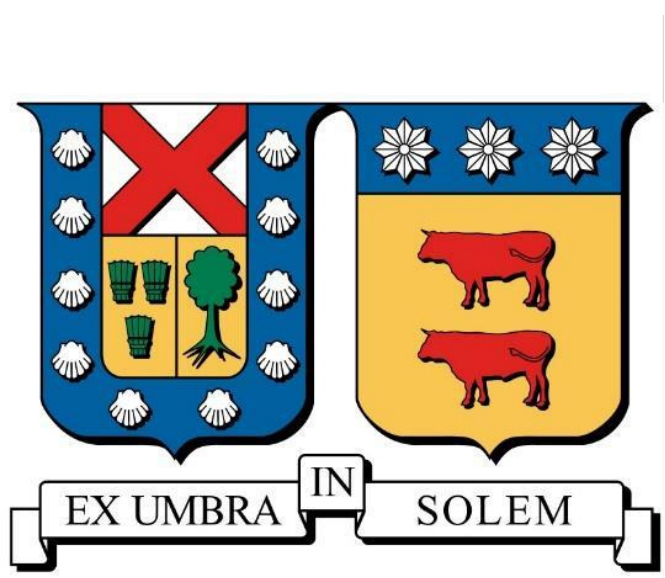
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO - CHILE



SISTEMA DE GENERACIÓN Y EVALUACIÓN DE ARCHIVOS MIDI PARA MÚSICA “HOUSE” MEDIANTE APRENDIZAJE PROFUNDO

JORGE IGNACIO LEFENDA MONTES

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELECTRÓNICO

PROFESOR GUÍA: MATÍAS ZAÑARTU, Ph.D

Valparaíso, mayo de 2023

Agradecimientos

A mis Padres, que durante toda mi vida han sido las personas que me han brindado hasta lo imposible y me han dado todo lo necesario para desarrollar mi carrera universitaria. Muchas gracias por todo lo que han hecho por mí.

A mi hermana que siempre me ha brindado su apoyo y ha ayudado en todo lo posible.

A mis compañeros y amigos, que siempre me han ayudado cuando lo necesitaba, sin esperar nada a cambio, convirtiendo a la Universidad en mi segunda casa.

A mis colegas músicos, que accedieron a colaborar conmigo en este trabajo sin esperar nada a cambio.

A mis Profesores, por brindar sus conocimientos, principio, valores y formarme como un Ingeniero, especialmente a mis Profesores correferente y guía, María José Escobar y Matías Zañartu respectivamente, que, a pesar de las adversidades, siempre me motivaron para llevar adelante este proyecto.

Resumen

La inteligencia artificial y la generación de contenido artístico mediante las técnicas de Deep Learning, son campos de estudio que al año 2023 están en pleno auge. Existen algoritmos y frameworks capaces de generar distintos estilos de música entrenados con distintos géneros y estilos, cada uno con sus particularidades rítmicas. Para poder crear Ritmos House, se ha incluido al dataset elementos de baterías que cumplen con el patrón 4/4 y ritmos percusivos. También se implementa un algoritmo que permite corregir los archivos generados a distinta temperatura (entiéndase temperatura como parámetro de proporcionalidad de probabilidades, explicado más adelante) para que cumplan con el patrón y evaluar el mejor valor de ellos para que las pistas cumplan con el criterio "Four on the Floor". Se establecen métodos de evaluación cuantitativos para la música, utilizando métricas como la Similitud del Coseno, la Distancia de Edición o de Levenshtein y una métrica específica en base al algoritmo utilizado nombrada Coeficiente de Similitud.

Palabras clave: Inteligencia Artificial, Música Electrónica, Composición Musical, AI, House.

Abstract

Artificial intelligence and the generation of artistic content through Deep Learning techniques are booming fields of study as of 2023. There exist algorithms and frameworks capable of generating various music styles trained on different genres and styles, each with its own rhythmic peculiarities. To create House rhythms, the dataset includes drum elements that follow the 4/4 time signature and percussive rhythms. Additionally, an algorithm is implemented to adjust the generated files at different temperatures (temperature understood as a parameter of probability proportionality, explained further ahead) to ensure compliance with the "Four on the Floor" criterion. Quantitative evaluation methods are established for music using metrics such as Cosine Similarity, Edit Distance or Levenshtein Distance, and a specific metric based on the algorithm employed, named the Similarity Coefficient.

Keywords: Artificial Intelligence, Electronic Music, Musical Composition, AI, House.

Índice general

Índice de ilustraciones

Índice de tablas

Capítulo 1

Introducción

1.1 Introducción

La motivación de este trabajo se inspira en la exploración de herramientas de vanguardia para el apoyo de composición musical de géneros de música electrónica. Para ello, los conocimientos necesarios son la Música Electrónica y los patrones rítmicos de la música “House”, el relacionar los mundos de la música y la Electrónica es un horizonte en exploración, y actualmente es posible diseñar herramientas que faciliten el trabajo de los compositores.

Con el crecimiento de la Música Electrónica en los últimos años, de la mano de la tecnología, también aumenta la exigencia de las composiciones, y eso requiere explorar técnicas de vanguardia que permitan mejores resultados, por tanto surgen la pregunta ¿Es posible generar ritmos mediante I.A que puedan respetar los ritmos del “House” o patrón “Four on the Floor”?, ¿Se puede medir y evaluar realmente si una Inteligencia Artificial está generando un determinado estilo?, ¿Se puede medir el arte?

Es por esto, que este estudio busca ser un paso al análisis de resultados artísticos para mejorar la precisión de los modelos de Aprendizaje Profundo enfocados a la generación de contenido.

1.2 Planteamiento del Trabajo

En la búsqueda de nuevas herramientas, se identificó la falta de patrones sólidos que respeten el ritmo del House en varias generaciones de secuencias rítmicas mediante Inteligencia Artificial, es por eso que nace el objetivo de mejorar estos resultados, ejecutando mecanismos para que los patrones cumplan los criterios correspondientes, implementando sistemas de mejora para la generación de ritmos “House”.

La particularidad de este estilo radica en la repetición de los elementos, donde el bombo suena cada un tiempo, la caja cada dos, partiendo desde el segundo, y el platillo cada uno, a contratiempo.

Si bien los algoritmos son capaces de generar estructuras en 4/4, compás correspondiente a la música House, este tiene la particularidad de que los elementos mencionados anteriormente deben estar siempre marcados, o bien, ausentarse una cantidad mínima para no perder el patrón repetitivo.

Los patrones que pueden definirse como “House” son variados, por lo que el trabajo se centra en analizar el “Four on the Floor” en su forma más estricta.

También, el desafío de una evaluación cuantitativa en generación de patrones artísticos es un desafío. Cuando se presentan secuencias de patrones que subjetivamente no presentan una similaridad con un estilo en particular, nace la necesidad de evaluar esto de una forma cuantitativa.

1.3 Estructura de la memoria

Capítulo 1:

Introducción, indicando la motivación y contenido de esta Memoria.

Capítulo 2:

Contexto y Estado del Arte, se abordan los temas a tratar en la memoria, con una revisión histórica de estos, entre ellos: Música Computacional, Deep Learning y proyectos actuales.

Capítulo 3:

Alternativas de la Solución. Se analizan las alternativas más relevantes para el desarrollo de los experimentos, seguido de la Alternativa Seleccionada.

Capítulo 4:

Metodología. En este capítulo, se define y describe la metodología a utilizar para realizar los experimentos, realizando un recorrido paso a paso.

Capítulo 5:

Experimentos y Resultados. Se describen los experimentos realizados y se ilustran los resultados obtenidos.

Capítulo 6:

Conclusiones y Trabajos Futuros. Se analizan los resultados obtenidos y se plantean recomendaciones para trabajos futuros.

Referencias:

Referencias y Bibliografía. Se citan los trabajos que dieron pie a la investigación.

Anexos:

Anexos. Contiene el script diseñado para la evaluación de los resultados e información gráfica en formato MIDI y partituras para entender mejor su funcionamiento.

Capítulo 2

Contexto y Estado del Arte

2.1 Composición Musical por Computador

En este apartado se tratan los temas e investigaciones realizadas en base a la composición algorítmica y la historia de esta aplicada en la música electrónica hasta la actualidad. La composición musical por computador tiene impacto en todos los géneros existentes, puesto que se puede utilizar para generar partituras, estructuras, arreglos, o muchas otras aplicaciones, pero es la música electrónica el género que depende 100% de la composición musical por computador, puesto que, en sus inicios, eran los sintetizadores quienes generaban el sonido interpretado por el autor, o bien, secuenciadores, que, con ciertas instrucciones, permitían interpretar una pieza musical en base a instrucciones dadas por el operador. Moog, quien en 1964 desarrolla un nuevo sistema para la creación musical electrónica basado en semi-conductores, añade que esta tecnología también puede ser útil “no solo para la composición de música electrónica directamente en cinta, sino para testear configuraciones de nuevos instrumentos musicales electrónicos para perfomance en vivo” [9]. La composición algorítmica, según Adam Alpern [10], se define como “la utilización de algún proceso formal para hacer música con una mínima intervención humana”. Previo a la

composición algorítmica, se denominaba “composición automatizada”, utilizando instrucciones y procesos formales para crear música, lo cual remonta a los antiguos filósofos de la humanidad: Pitágoras establecía una relación armónica en la naturaleza y el sonido, lo cual se interpretaba como música, mientras que Ptolomeo creía que las leyes matemáticas “subyacen en los sistemas de intervalos musicales y de los cuerpos celestes” y que cada modo musical “se corresponde con planetas concretos, sus distancias entre sí y sus movimientos” [11]. También, uno de los mayores influyentes musicales de la historia, Mozart, generó un juego musical en base a dados llamado “Musikalisches Wurfelspiel”, descrito por Alpern [10] que consistía en “ensamblar pequeños fragmentos musicales, y combinarlos de forma aleatoria, formando de este modo, una nueva pieza musical con partes elegidas al azar” En 2013, Carretero [12] investiga sobre la teoría y la aplicación práctica en la Composición Musical asistida por ordenador de dos técnicas bio-inspiradas de Inteligencia Artificial: los autómatas celulares y los P-sistemas. En su trabajo, enmarca la importancia de considerar las dimensiones no solamente técnicas, sino que también estéticas y estilísticas, buscando la manera buscando la manera de conseguir unos resultados musicales interesantes y al servicio de la expresión al mismo tiempo, enmarcados dentro del panorama de la Composición Musical actual y la herencia musical recibida

2.2 Representación y Formatos Musicales por Computador

La música es un arte que cumple patrones tanto armónicos como rítmicos, por tanto, sus partes son cuantificables. En esta sección se definen estudios de la representación en audio como simbólica, los conceptos principales de la música, con énfasis en los ritmos, los formatos que existen y la generación de datasets para el entrenamiento de las redes.

Los estudios se remontan al uso de composición algorítmica, que, en primer lugar, los primeros trabajos se basaban en Procesamiento Natural del Lenguaje, teniendo contribuciones de modelos intuitivamente plausibles como Chomsky, o modelos de n-gramas, que demostraron ser más confiables para muchos problemas del mundo real [13]. Actualmente, las técnicas se basan en aprendizaje profundo, siendo declaradas como el nuevo estado del arte en una variedad de tareas [14].

2.2.1 Formatos:

2.2.1.1 MIDI

En primer lugar, se tiene el formato MIDI, que se define como el “estándar técnico que describe un protocolo, una interfaz digital y conectores con interoperabilidad entre varios instrumentos musicales, softwares y dispositivos”[15]. Este lleva un mensaje que especifica la información en tiempo real de la nota actuada y su información de control.

- Nota Encendida: la información de una nota encendida contiene
    - Número del canal, que indica el instrumento o sonido especificado con un número entero dentro del conjunto {0,1,….,15}.
    - Número de Nota MIDI, que indica el pitch de la nota, especificado por un entero del conjunto entero {0,…,127}.
    - Velocidad, que indica la potencia de la nota a tocar, especificado por un entero del conjunto {0,…,127}

La nota encendida del protocolo MIDI se compone de 3 bytes, el primer byte es el de estado y los otros 2 son los bytes de información.

Tabla 1: Información MIDI.

BYTE DE ESTADO	BYTES DE INFORMACIÓN	
Byte #1	Byte #2	Byte #3
1 001 0000	00111100	01111111

El byte de estado describe la instrucción que se está transmitiendo al sintetizador maestro, identificando si la tecla fue oprimida o no.

El byte de información lleva el valor de la información transmitida que contiene:

- Número de Canales.
- Pitch de la nota oprimida.
- Velocidad de la Nota.

En el ejemplo ilustrado en la tabla, entonces se tiene:

Para el byte de estado, los cuatro bits más significativos (1001) corresponden al tipo de dato. El primer bit siempre es 1, que significa que es un byte de estado, los otros tres bits definen 8 categorías distintas de datos, mientras que los bits menos significativos, corresponden a 16 posibles canales de transmisión y recepción.

El Byte #2 (00111100) comienza con un 0 al igual que el #3 que significa que son bytes de información. Nuestra que la nota oprimida corresponde a un DO medio, y el byte #3 (01111111), que la nota fue oprimida con la fuerza máxima permitida [16].

- Nota Apagada: indica cuando la nota termina. En esta situación la velocidad indica que tan rápido la nota disminuye su potencia en el tiempo llamado "Release".

Tabla 2: Información MIDI DO medio.

BYTE DE ESTADO	BYTES DE INFORMACIÓN	
Byte #1	Byte #2	Byte #3
1 000 0000	00111100	01111111

En este caso, se están estudiando las dos categorías más importantes: encendido y apagado, donde el apagado se ve representado en el byte de estado como 000. Luego, el bus indica que se dejó de oprimir una nota DO medio con máxima potencia.

En 2018, Carofilis y Andrés [17], tras una serie de pruebas con distintos formatos, concluye que la mejor alternativa con los medios disponibles era utilizar música en formato MIDI.

En 2018 Hao-Wen Dong, Wen-Yi Hsiao y Yi-Hsuan Yang proponen un paquete de código abierto para Python que permite manejar pianorolls multipista para un

solo track. Definen el Piano Roll como una representación musical simbólica que registra la presencia de tonalidades en cada período de tiempo como una matriz binaria ente tonos y tiempo [18].

Esta representación permite tener una visión más gráfica de las notas musicales, que emula a lo que era un rollo de piano.

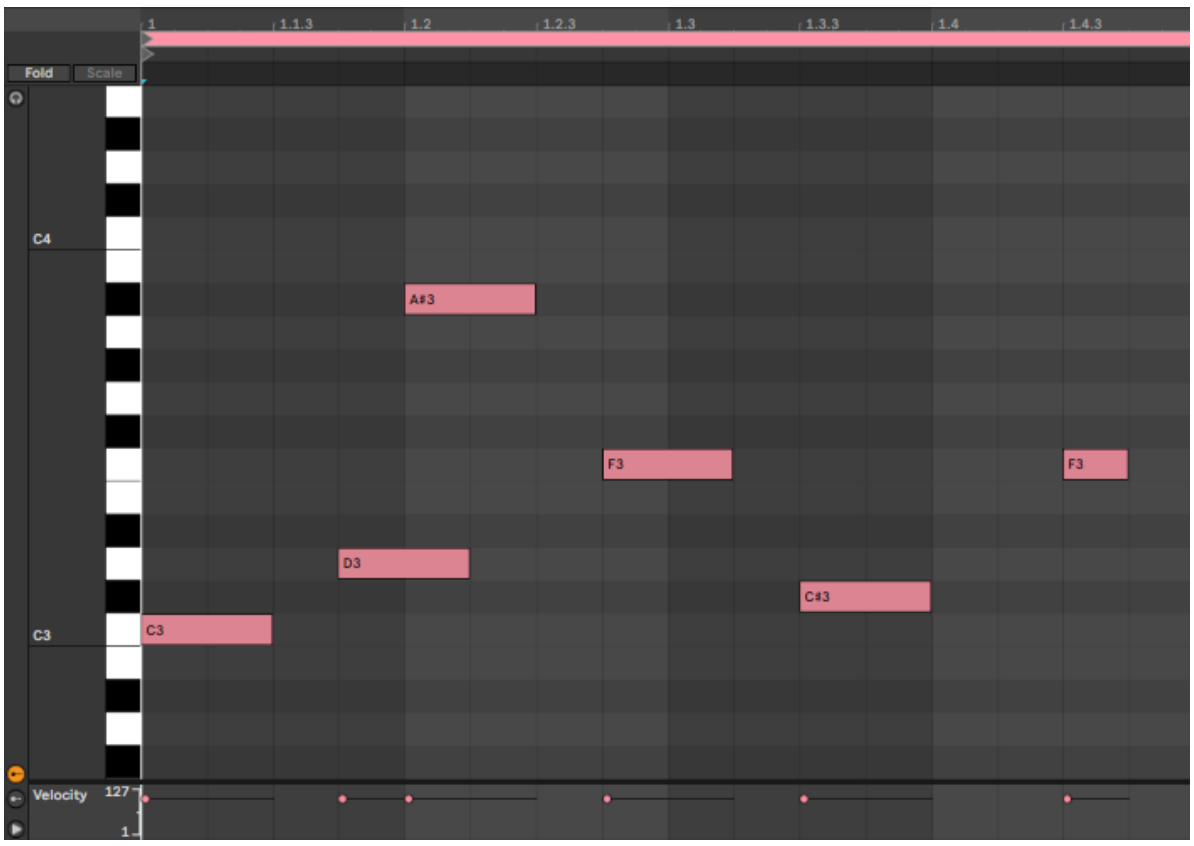


Ilustración 1: Información MIDI en formato Piano Roll. Fuente propia.

La figura corresponde al Piano Roll utilizado en el Software Ableton Live 11. Se muestra que el Eje Y corresponde a los valores de la tonalidad y el eje X corresponde al tiempo. Las líneas rojas en la parte inferior del eje representan la velocidad de la nota entre un valor de 1 a 127, correspondiente al byte de información número 3, como se definió anteriormente.

2.2.1.2 Texto en Notación americana

La notación americana es una forma muy frecuente de representar la música, sobre todo en música electrónica, que se utilizan patrones sencillos de armonía. Este sistema es más cuantificable que el tradicional, lo que permite una mejor aplicación en informática. Los modelos de Deep Learning reciben normalmente como dataset, distintos datos de forma textual, por lo que este formato es adaptable a todas las redes neuronales.

Se puede relacionar sistemas de generación automática de texto con sistemas de Deep Learning de composición musical que permiten un trabajo colaborativo de texto a música. Sturm emplea esta representación, pero añadiendo grupos de tokens para representas de manera adecuada las características de la obra [19].

La información de una nota es representada mediante vectores. En 2018, Juan Carlos García [20] desarrolla un modelo neuronal recurrente capaz de componer música de forma automática desde el enfoque del modelado del lenguaje con 20 Epochs de entrenamiento en tres pruebas distintas.

2.2.1.3 Música en formato XML

En el punto tres del apéndice se puede encontrar un ejemplo de archivo XML para una nota do.

XML es un formato que se ha utilizado para distintos proyectos de investigación. Extensible Markup Language es un metalenguaje diseñado para facilitar la definición de lenguajes relacionados con un dominio específico, como puede ser el caso de la música.

La principal ventaja de usar XML es la integración mejorada de la música y sus metadatos. En [5], se definen las ventajas que ofrece XML:

1- Se basa en la gramática.
2- Es declarativo.
3- Se estructura jerárquicamente.
4- Es modular.
5- Es extensible.
6- Es comprensible por los humanos.
7- Separa contenido y estructura de representación y comportamiento.

2.3 Arquitecturas de Deep Learning

Las investigaciones de Jean-Pierre Briot, Gaëtan Hadjeres y Francois-David Pachet se centran principalmente en el uso de Redes Neuronales [21].

Los modelos principales de Deep Learning que se utilizan actualmente en la generación musical son:

- RNN (Red Neuronal Recurrente)
- LSTM (Long Short Term Memory)
- Auto-Encoders
- Redes Convolucionales
- FeedForward
- Redes Generativas Adversarias
- Redes de Refuerzo
- Redes Neuronales Recurrentes

La arquitectura Recurrente sobresale por todas las demás por su amplio uso y capacidad de generar contenidos musicales. Estas redes fueron concebidas en 1986

pero no es hasta los últimos años que su uso se popularizó por la llegada de nuevas tecnologías, realizando distintos proyectos de inteligencia artificial en base a RNN.

Una red neuronal recurrente consiste en utilizar datos secuenciales de forma serial, la cual, en base a su memoria interna “Hiden State”, permite generar una historia de los procesos anteriores, utilizándolo para generar una nueva salida. Esto genera que la salida dependa de elementos anteriores dentro de la secuencia, generando una relación entrada-salida.

Dentro de las primeras aplicaciones de RNN en composición musical, se encuentran los trabajos realizados para el sistema CONCERT [23], seleccionando notas secuencialmente en base a una tabla de transición que especifica la probabilidad de la siguiente nota en función del contexto anterior. La red neuronal

autopredictiva es entrenada sobre un conjunto de piezas con el objetivo de extraer regularidades estilísticas.

- Redes RNN del tipo LSTM

Con la llegada de las redes LSTM (Long Short Term Memory), el trabajo de Hochreiter and Schmidhuber [24] permitió la implementación en la generación de música, que utilizaba una variante especial de redes recurrentes para decidir la cantidad de información que era tomada como información novedosa en la entrada, y cuál se mantenía como información más antigua, haciendo las redes mucho más eficientes.

Un uso de las LSTM fue aplicado en la improvisación de Blues [25], experimento en el cual se demostró que las LSTM destacaban por sobre las RNN, pues estas adolecen de una falta de estructura global ya que no pueden hacer seguimiento de la estructura musical. A diferencia, demostraron que las LSTM logran aprender rápido en el género que se les entrena. También hubo aplicación en el folk, que analizaba más de 23.000 piezas en notación de texto [26] de música celta para entrenar redes LSTM de 3 capas con 512 neuronas por cada una de ellas.

También, utilizando RNN del tipo LSTM se encuentra el trabajo denominado DeepBach [27] Es un modelo gráfico que modela música polifónica y, específicamente, piezas de himno al estilo Bach. La red fue entrenada en base a armonizaciones corales por Johann Sebastian Bach, por tanto, el modelo logra generar corales relacionados a dicho estilo, cumpliendo con restricciones de posición en las partes de soprano.

En 2018, Carofilis [17] determina que las redes neuronales LSTM demostraron ser las más efectivas en la generación de música con estructura coherente.

En 2019, Patarroyo y Yepes generan un aplicativo de una red neuronal RNN de Google Magenta, que permite a los usuarios acercarse al uso de esta aplicación, específicamente entrenando la red con canciones famosas de reggaetón. [28] Lo anterior demuestra el crecimiento del interés por la comunidad para acercar estas herramientas a todo público.

2.4 Métodos de Evaluación

La evaluación de las piezas musicales realizadas por máquinas ha estado en investigación durante todo el desarrollo de estas tecnologías, puesto que implica evaluar si la máquina realmente se comporta de manera creativa o no al ser una creadora de contenido del tipo artístico, ligándose a puntos de evaluación, por ejemplo, si la pieza musical es del agrado del evaluador, o si cumple con ciertas estructuras en base a un género musical. Los puntos anteriores generan que la evaluación sea un tanto subjetiva. Los primeros análisis de resultados de máquinas se remontan a los tiempos de Turing [29] con el conocido Test de Turing, el cual es un examen para medir la capacidad de comportamiento inteligente de una máquina, asimilándose al ser humano, o bien, indistinguible de este. El test consiste en que un humano debe evaluar una conversación en lenguaje natural entre un humano y una máquina diseñada para generar respuestas similares a las del ser humano limitada únicamente a un medio textual entre el teclado de un computador ay un monitor. El evaluador leerá la conversación entra la máquina y el humano, si este no nota diferencia entre quién es la máquina y quién es el humano, entonces, la máquina habrá pasado el Test de Turing y se podrá considerar como Inteligente, o bien, indistinguible del comportamiento humano.

En base a este test, es que ha habido distintas variaciones, por ejemplo, El Salón Chino [30] que consiste en dos interlocutores humanos comunicados a través de una computadora, pero que hablan en japonés y solamente uno d ellos interlocutores sabe el idioma, mientras que el segundo cuenta con un diccionario, que, según el texto que reciba, deberá buscar en él y responder lo que se indique. Al final del experimento se interroga al interlocutor sin conocimiento en el idioma, expresando el vago sentimiento de comunicación y que tan sólo se dedicó a transcribir y recibir instrucciones, lo cual sería el análogo de una máquina programada para comunicarse, entrando en jaque si las máquinas realmente pueden ser creativas o no. También existen métodos de métricas cuantitativas que miden el desempeño con técnicas de visualización de datos para definir, por ejemplo, el género al cual se asemeja evaluando la estructura, sin embargo, estos métodos no logran cubrir la evaluación en su totalidad, que comprende resultados psicoacústicos y sensoriales.

El problema de la evaluación de la creatividad se vuelve central, pues primero se debe definir si lo que se busca es que el observador considere que la máquina tuvo un comportamiento creativo, o si realmente se busca que la máquina sea creativa. Para lo primero, es necesario establecer ciertos parámetros que permitan definir una evaluación en el desempeño de la máquina y esta sea evaluada por un grupo de observadores. José Tapia, de la Universidad Internacional de la Rioja [31] generó un experimento de evaluación del desempeño de la generación de música flamenca a partir de redes neuronales, evaluando el desempeño en base al criterio de evaluadores profesionales, avanzados y aficionados en el rubro, determinando si existía una técnica y se cumplía la estructura del Flamenco en Soleá. Los evaluadores indicaban el porcentaje de falsetas que son consideradas flamencas, y cada porcentaje se calculaba como la media aritmética de los porcentajes obtenidos para las diferentes temperaturas, que era el parámetro que permitía complejizar en cantidad de notas la composición. Luego, se realizaban más experimentos con distintas redes y se comparaban para determinar qué red tenía el mejor desempeño para la composición de música flamenca para el estilo Soleá. Si bien la evaluación era completa en términos técnicos con distintos parámetros a evaluar, no lograba definir si la máquina había sido creativa o no.

A pesar del aporte de los trabajos mencionados anteriormente, los métodos de evaluación son subjetivos o psicológicos, por lo que es difícil seguir una rigurosidad en los resultados desde este punto de vista. Si bien, es necesaria una evaluación subjetiva en el contexto del arte y la música, también se requiere una evaluación objetiva de las pistas para poder tener una métrica que permita mejorar los resultados.

Una partitura se puede entender como una secuencia de notas, y cada nota como un evento que ocurre en un determinado momento. Bajo esta mirada, la evaluación de la música puede permitir métricas que han sido utilizadas para medir y comparar secuencias como texto o material genético. A continuación se presentan algunos de estos.

El paper titulado "A Metric for Music Notation Transcription Accuracy" [32] se centra en la propuesta de una métrica para evaluar la precisión de la transcripción de la notación musical. La métrica propuesta se basa en una **distancia de edición**, similar a las métricas utilizadas en bioinformática y lingüística, para comparar una transcripción musical con la partitura original.

La métrica se calcula en dos etapas. En la primera etapa, las dos partituras se alinean en función del contenido de la tonalidad. En la segunda etapa, se acumulan las diferencias entre las dos partituras, teniendo en cuenta doce aspectos diferentes de la notación musical: barras de compás, claves, firmas de tonalidad, firmas de tiempo, notas, notación de notas, duraciones de notas, direcciones de tallos, agrupaciones, silencios, duración de silencios y asignación de personal.

El diseño de la métrica fue guiado por un enfoque basado en datos y por la simplicidad. Para validar la relevancia y la utilidad de esta métrica, también se aplica un modelo de regresión lineal a los errores medidos por la métrica para predecir las evaluaciones humanas de las transcripciones.

El paper también discute la necesidad de una métrica objetiva para la transcripción de la notación musical, similar a la medida F estándar para la transcripción paramétrica. Además, se menciona que la evaluación subjetiva es un proceso que consume mucho tiempo y es difícil de escalar para proporcionar suficiente retroalimentación para mejorar aún más el sistema de transcripción.

El documento también menciona la **distancia Hamming**. En resumen, el documento utiliza la distancia de Hamming como una métrica de medición musical para comparar la similitud entre secuencias de música monofónica. Esta métrica se basa en contar las diferencias en los eventos musicales y puede ser extendida con una regla de compensación para abordar asimetrías específicas en la comparación de secuencias musicales.

Otras medidas de similitud ampliamente utilizadas son el Coeficiente de Dice, Jaccard y la similitud de coseno, tanto en texto, secuencias, documentos y algoritmos genéticos. Thada, V., & Jaglan, V. (2013) [33] comparan estos tres desde un punto de vista genético, cómo se pueden obtener documentos a partir de un documento o de una consulta utilizando los coeficientes de similitud.

La similitud del coseno se emplea para comparar características extraídas de las capas intermedias de una Red Neuronal Convolucional (CNN). Estas características se representan como vectores y, mediante el cálculo del producto punto entre dos vectores y su posterior normalización, se obtiene un valor entre -1 y 1. Un valor cercano a 1 indica una alta similitud, mientras que un valor cercano a 0 o -1 indica menor similitud o incluso oposición entre las piezas musicales.

En el estudio realizado por Sheikh Fathollahi y Razzazi (2021) [34], se diseñaron dos modelos de clasificación de géneros musicales utilizando esta métrica. El objetivo era extraer automáticamente características relevantes de las capas intermedias de la CNN para alimentar un sistema de recomendación de música.

La similitud del coseno resulta especialmente útil en este contexto, ya que mide la orientación de los vectores de características, independientemente de su magnitud. Esto permite determinar la similitud entre piezas musicales sin verse afectado por la longitud o estructura de las mismas.

En resumen, la similitud del coseno se ha aplicado exitosamente en la composición musical para clasificar géneros y generar sistemas de recomendación. Su capacidad para medir la similitud entre vectores de características extraídas de la música proporciona una herramienta eficaz en el análisis y comparación de piezas musicales en diversos contextos de investigación y desarrollo musical.

Un estudio clave en este campo propone un marco de evaluación que emplea características musicales para una evaluación objetiva y reproducible de estos modelos (Yang et al., 2020) [35]. Este marco utiliza métricas de conteo y dominio musical, así como la Divergencia de Kullback-Leibler (KLD) y el Área de Superposición (OA) para comparar las características extraídas de los datos generados con los del conjunto de entrenamiento. Este enfoque permite una comprensión más profunda de la eficacia de los modelos generativos y su capacidad para capturar aspectos relevantes de los datos de entrenamiento.

2.5 Desarrollos Actuales

La Industria Musical en la actualidad cuenta con una gran variedad de empresas digitales que están trabajando en la investigación y desarrollo de plataformas para la música mediante Inteligencia Artificial, esto ha llevado a un gran crecimiento en innovación dentro de los últimos años con plataformas de distintas características, algunas sirviendo como herramientas para intérpretes, y otras utilizando a la misma inteligencia artificial como intérprete. A continuación, se presenta una tabla con las empresas destacadas y su descripción:

Empresa	Proyecto	Año	País
Google	Magenta	2016	Estados Unidos
IRCAM-ACIDS	NeuroRack	2022	Francia
Spotify	Creator Technology Research Lab (CTRL)	2017	Suecia
Sony DrumGAN	Flow Machines by CSL	2016	Japón
OpenAI	JukeBox	2020	Estados Unidos

Tabla 3: Desarrollos Actuales.

GOOGLE MAGENTA

Debido a que las redes neuronales a utilizar pertenecen a Google Magenta, se procede a ahondar más en este proyecto:

Magenta es un proyecto de investigación de código abierto que utiliza el machine learning como una herramienta para el proceso creativo. El objetivo fundamental es dotar a los artistas de un conjunto de modelos y utilidades que los permitan desarrollar su creatividad, generando una comunidad entre investigadores, ingenieros y artistas. Esta información se encuentra en el sitio oficial de Google Magenta (https://magenta.tensorflow.org/).

En 2019, Patarroyo y Yepes generan un aplicativo de una red neuronal RNN de Google Magenta, que permite a los usuarios acercarse al uso de esta aplicación, específicamente entrenando la red con canciones famosas de reggaetón [28].

En el año 2021 hubo una serie de música en plataformas compuesta por estos algoritmos, entre ellos MJ Jacob, un rapero logró millones de streams en 2021. Aparna Kumar crea un proyecto de danza clásica india “Bharatantyam” donde la música utilizada fue en base a inteligencia artificial de Magenta. En 2021, José Tapia genera una comparación entre una red de Magenta y una LSTM para la composición de música flamenca [31]. Actualmente cuenta con proyectos visuales como también musicales. Proporciona una gran cantidad d emodelos para la composición musical, dentro de ellos están:

- Coconet: Red neuronal convolucional que completa partituras parciales
- Drums RNN: red neuronal LSTM que modela pistas de batería.
- GANSynth: Sintetiza audio mediante redes de tipo GAN.
- Melody RNN: red neuronal LSTM que modela pistas de melodías
- Improv RNN: las melodías generadas se ven condicionas de una progresión de acordes.

- Music VAE: genera un muestreo aleatorio a partir de una distribución a priori, interpolación y manipulación de secuencias.

- NSynth: modelo de síntesis de audio.

- Performance RNN: Activa, desactiva y cambia la velocidad de las notas.

- Onsets and Frames: Transcribe música de piano automáticamente

- RL Turner: Modelo LSTM que predice la nota siguiente de una melodía monofónica.

SONY DRUMGAN

J. Nistal, S. Lattner y G. Richard presentaron un trabajo titulado “DrumGAN: Synthesis of Drum Sounds with Timbral Feature Conditioning Using Generative Adversarial Networks” en las Actas de la 21ª Conferencia Internacional de la Sociedad para la Recuperación de Información Musical (ISMIR) en octubre de 2020 [36]. En este trabajo, los autores proponen un método para crear sonidos de batería sintéticos (como en las máquinas de ritmos) usando una GAN que se entrena con una gran colección de sonidos de bombo, caja y platillo. La GAN se condiciona con características perceptuales extraídas con un extractor de características público, lo que permite controlar el proceso de generación de forma intuitiva. Los autores muestran que su método mejora la calidad de los sonidos generados respecto a un trabajo previo basado en una arquitectura U-Net, y que la entrada condicional influye en las características perceptuales de los sonidos. Los autores proporcionan ejemplos de audio y el código usado en sus experimentos. Las métricas mencionadas (Inception Score, Kernel Inception Distance y Fréchet Audio Distance) se utilizan para evaluar la calidad y la diversidad de las muestras generadas por modelos generativos en el campo de la música. Estas métricas se basan en la clasificación de muestras, la comparación de características extraídas

y la evaluación de la similitud entre distribuciones de características. Se utilizan para evaluar la calidad del audio generado.

Capítulo 3

Objetivos

4.1 Objetivo General

El objetivo general de este trabajo es implementar un sistema de generación y evaluación de archivos MIDI a partir de una red neuronal que cumpla con el patrón “Four on the Floor” de la música electrónica.

4.2 Objetivos Secundarios

Para el desarrollo de este trabajo se establecen los siguientes objetivos específicos:

1. Aumentar diversidad y representatividad del Dataset.
2. Determinar rango de Temperatura para lograr equilibrio entre diversidad y coherencia de las pistas generadas.
3. Evaluar cuantitativamente los resultados mediante un sistema de Python.

Capítulo 4

Alternativas de la Solución

3.1 Alternativas

Se estudió y seleccionó el conjunto de alternativas que permites generar una estrategia viable para la solución al problema planteado. El objetivo consiste en analizar bajo distintos criterios las posibles soluciones al problema planteado. Se presentan los proyectos con los que se puede trabajar y se definen criterios y condiciones para identificar el alcance que tiene cada uno de ellos para atender la necesidad. Los proyectos deben ser susceptibles de generar por sí mismos beneficios en la investigación y evaluación de los resultados mediante la configuración de parámetros razonables en el período de tiempo presupuestado y cumplir con los objetivos. Los objetivos que deben alcanzar los proyectos son:

• Trabajar con algoritmos de Deep Learning como LSTM o CNN que permita crear ritmos de música House.

• Añadir información adicional correspondiente al House y analizar estructuras de codificación para generar Datasets en los formatos requeridos para el entrenamiento.

• Analizar distintos tipos de Evaluación de Creatividad y Estructura acorde al ritmo de la música House que permita la alternativa

• Crear composiciones musicales del estilo House utilizando las herramientas evaluadas

La Industria Musical en la actualidad cuenta con una gran variedad de empresas digitales que están trabajando en la investigación y desarrollo de plataformas para la música mediante Inteligencia Artificial, esto ha llevado a un gran crecimiento en innovación dentro de los últimos años con plataformas de distintas características, algunas sirviendo como herramientas para intérpretes, y otras utilizando a la misma inteligencia artificial como intérprete. Para su análisis, se enmarcó el estudio en las siguientes variables.

Presupuesto: Corresponde al cálculo anticipado del coste en inversión. La alternativa debe cumplir con el presupuesto acorde a lo definido por parte de los profesores para la realización de la Memoria de Titulación y se debe comparar con las demás alternativas para tomar una decisión que minimice los gastos para el cumplimiento del mismo objetivo.

Tiempo: Debido al acotado tiempo que se cuenta para realizar el proyecto, las alternativas deben cumplir con el tiempo presupuestado para una Memoria de Titulación. No debe exceder el tiempo asignado y debe tener relación con los resultados que se quieren alcanzar

Evaluación: El proyecto debe ser evaluable, es decir, debe permitir cierta libertad en sus parámetros para poder comparar y poder determinar el desempeño de su trabajo mediante análisis cuantitativo.

Parametrización: La alternativa debe cumplir con cierta libertad por parte del usuario que permita manipular diversos valores y así poder un estudio más complejo de la herramienta utilizada y cómo ésta permite la obtención de resultados expuesta a distintas pruebas

En el siguiente apartado se presentan los proyectos e ideas actuales que a priori cumplen con el objetivo del trabajo. La selección de estas alternativas se basó en trabajos previos enmarcados en el Estado del Arte de este proyecto

Generar red LSTM específica.

La generación de una red LSTM específica permite un flujo de transformación de los datos para cumplir con el objetivo específico y no depender de posibles procesos innecesarios y parametrizaciones irrelevantes en el camino a la creación de ritmos de música electrónica. Para este caso se deben definir los bloques que componen a la red, entrenarla y posteriormente probarla.

Presupuesto: Al tratarse del diseño e implementación de una red neuronal, no se requieren elementos ni licencias que no estén disponibles.

Tiempo: Debido a la complejidad del diseño de una red neuronal, el tiempo es una variable que afecta a esta alternativa, pues para el acotado tiempo que se tiene, los problemas que se pueden encontrar en el camino pueden dificultar mucho el lograr resultados.

Parametrización: La parametrización en este caso, está completamente diseñada para cumplir los objetivos específicos del trabajo, es por ello que es una red LST específica. En este caso, los parámetros a diseñar corresponden a un entrenamiento a partir de ritmos musicales de house y un factor de temperatura en el bloque de salida que permite generar ritmos más o menos complejas, con mayor o menor cantidad de elementos por segundo.

Evaluación: La evaluación realizada con esta herramienta es bastante completa, puesto que la manipulación de los parámetros fue establecida de tal manera que se cumpla lo especificado y lo que se busca como resultado.

Google Magenta

Magenta es un proyecto de investigación de código abierto que utiliza el machine learning como una herramienta para el proceso creativo. El objetivo fundamental es dotar a los artistas de un conjunto de modelos y utilidades que los permitan desarrollar su creatividad, generando una comunidad entre investigadores, ingenieros y artistas. Esta información se encuentra en el sitio oficial de Google Magenta (https://magenta.tensorflow.org/).

Magenta permite generar sistemas de generación musical como asistentes y también modelos autónomos a partir de distintos bloques, lo que lo convierte en una alternativa flexible y adaptable a todo tipo de género.

Actualmente cuenta con proyectos visuales como también musicales. Proporciona una gran cantidad de modelos para la composición musical, dentro de ellos están:

• Coconet: Red neuronal convolucional que completa partituras de manera parcial.

• Drums RNN: red neuronal LSTM que modela pistas de batería.

• GANSynth: Sintetiza audio mediante redes de tipo GAN.

• Melody RNN: red neuronal LSTM que modela pistas de melodías

• Improv RNN: las melodías generadas se ven condicionas de una progresión de acordes.

• Music VAE: genera un muestreo aleatorio a partir de una distribución a priori, interpolación y manipulación de secuencias.

• NSynth: modelo de síntesis de audio.

• Performance RNN: Activa, desactiva y cambia la velocidad de las notas. • Onsets and Frames: Transcribe música de piano automáticamente

• RL Turner: Modelo LSTM que predice la nota siguiente de una melodía monofónica.

• Score2perf: Colección de problemas Tensor2Tensor para generar interpretaciones musicales que pueden ser condicionadas por una partitura.

Presupuesto: cuenta con todo su repositorio abierto al público, por lo que no se debe invertir en aquello. Se debe tener en cuenta que, como existe poca información sobre su funcionamiento, se debe invertir en manuales de uso disponibles en internet.

Tiempo: El uso de magenta es la frontera entre lo complejo y lo amigable, pues está hecho tanto para músicos como para investigadores e ingenieros de sistemas, es por ello que, en temas de tiempo, se convierte en la mejor opción debido a la dificultad adaptable que puede generar y equilibrarlo con resultados complejos.

Parametrización: Magenta cuenta con distintas alternativas para la composición musical, entre ellas el modelo Drums_RNN, que a su vez cuenta con los modelos “one_drum” y “drum_kit” si se quiere utilizar elementos en singular o una paleta de elementos para generar los ritmos, y como cuenta con distintas redes, estas pueden ser combinadas para distintos resultados. Por ejemplo, MusicVAE permite interpolar entre dos patrones rítmicos distintos y GrooVAE permite añadir “Groove” a la secuencia, que se refiere a cambiar la posición de las notas a destiempo y variar la intensidad de esta. Estas combinaciones permiten resultados variados y manipulables.

Evaluación: Debido a la extensa parametrización y la existencia de redes exclusivas para la generación de ritmos, se puede evaluar claramente el objetivo y los distintos resultados a distintos parámetros para la creación de ritmos de música electrónica.

MuseNet

Es la alternativa de aprendizaje profundo que soporta el formato MIDI, puede generar pistas a partir de entradas, identificar estos patrones, y continuar desarrollando una composición en base a esta. Ambién puede trabajar con baterías. Su código Fuente está abierto al público y e formato de Jupyter Notebook y usuarios han colaborado en la facilitación de su código. Tiene hasta 9 insturmentos para entrenar y cuenta también con baterías.

Presupuesto: cuenta con todo su repositorio abierto al público, por lo que no se debe invertir en aquello.

Tiempo: Al igual que Magenta, funciona con archivos MIDI, que aceleran el proceso de entrenamiento, en vez de utilizar audio.

Parametrización: La parametrización de la generación de datos es más limitada que la de magenta, el cual cuenta con las variaciones mencionadas anteriormente, mientras que MuseNet se restringe a una generación en base al entrenamiento.

Evaluación: La evaluación de los resultados, al ser archivos MIDI se facilita, pues, se puede diseñar un algoritmo de análisis .csv para identificar los patrones en la salida.

Existen otras alternativas similares a Google magenta, como DrumGAN y Neural Drum Machines de Sony CSL, pero sus directorios se encuentran muy desactualizados y con un bajo número de interacciones por parte de la comunidad, por lo que se descartan de una selección más específica.

Al igual que los anteriores, existen otros frameworks a la fecha, como MuseNet de OpenAI, o bien MusicLM, pero estos no cuentan con el código abierto, por tanto también se descartan.

Elección de métricas de Evaluación

También, se debe realizar una selección de las métricas de evaluación para el caso particular de la elección del framework a utilizar. Las métricas analizadas en el estado del arte correspondían a métricas tanto para archivos de audio como para secuencias, entre las más relevantes están:

a. Similitud de coseno
b. Distancia de Hamming
c. Divergencia de Küllback-Leiber
d. Área de superposición
e. Coeficiente de dice
f. Coeficiente de Jaccard
g. Distancia de edición
h. Inception Score
i. Kernel Inception Distance
j. Frechet Audio Distance

Los criterios para definir una correcta métrica de evaluación debe ser la elección entre ellas que evite una alta multicolienalidad y una relación en cuanto al formato que se generará, puesto que hay algunas que son específicas para audio o bien secuencia de eventos.

3.2 Alternativa Seleccionada

Dentro de las características de comparación, el presupuesto, que corresponde al cálculo anticipado del coste de inversión, no presentaba una dificultad o desventaja en las 3 ideas planteadas. El Tiempo, sí corresponde a un factor determinante debido a la relación tiempo/resultados eligiendo las distintas opciones. La evaluación de los resultados también es determinante, debido al material que se puede entregar con las herramientas. Y finalmente, la parametrización también marca la diferencia si lo que se busca es enfocarse en la creación de ritmos de música electrónica.

la alternativa seleccionada es Google Magenta. Magenta cuenta con librerías de redes neuronales de código abierto para la generación de archivos MIDI rítmicos, lo que es bastante específico y es justamente el experimento que se quiere realizar en este trabajo. En base a esta alternativa seleccionada, el objetivo es generar pistas MIDI a partir de un dataset de música electrónica, específicamente, los Paquetes de Muestras utilizados por el autor para la generación de su propia música, utilizando esta herramienta con un enfoque en el género del autor. Luego, se entrenará la red neuronal en un entorno GPU. El tiempo de entrenamiento es de aproximadamente 5 horas con el dataset deseado. Finalmente, se evaluarán los resultados y se compararán también con los modelos preentrenados de magenta y ver cuál se adapta mejor al estilo del autor.

En cuanto a las métricas de evaluación, las métricas seleccionadas a utilizar deben ser de naturaleza de evaluación de secuencias, y deben evitar la multicolinealidad. Las métricas estudiadas que cumplen con este criterio son:

a) Similitud de Coseno
b) Distancia de Hamming
c) F1-Score

Cada una de ellas es explicada en la metodología.

Capítulo 5

Metodología

5.1 Metodología

A continuación, se describen los pasos dados y las herramientas utilizadas para conseguir los objetivos indicados anteriormente

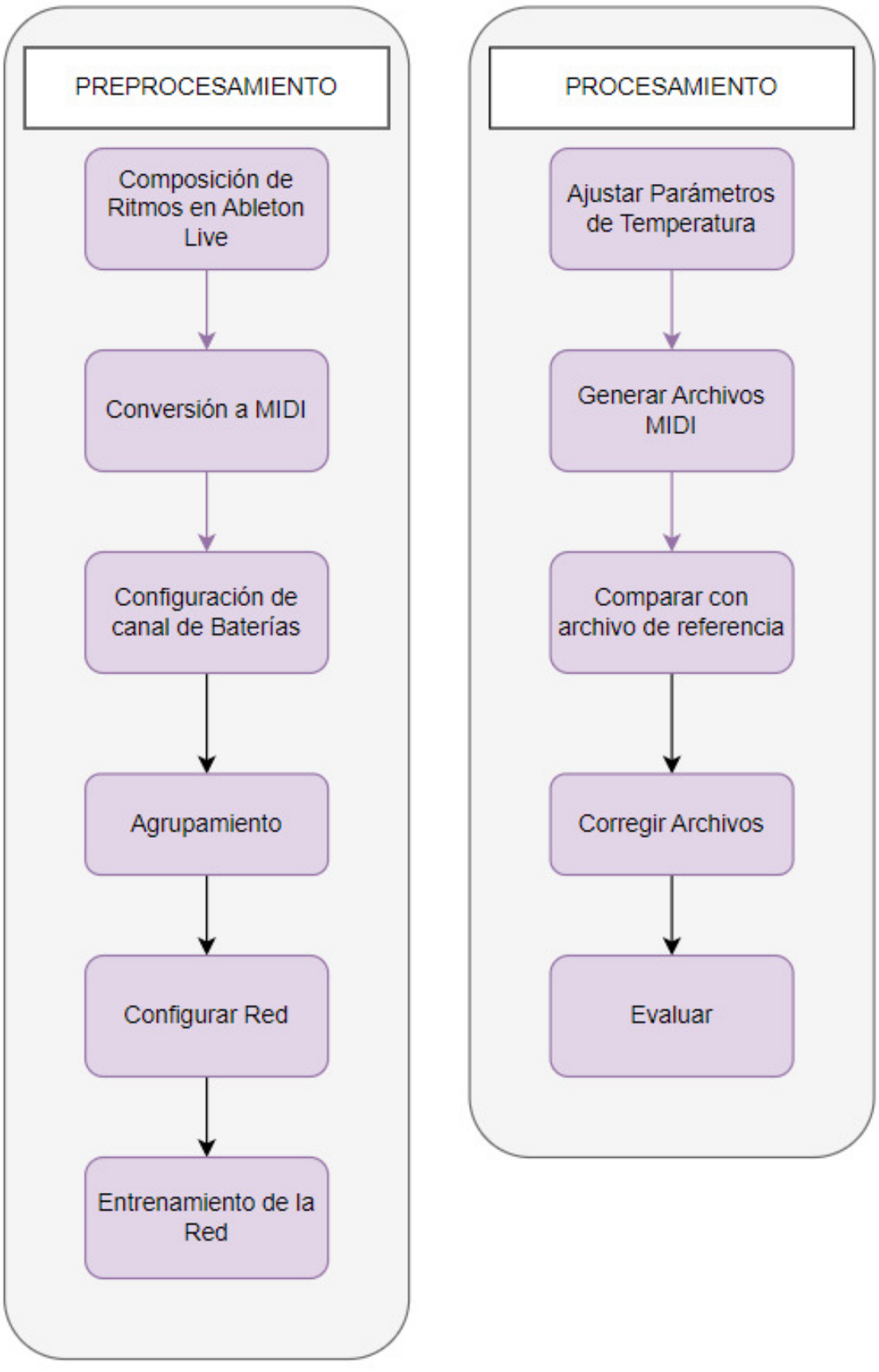


Ilustración 2: Diagrama de Flujo de pasos a realizar en el trabajo. Fuente propia.

5.1.1 Pre-Procesamiento

5.1.1.1 Composición de Ritmos

En primera instancia, se genera un dataset de entrenamiento mediante pistas del set abierto al público LMD-FULL Dataset, con un total de 512 pistas. Luego se genera otro dataset que corresponde al anterior más la suma de un 20% de canciones que cumplen con el criterio “Four on the Floo”. Estas se realizan de forma manual por parte del equipo conformado por 3 productores de música electrónica con altos conocimientos en composición musical utilizando el software Ableton Live. También, se explorarán datasets de archivos MIDI que cumplan con los ritmos House. Todas las pistas son para batería, por lo tanto, polifónicas, esto quiere decir que varias notas pueden tocarse simultáneamente.

5.1.1.2 Conversión a MIDI y Configuración de canal de Baterías

Los ritmos creados son posteriormente exportados como archivos .MIDI debido a la alta manipulación de este formato. Por defecto, el programa Ableton Live no permite exportar indicando el canal del instrumento, por lo que los archivos son recibidos por un algoritmo que modifica el valor del canal a 10, correspondiente a batería según el estándar MIDI.

5.1.1.3 Agrupamiento

El primer paso será convertir una colección de archivos MIDI en NoteSequences. Las NoteSequences son protocol buffers, un formato de datos rápido y eficiente, y más fácil de trabajar que los archivos MIDI. El dataset se convierte a un archivo .tfrecord.

A continuación, crearemos SequenceExamples. Los SequenceExamples se utilizan para alimentar al modelo durante el entrenamiento y la evaluación. Cada

SequenceExample contendrá una secuencia de entradas y una secuencia de etiquetas que representan una pista de batería.

Se generarán dos conjuntos de SequenceExamples, uno para entrenamiento y otro para evaluación, donde la fracción de SequenceExamples en el conjunto de evaluación se determina mediante el parámetro --eval_ratio. Con una relación de evaluación de 0.10, el 10% de las pistas de batería extraídas se guardarán en el conjunto de evaluación y el 90% se guardarán en el conjunto de entrenamiento. De esta manera, con un conjunto de prueba será posible evaluar el rendimiento del entrenamiento e identificar si está sobreajustado o no.

Los tamaños de los datasrt son los siguientes:

1) Dataset 1: 512 archivos de baterías de distintos estilos
2) Dataset 2: 704 archivos de baterías de distintos estilos enriquecido con ritmos House al 20%.

Cada uno con un conjunto de evaluación del 10%.

5.1.1.4 Configurar Red

Se configuran los elementos de entrenamiento y se define un 10% del dataset para su evaluación. Los hiperparámetros utilizados son los que entrega Google Magenta por defecto, en caso de obtener problemas de sobreajuste, estos serán modificados.

5.1.1.5 Entrenamiento de la Red

Se define el entrenamiento de la red Drums_RNN:

1) Se probarán distintas configuraciones proporcionadas por le modelo. El número de capas, neuronas por cada capa y tamaño de batch utilizado es el utilizado por defecto.

2) Se obtendrán los resultados del entrenamiento y la evaluación, en base a ello, se ajustarán los parámetros de entrenamiento y dataset para corregir posibles falencias.

6.1.2 Procesamiento

5.1.2.1 Ajuste de Parámetro de Temperatura

Se determinará el rango de Temperatura a evaluar y se configurará en la red. Este hiperparámetro lo poseen las LSTM. Este genera variabilidad en las predicciones generadas en la capa “Softmax”. Si es 0, los valores que se toman son los normalizados por la capa, a medida que aumenta, los valores de probabilidad pequeños disminuyen, y los grandes aumentan, generando una distribución de probabilidad más suave en las clases, excitando a la red y generando mayor diversidad, pero también más errores. En el contexto de la música, este parámetro añade mayor diversidad musical, añadiendo más instrumentos que los ingresados en la entrada, pero también mayor aleatoriedad, que puede dirigir a una composición muy distinta a la de entrada.

5.1.2.2 Generar Archivos MIDI.

Se generará un total de 10 archivos MIDI por cada valor de temperatura, estos serán agrupados para su posterior comparación.

5.1.2.3 Corregir Archivos

Los archivos serán entrada a un algoritmo de Python que determina si el archivo debe ser corregido o no, si este supera el 10% de diferencia con el patrón de referencia, que corresponde a un archivo midi 4/4 de la misma duración que el archivo de entrada. Al ser corregido, en caso de faltar elementos en el patrón, estos

son agregados para reducir la diferencia al 0%. En caso de superar el doble de uno de los elementos, se elimina el exceso de ese instrumento.

5.1.2.4 Evaluar

La evaluación se realizará de manera objetiva con tres métricas de comparación para medir la similitud de los resultados con una pista de referencia que contiene los elementos obligatorios para que una pista sea considerada “House”. Esta secuencia corresponde al patrón “Four on the Floor”. Las pistas serán comparadas con esta utilizando:

1) Distancia de Edición o de Levenshtein

El algoritmo transforma en arreglos a la pista generada y la pista de referencia. Luego, utilizando el algoritmo de Distancia de Edición, se determina el valor que define cuántas modificaciones se deben hacer para que una pista se convierta en otra. Mientras menor es el valor, menos ediciones se deben realizar, por tanto, mayor similitud. Y mientras mayor sea el valor, menor similitud tendrán las pistas.

Sea D la distancia de edición, x e y representan las dos cadenas que se están comparando.

$$D(x,y) = |x| si\ |y| = 0$$
$$D(x,y) = |y| si\ |x| = 0$$
$$D(x,y) = D(x[1], y[1:] si\ x[0] = y[0]$$
$$D(x,y) = 1 + \min\big(D(x[1:], y), D(x, y[1:]), D(x[1:], y[1:])\big)\ otro\ caso$$

2) Similitud de Coseno

Similar a la distancia de edición, las pistas son transformadas a vectores , y posteriormente normalizadas en one-hot encoding. De esta manera se calcula el coseno del ángulo mediante el algoritmo correspondiente y se asignan los valores.

Un valor cercano a 1 muestra similitud en las pistas, cercano a 0 muestra diferencia entre las pistas, y cercano a -1 muestra una oposición entre ellas. Sea A el vector de la pista de referencia y B el vector de la pista generada, entonces:

$$S.C = cos\theta = \frac{A * B}{|A| * |B|}$$

El coseno de similitud es una medida que va de -1 a 1, donde:

-1 indica similitud inversa o totalmente opuesta.

0 indica que no hay similitud entre los vectores.

1 indica similitud perfecta o completa identidad.

3) Coeficiente de Similitud de Eventos

Esta métrica es diseñada en el algoritmo que cuenta los falsos negativos (N), y a diferencia de las otras, permite obtener una medida de similitud que pondera en igual manera la importancia de los instrumentos equitativamente en la partitura.

El algoritmo cuenta la cantidad de eventos faltantes (falsos negativos) para cumplir con el criterio de la referencia, recorriendo la pista generada de izquierda a derecha e identificando los elementos que faltan para contener a la partitura de referencia dentro de ella.

$$CSP = \frac{100}{I}\sum_{i=1}^{I}\left(1 - \left(\frac{N_i}{M_i}\right)\right)$$

CSP: $Coeficiente\ de\ Similitud\ Promedio$

I: $Número\ de\ instrumentos\ totales$

Ni: $Número\ de\ eventos\ faltantes\ en\ la\ pista\ generada$

Mi: $número\ de\ eventos\ totales\ en\ la\ pista\ de\ referencia$

Este algoritmo se puede explicar utilizando el concepto de operaciones de conjuntos. Vamos a considerar dos conjuntos: el conjunto de eventos en la partitura de referencia (M) y el conjunto de eventos en la pista generada (N).

$$CSP = 100\% \text{ Si } N \cap M = N$$
$$CSP = 0\% \text{ Si } N \cap M = \emptyset$$

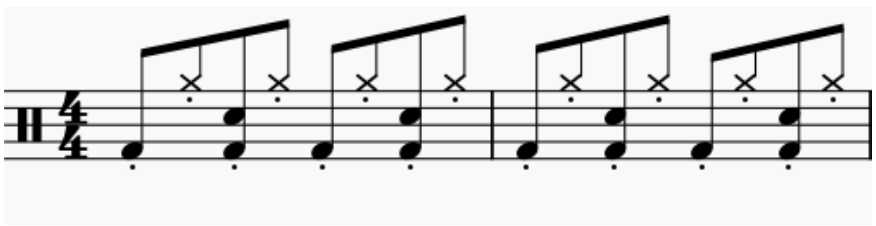


Ilustración 3: Partitura para dos compases del patrón "Four on the Floor". Fuente propia.

Sumado a estas métricas, se utilizarán las altamente utilizadas en evaluación de redes neuronales que son la Precisión, Recall y F1-Score. Estas métricas se basan en los conceptos de verdaderos positivos (TP), falsos positivos (FP) y falsos negativos (FN).

Un **verdadero positivo (TP)** ocurre cuando el modelo predice correctamente la presencia de un evento. En el contexto de la generación de música, un verdadero positivo podría ser una nota generada que coincide con una nota en una pieza musical de referencia.

Un **falso positivo (FP)** ocurre cuando el modelo predice incorrectamente la presencia de un evento. En el contexto de la generación de música, un falso positivo podría ser una nota generada que no coincide con ninguna nota en la pieza de referencia.

Un **falso negativo (FN)** ocurre cuando el modelo predice incorrectamente la ausencia de un evento. En el contexto de la generación de música, un falso negativo podría ser una nota en la pieza musical de referencia que no fue generada por el modelo.

La precisión P se calcula como el número de verdaderos positivos (TP) dividido por la suma de los verdaderos positivos y los falsos positivos (FP). Es una medida de cuántas de las notas generadas por el modelo son correctas, es decir, cuántas de las notas generadas coinciden con las notas en la secuencia de referencia. La fórmula para la precisión es:

$$P = \frac{TP}{TP + FP}$$

El recall R se calcula como el número de verdaderos positivos dividido por la suma de los verdaderos positivos y los falsos negativos (FN). Mide cuántas de las notas en la secuencia de referencia fueron correctamente generadas por el modelo. La fórmula para el recall es:

$$R = \frac{TP}{TP + FN}$$

El F1-score $F1$ combina la precisión y el recall en una sola métrica que proporciona un equilibrio entre ambas. Se calcula como la media armónica de la precisión y el recall. La fórmula para el F1-score es:

$$F1 = 2 * \frac{P * R}{P + R}$$

Al utilizar estas seis métricas, podremos obtener una evaluación completa y equilibrada de la calidad de las secuencias de batería generadas por el modelo.

5.3.3 Requisitos Funcionales

Para realizar los pasos anteriores, se dividió el trabajo en distintos módulos, de esta manera se puede probar de forma independiente, las entradas y salidas de cada uno de ellos.

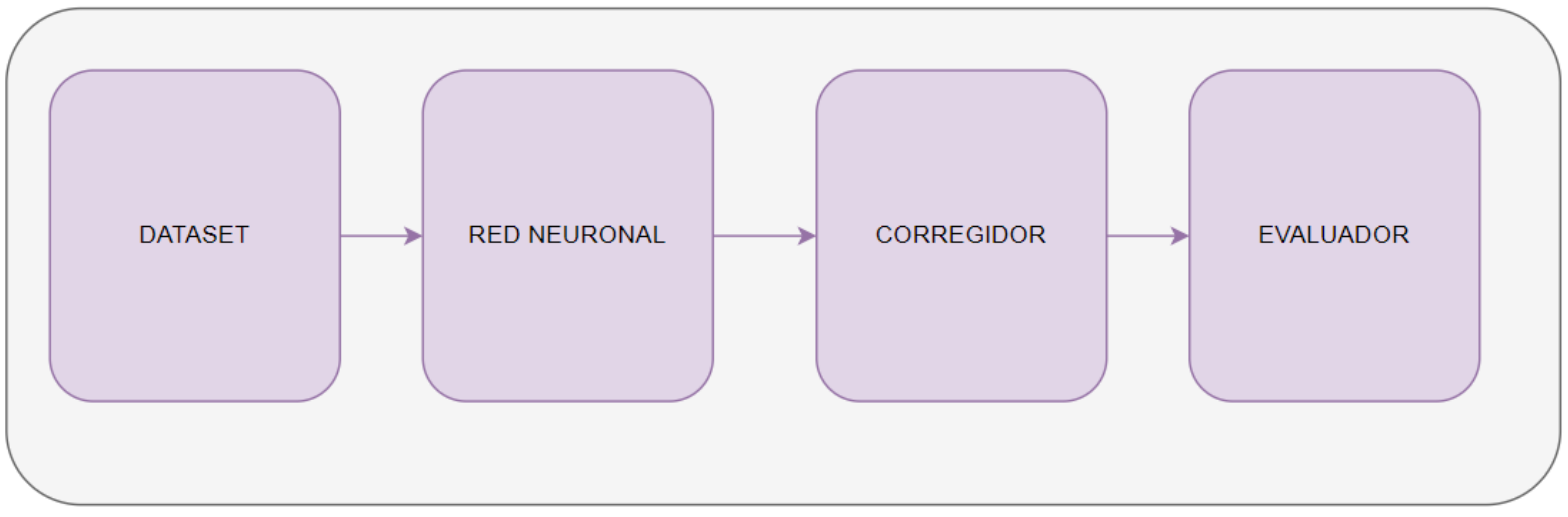


Ilustración 4: Módulos del Sistema. Fuente propia.

5.1.3.1 Requisito funcional DATASET

RP1: Se debe obtener un archivo. tfrecord que no descarte elementos por formato.

RP2: Se debe obtener archivos MIDI configurado en canal 10 según el estándar MIDI.

5.1.3.2 Requisito funcional RED NEURONAL

RP1: obtener un archivo de entrenamiento con los parámetros por defecto que permite la generación.

RP2: obtener los resultados de evaluación del entrenamiento con una Función Loss por debajo del 1 al finalizar el entrenamiento.

5.1.3.3 Requisito funcional CORREGIDOR

RP1: entregar la información en pantalla de los resultados de correlación

RP2: entregar 10 pistas MIDI corregidas que cumplan con el criterio “Four on the Floor”

5.1.3.4 Requisito funcional EVALUADOR

RP1: Mostrar en pantalla el valor de correlación por lote correspondiente a la temperatura seleccionada.

RP2: Mostrar la efectividad de las pistas en relación a la de referencia.

En base a los puntos anteriores, se muestra le diagrama de flujo que ilustra al algoritmo diseñado para la evaluación y corrección

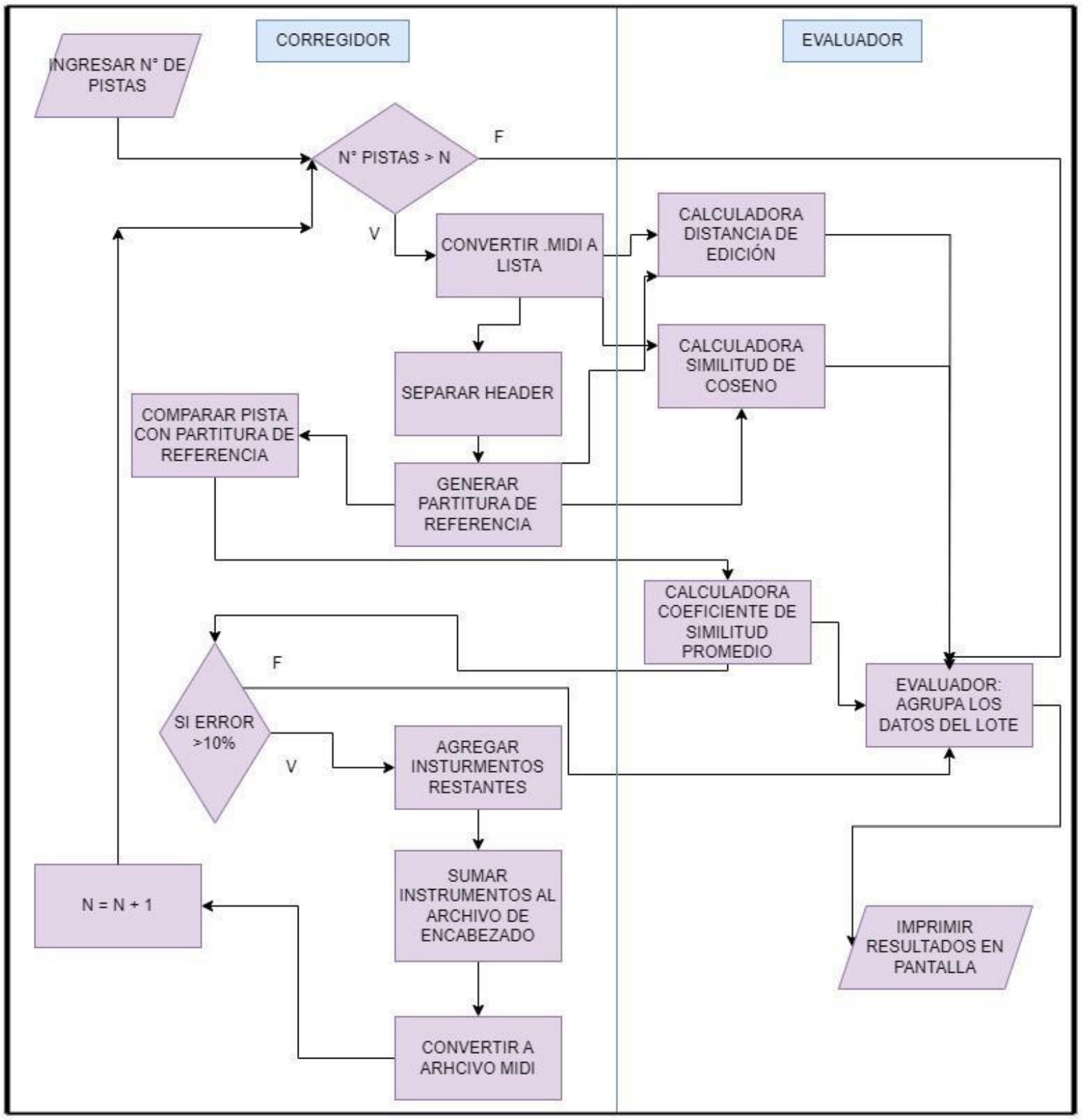


Ilustración 5: Diagrama de Flujo de algoritmo. Corregidor a la izquierda y Evaluador a la derecha. Fuente propia..

Capítulo 6

Experimentos y Resultados

6.1 Descripción de Experimento

El experimento consta en generar pistas luego de la etapa de pre-procesamiento. Se generaron lotes de 10 pistas a distintos valores de temperatura. Posteriormente, estos archivos son recibidos por el módulo corregidor que modifica las pistas para que cumplan con el criterio, y posterior a ello, se entrega el resultado de las métricas respecto a la referencia. Los resultados son promediados para determinar el CSP para cada temperatura y finalmente, determinar el rango adecuado de operación de temperatura para generar las pistas que cumplen con el criterio. Adicionalmente, se compara el resultado de las métricas con los dos dataset para determinar el porcentaje de mejora de la red.

6.2 Resultados

Los parámetros utilizados para el entrenamiento de la red con el dataset 2 son:

- Tamaño de batch: 64
- Etapas de entrenamiento: 20.000

La red fue entrenada hasta conseguir:

Precisión del Entrenamiento = 96%

Función Loss = 13.06%
Valor del 0.09 a la etapa 21140

Se utilizaron los valores de 0.25, 0.5, 1, 1.5, 2, 3, 4, 5 de variación del hiperparámetro de temperatura para los experimentos.

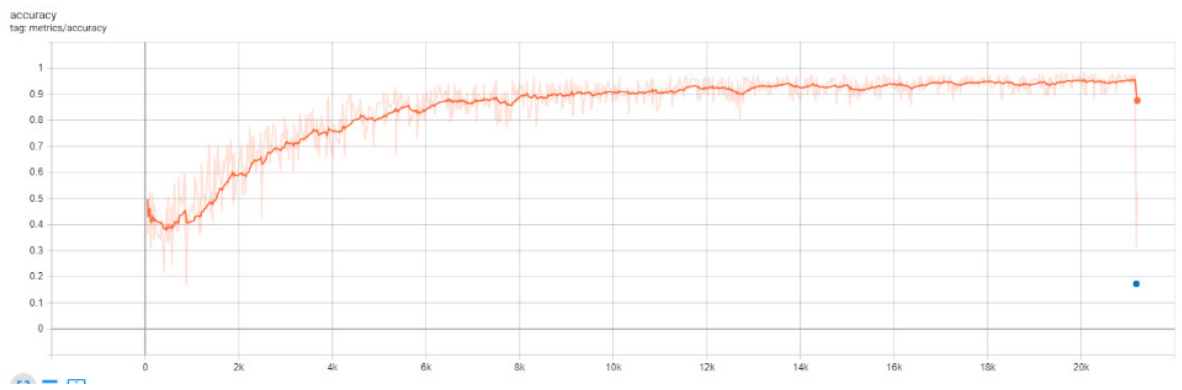


Ilustración 6: Gráfico de Precisión para dataset 2. Fuente Propia.

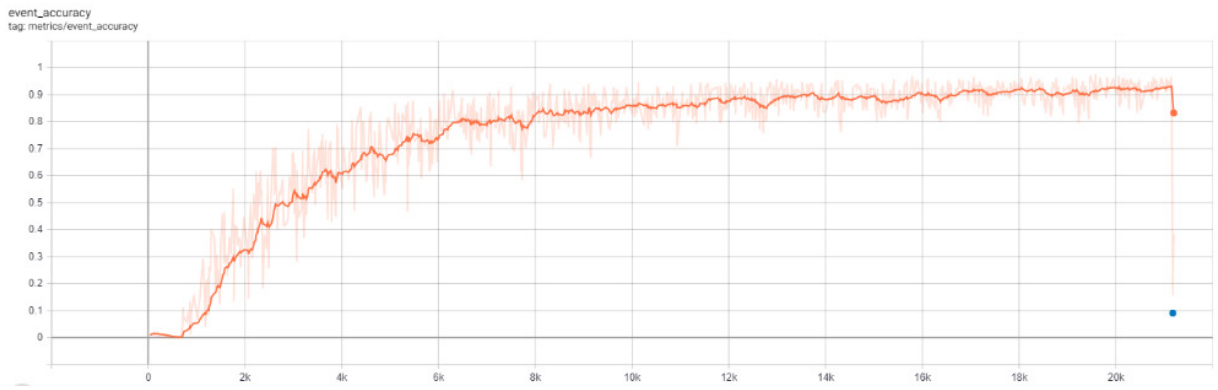


Ilustración 7: Gráfico de precisión de evento. Fuente propia.

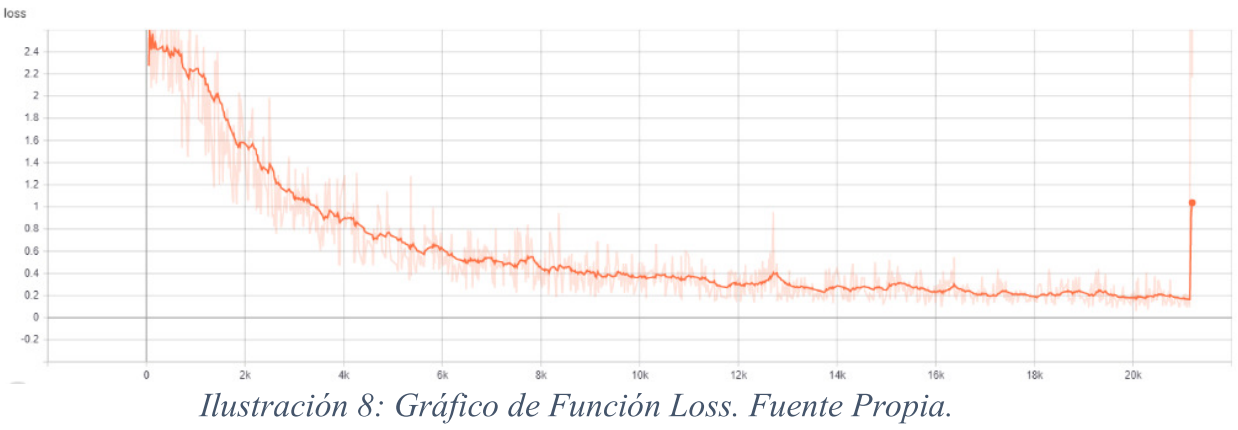


Ilustración 8: Gráfico de Función Loss. Fuente Propia.

EVALUACIÓN DE GENERACIÓN A PARTIR DE DATASET 1

Tabla 4: Evaluación de Generación a partir de dataset 1.

TEMPERATURA	CSP por Pista	C.S Promedio	D.E por Pista	D.E Promedio	S.C por Pista	S.C Promedio	P	R	F1
0.25	[49, 75, 54, 75, 49, 75, 75, 72, 60, 53]	63%	[107, 123, 109, 123, 107, 123, 123, 135, 121, 121]	119.2	[0.866, 0.866, 0.75, 0.866, 0.866, 0.866, 0.866, 0.75, 0.75, 0.75]	0.819	0.591	0.647	0.617
0.5	[53, 31, 45, 49, 49, 51, 49, 47, 31, 52]	51%	[134, 126, 126, 128, 128, 126, 145, 126, 128, 133]	130.0	[0.667, 0.667, 0.667, 0.667, 0.667, 0.667, 0.577, 0.667, 0.775, 0.866]	0.688	0.77	0.32	0.44
1	[77, 49, 49, 49, 46, 71, 61, 39, 75, 40]	55%	[88, 109, 107, 119, 151, 134, 133, 138, 123, 139]	124.1	[0.671, 0.866, 0.866, 0.671, 0.671, 0.75, 0.567, 0.75, 0.866, 0.53]	0.720	0.53	0.58	0.55
1.5	[44, 65, 73, 50, 20, 38, 55, 72, 46, 41]	50%	[136, 148, 132, 121, 131, 144, 118, 124, 121, 140]	131.5	[0.612, 0.567, 0.75, 0.567, 0.707, 0.671, 0.671, 0.671, 0.612, 0.612]	0.644	0.46	0.53	0.51
2	[48, 43, 54, 72, 64, 55, 45, 63, 52, 42]	53%	[152, 129, 145, 119, 136, 238, 117, 127, 198, 118]	147.9	[0.5, 0.567, 0.567, 0.612, 0.612, 0.5, 0.612, 0.612, 0.5, 0.612]	0.569	0.47	0.59	0.53
3	[48, 55, 48, 53, 47, 50, 46, 52, 44, 46]	48%	[208, 164, 244, 216, 212, 290, 242, 230, 239, 244]	228.9	[0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]	0.5	0.31	0.53	0.39
4	[58, 52, 38, 48, 40, 46, 39, 36, 43, 40]	44%	[343, 405, 357, 404, 351, 376, 357, 287, 344, 410]	363.4	[0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]	0.5	0.23	0.55	0.33
5	[46, 43, 44, 36, 39, 44, 46, 58, 39, 43]	43%	[531, 433, 440, 447, 429, 349, 429, 512, 512, 479]	456.1	[0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]	0.5	0.20	0.55	0.29

EVALUACIÓN DE GENERACIÓN A PARTIR DE DATASET 2

Tabla 5: Evaluación de generación a partir de dataset 2.

TEMPERATURA	CS por Pista	C.S Promedio	D.E por Pista	D.E Promedio	S.C por Pista	S.C Promedio	P	R	F1
0.25	[92, 92, 92, 92, 92, 92, 87, 92, 92, 92]	91%	[80, 80, 80, 80, 80, 80, 88, 80, 80, 80]	80.8	[0.866, 0.866, 0.866, 0.866, 0.866, 0.866, 0.866, 0.866, 0.866, 0.866]	0.86	1	0.89	0.94
0.5	[92, 84, 99, 84, 92, 99, 92, 92, 92, 95]	92%	[80, 94, 66, 94, 80, 81, 80, 80, 80, 74]	80.9	[0.866, 0.866, 0.866, 0.866, 0.866, 0.866, 0.866, 0.866, 0.866, 0.866]	0.86	0.99	0.90	0.94
1	[60, 87, 84, 84, 97, 83, 93, 99, 87, 92]	86%	[90, 86, 80, 94, 93, 106, 94, 68, 107, 80]	89.8	[0.707, 0.775, 0.775, 0.866, 0.75, 0.75, 0.866, 1.0, 0.567, 0.866]	0.79	0.88	0.86	0.87
1.5	[86, 86, 96, 94, 95, 79, 97, 87, 92, 48]	86%	[122, 112, 72, 112, 81, 72, 64, 94, 91, 119]	93.9	[0.567, 0.866, 0.866, 0.866, 0.671, 0.866, 0.866, 0.866, 0.866, 0.75]	0.804	0.82	0.857	0.837
2	[58, 60, 66, 73, 56, 54, 51, 65, 58, 72]	61%	[858, 786, 903, 845, 853, 825, 876, 935, 790, 858]	852.9	[0.542, 0.536, 0.541, 0.530, 0.528, 0.541, 0.536, 0.543, 0.523, 0.530]	0.534	0.168	0.731	0.273
3	[42, 57, 56, 57, 56, 56, 56, 59, 53, 65]	55%	[868, 907, 958, 885, 949, 880, 967, 889, 858, 976]	913.7	[0.509, 0.512, 0.511, 0.511, 0.510, 0.512, 0.511, 0.509, 0.510, 0.512]	0.51	0.149	0.684	0.244
4	[56, 54, 48, 50, 46, 50, 50, 59, 61, 52]	52%	[887, 924, 911, 942, 945, 894, 942, 903, 960, 932]	924.0	[0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]	0.5	0.149	0.676	0.244
5	[52, 57, 65, 46, 51, 61, 72, 49, 50, 56]	55%	[886, 923, 910, 941, 944, 894, 942, 903, 960, 932]	931.0	[0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]	0.5	0.15	0.679	0.253

Los siguientes gráficos muestran los resultados del experimento, que son analizados a continuación.

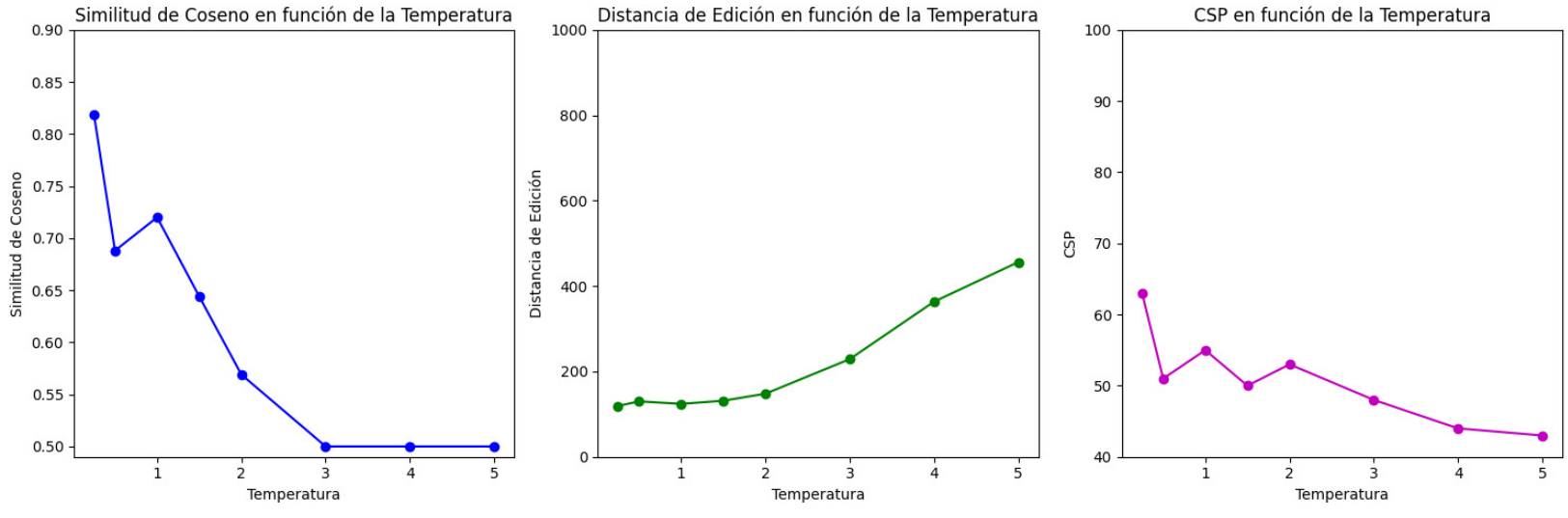


Ilustración 9: Resultados de métricas para dataset 1.

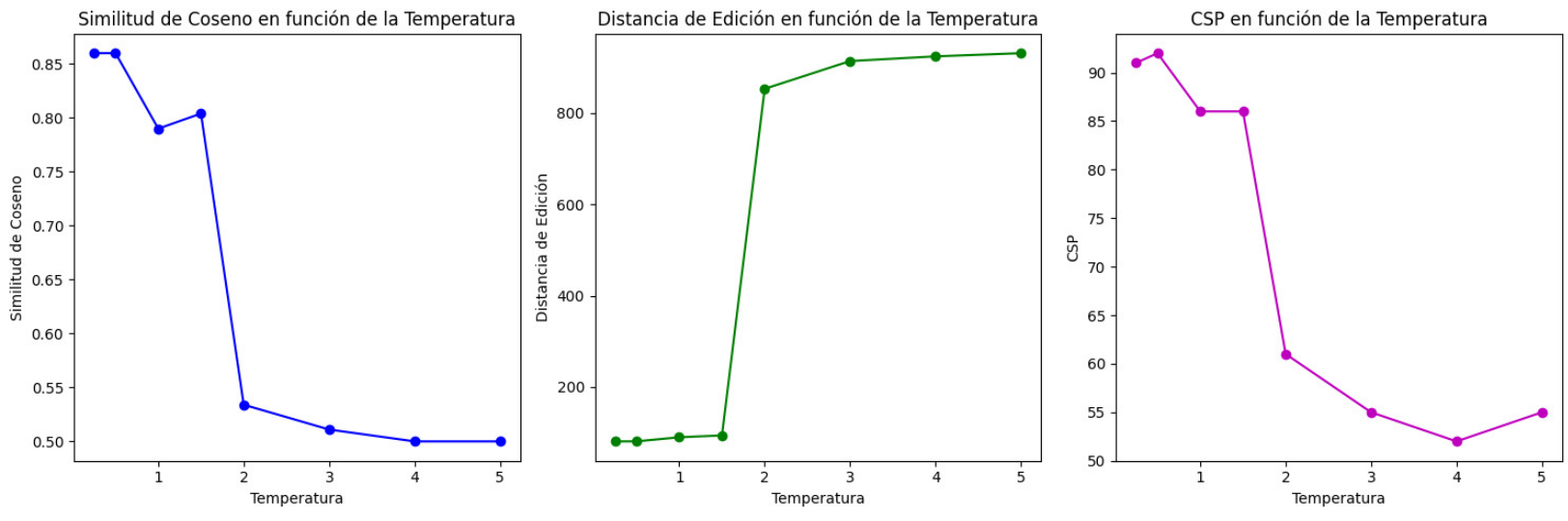


Ilustración 10: Resultados de métricas para dataset 2.

Los valores siguientes muestran el índice de correlación entre las variables para el primer y segundo conjunto, respectivamente.

	Similitud de Coseno	Distancia de Edición	CSP	F1
Similitud de Coseno	1.000	-0.748314	0.897346	0.84441817
Distancia de Edición	-0.74831	1.000	-0.791673	-0.8899395
CSP	0.897346	-0.791673	1.000	0.94068376
F1	0.84441817	-0.8899395	0.94068376	1

Tabla 6: Índices de correlación para métricas del conjunto de entrenamiento 1.

	Similitud de Coseno	Distancia de Edición	CSP	F1
Similitud de Coseno	1.000	-0.9912	0.9959	0.99702
Distancia de Edición	-0.9912	1.000	-0.98993	-0.99556687
CSP	0.9959	-0.98993	1.000	0.99279655
F1	0.99702	-0.99556687	0.99279655	1

Tabla 7: Índices de correlación para métricas del conjunto de entrenamiento 2.

De los índices de correlación se puede obtener la siguiente información:

1) Similitud de Coseno respecto a:
    a. Distancia de Edición: para el primer conjunto, mantiene una correlación negativa moderada, mientras que después de incrementar el dataset un 20%, esta se ajusta a una correlación negativa casi perfecta.
    b. CSP: con el CSP mantiene una correlación positiva fuerte, mientras que al aumentar el dataset un 20%, esta se transforma en una correlación positiva casi perfecta.
2) Distancia de Edición respecto a:
    a. Similitud de coseno: para el primer conjunto, mantiene una correlación negativa moderada, mientras que después de incrementar el dataset un 20%, esta se ajusta a una correlación negativa casi perfecta.
    b. CSP: mantiene una correlación negativa moderada, pero luego del incremento, esta asciende a una correlación negativa casi perfecta.
3) CSP:
    a. Similitud de Coseno: con la similitud de coseno se mantiene una correlación positiva fuerte, mientras que al aumentar el dataset un 20%, esta se transforma en una correlación positiva casi perfecta.
    b. Distancia de Edición: mantiene una correlación negativa moderada, pero luego del incremento, esta asciende a una correlación negativa casi perfecta.
4) F1-Score:
    a. El F1-Score mantiene una relación moderada frente a las demás métricas en el pimer dataset, pero luego, en el segundo, vemos cómo esta aumenta casi a la perfección, siendo una correlación negativa respecto a la distancia de edición y positiva frente a las demás.

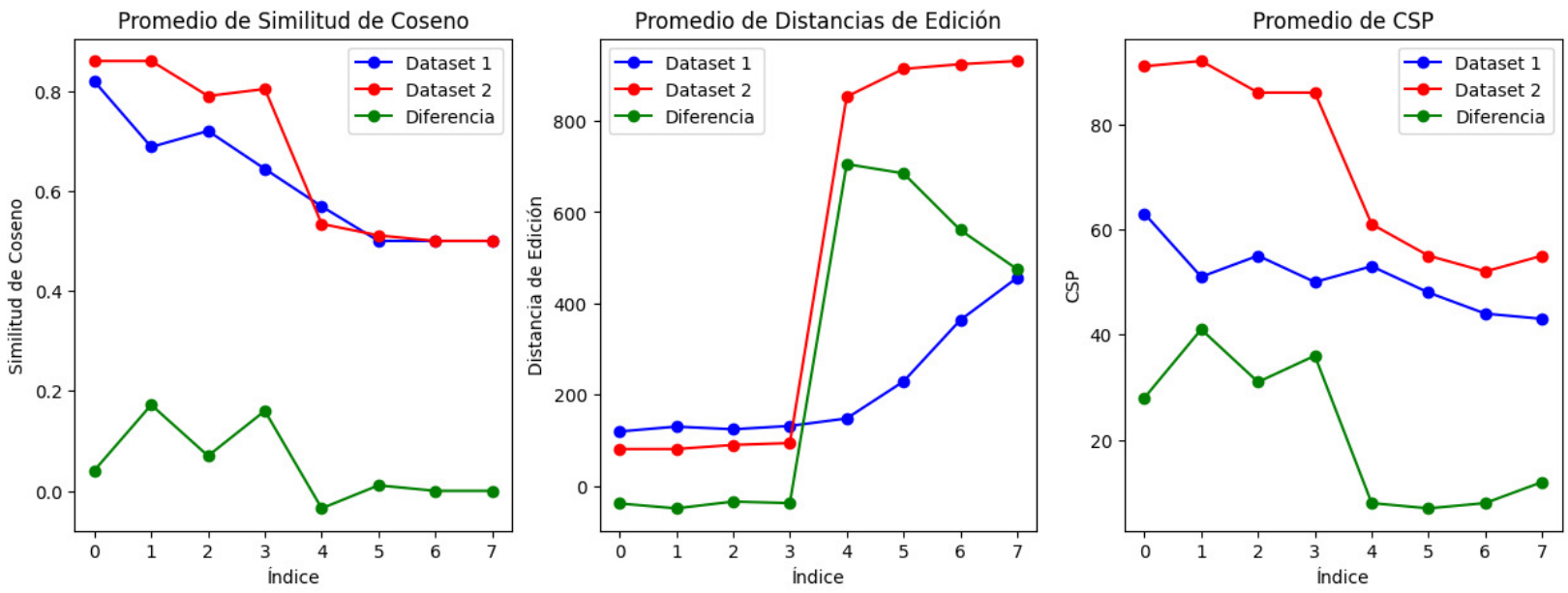


Ilustración 11: Comparación de métricas.

La línea verde muestra el valor absoluto de la diferencia.

Al analizar los resultados de los gráficos, podemos observar lo siguiente:

- Gráfico 1 - Promedio de Similitud de Coseno:
    - Ambos datasets (Dataset 1 y Dataset 2) muestran una variación en la similitud de coseno a medida que aumenta el índice.
    - En general, el Dataset 1 tiene valores de similitud de coseno más bajos en comparación con el Dataset 2.
    - Se observa un cambio en la relación con la temperatura cuando esta supera el valor de 2 de una manera más abrupta
- Gráfico 2 - Promedio de Distancias de Edición:
    - Al igual que en el gráfico anterior, podemos observar una tendencia de aumento en las distancias de edición a medida que aumenta el índice.
    - El Dataset 2 muestra valores mucho más altos de distancias de edición en comparación con el Dataset 1, esto se debe a la mayor complejidad de las pistas debido al enriquecimiento del dataset.
    - La línea que representa la diferencia entre los datasets muestra cómo varía la discrepancia en las distancias de edición entre ambos conjuntos de datos.

- Gráfico 3 - Promedio de CSP:

    - En este caso, podemos observar que tanto el Dataset 1 como el Dataset 2 muestran una variación en los valores de CSP a medida que aumenta el índice.
    - Se observa un aumento importante en el valor del CSP superior a 80% de similaridad antes del valor de temperatura 2.

De los casos anteriores, se puede observar el comportamiento de las métricas en función de las pistas generadas. Las pistas generadas del primer dataset, entregaban una menor cantidad de elementos al ser generadas a altas temperaturas, mientras que, con el enriquecimiento, esta complejidad aumentó, lo que indica un mayor valor de distancia de edición y un menor valor de similitud de coseno, pero un leve incremento en CSP. Eso es porque CSP cuenta los elementos restantes de la pista, pero aún así, restan casi los mismos elementos que en el dataset 1. Mientras tanto, para valores de temperatura menores a 1, se aprecia una mejora en las tres métricas. De los gráficos se observa que no se puede llegar a una conclusión analizando las métricas por separado, es por eso que tienen que ser analizadas en conjunto.

a. Las gráficas muestran valores inferiores y variaciones más suaves respecto a la del dataset 2, y también valores relativamente altos, pero no es suficiente para determinar si las pistas cumplen con el criterio analizándolas por separado, pues el CSP tiene valores aproximados del 60% cuando la similitud de coseno tiene valores del 80% aproximadamente, y un valor del 60% del CSP indica inmediatamente que la pista no cumple con el patrón. El alto valor de la similitud del coseno se puede deber a la similaridad de elementos y el reflejo de pistas similares en instrumentos. Y el bajo CSP se debe a que hay ciertos eventos importantes que no están ocurriendo. La similitud de coseno también cuenta lo que tienen en común, mientras que el CSP cuenta lo que le falta. Estos dos valores tampoco difieren mucho dentro de todo el rango de temperatura, es por esto que tienen una correlación positiva fuerte.

b. Para el caso del dataset 2, se puede observar un aumento en sus valores a la par del CSP y a la par de una disminución de la Distancia de Edición. Esto antes de una caída abrupta en la temperatura 2, que para todos los casos, esto se puede explicar que el valor de temperatura 2 puede ser un valor alto que ya comienza a trabajar con la complejidad de la red y genera resultados muy aleatorios y por ende, más complejos. Por esta razón los valores se estabilizan alejados de la referencia.
c. Bajo ese contexto, se puede analizar cómo la distancia de edición ilustra la cantidad de elementos que hay al generar pistas más complejas. Pues, a mayor distancia de edición, mayores son los cambios que hay que realizar
d. El CSP y la distancia de edición trabajan muy bien en conjunto, puesto que, para valores donde las pistas no cubren con la totalidad de eventos, el CSP entrega información porcentual sobre la similitud con la referencia, pero existen casos específicos donde puede cumplir con estos eventos, pero aún así tener una complejidad que la aleja mucho de lo que puede considerarse house, es ahí cuando es importante evaluar la distancia de edición, que indicará un mayor número si la pista es demasiado compleja, lo que entregará información para determinar si la pista cumple o no, puesto que individualmente es posible encontrar valores altos de CSP superior a temperatura 2.

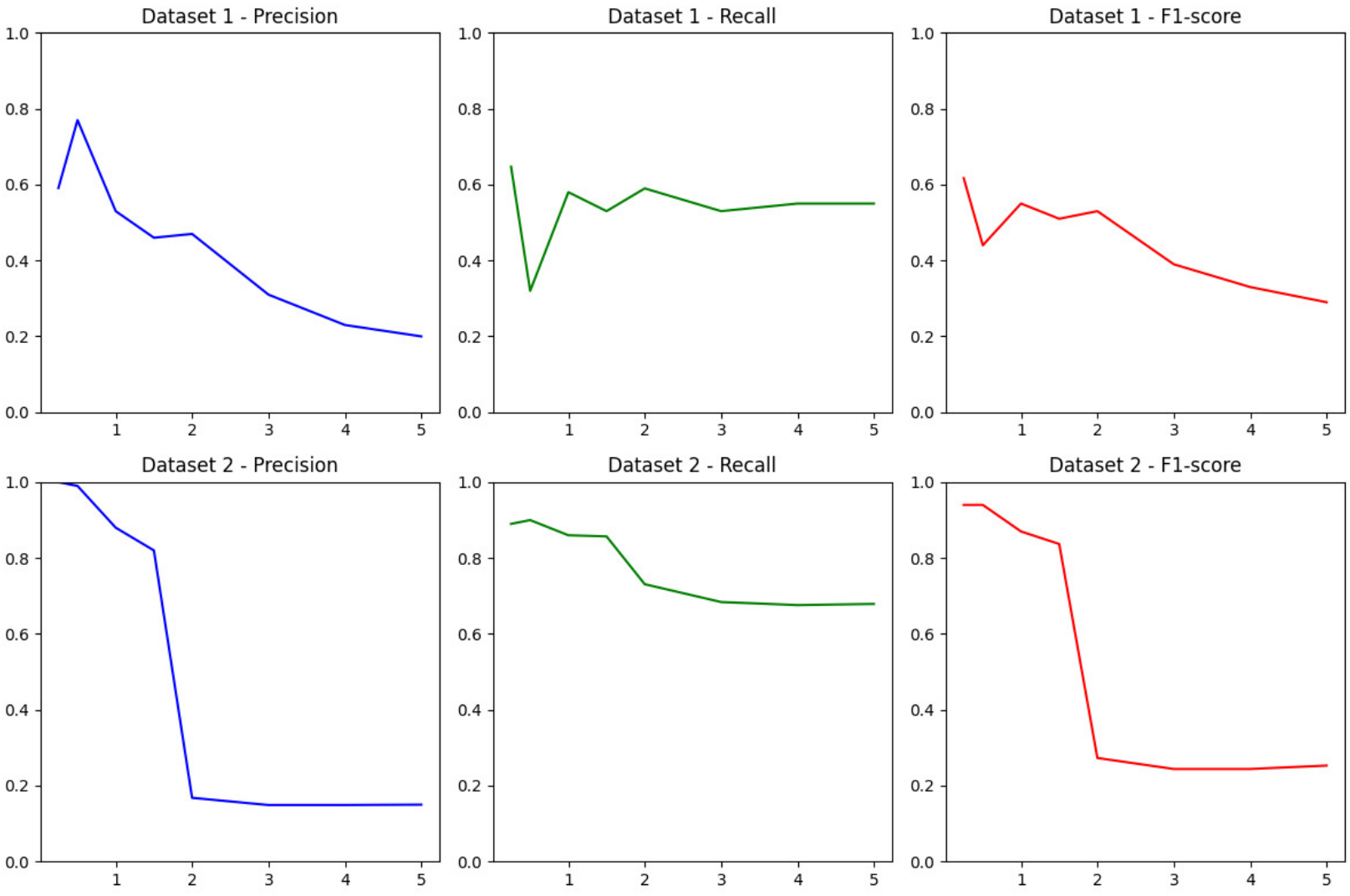


Ilustración 12: Gráficos de métricas de precisión, recall y F1 Score.

Precisión en el conjunto de datos 1:

Antes de la temperatura 2, la precisión disminuye a medida que aumenta la temperatura. Esto podría indicar que el modelo produce falsos positivos más altos a temperaturas más altas, lo que significa que elementos que no están presentes en la pista de referencia del género house en el Dataset 1.

Después de la temperatura 2, la precisión persiste en disminuir, lo que indica que el modelo continúa produciendo falsos positivos a temperaturas más altas.

Recall en el conjunto de datos 1:

Antes de la temperatura 2, el recall no muestra una tendencia evidente a medida que cambia la temperatura. El recall varía a diferentes temperaturas.

Después de la temperatura 2, el recall tampoco muestra una tendencia clara a medida que la temperatura cambia. El recall sigue siendo inconstante en todos los niveles de temperatura.

Puntaje F1 en el conjunto de datos 1:

Antes de la temperatura 2, el puntaje F1 muestra una tendencia a la baja a medida que aumenta la temperatura, lo que indica una disminución en el equilibrio entre precisión y recuerdo.
Después de la temperatura 2, el puntaje F1 continúa disminuyendo, lo que indica que el equilibrio entre precisión y recuerdo sigue disminuyendo a temperaturas más altas.

Precisión en el conjunto de datos 2:

Antes de la temperatura 2, la precisión se mantiene constante y alta. Esto demuestra que el modelo crea señales que cumplen con el género house con gran precisión.
Después de la temperatura 2, la precisión es muy baja. Esto podría indicar que el modelo comienza a generar más falsos positivos a temperaturas más altas, lo que significa que genera elementos que no están en la pista de referencia del género house.

Recall en el conjunto de datos 2:

Antes de la temperatura 2, el Recall se mantiene en un nivel elevado y constante. Esto indica que el modelo incorpora la mayor parte de las características que se encuentran en la pista de referencia del género house.
Después de la temperatura 2, el recuerdo disminuye un poco, pero sigue teniendo niveles bastante altos. Esto puede indicar que el modelo tiene dificultades para capturar todos los elementos pertinentes del género house a temperaturas más altas.
score F1 en Dataset 2

Antes de la temperatura 2, el puntaje F1 se mantiene estable y alto. Esto demuestra un buen equilibrio entre la precisión y el recall en la generación de pistas de la casa.

Después de la temperatura 2, el puntaje F1 disminuye, lo que indica que el equilibrio entre la precisión y el recuerdo sufre un impacto negativo a temperaturas más altas.
En resumen, se observa un rendimiento consistente y sólido en la generación de pistas del género en el Dataset 2 antes de la temperatura 2 con una alta precisión, recuerdo y puntaje F1.

Sin embargo, después de la temperatura 2, la precisión disminuye, lo que conduce a la generación de más falsos positivos, y tanto la puntuación de recuerdo como la puntuación F1 disminuyen ligeramente.

Estas observaciones destacan la importancia del rango de temperatura para el rendimiento del modelo e indican que las temperaturas más altas pueden tener un impacto negativo en la generación de pistas del género house.

Comparación de datasets

Se realiza la comparación entre el Dataset 2 y el Dataset 1 y se analiza cómo mejoró el desempeño en cada métrica en el rango de temperatura de 0 a 2:

En el rango de temperatura de 0 a 2:

Precisión:
Dataset 2: ((1 - 0.591) / 0.591) * 100 = 69.55% de mejora respecto al Dataset 1

Recall:
Dataset 2: ((0.89 - 0.647) / 0.647) * 100 = 37.53% de mejora respecto al Dataset 1

F1-score:
Dataset 2: ((0.94 - 0.617) / 0.617) * 100 = 52.39% de mejora respecto al Dataset 1

En el rango de temperatura de 0 a 2, el Dataset 2 muestra una mejora significativa en comparación con el Dataset 1 en todas las métricas de evaluación. La precisión mejora en un 69.55%, el recall en un 37.53% y el F1-score en un 52.39%.

Estos resultados indican que al enriquecer el Dataset 2 con un 20% de género house, se logra un desempeño mejorado en comparación con el Dataset 1 en el rango de temperatura de 0 a 2. El modelo generado con el Dataset 2 muestra una mayor precisión, recall y F1-score, lo que sugiere una mejor capacidad para generar pistas del género house con mayor coherencia y calidad en ese rango de temperatura.

En el rango de temperatura de 2 a 5, se observa un empeoramiento en las métricas de evaluación en comparación con el rango de temperatura de 0 a 2 en ambos datasets. A medida que aumenta la complejidad del dataset, como la inclusión de género house, también se introducen más variaciones y posibilidades en la generación de contenido. A temperaturas más altas, el modelo se vuelve más propenso a explorar estas variaciones y generar resultados más aleatorios y menos coherentes. Esto se debe a que la temperatura afecta la distribución de probabilidad en la generación de muestras, permitiendo una mayor diversidad y exploración de opciones menos probables.

Comparando el desempeño en el rango de temperatura de 2 a 5 entre el Dataset 2 y el Dataset 1, podemos analizar el cambio porcentual en cada métrica:

En el rango de temperatura de 2 a 5:

Precisión:
Dataset 2: ((0.15 - 1) / 1) * 100 = -85% de disminución respecto al rango de temperatura de 0 a 2

Recall:
Dataset 2: ((0.679 - 0.89) / 0.89) * 100 = -23.71% de disminución respecto al rango de temperatura de 0 a 2

F1-score:
Dataset 2: ((0.253 - 0.94) / 0.94) * 100 = -73.19% de disminución respecto al rango de temperatura de 0 a 2

En el rango de temperatura de 2 a 5, el Dataset 2 muestra una disminución significativa en comparación con el rango de temperatura de 0 a 2 en todas las métricas de evaluación. La precisión disminuye en un 85%, el recall en un 23.71% y el F1-score en un 73.19%.

Estos resultados indican que al aumentar la temperatura en el rango de 2 a 5, se produce un deterioro en el desempeño del modelo generado con el Dataset 2. Los valores aleatorios generados al aumentar la temperatura introducen más incertidumbre y variabilidad, lo que afecta negativamente la precisión, el recall y el F1-score.

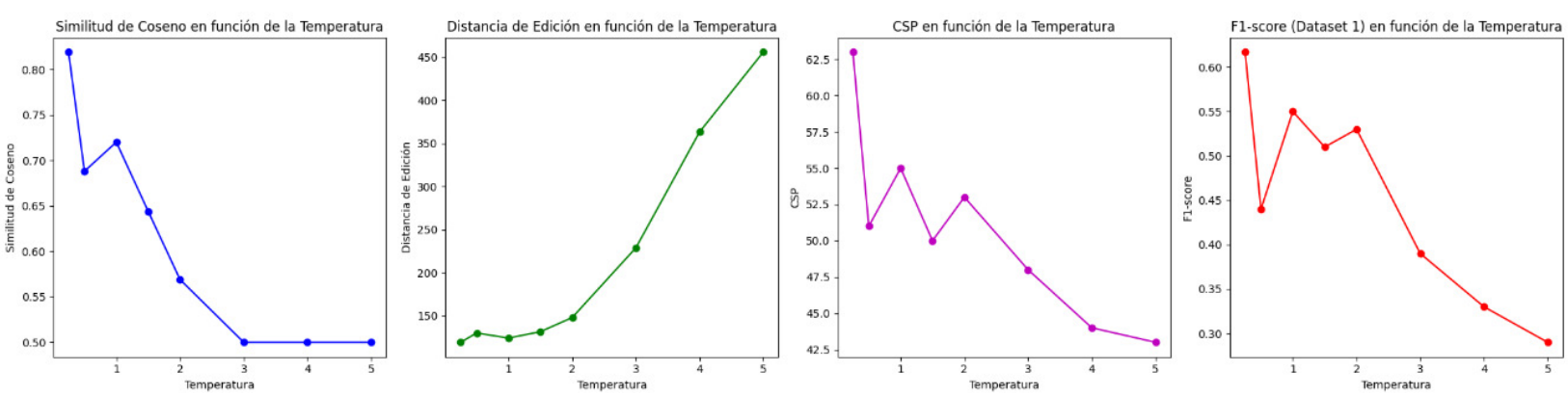


Ilustración 13: Gráficas de todas las métricas de dataset 1.

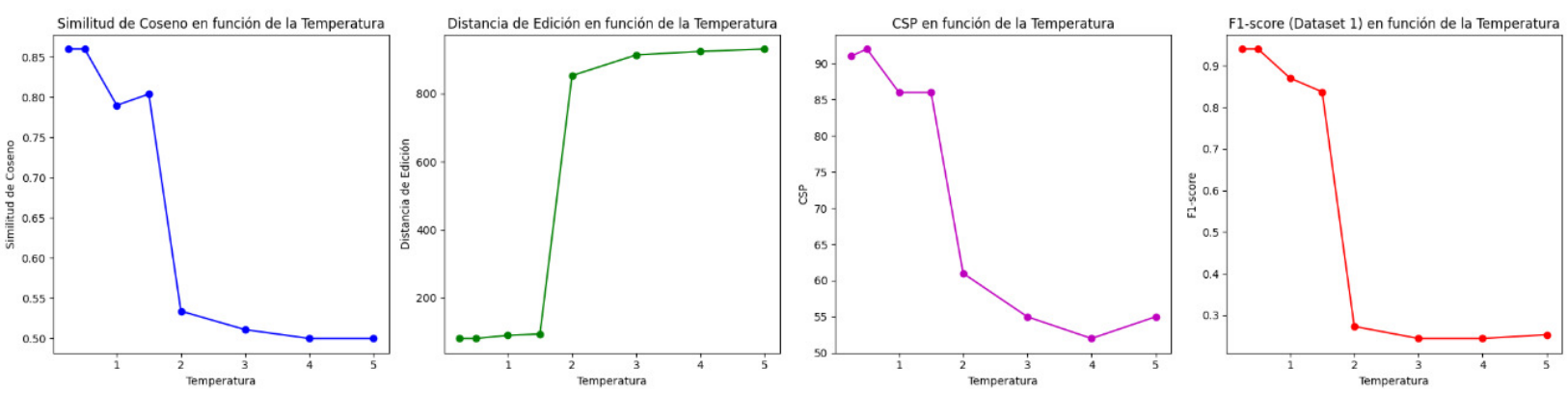


Ilustración 14: Gráficos de todas las métricas del dataset2.

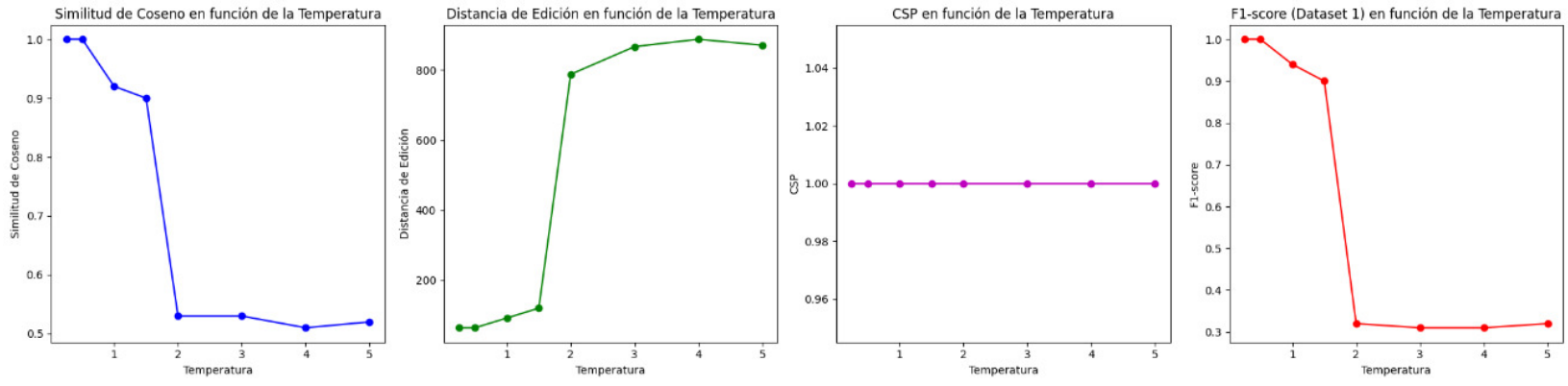


Ilustración 15: Resultados de Métricas para pistas corregidas.

En el gráfico para las pistas corregidas, se puede notar una diferencia en la métrica CSP, ya que es la métrica que utiliza el algoritmo para corregir, rellenando los falsos negativos que la pista no posee, y se puede ver una ligera mejora en las demás métrias, como el alcance del 100% para el F1 y el SC a valores de temperatura bajos, pero luego los valores disminuyen casi a los mismos valores de las pistas sin corregir, pues aumentan los falsos positivos. La justificación de la utilización de corrección de falsos negativos radica en que es obligatorio que el patrón esté en la pista, pero no se puede indicar como una corrección eventos que no están en la pista, porque son los que entregan diversidad a esta.

Capítulo 7

Conclusiones y Trabajos Futuros

7.1 Conclusiones

En este trabajo se realizó un experimento con el Framework de Google Magenta, utilizado la red DRUMS_RNN con el fin de implementar un sistema de generación de música “House” que cumpla con el criterio “Four on the Floor”. El desempeño de estos resultados fue medido en base a un índice de correlación entre las pistas generadas y una pista de referencia mediante un algoritmo de Lenguaje Python. Se logró implementar un sistema que genera pistas House, el cual corrige las pistas con errores y las adapta para que cumplan con el género, reteniendo la información de modificación para luego ser evaluadas.

Una vez obtenidos los resultados de evaluación, se observó un aumento en el desempeño para el dataset que fue enriquecido, como se observa en los gráficos de comparación, concluyendo que, a Temperatura 0, el enriquecimiento en un 20% del dataset, permite una mejora en el desempeño de 34% según el CSP, del 11% según la similitud de cosenos, y un 39% la Distancia de Edición para una temperatura inferior a 2. Es importante señalar que se deben analizar las métricas en conjunto para determinar si una pista cumple o no, porque estas miden distintos aspectos de la comparación de secuencias.

También, se observa que los mejores resultados ocurren para valores de Temperatura cercanos a 0, pues este valor causa que las pistas generadas varíen un mínimo en relación a la entrada, siendo cerca de un 91% igual a esta, la cual era un patrón “Four on the Floor”, pues mientras más cercano a cero, más determinística es la respuesta.

A medida que aumenta la temperatura, aumenta la cantidad de pistas con errores.

Luego de un valor de Temperatura = 2, los resultados disminuyen su desempeño drásticamente y se mantienen, por lo que el rango recomendable para no interferir de manera significativa el criterio es

$$0 < Temperatura < 2$$

Dentro de este rango, es posible observar que se cumple el objetivo de similitud del 80% por parte de todas las métricas, con un promedio total de 85% por parte de las métricas porcentuales.

El enriquecimiento del dataset también aumentó el índice de correlación entre las métricas, generando una mejor relación entre ellas y como consecuencia, establecer un análisis más certero de las pistas obtenidas.

Un enriquecimiento en el dataset muestra que a temperaturas superiores a 2, la red puede generar mejores resultados, a pesar de aumentar considerablemente su cantidad de elementos debido al aumento en la distancia de edición.

El enfoque one-hot asigna un valor binario (1 o 0) a cada elemento de la lista, indicando si está presente o no en la lista. Si una lista contiene más elementos en general y también incluye elementos adicionales en comparación con la otra lista, es posible que la similitud de coseno resultante sea más cercana a 0.5.

Esto se debe a que la similitud de coseno se calcula como el coseno del ángulo entre los vectores one-hot. Si los vectores tienen longitudes similares y están orientados en direcciones aproximadamente perpendiculares (ángulo cercano a 90 grados), el coseno del ángulo será cercano a 0, lo que se traduce en una similitud de coseno de alrededor de 0.5.
El análisis de las métricas (precisión, recall y F1-score) en relación con las temperaturas y los datasets proporciona información valiosa sobre el rendimiento de los modelos generativos en la generación de pistas de género house.

Al comparar los dos datasets, se observa que el Dataset 2, enriquecido con un 20% de género house, muestra una mejora significativa en términos de precisión, recall y F1-score en comparación con el Dataset 1. Esto indica que la inclusión de más ejemplos de género house en el dataset puede tener un impacto positivo en el rendimiento del modelo generativo en la generación de pistas de este género específico.

En general, al considerar 6 métricas, se puede obtener una visión más holística del rendimiento del modelo generativo en la generación de pistas de género house. Estas métricas adicionales pueden ayudar a evaluar la capacidad del modelo para generar contenido musical que sea similar en términos de estructura, características y estilo al género objetivo.

Con todo lo anterior, se puede concluir también que se obtuvo una mayor diversidad, ayudando a capturar una gama más amplia y variaciones y relaciones, lo que a su vez mejora la precisión de las métricas utilizadas. También se obtiene mayor representatividad, mitigando sesgos o desequilibrios en el dataset original, y como resultado, una mejora en la calidad de los resultados de las métricas utilizadas. Finalmente, se tiene una mejor generalización, pues un mayor número de ejemplos puede permitir una mejor captura de características relevantes y una mejor capacidad para generalizar a nuevos casos o datos no vistos anteriormente.

7.2 Trabajos Futuros

Respecto al trabajo futuros se puede trabajar con distintas redes neuronales y comparar entre ellas cuál entrega el mejor desempeño utilizando el mismo algoritmo diseñado en este trabajo.

También, el algoritmo diseñado puede ser configurado para analizar otros géneros en base a otras partituras de referencia y, con esto, poder tener una mayor visión de los resultados de una red neuronal que enmarque distintos estilos musicales.

Es importante también mencionar que el dataset puede ser aumentado para obtener un mejor entrenamiento, y así poder evaluar con mayor precisión distintos datasets para luego compararlos una vez la red genere las pistas.

El análisis puede ser realizado no solamente con el formato MIDI, sino que también se pueden probar distintos formatos que se acomoden mejor a otros estilos, en este caso particular el estudio se hizo en base a ritmos House, pues los productores de este

género principalmente utilizan archivos MIDI para la producción de sus obras, pero hay géneros, o no solamente puede ser música, que se puede lograr un mejor análisis utilizando, por ejemplo, archivos de audio, archivos XML o notación ABC, dependiendo cuáles se tengan al alcance del usuario y cuáles sean sus necesidades.

También, se puede realizar el análisis del tiempo de demora de la generación de ciertos patrones, y cómo estas redes se comportan en tiempo en base a los distintos formatos que se pueden utilizar para expresar la música a través de un computador.

Referencias

[1] Rhodes, M. (1961). An Analysis of Creativity. The Phi Delta Kappan, 42(7), 305–310. http://www.jstor.org/stable/20342603

[2] Getzels, J. W., & Jackson, P. W. (1962). *Creativity and intelligence: Explorationswith gifted students.* Wiley.

[3] Amabile, T. M. (1983). The social psychology of creativity: A componential conceptualization. *Journal of Personality and Social Psychology, 45*(2), 357– 376. https://doi.org/10.1037/0022-3514.45.2.357

[4] Colton S., Wiggins G. A.(2012). Computational creativity: The final frontier?. Ecai.

[5] Pearce, C. L., & Sims, H. P., Jr. (2002). Vertical versus shared leadership as predictors of the effectiveness of change management teams: An examination of aversive, directive, transactional, transformational, and empowering leader behaviors. *Group Dynamics: Theory, Research, and Practice, 6*(2), 172– 197. https://doi.org/10.1037/1089-2699.6.2.172.

[6] B. S. C. Ranjan, L. Siddharth, and A. Chakrabarti, “A systematic approach to assessing novelty, requirement satisfaction, and creativity,” Artificial Intelligence for Engineering Design, Analysis and Manufacturing, vol. 32, no. 4, pp. 390—414, 2018.

[7] Carolyn Lamb, Daniel G. Brown, and Charles L. A. Clarke. 2018. Evaluating Computational Creativity: An Interdisciplinary Tutorial. ACM Comput. Surv. 51, 2, Article 28 (March 2019), 34 pages. https://doi.org/10.1145/3167476

[8] Modelling Creativity: Identifying Key Components through a Corpus-Based Approach

Jordanous A, Keller B (2016) Modelling Creativity: Identifying Key Components through a Corpus-Based Approach. PLOS ONE 11(10): e0162959. https://doi.org/10.1371/journal.pone.0162959

[9] RO. A.. Moog, "Voltage Controlled Electronic Music Modules," J. Audio Eng. Soc.,vol.

13, no. 3, pp. 200-206, (1965 July.). doi: http://www.aes.org/e-lib/browse.cfm?elib=1204

[10] Alpern, A. (1995). Techniques for algorithmic composition of music. On the web: http://hamp. hampshire. edu/adaF92/algocomp/algocomp, 95(1995), 120.

[11] Grout, D. J., & Palisca, C. (2001). A History of Western Music. 6'h ed. New York.

[12] Carretero Aguado, A. (2013). El proceso de composición musical a través de las técnicas bio-inspiradas de inteligencia artificial: investigación desde la creación musical (Doctoral dissertation, Tesis doctoral (dirigida por M. Martínez y V. Calvo). Universidad Rey Juan Carlos).

[13] Manning, C., & Schutze, H. (1999). Foundations of statistical natural language processing. MIT press.

[14] Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., & Robinson,

T. (2013). One billion word benchmark for measuring progress in statistical language modeling. arXiv preprint arXiv:1312.3005.

[15] MIDI Manufacturers Association (MMA). MIDI Specifications, Accessed on 14/04/2017. https://www.midi.org/specifications.

[16] Ludeña Trujillo, M. H., & Valarezo Campoverde, S. F. (1998). Sistema informático musical con protocolo MIDI (Bachelor's thesis, QUITO/EPN/1998).

[17] Carofilis, V., & Andrés, R. (2018). Generación automática de música a través detécnicas de machine learning (Bachelor's thesis, Quito: UCE).

[18] Dong, H. W., Hsiao, W. Y., Yang, L. C., & Yang, Y. H. (2018, April). Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In Proceedings of the AAAI Conference on ArtificialIntelligence (Vol. 32, No. 1).

[19] Sturm, B. L. (2012, October). A survey of evaluation in music genre recognition.In International Workshop on Adaptive Multimedia Retrieval (pp. 29-66). Springer, Cham.

[20] García Álvarez, P. J. (2018). Aplicación de redes neuronales en la predicción de mortalidad por neumonía. Revista Médica Electrónica, 40(5), 1361-1379.

[21] Briot, J. P., & Pachet, F. (2020). Deep learning for music generation: challengesand directions. Neural Computing and Applications, 32(4), 981-993.

[22] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533-536.

[23] Mozer, M. C. (1994). Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scale processing. Connection Science, 6(2-3), 247-280.

[24] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[25] Eck, D., & Schmidhuber, J. (2002). A first look at music composition using lstm recurrent neural networks. Istituto Dalle Molle Di Studi Sull IntelligenzaArtificiale, 103, 48.

[26] Sturm, B. (2018). How stuff works: LSTM model of folk music transcriptions.In Joint Workshop on Machine Learning for Music, ICML.

[27] Hadjeres, G., Pachet, F., & Nielsen, F. (2017, July). Deepbach: a steerable model for bach chorales generation. In International Conference on Machine Learning (pp. 1362-1371). PMLR.

[28] Patarroyo Devia, J. L. (2019). Raitmo: generador musical para apoyar la composición musical basado en machine learning.

[29] Turing, A. M. (1950). Mind. Mind, 59(236), 433-460.

[30] Searle, J. R. (1982). The Chinese room revisited. Behavioral and brain sciences, 5(2), 345-348.

[31] Tapia-Velázquez, J. L. (2021). Composición de Música Flamenca para Guitarra mediante Técnicas de Inteligencia Artificial (Master's thesis).

[32] Lemström, K., & Ukkonen, E. (2000, April). Including interval encoding into edit distance based music comparison and retrieval. In Proc. AISB (pp. 53-60).

[33] Thada, V., & Jaglan, V. (2013). Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic

algorithm. International Journal of Innovations in Engineering and Technology, 2(4), 202-205.

[34] Sheikh Fathollahi, M., Razzazi, F. Music similarity measurement and recommendation system using convolutional neural networks. Int J Multimed Info Retr 10, 43–53 (2021).

[35] Yang, L.-C., & Lerch, A. (2020). On the evaluation of generative models in music. Neural Computing and Applications.

[36] J. Nistal, S. Lattner, and G. Richard, “DRUMGAN: synthesis of drum sounds with timbral feature conditioning using generative adversarial networks,” in Procceedings of the 21th International Society for Music Information Retrieval Conference (ISMIR), Oct. 2020, pp. 590–597

Anexos

9.1 Código de Corregidor/Evaluador

En el link jlefenda/magenta (github.com), se encuentra el GitHub con código en Lenguaje Python 3.7 para el desarrollo del Corregidor y Evaluador utilizado para realizar los experimentos.

9.2 Resultados gráficos de Archivos modificados

En esta sección, se presenta una representación gráfica de los archivos modificados, para identificar los cambios realizados en la partitura desde una percepción visual y facilitar la comprensión. Los archivos fueron cargados al Software MuseScore y Ableton Live para su representación en Partitura como en Piano Roll. El archivo utilizado fue el Track 2 de Temperatura 1 del Dataset 1, debido a la cantidad de elementos que estos poseían para más claridad.

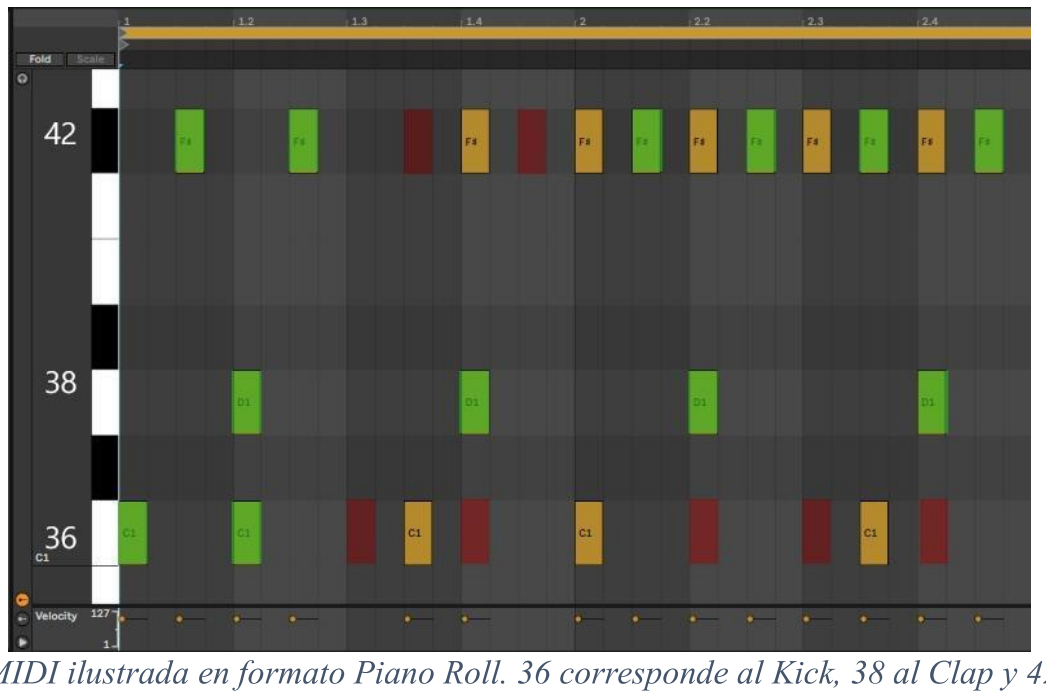


Ilustración 16: Pista MIDI ilustrada en formato Piano Roll. 36 corresponde al Kick, 38 al Clap y 42 al Hat. Fuente Propia.

En la presente imagen, se ilustra en color verde las notas que cumplen con el archivo de referencia, y el algoritmo los detecta como elementos presentes, mientras que los espacios

en rojo los detecta como los elementos que debe añadir para cumplir con el patrón “Four on the Flour” a cabalidad.

En la siguiente imagen, se muestra el archivo modificado por el algoritmo, añadiendo los elementos faltantes en los puntos rojos marcados.

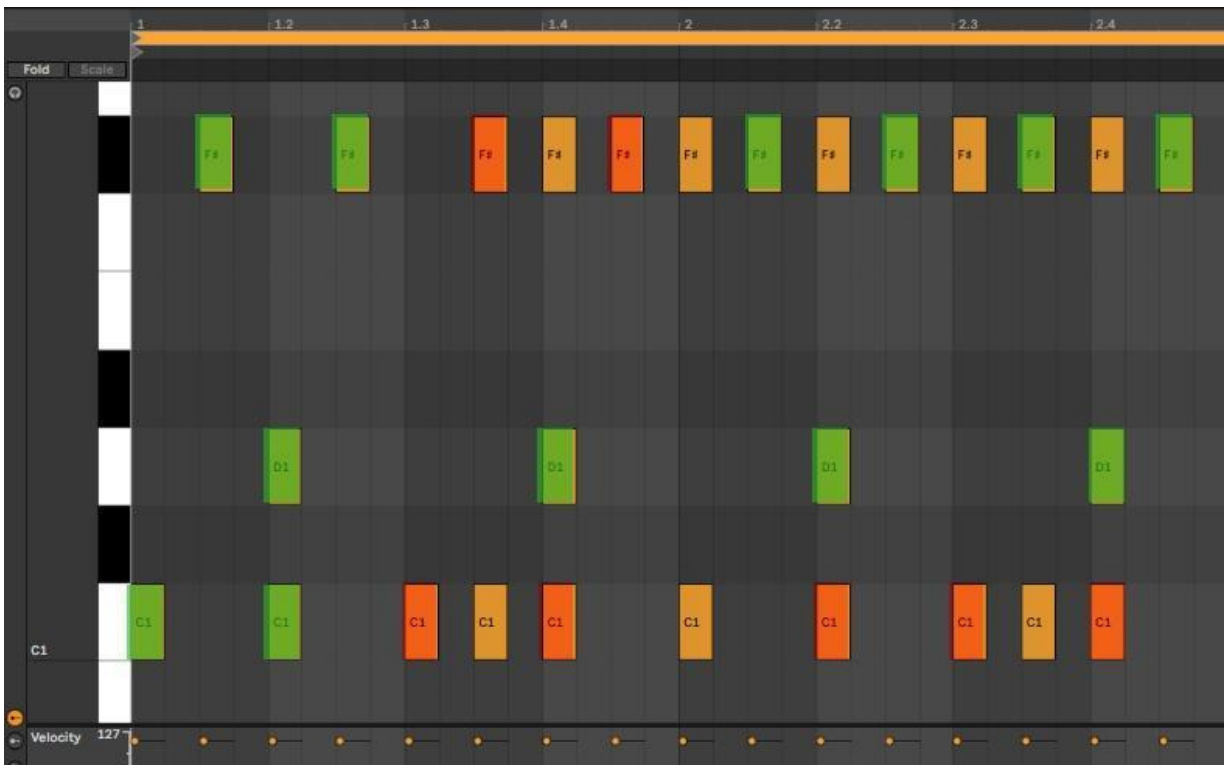


Ilustración 17: Pista MIDI modificada luego de pasar por el algoritmo. Fuente propia.

Análogamente, se obtienen los resultados en partitura para el archivo previo, y el modificado posteriormente:

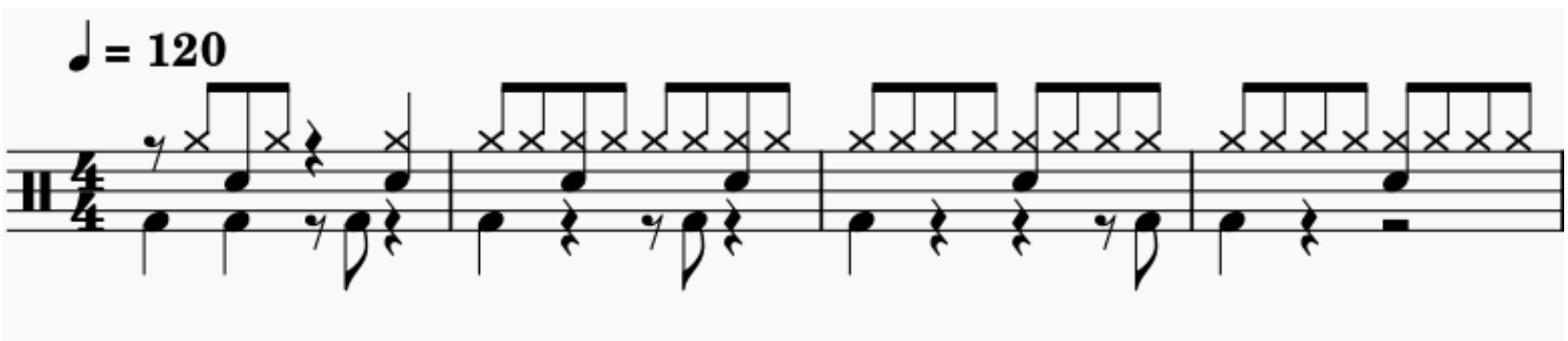


Ilustración 18: Partitura sin modificar. Fuente propia.

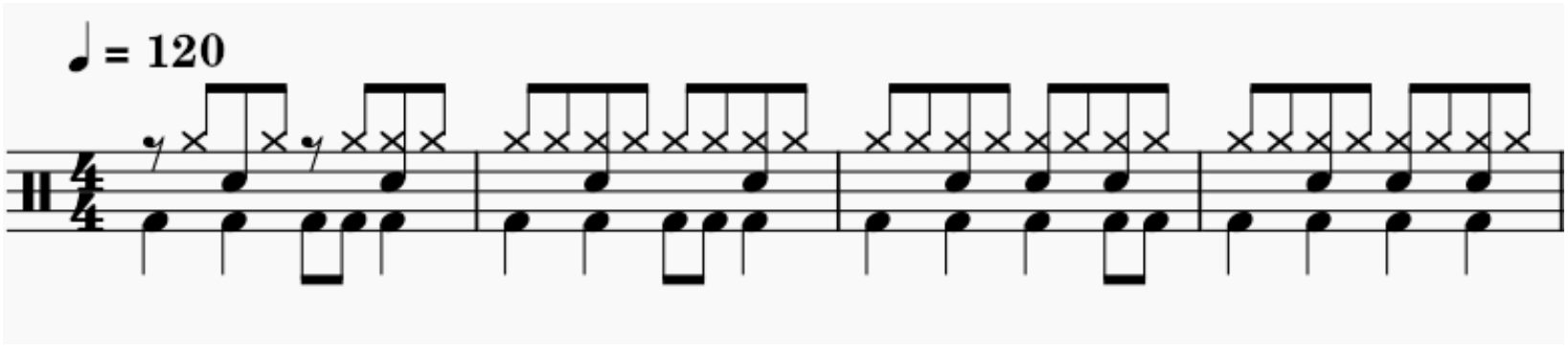


Ilustración 19: Partitura modificada. Fuente propia.

9.3 Ejemplo de archivo .XML

Se presenta el script de una partitura compuesta por una nota DO Medio (C3, o DO de tercera octava), del cual es posible observar la gran cantidad de metainformación que este sistema posee para definir las partituras.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE score-partwise PUBLIC "-//Recordare//DTD MusicXML 3.1
Partwise//EN" "http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise version="3.1">
  <identification>
    <encoding>
      <software>MuseScore 3.6.2</software>
      <encoding-date>2022-12-06</encoding-date>
      <supports element="accidental" type="yes"/>
      <supports element="beam" type="yes"/>
      <supports element="print" attribute="new-page" type="yes"
value="yes"/>
      <supports element="print" attribute="new-system" type="yes"
value="yes"/>
      <supports element="stem" type="yes"/>
      </encoding>
    </identification>
  <defaults>
    <scaling>
      <millimeters>6.99911</millimeters>
      <tenths>40</tenths>
      </scaling>
    <page-layout>
<page-height>1697.36</page-height>
<page-width>1200.15</page-width>
<page-margins type="even">
  <left-margin>85.7252</left-margin>
  <right-margin>85.7252</right-margin>
  <top-margin>85.7252</top-margin>
  <bottom-margin>85.7252</bottom-margin>
  </page-margins>
<page-margins type="odd">
  <left-margin>85.7252</left-margin>
  <right-margin>85.7252</right-margin>
  <top-margin>85.7252</top-margin>
  <bottom-margin>85.7252</bottom-margin>
  </page-margins>
</page-layout>
```

```
  <word-font font-family="Edwin" font-size="10"/>
  <lyric-font font-family="Edwin" font-size="10"/>
  </defaults>
<part-list>
  <score-part id="P1">
    <part-name>Piano</part-name>
    <part-abbreviation>Pno.</part-abbreviation>
    <score-instrument id="P1-I1">
      <instrument-name>Piano</instrument-name>
      </score-instrument>
    <midi-device id="P1-I1" port="1"></midi-device>
    <midi-instrument id="P1-I1">
      <midi-channel>1</midi-channel>
      <midi-program>1</midi-program>
      <volume>78.7402</volume>
      <pan>0</pan>
      </midi-instrument>
    </score-part>
  </part-list>
<part id="P1">
  <measure number="1" width="378.92">
    <print>
      <system-layout>
        <system-margins>
          <left-margin>50.00</left-margin>
          <right-margin>599.78</right-margin>
          </system-margins>
        <top-system-distance>70.00</top-system-distance>
        </system-layout>
      </print>
    <attributes>
      <divisions>1</divisions>
      <key>
        <fifths>0</fifths>
      </key>
    <time>
      <beats>4</beats>
      <beat-type>4</beat-type>
      </time>
    <clef>
      <sign>G</sign>
      <line>2</line>
      </clef>
    </attributes>
  <note default-x="84.22" default-y="-50.00" dynamics="111.11">
    <pitch>
      <step>C</step>
      <octave>4</octave>
      </pitch>
```

```
      <duration>1</duration>
      <voice>1</voice>
      <type>quarter</type>
      <stem>up</stem>
      <notations>
        <articulations>
          <staccato/>
          </articulations>
        </notations>
      </note>
    <note>
      <rest/>
      <duration>1</duration>
      <voice>1</voice>
      <type>quarter</type>
        </note>
      <note>
        <rest/>
        <duration>2</duration>
        <voice>1</voice>
        <type>half</type>
        </note>
      <barline location="right">
        <bar-style>light-heavy</bar-style>
        </barline>
      </measure>
    </part>
</score-partwise>
```