

# Composición musical usando algoritmos genéticos

## *Composing music through genetic algorithms*

### **JULIO CÉSAR CARVAJAL RAMÍREZ**

Ingeniero de Sistemas. Investigador de la Fundación Universitaria San Martín. Bogotá, Colombia. Contacto: [jc082006@ingenieria.sanmartin.edu.co](mailto:jc082006@ingenieria.sanmartin.edu.co)

### **FABIÁN ANDRÉS GIRALDO GIRALDO**

Ingeniero de Sistemas e Informática, candidato a magister en Ingeniería de Sistemas y Computación. Docente investigador de la Fundación Universitaria San Martín. Bogotá, Colombia. Contacto: [fabian.giraldo@ingenieria.sanmartin.edu.co](mailto:fabian.giraldo@ingenieria.sanmartin.edu.co)

Fecha de recepción: 31 de agosto de 2011

Fecha de aceptación: 14 de febrero de 2012

Clasificación del artículo: Investigación

**Palabras clave:** algoritmos genéticos, inteligencia artificial, música.

**Key words:** genetic algorithms, artificial intelligence, music.

## **RESUMEN**

Este artículo presenta un modelo de composición musical discreto, basado en las técnicas de escritura musical propuestas por Johann Sebastian Bach. El modelo fue implementado mediante el uso de la estrategia metaheurística: algoritmos genéticos implantados sobre el Framework JGap, el cual arroja resultados de composición con una sintaxis que cumple con las técnicas de escritura y composición musical. Se usará en este acercamiento la escala musical natural y los sonidos

sintetizados de la octava natural del piano. Se crearon acordes tomando los grados respectivos en la escala y se agruparon las notas y acordes para convertirlos en compases. Mediante el modelo propuesto se obtiene: 1) El conjunto de notas agrupadas por compases con su interpretación de escritura sintáctica siguiendo los lineamientos del Framework JFugue, y 2) la partitura de los fragmentos producidos sobre un pentagrama respetando las reglas de escritura musical formal. Se presentan tres MIDIS de composiciones hechas por el computador donde se muestra una fuerte

evolución de sonido y como tal un resultado audible de la composición.

## ABSTRACT

This paper presents a model of discrete musical composition based on musical writing techniques proposed by Johann Sebastian Bach. The model was implemented using the Meta-heuristics strategy, namely using Genetic algorithms implemented on the JGap Framework, producing musical results with a syntax that matches writing and music composition techniques. This approach is applied to

the natural musical scale, and also to synthesized sounds of the natural-piano octave. Chords were created by taking the corresponding grades in the scale; additionally, notes and chords were grouped to become actual score bars. Using the proposed model, the following is obtained: 1) A set of notes grouped in bars with their corresponding interpretation, following the guidelines of syntactic JFugue Framework; 2) the score of the fragments produced written on a staff that complies with the rules of formal notation. Three MIDI files corresponding to three compositions (made by the computer) are presented, showing a strong evolution of sound and, as such, an audible result of a song.

\* \* \*

## 1. INTRODUCCIÓN

La composición musical es un tema difícil de abordar desde los términos de la computación, desde los inicios de las artes se ha hecho una estigmatización de ellas con respecto al avance tecnológico que se vive en la actualidad. En este artículo se muestran los resultados del uso de una de las técnicas metaheurísticas más conocidas y aplicadas actualmente en la computación evolutiva: los algoritmos genéticos, con el fin de percibir cómo emerge la música desde un proceso evolutivo, que podría tardar décadas, tal vez siglos, en mostrar resultados con un ciclo de evolución musical a nivel de composición humano.

Ya varios estudios se han hecho sobre el tema, con resultados interesantes, como el caso de Risto Miikkulainen y Chun-Chi J. Chen, quienes aplicando redes neuronales recurrentes representan la notación musical y generan melodías basados en formulaciones matemáticas que les permiten construir un modelo de control de las restricciones de diversidad rítmica y medidas de restricción de densidad [1]; otro trabajo sobresaliente es el desarrollado por Reis et al [2], donde presentan

esquemas compuestos por algoritmos genéticos que aleatoriamente generan secuencias de notas ordenadas en la escala natural y luego son evaluadas por funciones matemáticas tales como las estudiadas en el cálculo diferencial (primera, segunda y tercera derivada de una función) que bajo criterio de los diseñadores son implementadas para seleccionar las mejores secuencias generadas por el algoritmo [2]. En el modelo presentado por la Universidad de Coruña, España, escrito por Pozos et al [3], se plasma la idea de usar los ordenadores para actividades artísticas como la composición musical y se presenta una estructura donde se emulan tres músicos artificiales, cada uno con un patrón que responde a un algoritmo genético para crear las composiciones de diferentes instrumentos como el violín, guitarra y la tuba, para luego ser evaluadas por una función fitness estructurada con métodos matemáticos discretos como teoremas de integración y derivación. El resultado que se obtiene es una base de datos con fragmentos musicales generados por el modelo.

Los resultados obtenidos por los autores indican que los algoritmos genéticos pueden diseñarse para obtener soluciones a este tipo de problemas,

pero desde el punto de vista de la teoría musical la falencia está en las funciones de evaluación de las melodías compuestas, ya que, en los trabajos mencionados, utilizaron estrictamente formulaciones matemáticas para considerar un fragmento generado por el algoritmo, pero están dejando a un lado la teoría y las técnicas de composición musical que están establecidas desde siglos atrás. Aunque las matemáticas tengan una relación directa con la composición musical, es necesario aclarar que se debe hacer uso de estas para dar posibles soluciones a problemas computacionales de composición, pero que las funciones de evaluación deben basarse en las reglas de escritura y sintaxis musical.

En el presente trabajo se expone el diseño de un algoritmo genético que produce composiciones musicales simulando la capacidad de composición humana, emitiendo sonidos sintetizados a partir del piano donde el computador se convierte en un instrumento virtual y en un compositor con capacidad para gestar diversos fragmentos musicales que se convertirán finalmente en una pieza musical terminada.

Como resultado final se expone un conjunto de notas que representan los compases generados a partir de la evolución del algoritmo, una tablatura final con el fragmento evolucionado, y una partitura que cumple con los estándares de escritura musical; tres MIDIS compuestos por el algoritmo donde se muestran diferentes fases del proyecto y una evolución de sonido considerable con respecto a las anteriores.

En este artículo, en cambio, no solo se exponen las funciones utilizadas para generar las notas que hacen parte de la composición, también se tienen en consideración elementos musicales tales como: tempos, compases, acordes y escritura formal en el pentagrama, para posteriormente ser utilizados en procesos de evaluación automática y, de esta forma, dar origen a los fragmentos audi-

bles en formato MIDI que respaldan los postulados de este trabajo y complementan los resultados teóricos que se están generando.

El artículo está organizado de la siguiente forma, en la sección 1: introducción, en la sección 2: conceptos musicales, en la sección 3: metodología, en la sección 4: diseño del algoritmo, en la sección 5: implementación, en la sección 6: resultados, y por último en la sección 7: conclusiones.

## 2. CONCEPTOS MUSICALES

Los siguientes párrafos contienen los conceptos necesarios para que el lector obtenga unas bases sólidas del punto de partida y dónde se quiere llegar con el uso de los algoritmos genéticos y la teoría musical aplicando las herramientas computacionales que ofrece la tecnología actualmente.

### 2.1 Técnica musical

La música, como toda manifestación artística, es un producto cultural. El fin de este arte es suscitar una experiencia estética en el oyente, expresar sentimientos, circunstancias, pensamientos o ideas. La música es un estímulo que afecta el campo perceptivo del individuo [4].

Para representar la música existen diversas grafías que se concibieron con el fin de poder escribir y trascender la música, estos símbolos representan las notas musicales y su respectiva duración, estos escritos ordenados sobre líneas representan su valor sonoro y generan una notación denominada pentagrama.

El pentagrama es el símbolo gráfico en el que se centra toda la grafía musical. Es en él, donde se escriben las notas musicales y otros signos como los compases o las fórmulas de compás. El pentagrama está compuesto por cinco líneas horizon-



**Figura 1.** Pentagrama con sus respectivas líneas y espacios.

Fuente: elaboración propia

tales y paralelas; además de equidistantes. Estas cinco líneas paralelas forman cuatro espacios entre ellas. En estos espacios también se ubican las notas musicales. En la práctica, hay cinco líneas y cuatro espacios. Estas líneas y espacios se nombran de abajo a arriba; así, por ejemplo, la línea de más abajo la podemos nombrar como primera línea [5], como lo muestra la figura 1.

## 2.2 Figura musical

Son signos musicales que indican la duración proporcional de los sonidos, como se muestra en la figura 2. Los silencios son signos para indicar una duración sin sonido, de ahí su nombre.

Una nota es un concepto musical que sirve para expresar la altura o tono de un sonido. Concreta-

	○	♪	♩	♪	♫	♬	♭
REDONDA	○	1	2	4	8	16	32
BLANCA	♪	1/2	1	2	4	8	16
NEGRA	♩	1/4	1/2	1	2	4	8
CORCHEA	♪	1/8	1/4	1/2	1	2	4
SEMICORCHEA	♫	1/16	1/8	1/4	1/2	1	2
FUSA	♬	1/32	1/16	1/8	1/4	1/2	1
SEMIFUSA	♭	1/64	1/32	1/16	1/8	1/4	1/2

**Figura 2.** Notas musicales y sus respectivas equivalencias entre ellas mismas.

Fuente: elaboración propia

mente, son nombres para ciertas frecuencias, que permiten decir que un sonido musical corresponde a cierto símbolo. Una combinación de varias notas está entre dos notas. Así una nota puede denotarse según la convención musical o mediante la expresión de su frecuencia, por ejemplo la 4 es igual a 440 hercios, o vibraciones por segundo en afinación estándar, o 444 hercios en afinación de cámara [6].

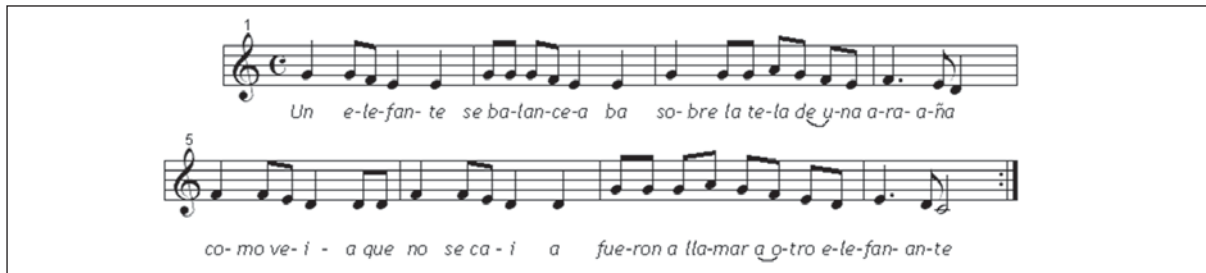
La música académica occidental ha desarrollado un método de escritura basado en dos ejes: el horizontal, que representa el transcurso del tiempo, y el vertical, que representa la altura del sonido; la duración de cada sonido está dada por la forma de las figuras musicales [6], como se puede observar en la figura 3 donde están expuestas las figuras musicales ordenadas a través de una línea horizontal que simboliza el tiempo de duración.

## 3. METODOLOGÍA

A continuación se desglosarán las diferentes técnicas utilizadas y las tecnologías que se fusionaron para el desarrollo del proyecto de composición musical usando algoritmos genéticos.

### 3.1 Representación de los individuos

Los individuos en los algoritmos genéticos se representan generalmente en cadenas binarias; pero en la solución presentada en este trabajo,



**Figura 3.** Método de escritura musical formal, utilizando el pentagrama.

Fuente: elaboración propia

serán representadas en cadenas de tipo entero que es otra de las técnicas comunes utilizadas para la representación de los individuos, o cromosomas, cada nota se compone de tres genes los cuales pueden tomar diversos valores según se configuren, como se muestra en la figura 4, de esta forma se genera un arreglo de tres posiciones; donde la primera posición o primer gen, hace referencia a la nota musical, y los posibles valores que puede tomar se denotan en el siguiente conjunto Notas = {D,E,F,G,A,B,C}, donde cada letra es la representación de las notas musicales en su escritura formal o un posible alelo para el componente número uno del gen; la segunda posición hace referencia al semitono manejado por la nota, y los posibles valores se denotan en el conjunto Semitono = {#, ##, b, bb}, donde el numeral simboliza un sostenido y la b un bemol; en la tercera posición se asigna la duración de la nota y los posibles valores se denotan en el conjunto Tiempo = {w, h, q, i, s, t, x, o}, donde cada letra hace referencia a un tiempo de duración. Para la generación de los compases se agrupan las notas según su duración, y de esa forma determinar el tiempo adecuado de cada compas.



**Figura 4.** Cromosoma diseñado.

Fuente: elaboración propia

Para generar un acorde mayor es necesario tomar el primero, tercero y quinto grado de la escala a la cual pertenezca, o las variaciones de las tres notas encontradas en estos grados, esto exactamente es lo que hace el algoritmo para generar acordes, de forma aleatoria busca los cromosomas cuyas notas han sido generadas con ausencia de semitono, en este caso los alelos comprendidos en el conjunto Semitono = {#, ##, b, bb}, y en esa posición reemplaza la nota existente por un acorde de la escala a la que pertenece la nota. En esta entrega se generarán acordes mayores de la escala fundamental o escala natural, claro está la teoría musical muestra detalladamente acordes que van desde dos hasta siete notas, pero en tal caso la complejidad del algoritmo aumentaría por tanto para esta entrega se hacen acordes de tres notas.

### 3.2 Grado de adaptación de los individuos

En el caso de algoritmos genéticos, la adaptación de un individuo coincide con el valor de la función objetivo a maximizar. Por lo tanto, el cálculo de la adaptación consiste en decodificar el cromosoma del individuo para identificar la melodía que representa y calcular el valor de la función objetivo en ese punto.

En esta entrega, el grado de adaptación de los individuos es calificado por el usuario, de tal forma que, en la función de evaluación, se pide un parámetro de 1 a 100, el cual va a representar una

calificación para cada grupo de cromosomas generados, donde los mejores puntuados serán los seleccionados como los más adaptados.

Es necesario especificar las condiciones en las que el algoritmo deja de evolucionar y presenta la mejor solución encontrada. Estas condiciones hacen referencia al punto de convergencia del algoritmo. En este caso se utilizó la condición de terminación más sencilla, donde al alcanzar un determinado número de generaciones de evolución el algoritmo muestra la solución o el cromosoma más adaptado.

## 4. IMPLEMENTACIÓN

Para el diseño del algoritmo genético se utilizó el Framework JGap, ya que es una herramienta de uso libre, y hace que la configuración del algoritmo sea rápida, comprensible, posee una arquitectura basada en el paradigma de programación orientada por objetos y patrones de diseño de software que lo hacen muy robusto, modificable y escalable, por otra parte, está escrito en el lenguaje de programación Java y esto otorga una integración limpia con los demás Frameworks usados.

### 4.1 Generación de la población

La población inicial se genera de forma completamente aleatoria, dando como resultado cadenas, como lo muestra la figura 3, que representan las notas a ser utilizadas en la solución. En el Framework JGap esta característica se obtiene utilizando el método `randomInitialGenotype`, provisto por la clase `Genotype`.

### 4.2 Operadores de selección

Los operadores de selección (el proceso de selección utilizado fue elitismo, que bajo el Framework

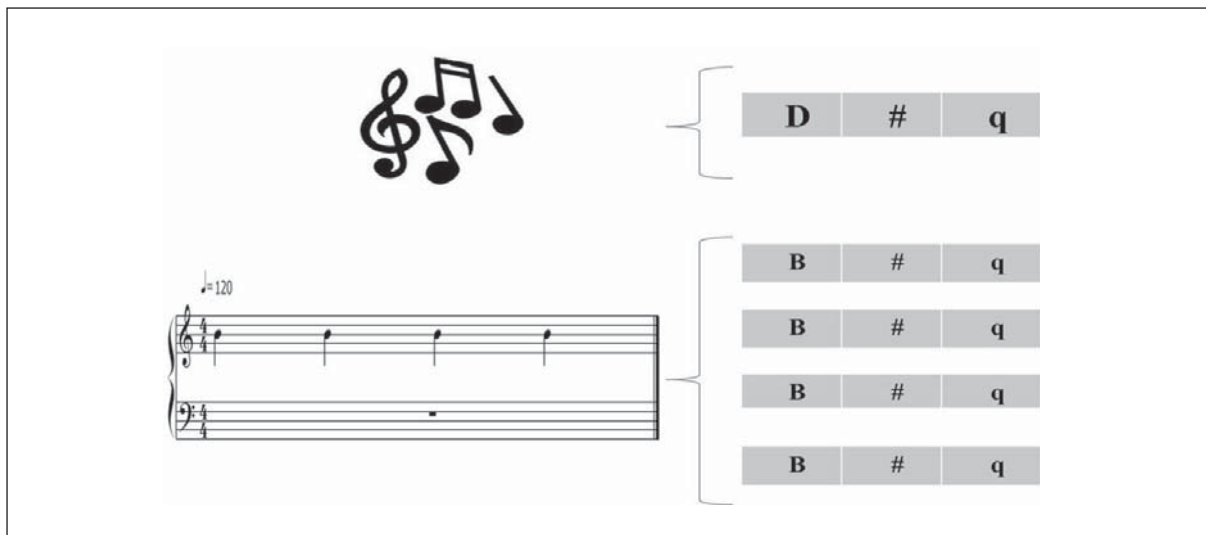
JGap está modelado bajo la clase `BestChromosomeSelector`) determinan qué individuos son los más adaptados al medio para darles trascendencia a la siguiente generación. En este acercamiento se configuró un operador de selección, que proporciona JGap, y se escribió un complemento para que el usuario sea partícipe de dicha selección, luego de escuchar la producción musical de una generación, otorga una calificación entre 1 y 100, siendo 1 la calificación más baja y 100 la más alta, a las cadenas generadas por el algoritmo genético, es así como se eligen las mejores melodías a perseverar en la línea de vida y pasar a la próxima generación, y de esta forma ser nuevamente evaluadas por el operador de selección anteriormente descrito.

### 4.3 Operadores de cruce y mutación

El Framework JGap permite usar los dos operadores de evolución como son: el cruce y la mutación de manera transparente. Para el diseño y desarrollo de este algoritmo genético se usaron estos dos operadores; el primero de ellos, el cruce monopunto consiste en tomar individuos de la población, dos vectores de notas que los identifica como el padre y la madre, y aplicarles el operador cruce, que consiste en tomar la primera mitad del padre y la segunda de la madre para engendrar el primer hijo y viceversa para el segundo [7]. El segundo operador, la mutación (bit a bit) consiste en tomar un individuo de la población y alterar su cadena genética, en este caso el cromosoma, dando diferentes valores a los alelos del mismo, que fluctúan entre los conjuntos anteriormente mencionados [7].

### 4.4 Representación de los conceptos musicales en la librería JFugue

En el algoritmo la gramática musical es representada por medio de cromosomas que son la base fundamental del funcionamiento de la algorítmica



**Figura 5.** Representación de las notas y los compases musicales generados por el algoritmo.

Fuente: elaboración propia

ca implementada, como se muestra en la figura 5, donde las notas musicales son una cadena, como se muestra en la figura 4. El Framework JFugue es el encargado de la comprensión del lenguaje gramatical musical y toda su interpretación de escritura en el lenguaje de programación Java. El Framework permite la fácil interpretación musical en un ámbito computacional, y aplicado a esta investigación se encarga de interpretar las cadenas que genera el algoritmo diseñado para tal fin. En su versión 1.0 demuestra ser una herramienta robusta para la programación musical, gracias a su facilidad de codificación, la documentación proporcionada por los desarrolladores y posibilidad de escalabilidad [8].

#### 4.5 Implementación de acordes

Para la generación de acordes usando el Framework JFugue, se debe concatenar con el símbolo + las notas que componen dicho acorde, un ejemplo de generación básico es un acorde en la escala natural donde las notas son C, E, G; la representación del acorde en la escritura del lenguaje Java usando JFugue como interprete sería

Cq+Eq+Gq, donde C es la fundamental, E la tercera y G la quinta, la q representa el tiempo de duración de cada nota en este caso  $\frac{1}{4}$ , el signo + indica que las notas están concatenadas, sonarán al tiempo y formarán un acorde mayor de la escala natural.

#### 4.6 Los tiempos de las notas

Las notas musicales también están catalogadas por los tiempos de duración en su frecuencia de sonido, es decir, cada una de ellas tiene un equivalente en duración: la redonda dura 1 tiempo, la blanca dura  $\frac{1}{2}$  tiempo, la negra dura  $\frac{1}{4}$  tiempo y así sucesivamente, como se muestra en la figura 6. En JFugue las duraciones de los tiempos están representados con letras w para redonda, h para la blanca, q para la negra, i para la corchea, s para la semicorchea, t para la fusa y x para la semifusa [9], letras que hacen parte de un conjunto llamado *time* anteriormente nombrado. De esta forma la programación de los tiempos por cada nota se hace muy agradable. El algoritmo tiene la capacidad de: generar los tiempos aleatoriamente siguiendo las reglas de composición de la teoría



**Figura 6.** Notas musicales básicas y su respectiva duración sobre un pentagrama musical.

Fuente: elaboración propia

musical y asignarle una duración adecuada a cada nota generada.

## 4.7 Compases

La generación de compases se hace concatenando las notas o cromosomas en el algoritmo, ya que se agrupan notas cuya duración de tiempos no supere los cuatro cuartos, esto garantiza estar generando, en esta primera entrega, compases en una medida de las más utilizadas en la música actual como el rock, pop [10].

## 4.8 Interpretación de compases

Se codificó un reproductor del Framework JFugue, que permite interpretar las notas generadas por el algoritmo en tiempo de ejecución para que la evaluación hecha por el usuario final sea más objetiva y acertada.

Este también permite escuchar de manera más clara los resultados que arroja el algoritmo por cada iteración, también acompaña al usuario final en un aprendizaje constante entre la gramática musical generada y los sonidos que están asociados a ella.

Esto, en términos generales, es aceptable, ya que por un lado genera composiciones musicales pero a su vez dichas composiciones pueden ser escuchadas en tiempo real, así, el usuario final tendrá un dominio más global de la problemática de la composición donde el computador genera cade-

nas con escritura discreta y sonidos sintetizados para su mayor comprensión.

## 4.9 MIDIS finales

Como último resultado se configura un conversor de la gramática generada por el algoritmo a un archivo MIDI, éste permitirá al usuario guardar cada uno de los patrones que se compusieron después de todas las iteraciones; es decir, la mejor solución encontrada por el algoritmo.

Tener este registro permite conservar las composiciones hechas por el algoritmo de una forma audible. Poder reproducir dichas composiciones en diferentes ordenadores y compartirlas, reproducir las cadenas de notas generadas en los reproductores de música portátiles.

Como parte de los resultados se generaron tres MIDIS, que muestran la evolución del proyecto en general, y se muestra la etapa de inicio con un track titulado Toccata N. 1 in C Major, la etapa central con un track titulado Molto Allegro in C Major, y la etapa final con un track titulado Computer Sonata Allegro in C Major; estos MIDIS son la prueba de la evolución del desarrollo y un eje fundamental para la continuidad de proyecto.

## 5. RESULTADOS

Los resultados que se obtuvieron después del diseño y la implementación del algoritmo y su respectiva función de evaluación fueron satisfac-



torios. En esta entrega se hizo un acercamiento a la composición musical aplicando algoritmos genéticos y técnicas musicales propuestas por Bach.

A grandes rasgos se obtuvieron resultados tales como: un algoritmo escrito en el lenguaje de programación Java, que genera composiciones musicales basadas en las técnicas musicales gramaticales y sintácticas propuestas por Johann Sebastian Bach, generando sonidos sintetizados a partir del piano donde el computador se convirtió en un instrumento y a su vez en un compositor simulando la capacidad de composición humana.

Teniendo en cuenta estos ítems obtenidos y de la generación propia de composiciones hechas por el algoritmo, hay tres productos finales para desglosar y mostrar.

### 5.1 Compases compuestos por notas musicales

El algoritmo genera, a partir de la implementación de cromosomas y funciones evaluadoras, cadenas de notas que cumplen con las características básicas y que se agrupan para componer compases.

En el siguiente análisis cuantitativo se muestra la evolución y convergencia del algoritmo implementado con diferentes escenarios de búsqueda, población y criterios de finalización. La población se evalúa después de cada generación, donde se revisan los contadores de adaptación relativa y puntuación acumulada de los individuos de la población. Así mismo, se calcula la adaptación global de la población y la posición del mejor individuo. El cromosoma final formado por los individuos más adaptados, en este caso puntuados por el usuario después de escucharlos, es el que se toma como base para la muestra y se relacionan en la tabla 1.

**Tabla 1.** Comparación individuos población, puntaje máximo y mínimo.

Tiempo	Tamaño	Puntaje Máximo	Puntaje Mínimo	Población Inicial
291612	80	70	1	50
311511	80	70	2	60
435671	80	60	4	70
491634	80	60	4	80
561711	80	70	4	90
594321	80	70	4	100

Fuente: elaboración propia

**Tabla 2.** Comparación individuos población, puntaje máximo y mínimo.

Tiempo	Tamaño	Puntaje Máximo	Puntaje Mínimo	Población Inicial
872345	160	70	1	150
932344	160	70	4	160
1298977	160	60	4	170
1476655	160	60	4	180
1687766	160	70	5	190
1778899	160	70	5	200

Fuente: elaboración propia

En la tabla 2 se muestra una comparación de los resultados obtenidos, después de correr 12 veces el algoritmo, donde hay clara evidencia que, alterando significativamente la población inicial, se obtienen resultados más satisfactorios, la muestra clara es el puntaje mínimo ya que al crecer la población los resultados generados son más agradables para el usuario final.

La sintaxis con la que se muestran los resultados de las evoluciones del algoritmo, cumple los estándares de la escritura musical y son comprensibles, este era uno de los aspectos que se propuso como objetivo ya que hará que el proyecto tenga escalabilidad y los resultados sean entendibles para el usuario final, como se presenta en las figuras 7,8 y 9.

Melody # 1

Genetic Algorithm

**Figura 7.** Fragmento de partitura final compuesta por notas generadas por el algoritmo, configurado con 30 iteraciones, una población inicial de 50 individuos y un tiempo de ejecución estimado de 218967 ms.

Fuente: elaboración propia

Tocata

Genetic Algorithm

**Figura 8.** Fragmento de partitura donde se incluyen notas en clave de sol y en clave de fa, configurado con 30 iteraciones, una población inicial de 50 individuos y un tiempo de ejecución estimado de 293456 ms.

Fuente: elaboración propia

Sonata

Genetic Algorithm

**Figura 9.** Fragmento de partitura donde se incluyen notas en clave de sol y en clave de fa, configurado con 30 iteraciones, una población inicial de 50 individuos y un tiempo de ejecución estimado de 415571 ms.

Fuente: elaboración propia

Todos los fragmentos producidos por el algoritmo genético pueden ser escuchados ingresando a la dirección electrónica disponible en: <http://cmag.wikispaces.com>.

## 6. CONCLUSIONES

Después de haber realizado el proceso de desarrollo anteriormente descrito y tomando en consideración los resultados obtenidos se pueden extraer las siguientes conclusiones:

El proceso de composición musical visto desde la perspectiva de la computación evolutiva demues-

tra ser una propuesta prometedora, dado que se pueden obtener fragmentos musicales semejantes a los producidos por humanos con un buen conocimiento musical. Sin embargo, es importante indicar que se deben plantear mecanismos de evaluación automática del fitness, considerando no solo elementos matemáticos como los desarrollados por los autores presentados en el artículo, sino, tomando también en consideración reglas heurísticas utilizadas por personas experimentadas en el campo musical que ayudan a penalizar fragmentos que no son agradables al oído.

Tomando en consideración, los comentarios de los músicos que interpretaron la partitura final,

se puede concluir que el algoritmo satisface las técnicas de escritura musical y se obtiene un ahorro de tiempo considerable al producir una obra musical. Que la partitura es de fácil comprensión para los intérpretes, esto a raíz de la experiencia con una persona ajena al proyecto que interpretó la melodía producida sin ningún problema.

En el diseño del algoritmo no se tuvo en cuenta la figura musical B Cuadro esto puede ocasionar que, en la escala natural, cuando hay un paso de generación, aleatoriamente en el operador de cruce se puede generar, por ejemplo un D sostenido, lo que indica que de ahí en adelante todos los D deben ir sostenidos, hasta que haya la aparición de un B Cuadro, esto lo que ocasiona es: que los acordes o notas generadas se salgan de la octava natural del piano y por tanto puede causar un confusión en la interpretación.

La interpretación y composición de compases a partir de notas musicales y la generación de archivos MIDI son resultados esperados y son un incentivo para seguir trabajando en el tema metaheurístico para la composición musical, hay muchas otras técnicas que pueden ser aplicadas e implementadas al proyecto, como redes neuronales, computación evolutiva (programación ge-

nética, programación genética gramatical) y PSO (Particle Swarm Optimization) con los cuales se podrían dar diferentes tipos de soluciones a la misma problemática y de esta forma mejorar los resultados obtenidos en este acercamiento.

## 7. TRABAJO FUTURO

Se está trabajando un una función fitness que tenga la capacidad de hacer la evaluación completa de la obra generada por el algoritmo, los criterios que se establecen para ella están direccionados desde el punto de vista musical para, de esta forma, generar funciones de evaluación basadas en penalizaciones dado el incumplimiento de los criterios musicales establecidos. En una futura versión se implementara una o varias de estas técnicas para la evaluación de los compases generados por el algoritmo, ya que el Framework ya contiene estos operadores de selección de esta forma la evaluación será automática y aleatoria.

## 8. FINANCIAMIENTO

Financiamiento y aval, Fundación Universitaria San Martín Facultad de Ingeniería.

---

## REFERENCIAS

- [1] J. Chun-Chi, R. Miiikkulainen, "Creating Melodies with Evolving Recurrent Neural Networks", *International Joint Conference on Neural Networks*, Vol. 3, pp. 2241-2246, agosto, 2001.
- [2] G. Reis, N. Fonseca, F. Fernandez, A. Ferreira, "A Genetic Algorithm Approach with Harmonic Structure Evolution for Polyphonic Music Transcription" *IEEE International Symposium on Signal Processing and Information Technology*, pp. 491-496, diciembre, 2008.
- [3] A. Pazos, A. Santos Del Riego, A. Dorado, J. Romero, "Genetic Music Composer", *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress*, Vol. 2, pp. 1-6, 2011.
- [4] A. Rodríguez, *La Dimensión sonora del lenguaje audio visual*. España: Paidós, 2009, pp. 1-100.
- [5] S. Torres y F. Javier, *Tecnologías en la composición de bandas sonoras*. España:

- Fundación Autor - Sociedad General de Autores y Editores, 2011, pp. 23-70.
- [6] R. de Cande, *Invitación a la música*. México: Alianza, 2008, pp. 10 - 50.
- [7] L. Araujo, C. Cervigón, *Algoritmos evolutivos un enfoque práctico*. Bogotá: Alfaomega, 2009, pp. 1-88.
- [8] D. Koelle, *the Complete Guide to JFugue*. Estados Unidos: Independiente, 2010, pp. 1-118.
- [9] G. Kovács y Z. Pásztor, *Ejercicios preparatorios para instrumentistas*. Madrid: Grao, 2010, pp. 1 - 100.
- [10] L, Lozano, A. Medaglia y N. Velasco, “Generation of Pop-Rock Chord Sequences Using Genetic Algorithms and Variable Neighborhood Search”, *Centro de Optimización y Probabilidad Aplicada Universidad de los Andes*, pp. 573-578, agosto, 2009.