

# *Generación de nuevas melodías con el uso de algoritmos genéticos*

|  |  |  |
|--|--|--|
| De Gracia González, Jesús Aquiles                      | Ortega Coronado, Yeny                                  | Obaldia Ruiz, Kaiser Javier                            |
| Facultad de Informática, Electrónica y Comunicaciones. | Facultad de Informática, Electrónica y Comunicaciones. | Facultad de Informática, Electrónica y Comunicaciones. |
| Escuela de Informática.                                | Escuela de Informática.                                | Escuela de Informática.                                |
| Ciudad de Panamá, Panamá                               | Ciudad de Panamá, Panamá                               | Ciudad de Panamá, Panamá                               |
| jaquideg08@gmail.com                                   | yennycoronado199727@gmail.com                          | thekaiser26@gmail.com                                  |

**Abstract—** Music students are looking for some kind of system that allows them to compose new melodies based on a classical music piano chord dataset. For this work, artificial intelligence will be developed, taking as a model the genetic algorithm to be able to create new melodies automatically, applying its different characteristics (population, fitness calculation, mating pool, parents selection, crossover, mutating) in the sets of mentioned data in MIDI format. Obtaining as a result the generation of new music based on the selected chord files.

## **RESUMEN**

Estudiantes de música están buscando algún tipo de sistema que les permita componer nuevas melodías utilizando como base un dataset de acordes de piano de música clásica. Para este trabajo, se estará desarrollando una inteligencia artificial, tomando como modelo el algoritmo genético para poder crear nuevas melodías de manera automática, aplicando sus diferentes características (population, fitness calculation, mating pool, parents selection, crossover, mutating, radiation) en los conjuntos de datos mencionados en formato MIDI. Obteniendo como resultado la generación de nuevas melodías en base a los archivos de acordes seleccionados.

## **PALABRAS CLAVES**

Algoritmo genético, crossover, mutación, población, función de optimización, Formato MIDI.

## **I - INTRODUCCIÓN**

La música es una de las expresiones más antiguas de la humanidad, la cual desempeña un papel importante. crear música y arte en general es una forma de expresar sensaciones, sentimientos, ideas y otras emociones que

solo el ser humano es capaz de entender. Pero hoy día, con el avance de la tecnología y metodologías de aprendizaje automático, la creación de la música no solo podría ser del hombre.

En el presente trabajo se explicará cómo se aplicó el algoritmo genético que produce nuevas melodías simulando capacidad de composiciones humanas.

## **II - OBJETIVO**

Desarrollar un programa basado en algoritmo genético que logre generar melodías, basadas en un dataset establecido. Permitiendo utilizar las salidas para inspiración de nuevos artistas. Logrando así encontrar una melodía adecuada que se pueda pulir para satisfacer las características de la música clásica.

## **III - MATERIALES Y MÉTODOS**

### **A. Dataset**

El conjunto de datos utilizado en el estudio se obtuvo de archivos de música clásica MIDI.

### **B. Métodos**

**Algoritmos Genéticos:** son una familia de modelos computacionales basados en la evolución, estos codifican una posible solución a un problema específico en una simple estructura cromosómica de datos y aplica mecanismos de recombinación a las estructuras para conservar la información relevante. Los algoritmos genéticos de información los podemos ver como optimizadores de funciones, aunque el rango de problemas a los que se han aplicado algoritmos genéticos es bastante amplio.

Aprovechando el potencial del algoritmo genético utilizamos las siguientes características:

1. **Generación de la población inicial:** la población será un subconjunto de soluciones de la generación actual. La población inicial, se

convertiría en la primera generación, creada generalmente de forma aleatoria. Después de una serie de operaciones se continuaría generando nuevas generaciones de población.

2. **Nueva Generación:** Es un proceso de selección, aplicado en la generación de nuevas poblaciones, en donde se toman los grupos de padres en base a la función de optimización para cada individuo de la población, de manera que los individuos más aptos puedan pasar hacia las generaciones posteriores.
3. **Cruce de padres (crossover):** el cruce se realiza entre 2 padres de la población de padres (la selección de los padres se realizan generalmente de forma aleatoria para que un padre pueda producir varios hijos) parte de la genética de los padres se le pasarían a sus hijos.
4. **Mutación:** Es la probabilidad de que un gen en un cromosoma mute al azar.
5. **Radiación:** Proceso donde se reestructura la generación cuando llegan varias a un estado de estancamiento.

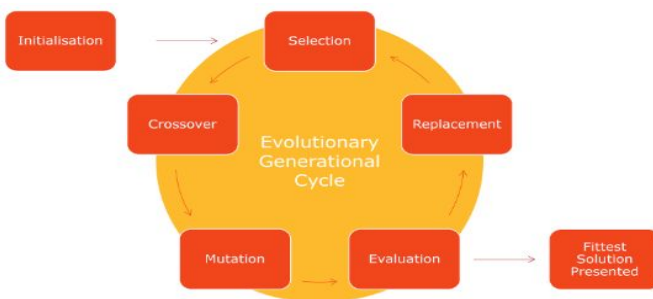


Imagen 1: proceso del algoritmo genético

### C. Librerías y Extensiones

**Módulo Midi - Pygame:** Este módulo permite enviar salidas midi y obtener entrada midi. También puede enlistar los midi en el sistema. El módulo midi admite trabajar con midi reales, virtuales, desde csv y midi txt.

**Midiutil:** es una librería de Python que permite la escritura de archivos de interfaz digital de instrumentos musicales (MIDI) para su manipulación en Python.

**Textwrap:** se utiliza para ajustar y utilizar texto sin formato. Lo que proporciona lectura de texto ajustando los saltos de línea en el párrafo de entrada y manipulación de transcripción.

## IV - Metodologías

1. Lo primero que hacemos es establecer el

manejo de los archivos midi usados para generar la nueva melodía. Pasamos la data a audio con ayuda de la librería **midiutil** y probamos administramos esos audios con **pygame mixer**.

2. Definimos las constantes que serán las encargadas de generar la melodía inicial. Utilizamos el tempo, el rango de notas, y los parámetros de los individuos y mutaciones del algoritmo genético.
3. Creamos la función para leer la carpeta donde tenemos nuestros audios midi que usaremos para generar melodías con **joinpath** y **randomfiles**.
4. Generamos el primer individuo que se encargará de realizar la primera frase musical, y lo regulamos con frecuencias para evitar vacíos y notas muy agudas, siguiendo los rangos de música clásica en piano.
5. Creamos el midi donde se almacenará los resultados de la población a generar.
6. Generamos una población inicial y nos encargamos de que se utilicen los parámetros establecidos al inicio del código. Lo que dará como resultado una puntuación máxima de generación.
7. Luego contemplamos los resultados individuales de una nueva generación y lo regulamos con la anterior, el resultado se guarda en una matriz de clasificación.
8. Mutamos los mejores resultados individuales, generamos nuevos individuos
9. Podemos mutar una frase musical tantas veces como hayamos especificado en mutaciones. Cada mutación elige una nota aleatoria y le da un nuevo tono y una nueva duración aleatoria.
10. Los resultados de frases musicales utilizando las características. descubrimos que tan agradable suenan las notas juntas, y damos la opción si el usuario se queda con esa generación o si quiere seguir buscando nuevas tonalidades.
11. Repetimos los pasos anteriores y el programa guardará el archivo midi de la generación que obtenga la puntuación más alta y no guardará otra hasta que aparezca una mejor puntuación hasta que llegue a 100 o el usuario decida pararlo.
12. En caso de que se generen 2000 generaciones con la misma puntuación alta se hará una radiación masiva que provoca una mayor probabilidad de mutación para el estancamiento
13. El usuario tendrá la opción de salir después de

cada radiación o esperar llegar a 100.

14. Por último en la carpeta “Corrida Actual” estarán todos los resultados por puntuación máxima única y podrá elegir cualquiera de las melodías para inspiración.

## V - Análisis de Resultados

### I. Primera Generación:

La primera generación siempre da como resultado un sonido muy desbalanceado, mezcla de notas altas y bajas que no dan mucha inspiración, pero es funcional.

En la **imagen2** podemos observar los parámetros de la generación y la puntuación inicial alta que es de 17 que vemos que es muy pobre para lo que algoritmo decide que es una buena melodía.

La **imagen3** por su parte muestra el archivo generado midi que es la que los estudiantes podrán escuchar y decidir si quieren escuchar más tonalidades en una nueva generación o salirse.

```
-----:
:G U A R D A N D O-- Corrida-Actual/0.mid:
-----:
Generación: 0
Puntuación máxima actual: 17
Puntaje más bajo: -45
Media de puntuaciones: -13.78
Calificaciones: [-45, -44, -42, -39, -38, -36, -33, -30, -29, -29, -28, -27, -16, -16, -16, -15, -15, -14, -14, -14, -14, -14, -14, -13, -13, -13, -5, -5, -5, -4, -3, -3, -3, -2, -2, -1, 0, 0, 0, 0, 0, 1, 4, 9, 10, 10, 17]
Ejecuciones de puntuación coherentes: 0
Tamaño de la población: 100
Fuerza de las mutaciones: 8
Nueva población aleatoria / mutada: 0.25/0.55
Deseas buscar una mejor puntuación de melodía? (y/n)
:
```

Imagen 2: Generación 0

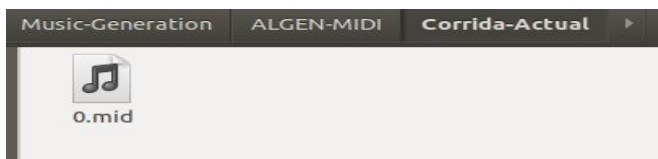


Imagen 3: Carpeta contenedora inicia

### II. Comparaciones iniciales:

Luego de seguir probando mejorar las melodías pudimos observar que las primeras generaciones son muy desiguales, las notas son muy variadas lo que es bueno para los estudiantes ya que podrán elegir diferentes interpretaciones de una melodía en diferentes tonos y elegir cualquiera para sus músicas

### II. Comparaciones Finales:

En la siguiente **imagen4** tenemos una corrida que llegó al 98 de puntuación máxima en la generación 9050010 donde pasó por muchos procesos de radiación. Lo que nos ayudó a conocer que entre más cercano a lo óptimo está la melodía más parecidas son, debido a que busca cumplir lo establecido en los parámetros del algoritmo genético.



imagen 4: generación más alta registrada

## VI - Conclusiones

Como resultado final pudimos obtener el ideal deseado, generamos melodías que sirven de inspiración, y además pudimos estudiar las variaciones de una misma melodía y así acaparar más gustos melódicos en los estudiantes.

Elegimos el algoritmo genético porque no solo nos ayuda a generar melodías, sino que nos permite crear nuevas tonalidades más complejas utilizando piezas reales como patrón de entrenamiento, ya sean aplicadas aleatorio mente o siguiendo pasos determinados. Sin embargo, las piezas generadas por radiación tienden a ser repetitivas lo que causa generaciones innecesarias. Pero en general, las guardadas son muy diferentes.

## VI - Bibliografías

Generación y Evaluación de Secuencias Melódicas Mediante Inteligencia Artificial. (s.f.). Obtenido de [http://oa.upm.es/53396/1/TFG\\_SILVIA\\_JIMENEZ\\_GOMEZ.pdf](http://oa.upm.es/53396/1/TFG_SILVIA_JIMENEZ_GOMEZ.pdf)

Santiago Perez Toledo, F. M. (s.f.). Proyecto de grado . Obtenido de [https://www.fing.edu.uy/inco/grupos/pln/prygrado/informe\\_pgcumbiacheta.pdf](https://www.fing.edu.uy/inco/grupos/pln/prygrado/informe_pgcumbiacheta.pdf)

Machuca, M. H. (s.f.). Corsera . Obtenido de <https://es.coursera.org/lecture/composicion-algoritmica-supercollider/sonificacion-musificacion-de-datos-PCveI>

Caparrini, F. S. (4 de 11 de 2019). Fernando Sancho Caparrini. Obtenido de <http://www.cs.us.es/~fsancho/?e=65>

Dr. Juan Carlos Seck Tuoh Mora, D. N. (s.f.). Centro de Cómputo Académico. Obtenido de [http://ceca.uaeh.edu.mx/algoritmos\\_geneticos/](http://ceca.uaeh.edu.mx/algoritmos_geneticos/)