



Universidad Internacional de La Rioja
Máster Oficial en Investigación Musical

Composición algorítmica de una pieza de metal instrumental desde el análisis musical informatizado

**Trabajo fin de
máster presentado** William Ricardo Vargas Castellanos
por:

Director/a: Daniel Moro Vallina

Bogotá, Colombia
25 de enero de 2017
Firmado por: William Ricardo Vargas Castellanos

A handwritten signature in black ink, which appears to read "William", is positioned below the text. A horizontal line is drawn underneath the signature.

Resumen

En este trabajo se presenta el proceso de composición algorítmica de una pieza de *metal* instrumental desde el análisis musical informatizado, para lo cual se ha dividido el desarrollo en dos fases: análisis y composición. Para la primera fase, el análisis, se parte de la teoría neo-Riemanniana la cual es una teoría transformacional que se ha encontrado apropiada para el análisis del género. A partir del análisis informatizado de tres piezas representativas de *metal* instrumental de la banda Metallica se aplican las operaciones neo-Riemannianas que conducen de un acorde a otro en las guitarras. Se hace un recuento de estas para poder generar cadenas de Markov de primer orden, que representan la probabilidad de aplicar una operación de transformación básica o compuesta.

Para la segunda fase, la composición, se ha elegido una forma musical derivada del estudio de las canciones de Metallica y se han generado progresiones de acordes automáticamente usando las cadenas de Markov resultantes del análisis. Las duraciones de los acordes son decididas por el compositor entre patrones creados manualmente y se automatiza de forma muy básica el acompañamiento del bajo eléctrico, siguiendo la nota más baja de cada acorde. La pieza generada es una en particular de entre varias ejecuciones del software pues por su construcción resulta en un algoritmo no determinista.

Se constata que es posible generar la pieza que se perseguía pero no es suficiente con modelar las progresiones de acordes, hace falta la integración de otros conceptos musicales. Aún cuando la composición es algorítmica, el papel del compositor es importante en decisiones estructurales y sobre todo en darle una intención al proceso compositivo.

Palabras Clave: composición algorítmica, cadenas de Markov, teoría neo-Riemanniana, Metal Instrumental, Análisis musical informatizado.

Abstract

This work presents the algorithmic composition process of an instrumental metal song based on computer music analysis. Development has been divided into two phases: analysis and composition. The analysis phase is based on Neo-Riemannian Theory, which is a transformational theory that seems appropriate for the analysis of the genre. Three representative songs of the band Metallica are analyzed to find neo-Riemannian operations that transform one chord into another for the guitars. These operations are accounted for and first order Markov chains are constructed, these chains represent the transition probability from one chord to another by applying basic or compound operations.

For the second phase, composition, a musical form found in the songs analyzed is used and chord progressions are generated automatically using the resulting Markov chains from the analysis. Chord durations are decided by the composer among manually created patterns. Bass accompaniment is automated in a very basic way, following the lowest note of each chord. A generated song is presented, which is one of many others, selected among different software executions as the algorithm is non-deterministic.

It is found that it is possible to generate the song wanted but is not enough to model chord progressions, the integration of musical concepts it's also needed. Although the composition is algorithmic, the role of the composer is important in structural decisions and more important in giving an intention to the composition process.

Keywords: algorithmic composition, Markov chains, Neo-Riemannian Theory, Instrumental Metal, Computer Music Analysis.

Agradecimientos

Agradezco a la Universidad Internacional de la Rioja y a las directivas del Máster en Investigación Musical por darme la oportunidad de participar de estos estudios.

Doy las gracias también a mi director, Daniel Moro Vallina, por su siempre oportuna y juiciosa revisión y su acertada guía en cómo abordar y desarrollar este proyecto.

*A Gina, mi esposa,
por su apoyo incondicional y motivación cuando casi tuve que abandonar el proyecto.*

*A Verónica y Emilio, mis pequeños hijos,
por sus sonrisas y rostros de sorpresa al escuchar la pieza final.*

Índice

<i>Resumen</i>	2
<i>Abstract</i>	3
<i>Agradecimientos</i>	4
<i>1 Introducción</i>	9
1.1 Justificación y problema	9
1.2 Objetivos generales y específicos	10
<i>2 Marco Teórico</i>	11
2.1 El género metal	11
2.1.1 Breve recorrido histórico	11
2.1.2 Características e instrumentos	12
2.2 Análisis musical informatizado y Teoría neo-Riemanniana	14
2.3 Composición musical algorítmica	18
2.3.1 Breve recorrido histórico	18
2.3.2 Técnicas de composición algorítmica	19
<i>3 Análisis</i>	23
<i>4 Composición</i>	32
<i>5 Discusión y Conclusiones</i>	37
5.1 Limitaciones	38
5.2 Prospectiva	39
<i>6 Referencias bibliográficas</i>	40
<i>7 Anexos</i>	43
7.1 Matrices de transición	43
7.2 Discografía	49
7.3 Desarrollo informático	51
7.4 Partitura	52

Índice de figuras

Figura 1: Operaciones P, R, L en C mayor	14
Figura 2: Operaciones de transformación en el <i>tonnetz</i>	15
Figura 3: <i>Tonnetz</i> con nombres de clases de alturas	16
Figura 4: Matriz y grafo de transición para una cadena de Markov	21
Figura 5: Forma musical. <i>The call of Ktulu</i>	23
Figura 6: Forma musical. <i>Orion</i>	23
Figura 7: Forma musical. <i>To live is to die</i>	23
Figura 8: Distribución de tonos. <i>The call of Ktulu</i>	24
Figura 9: Distribución de duraciones. <i>The call of Ktulu</i>	24
Figura 10: Distribución de tonos. <i>Orion</i>	25
Figura 11: Distribución de duraciones. <i>Orion</i>	25
Figura 12: Distribución de tonos. <i>To live is to die</i>	26
Figura 13: Distribución de duraciones. <i>To live is to die</i>	26
Figura 14: Análisis desde transformaciones neo-Riemannianas	27
Figura 15: Cálculo para la transición en una matriz	31
Figura 16: Fragmento de <i>Experimento 0</i>	32
Figura 17: Fragmento de <i>Experimento 1</i>	32
Figura 18: Fragmento de <i>Experimento 2</i>	33
Figura 19: Fragmento de <i>Experimento 3</i>	34

Índice de tablas

Tabla 1: Nombres de conjuntos de clases de alturas	16
Tabla 2: Matriz de recuento de transiciones	28
Tabla 3: Matriz de transición	28
Tabla 4: Conversión de transformaciones compuestas en básicas	28
Tabla 5: Matriz de transición entre clases de alturas	29
Tabla 6: Matriz de transición entre operaciones compuestas	29
Tabla 7: Matriz de transición entre operaciones básicas	30
Tabla 8: Patrones de duración para la composición	34
Tabla 9: Estructura formal de la composición	35

1 Introducción

1.1 Justificación y problema

El *rock* en general y el *metal* en particular son géneros poco estudiados desde el punto de vista del análisis musical, para Lilja (2004) los estudios existentes están más orientados hacia aspectos sociales y culturales. El enfoque de este trabajo es práctico y se dirige al uso del análisis musical informatizado como base para una composición algorítmica, en nuestro caso no nos acercamos a esta buscando sonidos novedosos o experimentales sino buscando reglas que nos permitan construir un discurso musical coherente dentro del género.

El trabajo se enmarca dentro de la composición asistida por computador, pues las decisiones estructurales son libertad del compositor mientras que las decisiones de más detalle son calculadas con algoritmos no deterministas. Los roles en este caso son diversos, el compositor ejerce de programador y músico, de crítico y espectador de su propia obra. El rol de la computadora también es diverso, asiste tanto en el análisis como en la composición, permite un manejo de altos volúmenes de datos, realiza la transformación entre diferentes representaciones y ejerce de intérprete de una obra digital.

La aproximación de este trabajo es similar a lo presentado en ASSA (*Automatic Style-Specific Accompaniment*) (Chuan y Chew, 2011, citado en Moro, 2016c) en donde se busca generar automáticamente el acompañamiento para una melodía de rock usando cadenas de Markov junto con otras técnicas. También se acerca al trabajo de Padilla (2012) en el que se exploran técnicas de composición algorítmicas para generar ideas musicales para piezas de corte más clásico.

Ahora bien ¿para qué querríamos investigar en composición algorítmica? ¿Es esta una investigación en arte? ¿En tecnología? Esta sociedad tiende a exagerar el valor de la tecnología y ha olvidado la importancia de las artes y las humanidades. En nuestro mundo tecnificado quizás esta investigación pueda resaltar la importancia de las artes, paradójicamente a través de la tecnología y, más que el resultado o las técnicas, ojalá este trabajo sirva para mostrar nuevos caminos profesionales entre la dura competencia y la falta de oportunidades en nuestras sociedades, lo que es el fin último de la investigación: impactar positivamente en la vida de las personas.

Esperamos que este proyecto aporte a aquellos que se acercan desde el área musical, mostrando posibles usos del análisis musical informatizado y de la composición algorítmica y también, a aquellos que se aproximan desde el área informática, presentando áreas alternativas de trabajo e investigación dentro del desarrollo de software.

Este documento está estructurado de la siguiente manera: en la primera parte se exponen además de esta introducción los objetivos generales y específicos que guían el desarrollo del proyecto. En la segunda parte se presenta el marco teórico en el que se hace un corto recorrido histórico por el género *metal* y se presentan sus características musicales. Se expone también el análisis musical informatizado y la teoría neo-Riemanniana y finalmente conceptos de composición algorítmica y cadenas de Markov.

En la tercera parte se describe el proceso de análisis informatizado de un pequeño corpus de estudio que es usado para la cuarta parte, en donde se presenta el proceso de composición de la pieza. En la quinta parte se exponen las conclusiones, limitaciones y trabajo futuro. Para acompañar el trabajo se presentan anexos con la partitura de la pieza, una descripción del desarrollo informático, las cadenas de Markov representadas como matrices de transición y una breve discografía de piezas representativas de *metal* instrumental.

1.2 Objetivos generales y específicos

General

Componer una pieza musical del género *metal* instrumental desde el análisis musical informatizado y la composición algorítmica.

Específicos

- Realizar un breve recorrido histórico sobre el género metal y presentar sus características musicales más importantes.
- Aplicar técnicas de análisis musical informatizado sobre piezas representativas del género, apoyadas en un modelo teórico de análisis musical.
- Explorar diferentes técnicas de composición algorítmica para elegir una o varias susceptibles de implementar informaticamente.
- Presentar una aplicación práctica del análisis musical informatizado y de la composición algorítmica en un género de música popular.

2 Marco Teórico

El marco teórico de esta investigación cubre tres aspectos. Por una parte se fundamentará desde una perspectiva más musical, haciendo un breve recorrido por el metal y mencionando algunos conceptos sobre su instrumentación y estructura musical. Por otra parte se expondrán conceptos y técnicas del análisis musical desde la Teoría neo-Riemanniana. Finalmente se mencionarán algunas aproximaciones a la composición algorítmica para profundizar en las cadenas de Markov.

2.1 El género metal

Los límites por los cuales una música se puede clasificar como rock o metal son muy difusos pues ni las mismas bandas ni sus seguidores tienen un criterio único. Para Berger el metal es un grupo de géneros musicales que emplean guitarras distorsionadas, muestras de virtuosismo individual y formas complejas en sus canciones (1999, citado en Lilja, 2004, p. 5). El metal se aparta del rock y se caracteriza porque abandona la influencia directa del blues (Lilja, 2004).

2.1.1 Breve recorrido histórico

A continuación trazaremos los inicios y desarrollo del género metal y sus principales exponentes, para a continuación describir en qué consiste el metal instrumental y sus principales rasgos desde el punto de vista de la técnica musical, objetivo de este trabajo. El siguiente recorrido histórico es traducido y resumido de Pearlin (2014).

El metal tiene sus primeras semillas en los años 60 dentro de una gran explosión de música popular. Bandas como The Beatles, The Rolling Stones, The Who o The Kinks contribuyeron a formar el prototipo de lo que sería una banda de rock. Pero son bandas como Cream, Led Zeppelin y The Jimi Hendrix Experience las que serían la semilla del metal, con sonidos disruptivos, letras socialmente agresivas y altos volúmenes en sus amplificadores.

En Inglaterra es donde podemos ubicar el nacimiento del metal con Black Sabbath en los años 70, presentando un sonido oscuro y amenazante contrario a la música pop de la época y unas letras que tocaban temas tabú como la corrupción política, las drogas y la exclusión social. A finales de esta misma década surge Judas Priest que combina la oscuridad e intensidad de Black Sabbath con la calidad musical y complejidad de Deep Purple. Judas Priest aporta un metal a la vez rítmico y melódico y la ahora conocida estética metal de cuero y tachas.

En los años 80 aparece lo que se ha denominado la nueva ola del metal británico con bandas como Iron Maiden, Motörhead, Saxon y Diamond Head. En estas es menos notoria la influencia del blues y sí la del punk británico; sus letras, por otra parte, exploran temas fantásticos y mitológicos. A finales de esta década surgen dos aproximaciones al metal en Norteamérica, por una parte surge el *hair metal* con un sonido simplificado, más comercial y de puro espectáculo con letras sobre las fiestas y la buena vida. Ejemplos son Poison, Mötley Crüe y RATT. Por otra parte surge una postura totalmente opuesta: el *thrash metal*. Bandas como Metallica, Exodus, Testament, Megadeth, Slayer, Metal Church y Annihilator presentan un metal más rudo, rápido y ruidoso. Siendo un

metal más rítmico que melódico su enfoque es la ejecución veloz de complicados riffs y sus letras giran alrededor del descontento social y la crítica política. En esta época el metal gana popularidad en Europa, Norte y Sur América.

Durante los 90 el *hair metal* pierde fuerza al igual que el *thrash metal* audiencias. A la vez crece el grunge que, con un sonido más simple y letras sobre el desencanto y la angustia opaca al metal. Sólo bandas como Metallica y Megadeth que optan por un sonido más comercial se mantienen, junto con Pantera que realiza una exitosa combinación de metal y *hardcore*. En la escena underground surgen géneros más extremos como el *black metal* y el *death metal*, que tendrían en adelante pocos pero fieles seguidores.

Después del papel poco protagonista del metal en la década de los 90, a principios del 2000 surge la libertad para explorar y es así como surgen varios subgéneros como el *symphonic*, *folk*, *melodic death*, *progressive death*, *technical death* ... la mayor parte de este esfuerzo creativo se da en Europa con bandas como In Flames, Opeth y Therion (Suecia), Nightwish y Children of Bodom (Finlandia) y Dimmu Borgir (Noruega).

Metal instrumental

Presentado este recorrido histórico podemos encontrar al metal instrumental como un enfoque dentro del metal que enfatiza en el uso de los instrumentos musicales y que consecuentemente tienen una nula o mínima participación de la voz. Prácticamente ninguna banda de metal compone únicamente temas instrumentales, más bien incluyen a veces temas instrumentales en sus producciones. Las características de estas piezas instrumentales son tan diversas como el mismo género, desde cortas y melódicas hasta largas y complejas. Invitamos al lector y oyente curioso a explorar el Anexo 2 en el que se mencionan algunas piezas representativas del *metal* instrumental.

2.1.2 Características e instrumentos

Van Valkenburg (2010) expresa que en el metal se resaltan las capacidades musicales y el manejo del instrumento de cada miembro de la banda y esto se refleja en la estructura de las canciones que hacen énfasis en el solo de guitarra, una especie de interludio a mitad de la canción. También menciona otra característica del metal: la velocidad, que puede variar entre 120bpm y 250 bpm. El papel de la voz en el metal es importante y se encuentra entre los extremos desde el canto gutural hasta el canto soprano. No mencionaremos la voz en este trabajo por el enfoque en piezas instrumentales.

La guitarra

El metal se caracteriza por el uso de fuerte distorsión en la guitarra eléctrica, lo que genera un sonido más denso y sostenido que permite la expresión de la guitarra como instrumento rítmico y melódico, a la vez que sirve como metáfora de poder físico (Van Valkenburg, 2010).

Una estructura que se encuentra frecuentemente en el metal, es el llamado *power chord* que consiste en la ejecución de dos o más notas en la guitarra, compuestas de la base y la quinta de un acorde dado y conteniendo sólo intervalos perfectos entre notas. El power chord no tiene por sí cualidad de mayor o menor y difuminan la distinción tradicional entre armonía y melodía (Pillsbury, 2006 citado en Van Valkenburg, 2010). Lilja (2004) y Van Valkenburg (2010) proponen que debido a la distorsión en la guitarras se evita el uso de power chords de más de dos notas, para no saturar el sonido con demasiados componentes armónicos. Para este último, las configuraciones de los power chords también resultan de consideraciones como la comodidad y practicidad en la ejecución.

Otra estructura común en el metal es el *riff*, un “corto y melódico ostinato que puede ser repetido bien intacto o variado para ajustar un patrón armónico subyacente” (*Grove Music Online*, citado en Van Valkenburg, p. 37). En metal un riff puede servir para propósitos tanto melódicos como armónicos. Los riffs en el metal tienden a estar centrados en la cuerda E más baja como tónica para tener el sonido más denso posible.

Una técnica frecuente en la ejecución de guitarra en el metal es el *palm muting*. Este consiste en tocar con la palma de la mano las cuerdas para dar un sonido más percusivo a la guitarra, reduciendo las frecuencias de rango medio. También permite un rápido tiempo de caída que le permite sincronizarse con la batería dando un efecto más denso (Van Valkenburg, 2010).

El bajo

En metal las inversiones de acordes no tienen un papel importante, así que la construcción del bajo se acerca a la práctica del bajo fundamental de Rameau (Lilja, 2004). Bajo esta teoría el acorde se lleva a través de rotaciones a una pila de terceras y se determina la base del acorde como la nota más baja. El bajo entonces interpreta la base del acorde una octava abajo y se usa para definir y mantener el pulso. El papel del bajo es importante pues junto con el bombo dan guía al oyente para que no se pierda dentro de la canción (Starr, 2009). Normalmente el oyente no debe sentir que el bajo tiene la prioridad en la canción excepto en momentos destinados a la muestra de virtuosismo del intérprete, lo cual es común en el metal.

La batería

El metal se basa en patrones básicos del rock para la batería con algunas variaciones. El patrón más típico es la marca del redoblante en los tiempos 2 y 4, en una métrica de 4/4. El bombo marca otros tiempos diferentes y alternos al redoblante y el hi-hat marca continuamente cada cuarto o en algunos casos cada octavo. El hi-hat es un punto claro de referencia métrica y da una sensación de energía y movimiento continuo (Van Valkenburg, 2010).

El platillo crash se usa como punto de referencia para indicar cambios en estructuras de mayor tamaño, por ejemplo finales de secciones y también como acento en algunos pasajes. El sonido del

platillo siempre se hace simultáneo a un golpe de bombo para crear un sonido más completo en el espectro (Van Valkenburg, 2010).

La marca del redoblante crea una sensación de estabilidad, sin embargo esta se debe romper para indicar transiciones o cierres. Estas rupturas se denominan *drum fills* y generalmente después de ellos se regresa al patrón inicial (Van Valkenburg, 2010). Los fills pueden incluir y combinar redoblante, toms y en algunos casos el bombo.

Finalmente una característica básica del metal es el uso de doble pedal o doble bombo en la batería para poder sostener los rápidos tempos (Van Valkenburg, 2010).

2.2 Análisis musical informatizado y Teoría neo-Riemanniana

Análisis musical

Sobre el análisis musical podemos encontrar varias definiciones: “(...) es la resolución de la estructura musical en elementos constituyentes relativamente más simples, y el estudio de la función de estos elementos en dicha estructura” (Bent, 1980, citado en Moro, 2016a p. 1); “(...) es toda segmentación, comentario e interpretación de una obra o de un conjunto homogéneo de obras” (*Die Musik in Geschichte und Gegenwart*, citado en Moro, 2016a p. 1). Para Cook (1999, citado en Moro, 2016a p. 8) el análisis musical es acientífico y no ofrece respuestas universales, sólo una posible interpretación a la construcción de la obra.

Diversos autores plantean varias tipologías para el análisis musical. Bent (1980, citado en Moro, 2016a p. 11) propone tres aproximaciones al análisis. En primer lugar, según la sustancia musical: viendo a la pieza como una sucesión de estructuras. En segundo lugar, según los métodos operativos: que pueden ser reducción, comparación, segmentación y recuento estadístico, entre otros. Finalmente, según los medios de representación: partituras comentadas, listas, gráficos y descripciones verbales.

Solare (2007) propone que los métodos, perspectivas, lenguaje y forma de presentación del análisis musical deben responder a quién lo hace y para qué se hace el análisis, en el caso de este trabajo y siguiendo su propuesta se quiere “analizar para sintetizar, (para) derivar mecanismos que nos conduzcan a componer nuestra propia obra” (p. 46).

Análisis musical informatizado

Moro (2016a) establece que el análisis musical informatizado mejora el trabajo del analista brindándole nuevas posibilidades de representación y permite presentar los datos de manera objetiva de acuerdo con un proceso algorítmico. Con todo, sigue siendo necesaria una interpretación musical, subjetiva, de los datos. Por otra parte resalta algunos problemas del análisis musical informatizado como la dualidad entre la representación simbólica y la señal de audio y la gran variedad de aproximaciones al análisis musical.

Para el presente trabajo nos interesa una de las aplicaciones del análisis musical informatizado: la emulación de estilos. Estos sistemas suelen incluir tanto un análisis estadístico como una infraestructura teórica para la generación automática de acompañamientos (Moro, 2016b).

Prácticamente todas las metodologías de Análisis Musical se basan en la reducción y segmentación de la música en unidades estructurales para su posterior comparación, el objetivo es encontrar patrones recurriendo en algunos casos a métodos estadísticos (Moro, 2016c). Considerando una pieza musical como “un campo de datos en el que buscar modelos” (Igoa, 1999. p. 71), podemos aplicar a su estudio procesos de estadística descriptiva.

Uno de los enfoques del análisis musical es la aproximación transformacional y una de sus teorías es la Teoría neo-Riemanniana. Diversos autores como Capuzzo (2004), Felicity (2013) o Lilja (2004) plantean que es una teoría apropiada para el análisis de rock y pop así que la describiremos a continuación.

Teoría neo-Riemanniana

Hugo Riemann (1814 - 1919) establece los fundamentos de la teoría transformacional al definir acordes triada sin referencia a una clave o registro específico. David Lewin hace visible el trabajo de Riemann con su libro *Generalized Music Intervals and Transformations* (1987). Otros autores como Brian Hyers, Henry Klumpenhouwer y Richard Cohn aportan a la teoría creando lo que se ha denominado la Teoría neo-Riemanniana (Felicity, 2013).

La Teoría neo-Riemanniana¹ aborda el estudio de las progresiones de acordes desde la teoría de grupos en matemáticas en donde a partir de operaciones de transformación es posible encontrar relaciones entre acordes (Felicity, 2013), las operaciones neo-Riemannianas que presentamos son las establecidas por Cohn (1997, citado en Hascher, 2007) estas son tres operaciones de transformación básicas: Paralela (P), Relativa (R) y Cambio de tono líder (L) a partir de las cuales se pueden construir todas las transformaciones originalmente presentadas por Riemann (1880, citado en Hascher, 2007). La transformación paralela convierte una triada mayor en menor y viceversa, moviendo la tercera un semitono. La transformación relativa convierte una triada mayor en menor moviendo la quinta un tono y viceversa o una triada menor en mayor moviendo la base un tono. La transformación de cambio de tono líder convierte una triada mayor en menor moviendo la base un semitono y viceversa. Vemos algunos ejemplos de transformaciones en la Figura 1 (Felicity, 2013, p. 7.)

¹ TNR en adelante.

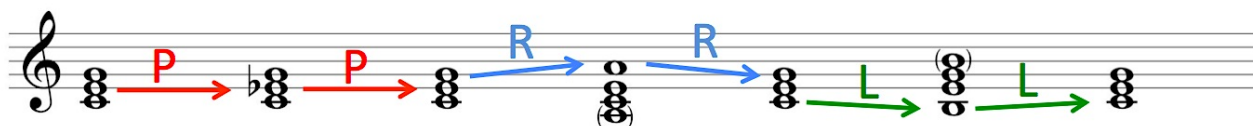


Figura 1: Operaciones P, R, L en C mayor

Cohn (1997, citado en Felicity, 2013) introduce el término parsimonia, que se usa cuando dos acordes comparten dos tonos. Las operaciones neo-Riemannianas son parsimónicas y también son involuciones, lo que significa que al aplicar la misma transformación sobre el resultado da de nuevo el valor original (Felicity, 2013).

El *tonnetz* es una representación geométrica a manera de retícula de tonos interconectados y es útil para presentar gráficamente las operaciones de transformación. El primer *tonnetz* se atribuye a Leonhard Euler en 1739, posteriormente Hostinský presentó una versión inclinada en 1879 y luego Riemann lo adapta en 1880 (Felicity, 2013). El *tonnetz* es una figura de superficie infinita, es decir se conecta con si mismo en todos sus bordes. En la figura 2 podemos ver el *tonnetz* con los nombres de tonos, cada triángulo expresa un acorde y podemos ver un detalle de las operaciones de transformación L, P, R (Felicity, 2013, p. 20.).

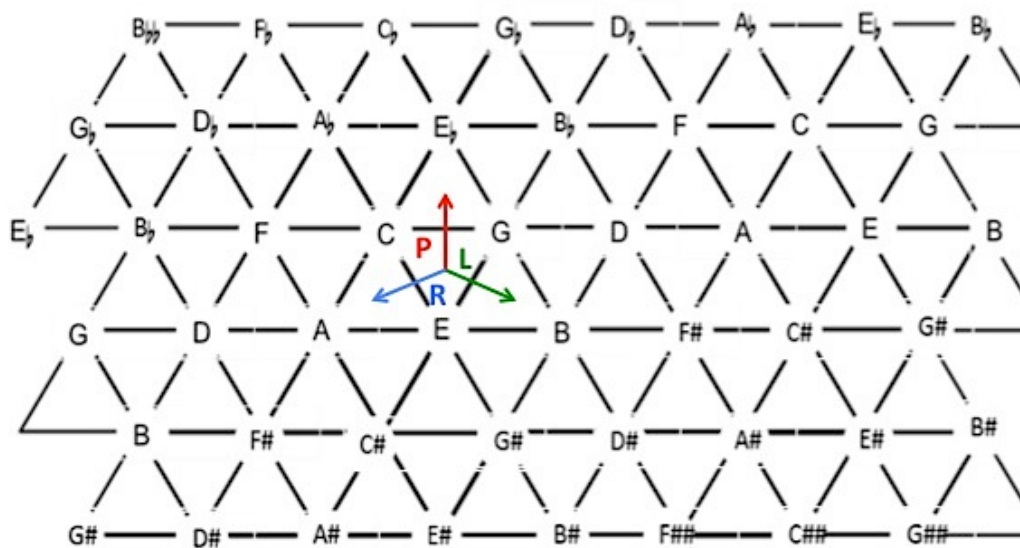


Figura 2: Operaciones de transformación en el tonnetz

En la figura 3 podemos ver un *tonnetz* con nombres de clases de alturas, estas provienen del modelo de análisis de conjuntos de clases de alturas (*Pitch-Class Set Theory*) y simplifica el trabajo

con el *tonnetz* gracias a la equivalencia enarmónica (Felicity, 2013, p. 22.). La tabla de equivalencias se puede ver en la tabla 1 (Tucker, 2001).

Nombre tradicional	Nombre de clase
B# - C	0
C# - Db	1
D	2
D# - Eb	3
E - Fb	4
E# - F	5
F# - Gb	6
G	7
G# - Ab	8
A	9
A# - Bb	10
B - Cb	11

Tabla 1: Nombres de conjuntos de clases de alturas

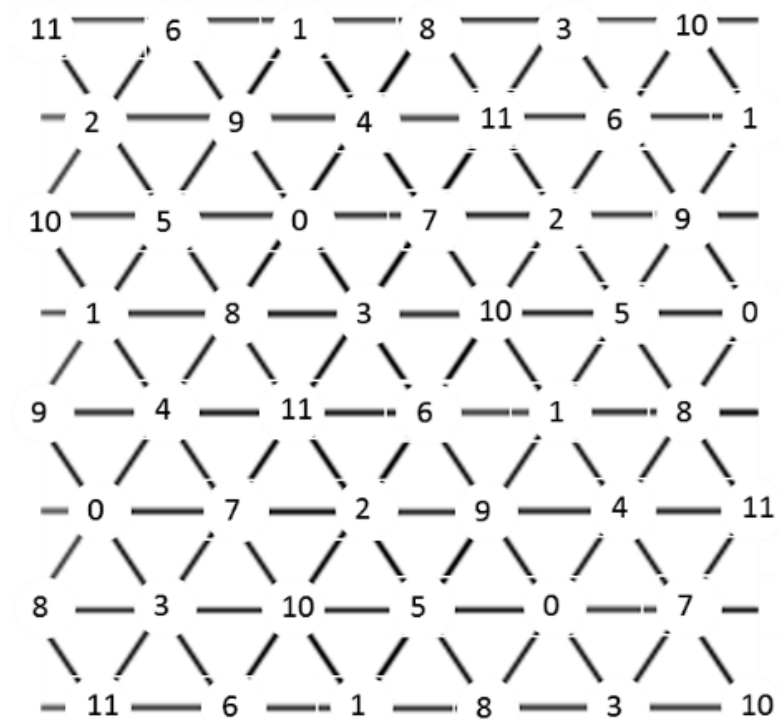


Figura 3: Tonnetz con nombres de clases de alturas

Teoría neo-Riemanniana en el pop y el rock

Así como la música rock es relativamente reciente, también lo son sus estudios analíticos y aún no hay consenso sobre cuál es la mejor aproximación a su análisis.

Capuzo (2004) se aproxima desde la TNR argumentando que las progresiones de acordes en el rock y pop “invitan a una aproximación transformacional” (pp., 177). Por otra parte argumenta que las consideraciones táctiles tienen un papel importante en el rock, pues mucha de la música se compone directamente sobre el diapasón lo que le da al compositor un repertorio limitado de acordes, este repertorio se caracteriza en general por mínimos movimientos de dedos y esto la hace susceptible de analizar con modelos transformacionales que pueden dar mejores resultados que otros tipos de análisis en música rock.

Felicity (2013) propone que la TNR puede ser útil al analizar géneros como el pop y el rock que no se ajustan a modelos de análisis más clásicos. Además comenta que las transformaciones compuestas, que son la combinación de dos o más transformaciones básicas, se pueden encontrar con frecuencia en el género rock. Por último Lilja (2004) indica que algunas progresiones en el metal se acercan a la operación neo-Riemanniana de cambio de tono líder “L”.

2.3 Composición musical algorítmica

Para Padilla la composición musical *algorítmica* es aquella composición que se basa en reglas formales (2012). Alpern propone que cuando en la anterior hay una mínima intervención humana podemos llamarle composición musical *automática* (1995). Podemos encontrar otros términos similares como música *generativa* y composición *asistida* por computador. El primero se refiere a la generación de música con reglas pero que no es repetible ni predecible, como una forma abierta de arte (Eno, 1996). El segundo se refiere a la participación del computador como herramienta de apoyo, pero el compositor mantiene activa participación en las decisiones del proceso.

En general la composición algorítmica es una técnica para “crear motivos, fragmentos o incluso obras completas musicales a partir de métodos formales” (Sirvent, 2014).

2.3.1 Breve recorrido histórico

En el siglo X encontramos los primeros intentos de composición algorítmica con Guido d' Arezzo, quien diseñó un sistema de generación automática de melodías a partir de textos, que consistía en asignar diferentes notas en la escala a las vocales en los textos (Padilla, 2012). Luego, en el siglo XIV, Philippe de Vitry (1291 – 1361) presentó el motete isorrítmico que por la relación formal entre talea (ritmo) y color (melodía) se puede considerar como un “ejemplo de composición algorítmica simple” (Padilla, 2012, p. 15).

Atanasius Kircher (1602 – 1680) describe en su trabajo *Musurgia Universalis* de 1650, una hipotética máquina musical: el *Arca Musarithmica*:

(...) una máquina capaz de inventar diferentes canciones según la ocasión (...) se programaba el aparato con diferentes tablas de madera según la carga emocional adecuada para el texto, el ritmo, sonidos, número de sílabas (...) (Padilla, 2012, p.16).

Los conceptos de aleatoriedad y juego en la composición musical aparecen en el siglo XVIII, listamos algunos ejemplos extraídos de Fuch (2016):

- Ever-Ready Minuet and Polonaise Composer, (*Der allezeit fertige Menuetten- und Polonaisencomponist*) por Johann Philipp Kirnbenger en 1757.
- A Method for Making Six Bars of Double Counterpoint at the Octave Without Knowing the Rules (*Einfall, einen doppelten Contrapunct in der Octave von sechs Tacten zu machen ohne die Regeln davon zu wissen*) por Carl Philipp Emanuel Bach en 1758.
- Table for composing minuets and trios to infinity, by playing with two dice (*Table pour Composer des minuets et des trios à l'infinie; avec deux dez à jouer*) por Maximilian Stadler en 1780.
- Game of Harmony, or an Easy Method for Composing an Infinite Number of Minuet-Trios, Without Any Knowledge of Counterpoint (*Gioco filarmonico, o sia maniera facile per comporre un infinito numero di minuetti e trio anche senza sapere il contrapunto : da eseguirsi per due violini e basso, o per due flauti e basso*) por Haydn en 1790.
- Musical dice game (*Musikalisches Würfelspiel*) atribuido a Mozart en 1792.

Ada Lovelace (1815 – 1852), considerada como la primera programadora informática de la historia, sugería lo siguiente en sus apuntes:

Suponiendo, por ejemplo, que las relaciones del sonido tonal en los símbolos de armonía y composición musical fueran susceptibles de tales expresiones y adaptaciones, la máquina podría componer piezas de música elaboradas y científicas de cualquier grado de complejidad o extensión (Lovelace (s.f.), citada en Alpern, 1995, p.1).

Llama la atención su gran visión pues en el siglo XIX aún no existían las computadoras y lo más cercano a ellas era la máquina calculadora de Charles Babbage.

No podemos dejar de mencionar a Iannis Xenakis (1922 – 2001) quien usó modelos matemáticos para sus composiciones en obras como *Metastasis* (1952), *Pithoprakta* (1955-56) y *Achorripsis* (1956-57) y fue pionero en el uso de la computadora para la composición musical (Padilla, 2012). Xenakis aplicó la teoría de probabilidad para componer música y él mismo acuñó el término de música estocástica (Sirvent, 2014).

En 1956 se presenta la *Illiad Suite*, que se acepta como “la primera composición asistida por computador para instrumentos tradicionales” (Di Nunzio, 2011). Esta fue compuesta por Lejaren Hiller y Leonard Isaacson usando un computador ILLIAC I.

2.3.2 Técnicas de composición algorítmica

Existen múltiples técnicas de composición algorítmica, unas inspiradas en modelos matemáticos y estadísticos, otras inspiradas en fenómenos biológicos o físicos, otras basadas en teorías de gramática y otras más en técnicas de inteligencia artificial. A continuación mencionaremos

brevemente algunas técnicas de composición para finalmente profundizar en los modelos de Markov que serán utilizados en este trabajo.

Combinatoria, aleatoriedad y estocástica

Brown (2005) llama combinatoria musical al proceso de reorganizar fragmentos musicales preexistentes de diferentes formas, este proceso puede ocurrir durante la composición, que denominaríamos composición probabilística o bien en el proceso de interpretación, que llamaríamos música móvil. La música que se apoya en procesos aleatorios la denomina música aleatoria y, cuando de la aleatoriedad simple se pasa a un modelo probabilístico, la denomina como música estocástica.

Gramáticas generativas

Este método basado en la teoría de gramática generativa de Noam Chomsky, se utiliza tanto en análisis como en composición. Está basado en un conjunto de reglas que es capaz de predecir combinaciones correctas dentro de un determinado lenguaje (Sirvent, 2014). Las gramáticas generativas pueden usarse en composición para crear reglas que permitan ciertas combinaciones musicales (Sirvent, 2014).

Teoría del caos y Teoría de fractales

La teoría del caos, mal entendido como desorden, estudia el comportamiento complejo de sistemas dinámicos no lineales, en los que pequeños cambios en las condiciones iniciales generan grandes cambios en el comportamiento futuro del sistema (Nierhaus, 2009). Este tipo de sistemas fue descrito inicialmente por Jules Henri Poincaré (1854 – 1912).

Por su parte la teoría de fractales estudia conjuntos que tienen un alto grado de similitud, es decir que su forma varía poco con la escala. Este tipo de conjuntos se pueden encontrar naturalmente en cristales de hielo, en algunos vegetales o en la forma de algunas costas (Nierhaus, 2009). El término fractal fue introducido por Benoit Mandelbrot cuyo conjunto está entre los más conocidos junto con el conjunto de Cantor y la curva de Koch (Nierhaus, 2009).

En composición algorítmica su uso varía según la relación que el compositor construya entre las estructuras musicales y los conceptos matemáticos.

Algoritmos genéticos

Los algoritmos genéticos son una clase algoritmos evolutivos inspirados en la teoría de evolución de Charles Darwin. El objetivo es encontrar soluciones a problemas a través de operaciones como selección, mutación y cruce. En composición algorítmica se pueden usar para generar material melódico a partir de un patrón inicial (Nierhaus, 2009) o bien el proceso compositivo puede

consistir en aplicar operaciones de mutación y cruce sin aplicar necesariamente una operación de selección (Brown, 2005).

Autómatas celulares

Los autómatas celulares se usan para modelar sistemas dinámicos discretos, estos se modelan en un espacio en donde el estado futuro de cada celda depende de su estado actual y el de sus celdas vecinas (Nierhaus, 2009). El ejemplo más conocido de autómatas celulares es el llamado Juego de la Vida planteado por John Horton Conway en 1970.

En composición algorítmica se pueden usar asignando tonos o ritmos a estados de las celdas o también asignando parámetros de síntesis de sonido a los estados de las celdas (Nierhaus, 2009).

Redes neuronales artificiales

Las redes neuronales artificiales están inspiradas en las redes neuronales naturales del cerebro humano. La solución a un problema se encuentra a partir del cambio en la estructura de componentes simples interconectados que actúan como neuronas artificiales (Nierhaus, 2009).

Cada neurona artificial tiene una interfaz por la cual recibe información del exterior o de otras neuronas, una función de propagación que combina todas las entradas de la neurona en una sola pieza de información, una función de activación que determina el estado de activación de la neurona y una función de salida que determina el valor a enviar a las neuronas conectadas (Nierhaus, 2009). Según como estén interconectadas las neuronas artificiales se habla de diferentes arquitecturas o topologías de la red neuronal.

Las redes neuronales tienen aplicaciones como reconocimiento de patrones, predicción, optimización y clasificación automática. En composición se hace necesario adaptar y escoger la arquitectura de red apropiada para representar propiedades estructurales de la música que pueden ser por ejemplo probabilidades o características musicales propiamente dichas (Nierhaus, 2009).

Inteligencia Artificial

La inteligencia artificial agrupa diferentes técnicas como búsquedas en espacios de estados, lógica difusa, razonamiento lógico, sistemas basados en reglas, aprendizaje de máquina, simulación basada en agentes y sistemas expertos entre otros (Nierhaus, 2009). En composición se pueden usar para imitación de estilo o para modelar procesos creativos musicales.

Cadenas de Markov

Las cadenas de Markov modelan procesos aleatorios dependientes del tiempo. Fueron desarrolladas por el matemático ruso Andrey Andreyevich Markov (1856-1922). En una cadena de Markov la probabilidad de un estado futuro de un sistema depende sólo de estados inmediatamente anteriores. Estas pueden ser representadas mediante un grafo o matriz de transición, en el grafo los nodos representan los posibles estados y las aristas las probabilidades de

transición, en la matriz las filas representan el estado actual y las columnas el estado siguiente. Las celdas contienen la probabilidad de esa transición. En la figura 4 se pueden observar una matriz y un grafo para la misma cadena (Sirvent, 2014. p. 24.)

Cuando el estado futuro depende solamente del estado actual se denominan cadenas de Markov de primer orden, cuando depende de los dos estados anteriores se denominan de segundo orden y así sucesivamente.

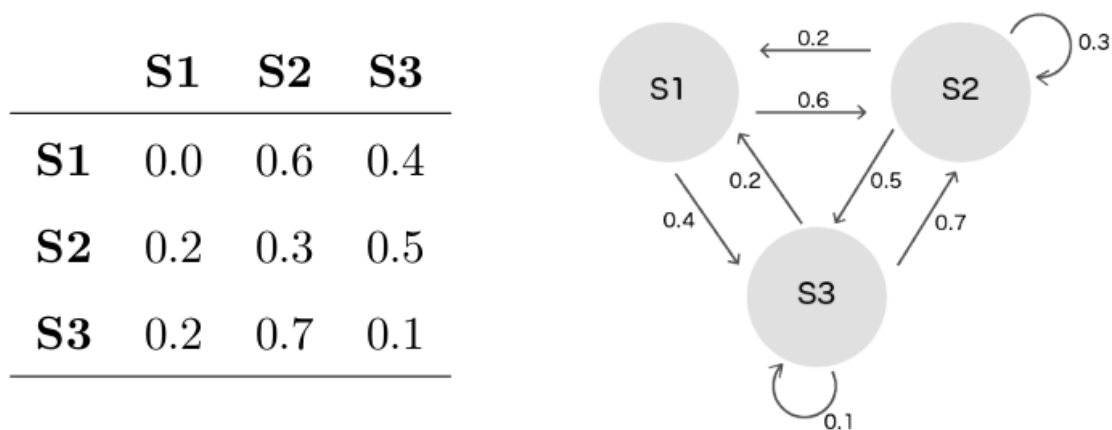


Figura 4: Matriz y grafo de transición para una cadena de Markov

Otro tipo de cadena de Markov son los modelos ocultos de Markov, en estos los estados y transiciones no son visibles directamente, sólo se pueden conocer ciertas salidas observables del sistema, que se denominan emisiones de probabilidad (Nierhaus, 2009).

Las cadenas de Markov se pueden usar en composición algorítmica para modelar las probabilidades de transición entre alturas, duraciones u otros estados que se puedan extraer de un análisis previo de un corpus musical (Nierhaus, 2009). Algunos ejemplos pioneros de aplicación de cadenas de Markov en composición algorítmica los encontramos en el *Electronic Music Synthesizer* de Harry Olson (1950), en la *ILLIAC Suite* de Hiller e Isaacson (1955) y en *Analogique A* de Iannis Xenakis (1958).

Un tipo especial de cadena de Markov es el paseo aleatorio en el que los estados están ordenados. De un estado sólo se puede pasar a los estados vecinos y un estado no puede repetirse de forma sucesiva; el paseo aleatorio puede usarse en composición con alturas musicales para generar melodías o decidir duraciones (Sirvent, 2014).

3 *Análisis*

Corpus de estudio

Sin duda una de las bandas más representativas del metal es Metallica, considerada la primera banda de *thrash metal* (Van Valkenburg, 2010). Metallica ha incluido varios temas instrumentales en su repertorio razón por la cual la hemos elegido como fuente para el análisis musical en este trabajo². Sus piezas instrumentales son:

- *Anesthesia - Pulling Teeth* (Kill'em All, 1983).
- *The Call Of Ktulu* (Ride The Lightning, 1984).
- *Orion* (Master of Puppets, 1986).
- *To Live Is To Die* (...And Justice for All, 1988).
- *Suicide & Redemption* (Death Magnetic, 2008).

De estas piezas hemos descartado la primera por ser una versión que hace un fuerte uso de bajo eléctrico con distorsión y la última por ser la más reciente y distanciada temporalmente de las demás.

Van Valkenburg (2010) destaca como características de las canciones de Metallica la variabilidad en el tempo, el uso de guitarras sin distorsión y guitarras acústicas en algunos pasajes y la complejidad de sus canciones que normalmente exceden el tiempo promedio de una canción metal, que es de unos 4 minutos. Sus piezas instrumentales son aún más extensas, el promedio de duración de las tres elegidas es de 9 minutos.

Abordaremos el análisis musical desde las tres perspectivas propuestas por Bent (1980) en nuestro marco teórico. Según la sustancia musical vamos a estudiar las piezas como una sucesión de estructuras, en este caso de acordes para la guitarra rítmica y de tonos para la guitarra solista. Según los métodos operativos vamos a utilizar procesos de reducción para simplificar las piezas y de segmentación para encontrar fragmentos de interés. También usaremos recuento estadístico para poder generar probabilidades. Según los medios de representación usaremos la representación de audio, partituras comentadas, listas, gráficos, descripciones verbales y código fuente.

Haremos el análisis informatizado a partir de las transcripciones en formato MIDI hechas por *fans* y encontradas en la red (www.midiworld.com) de las piezas elegidas. Somos conscientes de que pueden existir errores en las transcripciones, pero a falta de una representación simbólica original, consideramos que para nuestros propósitos estas fuentes son suficientemente válidas.

² Consideramos que el estudio de estas piezas protegidas por derechos de autor se enmarca dentro de la legislación norteamericana del *fair use* (www.law.cornell.edu/uscode/text/17/107) por tener un propósito de investigación sin ánimo de lucro.

También nos enmarcaremos dentro de lo propuesto por Moro (2016c) para el análisis musical informatizado en donde basados en un modelo teórico (la teoría neo-Riemanniana) procederemos a realizar un tratamiento estadístico - probabilístico.

Forma musical

Van Valkenburg (2010) establece que las piezas de Metallica suelen tener estructuras típicas de música popular como AABA, AABA', AAB' o AA'B'. Por supuesto ese análisis se ha hecho sobre piezas con lírica, en donde se puede identificar claramente el coro y las estrofas. Para nuestro caso hemos tenido en cuenta secciones más o menos diferenciadas por el papel de la guitarra solista y la repetición y variación de riffs en las secciones.

Presentamos a continuación el análisis de la forma musical sobre la representación de forma de onda en formato mono de las tres piezas elegidas³.

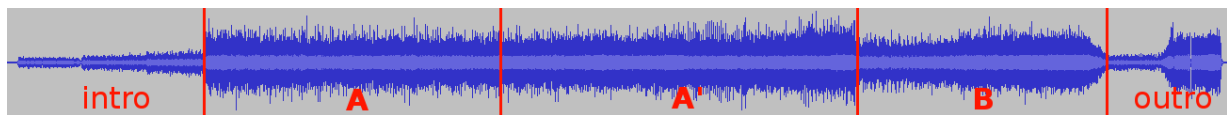


Figura 5: Forma musical. *The call of Ktulu*

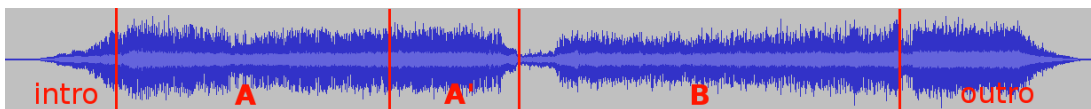


Figura 6: Forma musical. *Orion*

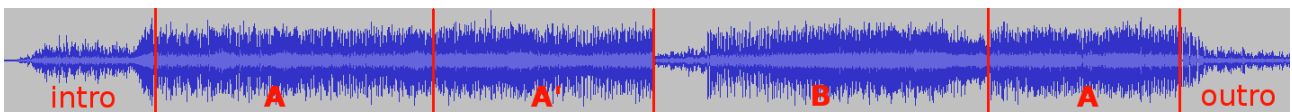


Figura 7: Forma musical. *To live is to die*

Análisis estadístico

En nuestro caso el análisis estadístico nos proporcionará guía para las decisiones de escalas y duraciones. Presentamos a continuación los histogramas de tonos y duraciones de cada pieza, incluyendo únicamente guitarras.

³ Para la generación de las formas de onda se ha usado el editor de audio open source Audacity. Más información en www.audacityteam.org

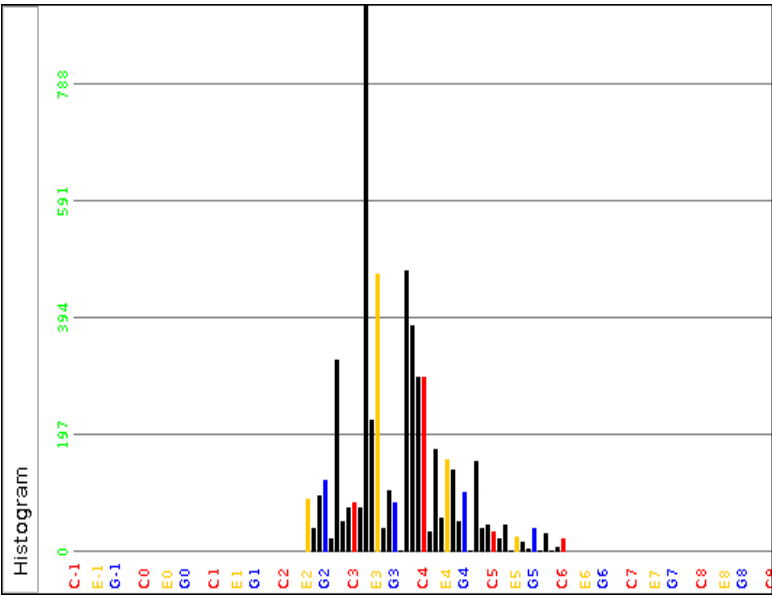


Figura 8: Distribución de tonos. *The call of Ktulu*

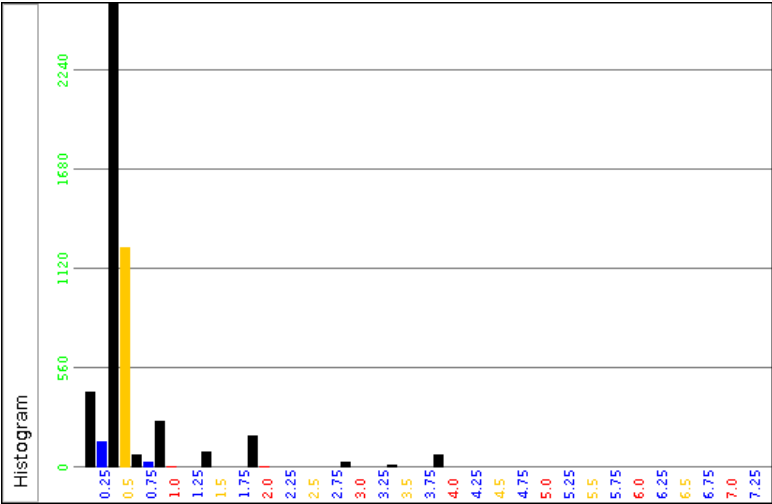


Figura 9: Distribución de duraciones. *The call of Ktulu*

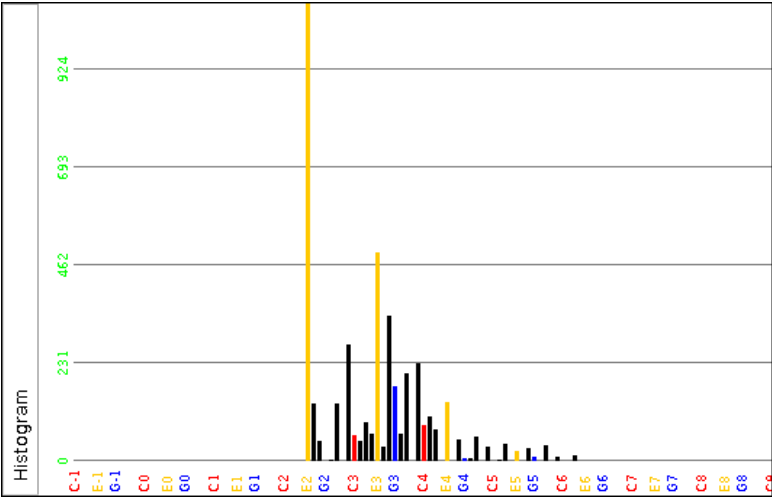


Figura 10: Distribución de tonos. Orion

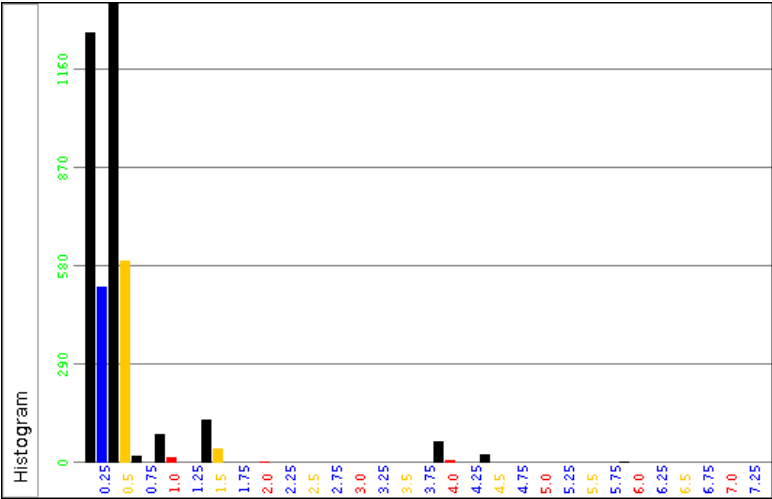


Figura 11: Distribución de duraciones. Orion

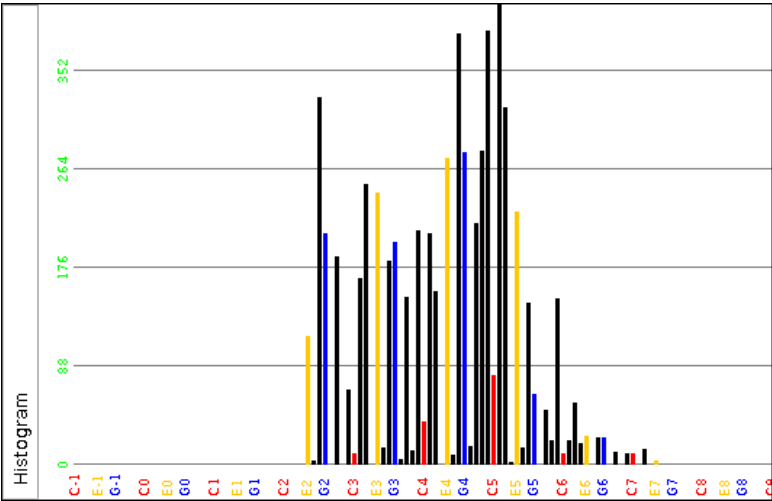


Figura 12: Distribución de tonos. To live is to die

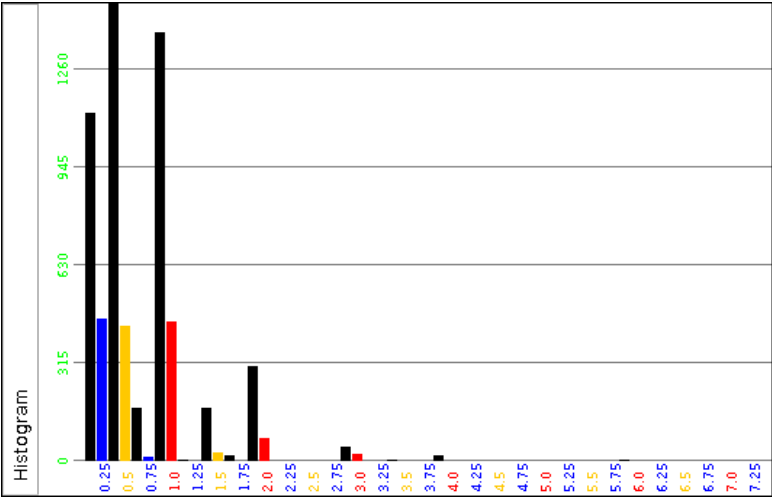


Figura 13: Distribución de duraciones. To live is to die

Análisis desde las transformaciones neo-Riemannianas

En términos generales nuestra aproximación al análisis consiste en las siguientes operaciones: el primer paso es reducir cada pista a sus acordes. El segundo paso es representar los acordes como conjuntos de clases de alturas. El tercer paso es ajustar y ubicar los conjuntos en el *tonnetz* de clases de alturas para hallar la transformación básica o compuesta que lleva de uno a otro. El cuarto paso es hacer un recuento sobre las transformaciones halladas para generar una cadena de Markov representada de forma matricial. A continuación haremos un ejercicio de este proceso a pequeña escala, eligiendo un riff típico de la canción *To Live is to Die* (Metallica, 1988). En la figura 14 presentamos la partitura comentada⁴.

♩ = 120

[1 6 10] [1 6 10] [1 2 6 10][1 2 6 10] [4 11] [1 6] [2 7 11] [2 7 11] [1 2 6 10][1 2 6 10] [2 7 11]

[1 6 10] [1 6 10] [1 6 10] [1 6 10] [4 7 11] [1 6 9] [2 7 11] [2 7 11] [1 6 10] [1 6 10] [2 7 11]

T T T LRLRP LRLR RLR T RLRP T LRPR

Figura 14: Análisis desde transformaciones neo-Riemannianas

La primera línea de la partitura presenta los conjuntos de clases de alturas originales de cada acorde. La segunda línea presenta los conjuntos ajustados a triadas del *tonnetz*. La última línea presenta las transformaciones compuestas necesarias para ir de un acorde a otro. Hemos introducido la operación “T” como una operación neutra para poder tener en cuenta las repeticiones inmediatas de acordes⁵. A partir de este análisis calcularemos las probabilidades de transición contando las transiciones de una transformación a otra en una matriz como se muestra en la tabla 2.

⁴ Para la edición de partituras en este proyecto se ha usado el editor open source MuseScore. Más información en musescore.org/es

⁵ La operación de transformación T es neutra en el sentido que no se realiza en realidad ninguna transformación. Así T aplicada sobre una triada nos conduce a ella misma, p.ej. [1 6 10] → T → [1 6 10]. El nombre T obedece a la primera letra de la palabra *this*, usada en programación orientada por objetos para referirse al objeto de ejecución actual.

	T	LRLRP	LRLR	RLR	RLRP	LRPR
T	2	1	0	0	1	1
LRLRP	0	0	1	0	0	0
LRLR	0	0	0	1	0	0
RLR	1	0	0	0	0	0
RLRP	1	0	0	0	0	0
LRPR	0	0	0	0	0	0

Tabla 2: Matriz de recuento de transiciones

Como el valor de probabilidad debe estar entre 0 y 1, normalizamos la matriz dividiendo cada celda sobre la suma de los valores de la fila. Adicionalmente hemos considerado que el riff se repite yendo del último al primer acorde, esto permite modelar la cadena correctamente pues todas las filas deben sumar 1. La matriz resultante se muestra en la tabla 3 y ya representa correctamente una cadena de Markov de primer orden.

	T	LRLRP	LRLR	RLR	RLRP	LRPR
T	0,4	0,2	0	0	0,2	0,2
LRLRP	0	0	1	0	0	0
LRLR	0	0	0	1	0	0
RLR	1	0	0	0	0	0
RLRP	1	0	0	0	0	0
LRPR	1	0	0	0	0	0

Tabla 3: Matriz de transición

Como el análisis informatizado en este trabajo tiene el objetivo de componer una pieza musical hemos generado otros dos tipos de cadena. Una consiste en generar a partir de las transformaciones compuestas, una cadena que solo contenga transformaciones básicas. En la tabla 4 presentamos un ejemplo de conversión de transformaciones compuestas en básicas. El otro tipo de cadena se genera contando las transiciones directas entre clases de alturas, o mejor entre conjuntos unitarios de clases de alturas, por ejemplo [5] → [11] → [3].

Transformaciones compuestas	Transformaciones básicas
$T \rightarrow LRLRP \rightarrow LRLR$	$T \rightarrow L \rightarrow R \rightarrow L \rightarrow R \rightarrow P \rightarrow L \rightarrow R \rightarrow L \rightarrow R$

Tabla 4: Conversión de transformaciones compuestas en básicas

El proceso anteriormente descrito presenta a pequeña escala los pasos a seguir para el análisis musical informatizado en el presente trabajo. El mismo proceso se ha aplicado a mayor escala sobre cada una de las pistas de cada archivo MIDI y para esto se ha hecho un desarrollo

informático en Java, descrito con detalle en el Anexo 4. A grandes rasgos el software extrae a partir de un archivo MIDI los acordes, los convierte en conjuntos de clases de alturas y luego busca las transformaciones en una representación informática del *tonnetz*. El conteo de transiciones se va acumulando en una matriz que luego se normaliza para representar una cadena de Markov de alguno de los tres tipos propuestos que son, resumiendo, matrices de transición directa entre clases de alturas, entre operaciones neo-Riemannianas básicas o compuestas y entre operaciones básicas únicamente. La totalidad de matrices generadas se puede consultar en el Anexo 1. A continuación presentamos como referencia una matriz de cada tipo.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]
[0]	0	0	0,15	0	0,15	0,11	0,07	0,1	0	0,11	0,25	0,06
[1]	0	0	1	0	0	0	0	0	0	0	0	0
[2]	0,05	0	0,05	0,02	0,04	0,11	0,02	0,09	0	0,53	0,05	0,05
[3]	0	0	0	0,79	0,03	0	0	0,03	0,04	0,11	0	0
[4]	0	0	0	0,11	0,23	0,08	0	0,06	0	0,13	0,23	0,17
[5]	0	0	0,34	0	0	0,34	0	0,04	0	0,21	0,06	0
[6]	0	0,14	0	0	0	0	0,57	0	0	0,14	0	0,14
[7]	0,11	0	0,17	0,03	0,08	0,03	0	0,39	0	0,12	0,06	0
[8]	0,33	0	0	0	0	0	0	0	0,67	0	0	0
[9]	0,1	0	0,49	0,01	0,09	0	0	0,04	0	0,18	0,05	0,05
[10]	0,29	0,06	0,17	0,02	0,25	0	0	0,02	0	0,1	0,1	0
[11]	0,56	0	0,22	0	0	0	0,11	0,11	0	0	0	0

Tabla 5: Matriz de transición entre clases de alturas

	LP	LPLP	LPLR	LPR	LPRP	LR	LRLR	PL	PLPR	PR	R	RL	RLR	RP	T
LP	0	0	0,03	0	0,4	0	0	0,05	0	0	0,07	0	0	0	0,45
LPLP	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
LPLR	0	0	0	0	0	0	0,4	0	0,6	0	0	0	0	0	0
LPR	0,6	0	0	0	0	0	0	0	0,2	0	0	0,2	0	0	0
LPRP	0	0,13	0	0	0,02	0,02	0	0	0,11	0	0,02	0	0	0	0,7
LR	0	0	0	0	0,93	0	0	0	0	0	0	0	0	0	0,07
LRLR	0,78	0	0	0	0,09	0	0	0	0	0	0	0	0	0,09	0,04
PL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
PLPR	0	0	0,06	0	0,12	0	0,02	0,29	0,1	0	0	0	0	0	0,41
PR	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
RL	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RLR	0	0	0	0,2	0	0	0	0	0	0	0	0	0,8	0	0
RP	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
T	0,07	0	0	0	0	0,09	0,09	0	0,16	0	0	0	0	0	0,58

Tabla 6: Matriz de transición entre operaciones compuestas

	L	P	R	T
L	0	0,62	0,32	0,06
P	0,39	0,03	0,39	0,19
R	0,47	0,38	0,03	0,11
T	0,25	0,16	0	0,58

Tabla 7: Matriz de transición entre operaciones básicas

A través del análisis informatizado se han generado 38 matrices. Se han descartado 17 de ellas por dos motivos. El primero es la presencia de ciclos infinitos, esto se da cuando en la diagonal de la matriz tenemos algún valor de probabilidad igual a uno. El segundo motivo es la presencia de estados finales, esto se da cuando la probabilidad acumulada en una fila es igual a cero.

Contamos entonces con 21 matrices válidas que serán usadas para la composición, estas se han grabado en archivos que serán leídos en la fase compositiva para acelerar el proceso, así el análisis que toma unos 20 segundos, se realiza una sola vez.

Sería posible realizar un análisis más profundo sobre las matrices generadas para encontrar, por ejemplo, las transiciones de mayor probabilidad y las de menor probabilidad, las matrices de mayor y menor dimensión o bien el grado de similitud o diferencia entre matrices. Como el propósito de este trabajo es compositivo, no entraremos en más detalles y dejaremos el análisis hasta este punto.

4 Composición

En la fase de análisis hemos construido varias cadenas de Markov, unas representan la probabilidad de transición entre acordes a través de operaciones neo-Riemannianas (básicas y compuestas) y otras representan la probabilidad de transición entre alturas directamente. Ahora, en la fase de composición estas serán usadas para generar secuencias de acordes para las partes de guitarra rítmica y secuencias de tonos para los puentes y solos de guitarra.

Para las cadenas de operaciones neo-Riemannianas el proceso de generación consiste en que a partir de un punto inicial aleatorio en la matriz, se genera una secuencia de transformaciones básicas o compuestas de una longitud especificada. Esta secuencia se aplica sobre la representación informática del *tonnetz* para generar la secuencia de acordes. Para las cadenas de transición directa entre clases de alturas se genera un punto inicial aleatorio y se genera una secuencia de clases. Las triadas y clases generadas se van agregando a la estructura de datos que representa una partitura, para finalmente ser escrita en un archivo midi junto con las pistas de bajo eléctrico y batería.

El algoritmo para calcular las secuencias en una matriz de transición es el siguiente: se genera un número aleatorio entre 0 y 1, luego se va calculando la probabilidad acumulada de la fila hasta que el número generado sea mayor o igual a esta. En ese momento el siguiente paso corresponderá al índice de la columna actual. Este método es usado por Brown (2009) y Sirvent (2014) de quien tomamos la figura 15 (p. 54) para una explicación visual.

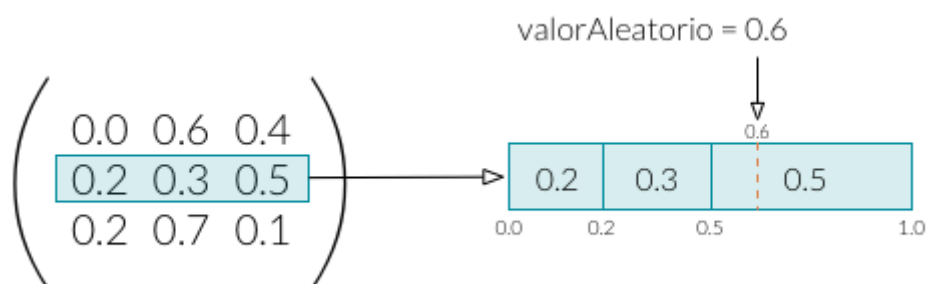


Figura 15: Cálculo para la transición en una matriz

Después de varios experimentos hemos llegado a la partitura del Anexo 3. Son muchas las posibilidades que se generan para una composición algorítmica, cada parámetro, cada valor, cada configuración del software influyen en la composición y conducen a diferentes resultados. Incluso la partitura mencionada es sólo una elegida entre las tantas posibles ejecuciones del software, pues por su construcción resulta en un algoritmo no determinista. Vamos a enumerar una serie de

experimentos que describen los retos y soluciones que surgieron durante el proceso compositivo y que llevaron de una u otra forma al resultado final.

Experimento 0

El primero de los experimentos fue generar una secuencia de acordes de igual duración para cada una de las matrices, esto con el objetivo de *escuchar* cada matriz y quizás identificar sus características para usar a la hora de componer. Sin embargo dada la cantidad de matrices (y por qué no, la poca experiencia musical de quien escribe) fue difícil reconocer y acaso memorizar sus características, así que se optó por una composición que incluyera de alguna forma todas las matrices.



Figura 16: Fragmento de Experimento 0

Experimento 1

En ese orden de ideas este experimento consistió en incluir una secuencia de acordes de cada matriz una tras otra, teniendo en cuenta las dos matrices de clases de alturas para construir puentes de guitarra. Para elegir las duraciones se construyó un generador de valores de una distribución de probabilidad normal con valores de media y desviación parametrizables. En este experimento ya se incluyó la pista de bajo, siguiendo la nota más baja de cada acorde de duración superior a una corchea; no se incluyó aún la pista de batería. También se incluyó programación para poder repetir y transponer secuencias de acordes generados.

El resultado de este experimento fue más o menos satisfactorio, la pieza carecía de una estructura clara acercándose un poco a algo de corte más *progresivo*, en el sentido de que la canción resulta más como un viaje sonoro. Como nuestra intención es producir una pieza de estructura más *reconocible* y no tan experimental, proseguimos al siguiente experimento.



Figura 17: Fragmento de Experimento 1

Experimento 2

A partir del anterior lo que se hace en este experimento es reducir la longitud de las secuencias generadas y aumentar las repeticiones y transposiciones. La pieza es un poco más estable pero aún no cuenta con una estructura clara. En este experimento se incluye manualmente una pista de batería con un patrón típico de rock a 4/4 que no combina muy bien y resalta la falta de estructura de la pieza. Se intenta también automatizar la pista de batería haciendo que el bombo siga al bajo. Se probó también con otra distribución de probabilidad para las duraciones, una distribución uniforme pero sin mejores resultados. Por estos resultados no del todo satisfactorios se continuó con el experimento 3.



Figura 18: Fragmento de Experimento 2

Experimento 3

Para el experimento 3 se partió de una revisión del análisis hecho para dar estructura a la pieza como primer paso. La estructura elegida fue una estructura de tipo AA'BA como la presentada en la figura 7. Esto nos permitió concentrarnos en cada parte de la composición por separado. Adicionalmente se descartó en este experimento la generación de duraciones con funciones de probabilidad y se optó por partir de un patrón rítmico de la batería como base para la composición. Inicialmente se esperaba generar la pista de batería manualmente, pero al resultar una composición no determinista no era viable hacer una pista para cada ejecución del software.

La base de la canción quedó entonces definida por los patrones de duración para el bombo, estos patrones se usan para la duración de los acordes generados y los acordes son la base para la pista del bajo. La ventaja de este método es que los patrones permiten generar compases completos de cuatro tiempos dando más orden a la pieza, recordando que nuestro objetivo no es ir hacia lo experimental sino a algo más *tradicional*, si cabe la palabra.

Al ser una composición no determinista hemos elegido una de tantas ejecuciones del software. En general no es tanta la variación entre ejecuciones, pues la estructura a gran escala permanece fija.

Figura 19: Fragmento de Experimento 3

A continuación en la tabla 8 presentamos los patrones de bombo usados, los valores numéricos son decimales de 0 a 4 donde 4 representa una redonda, 2 una blanca, 1 una negra, 0.5 una corchea y 0.25 una semicorchea. Los valores negativos representan silencios.

Patrón	Duraciones de bombo
Rock	1, -1, 1, -1
Doble	0.25, 0.25, -1.5, 0.25, 0.25, -1.5
Galope	0.5, 0.25, 0.25, -0.5, 0.25, 0.25, 0.5, 0.25, 0.25, -0.5, 0.25, 0.25
Dieciséis	0.25, 0.25, 0.25, 0.25, -0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, -0.25, 0.25, 0.25, 0.25
Solo	0.25, -0.25, -0.25, -0.25, -0.25, -0.25, -0.25, -0.25, 0.25, -0.25, -0.25, -0.25, -0.25, -0.25, -0.25, -0.25

Tabla 8: Patrones de duración para la composición

En la tabla 9 describimos cada una de las secciones de la pieza final, la forma de presentación es adaptada desde Van Valkenburg (2010, p.32) e incluye adicionalmente las matrices y el patrón de duración usado. La pieza está disponible en línea a través de soundcloud (soundcloud.com/codificador-nocturno/experimento-3)

Nombre	Compases	Duración	Matrices	Patrón	Comentarios
Apertura	1-2	2mm	19 (clases)	Rock	Sólo guitarra
Intro 1	3-6	4mm	6 (clases)	Rock	
Intro 2	7-10	4mm	19 (clases)	Rock	
Intro 3	11-12	2mm	19 (clases)	Rock	Sólo batería
A	13-52	40mm	1, 3, 5, 8, 10, 12, 14, 16, 18, 21 (básicas)	Galope	Un compás por matriz que suena 4 veces
Puente A1	53-54	2mm	19 (clases)	Galope	Sólo batería
Puente A2	55-58	4mm	6 (clases)	Doble	
Puente A3	59-60	2mm	19 (clases)	Doble	Sólo batería
A'	61-96	36mm	2, 4, 7, 9, 11, 13, 15, 17, 20 (compuestas)	Galope	Un compás por matriz que suena 4 veces: 2 veces sin transposición, 1 vez transpuesto y 1 vez sólo guitarra
Puente A'1	97-98	2mm	19 (clases)	Galope	Sólo batería
Solo 1	99-102	4mm	6 (clases)	Solo	
Solo 2	103-106	4mm	6 (clases)	Solo	
Puente A'2	107-108	2mm	19 (clases)	Doble	Sólo batería
Puente A'3	109-112	4mm	19 (clases)	Doble	Sólo bajo
B	113-160	48mm	Una matriz aleatoria (compuestas)	Dieciséis	48 compases de una misma matriz
Puente B1	161-162	2mm	19 (clases)	Dieciséis	Sólo batería
Puente B2	163-166	4mm	19 (clases)	Doble	
Puente B3	167-168	2mm	19 (clases)	Doble	Sólo batería
A	169-188	20mm	1, 3, 5, 8, 10, 12, 14, 16, 18, 21 (básicas)	Galope	Un compás por matriz que suena 2 veces
Outro 1	189-190	2mm	19 (clases)	Galope	Sólo batería
Outro 2	191-194	4mm	6 (clases)	Rock	
Outro 3	195-198	4mm	19 (clases)	Rock	
Cierre	199-200	2mm	19 (clases)	Rock	Sólo guitarra

Tabla 9: Estructura formal de la composición

5 *Discusión y Conclusiones*

Hemos presentado un género de música popular desde una perspectiva formal, el *metal*, que aunque enmarcado dentro del género rock se aleja un poco de este y posee su propia identidad. Surge como respuesta al rock más *hippie* con la bandera del amor y la paz, poniendo sobre la mesa otros temas más incómodos para la sociedad como la guerra, la violencia, la opresión y la corrupción política. También hemos hecho visible un subgénero poco conocido como es el *metal* instrumental.

Se ha mostrado que el análisis musical informatizado debe tener un fundamento teórico, pues de otra forma sólo tendríamos datos sin sentido. Se ha podido constatar también su utilidad al analizar grandes conjuntos de piezas, pues el análisis manual del corpus de este trabajo no hubiera sido viable. Para el análisis en este proyecto se eligió una representación simbólica y no en audio pues la consideramos más apropiada para procesar computacionalmente.

Varios autores de la bibliografía consultada recomiendan las aproximaciones transformacionales para el análisis de música rock. En este sentido, aunque la Teoría neo-Riemanniana no surge precisamente para el análisis del *metal*, sí ha probado ser útil y apropiada para el desarrollo de este trabajo, además la existencia de una representación geométrica como el *tonnetz* permite una comprensión del análisis musical desde lo visual. Debemos destacar que se introdujo la noción de una operación de transformación neutra para modelar la repetición inmediata de acordes, esto permitió modelar mejor el género metal que es mayormente rítmico como se puede ver con la alta probabilidad de la transformación neutra “T” en todas las matrices.

El término composición algorítmica se ha mostrado difícil de definir por los límites tan difusos con otros términos relacionados como música móvil, obra abierta, composición automática o música generativa. Los métodos de composición algorítmica son formales, en el sentido de que se pueden expresar matemáticamente y documentar, pero son totalmente subjetivos debido a que la relación entre lo formal y lo musical se establece arbitrariamente por el compositor. En nuestro caso elegimos una técnica viable computacionalmente en nuestro marco de tiempo como las cadenas de Markov, apenas mencionando otras posibilidades.

En otros casos el compositor experimentado busca nuevos caminos desde la composición algorítmica, pero en el nuestro la composición nunca estuvo orientada a algo experimental sino por el contrario buscando algo más verosímil o tradicional si cabe la palabra.

Proponemos que en nuestro caso tenemos una composición algorítmica asistida por computador pues aunque la parte de guitarra se ha automatizado desde el análisis informatizado, las demás decisiones como los algoritmos, las duraciones y la estructura general de la pieza han sido decisiones voluntarias del compositor.

Son varias las técnicas para la composición algorítmica, podríamos clasificarlas en dos grandes grupos para nuestros propósitos: el primero un grupo de técnicas que buscan generar música formal con resultados novedosos y el otro un grupo de técnicas que buscan modelar música

existente. Entre el primero encontramos por ejemplo, caos, fractales y autómatas celulares, en el segundo, redes neuronales, gramáticas generativas y cadenas de Markov. Hemos elegido esta última técnica por tres motivos, en primer lugar su facilidad de implementación computacional, en segundo lugar su clara representación en forma matricial y en tercer lugar su utilidad para modelar un proceso creativo existente como un proceso estocástico.

Gracias a la disponibilidad de librerías *open source* como JMusic y lenguajes de programación como Java hemos logrado implementar en corto tiempo el software necesario para el análisis y la composición. Aunque existen lenguajes orientados a audio o música específicamente, un lenguaje de propósito general como Java ha probado ser útil por la capacidad de abstracción que nos permite el modelo de programación orientada por objetos.

Se ha estructurado el software de la mejor manera posible para permitir experimentar con la composición. El análisis musical de las tres piezas y la generación de todas las matrices toma unos veinte segundos y la generación de la pieza toma menos de un par de segundos en un equipo de cómputo de escritorio, esto nos informa acerca de las posibilidades que tenemos con el software y hardware actual.

Hemos logrado el objetivo de componer una pieza de metal instrumental desde el análisis informatizado. Presentamos una de tantas ejecuciones del software, aunque en realidad la pieza no cambia mucho entre ejecuciones pues su estructura permanece fija. En los primeros intentos de composición ingenuamente pensamos que era suficiente modelar las progresiones de acordes, pero durante el proceso se mostró que los conocimientos teóricos musicales, la decisión de la estructura y la parte rítmica eran fundamentales para producir una pieza *creíble*. Con esto podemos ver que el papel del compositor, su experiencia, conocimientos o intenciones aparecen con notoriedad aún cuando la composición es algorítmica.

Finalmente, la motivación de este trabajo como todo trabajo de investigación, tiene algo de personal. El objetivo de pasar de ser un oyente pasivo a un creador, con apoyo en la tecnología se ha cumplido. La curiosidad por aplicar la informática lejos del lado empresarial y cerca del lado artístico se ha satisfecho.

5.1 Limitaciones

Es cierto que a falta de un modelo de análisis específico para el género hemos dejado de lado algunos detalles importantes como los *power chords* y hemos de alguna forma llevado la música hacia un modelo de análisis existente que encontramos apropiado.

Respecto al software, no se implementó una interfaz gráfica de usuario para mantener dentro del tiempo estipulado el desarrollo de este proyecto, sin embargo se echa de menos para permitir al usuario no programador experimentar con la composición.

Descartamos el realizar un análisis o evaluación sobre la pieza final, porque creemos que el proceso ha sido más interesante que el producto final. Aún muchos detalles se podrían agregar para dar

más carácter a la pieza y dejamos de lado muchas técnicas que corresponden más al lado de la posproducción musical y lo performativo que a la composición algorítmica que nos atañe.

Para hacer este proyecto factible no se ha analizado con la misma profundidad el papel del bajo y la batería ni la relación que surge cuando se ejecutan dos guitarras.

5.2 Prospectiva

Hemos mencionado superficialmente la historia y características del metal pues no es fácil encontrar literatura sobre el género, un trabajo futuro con un enfoque más histórico queda pendiente. Poco a poco el análisis de géneros de música popular se abre camino en la academia y esperamos que hayamos aportado nuestro grano de arena en este sentido.

Como dijimos en la sección de limitaciones, se ha usado un modelo de análisis existente, para el futuro se podría construir algún modelo de análisis para el *metal*, quizá teniendo muy en cuenta las características físicas de la guitarra, definiendo alguna geometría musical en relación al diapasón y en general teniendo en cuenta un punto de vista más del lado performativo.

El análisis musical informatizado puede resultar muy útil para el analista más experimentado, por esto se podría continuar con la implementación de una interfaz gráfica que permita al usuario no informático aprovechar el software que se ha desarrollado, bien para análisis propiamente dicho o para aplicaciones con propósitos pedagógicos.

Finalmente, no hemos usado las matrices para sacar conclusiones sobre el género o el corpus de estudio, pero esto sería totalmente factible para un trabajo futuro orientado hacia el análisis.

6 Referencias bibliográficas

- Alpern, A. (1995). *Techniques for Algorithmic Composition of Music*. Recuperado de <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.23.9364&rep=rep1&type=pdf>
- Brown, A. (2009). *Making Music with Java – An introduction to computer music, Java programming, and the jMusic library*. Lulu Press.
- Capuzzo, G. (2004). Neo-Riemannian Theory and the Analysis of Pop-Rock Music. *Music Theory Spectrum*. 26.2, pp. 177-179. Recuperado de https://libres.uncg.edu/ir/uncg/f/G_Capuzzo_Neo_2004.pdf
- Di Nunzio, A. (2011). *Illiac Suite*. Recuperado el 19 de octubre de 2016 de <http://www.musicainformatica.org/topics/illiac-suite.php>
- Eno, B. (1996). *Generative Music*. Recuperado el 19 de octubre de 2016 de <http://www.inmotionmagazine.com/en01.html>
- Felicity, L. (2013). *Essential Neo-Riemannian Theory for Today's Musician*. (Tesis de Maestría). Universidad de Tennessee, Knoxville. Recuperado de http://trace.tennessee.edu/cgi/viewcontent.cgi?article=2692&context=utk_gradthes
- Fuchs, M. (2016). Century of Play: 18th Century Precursors of Gamification. *Kinephanos, Special Edition Exploring the Frontiers of Digital Gaming: Traditional Games, Expressive Games, Pervasive Games*, 9-33. Recuperado de http://www.kinephanos.ca/Revue_files/2016_Fuchs.pdf
- Hascher, X. (2007). El neo-riemannismo americano. *Doce notas preliminares*, (19-20) 108-120. Recuperado de: http://www.docenotas.com/pdf/DOCENOTAS_Preliminares_19y20.pdf
- Igoa, E. (1999). Análisis estadístico, *Quodlibet*, 13, (pp. 71-79).
- Joel, R. (2014). *Some Really Great Thrash Metal Instrumental Songs*. Recuperado el 9 de noviembre de 2016 de <https://stayheavyalways.wordpress.com/2014/11/10/some-really-great-thrash-metal-instrumental-songs/>
- Lilja, E. (2004). *Characteristics of Heavy Metal Chord Structures*. (Tesis de Licenciatura). Universidad de Helsinki. Recuperado el 28 de octubre de 2016 de <https://helda.helsinki.fi/bitstream/handle/10138/19617/characte.pdf>

Moro, D. (2016a). *Tema 2: Teorías del Análisis Musical*. Material no publicado. Recuperado el 28 de octubre de 2016 de http://humanidadesonline.unir.net/cursos/lecciones/ARCHIVOS_COMUNES/versiones_para_imprimir/musicalo3/tema2.pdf

Moro, D. (2016b). *Tema 6: Análisis Musical Computarizado*. Material no publicado. Recuperado el 28 de octubre de 2016 de http://humanidadesonline.unir.net/cursos/lecciones/ARCHIVOS_COMUNES/versiones_para_imprimir/musicalo3/tema6.pdf

Moro, D. (2016c). *Tema 7: Herramientas informáticas para el análisis de la forma*. Material no publicado. Recuperado el 28 de octubre de 2016 de http://humanidadesonline.unir.net/cursos/lecciones/ARCHIVOS_COMUNES/versiones_para_imprimir/musicalo3/tema7.pdf

Nierhaus, G. (2009). *Algorithmic Composition - Paradigms of Automated Music generation*. New York: SpringerWien.

Padilla, V. (2012). *Probabilidad, Redes Neuronales e Inteligencia Artificial en Composición Musical. Desarrollo de los Sistemas MusicProb y MusicNeural*. (Tesis doctoral). Universidad Rey Juan Carlos, Madrid. Recuperado de <https://eciencia.urjc.es/handle/10115/11881>

Pearlin, J. (2014). *MIT Heavy Metal 101*. Recuperado el 17 de noviembre de 2016 de <http://metal.mit.edu/>

Sirvent, L. (2014). *Estudio e implementación de métodos de composición algorítmica con propósitos explorativos*. (Trabajo Fin de Grado). Universidad de Alicante, Alicante. Recuperado de <http://grfia.dlsi.ua.es/cm/pfc/lSirvent/memoria.pdf>

Solare, J. (2007). Analizando el análisis. *Doce notas preliminares*, (19-20) 36-47. Recuperado de: http://www.docenotas.com/pdf/DOCENOTAS_Preliminares_19y20.pdf

Starr, E. (2009). *The Everything Music Composition Book - A step-by-step guide to writing music*. Massachusetts: Adams Media.

Tucker, G. (2001). *A brief introduction to pitch class set analysis*. Recuperado el 9 de noviembre de http://www.mta.ca/pc-set/pc-set_new/pages/introduction/toc.html

Tucker, D. (2014). *Instru-Metal symphony: The 20 greatest heavy metal & hard rock instrumental songs*. Recuperado el 9 de noviembre de 2016 de <http://www.vh1.com/news/52872/20-greatest-heavy-metal-hard-rock-instrumentals/>

Van Valkenburg, A. (2010). *Musical Process and the Structuring of Riffs in Metallica*. (Tesis de Maestría). Universidad de Baylor. Recuperado el 28 de octubre de 2016 de https://baylor-ir.tdl.org/baylor-ir/bitstream/handle/2104/7950/aaron_vanvalkenburg_mm.pdf

Wagner, J. (2013). *10 Best Heavy Metal Instrumentals*. Recuperado el 9 de noviembre de 2016 de <http://noisecreep.com/best-metal-instrumentals/>

7 Anexos

7.1 Matrices de transición

A continuación se presentan las matrices de transición resultantes del análisis musical informatizado.

Orion_guitar_1_single				
	L	P	R	T
L	0,11	0,24	0,45	0,21
P	0,21	0,08	0,45	0,26
R	0,3	0,28	0,39	0,04
T	0,45	0,5	0,05	0

Orion_guitar_2_multiple																
	L	LP	LPLP	LPLR	LPR	LR	LRL	LRLR	PL	PLP	PLR	PR	R	RL	RP	T
L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
LP	0	0,07	0	0	0	0,5	0	0,21	0	0	0	0	0	0	0	0,21
LPLP	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
LPLR	0	0	0	0	0	0	0	0,17	0,11	0	0	0	0	0	0	0,72
LPR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
LR	0	0	0	0	0	0	0	0	0,17	0	0	0	0	0	0,52	0,31
LRL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
LRLR	0	0,12	0	0	0,12	0	0	0	0	0	0	0	0	0	0	0,75
PL	0	0,09	0	0,09	0	0	0	0	0	0	0	0	0	0,55	0	0,27
PLP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
PLR	0	0	0	0	0	0	0,5	0	0	0	0,25	0	0	0	0	0,25
PR	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
RL	0,04	0,21	0	0	0	0,08	0	0,08	0	0	0	0	0	0	0	0,58
RP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T	0	0,03	0	0,08	0,05	0,09	0	0	0,02	0,04	0,01	0,06	0	0,02	0	0,59

Orion_guitar_2_single				
	L	P	R	T
L	0,07	0,35	0,46	0,12
P	0,5	0	0,24	0,25
R	0,3	0,18	0,21	0,31
T	0,25	0,14	0,03	0,59

Orion_guitar_3_multiple																
	L	LP	LPLP	LPLR	LPR	LR	LRL	LRLR	PL	PLP	PLR	PR	R	RL	RP	T
L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
LP	0	0,07	0	0	0	0,5	0	0,21	0	0	0	0	0	0	0	0,21
LPLP	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
LPLR	0	0	0	0	0	0	0	0,17	0,11	0	0	0	0	0	0	0,72
LPR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
LR	0	0	0	0	0	0	0	0	0,17	0	0	0	0	0	0,52	0,31
LRL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
LRLR	0	0,12	0	0	0,12	0	0	0	0	0	0	0	0	0	0	0,75
PL	0	0,09	0	0,09	0	0	0	0	0	0	0	0	0	0,55	0	0,27
PLP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
PLR	0	0	0	0	0	0	0,5	0	0	0	0,25	0	0	0	0	0,25
PR	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
RL	0,04	0,21	0	0	0	0,08	0	0,08	0	0	0	0	0	0	0	0,58
RP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T	0	0,03	0	0,08	0,05	0,09	0	0	0,02	0,04	0,01	0,06	0	0,02	0	0,59

Orion_guitar_3_single				
	L	P	R	T
L	0,07	0,35	0,46	0,12
P	0,5	0	0,24	0,25
R	0,3	0,18	0,21	0,31
T	0,25	0,14	0,03	0,59

The_Call_Of_Ktulu_guitar_2_classes												
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]
[0]	0	0	0,15	0	0,15	0,11	0,07	0,1	0	0,11	0,25	0,06
[1]	0	0	1	0	0	0	0	0	0	0	0	0
[2]	0,05	0	0,05	0,02	0,04	0,11	0,02	0,09	0	0,53	0,05	0,05
[3]	0	0	0	0,79	0,03	0	0	0,03	0,04	0,11	0	0
[4]	0	0	0	0,11	0,23	0,08	0	0,06	0	0,13	0,23	0,17
[5]	0	0	0,34	0	0	0,34	0	0,04	0	0,21	0,06	0
[6]	0	0,14	0	0	0	0	0,57	0	0	0,14	0	0,14
[7]	0,11	0	0,17	0,03	0,08	0,03	0	0,39	0	0,12	0,06	0
[8]	0,33	0	0	0	0	0	0	0	0,67	0	0	0
[9]	0,1	0	0,49	0,01	0,09	0	0	0,04	0	0,18	0,05	0,05
[10]	0,29	0,06	0,17	0,02	0,25	0	0	0,02	0	0,1	0,1	0
[11]	0,56	0	0,22	0	0	0	0,11	0,11	0	0	0	0

The_Call_Of_Ktulu_guitar_2_multiple															
	LP	LPLP	LPLR	LPR	LPRP	LR	LRLR	PL	PLPR	PR	R	RL	RLR	RP	T
LP	0	0	0,03	0	0,4	0	0	0,05	0	0	0,07	0	0	0	0,45
LPLP	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
LPLR	0	0	0	0	0	0	0,4	0	0,6	0	0	0	0	0	0
LPR	0,6	0	0	0	0	0	0	0	0,2	0	0	0,2	0	0	0
LPRP	0	0,13	0	0	0,02	0,02	0	0	0,11	0	0,02	0	0	0	0,7
LR	0	0	0	0	0,93	0	0	0	0	0	0	0	0	0	0,07
LRLR	0,78	0	0	0	0,09	0	0	0	0	0	0	0	0	0,09	0,04
PL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
PLPR	0	0	0,06	0	0,12	0	0,02	0,29	0,1	0	0	0	0	0	0,41
PR	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
RL	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RLR	0	0	0	0,2	0	0	0	0	0	0	0	0	0,8	0	0
RP	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
T	0,07	0	0	0	0	0,09	0,09	0	0,16	0	0	0	0	0	0,58

The_Call_Of_Ktulu_guitar_2_single				
	L	P	R	T
L	0	0,62	0,32	0,06
P	0,39	0,03	0,39	0,19
R	0,47	0,38	0,03	0,11
T	0,25	0,16	0	0,58

The_Call_Of_Ktulu_guitar_4_multiple												
	LP	LPLR	LPR	LPRP	LR	LRLR	PLPR	R	RL	RLR	RP	T
LP	0	0	0	0	0	0	0	0	0	0	0	1
LPLR	0	0	0	0	0	0	0	0	0	0,5	0	0,5
LPR	0	0,03	0	0	0	0	0	0	0,03	0	0	0,93
LPRP	0	0	0	0	0	0	0	0	0	0	0	1
LR	0	0	0	1	0	0	0	0	0	0	0	0
LRLR	0,75	0	0	0	0	0	0	0	0	0	0,25	0
PLPR	0	0	0	0	0	1	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	1
RL	0	0	0	0	0	0	0	0	0	0	0	1
RLR	0	0	1	0	0	0	0	0	0	0	0	0
RP	0	0	0	0	0	0	0	0	0	0,33	0	0,67
T	0,07	0	0,08	0	0,02	0	0,02	0,08	0	0	0	0,71

The_Call_Of_Ktulu_guitar_4_single				
	L	P	R	T
L	0	0,73	0,26	0,01
P	0,1	0	0,48	0,42
R	0,33	0,1	0,04	0,53
T	0,18	0,02	0,09	0,71

The_Call_Of_Ktulu_guitar_5_multiple													
	LP	LPLR	LPR	LPRP	LR	LRLR	PLPR	R	RL	RLR	RP	T	
LP	0	0	0	0	0	0	0	0	0	0	0	0	1
LPLR	0	0	0	0	0	0	0	0	0	0	0,5	0	0,5
LPR	0	0,03	0	0	0	0	0	0	0	0,03	0	0	0,93
LPRP	0	0	0	0	0	0	0	0	0	0	0	0	1
LR	0	0	0	0	1	0	0	0	0	0	0	0	0
LRLR	0,75	0	0	0	0	0	0	0	0	0	0	0,25	0
PLPR	0	0	0	0	0	0	1	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	1
RL	0	0	0	0	0	0	0	0	0	0	0	0	1
RLR	0	0	1	0	0	0	0	0	0	0	0	0	0
RP	0	0	0	0	0	0	0	0	0	0	0,33	0	0,67
T	0,07	0	0,08	0	0,02	0	0,02	0,08	0	0	0	0	0,71

The_Call_Of_Ktulu_guitar_5_single				
	L	P	R	T
L	0	0,73	0,26	0,01
P	0,1	0	0,48	0,42
R	0,33	0,1	0,04	0,53
T	0,18	0,02	0,09	0,71

To_Live_Is_To_Die_guitar_1_multiple												
	L	LR	LRL	LRP	PLPR	PR	R	RL	RLR	RP	T	
L	0	0	0	0	0	0	0,71	0	0	0	0	0,29
LR	0	0	0	0	0	0	0	0	0,25	0,25	0	0,5
LRL	0	0	0	0	0	0	0	0	0	0	0	1
LRP	0,06	0	0,33	0,33	0	0	0	0	0	0	0	0,28
PLPR	0	0	0	0	0	1	0	0	0	0	0	0
PR	0,71	0,14	0	0	0	0	0	0	0	0	0	0,14
R	0	0	0	0	0	0,1	0	0	0,1	0	0	0,8
RL	0	0,17	0	0	0	0	0	0	0	0	0	0,83
RLR	0	0	0	0	0	0	0	0	0	0	0	1
RP	0	0	0	0,55	0	0,45	0	0	0	0	0	0
T	0,02	0,03	0	0,1	0,02	0	0,08	0,08	0,07	0,16	0	0,44

To_Live_Is_To_Die_guitar_1_single				
	L	P	R	T
L	0,02	0,04	0,7	0,24
P	0,53	0,12	0,23	0,12
R	0,31	0,43	0,04	0,22
T	0,15	0,02	0,39	0,44

To Live Is To Die guitar 2 multiple																						
	L	LP	LPLP	LPLPR	LPLR	LPR	LPRP	LR	LRL	LRLR	LRP	P	PL	PLP	PLPR	PLR	PR	PRP	R	RL	RLR	T
L	0,04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,04	0	0	0	0	0	0,93
LP	0	0	0	0	0	0	0,03	0	0,03	0	0	0,03	0,37	0,13	0	0	0	0	0	0	0	0,4
LPLP	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LPLPR	0	0	0	0	0	0	0	0	0	0,91	0	0	0	0	0	0	0	0,02	0	0	0	0,07
LPLR	0	0	0	0	0	0	0,06	0	0	0,12	0	0	0,47	0	0	0	0	0	0	0	0	0,35
LPR	0	0	0	0,25	0	0,5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,25
LPRP	0	0,15	0	0	0,07	0	0,07	0,05	0	0	0,02	0	0	0,02	0,44	0	0	0	0	0	0	0,18
LR	0	0	0	0	0	0	0,5	0	0	0	0	0,12	0	0	0	0	0	0	0	0	0	0,38
LRL	0	0	0	0	0,5	0	0	0	0,5	0	0	0	0	0	0	0	0	0	0	0	0	0
LRLR	0,31	0	0	0	0,01	0	0	0	0	0	0	0	0	0	0,16	0	0	0	0,01	0	0,39	0,1
LRP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
P	0,5	0	0	0	0	0	0,5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PL	0,03	0,06	0,12	0	0	0	0	0	0	0	0	0	0	0	0,25	0	0	0	0	0	0	0,53
PLP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
PLPR	0	0	0	0	0	0	0,1	0	0	0	0	0	0	0	0	0	0,05	0	0	0	0	0,84
PLR	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
PR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
PRP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R	0,5	0	0	0	0	0	0,5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RL	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RLR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T	0	0,05	0	0,1	0,02	0	0,07	0,01	0	0,06	0	0	0,03	0	0,08	0	0	0	0	0	0	0,57

To Live Is To Die guitar 2 single				
	L	P	R	T
L	0,02	0,54	0,36	0,08
P	0,45	0,09	0,4	0,06
R	0,44	0,2	0,07	0,29
T	0,32	0,11	0	0,57

Live Is To Die guitar 3 multiple			
	LPLR	P	T
LPLR	0	0	1
P	0	0	1
T	0	0,2	0,8

To Live Is To Die guitar 3 single				
	L	P	R	T
L	0	0,5	0,5	0
P	0,5	0	0	0,5
R	0	0	0	1
T	0	0,2	0	0,8

To_Live_Is_To_Die_guitar_4_classes												
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]
[0]	0,22	0,06	0,14	0	0,14	0	0	0,14	0	0,08	0	0,22
[1]	0,11	0,06	0,17	0	0,11	0	0,33	0	0	0	0	0,22
[2]	0,08	0,05	0,21	0,03	0,08	0,08	0,05	0,21	0	0	0	0,21
[3]	0	0	0	0,75	0,25	0	0	0	0	0	0	0
[4]	0,15	0,4	0,07	0,03	0,15	0,03	0,05	0,03	0	0,05	0	0,05
[5]	0	0	0,88	0	0,12	0	0	0	0	0	0	0
[6]	0	0,1	0,06	0,02	0,08	0,06	0,41	0,03	0	0,08	0	0,16
[7]	0,03	0	0,04	0	0,04	0	0,05	0,52	0,07	0,05	0	0,19
[8]	0	0	0	0	0	0	0,45	0,36	0	0,18	0	0
[9]	0,09	0	0	0	0,14	0	0	0,12	0,14	0,37	0	0,14
[10]	0,08	0	0	0	0	0	0	0	0	0,31	0,62	0
[11]	0,11	0,11	0,04	0,01	0,03	0	0,17	0,14	0	0,1	0,07	0,22

To_Live_Is_To_Die_guitar_6_multiple								
	L	LR	LRL	P	PLR	R	RLR	T
L	0	0	0	0	1	0	0	0
LR	0	0	0	0	1	0	0	0
LRL	1	0	0	0	0	0	0	0
P	0	0	0,2	0	0	0,5	0	0,3
PLR	0	0	0	1	0	0	0	0
R	0	0	0	0	0	0	1	0
RLR	0	0	0	0	1	0	0	0
T	0	1	0	0	0	0	0	0

To_Live_Is_To_Die_guitar_6_single				
	L	P	R	T
L	0,09	0,09	0,83	0
P	0,6	0	0,25	0,15
R	0,22	0,63	0,15	0
T	1	0	0	0

7.2 Discografía

Análisis musical

La siguiente discografía corresponde a las obras analizadas de la banda Metallica, está disponible en Spotify (play.spotify.com/user/cod_noc/playlist/7lo3NiQvgqo85fYzZriOXp).

Metallica. (1984). *The Call Of Ktulu*. En Ride The Lightning. [streaming]. Blackened Recordings.

Metallica. (1986) *Orion*. En Master Of Puppets. [streaming]. Blackened Recordings.

Metallica. (1988). *To Live Is To Die*. En ...And Justice For All. [streaming]. Blackened Recordings.

Piezas representativas

La siguiente discografía se ha condensado desde Tucker (2014), Joel (2014) y Wagner (2013), a partir de sus listados de los mejores temas instrumentales del metal. Esta lista se encuentra disponible en Spotify (play.spotify.com/user/cod_noc/playlist/oCSUzZ6Ff3yFtcXcYdaoSx).

Led Zeppelin. (1969). *Moby Dick*. En Led Zeppelin II. [streaming]. Swan Song – Atlantic Recording.

Black Sabbath. (1970). *Rat salad*. En Paranoid. [streaming]. Sanctuary Records – BMG.

Iron Maiden. (1981). *Genghis Khan*. En Killers. [streaming]. Iron Maiden Holdings – Parlophone Records.

Yngwie Malmsteen. (1985). *Black Star*. En Rising Force. [streaming]. Polydor.

Agent Steel. (1985). *The day at Guyana*. En Unstoppable Force. [streaming]. Columbia Records.

Metallica. (1986) *Orion*. En Master Of Puppets. [streaming]. Blackened Recordings.

Cacophony. (1987). *Speed Metal Symphony*. En Speed Metal Symphony. [streaming]. Shrapnel Records.

Dokken. (1987). *Mr. Scary*. En Back for The Attack. [streaming]. Elektra/Asylum Records – WEA International.

Death Angel. (1987). *The Ultra-Violence*. En *The Ultra-Violence*. [streaming]. Rykodiscs – Rhino Entertainment.

Megadeth. (1988). *Into the Lungs of Hell*. En *So Far, So Good ...So What!*. [streaming]. Capitol Records.

Testament. (1989). *Confusion Fusion*. En *Practice What You Preach*. [streaming]. Atlantic Recording – Anthem Entertainment – WEA International.

Anthrax. (1990). *Intro to Reality*. En *Persistence Of Time*. [streaming]. The Island Def Jam Music Group.

Dream Theater. (1994). *Erotomania*. En *Awake*. [streaming]. Atlantic Record – WEA.

In Flames. (1996). *The Jester's Dance*. En *The Jester Race*. [streaming]. Century Media Records.

Death. (1998). *Voice of the Soul*. En *The Sound of Perseverance*. [streaming]. Relapse Records.

Opeth. (1998). *Epilogue*. En *My Arms Your Hearse*. [streaming]. Candlelight Records.

Racer X. (1999). *Technical difficulties*. En *Technical difficulties*. [streaming]. Shrapnel Records.

Animals as Leaders. (2009). *Cafo*. En *Animals as Leaders*. [streaming]. Prosthetic Records.

7.3 Desarrollo informático

El desarrollo informático para este trabajo se ha realizado en el lenguaje de programación Java⁶ con apoyo de la librería jMusic⁷. A continuación describimos las clases más importantes que se han modelado.

ChordsAnalysis	Esta clase está encargada de leer los archivos MIDI. Usando la estructura de datos musicales de jMusic, hace un conteo de acordes de una partitura
HistogramAnalysis	Esta clase se usa para generar histogramas de tonos y duraciones
SetsAnalysis	Esta clase se usa para extraer conjuntos de clases de alturas de determinada longitud
Chord	Esta clase modela un acorde como una lista de notas
Matrix	Esta clase modela una matriz de transición
PCSNODE	Esta clase modela un nodo del <i>tonnetz</i> de clases de alturas, contiene una instancia de tipo PCSet y referencias a los nodos de las transformaciones L, R y P
Tonnetz	Esta clase es una representación informática del <i>tonnetz</i> de clases de alturas, se ha modelado como un grafo de referencias a objetos PCSNode
PCSet	Esta clase modela un conjunto de clases de alturas, contiene una lista de números enteros
Analysis	Esta clase contiene un método main, su función es realizar el proceso de análisis de los archivos midi
Composition	Esta clase contiene un método main, su función es realizar el proceso de composición a partir de las matrices generadas

El fragmento de código fuente Java con que se genera la pieza se presenta a continuación, el código fuente completo se encuentra disponible públicamente en github (github.com/codificador-nocturno/tfm).

⁶ Java es un lenguaje de programación orientado por objetos, marca registrada de Oracle Inc. Más información en java.com/es

⁷ JMusic es una librería open source para el desarrollo de software musical y sonoro en Java. Más información en sourceforge.net/projects/jmusic

/home/casa/William/Dropbox/MIM/C2/TFM/code/tfm/src/tfm/Composition.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package tfm;

import java.util.ArrayList;
import java.util.List;
import java.util.Random;
import jm.constants.Pitches;
import jm.constants.ProgramChanges;
import jm.music.data.Note;
import jm.music.data.Part;
import jm.music.data.Phrase;
import jm.music.data.Score;
import jm.music.tools.Mod;
import jm.util.Write;
import tfm.model.chords.Chord;
import tfm.model.markov.Matrix;
import tfm.model.nrt.PCSNode;
import tfm.model.pcst.PCSet;
import tfm.probability.NormalDistribution;

/**
 *
 * @author william
 */
public class Composition {

    public static void main(String[] args) throws Exception {
        Composition c = new Composition();
        c.create();
        c.write("experimento-3");
    }

    private List<Matrix> matrices;
    private Phrase guitarPhrase;
    private Phrase bassPhrase;
    private Score score;
    //drums
    private Part drumsPart;
    private Phrase hiHatPhrase;
    private Phrase snarePhrase;
    private Phrase kickPhrase;

    private int TEMPO = 120;
    private double[] gallopTimes = {0.5, 0.25, 0.25, -0.5, 0.25, 0.25, 0.5, 0.25, 0.25, -0.5, 0.25, 0.25};
    private double[] doubletimes = {0.25, 0.25, -1.5, 0.25, 0.25, -1.5};
    private double[] sixteenTimes = {0.25, 0.25, 0.25, 0.25, -0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, -0.25, 0.25, 0.25, 0.25};
    private double[] soloTimes = {0.25, -0.25, -0.25, -0.25, -0.25, -0.25, -0.25, -0.25, 0.25, -0.25, -0.25, -0.25, -0.25, -0.25, -0.25};
    private double[] rockTimes = {1, -1, 1, -1};

    private double[] durationArray;

    public Composition() throws Exception {
        matrices = new ArrayList<>();
        score = new Score();
        score.setDenominator(4);
        score.setNumerator(4);
        score.setTempo(TEMPO);

        Part guitarPart = new Part("Guitar");
        Part bassPart = new Part("Bass");

        guitarPhrase = new Phrase();
        bassPhrase = new Phrase();

        guitarPart.add(guitarPhrase);
        bassPart.add(bassPhrase);

        score.add(guitarPart);
        score.add(bassPart);

        guitarPart.setInstrument(ProgramChanges.DISTORTED_GUITAR);
        bassPart.setInstrument(ProgramChanges.ELECTRIC_BASS);

        drumsPart = new Part("Drums", 0, 9); // 9 = MIDI channel 10
        score.addPart(drumsPart);

        hiHatPhrase = new Phrase();
        snarePhrase = new Phrase();
        kickPhrase = new Phrase();

```

```

drumsPart.add(hiHatPhrase);
drumsPart.add(snarePhrase);
drumsPart.add(kickPhrase);

loadMatrices();
}

private void create() throws Exception {
    durationArray = rockTimes;
    //apertura
    add(generateChords(19, durationArray.length, 0), true, false, false, 2);
    //intro 1
    add(generateChords(6, durationArray.length, 0), 4);
    //intro 2
    add(generateChords(19, durationArray.length, 0), 4);
    //intro 3
    add(generateChords(19, durationArray.length, 0), false, false, true, 2);

    //A
    durationArray = gallopTimes;
    for (int i = 1; i < matrices.size(); i++) {
        if (matrices.get(i).getName().endsWith("_single")) {
            List<Chord> chords = generateChords(i, durationArray.length, -1);
            add(chords, 4);
        }
    }

    //Puente A1
    add(generateChords(19, durationArray.length, 0), false, false, true, 2);
    //Puente A2
    durationArray = doubleTimes;
    add(generateChords(6, durationArray.length, 0), 4);
    //Puente A3
    add(generateChords(19, durationArray.length, 0), false, false, true, 2);

    //A'
    durationArray = gallopTimes;
    for (int i = 1; i < matrices.size(); i++) {
        if (matrices.get(i).getName().endsWith("_multiple")) {
            List<Chord> chords = generateChords(i, durationArray.length, -1);
            add(chords, 2);
            List<Chord> copy = Refs.chords.duplicate(chords);
            Refs.chords.transpose(copy, +1);
            add(copy, 1);
            add(chords, true, false, false, 1);
        }
    }

    //Puente A'1
    add(generateChords(19, durationArray.length, 0), false, false, true, 2);
    durationArray = soloTimes;
    //Solo 1
    add(generateChords(6, durationArray.length, +1), 4);
    //Solo 2
    add(generateChords(6, durationArray.length, +1), 4);
    durationArray = doubleTimes;
    //Puente A'2
    add(generateChords(19, durationArray.length, 0), false, false, true, 2);
    //Puente A'3
    add(generateChords(19, durationArray.length, 0), false, true, false, 4);

    //B
    durationArray = sixteenTimes;
    int[] multiple = {2, 4, 7, 9, 11, 13, 15, 17, 20};
    int bMatrix = new Random().nextInt(multiple.length);
    for (int i = 0; i < 48; i++) {
        add(generateChords(multiple[bMatrix], durationArray.length, -1), 1);
    }

    //Puente B1
    add(generateChords(19, durationArray.length, 0), false, false, true, 2);
    //Puente B2
    durationArray = doubleTimes;
    add(generateChords(19, durationArray.length, 0), 4);
    //Puente B3
    add(generateChords(19, durationArray.length, 0), false, false, true, 2);

    //A
    durationArray = gallopTimes;
    for (int i = 1; i < matrices.size(); i++) {
        if (matrices.get(i).getName().endsWith("_single")) {
            List<Chord> chords = generateChords(i, durationArray.length, -1);
            add(chords, 2);
        }
    }
}

```

```

//Outro 1
add(generateChords(19, durationArray.length, 0), false, false, true, 2);
durationArray = rockTimes;
//Outro 2
add(generateChords(6, durationArray.length, 0), 4);
//Outro 3
add(generateChords(19, durationArray.length, 0), 4);
//Cierre
add(generateChords(19, durationArray.length, 0), true, false, false, 2);
}

private void loadMatrices() throws Exception {
    String[] names = {
        "Orion_guitar_1_single",
        "Orion_guitar_2_multiple",
        "Orion_guitar_2_single",
        "Orion_guitar_3_multiple",
        "Orion_guitar_3_single",
        "The_Call_Of_Ktulu_guitar_2_classes",
        "The_Call_Of_Ktulu_guitar_2_multiple",
        "The_Call_Of_Ktulu_guitar_2_single",
        "The_Call_Of_Ktulu_guitar_4_multiple",
        "The_Call_Of_Ktulu_guitar_4_single",
        "The_Call_Of_Ktulu_guitar_5_multiple",
        "The_Call_Of_Ktulu_guitar_5_single",
        "To_Live_Is_To_Die_guitar_1_multiple",
        "To_Live_Is_To_Die_guitar_1_single",
        "To_Live_Is_To_Die_guitar_2_multiple",
        "To_Live_Is_To_Die_guitar_2_single",
        "To_Live_Is_To_Die_guitar_3_multiple",
        "To_Live_Is_To_Die_guitar_3_single",
        "To_Live_Is_To_Die_guitar_4_classes",
        "To_Live_Is_To_Die_guitar_6_multiple",
        "To_Live_Is_To_Die_guitar_6_single"
    };

    matrices.add(null); //indices from 1 :P

    for (String n : names) {
        //read serialized matrix
        matrices.add(Refs.matrices.readFromDisk(n));
    }
}

private List<Chord> generateChords(int matrixNumber, int length, double duration, int transposition) {
    List<Chord> chords = gen(matrixNumber, length, transposition);
    Refs.chords.applyDuration(chords, duration);
    return chords;
}

private List<Chord> generateChords(int matrixNumber, int length, double mean, double deviation, int transposition) {
    List<Chord> chords = gen(matrixNumber, length, transposition);
    Refs.chords.applyDuration(chords, new NormalDistribution(mean, deviation));
    return chords;
}

private List<Chord> generateChords(int matrixNumber, int length, int transposition) {
    List<Chord> chords = gen(matrixNumber, length, transposition);
    Refs.chords.applyDuration(chords, durationArray);
    return chords;
}

private List<Chord> gen(int matrixNumber, int length, int transposition) {
    Matrix m = matrices.get(matrixNumber);
    List<Chord> chords;

    if (m.getName().contains("_classes")) {
        //single pitch class
        //generate sets
        List<PCSet> sets = m.generateSets(length);
        chords = Refs.chords.convertSetsToChords(sets);
    } else {
        //operations
        List<String> operations = m.generateSteps(length - 1);
        //from ops createSong tonnetz node list
        List<PCSNODE> nodes = Refs.tonnetz.generateNodes(operations);
        //from nodes createSong list of chords
        chords = Refs.chords.convertNodesToChords(nodes);
    }

    Refs.chords.transpose(chords, transposition);

    return chords;
}

```

```

private List<Chord> generateBass(List<Chord> chords) {
    List<Chord> lowChords = new ArrayList<>();

    //copy lowest note
    for (Chord c : chords) {
        Note lower = c.getNote(0);
        Note bass = new Note(lower.getNote());
        Chord cn = new Chord();

        //if (c.getDuration() <= 0.25) {
        Mod.transpose(bass, (-2 * 12));
        // } else {
        //     bass.setPitch(Pitches.REST);
        // }

        cn.setDuration(c.getDuration());
        cn.add(bass);

        lowChords.add(cn);
    }

    return lowChords;
}

private void generateDrums() {
    generateHat();
    generateKick();
    generateSnare();
}

private void generateHat() {
    // make hats
    for (int i = 0; i < 4; i++) {
        Note note = new Note(42, 1);
        hiHatPhrase.addNote(note);
    }
}

private void generateKick() {
    // make bass drum
    for (int j = 0; j < durationArray.length; j++) {
        if (durationArray[j] < 0.0) {
            kickPhrase.addNote(new Note(Pitches.REST, Math.abs(durationArray[j])));
        } else {
            kickPhrase.addNote(new Note(36, durationArray[j]));
        }
    }
}

private void generateSnare() {
    Note rest = new Note(Pitches.REST, 1);
    snarePhrase.addNote(rest);
    Note note = new Note(38, 1);
    snarePhrase.addNote(note);
    snarePhrase.addNote(rest);
    snarePhrase.addNote(note);
}

private void add(List<Chord> chords, int times) {
    for (int i = 0; i < times; i++) {
        //chords
        guitarPhrase.addNoteList(Refs.chords.convertChordsToPhrase(chords).getNoteList(), true);
        //bass
        bassPhrase.addNoteList(Refs.chords.convertChordsToPhrase(generateBass(chords)).getNoteList(), true);
        //drums
        generateDrums();
    }
}

private void write(String name) {
    Write.midi(score, name + ".mid");
}

private void add(List<Chord> chords, boolean guitar, boolean bass, boolean drums, int times) {
    for (int i = 0; i < times; i++) {
        //chords
        if (guitar && !bass && !drums) {
            guitarPhrase.addNoteList(Refs.chords.convertChordsToPhrase(chords).getNoteList(), true);
            bassPhrase.addNote(new Note(Pitches.REST, 4.0));
            hiHatPhrase.addNote(new Note(Pitches.REST, 4.0));
            snarePhrase.addNote(new Note(Pitches.REST, 4.0));
            kickPhrase.addNote(new Note(Pitches.REST, 4.0));
        }

        if (bass && !guitar && !drums) {

```

```
guitarPhrase.addNote(new Note(Pitches.REST, 4.0));
bassPhrase.addNoteList(Refs.chords.convertChordsToPhrase(generateBass(chords)).getNoteList(), true);
hiHatPhrase.addNote(new Note(Pitches.REST, 4.0));
snarePhrase.addNote(new Note(Pitches.REST, 4.0));
kickPhrase.addNote(new Note(Pitches.REST, 4.0));
}

if (drums && !guitar && !bass) {
    guitarPhrase.addNote(new Note(Pitches.REST, 4.0));
    bassPhrase.addNote(new Note(Pitches.REST, 4.0));
    generateDrums();
}
}
}
```


7.4 Partitura

A continuación se presenta la partitura de la pieza compuesta, se ha generado automáticamente desde el archivo midi producido por el software con el editor de partituras MuseScore.

Experimento 3

Composición algorítmica de metal instrumental

William Vargas

$\text{♩} = 120$

Guitarra eléctrica

Bajo eléctrico

Set de percusión

8

6

Guit. El.

Baj. El.

Perc.

8

12

Guit. El.

Baj. El.

Perc.

8

15

Guit. El.

Baj. El.

Perc.

8

17

Guit. El.

Baj. El.

Perc.

19

Guit. El.

Baj. El.

Perc.

21

Guit. El.

Baj. El.

Perc.

23

Guit. El.

Baj. El.

Perc.

25

Guit. El.

Baj. El.

Perc.

27

Guit. El.

Baj. El.

Perc.

29

Guit. El.

Baj. El.

Perc.

31

Guit. El.

Baj. El.

Perc.

33

Guit. El.

Baj. El.

Perc.

35

Guit. El.

Baj. El.

Perc.

37

Guit. El.

Baj. El.

Perc.

39

Guit. El.

Baj. El.

Perc.

41

Guit. El.

Baj. El.

Perc.

43

Guit. El.

Baj. El.

Perc.

45

Guit. El.

Baj. El.

Perc.

47

Guit. El.

Baj. El.

Perc.

49

Guit. El.

Baj. El.

Perc.

8

51

Guit. El.

Baj. El.

Perc.

8

53

Guit. El.

Baj. El.

Perc.

8

56

Guit. El.

Baj. El.

Perc.

8

60

Guit. El.

Baj. El.

Perc.

62

Guit. El.

Baj. El.

Perc.

64

Guit. El.

Baj. El.

Perc.

66

Guit. El.

Baj. El.

Perc.

68

Guit. El.

Baj. El.

Perc.

8

70

Guit. El.

Baj. El.

Perc.

8

71

Guit. El.

Baj. El.

Perc.

8

72

Guit. El.

Baj. El.

Perc.

8

74

Guit. El.

Baj. El.

Perc.

76

Guit. El.

Baj. El.

Perc.

79

Guit. El.

Baj. El.

Perc.

81

Guit. El.

Baj. El.

Perc.

83

Guit. El.

Baj. El.

Perc.

83

84

85

Guit. El.

Baj. El.

Perc.

85

86

87

Guit. El.

Baj. El.

Perc.

87

88

88

Guit. El.

Baj. El.

Perc.

88

89

90

Guit. El.

Baj. El.

Perc.

92

Guit. El.

Baj. El.

Perc.

94

Guit. El.

Baj. El.

Perc.

96

Guit. El.

Baj. El.

Perc.

98

Guit. El.

Baj. El.

Perc.

Measure 98: The guitar and bass parts have a melodic line starting on a whole note, followed by eighth notes. The percussion part has a steady eighth-note pattern with accents.

Measure 99: The guitar and bass parts continue the melodic line. The percussion part has a steady eighth-note pattern with accents.

100

Guit. El.

Baj. El.

Perc.

Measure 100: The guitar and bass parts continue the melodic line. The percussion part has a steady eighth-note pattern with accents.

Measure 101: The guitar and bass parts continue the melodic line. The percussion part has a steady eighth-note pattern with accents.

101

Guit. El.

Baj. El.

Perc.

Measure 102: The guitar and bass parts continue the melodic line. The percussion part has a steady eighth-note pattern with accents.

Measure 103: The guitar and bass parts continue the melodic line. The percussion part has a steady eighth-note pattern with accents.

102

Guit. El.

Baj. El.

Perc.

Measure 104: The guitar and bass parts continue the melodic line. The percussion part has a steady eighth-note pattern with accents.

Measure 105: The guitar and bass parts continue the melodic line. The percussion part has a steady eighth-note pattern with accents.

103

Guit. El.

Baj. El.

Perc.

104

Guit. El.

Baj. El.

Perc.

105

Guit. El.

Baj. El.

Perc.

106

Guit. El.

Baj. El.

Perc.

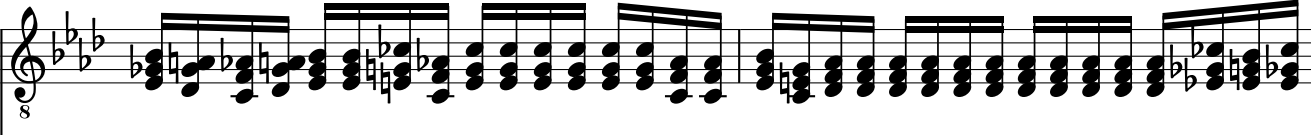
109


Guit. El. 


Baj. El. 

Perc. 

113

Guit. El. 

Baj. El. 

Perc. 

115

Guit. El. 

Baj. El. 

Perc. 

116

Guit. El. 

Baj. El. 

Perc. 

117

Guit. El.

Baj. El.

Perc.

8

118

Guit. El.

Baj. El.

Perc.

8

119

Guit. El.

Baj. El.

Perc.

8

120

Guit. El.

Baj. El.

Perc.

8

121

Guit. El.

Baj. El.

Perc.

8

122

Guit. El.

Baj. El.

Perc.

8

123

Guit. El.

Baj. El.

Perc.

8

124

Guit. El.

Baj. El.

Perc.

8

125

Guit. El.

Baj. El.

Perc.

126

Guit. El.

Baj. El.

Perc.

127

Guit. El.

Baj. El.

Perc.

128

Guit. El.

Baj. El.

Perc.

129

Guit. El.

Baj. El.

Perc.

130

Guit. El.

Baj. El.

Perc.

131

Guit. El.

Baj. El.

Perc.

132

Guit. El.

Baj. El.

Perc.

133

Guit. El.

Baj. El.

Perc.

134

Guit. El.

Baj. El.

Perc.

135

Guit. El.

Baj. El.

Perc.

136

Guit. El.

Baj. El.

Perc.

137

Guit. El.

Baj. El.

Perc.

138

Guit. El.

Baj. El.

Perc.

139

Guit. El.

Baj. El.

Perc.

140

Guit. El.

Baj. El.

Perc.

141

Guit. El.

Baj. El.

Perc.

142

Guit. El.

Baj. El.

Perc.

143

Guit. El.

Baj. El.

Perc.

144

Guit. El.

Baj. El.

Perc.

145

Guit. El.

Baj. El.

Perc.

146

Guit. El.

Baj. El.

Perc.

147

Guit. El.

Baj. El.

Perc.

148

Guit. El.

Baj. El.

Perc.

149

Guit. El.

Baj. El.

Perc.

8

150

Guit. El.

Baj. El.

Perc.

8

151

Guit. El.

Baj. El.

Perc.

8

152

Guit. El.

Baj. El.

Perc.

8

153

Guit. El.

Baj. El.

Perc.

154

Guit. El.

Baj. El.

Perc.

155

Guit. El.

Baj. El.

Perc.

156

Guit. El.

Baj. El.

Perc.

157

Guit. El.

Baj. El.

Perc.

158

Guit. El.

Baj. El.

Perc.

159

Guit. El.

Baj. El.

Perc.

160

Guit. El.

Baj. El.

Perc.

161

Guit. El.

Baj. El.

Perc.

8

164

Guit. El.

Baj. El.

Perc.

8

169

Guit. El.

Baj. El.

Perc.

8

170

Guit. El.

Baj. El.

Perc.

8

171

Guit. El.

Baj. El.

Perc.

173

Guit. El.

Baj. El.

Perc.

175

Guit. El.

Baj. El.

Perc.

177

Guit. El.

Baj. El.

Perc.

179

Guit. El.

Baj. El.

Perc.

181

Guit. El.

Baj. El.

Perc.

183

Guit. El.

Baj. El.

Perc.

185

Guit. El.

Baj. El.

Perc.

187

Guit. El.

Baj. El.

Perc.

189

Guit. El.

Baj. El.

Perc.

194

Guit. El.

Baj. El.

Perc.

200

Guit. El.

Baj. El.

Perc.