos parezca notables.
Instrumentos	<ul style="list-style-type: none"> <li>- Tascam DR-05, equipo de grabación</li> <li>- Computador</li> <li>- Google documents, Hojas de cálculo para mapeo</li> <li>- Cubase 5, Digital Audio Workstation</li> <li>- SuperCollider</li> </ul>
Descripción	<p>La actividad tiene como finalidad crear una obra musical a partir de una investigación etnográfica, donde la unidad de análisis, niveles de observación, metodología e instrumentos se definieron en este documento.</p> <p>Para este primer avance se entregará un documento que nos permitirá establecer al espacio sociofónico como un ciberespacio de primer orden, a través de las distintas definiciones que han surgido en el tiempo y reflexiones personales para darle sentido a la noción que usaremos en adelante. Un ciberespacio donde las personas interactúan con tecnologías que tienen singularidades sonoras que registramos.</p> <p>En una segunda etapa se trabajarán temas estéticos, extrayendo singularidades que nos interesen usando un DAW y luego procesándolas, sintetizando un ciberespacio de segundo orden. A través de una algoritmia y técnicas de</p>

	especialización y procesamiento, se planteará la obra finalmente.
--	---

El resultado final de esta reflexión cibemética, en combinación de una predisposición algorítmica, ha sido la elaboración de un flujo de datos que permita dar sentido a la improvisación usando SC, el esquema de la Figura 4-15 es un bosquejo de este flujo. Un conjunto sencillo de reglas y condiciones obtenidas desde las observaciones cotidianas del viaje en el metro de Valparaíso. Se han agregado algunos sintetizadores a gusto para la experimentación. El código resultante, es decir, con los últimos cambios luego de improvisar y haber generado un audio con el mismo software puede verse en el Anexo C.

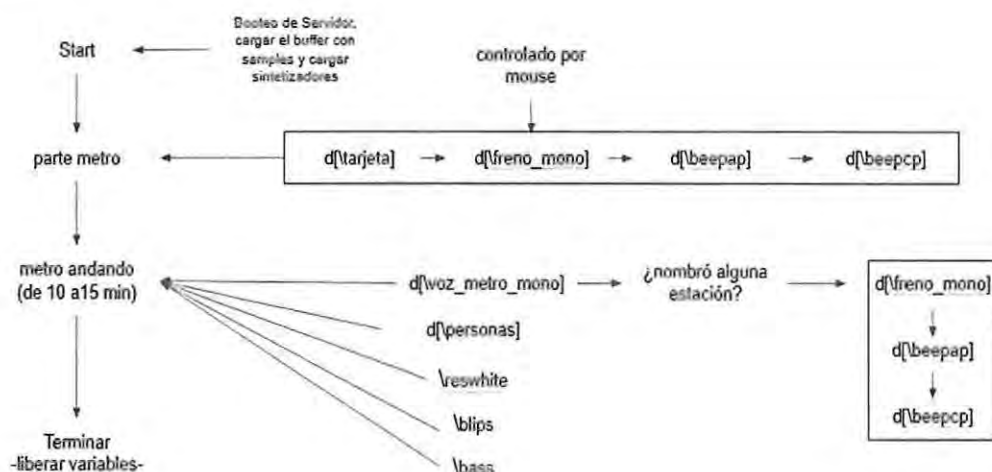


Figura 4-15 Esquema general para improvisar en pieza: Cibermetroval.

Los archivos de audio generados con este flujo fueron generados con la misma herramienta, primero en formato *aif* y luego comprimidos a *mp3*. Dentro de las cosas mejorables, principalmente están el uso de las granulaciones y procesamientos de los samples. Un problema que surgió y no se pudo solucionar para la entrega del avance tiene que ver con algunos ruteos de samples que dieron error, sin embargo esto se pudo sopesar al no ser mayormente archivos críticos, a excepción del audio de llegada del metro, que tenía una mayor duración. También se puede seguir extrayendo más y mejor material de las tomas hechas, que puedan aportar a más reflexiones y generar otras relaciones que aporten condiciones o reglas para el esquema general.

## Conclusión y discusión final

En este trabajo se documentó un resumen a grandes rasgos de la historia de *computer music*, que permite contextualizar desde donde nos hemos planteado como creadores de piezas musicales y a qué tradición hemos adherido. También hemos mencionado la composición algorítmica que da explicación a las estrategias de creación con la plataforma elegida. Por último hemos descrito esta última, el software SuperCollider, si bien este documento no entra en detalles de uso, como lo haría un tutorial, hemos documentado cómo ha sido el proceso creativo con esta herramienta.

Sobre la música interactiva y live coding, podemos concluir una diferencia esencial que ayudará en su comprensión: la estrategia para el tratamiento del algoritmo con el que se está trabajando. Si es que el algoritmo que corre en la computadora está siendo alimentado con datos para obtener resultados musicales, usualmente mediante conversores análogo-digital (AD), entonces lo que se escucha se trata de música interactiva<sup>30</sup>. En cambio, si el algoritmo está siendo modificado mediante código y al vuelo (“on the fly”), es decir, mientras está ejecutándose, la música se tratará de live coding. Sin embargo, nada impide la mezcla de ambas estrategias.

El bucle de interacción y retroalimentación de la Figura 4-1 resolvió en gran medida el realizar una documentación acorde a nuestros objetivos, sin llegar a la categoría de composición, pero manteniendo un rigor académico. La autoetnografía ha probado ser aquí una herramienta capaz de facilitar la concentración y entrega de información de la investigación artística.

Si bien este trabajo no pretendía convertirse en un manual del software SuperCollider, se describió de forma general, dando posibilidades y terminología para su uso y exploración. Esta herramienta tiene muchas más aplicaciones que las que se pudieron mostrar, pero eso no está dentro de los márgenes de investigación. Hay muchas opciones para aprender a utilizarlo, una es la misma ventana de ayuda; también los canales en el sitio Youtube, como el de Eli Fieldsteel, Music Engraver, y otros, mencionados en la introducción.

Tanto el contenido del marco teórico y las reflexiones entregadas en esta tesis pueden ayudar a comprender el lugar desde donde aparece la música generada, el cruce arte-ciencia. Las personas a las que aquí pudimos recordar pueden otorgarnos nuevos

---

<sup>30</sup> También puede haber un proceso inverso.

estereotipos a los que mirar. Se pudo abarcar, a lo largo del texto, a músicos, ingenieros y matemáticos, sin embargo, como se vio en la Figura 1-1, la inspiración de *computer music* puede venir de otros caminos disciplinares también.

Para la realización de prácticas de creativas aquí mostradas, ligadas directamente con el trabajo en SuperCollider, hemos de considerar algunas competencias relacionadas al uso de este software. En primer lugar, tener un dominio intermedio del inglés, no sólo porque la herramienta está hecha a partir de este idioma, si no que las principales fuentes de estudios se han producido en él. Luego, para el diseño de instrumentos, desde la programación se deben tener nociones de la programación orientada a objetos. Lo básico del procesamiento de señales, para así entender que es lo que sucede al operar con las diferentes señales que se generen (ej., modulaciones mediante multiplicación), y como administrarlas (ej., uso de expansión de multicanal). Por último, para la ejecución de patrones que usen los instrumentos diseñados, asumiendo que el usuario entiende sus creaciones, la intuición y voluntad musical es la competencia principal: el manejo de las duraciones, intensidades, alturas, alteración de timbre.

En las primeras décadas del siglo en curso el uso de sistemas computarizados ha permeado la cotidianidad humana. La ley de Moore, aquella promesa a los inversores de que la capacidad de cómputo (relacionado a la densidad de componentes) se duplicaría cada dos años manteniendo los costos de fabricación de tales sistemas, aún se mantiene. Algunas consecuencias de esto son: los avances de en tecnologías de telecomunicación, que nos permiten establecer relaciones con variadas culturas. También la generación de sistemas autónomos y máquinas que generan máquinas.

Para finalizar, recordamos el interés del músico francés Edgar Varèse: 'le computer, c'est la nouvelle frontier' (Risset, 2004) (la computadora, es la nueva frontera), esperamos que este documento de proyecto de título pueda ayudar en algo a la comprensión de estas palabras y su visión de liberación del sonido, a través del relato de las personalidades que habitaron en algún momento o habitan aún en el cruce arte-ciencia, las referencias investigadas y por último, el trabajo logrado, esto es, la música de *computer music* creada mediante el software SuperCollider.

## Referencias en orden alfabético

- Bidlack, R. (1992). Chaotic Systems as Simple (But Complex) Compositional Algorithms. *Computer Music Journal*, 16(3), 33-47.  
<https://doi.org/10.2307/3680849>
- Brown, A. R. (2016). Performing with the other: the relationship of musician and machine in live coding. *International Journal of Performance Arts and Digital Media*, 12(2), 179-186.  
<https://doi.org/10.1080/14794713.2016.1227595>
- Bunge, M. (2005). *Diccionario de filosofía* (3ra ed.). Siglo XXI.
- Canal Eulerroom. (obtenido 8 de Julio 2022). on-the-fly.documentary [Archivo de Vídeo]. *Youtube*. <https://www.youtube.com/watch?v=ntFMuvv2-TY>
- Chairman of the Joint Chiefs of Staff (CJCS). (2018). *DOD dictionary of military and associated terms*. Washington, DC: Joint Chiefs of Staff.  
[https://www.jcs.mil/Portals/36/Documents/Doctrine/pubs/jp3\\_12.pdf](https://www.jcs.mil/Portals/36/Documents/Doctrine/pubs/jp3_12.pdf)
- Collins, N. (2010). *Introduction to Computer Music*. Wiley.
- Collins, N. (2016). Live coding and teaching SuperCollider. *Journal of music, technology and education*, 9(1), 5-16. [https://doi.org/10.1386/jmte.9.1.5\\_1](https://doi.org/10.1386/jmte.9.1.5_1)
- Cottle, D. M. (2005). *Computer Music with examples in SuperCollider 3*.  
[http://rhoadley.net/courses/tech\\_resources/supercollider/tutorials/cottle/CMSC7105.pdf](http://rhoadley.net/courses/tech_resources/supercollider/tutorials/cottle/CMSC7105.pdf)
- Croft, J. (2015). Composition is not research. *Tempo*, 69(272), 6-11.  
<https://doi.org/10.1017/S0040298214000989>
- Dannenberg, R. (2018). Languages for Computer Music. *Frontiers in Digital Humanities*, 5. <https://doi.org/10.3389/fdigh.2018.00026>
- Dean, R. T. (Ed.). (2011). *The Oxford Handbook of Computer Music*. Oxford University Press.
- Garnett, G. (2001). The Aesthetics of Interactive Computer Music. *Computer Music Journal*, 25(1), 21-33. <https://doi.org/10.1162/014892601300126089>

- Haugeland, J. (1981). Semantic engines: An introduction to mind design. In J. Haugeland (Ed.), *Mind Design: Philosophy, Psychology, Artificial Intelligence* (pp. 1-34). MIT Press.
- Heims, S. J. (1982). *John Von Neumann and Norbert Wiener: From Mathematics to the Technologies of Life and Death*. MIT Press.
- Henríquez, D. (2022). *Diseño de sensor textil para obra musical* [Informe de proyecto de título de ingeniería electrónica]. Pontificia Universidad Católica de Valparaíso.
- Hiller, L. (1981). Composing with Computers: A Progress Report. *Computer Music Journal*, 5(4), 7-21. <https://doi.org/10.2307/3679501>
- Keislar, D. (2011). A Historical View of Computer Music Technology. In R. T. Dean (Ed.), *The Oxford Handbook of Computer Music* (pp. 11-43). Oxford University Press. <https://doi.org/10.1093/oxfordhb/9780199792030.013.0002>
- Ledesma, E. (2015). The Poetics and Politics of Computer Code in Latin America: Codework, Code Art, and Live Coding. *Revista de Estudios Hispánicos*, 49(1), 91-120. <https://doi.org/10.1353/rvs.2015.0016>
- López-Cano, R., & Opazo, Ú. S. C. (2014). *Investigación artística en música. Problemas, métodos, experiencias y modelos* (1ra ed.). ESMUC.
- Mathews, M. V. (1963). The Digital Computer as a Musical Instrument. *Science*, 142(3592), 553-557. <https://doi.org/10.1126/science.142.3592.553>
- McCartney, J. (2002). Rethinking the Computer Music Language: SuperCollider. *Computer Music Journal*, 26(4), 61-68. <https://doi.org/10.1162/014892602320991383>
- McLean, A., & Dean, R. T. (Eds.). (2018). *The Oxford Handbook of Algorithmic Music*. Oxford University Press.
- McLuhan, M. (1964). *Understanding media: The extensions of man*. McGraw-Hill.
- Microsoft. (2023, January 3). *Object-Oriented Programming (C#)*. Microsoft Learn. Retrieved January 6, 2023, from <https://docs.microsoft.com/en-us/dotnet/csharp/fundamentals/tutorials/oop>
- Ministerio de Defensa Nacional (MDN). (n.d.). *Ciberdefensa*. <https://www.defensa.cl/temas-de-contenido/ciberdefensa/>
- Moore, F. R. (1990). *Elements of computer music*. Prentice Hall.

- Nierhaus, G. (Ed.). (2015). *Patterns of Intuition: Musical Creativity in the Light of Algorithmic Composition*. Springer Netherlands.  
<http://dx.doi.org/10.1007/978-94-017-9561-6>
- Olano, T. (2013, February 6). Lecture 5 - [Part 1/3] Karlheinz Stockhausen - Four Criteria of Electronic Music (KONTAKTE), (1972) [Video]. *YouTube*.  
<https://www.youtube.com/watch?v=7xyGtI7KKIY&t=23s>
- Ottis, R. & Lorents, P. (2010). *Cyberspace: Definition and Implications*. In *Proceedings of the 5th International Conference on Information Warfare and Security*, Dayton, OH, US, 8-9 April. Reading: Academic Publishing Limited, 267-270.
- Ottis, R. & Lorents, P. (n.d.). *Cyberspace: Definition and Implications*. CCDCOE.  
<https://ccdcoe.org/library/publications/cyberspace-definition-and-implications/>
- Patterson, D. A., & Hennessy, J. L. (2017). *Computer Architecture: A Quantitative Approach* (6th ed.). Elsevier Science.
- Pierce, J. (1996). Computer Music, Coming and Going. *Computer Music Journal*, 20(1), 49-51. <https://doi.org/10.2307/3681270>
- Pizarro, F., Piero, V., Yunge, D., Rodríguez, M., Hermosilla, G., & Leiva, A. (2018, Abril 13). Easy-to-Build Textile Pressure Sensor. *Sensors*, 18(4), 1190.  
<https://doi.org/10.3390/s18041190>
- Rajsbaum, S. & Morales, E. (2016). *Norbert Wiener y el origen de la Cibernética*. Revista Ciencia. México D.F.
- Sorensen, A., Swift, B., & Riddell, A. (2014). The Many Meanings of Live Coding. *Computer Music Journal*, 38(1), 65-76.  
[https://doi.org/10.1162/COMJ\\_a\\_00230](https://doi.org/10.1162/COMJ_a_00230)
- Strate, L. (1999). The varieties of cyberspace: Problems in definition and delimitation, *Western Journal of Communication*, 63:3, 382-412,  
<https://doi.org/10.1080/10570319909374648>
- Supper, M. (2001). A Few Remarks on Algorithmic Composition. *Computer Music Journal*, 25(1), 48-53. <https://doi.org/10.1162/014892601300126106>
- Valle, A. (2016). *Introduction to SuperCollider* (M. Koutsomichalis & A. Valle, Trans.). Logos Verlag Berlin.

- Varela, F. J. (1990). *Conocer: las ciencias cognitivas : tendencias y perspectivas : cartografía de las ideas actuales* (C. Gardini, Trans.). Gedisa.
- Varèse, E., & Wen-chung, C. (1966). The Liberation of Sound. *Perspectives of New Music*, 5(1), 11-19. <https://doi.org/10.2307/832385>
- Wiener, N. (1948). *CYBERNETICS*. *Scientific American*, 179(5), 14–19. <http://www.jstor.org/stable/24945913>.
- Wiener, N. (1961[1948]). *Cybernetics: Or Control and Communication in the Animal and the Machine* (2nd ed.). Cambridge, MA: MIT Press.
- Wiener, N. (1964). *I Am a Mathematician: The Later Life of a Prodigy*. M.I.T. Press.
- Wilson, S., Cottle, D., & Collins, N. (Eds.). (2011). *The SuperCollider Book*. MIT Press.
- Xenakis, I. (1992). *Formalized Music: Thought and Mathematics in Composition* (Rev. ed.). Pendragon Press.

# Índice de Figuras, Tablas y Ecuaciones

## Figuras

Figura 1-1 Contexto disciplinario de computer music (subdisciplinas esenciales en <i>itálica</i> )(trad.) .....	9
Figura 1-2 Ordenador IBM 704 .....	12
Figura 1-3 Diagrama esquemático de una típica unidad-instrumento en la orquesta de computadora.....	13
Figura 1-4 Arquitectura Von Neumann. ....	14
Figura 1-5 Composición mediada por computadora. (Trad.).....	16
Figura 2-1 Poster décimo aniversario Algorave.....	19
Figura 2-2 Imágenes desde registros fotográficos; desde arriba hacia abajo: “Strings” (Andres Rivera, Camilo Lillo); “Sillón Sonoro” (Georgia del Campo, José Candela); “Rueda de Canto” (Felipe Barrera, Oscar Santis).....	20
Figura 3-1 Línea de tiempo de lenguajes representativos e históricamente significativos para computer music. ....	21
Figura 3-2 Estructura de la aplicación SuperCollider. ....	22
Figura 3-3 Sclang como cliente de alto nivel.....	23
Figura 3-4 Árbol de nodos de {SinOsc.ar}.play. ....	23
Figura 3-5 Resultados gráficos de {SinOsc.ar}: .scope(2) a la izquierda y .plot(0.01) a la derecha. ....	24
Figura 3-6 SuperCollider IDE v3.12.2.....	25
Figura 3-7 Diagrama de audio en Windows. ....	26
Figura 3-8 Diagramas de para la representación de sintetizadores.....	27
Figura 3-9 Códigos y grafo para el sintetizador \pulseSine. ....	28
Figura 4-1 Bucle de interacción y retroalimentación entre práctica creativa y reflexión.....	30
Figura 4-2 Hoja para registro de audiciones. ....	33
Figura 4-3 Diagrama de composición mediada por computadora para las prácticas de live coding.....	40
Figura 4-4 Flujo de carga/descarga de archivos entre GitHub y el computador local. ....	41
Figura 4-5 Forma de la envolvente de \resnoise. ....	43
Figura 4-6 Espectrograma del instrumento \renoise con el patrón de la Tabla 4-4.44	
Figura 4-7 Forma de la envolvente de \resnoise. ....	45
Figura 4-8 Espectrograma del instrumento \rlpfimp con el patrón de la Tabla 4-6.46	
Figura 4-9 Espectrograma del instrumento \rlpfimp con el patrón de la Tabla 4-6.47	
Figura 4-10 Forma de la envolvente de \klang2.....	48
Figura 4-11 Espectrograma del instrumento \klang2 con el patrón de la Tabla 4-9 y diferentes valores de \n. ....	49
Figura 4-12 Forma de la envolvente de \blips.....	50
Figura 4-13 Espectrograma del instrumento \blips con el patrón de la Tabla 4-10.51	
Figura 4-14 Los órdenes del ciberespacio.....	55
Figura 4-15 Esquema general para improvisar en pieza: Cibermetroval.....	57

**Tablas**

Tabla 3-1 Correspondencias entre programación y música respecto a código SC.29	
Tabla 4-1 Booteo y configuración inicial. ....	42
Tabla 4-2 Instrumento \resnoise.....	43
Tabla 4-3 Patrón para instrumento \resnoise.....	44
Tabla 4-4 Instrumento \rlpfimp.....	45
Tabla 4-5 Patrón para instrumento \rlpfimp.....	45
Tabla 4-6 Instrumento \rlpfimp.....	46
Tabla 4-7 Instrumento \klang2. ....	48
Tabla 4-8 Patrón para instrumento \klang2.....	49
Tabla 4-9 Instrumento \blips. ....	50
Tabla 4-10 Patrón para instrumento \blips.....	51
Tabla 4-11 Ficha para inscripción de obra.....	55

**Ecuaciones**

Ecuación 4-1 Factor de calidad $r_q$ .....	44
--	----

## Anexo A

```
1  (  
2  SynthDef.new(\sinmix, {  
3    var sig, env;  
4    //sig = SinOsc.ar(rrand(200, 1000));  
5    sig = Mix.fill(20, {VarSaw.ar(rrand(200, 2000), 0)});  
6    env = EnvGen.kr(Env.linen(1, 2, 3, 0.6), doneAction: Done.freeSelf);  
7    sig = sig * env;  
8    //sig = Resonz.ar(sig, \freqres.kr(440), \bwr.kr(1.0));  
9    //sig = RHPF.ar(sig, \freqres.kr(440), \rq.kr(1.0));  
10   sig = RHPF.ar(sig, MouseY.kr(1000,4000), MouseX.kr(0.1,1.0));  
11   sig = Pan2.ar(sig, \pan.kr(0.0), \amp.kr(0.2));  
12   Out.ar(0, sig)  
13 }).add;  
14  
15 SynthDef.new(\bass, {  
16   | gate=1, curve=-8.0 |  
17   var sig, env;  
18   sig = SinOsc.ar(\freq.kr(100));  
19   env = EnvGen.kr(Env.perc(\atk.kr(0.001), \rel.ir(1), curve: curve), gate,  
20 doneAction: Done.freeSelf);  
21   sig = sig * env;  
22   sig = FreeVerb.ar(sig, \mix.kr(0.33), \room.kr(0.5), \damp.kr(0.5));  
23   sig = Pan2.ar(sig, \pan.ir(0.0), \amp.ir(0.3));  
24   Out.ar(0, sig)  
25 }).add;  
26  
27 SynthDef.new(\drum, {  
28   | gate=1, curve=-8.0, mul=1 |  
29   var sig, env, rlpf;  
30   sig = BrownNoise.ar(mul);  
31   sig = RHPF.ar(sig, \freq.kr(100), \rq.kr(1.0));  
32   env = EnvGen.kr(Env.perc(\atk.kr(0.001), \rel.ir(1), curve: curve), gate,  
33 doneAction: Done.freeSelf);  
34   sig = sig * env;  
35   sig = FreeVerb.ar(sig, \mix.kr(0.33), \room.kr(0.5), \damp.kr(0.5));  
36   sig = Pan2.ar(sig, \pan.ir(0.0), \amp.ir(0.3));  
37   Out.ar(0, sig)  
38 }).add;  
39  
40 SynthDef.new(\drum2, {  
41   | gate=1, curve=-8.0, mul=1 |  
42   var sig, env;  
43   sig = WhiteNoise.ar(mul);  
44   env = EnvGen.kr(Env.perc(\atk.kr(0.001), \rel.ir(1), curve: curve), gate,  
45 doneAction: Done.freeSelf);  
46   sig = sig * env;  
47   sig = FreeVerb.ar(sig, \mix.kr(0.33), \room.kr(0.5), \damp.kr(0.5));  
48   sig = Pan2.ar(sig, \pan.ir(0.0), \amp.ir(0.3));  
49   Out.ar(0, sig)  
50 }).add;  
51  
52 SynthDef.new(\mel, {  
53   | gate=1, curve=-8.0, mul=1 |
```

```

54   var sig, env;
55   sig = VarSaw.ar(\freq.kr(100));
56   env = EnvGen.kr(Env([0,1,0], [1,1]), gate, doneAction: Done.freeSelf);
57   sig = sig * env;
58   sig = Pan2.ar(sig, \pan.ir(0.0), \amp.ir(0.3));
59   Out.ar(0, sig)
60   }).add;
61
62   SynthDef.new(\bass2, {
63     | gate=1, curve=-8.0 |
64     var sig, env;
65     sig = SinOsc.ar(\freq.kr(100));
66     env = EnvGen.kr(Env.perc(\atk.kr(0.001), \rel.ir(1), curve: curve), gate,
67 doneAction: Done.freeSelf);
68     sig = sig * env;
69     sig = FreeVerb.ar(sig, \mix.kr(0.33), \room.kr(0.5), \damp.kr(0.5));
70     sig = Pan2.ar(sig, \pan.ir(0.0), \amp.ir(0.3));
71     Out.ar(0, sig)
72   }).add;
73
74   SynthDef.new(\coin, {
75     arg prob=0.1;
76     var trig;
77     trig = Impulse.ar(\freqimp.kr(9.0), 0, SinOsc.kr(0.01, 0, 1, 1));
78     trig = EnvGen.ar(Env.perc(0.01, 1.0, 1.0, -8.0), doneAction: Done.freeSelf) *
79     trig;
80     trig = Mix.fill(7.0, {Ringz.ar(CoinGate.ar(prob, trig), #[0.5, 0.75] * Rand(3000,
81 9000), 0.05)});
82     trig = NHHall.ar(trig, rt60: 0.45, lowRatio: 0.4, earlyDiffusion: 0.1,
83 lateDiffusion: 0.65, modRate: 0.2, modDepth: 0.4);
84     Out.ar(0, trig * 0.75)
85   }).add;
86
87   SynthDef.new(\bufplay, {
88     arg buf=0, rate=1, amp=1, pan=0;
89     var sig;
90     sig = PlayBuf.ar(2, buf, BufRateScale.ir(buf) * rate, doneAction:2);
91     sig = Pan2.ar(sig, pan, amp);
92     Out.ar([0,1], sig);
93   }).add;
94
95   SynthDef.new(\blips, {
96     var sig, env;
97     sig = Blip.ar(\freq.kr(200), \harm.kr(30));
98     env = EnvGen.kr(Env.perc(\atk.kr(0.001), \rel.kr(0.5)), doneAction:
99 Done.freeSelf);
100    sig = sig * env;
101    sig = Pan2.ar(sig, \pan.kr(0.0), \amp.kr(0.2));
102    Out.ar(0, sig)
103  }).add;
104
105  SynthDef.new(\klang, {
106    arg n;
107    var sig, env;
108    sig = Klang.ar(' [ {exprand(400, 2000)} .dup(12) * n, nil, nil ], 1, 0);
109    env = EnvGen.kr(Env([0,1,0], curve: 'lin'), doneAction: Done.freeSelf);
110    sig = sig * env;
111    sig = Pan2.ar(sig, \pan.kr(0.0), \amp.kr(0.2));
112    Out.ar(0, sig)
113  }).add;
114

```

```

115 SynthDef.new(\klang2, {
116   arg n;
117   var sig, env;
118   sig = Klang.ar([ {exprand(400, 2000)}.dup(24) * n, nil, nil ], 1, 0);
119   env = EnvGen.kr(Env.sine(6), doneAction: Done.freeSelf);
120   sig = sig * env;
121   sig = Pan2.ar(sig, \pan.kr(0.0), \amp.kr(0.2));
122   Out.ar(0, sig)
123 }).add;
124
125 SynthDef.new(\twobandsq, {
126   arg n, d, f;
127   var sig, env;
128   //env = XLine.kr(8000,200,2, doneAction: Done.freeSelf);
129   env = EnvGen.kr(Env([8000,1000,50]), doneAction: Done.freeSelf) * n;
130   sig = Impulse.ar([100,250] * d);
131   sig = RLPF.ar(sig, env * f, \rq.kr(0.05));
132   sig = Pan2.ar(sig, \pan.kr(0.0), \amp.kr(0.2));
133   Out.ar(0, sig)
134 }).add;
135
136 SynthDef(\rcswhite, {
137   var sig, env;
138   sig = ClipNoise.ar;
139   env = XLine.kr(1, 0.0001, \dur.kr(8), doneAction: Done.freeSelf);
140   sig = Resonz.ar(sig, \freq.kr(2000), env);
141   sig = Pan2.ar(sig, \pan.kr(0.0), \amp.kr(0.2));
142   Out.ar(0, sig)
143 }).add;
144
145 SynthDef(\bassdrumgpt, {
146   arg freq=100;
147   var env, sig;
148   env = EnvGen.kr(Env.perc(\atk.kr(0.01), \rel.kr(0.5)), doneAction:2);
149   sig = BPF.ar(Impulse.ar(freq), freq*2, 0.5);
150   sig = sig * env;
151   //sig = NHHall.ar(sig, 0.1, 1, 100, 0.75, 2000, 1, 0.10);
152   sig = Pan2.ar(sig, \pos.kr(0.0), \amp.kr(0.2));
153   Out.ar(0, sig);
154 }).add;
155 )

```

Tabla A-1 – Colección de sintetizadores.

## Anexo B

```
1 //////////////////////////////////////////////////
2 // Configuración de server
3 //////////////////////////////////////////////////
4 (
5 t = TempoClock.new(72/60).permanent_(true);
6 //72/60 // tempo inicial 1.2 bpm
7
8 p = ProxySpace.new(s, clock: t);
9 p.push;
10 p.quant;
11 )
12
13 //////////////////////////////////////////////////
14 /// Extensiones y quarks
15 //////////////////////////////////////////////////
16
17 // StageLimiter.activate; //descomentar si está instalado el limitador
18 // SpectrogramWindow.new; //descomentar si está instalado el espectrograma
19
20 ///////////////
21 //Patrones
22 ///////////////
23
24 (
25 ~reson = Pbind(
26   \instrument, \reswhite,
27   \dur, Prand([1/8,1/4], inf),
28   \note, Pseq([1,3,5,7,9],inf),
29   \pan, Pwhite(-0.5,0.5,inf),
30   \amp, Pxrand([0.0,0.08,0.1,0.5,0.7,0.9],inf)
31 )
32 )
33
34
35 (
36 ~klang = Pbind(
37   \instrument, \klang2,
38   \dur, Pxrand([1],4),
39   \n, Pwhite(0.2,8,inf),
40   \pan, Pwhite(-1,1,inf),
41   \amp, Pwhite(0.01,0.07, inf) * 0.2
42 )
43 )
44
45
46 (
47 ~blips = Pbind(
48   \instrument, \blips,
49   \dur, Prand([1/8,1/4,1/2],inf),
50   \note, Pseq([
51     Pxrand([1,3,6], 24),
52     Pxrand([2,8,9], 24)], inf) + [0.0,0.03] - 10,
```

```

53     \harm, Pxrand([1,3,5,9,10,12,15], inf) + 2,
54     \atk, 0.01,
55     \rel, Pxrand([0.1,0.2,0.3],inf),
56     \curve, -4.0,
57     \pan, Pwhite(-0.8,0.8, inf),
58     \amp, Pwhite(0.2,0.6, inf) * 0.7
59 )
60 )
61
62 (
63 ~sq2b = Pbind(
64     \instrument, \twobandsq,
65     \dur, Pxrand([1/2],inf),
66     \note, Pseq([1,3,7,9,12],inf) + [0.1,0.01,0.001] - 15,
67     \rq, Pwhite(0.04,0.06,inf) * 0.01,
68     \d, Pwhite(0.01,1,inf), //entre alto más agudo
69     \f, Pwhite(0.3,0.5,inf), // al pasar los 1000 empieza a dar problemas
70     \n, Pwhite(0.1,50,inf),
71     \pan, Pwhite(-1,1, inf) ,
72     \amp, Pwhite(0.2,0.6, inf) * 0.02
73 )
74 )
75
76 (
77 ~bassdrumgpt = Pbind(
78     \instrument, \bassdrumgpt,
79     \note, Pxrand([4,1,0,-3,-5,-7,-10,-14,-15,-17,-20],inf) + [0.0,0.01] - 20,
80     \atk, 0.05,
81     \rel, 0.05,
82     \dur, Prand([1/8],inf),
83     \pan, Pwhite(0.0,1.0,inf),
84     \amp, Pwhite(5,10) * 1.2
85 )
86 )
87
88
89 (
90 ~caja = Pbind(
91     \instrument, \drum2,
92     \dur, Pseq([1/8],40),
93     \mix, Pwhite(0.1,0.7,inf),
94     \room, 0.5,
95     \atk, 0.001,
96     \rel, 0.1,
97     \pan, Prand([-0.8,0.8],inf),
98     \amp, Pwhite(0.01,1.5, inf)
99 )
100 )
101
102
103 //////////////////////////////////////////////////
104 // Cambio de tempo
105 //////////////////////////////////////////////////
106
107 t.tempo = 1.2;
108
109
110 //////////////////////////////////////////////////
111 // Liberar variables (fadeTime)
112 //////////////////////////////////////////////////
113 ~reson.free(2);

```

114	~klang.free(2);
115	~blips.free(10);
116	~sq2b.free(2);
117	~bassdrumgpt.free(5);
118	~caja.free(2)

Tabla B-1 Código para improvisar con Tabla A-1.

## Anexo C

```
1  //////////////////////////////////
2  // Setup server
3  //////////////////////////////////
4
5  (
6  Buffer.freeAll;
7  ServerTree.removeAll;
8  ServerQuit.removeAll;
9  Tdef.removeAll;
10 t = TempoClock.new(72/60).permanent_(true);
11 p = ProxySpace.new(s, clock: t);
12 p.push;
13 // StageLimiter.activate; //protección: el volumen no excede los 0dB (activar si
14 se tiene la extensión)
15
16 s.reboot;
17 s.waitForBoot;
18 )
19
20
21 //////////////////////////////////
22 // Samples
23 // (ojo con el path)
24 //////////////////////////////////
25
26 (
27 d = Dictionary.new;
28 d.add(\tarjeta -> PathName("C:/Users/diego/OneDrive/Escritorio/apuntes
29 uv/etnografia 2/samples_metro/bceptarjeta/").entries.collect({
30   arg sf;
31   Buffer.read(s, sf.fullPath);
32 });
33 );
34
35 d.add(\tarjeta_mono -> PathName("C:/Users/diego/OneDrive/Escritorio/apuntes
36 uv/etnografia 2/samples_metro/bceptarjeta/mono/").entries.collect({
37   arg sf;
38   Buffer.read(s, sf.fullPath);
39 });
40 );
41
42 d.add(\metrollega -> PathName("C:/Users/diego/OneDrive/Escritorio/apuntes
43 uv/etnografia 2/samples_metro/metro_llega/").entries.collect({
44   arg sf;
45   Buffer.read(s, sf.fullPath);
46 });
47 );
48
49 d.add(\aire -> PathName("C:/Users/diego/OneDrive/Escritorio/apuntes
50 uv/etnografia 2/samples_metro/aire/").entries.collect({
51   arg sf;
52   Buffer.read(s, sf.fullPath);
53 });
```

```

54 );
55
56 d.add(\aire_mono -> PathName("C:/Users/diego/OneDrive/Escritorio/apuntes
57 uv/etnografia 2/samples_metro/aire/mono").entries.collect({
58   arg sf;
59   Buffer.read(s, sf.fullPath);
60 });
61 );
62
63 d.add(\beepmetro -> PathName("C:/Users/diego/OneDrive/Escritorio/apuntes
64 uv/etnografia 2/samples_metro/beepmetro").entries.collect({
65   arg sf;
66   Buffer.read(s, sf.fullPath);
67 });
68 );
69 d.add(\beepcp -> PathName("C:/Users/diego/OneDrive/Escritorio/apuntes
70 uv/etnografia 2/samples_metro/beepcierrepuerta").entries.collect({
71   arg sf;
72   Buffer.read(s, sf.fullPath);
73 });
74 );
75 d.add(\beepcp_mono -> PathName("C:/Users/diego/OneDrive/Escritorio/apuntes
76 uv/etnografia 2/samples_metro/beepcierrepuerta/mono").entries.collect({
77   arg sf;
78   Buffer.read(s, sf.fullPath);
79 });
80 );
81
82 d.add(\beepap -> PathName("C:/Users/diego/OneDrive/Escritorio/apuntes
83 uv/etnografia 2/samples_metro/beepabrepuerta").entries.collect({
84   arg sf;
85   Buffer.read(s, sf.fullPath);
86 });
87 );
88 d.add(\beepap_mono -> PathName("C:/Users/diego/OneDrive/Escritorio/apuntes
89 uv/etnografia 2/samples_metro/beepabrepuerta/mono").entries.collect({
90   arg sf;
91   Buffer.read(s, sf.fullPath);
92 });
93 );
94 d.add(\personas -> PathName("C:/Users/diego/OneDrive/Escritorio/apuntes
95 uv/etnografia 2/samples_metro/personas").entries.collect({
96   arg sf;
97   Buffer.read(s, sf.fullPath);
98 });
99 );
100 d.add(\personas_mono ->
101 PathName("C:/Users/diego/OneDrive/Escritorio/apuntes uv/etnografia
102 2/samples_metro/personas/mono").entries.collect({
103   arg sf;
104   Buffer.read(s, sf.fullPath);
105 });
106 );
107 d.add(\voz_metro -> PathName("C:/Users/diego/OneDrive/Escritorio/apuntes
108 uv/etnografia 2/samples_metro/voz_metro").entries.collect({
109   arg sf;
110   Buffer.read(s, sf.fullPath);
111 });
112 );
113 d.add(\voz_metro_mono ->
114 PathName("C:/Users/diego/OneDrive/Escritorio/apuntes uv/etnografia

```

```

115 2/samples_metro/voz_metro/mono").entries.collect({
116   arg sf;
117   Buffer.read(s, sf.fullPath);
118 });
119 );
120 d.add(\freno_mono -> PathName("C:/Users/diego/OneDrive/Escritorio/apuntes
121 uv/etnografia 2/samples_metro/freno/mono").entries.collect({
122   arg sf;
123   Buffer.read(s, sf.fullPath);
124 });
125 );
126 )
127 d[\aire].play;
128
129 ////////////////
130 // SynthDefs
131 ////////////////
132
133 (
134 SynthDef.new(\bufplay, {
135   arg buf=0, rate=1, amp=1, pan=0;
136   var sig;
137   sig = PlayBuf.ar(2, buf, BufRateScale.ir(buf) * rate, doneAction:
138 Done.freeSelf);
139   sig = Pan2.ar(sig, pan, amp);
140   Out.ar([0,1], sig);
141 }).add;
142
143 SynthDef.new(\granoise, {
144   arg b, n;
145   var trate, dur, clk, pos, pan, sig;
146   trate = MouseY.kr(40,120,1) * n;
147   dur = 12 / trate;
148   clk = Impulse.kr(trate);
149   pos = MouseX.kr(0,BufDur.kr(b)) + TRand.kr(0, 0.01, clk);
150   sig = TGrains.ar(4, clk, b, 1, pos, dur, ClipNoise.kr(0.5), \amp.kr);
151   Out.ar([0,1], sig)
152 }
153 ).add;
154
155 SynthDef(\reswhite, {
156   var sig, env;
157   sig = ClipNoise.ar;
158   env = XLine.kr(1, 0.0001, \dur.kr(8), doneAction: Done.freeSelf);
159   sig = Resonz.ar(sig, \freq.kr(2000), env);
160   sig = Pan2.ar(sig, \pan.kr(0.0), \amp.kr(0.2));
161   Out.ar(0, sig)
162 }).add;
163
164 SynthDef.new(\bass, {
165   | gate=1, curve=-8.0 |
166   var sig, env;
167   sig = SinOsc.ar(\freq.kr(100));
168   env = EnvGen.kr(Env.perc(\atk.kr(0.001), \rel.ir(1), curve: curve), gate,
169 doneAction: Done.freeSelf);
170   sig = sig * env;
171   sig = FreeVerb.ar(sig, \mix.kr(0.33), \room.kr(0.5), \damp.kr(0.5));
172   sig = Pan2.ar(sig, \pan.ir(0.0), \amp.ir(0.3));
173   Out.ar(0, sig)
174 }).add;
175
176

```

```

177 SynthDef.new(\blips, {
178   var sig, env;
179   sig = Blip.ar(\freq.kr(200), \harm.kr(30));
180   env = EnvGen.kr(Env.perc(\atk.kr(0.001), \rel.kr(0.5)), doneAction:
181   Done.freeSelf);
182   sig = sig * env;
183   sig = Pan2.ar(sig, \pan.kr(0.0), \amp.kr(0.2));
184   Out.ar(0, sig)
185 }).add;
186
187 SynthDef.new(\twobandsq, {
188   arg n, d, f;
189   var sig, env;
190   //env = XLine.kr(8000,200,2, doneAction: Done.freeSelf);
191   env = EnvGen.kr(Env([8000,1000,50]), doneAction: Done.freeSelf) * n;
192   sig = Impulse.ar([100,250] * d);
193   sig = RLPF.ar(sig, env * f, \rq.kr(0.05));
194   sig = Pan2.ar(sig, \pan.kr(0.0), \amp.kr(0.2));
195   Out.ar(0, sig)
196 }).add;
197 )
198
199 ///////////////////////////////////////////////////
200 ///datos de los audios
201 ///////////////////////////////////////////////////
202
203 d.atAll(d.order);
204 d.order;
205 d[\voz_metro].at(0).duration;
206 d[\voz_metro].at(0).numFrames;
207 d[\voz_metro].at(0).bufnum;
208 d[\voz_metro].at(0).sampleRate;
209
210
211 (
212 ~tarjeta = Pbind(
213   \instrument, \bufplay,
214   \dur, Pseq([2],inf),
215   \buf, Pseq([d[\tarjeta]],1),
216   \rate, Prand([1.5,1,0.5],inf),
217   \amp, Pseq([0.3],inf)
218 )
219 )
220 ~tarjeta.free(3);
221
222 (
223 ~personas = Pbind(
224   \instrument, \bufplay,
225   \dur, Pseq([1],inf),
226   \buf, Pseq([d[\personas].choose],5),
227   \rate, Prand([1,1.3,0.8],inf),
228   \amp, Pseq([0.5],inf)
229 )
230 )
231 ~personas.free(3);
232
233 (
234 ~personas = Pbind(
235   \instrument, \bufplay,
236   \dur, Pseq([2],inf),
237   \buf, Pseq([d[\personas].at(12)],3),

```

```

238     \rate, Prand([1],inf),
239     \amp, Pseq([0.5],inf) * 2
240 )
241 )
242 ~personas.free(3);
243
244 (
245 ~voz_metro = Pbind(
246     \instrument, \bufplay,
247     \dur, Pseq([2],inf),
248     \buf, Prand([d[\voz_metro].choose],1),
249     \rate, Pxrand([0.8,1.0,1.3,1.5],inf),
250     \amp, Pseq([0.6],inf)
251 )
252 )
253 ~voz_metro.free(3);
254
255 (
256 ~freno = Pbind(
257     \instrument, \granoise,
258     \dur, Pseq([1],1),
259     \n, 1,
260     \b, Pseq([d[\freno_mono].at(0)],1),
261     \rate, Pseq([0.2],inf),
262     \amp, Pseq([0.2],inf)
263 )
264 )
265 ~freno.free(2);
266 t.tempo = 1.2
267 s.freeAll(5);
268
269 (
270 ~beepap = Pbind(
271     \instrument, \bufplay,
272     \dur, Pseq([2],inf),
273     \buf, Pseq([d[\beepap].at(1)],2),
274     \rate, Prand([0.6],inf),
275     \amp, Pseq([0.5],inf)
276 )
277 )
278 ~beepap.free(3);
279
280 (
281 ~beepcp = Pbind(
282     \instrument, \bufplay,
283     \dur, Pseq([2],inf),
284     \buf, Pseq([d[\beepcp].at(2)],1),
285     \rate, Prand([0.5],inf),
286     \amp, Pseq([0.5],inf)
287 )
288 )
289 ~beepcp.free(3);
290
291 (
292 ~bass = Pbind(
293     \instrument, \bass,
294     \dur, Pxrand([1],inf),
295     \note, Pseq([0],inf) + [-0.1,0.25] - 28,
296     \pan, 0.0,
297     \amp, Pseq([1.0,0.5], inf) * 0.09
298 )

```

```

299 )
300 ~bass.free(2)
301
302 (
303 ~reson = Pbind(
304   \instrument, \reswhite,
305   \dur, Pxrand([1/8,1/8,1/4], inf),
306   \note, Prand([1,3,5,7,9],inf) - 25,
307   \pan, Pwhite(-0.5,0.5,inf),
308   \amp, Pxrand([0.2,0.5,0.7,0.9],inf) * 0.5
309 )
310 )
311 ~reson.free(5);
312
313 (
314 ~blips = Pbind(
315   \instrument, \blips,
316   \dur, Prand([1/8],inf),
317   \note, Pxrand([2], inf) + [-0.15,0.15] - 20,
318   \harm, Pxrand([1,3,5,9,10,12,15], inf) + 5,
319   \atk, 0.001,
320   \rel, 0.15,
321   \curve, -12,
322   \pan, Pwhite(-0.8,0.8, inf),
323   \amp, Pwhite(0.2,0.6, inf) * 0.1
324 )
325 )
326 ~blips.free(2);

```

Tabla C-1 Código de la pieza Cibermetroval.

## INFORME DE TESIS DE GRADO

Pablo Palacios Torres Profesor Guía

1. Identificación del alumno y Carrera. Nombre : Diego Henríquez

Carrera : Música

2. Título de la tesis de grado.

“CREACIÓN DE PIEZAS DE COMPUTER MUSIC EN SUPERCOLLIDER”

3. Contenido de cada una de sus partes o capítulos.

Introducción (pregunta de investigación y objetivos).

La problemática está construida sobre preguntas pertinentes y problematizadoras, cumple con la suficiencia metodológica para llevar a cabo este tipo de trabajo. Los objetivos son acorde a las preguntas y están bien diseñados. La redacción de la introducción es acorde a lo que se pide en un trabajo de pregrado. Interesante es el marco teórico, el cual denota una aproximación profunda a los conceptos de computer music y el uso de programas como supercollider. Esto posibilita poder comprender de mejor forma el rol del creador/programador en tiempo real.

En el apartado metodológico podemos encontrar una definición adecuada al tipo de trabajo realizado, particularmente la autoetnografía, la que el estudiante pudo seguir y ordenar su proceso creativo poético.

Capítulos 1, 2 y 3

Responde a la descripción de los conceptos y paradigma creativo al que se adscribe el tesista, además sirven de marco teórico al lector, ya que responde a una breve introducción a los problemas y creadores del área de computer music.

Quisiera destacar la significativa revisión bibliográfica del postulante, ya que da cuenta de un excelente estado del arte y del conocimiento de las obras y creadores clásicos del género a tratar. En el caso de poder mejorar esto, se podría optar por la utilización de mayor cantidad de elementos visuales que orienten la comprensión de un contenido eminentemente técnico.

## Capítulo 4

Este capítulo presenta la descripción de las obras que serán creadas desde el live coding, esto es la codificación como arte o creación. En esta parte del trabajo se observa un cuidado con dos cosas, una de ellas conocer los referentes sonoros, para esto el autor utiliza la autoetnografía, la que desde la experiencia de Rubén López Cano sirve para conocer de mejor forma la aproximación a la tradición creativa de la especialidad, a la escucha racional del creador.

A pesar que el lenguaje de la codificación es de una naturaleza críptica para no iniciado en este lenguaje, se puede comprender gracias a la descripción del proceso la sucesión de eventos y decisiones que se debieron tomar para diseñar (codificar) los sonidos.

Cabe destacar las páginas finales sobre la cuestión teórica del ciber espacio, creo que desde el punto de vista de las humanidades es un aporte interesante poder problematizar estos conceptos dada la naturaleza de la vida y el conocimiento actual. Las páginas dedicadas a este tema son a mi modo de ver magistrales, ya que revelan las distintas concepciones sobre este tan usado concepto.

## Conclusiones

Las conclusiones responden perfectamente a la naturaleza de los objetivos, metodología y temas planteados. Por otra parte, creo que se debe mencionar que la tesis presenta en el espacio creativo latinoamericano una excelente trabajo que acerca este tipo de práctica, más bien anglosajona, a nuestra cultura. Una de la conclusiones no planteadas, pero relevantes, es si es posible pensar en un desarrollo del live coding con tintes latinos, ¿es posible desarrollar la computer music desde nuestro continente?

---

### 4. Importancia, originalidad e interés profesional o cultural del trabajo.

Se valora: realizar una investigación en arte a través de la auto etnografía en una espacio de conocimiento nuevo para el mundo de la música chilena. El trabajo es un real aporte al conocimiento del tema.

### 5. Amplitud, suficiencia y racionalidad crítica en el tratamiento del tema.

La amplitud es suficiente para una tesis de pregrado, presenta un excelente nivel de sistematización de la información, y su grado de racionalidad científica sobresaliente. Se destaca una óptima redacción.

**6. Método de investigación, fuentes de información empleadas, consistencia lógica en el desarrollo de la tesis, redacción y uso adecuado de la terminología especializada.**

La consistencia lógica del trabajo es excelente, hay un desarrollo de la tesis ad hoc a pregrado, uso de términos propios de la especialidad, tanto en inglés como en castellano, además de una excelente revisión de bibliografía.

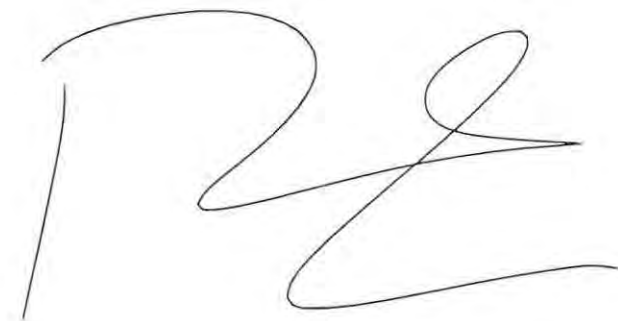
**7. Contribución en cuanto a nuevas ideas o hipótesis de trabajo que sea necesario destacar.**

Los objetivos son acordes al nivel de pregrado, pero cabe destacar que el trabajo abre un campo de posibilidades de desarrollo a nivel de posgrado al estudiante. Por otra parte, plantea el tema de la computer music y live coding en el marco de la composición/creación, lo que genera una apuesta alta en el marco de la tradición chilena.

**8. Conclusiones y calificación de acuerdo con la escala de notas vigente.**

Las conclusiones son acordes al trabajo y revelan un interés real y profundo por el tema planteado.

**NOTA: 7,0 (siete como cero), en escala de uno a siete.**

A handwritten signature in black ink, consisting of a large, stylized 'P' followed by a series of loops and a long horizontal stroke at the end.

**Pablo Palacios Torres.**

## INFORME DE TESIS

Carrera de Música  
Instituto de Filosofía  
Facultad de Humanidades y Educación  
Universidad de Valparaíso



### Nombre del Estudiante

DIEGO ANDRÉS HENRÍQUEZ NEIRA

### Título de la Tesis

CREACIÓN DE PIEZAS DE COMPUTER MUSIC EN SUPERCOLLIDER

La introducción muestra un objetivo general que es la creación de obras que se enmarquen en el género de Computer Music y dos objetivos específicos. Muestra claramente la fundamentación de la investigación y la metodología que se utilizará en la investigación y creación.

El capítulo uno está centrado en el computador como instrumento, como búsqueda interdisciplinaria, como múltiples preocupaciones estéticas y Max Mathews. En este capítulo profundiza sobre los antecedentes del Computer Music. El capítulo dos nos habla de la composición algorítmica, donde destaco el trabajo realizado con Cemla en el uso de los sensores y las telas. El capítulo tres nos relata sobre el Software SuperCollider, sus potencialidades, sus elementos técnicos, sus usos.

El desarrollo, está determinado por el capítulo cuatro, que relata la creación de piezas musicales para Computer Music. Este capítulo muestra una creación realizada en un trayecto definido por el metro tren Villa Alemana – Valparaíso y como este recorrido realizado ha ido cambiando con el tiempo, sumando y suprimiendo sonidos. Esto permite generar un relato artístico estableciendo una serie de procedimientos técnicos y poéticos a la obra.

La Conclusión da respuesta a los objetivos trazados y deja las puertas abiertas a futuras investigaciones en el tema. Creo que esta tesis es un gran aporte e invita a sumergirse a nuevas maneras de crear, analizar y establecer relatos artísticos desde nuestro cotidiano tecnológico.

Nota: seis coma cinco (6,5)

A handwritten signature in black ink, appearing to read 'Paul Hernández'.

Paul Hernández Mendoza  
Profesor Informante