



**Universidad CENFOTEC**

Maestría en Ingeniería del Software con Énfasis en Inteligencia  
Artificial Aplicada

**Exploración de la técnicas de Inteligencia  
Artificial aplicadas a la composición musical.**

Santiago Mejía Osorio

## TRIBUNAL EXAMINADOR

Este proyecto fue aprobado por el Tribunal Examinador de la carrera: **Maestría Profesional en Ingeniería del Software con énfasis en Inteligencia Artificial Aplicada**, requisito para optar por el título de grado de **Maestría**, para el estudiante: **Mejía Osorio Santiago**.



Digitally signed by  
MIGUEL PEREZ  
MONTERO (FIRMA)  
Date: 2024.02.26  
20:48:37 -06'00'

*M.Sc. Miguel Pérez Montero*  
**Tutor**

**JORGE ARTURO  
GARNIER  
ROVIRA (FIRMA)**

Firmado digitalmente por  
JORGE ARTURO GARNIER  
ROVIRA (FIRMA)  
Fecha: 2024.02.26  
21:50:10 -06'00'

*MBA. Jorge Arturo Garnier Rovira*  
**Lector 1**

**CHRISTIAN  
JOSE SIBAJA  
FERNANDEZ  
(FIRMA)**

Firmado digitalmente  
por CHRISTIAN JOSE  
SIBAJA FERNANDEZ  
(FIRMA)  
Fecha: 2024.03.06  
07:28:31 -06'00'

*Ing. Christian Sibaja Fernández, PH.D.*  
**Lector 2**



San José, Costa Rica, 26 de febrero de 2024

---

# Exploración de la técnicas de Inteligencia Artificial aplicadas a la composición musical.

Ing. Santiago Mejía Osorio<sup>1</sup>, MSc. Miguel Pérez Montero<sup>2</sup>

<sup>1</sup> [smejiao@ucenfotec.ac.cr](mailto:smejiao@ucenfotec.ac.cr)

<sup>2</sup> [mperez@ucenfotec.ac.cr](mailto:mperez@ucenfotec.ac.cr)

**RESUMEN** Este artículo examina cómo las técnicas de Inteligencia Artificial (IA) se utilizan en la composición musical. Se busca comprender tanto sus capacidades como sus limitaciones a través de un análisis de las metodologías de IA utilizadas en este contexto y su evolución histórica. Además, se examinan ejemplos concretos de algoritmos subyacentes, brindando una comprensión completa de su uso en la composición musical moderna. Los objetivos particulares del estudio comprenden enumerar las diferentes técnicas de IA utilizadas, explicar los principios fundamentales que sustentan estas técnicas, analizar sus limitaciones para encontrar oportunidades de mejora y evaluar el impacto de la IA en la composición musical actual, contribuyendo así a la comprensión de este campo en constante evolución.

**ABSTRACT** This article examines how Artificial Intelligence (AI) techniques are used in musical composition. It seeks to understand both its capabilities and its limitations through an analysis of the AI methodologies used in this context and its historical evolution. In addition, concrete examples of underlying algorithms are examined, providing a comprehensive understanding of their use in modern musical composition. The specific objectives of the study include listing the different AI techniques used, explaining the fundamental principles underpinning these techniques, analyzing their limitations to find opportunities for improvement and assessing the impact of AI on the current musical composition, thus contributing to the understanding of this constantly evolving field.

**PALABRAS CLAVE** Inteligencia Artificial, composición musical, técnicas de Inteligencia Artificial, capacidades, limitaciones.

## I. INTRODUCCIÓN

A pesar de los avances recientes, persisten brechas sustanciales en el conocimiento sobre la aplicación de la IA en la composición musical, incluyendo limitaciones en las metodologías existentes y oportunidades sin explotar en la generación de música por medio de la IA. La aplicación de la IA en la composición musical es un desafío multidisciplinario que requiere la integración de conocimientos en teoría musical,

percepción humana de la música y la creación de algoritmos de IA sofisticados. Esta investigación busca abordar estas brechas de conocimiento y comprender mejor el uso de la IA en la composición musical por medio de la literatura disponible y la experimentación.

## II. MÉTODOS

La sección de métodos de esta investigación sobre la composición musical asistida por

Inteligencia Artificial se estructuró con un enfoque meticuloso para comprender el estado actual de la investigación en este campo en crecimiento. Inicialmente, se formuló una pregunta de investigación precisa que guió el proceso de revisión sistemática de la literatura. Esta revisión buscó identificar áreas para la expansión y resaltar debilidades en el cuerpo de trabajo existente. Además, se aplicaron criterios rigurosos para la selección de fuentes, que incluyeron la relevancia directa con el tema de investigación, la credibilidad y fiabilidad de las fuentes, la actualidad de la información, la diversidad de perspectivas y la accesibilidad de las fuentes.

El proceso de experimentación también desempeñó un papel fundamental en esta investigación. El autor llevó a cabo experimentos prácticos utilizando ejemplos de código y metodologías de IA específicas para la composición musical. Estos ejemplos permitieron una comprensión más profunda de cómo se aplican las técnicas de IA en la creación musical y proporcionaron ejemplos concretos de su eficacia.

### III. METODOLOGÍAS DE IA EN LA COMPOSICIÓN MUSICAL

A manera de elucidar los principios fundamentales que sustentan las diversas metodologías de Inteligencia Artificial empleadas en el ámbito de la composición musical, se proporcionan a continuación las siguientes figuras con la organización de los principios relevantes para cada metodología:

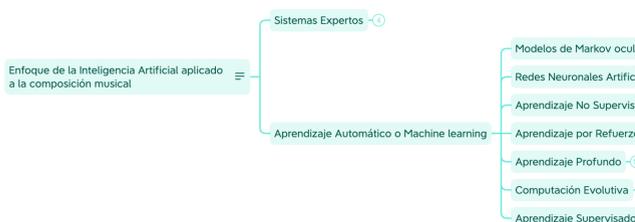


Figura 1 Listado de metodologías. Fuente: Elaboración propia.

#### SISTEMAS EXPERTOS



Figura 2 Principios de Sistemas Expertos. Fuente: Elaboración propia.

#### MODELOS DE MARKOV OCULTOS

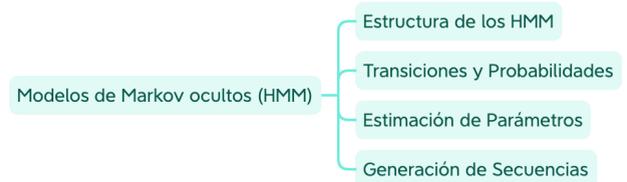


Figura 3 Principios de Modelos de Markov Ocultos. Fuente: Elaboración propia.

#### REDES NEURONALES ARTIFICIALES

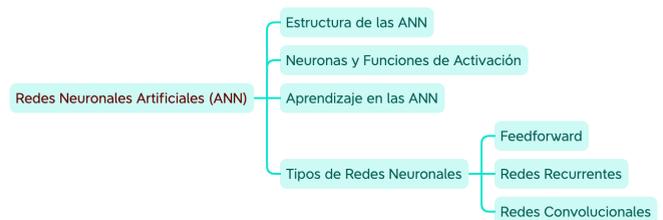


Figura 4 Principios de Redes Neuronales Artificiales. Fuente: Elaboración propia.

#### APRENDIZAJE SUPERVISADO

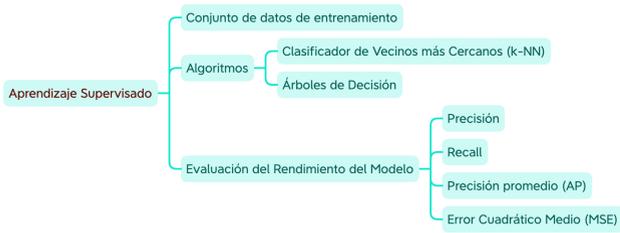


Figura 5 Principios de Aprendizaje Supervisado. Fuente: Elaboración propia.

## APRENDIZAJE NO SUPERVISADO



Figura 6 Principios de Aprendizaje No Supervisado. Fuente: Elaboración propia.

## APRENDIZAJE POR REFUERZO



Figura 7 Principios de Aprendizaje por Refuerzo. Fuente: Elaboración propia.

## APRENDIZAJE PROFUNDO

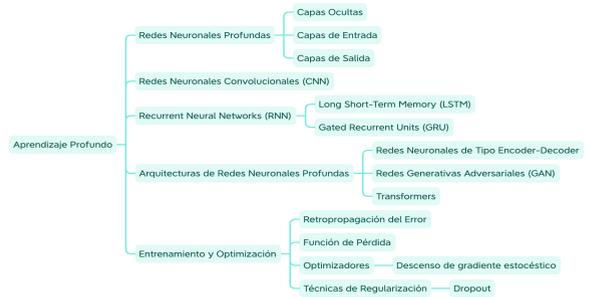


Figura 8 Principios de Aprendizaje Profundo. Fuente: Elaboración propia.

## COMPUTACIÓN EVOLUTIVA

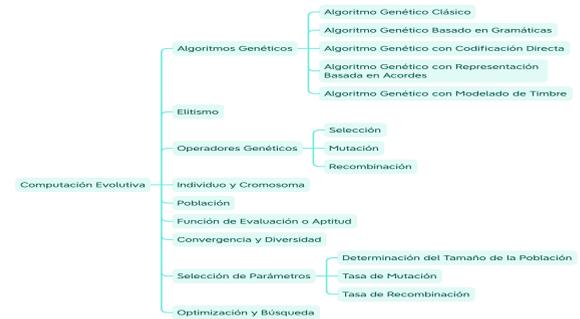


Figura 9 Principios de Computación Evolutiva. Fuente: Elaboración propia.

## IV. DESARROLLO HISTÓRICO DE LA INTELIGENCIA ARTIFICIAL EN LA COMPOSICIÓN MUSICAL

Se llevó a cabo una revisión exhaustiva de la literatura relacionada con la inteligencia artificial en la composición musical, recopilando información de diversas fuentes académicas y especializadas. Aquí se destacan los hitos más importantes en el desarrollo de técnicas de inteligencia artificial en la composición musical y se realiza un análisis crítico de la literatura disponible.

---

## **SISTEMAS EXPERTOS**

Los orígenes de los Sistemas Expertos en composición musical se remontan a la década de 1960, con un hito importante en 1961: la creación del sistema "Harmonet" por Hiller e Isaacson, que buscaba producir música tonal mediante reglas predeterminadas (Hiller & Isaacson, 1962). Este esfuerzo inicial estableció las bases para avances futuros. En la década de 1970, se produjo un avance significativo con la aparición de algoritmos capaces de analizar composiciones musicales existentes, luego evolucionaron a sistemas basados en reglas utilizados por Cope en 1997 para examinar y generar música al estilo de diferentes compositores (Kostka & Cope, 1997). Este estudio demostró el potencial de los sistemas expertos para replicar la creatividad de compositores humanos. La evolución de los sistemas expertos en composición musical ha sido notable a lo largo de su historia, desde sistemas basados en reglas iniciales hasta la incorporación posterior de enfoques de aprendizaje automático y aprendizaje profundo.

## **MODELOS DE MARKOV OCULTOS**

El desarrollo histórico de los HMM se puede remontar a finales de la década de 1960, cuando Leonard E. Baum y sus colegas introdujeron el algoritmo Baum-Welch, un algoritmo fundamental para el entrenamiento de HMMs. En los años 2000, Simon Dixon marcó un hito al emplear HMMs en la transcripción automática de música (Dixon, 2001). Un ejemplo notable de aplicación más reciente es DeepBach de Gatan Hadjeres y François Pachet, que emplea redes neuronales recurrentes (Hadjeres & Pachet, 2017). En los últimos años, se han propuesto modelos híbridos que combinan HMMs con redes neuronales profundas para generar secuencias musicales expresivas (Hawthorne et al., 2018).

## **REDES NEURONALES ARTIFICIALES**

En los años 80, Todd empleó redes conectivistas para extraer patrones musicales (Bharucha & Todd, 1989). La década de 1990 vio la llegada de redes neuronales recurrentes (RNN) y redes LSTM (Long Short - Term Memory) para

representar conexiones temporales en la música (Hochreiter & Schmidhuber, 1997). Luego, las redes convolucionales (CNNs) y las GANs (Redes Generativas Adversariales) se usaron para mejorar las capacidades de las ANNs en la creación y análisis musical (Heaton, 2017). Estos avances han impulsado el uso de ANNs en la composición musical, desde las primeras redes conectivistas hasta las modernas arquitecturas de aprendizaje profundo.

## **APRENDIZAJE SUPERVISADO**

A principios de la década de 1980, se exploraron algoritmos de aprendizaje supervisado para crear música. Pero se puede retroceder hasta la década de los 50s donde Hiller e Isaacson (1959) presentaron la idea de usar computadoras para componer música basada en reglas. Todd (1989) avanzó en esto con redes neuronales de retropropagación. Biles (1997) creó GenJam, que improvisaba jazz basándose en músicos humanos. Tzanetakis y Cook (2002) usaron técnicas supervisadas para clasificar géneros musicales. Mozer (2007) introdujo "Concerto", una red neuronal recurrente que aprendía patrones musicales. En los últimos años, las redes adversarias generativas han ganado popularidad en la composición musical, como "MuseGAN" de Huang et al. (2018) que generó música de piano *multitrack*.

## **APRENDIZAJE NO SUPERVISADO**

Iannis Xenakis en la década de 1950 utilizó técnicas estocásticas en "*Metastaseis*" para crear música innovadora (Xenakis, 1955). En la década de 1980, David Cope aplicó algoritmos de aprendizaje sin supervisión en "Experiments in Musical Intelligence" para analizar y replicar el estilo de compositores (Cope, 1996). Tzanetakis y Cook (2002) contribuyeron al análisis musical no supervisado usando algoritmos de agrupación. En el año 2020, se introdujo un algoritmo de aprendizaje profundo sin supervisión para interpretación de audio (Wang et al., 2020), que demostró una alta precisión en la clasificación de entornos acústicos.

---

## APRENDIZAJE POR REFUERZO

Uno de los mayores hitos del aprendizaje supervisado en la composición musical comenzó con Mozer (2007) explorando la generación de melodías y armonías mediante una variante de Q-learning. En 2017, Hadjeres y Pachet presentaron "DeepBach", que empleaba redes neuronales profundas para armonizar en el estilo de Bach. Luego, Huang et al. (2017) creó un sistema de improvisación musical basado en aprendizaje por refuerzo. Un avance más reciente en 2021 (Kritsis et al., 2021) se centró en la improvisación en tiempo real en jazz, utilizando redes neuronales recurrentes y aprendizaje por refuerzo. Entrenaron a las redes para crear acompañamientos coherentes y estéticamente adecuados, recompensándolas por producir resultados de alta calidad.

## APRENDIZAJE PROFUNDO

El aprendizaje profundo en la composición musical comenzó a principios de los años 2000, explorando la generación de música más compleja y coherente. Hinton et al. (2006) introdujeron redes de creencia profunda (DBNs) que permitieron modelar patrones y estructuras musicales complicadas, facilitando composiciones más complejas. Los investigadores buscaron la combinación de redes neuronales profundas con otros paradigmas de IA. Huang et al. (2016) propusieron un modelo que fusiona redes de creencia profundas con máquinas Boltzmann restringidas. Se exploraron Redes Neuronales Convolucionales (CNNs) y Redes Neuronales Recurrentes (RNNs) en la creación musical. WaveNet, desarrollado por van den Oord et al. (2016), empleó CNNs para producir música realista y de alta calidad. Luego, se refinaron las técnicas de aprendizaje profundo con la introducción de redes adversarias generativas en la generación musical (Briot et al., 2017). Simon et al. (2017) propusieron Performance RNN, un modelo basado en redes LSTM para generar música expresiva.

## COMPUTACIÓN EVOLUTIVA

En el año 2000, se presentó Vox Populi, un sistema interactivo que utiliza algoritmos genéticos para generar música colaborativamente (Moroni et al., 2000). Este enfoque demostró ser viable y eficaz en la producción de música algorítmica, sugiriendo la posibilidad de crear música innovadora mediante sistemas evolutivos interactivos. Posteriormente, Gartland y Copley (2003) investigaron el uso de algoritmos genéticos en la creación musical y concluyeron que los algoritmos genéticos pueden ser herramientas valiosas, pero requieren más estudio para optimizar su uso. En 2021, Guo propuso el uso de algoritmos genéticos interactivos (IGAs) para generar música, utilizando una función de *fitness* basada en similitud de intervalos (Guo, 2021).

## V. ALGORITMOS

A continuación se explican los mecanismos y algoritmos subyacentes con ejemplos de código, utilizados por diversas técnicas de generación de música basadas en Inteligencia Artificial. En esta sección se proveen ejemplos de las formas más básicas de las metodologías para facilitar la comprensión de los conceptos y técnicas.

### SISTEMA EXPERTO CON LÓGICA DE INFERENCIA

El código de este ejemplo, ha sido puesto a disposición del público en el siguiente enlace: <https://zenodo.org/records/10162336>.

El proceso implica una base de conocimientos con información sobre escalas, acordes, ritmo y firma de tiempo. Las reglas establecen condiciones y acciones. Cuando se cumple una condición, se ejecuta la acción correspondiente. En este caso, hay tres reglas: cambiar el ritmo de 120 a 140, modificar la firma de tiempo de 4/4 a 3/4 y eliminar la nota "C" en la escala mayor. Un motor de inferencia revisa las reglas y aplica las acciones. El resultado es una base de conocimientos actualizada.

### MODELO DE MARKOV OCULTO SIMPLIFICADO

---

El código de este ejemplo, ha sido puesto a disposición del público en el siguiente enlace: <https://zenodo.org/records/10162359>.

El proceso comienza estableciendo matrices de transición y emisión junto con las probabilidades de estado inicial. Luego, se define 'num\_notes' para determinar la cantidad de notas a generar. Se elige el estado inicial de manera estocástica usando 'random.choices()' según las probabilidades. Luego, se itera 'num\_notes' veces, registrando el estado actual en 'music\_sequence' en cada iteración y actualizándose según las probabilidades de transición. Finalmente, se selecciona aleatoriamente una nota de cada estado basándose en las probabilidades de emisión para convertir la secuencia en notas.

### **RED NEURONAL ARTIFICIAL CON LSTM**

El código de este ejemplo, ha sido puesto a disposición del público en el siguiente enlace: <https://zenodo.org/records/10162371>.

El proceso comienza importando bibliotecas como 'numpy' y 'Keras'. Los datos de entrada son dos escalas musicales: C mayor y menor. Los datos de salida son similares, pero cada nota se desplaza una hacia la siguiente en la escala. Se convierten en matrices y se normalizan. Se define un modelo RNN con una capa LSTM de 128 unidades y una capa densa lineal. Se utiliza el error cuadrado medio (MSE) como función de pérdida y el optimizador Adam. El modelo se entrena durante 1000 epochs con un tamaño de lote de 2, utilizando los datos de entrada y salida. Después del entrenamiento, se genera una nueva secuencia musical insertándose en el modelo y ajustando la salida para que esté en el rango original.

### **APRENDIZAJE SUPERVISADO CON K-NN**

El código de este ejemplo, ha sido puesto a disposición del público en el siguiente enlace: <https://zenodo.org/records/10162377>.

El proceso se describe de la siguiente manera: se importan las bibliotecas necesarias, como 'numpy' para operaciones numéricas y

'KNeighborsClassifier' del módulo 'sklearn.neighbors' para implementar el algoritmo KNN. Se preparan los datos de entrenamiento, llamados 'training\_data', como una matriz bidimensional en la que cada fila representa un punto de datos con dos características, como ritmo y energía. Las etiquetas correspondientes se almacenan en el arreglo 'labels', por ejemplo, 'Clásico' o 'Rock'. Se crea una instancia de la clase 'KNeighborsClassifier' con un valor de 'k=3' para el clasificador KNN. Se procede a entrenar el clasificador usando el método 'fit()', que toma los datos de entrenamiento y las etiquetas como argumentos. Se prepara un nuevo conjunto de datos, 'test\_data', para la clasificación, asegurándose de que tenga el mismo número de características que los datos de entrenamiento. Finalmente, se determina la etiqueta del nuevo conjunto de datos utilizando el método 'predict()' del clasificador KNN.

### **APRENDIZAJE NO SUPERVISADO CON FACTORIZACIÓN DE MATRIZ NO NEGATIVA (NMF)**

El código de este ejemplo, ha sido puesto a disposición del público en el siguiente enlace: <https://zenodo.org/records/10162380>.

Para empezar, se incluyen las bibliotecas esenciales para el análisis. Específicamente, se importa 'numpy' para facilitar los cálculos numéricos y 'NMF' de 'sklearn.decomposition' para ejecutar la Factorización de Matriz No Negativa. Para empezar en este ejemplo, el conjunto de datos de música debe cargarse desde un archivo llamado 'music\_dataset.txt'. Para generar un modelo de NMF, se crea una instancia del modelo con el número especificado de componentes. En este caso particular, el valor asignado al parámetro 'n\_components' es 10. El modelo NMF se incorpora al conjunto de datos de música entrenándolo utilizando el método 'fit'. Las características aprendidas y los vectores básicos se extraen utilizando los atributos 'transform' y 'components\_' posteriores al proceso de ajuste del modelo.

### **APRENDIZAJE POR REFUERZO CON Q-TABLE**

---

El código de este ejemplo, ha sido puesto a disposición del público en el siguiente enlace: <https://zenodo.org/records/10162392>.

El código se divide en tres fases. En la primera, se importan bibliotecas y se establecen parámetros, estados y acciones. En la segunda, ocurre el entrenamiento a través de episodios donde se actualiza la Tabla Q usando la ecuación Q-Learning. En la tercera, se emplea la Tabla Q para generar una melodía en la función 'generate\_melody'.

#### **APRENDIZAJE PROFUNDO SIMPLIFICADO**

El código de este ejemplo, ha sido puesto a disposición del público en el siguiente enlace: <https://zenodo.org/records/10162395>.

En este código, primero se importan bibliotecas esenciales, como 'numpy' para operaciones numéricas y varias clases de 'keras' para construir el modelo. Luego, se define el conjunto de datos, que consta de secuencias de notas. Las secuencias de entrada y salida se normalizan y codifican según las necesidades del modelo. A continuación, se construye la arquitectura del modelo, que incluye una capa LSTM y una capa densa. El modelo se compila con parámetros específicos. Finalmente, el código utiliza semillas de entrada para predecir notas musicales subsiguientes y crear composiciones musicales.

#### **COMPUTACIÓN EVOLUTIVA SIMPLIFICADA**

El código de este ejemplo, ha sido puesto a disposición del público en el siguiente enlace: <https://zenodo.org/records/10162401>.

El programa busca generar una melodía de destino especificada. Se configuran parámetros como el tamaño de la población, el número de generaciones, y el rango de notas posibles. La aptitud se mide calculando las discrepancias absolutas entre las notas de la melodía deseada y la proporcionada. Se genera una población inicial de melodías de manera aleatoria. En cada generación, se calcula la aptitud de cada melodía. Luego, se seleccionan padres a través de un proceso de torneo. La descendencia se crea

mediante *crossover*, combinando partes de los padres. La nueva generación puede experimentar mutaciones aleatorias en sus notas. Al final de las generaciones, se identifica la melodía con la mayor aptitud como la óptima.

## **VI. LIMITACIONES**

### **SISTEMAS EXPERTOS**

La producción musical a través de sistemas expertos se ve limitada por su incapacidad para reflejar plenamente la gama de emociones humanas y la creatividad musical, argumentando Papadopoulos y Wiggins (1999) que estos sistemas se centran en procesos matemáticos y lógicos, dejando de lado los aspectos emocionales y creativos. Además, carecen de versatilidad y adaptabilidad a las tendencias musicales cambiantes, lo que hace que su creación y mantenimiento sean costosos según Kersten y Bloechliger (2007).

### **MODELOS DE MARKOV OCULTOS**

Una de las limitaciones de los modelos de Markov ocultos, señalada por Bello y Pickens (2005), es la suposición de estacionalidad, donde las distribuciones de probabilidad de los estados permanecen constantes a lo largo del tiempo. Esta suposición puede no ser aplicable en el contexto musical, ya que las características de una composición musical evolucionan con el tiempo, lo que sugiere que los modelos ocultos de Markov podrían no captar con precisión la dinámica de las composiciones. Determinar el número óptimo de estados es una tarea compleja, y la elección inapropiada de características puede llevar a la incorrecta emulación de patrones musicales.

### **REDES NEURONALES ARTIFICIALES**

Una de las principales limitaciones de las redes neuronales artificiales, como indican Huang, Yang y Wu (2016), radica en su necesidad de grandes cantidades de datos de alta calidad para lograr un rendimiento óptimo en la generación de composiciones musicales coherentes y estéticamente atractivas. Además, se enfrenta el

---

desafío de generar obras musicales verdaderamente distintivas e innovadoras, ya que las redes neuronales artificiales tienden a basarse en patrones preexistentes, lo que puede resultar en composiciones con similitudes a obras anteriores.

### **APRENDIZAJE SUPERVISADO**

La tarea de etiquetar datos musicales es laboriosa y demanda conocimientos especializados, lo que dificulta la creación de conjuntos de datos extensos adecuados para el entrenamiento. Estas limitaciones y desafíos destacan la necesidad de que los investigadores exploren enfoques alternativos que superen estas restricciones y mejoren la eficacia de los modelos en el contexto de la composición musical.

### **APRENDIZAJE NO SUPERVISADO**

La ausencia de una métrica de evaluación es una de las desventajas más significativas. Es difícil evaluar la calidad de la salida generada por un algoritmo de aprendizaje sin supervisión. Como resultado, resulta difícil determinar si el algoritmo ha aprendido o no algo de valor. Además un gran conjunto de datos es necesario para que los algoritmos de aprendizaje no supervisados descubran patrones y correlaciones significativas. Puede ser problemático localizar tal conjunto de datos en el campo de la composición musical.

### **APRENDIZAJE POR REFUERZO**

La dificultad para definir una función de recompensa que refleje la calidad de la música es un desafío clave en el uso del aprendizaje por refuerzo en la composición musical. Además, la complejidad intrínseca de la creación musical dificulta el desarrollo de un modelo de aprendizaje por refuerzo que pueda capturar con precisión sus sutilezas, como señalaron Sturm y Ben-Tal (2017). La escasez de datos de alta calidad también se convierte en un obstáculo, ya que las composiciones musicales son poco comunes y efímeras en comparación con otras disciplinas, lo que dificulta la formación de modelos de aprendizaje por refuerzo con

conjuntos de datos adecuados para representar las características musicales deseadas.

### **APRENDIZAJE PROFUNDO**

La falta de interpretación de estos modelos, como señalan Choi et al. (2017), los hace parecer cajas negras, lo que dificulta comprender los mecanismos subyacentes que generan las salidas del modelo. Esto puede ser problemático para los compositores, ya que la interpretación es esencial para crear composiciones significativas. Además, la necesidad de conjuntos de datos sustanciales, resaltada por Hadjeres y Pachet (2017), se vuelve complicada en el ámbito musical debido a la escasez de datos, especialmente en géneros especializados. La falta de precisión en las salidas de los modelos de aprendizaje profundo también representa un desafío en el proceso de formación.

### **COMPUTACIÓN EVOLUTIVA**

La complejidad de la música combinada con la dificultad de codificar, como señalan Pachet et al. (2011), es una barrera significativa para la eficacia de estos algoritmos. Además, como señalan Latorre-Biel et al. (2015), la ausencia de una función de *fitness* claramente definida hace que sea difícil evaluar los resultados en la composición musical ya que no existe una manera cuantificable de medir la calidad musical. Definir los parámetros que determinan la calidad musical se convierte en una tarea compleja. Además, los algoritmos de Computación Evolutiva requieren recursos de procesamiento sustanciales y tiempo para producir composiciones musicales, lo que resulta en altos costos computacionales.

## **VII. CONCLUSIONES**

Como notas finales cabe resaltar que el autor ha podido identificar áreas de mejora que tienen amplia aplicabilidad en estas disciplinas. Se ha descubierto que todos estos métodos tienen una cosa en común: se necesita mejorar urgentemente la calidad y cantidad de los datos de entrada, lo que es crucial para el proceso creativo asistido por la inteligencia artificial. A nivel personal, el autor recomienda para el entusiasta que desea

---

iniciar a componer música por medio de inteligencia artificial, ver algunos modelos de IA generativos que utilizan la arquitectura de *Transformers* dentro de la categoría de *Deep Learning*, además de estudiar a fondo las Redes Neuronales Artificiales.

También está claro cuán crucial es analizar y comprender los mecanismos subyacentes de los métodos utilizados. La necesidad de mayor claridad y transparencia en los procedimientos seguidos y los resultados de los algoritmos utilizados es un reflejo de este requisito. La complejidad de la codificación de datos musicales presenta un nivel adicional de dificultad que requiere una mejora y atención constantes.

Esta investigación y la serie de pruebas ha reforzado la creencia del autor en las posibilidades de estas metodologías. A pesar de tener varios retos, existe una alentadora perspectiva para el avance en la composición musical a medida que se llenan las lagunas en el conocimiento, se mejora la calidad de los datos y se perfeccionan los métodos de interpretación.

## REFERENCIAS

- [1] Baum, L. E., Petrie, T., Soules, G., & Weiss, N. (1970, February). A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *The Annals of Mathematical Statistics*, 41(1), 164–171. <https://doi.org/10.1214/aoms/1177697196>.
- [2] Bello, J. P., & Pickens, J. (2005). A robust mid-level representation for harmonic content in music signals. In *Proceedings of the 6th International Conference on Music Information Retrieval* (pp. 304-311). London: Queen Mary, University of London.
- [3] Bharucha, J. J., & Todd, P. M. (1989). Modeling the Perception of Tonal Structure with Neural Nets. *Computer Music Journal*, 13(4), 44. <https://doi.org/10.2307/3679552>.
- [4] Biles, J. A. (1997, November 1). GenJam: An interactive genetic algorithm jazz improviser. *The Journal of the Acoustical Society of America*, 102(5\_Supplement), 3181–3181. <https://doi.org/10.1121/1.420841>.
- [5] Briot, J. P., Hadjeres, G., & Pachet, F. D. (2019, November 8). *Deep Learning Techniques for Music Generation*. Springer.
- [6] Cope, D. (1996, January 1). *Experiments in Musical Intelligence*.
- [7] Dixon, S. (2001, March 1). Automatic Extraction of Tempo and Beat From Expressive Performances. *Journal of New Music Research*, 30(1), 39–58. <https://doi.org/10.1076/jnmr.30.1.39.7119>.
- [8] Dong, H. W., Hsiao, W. Y., Yang, L. C., & Yang, Y. H. (2018, April 25). MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1). <https://doi.org/10.1609/aaai.v32i1.11312>.
- [9] Gartland-Jones, A., & Copley, P. (2003, September). The Suitability of Genetic Algorithms for Musical Composition. *Contemporary Music Review*, 22(3), 43–55. <https://doi.org/10.1080/0749446032000150870>.
- [10] Guo, Q. (2021, November 1). Computer-assisted Music Composition Algorithm Design Dependent on Interactive Genetic Algorithm with Interval Fitness. *Journal of Physics: Conference Series*, 2066(1), 012035. <https://doi.org/10.1088/1742-6596/2066/1/012035>.
- [11] Hadjeres, G., Pachet, F., & Nielsen, F. (2017). DeepBach: a steerable model for Bach chorales generation. *arXiv preprint arXiv:1612.01010*.
- [12] Hawthorne, C., Elsen, E., Song, J., Roberts, A., Simon, I., Raffel, C., & Engel, J. (2018). Onsets and frames: Dual-objective piano transcription. *arXiv preprint arXiv:1710.11153*.
- [13] Heaton, J. Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. *Genet Program Evolvable Mach* 19, 305–307 (2017). <https://doi.org/10.1007/s10710-017-9314-z>.
- [14] Hinton, G. E., & Salakhutdinov, R. R. (2006, July 28). Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786), 504–507. <https://doi.org/10.1126/science.1127647>.

- 
- [15] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006, July). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7), 1527–1554. <https://doi.org/10.1162/neco.2006.18.7.1527>.
- [16] Hochreiter, S., & Schmidhuber, J. (1997, November 1). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [17] Huang, A., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., ... & Eck, D. (2017). Music Transformer: Generating music with long-term structure. arXiv preprint arXiv:1809.04281.
- [18] Huang, Y., & Wu, Y. (2016). Deep learning for music. arXiv preprint arXiv:1606.04930.
- [19] Huang, Y., & Wu, Y. (2016). Deep learning for music. arXiv preprint arXiv:1606.04930.
- [20] Kersten, G., & Bloechliger, A. (2007). An empirical evaluation of generative music systems. *Proceedings of the 11th International Conference on Music Information Retrieval*, 275–282.
- [21] Kostka, S., & Cope, D. (1997, September). Experiments in Musical Intelligence. *Notes*, 54(1), 98. <https://doi.org/10.2307/899968>.
- [22] Kritsis, K., Kylafi, T., Kaliakatsos-Papakostas, M., Pikrakis, A., & Katsouros, V. (2021, February 12). On the Adaptability of Recurrent Neural Networks for Real-Time Jazz Improvisation Accompaniment. *Frontiers in Artificial Intelligence*, 3. <https://doi.org/10.3389/frai.2020.508727>.
- [23] L. Sturm, B., & Ben-Tal, O. (2017, September 1). Taking the Models back to Music Practice: Evaluating Generative Transcription Models built using Deep Learning. *Journal of Creative Music Systems*, 2(1). <https://doi.org/10.5920/jcms.2017.09>.
- [24] Moroni, A., Manzolli, J., Zuben, F. V., & Gudwin, R. (2000, December). Vox Populi: An Interactive Evolutionary System for Algorithmic Music Composition. *Leonardo Music Journal*, 10, 49–54. <https://doi.org/10.1162/096112100570602>.
- [25] Mozer, M. C. (2007). Neural Network Music Composition by Prediction: Exploring the Benefits of Psychoacoustic Constraints and Multi-scale Processing. *Connection Science*, 6(2–3), 247–280. <https://doi.org/10.1080/09540099408915726>.
- [26] Papadopoulos, C., & Wiggins, G. (1999). Musical creativity and expert systems. *Journal of New Music Research*, 28(1), 3–21.
- [27] Schot, J. W., Hiller, L., & Isaacson, L. M. (1962, October). Experimental Music Composition with an Electronic Computer. *Mathematics of Computation*, 16(80), 507. <https://doi.org/10.2307/2003155>.
- [28] Todd, P. M. (1989). A Connectionist Approach to Algorithmic Composition. *Computer Music Journal*, 13(4), 27. <https://doi.org/10.2307/3679551>.
- [29] Tzanetakis, G., & Cook, P. (2002, July). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5), 293–302. <https://doi.org/10.1109/tsa.2002.800560>.
- [30] Tzanetakis, G., & Cook, P. (2002, July). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5), 293–302. <https://doi.org/10.1109/tsa.2002.800560>.
- [31] van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... & Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499.
- [32] Wang, M., Zhang, X. L., & Rahardja, S. (2020, March 19). An Unsupervised Deep Learning System for Acoustic Scene Analysis. *Applied Sciences*, 10(6), 2076. <https://doi.org/10.3390/app10062076>.
- [33] Westergaard, P., Hiller, L. A., & Isaacson, L. M. (1959, November). Experimental Music Composition with an Electronic Computer. *Journal of Music Theory*, 3(2), 302. <https://doi.org/10.2307/842857>.
- [34] Xenakis, I. (1955). *Metastaseis*. Paris, France: Editions Salabert.