



Universidad Central de Venezuela

Facultad de Ciencias

Escuela de Computación

# **SISTEMA SEMI-INTERACTIVO PARA LA EVOLUCIÓN DE IDEAS MELÓDICAS**

Br. Jorge Alejandro Fuentes Figueroa

Héctor Navarro, Tutor

Caracas, 13 de mayo del año 2015





Universidad Central de Venezuela

Facultad de Ciencias

Escuela de Computación

# **SISTEMA SEMI-INTERACTIVO PARA LA EVOLUCIÓN DE IDEAS MELÓDICAS**

Br. Jorge Alejandro Fuentes Figueroa

Héctor Navarro, Tutor

Caracas, 13 de mayo del año 2015

**SISTEMA SEMI-INTERACTIVO PARA LA EVOLUCIÓN DE IDEAS  
MELÓDICAS**

Br. Jorge Alejandro Fuentes Figueroa

*Trabajo Especial de Grado presentado  
ante la ilustre Universidad Central de Venezuela  
como requisito parcial para optar al título de  
**Licenciado en Computación.***

---

Héctor Navarro, Tutor

# ACTA DEL VEREDICTO

Quienes suscriben, miembros del Jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado presentado por el **Br. Jorge Alejandro Fuentes Figueroa**, titulado: “**Sistema Semi-Interactivo para la Evolución de Ideas Melódicas**” para optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído como fue, dicho trabajo por cada uno de los miembros del Jurado, se fijó el día 8 de Mayo de 2015 a las 14:00 horas, para que su autor lo defendiera en forma pública, lo que se hizo en el Centro de Computación Gráfica de la Escuela de Computación, en la Facultad de Ciencias de la Universidad Central de Venezuela, mediante una exposición oral de su contenido, luego de lo cual, respondió las preguntas formuladas por el jurado y público en general. Finalizada la defensa pública del Trabajo Espacial de Grado, el jurado decidió aprobarlo.

En fe de lo cual se levanta la presente acta, en Caracas a los ocho días del mes de mayo de dos mil quince, dejándose también constancia de que actuó como Coordinador del jurado el Profesor Tutor Héctor Navarro.

---

Héctor Navarro, Tutor

---

Walter Hernández

---

Esmitt Ramirez

*Para mi madre y mi hermana.*

*Mi mejor equipo.*

# Agradecimientos

Comienzo por agradecer a mi madre Rosa Aura por su constancia, dedicación, apoyo y amor incondicional. Sin su esfuerzo nunca me hubiese podido convertir en la persona que soy hoy. Gracias Ma.

A mi hermana Carmen que con sus consejos y su ayuda he logrado construir y reconstruir mi camino en múltiples ocasiones.

A mi novia Marjorie, que con su apoyo y perseverancia, siempre me muestra que puedo ser más y mejor.

A los 5, gracias a ustedes principalmente por estar conmigo hasta el final.

A mis amigos por enseñarme a ser mejor día tras día, y a todo el que me recuerda diariamente realmente quien soy y lo que puedo lograr gracias, porque no tengo muy buena memoria con eso por lo general.

A la Universidad Simón Bolívar, por mostrarme que la excelencia es el camino al éxito, y a la Universidad Central de Venezuela, por enseñarme como alcanzarla.

A mi tutor, Héctor Navarro, por asistirme, guiarme y tenerme toda la paciencia necesaria durante el desarrollo de este trabajo.

Finalmente, quiero agradecer a mi familia por estar a mi lado en todo momento y brindarme su apoyo hasta en los momentos más difíciles. Especialmente a mi tío Fernando, que me transmitió siempre el amor por la música y me ayudo a comprenderla mejor para poder llevar a cabo este trabajo.

Por ultimo, al 2013 por ser el peor y el mejor año hasta los momentos.

## Resumen

La demanda de contenido musical es alta actualmente, gracias a la gran popularidad de los medios de comunicación audiovisuales y multimedia. El presente trabajo especial de grado propone la utilización de algoritmos genéticos para la generación y mejora evolutiva de melodías, con el fin de generar contenido musical base. Este sistema está constituido por tres módulos: un módulo “generador de melodías aleatorias”, un módulo de “Reproducción musical” de las melodías generadas por el algoritmo, y un módulo “evolutivo”, que mejore las melodías generadas. Esta solución permite al usuario explorar en un espectro de búsqueda determinado, mejorando las melodías a su conveniencia hasta conseguir una de su agrado. Los resultados obtenidos revelan que la herramienta crea individuos utilizables y adaptables a las necesidades del usuario final.

# Índice General

<b>Resumen</b>	<b>vi</b>
<b>Índice General</b>	<b>vii</b>
<b>1. Problema de Investigación</b>	<b>1</b>
1.1. Planteamiento del problema . . . . .	1
1.2. Alcance . . . . .	4
1.3. Solución . . . . .	6
1.4. Objetivo general . . . . .	6
1.5. Objetivos específicos . . . . .	6
<b>2. Algoritmos Genéticos</b>	<b>8</b>
2.1. Acerca de la definición de Algoritmos Genéticos . . . . .	8
2.1.1. Antecedentes . . . . .	9
2.2. Estructura de un Algoritmo Genético . . . . .	11
2.2.1. Representación del Cromosoma . . . . .	12
2.2.2. Función de Aptitud . . . . .	14
2.2.3. Proceso de Selección . . . . .	17
2.2.4. Operadores Genéticos . . . . .	20
2.3. Usos y aplicaciones . . . . .	23
<b>3. La Música</b>	<b>25</b>
3.1. Algunos conceptos musicales básicos . . . . .	25
3.1.1. Música y Sonido . . . . .	26
3.1.2. Cualidades Principales del Sonido . . . . .	27

3.1.3.	Conceptos Musicales Varios . . . . .	29
3.2.	Criterios usados para la composición y evaluación musical . . . . .	36
3.2.1.	Consonancia y Disonancia . . . . .	36
3.2.2.	Tonalidad Musical . . . . .	38
3.2.3.	Escalas Musicales . . . . .	38
3.2.4.	Diferencias y relaciones entre tonalidad musical y escalas . . . . .	40
3.3.	Musical Instrument Digital Interface (MIDI) . . . . .	40
<b>4.</b>	<b>Algoritmos Genéticos Musicales</b>	<b>42</b>
4.1.	Convergencia audible para la extensión óptima de melodías base con evaluación estadística de distancia interválica específica del género, Ronald Hochreiter[18] . . . . .	42
4.2.	Un algoritmo genético para componer música, Dragan Matic [6] . . . . .	44
4.3.	GenJam: un algoritmo genético para generar solos de jazz, John A. Biles [8] . . . . .	49
<b>5.</b>	<b>Desarrollo e Implementación</b>	<b>54</b>
5.1.	Conceptualización y estructuración de la solución . . . . .	54
5.1.1.	Metodología Planteada . . . . .	54
5.1.2.	Conceptualización . . . . .	55
5.1.3.	Implementación y estructura . . . . .	56
5.1.4.	Diagrama de Clases y Formularios . . . . .	60
5.2.	Módulo “Generador” . . . . .	60
5.2.1.	Implementación . . . . .	61
5.3.	Módulo “Evolutivo” . . . . .	62
5.3.1.	Estructura del Algoritmo Genético . . . . .	62
5.3.2.	Representación . . . . .	63
5.3.3.	Aptitud . . . . .	66
5.3.4.	Selección . . . . .	69
5.3.5.	Operadores Genéticos . . . . .	70
5.3.6.	Reemplazo . . . . .	72
5.4.	Módulo “Reproductor Musical” . . . . .	73

<i>Índice General</i>	ix
5.4.1.    C# Midi Toolkit . . . . .	73
5.4.2.    Implementación . . . . .	74
<b>6. Resultados</b>	<b>75</b>
6.1.    Ambiente de Trabajo . . . . .	75
6.2.    Módulo de pruebas . . . . .	76
6.2.1.    Creador de casos de prueba . . . . .	76
6.2.2.    Pantalla de pruebas de usuario . . . . .	77
6.3.    Resultados cuantitativos . . . . .	78
6.4.    Resultados cualitativos . . . . .	79
6.5.    Resultados musicales . . . . .	81
<b>7. Conclusiones y trabajos futuros</b>	<b>84</b>
7.1.    Conclusiones . . . . .	84
7.2.    Trabajos futuros y recomendaciones . . . . .	85
<b>Bibliografía</b>	<b>87</b>

## Capítulo 1

# Problema de Investigación

Este capítulo tiene como objetivo presentar el contexto de la situación sobre la cual se realizó la investigación, así como también las alternativas de abordaje y soluciones propuestas. Se presenta también el alcance del sistema, el objetivo general y los objetivos específicos que sirvieron de guía para la realización de la presente investigación.

### 1.1. Planteamiento del problema

El presente trabajo se propone unir a través de la investigación y la exploración, dos variables que han sido estudiadas de manera individual a profundidad: los algoritmos genéticos y la música, pero que al ser unidas, representan un nuevo universo de estudio amplio e interesante, que pese a haber sido estudiado en varias ocasiones a lo largo de los últimos años, contiene detalles que recién comienza a ser explorados.

Grandes descubrimientos científicos y tecnológicos permiten el intercambio y complementación de información entre campos del saber, (por ejemplo teléfonos inteligentes, el internet, las redes mundiales de información, entre otros). El modo de comunicarse también ha cambiado: el Internet ha facilitado esto en gran medida. Gracias a las redes sociales, la facilidad de comunicar e informar cada día es mayor. El mundo se ha convertido en una aldea global en la que viven millones de seres humanos conectados

por una gran red de información. Cada día nuevos proyectos son creados en base a estas interacciones humanas, lo que requiere de una cantidad de recursos y material interactivo exponencialmente creciente.

En este sentido, resulta interesante el planteamiento de utilizar modelos evolutivos aplicados a un área compleja y profesional como lo es la música, enfocándose específicamente en el área de creación musical o en el mejoramiento de composiciones musicales ya existentes, sobre la base biológica de la evolución. El divulgador científico Philip Ball [1] señala al respecto:

*“..La mayoría de compositores musicales prefiere creer que sus creaciones son fruto de una mezcla de intuición e inspiración, cuando en realidad las melodías que se les ocurren son amalgamas y mutaciones de otras que ya han oído.”*

De esta manera, siendo la música una de las artes con más formación matemática en sus cimientos, es posible que existan tanta variedad de combinaciones que pueden llegar a ser musicalmente agradables como las que podría ofrecer, desde la biología, el proceso de evolución desde individuos menos aptos a individuos bien adaptados.

Como lo señalan Fernández y Vico [2]

*“Resulta discutible el que un sistema computacional pueda ser realmente creativo, sin embargo, el desarrollo de sistemas de composición algorítmica capaces de exponer creatividad independiente, expande ampliamente el proceso de composición musical, y por consiguiente el mercado musical. Esto no debe ser visto como un caso más de los ordenadores reemplazan los seres humanos, o una sustitución potencialmente radical en la forma en que los compositores realizan su trabajo, más bien, debe ser visto como un sistema experto pedagógico, que no reemplaza el papel de los profesionales humanos, sino que les ayuda y otorga nuevas formas de hacer su trabajo.”*

Ahora bien, ¿Para qué es necesario producir música computacionalmente? ¿Qué razones justificarían esta combinación de elementos?

En primer lugar, es necesario programar, idear, crear y producir recursos para una audiencia local y mundial cada vez más exigente, interactiva, globalizada e interconectada, todo ello representa un enorme reto.

En un segundo plano, esta nueva realidad respecto al uso creciente de la tecnología, genera espacios de aprendizaje y difusión cada vez más diversos, como por ejemplo, el desarrollo de videojuegos, la producción de recursos audiovisuales (películas, videos, cortos), entre otros campos, lo que abre las puertas al aprendizaje, investigación y creación independientes.

Todo lo antes expuesto, ahonda la demanda creciente en los potenciales consumidores, usuarios o desarrolladores que necesitan crear o dar uso a ideas musicales o melódicas.

Por ejemplo, para la realización de un proyecto simple (videos, cortos, cine, videojuegos, entre otros) es necesario contar con cierta cantidad de competencias, habilidades y recursos, que pudieran estar ausentes en el equipo de trabajo. En general, se busca enriquecer al equipo agregando a personas que tengan las habilidades que se requieren. Pero existen casos en los que esto resulta difícil, bien sea por falta de capital, mano de obra, espacio, entre otras razones. En este tipo de ocasiones podría resultar útil este tipo de aplicación.

Por otro lado, la originalidad y diversidad del material generado, podría a la vez ser usado por los mismos músicos, compositores, y profesionales, como punto de partida para nuevas obras, o como segmentos de ellas, incluso como inspiración.

Finalmente este programa enriquece las herramientas musicales de las que disponen los desarrolladores y creativos, al ser necesario musicalizar diversos momentos de la producción de películas, cortos o videojuegos (apertura, créditos, menú, historia, las transiciones, efectos, entre otros).

Como es posible apreciar, existen múltiples ventajas y oportunidades que esta propuesta investigativa puede ofrecer a futuros emprendedores, más deben ser vistas paso

a paso, a manera de considerar también los múltiples obstáculos, errores o dificultades que la investigación pueda acarrear en sí misma, ofreciendo así también, oportunidades para que sea continuada y ampliada. En cuanto al ámbito musical, debido a su complejidad de conceptos y géneros, esta iniciativa de generación de contenido no ha sido muy explorada, razón por la cual la presente investigación desea ser un pequeño aporte exploratorio.

Una vez expuestas las razones que motivan el presente estudio, se presentan algunas de las interrogantes que guiarán la investigación:

- ¿Se pueden escoger parámetros básicos que hacen que un algoritmo genético sea capaz de crear música?
- ¿A través de que procedimientos se puede ajustar el AG para permitir la participación del usuario, en el proceso de evolución de composiciones musicales básicas?
- ¿Cómo se puede crear un AG en base a parámetros musicales melódicos definidos?
- ¿Es posible escuchar en el programa una composición producida por el algoritmo?
- ¿Qué limitaciones y oportunidades se desprenden de esta investigación?

## **1.2. Alcance**

Al trabajar en áreas de investigación como la música por un lado y los algoritmos evolutivos por el otro, es necesario admitir la abundancia de variables y parámetros a tomar en cuenta, profundidad de los planteamientos, y en algunos casos subjetividad de conceptos.

Por tal motivo, se hace imprescindible delimitar el área de extensión de la investigación con la finalidad de precisar los aportes que se vayan alcanzando en cada

prototipo. En tal sentido se propone ciertas limitaciones que conforman el alcance de la investigación:

La solución o programa generado deberá permitir al usuario inicializar una población de melodías reproducibles y generadas aleatoriamente. Durante este proceso el programa permitirá la configuración de parámetros básicos de inicialización de un Algoritmo genético como lo son el tamaño de la población, o las probabilidades de cruce y mutación. También se podrá configurar otras opciones relacionadas a las funciones de aptitud musical establecida.

Se realizara un desarrollo de prototipos de la aplicación con pruebas de distintos esquemas de reemplazo de la población, estrategias de selección y operadores genéticos, para determinar cuál es la combinación que logre permitir al usuario explorar diversas melodías, o si así lo prefiere, evolucionar una melodía específica.

El programa también permitirá al usuario marcar una o varias melodías como “agradable o preferida”, condicionando al algoritmo para que genere más melodías parecidas a estas en la siguiente generación. El usuario también podrá en todo momento reproducir y ver una representación visual del cromosoma que se encuentre seleccionado en ese momento.

Se desarrollara también una opción que permita obtener varias composiciones producidas por varias corridas sucesivas del algoritmo y que con estas canciones se pueda realizar una prueba para contrastar la función de aptitud escogida contra el gusto de varios usuarios.

En resumen, de acuerdo a los objetivos propuestos, el alcance de la presente investigación se limita a ser un pequeño aporte o ayuda exploratoria e interactiva, en el vasto universo de la creación de ideas melódicas a través de algoritmos evolutivos, un punto de inicio, una semilla que podría germinar.

### 1.3. Solución

Se propuso la construcción de un programa, que cree ideas melódicas cortas referentes a una escala musical específica, y que haciendo uso de un algoritmo genético con una función de aptitud multi-objetivos, y operadores genéticos musicales, evolucione dichas melodías y obtenga resultados musicalmente aceptables.

Se propone también que dicho sistema permita la selección preferencial del usuario y muestre los resultados de manera visual y audible, logrando con esto que el usuario interactúe con la aplicación hasta conseguir resultados que sean más de su agrado.

### 1.4. Objetivo general

Construir un sistema evolutivo aplicando un algoritmo genético semi-interactivo, que permite evolucionar ideas melódicas básicas, generadas aleatoriamente.

### 1.5. Objetivos específicos

- Definir e implementar los aspectos básicos del algoritmo genético:
  - Representación del cromosoma (idea melódica): representación de la información genética.
  - Método de selección: selección por torneo o ruleta.
  - Función de aptitud y método de evaluación: función multi-objetivo basada en la evaluación interválica de consonancia y disonancia, varianza de tonos y/o evaluación semi-interactiva.
  - Operadores genéticos: Cruces simples o multipuntos, y mutaciones no dependientes del contexto musical.

- Crear un método para generar individuos procedimentalmente para ser usado como estrategia de generación de la población inicial.
- Definir los parámetros específicos de configuración del algoritmo genético:
  - Tamaño de la población
  - Cantidad de iteraciones
  - Factores de peso de las diversas funciones de aptitud
  - Escala musical de referencia para la melodía a generar
- Crear un sistema que permita la participación semi-interactiva del usuario.
- Crear en el programa la posibilidad de exportar las melodías en formato MIDI y reproducirlas.
- Precisar las oportunidades y limitaciones de este sistema.

## Capítulo 2

# Algoritmos Genéticos

La “Computación evolutiva” es una rama de la inteligencia artificial inspirada en la teoría de la evolución propuesta por científicos como Charles Darwin en 1859. Comprende a aquellos algoritmos que se basan en el proceso de evolución de las especies y de selección natural, para explorar espacios de búsqueda amplios y resolver problemas de optimización y búsqueda.

La implementación más amplia y extensamente aceptada en esta rama de la inteligencia artificial son los Algoritmos Genéticos. En este capítulo explicaremos todo lo referente a ellos, sus características, conceptos y aspectos más importantes.

### 2.1. Acerca de la definición de Algoritmos Genéticos

En el estado del arte actual del ámbito de la computación evolutiva, no existe una definición exacta sobre lo que son los Algoritmos Genéticos. Cada autor crea su “versión” de estos algoritmos de acuerdo con los requerimientos de su investigación y el área de estudio en la cual lo vayan a aplicar.

Sin embargo, existe un trasfondo histórico en el área de la computación evolutiva que vale la pena exponer, debido a que permite descubrir la existencia de ciertos lineamientos y criterios mínimos para clasificar o evaluar si el algoritmo que está siendo creado entra dentro de la clasificación de Algoritmo Genético o no.

A continuación se hará referencia a las primeras investigaciones así como varias definiciones de diversos estudiosos en el área de la inteligencia artificial con respecto a los algoritmos genéticos su definición y funcionamiento, a manera de tener una visión más amplia de cómo han venido evolucionando a través de los años.

### 2.1.1. Antecedentes

Los algoritmos genéticos fueron inventados por John Holland aproximadamente en los años 60 y fueron desarrollados entre 1960 y 1970 por Holland y sus estudiantes. A diferencia de la investigación que se llevaba a cabo en el ámbito de inteligencia artificial en esa época, el trabajo de Holland [3] tenía como objetivo estudiar a manera de simulación el proceso de adaptación que ocurre en la naturaleza en poblaciones de individuos, y como este proceso puede adaptarse a los sistemas computarizados.

En su libro “Introduction to Genetic Algorithms” Mitchell[4] describe el trabajo de Holland [3] de la siguiente forma:

“El algoritmo genético de Holland es un método para mover de una población de “cromosomas” (ejemplo: cadenas de ceros y unos, o “bits”) a una nueva población usando una especie de “Selección Natural”, junto con operadores de *cruce*, *mutación*, e *inversión* inspirados en la genética. ... El operador de *selección* escoge cuales de esos cromosomas en la población serán capaces de reproducirse, y en promedio, los cromosomas más *aptos* producirán más hijos, que los menos *aptos*.”

Otra definición clásicamente aceptada es la dada por Goldberg [5]:

“los Algoritmos Genéticos son algoritmos de búsqueda basados en la mecánica de selección natural y de la genética natural. Combinan la supervivencia del más apto entre estructuras de secuencias con un intercambio de información estructurado, aunque aleatorizado, para constituir así un algoritmo de búsqueda que tenga algo de las genialidades de las búsquedas humanas”

De estas definiciones se puede observar que inicialmente se manejaba una definición bastante ligada al aspecto biológico del proceso de evolución y selección natural. Sin embargo con el avance de la comprensión tecnológica y de los sistemas automatizados, poco a poco la definición se va tornando más técnica. Aun así, hasta el día de hoy la definición ha guardado relación con algunos conceptos análogos en el área de la biología evolutiva. Esto puede evidenciarse en el trabajo de Matic (2013) [6], en el cual se definen también los Algoritmos genéticos de la siguiente manera:

“Los Algoritmos Genéticos (GA) son algoritmos complejos y adaptativos usualmente usados en resolver problemas robustos de optimización. Básicamente involucran trabajo con poblaciones de individuos en las cuales cada individuo puede representar una solución (óptima), y cada población es un subconjunto del espacio total de búsqueda. Las poblaciones van cambiando en el proceso iterativo (los nuevos individuos reemplazan a los viejos, al ser potencialmente mejores). A cada individuo se le asigna un valor llamado *aptitud*, el cual indica la calidad del mismo. Durante el proceso de iteración los buenos individuos son *seleccionados* para *reproducirse* y con esto reemplazar antiguos individuos con individuos *mejores*, al aplicar operadores de *cruce* y *mutación*.”

En síntesis, una definición bastante completa, simple y explicativa es la que propone Haupt R. [7]:

“El Algoritmo genético (AG) es una técnica de optimización y búsqueda basada en los principios de genética y selección natural. Un AG permite que una población compuesta por muchos individuos evolucione bajo reglas de selección específicas hacia un estado que maximice su aptitud.”

En el presente trabajo se ha decidido adoptar una definición de algoritmo genético amplia, que no limite la usabilidad de estos tan solo a la optimización o búsqueda de individuos, sino también que sea abierto a la exploración del espacio de búsqueda como tal. Esto debido a que el espacio de búsqueda seleccionado para este proyecto es nada menos que el entorno musical y auditivo, en el cual no se está completamente definido en base a que parámetros una composición musical es óptima o no.

En cuanto a esto, se tiene que un Algoritmo Genético, es un proceso iterativo en el cual una población de individuos estructurados de acuerdo a una cierta *Representación*, evoluciona y cambia por medio de ciertos operadores. Cada uno de estos individuos tiene asociado un valor de *aptitud* en base a su representación. En cada iteración se “actualizará” la población. En otras palabras, se elegirá, de acuerdo a una *estrategia de selección* a los individuos más prominentes (más aptos). Se aplicarán entonces *operadores genéticos* varios para generar nuevos individuos y se reemplazarán los individuos generados, por los que sean menos aptos de la población, creando con esto una nueva población.

A continuación se hablará con más detalle de cada una de las características que conforman esta definición, las cuales serán tomadas en cuenta para la realización del presente trabajo de investigación.

## 2.2. Estructura de un Algoritmo Genético

Como se expuso anteriormente, los Algoritmos Genéticos no cuentan con una definición específica y única. Sin embargo, la mayoría de las definiciones desarrolladas a lo largo de los años cuentan con un esquema de funcionamiento bastante similar. Dicho esquema puede ser apreciado en la Figura 2.1

En base a ese esquema y a las definiciones previamente planteadas, puede apreciarse que la mayoría de las implementaciones de Algoritmos Genéticos cuentan con ciertos elementos en común:

- Una representación específica del “Individuo” o “Cromosoma”
- Definición de “población” y sus Características y parámetros
- Una “Función de Aptitud” o evaluación dependiente de la representación
- Un conjunto de “operadores genéticos” que generan variación del material genético de la población y generan nuevos individuos en base a los ya existentes

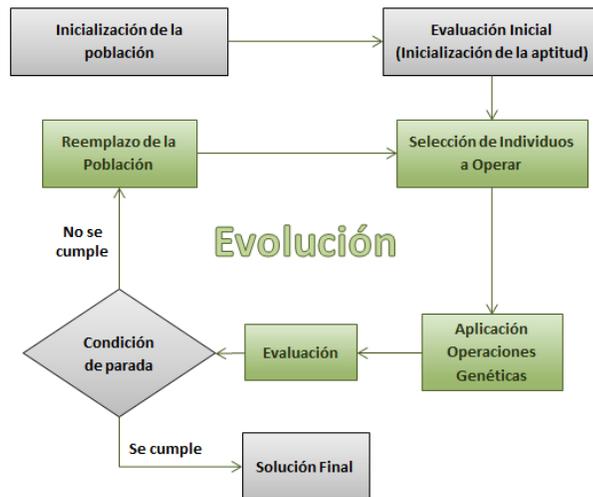


Figura 2.1: Esquema General de un Algoritmo Genético

- Una “Estrategia de Selección” en base a la cual se elijan a los individuos más prominentes

Es muy importante, sin embargo, al momento de utilizar o crear un algoritmo genético, que cada uno de estos sean definidos y estipulados, teniendo en cuenta el espacio de búsqueda, o el problema de investigación al cual se vaya a aplicar el algoritmo. En esta sección hablaremos un poco de cada uno de estos elementos.

### 2.2.1. Representación del Cromosoma

Según Haupt (2004) en su libro “Practical Genetic Algorithms” [7], Los Algoritmos Genéticos (AG) son “una técnica de optimización y búsqueda basada en los principios de genética y selección natural. Un AG permite que una población compuesta por muchos individuos evolucione bajo reglas de selección específicas hacia un estado que maximice su aptitud”. En esta sección se explicarán los detalles acerca de la representación y creación de los individuos de un Algoritmo genético, como tal.

## Representación

Los Algoritmos genéticos al centrarse en la evolución de individuos, requieren de una representación de los individuos, o candidatos de solución al problema que se está estudiando. Dicha representación debe ser hecha de tal manera que logre codificar información varia sobre el dominio estudiado a fin de evaluarla y sacarle provecho seguidamente. El diseñador del Algoritmo genético debe ser capaz de crear una representación lo bastante amplia y no engorrosa, que sea apropiada al problema (usualmente una codificación usada en un método ya existente es una buena solución). La representación del individuo depende entonces del problema resuelto, aun así es posible mencionar algunas formas de representación ampliamente utilizadas:

- Cadenas de caracteres compuestos por números reales, escogidos dependiendo del problema a estudiar
- Cadenas de números, representando valores en bits (unos y ceros)
- Cadenas de bits directamente utilizados
- Expresiones lógicas

Haupt [7] también clasifica a los algoritmos genéticos de acuerdo al tipo de representación utilizada. Haupt define como “Algoritmos Genéticos Binarios”, a aquellos que usan una representación de los individuos a nivel de bits, y define como “Algoritmos Genéticos Continuos” a aquellos que usan tipos de datos continuos (Integer, Floats, Strings codificados semánticamente a nivel de caracteres, entre otros).

Para generar nuevos individuos, los algoritmos genéticos binarios usan operadores que modifican el material a nivel de bit. Haciendo más difícil la evaluación de sus características y rasgos estudiables, pero maximizando la optimización en cuanto a operaciones entre individuos. Por su parte los AG Continuos, usarían operadores muy parecidos a los binarios, solo que aplicarían las operaciones a nivel de datos continuos.

Estos resultan mejores cuando se trata de lidiar con sistemas que necesitan definir variables con valores continuos. En este caso representaciones como las de punto flotante suelen ser más apropiadas.

### Inicialización de la población

El mecanismo de inicialización de la población depende, claramente del problema de estudio y de la representación de los individuos que ha sido definida para el algoritmo. Generalmente se usa un mecanismo de generación aleatoria, para generar individuos que caigan dentro de las reglas de representación de individuos previamente definidas. Sin embargo también existen aplicaciones de algoritmos genéticos que implementan heurísticas con el único prerrequisito de que estas no afecten muy notablemente en el rendimiento [6].

Un caso de estudio bastante particular es el de GENJAM [8] un sistema basado en la Computación evolutiva para generar solos de jazz interactivamente. Este sistema utiliza para su inicialización un sistema basado en cadenas de Markov y otros algoritmos complejos que garantizan la obtención de buenos individuos iniciales. Esto le permite aligerar la carga del algoritmo al momento de evaluar los individuos, y centrar la potencia de ejecución en la creación y mezcla de buenos individuos (tomando en cuenta que también es un algoritmo genético interactivo, lo cual será explicado más adelante). La premisa de Biles [8] es “Dos *buenos* individuos padres al cruzarse y entremezclarse, tenderán a producir *buenos* individuos hijos”. A este tipo de inicialización de le conoce como “Inicialización Heurística” o “inteligente”

#### 2.2.2. Función de Aptitud

Se definirá por aptitud o adecuación a la capacidad de reproducirse que tiene un individuo con ciertas características genéticas. Este concepto es aplicable principalmente al área biológica más que la computacional. En el área de la inteligencia

artificial puede decirse que la aptitud representa a la calidad que tiene un individuo determinado, referida directamente a que tan bien resuelve el problema planteado.

Wiggins y Papadopoulos (1998)[9] dicen que la función de aptitud “provee una medida de la calidad de un cromosoma”. También exponen que “hallar tal función es un problema no-trivial, especialmente en el caso de soluciones multi-objetivo, donde varias medidas de aptitud deben de ser comparadas unas con otras en el proceso de selección y evaluación de un cromosoma”

La función de evaluación, establece una medida numérica de la bondad de un individuo. Normalmente al trabajar con algoritmos genéticos esta función se debe maximizar o minimizar, para luego compararla con un valor objetivo u óptimo. Esto se hace con el fin de encontrar “la mejor solución” al problema planteado, objetivo común de todos los algoritmos de búsqueda y optimización.

### **Función de Aptitud Automatizada**

Es la forma más común de plantear la función de aptitud. Se basa en cálculos matemáticos, y comparaciones con un valor objetivo, para implementar la maximización (o minimización) del valor de la aptitud en los individuos de la población. Usualmente depende mucho de la representación del cromosoma, así como del espacio de búsqueda que se esté utilizando. Suelen tener mucha relación con el campo de la matemática y la optimización de funciones.

Como un ejemplo se exhibe el caso específico de una función de aptitud para un algoritmo genético que busca resolver el rompecabezas del 8-puzzle. En este caso una heurística muy usada para evaluar la calidad de la solución suele ser la distancia acumulada de manhattan para las piezas del puzzle. Esto se hace con la formula:

$$h(s) = \sum_{i=1}^8 (|x_i(s) - \bar{x}_i| + |y_i(s) - \bar{y}_i|) \quad (2.1)$$

en donde  $x_i(s)$  y  $y_i(s)$  son las posiciones de una piza en el estado  $s$ , y  $\bar{x}_i$  y  $\bar{y}_i$  son las posiciones de una pieza en el estado final. Mientras menor sea el valor de esta medida, más cerca está el puzzle de ser resuelto.

Generalmente y para mejorar la calidad del algoritmo, se usa no una sino varias funciones de aptitud en conjunto, ponderadas de acuerdo a su importancia (definida por el creador del AG, o el usuario), generando un valor compuesto llamado “Función de Aptitud Compuesta”. Esto le da flexibilidad y precisión a la búsqueda y permite dar mayor control del usuario sobre la exploración del espacio muestral.

Como puede observarse estas funciones heurísticas poseen una lógica matemática bastante específica en muchos casos, lo cual pese a ser ventajoso al momento de explorar certeramente el espacio de búsqueda, requieren de un gran inversión y esfuerzo en el diseño y creación de las mismas, más que todo en espacios de búsqueda complejos y poco definidos, simétricamente. Un ejemplo de un espacio para el cual puede resultar difícil el uso de este tipo de funciones, puede ser en las artes (que suelen ser espacios con características subjetivas más que objetivas), la escritura (en aspectos semánticos), entre otros.

### **Función de Aptitud Interactiva**

Muchas veces la complejidad del problema a evaluar es bastante elevada para ser calculada computarizadamente o por métodos matemáticos, así como también existen campos de estudio que requieren de evaluación subjetiva, más que objetiva y funcional. En estos casos se usa la opinión del usuario final, a manera de evaluadores de la aptitud de los individuos. En este caso se dice que se hace uso de una estrategia de evaluación interactiva, dando origen a un Algoritmo Genético Interactivo (IGA).

En los algoritmos genéticos interactivos, el usuario final tiene la posibilidad de observar, y dar un valor de aptitud a los individuos de la población, ligando la evolución de las poblaciones al juicio personal del usuario. Sin embargo, esto puede llegar una

gran carga para el usuario al tener muchos individuos por población o un número incrementado de generaciones.

En general cuando el problema de estudio es bastante complejo, se implementan las funciones de aptitud automatizadas necesarias para que el algoritmo pueda funcionar de forma independiente, y se potencia el mismo con la posibilidad de que el usuario evalúe o seleccione los individuos de la población de su preferencia, dándoles un mayor nivel de prioridad que el otorgado por las funciones de aptitud automáticas.

Un ejemplo bastante claro de este tipo de función de aptitud, es evidenciado en el trabajo “MorphLab” de la página web <http://alife.fusebox.com/> en el cual se usa un algoritmo genético y una gramática de Lindenmayer, para generar poblaciones de individuos en forma de plantas, los cuales son presentados paralelamente al usuario en una misma vista, y el usuario tiene la posibilidad de elegir al individuo más prominente, y en base a ese individuo se genera la nueva población.

### **2.2.3. Proceso de Selección**

Luego de que cada individuo de la población cuenta con un valor de aptitud, el algoritmo genético debe aplicar un proceso de selección que garantiza cuales individuos de la población perpetuaran sus características hacia las siguientes generaciones. Este proceso de selección debe ser definido al momento de la construcción del algoritmo genético tomando en cuenta la codificación utilizada, la función de aptitud, y los operadores genéticos, con el fin de obtener un algoritmo balanceado, que no tienda a la convergencia prematura, o hacia la aleatoriedad extrema.

#### **Estrategias de Selección**

Para definir las estrategias de selección, es necesario definir previamente varios conceptos y terminologías comúnmente usadas en cuanto a selección se refiere.

Se entiende por “Presión de Selección” a la probabilidad del mejor individuo de ser seleccionado, en comparación con el promedio de probabilidad de elección de toda la población. En general, se refiere a la forma en que el método de selección elige al mejor individuo con preferencia que a individuos promedio. Cuando esta tendencia es mayor, el mejor individuo será seleccionado una alta cantidad de veces, llenando rápidamente la población y produciendo convergencia prematura. De ser muy baja, los individuos de baja calidad pueden reproducirse de más y retrasar la convergencia.

También se entiende por “convergencia prematura” el estancamiento de la búsqueda realizada por un algoritmo genético gracias a la sobrepoblación de individuos extra aptos muy rápidamente. Al comienzo del algoritmo, la varianza de la aptitud entre los individuos es alta, (algunos individuos son mucho más aptos que otros). Esto hace que bajo estos métodos mencionados, estos individuos aptos se multipliquen rápidamente llenando la población con descendientes parecidos a ellos, previniendo así la exploración del espacio de búsqueda [4]. En algunos casos esto no es útil, si el algoritmo genético se enfoca más en la exploración que en la optimización y en ocasiones inclusive en problemas de optimización inhibe la multiplicidad de soluciones.

En la literatura existen diferentes estrategias de selección desarrolladas y mejoradas con el paso de los años. A continuación se describirá algunas de las más importantes popularmente usadas:

**Selección por ruleta:** Se suman los valores de aptitud de toda la población obteniendo así un total  $S$  y se obtiene un número aleatorio  $r$  entre 0 y  $S$ . Luego se recorre progresivamente la población sumando acumuladamente en una variable  $k$  los valores de aptitud de cada individuo, y cuando  $k \geq r$ , se selecciona este individuo para reproducirse. A este proceso se le conoce como “Giro de la ruleta” lo que da nombre al método. Este proceso es la base de todas las demás estrategias o métodos de selección, siendo aplicado de una u otra manera con variaciones en los valores usados para crear la “ruleta”.

**Selección por truncamiento:** Se ordenan los individuos de acuerdo a su función de aptitud, y se toma la proporción  $1/p$  de individuos de la población más aptos. Este proceso es suficientemente simple, pero al ser tan simple y falto de evaluación inhibe la variabilidad en la población y puede llegar a favorecer la convergencia prematura.

**Selección por Jerarquía:** Es ideada también para evitar la convergencia prematura. Coello [10] describe el proceso de la siguiente manera: Se ordenan o jerarquizan los individuos de menor a mayor dependiendo de su aptitud y se le asignan valores del 1 al  $N$  siendo  $N$  el número de individuos. Se elige  $Max = (1 \leq Max \leq 2)$  y  $Min = 2 - Max$ . El valor esperado de cada individuo será:

$$Valesp(i, t) = Min + (Max - Min) \frac{r(i, t) - 1}{N - 1} \quad (2.2)$$

Siendo  $j(i, t)$  el valor de jerarquía para el individuo  $i$ . En base a este valor se procede al proceso de selección.

**Selección de Torneo:** Se seleccionan  $n$  individuos aleatoriamente de la población, de los cuales se selecciona el mejor [11]. También se puede aleatorizar este tipo de selección reduciendo el número de individuos por “torneo” a dos, y escogiendo un número aleatorio  $r$  entre 0 y 1. Si  $r$  es mayor que una cierta variable  $k$  de probabilidad, definida por el algoritmo, el individuo más apto es escogido, de lo contrario se escogerá el menos apto.

Independiente del método de selección definido, también es necesario hacer énfasis y definir la estrategia de reemplazo o re-población que se llevara al momento de aplicar los operadores genéticos a los padres seleccionados, se decir al momento de producir descendencia.

## Estrategias de reemplazo de la Población

Junto con el proceso de selección, el algoritmo genético debe escoger también la “Estrategia de reemplazo de la población”. Con ella se quiere definir bajo qué proceso la descendencia producida luego de aplicar los operadores genéticos, reemplazara a la población actual.

A continuación se procede a definir las diversas estrategias más utilizadas en la literatura: (falta por completar)

**Reemplazo Completo:** Luego de generada, la nueva población sustituye completamente a la población anterior. Este proceso conlleva a la variabilidad de soluciones y la aleatoriedad, pero permite un máximo desempeño en la exploración del espacio de búsqueda.

**Reemplazo Steady-state:** Se busca reemplazar una cantidad  $k$  de individuos, con la descendencia del cruce entre los más aptos. Este tipo de reemplazo se usa en sistemas que se encargan de evolucionar sistemas basados en reglas. Ya que el sistema no debe olvidar las reglas que logren resolver o acercarse al problema planteado, una vez aprendidas [4].

**Elitismo:** En esta estrategia se preserva una pequeña porción de los mejores individuos de la población actual y el resto de los individuos se reemplaza, aplicando operaciones genéticas entre los individuos seleccionados. Esto permite garantizar la conservación individuos prominentes. Sin embargo esto puede promover la convergencia prematura del algoritmo, pero dependiendo del problema de estudio, puede llegar a aumentar la efectividad del algoritmo genético [4].

### 2.2.4. Operadores Genéticos

Como se mencionó antes, los algoritmos genéticos suelen hacer uso de operadores genéticos que actúan sobre los individuos de la población (generalmente a nivel de

bits), para generar nuevos individuos que puedan contrastar y opacar los actuales con respecto a su aptitud.

La forma en la que estos operadores actúan, debe definirse, de acuerdo con la representación cromosómica estipulada en el algoritmo, lo cual definirá a que nivel trabajan los operadores genéticos.

### Operador de Cruce

Es el operador análogo a la reproducción ente individuos, presente en la naturaleza. Según Mitchell [4] Este operador trata de imitar muy crudamente lo que es el proceso de recombinación genética entre los cromosomas de 2 organismos.

Consiste básicamente en el intercambio de parte de la información genética de 2 cromosomas padres para generar 1 o más cromosomas hijos con una mezcla de información de ambos padres. Generalmente ocurre con un cierto grado de probabilidad, al cual se le denomina “Probabilidad de Cruce”. Este factor determina si efectivamente 2 cromosomas padres generaran o no descendencia por medio de este operador.

Existen diversas implementaciones de este operador que actúan de manera diferente de acuerdo al punto del cromosoma en el cual se cruzan:

- Cruce de un solo punto: Se escoge un solo punto de cruce aleatoriamente y se intercambian las partes de ambos padres que estén luego del punto elegido. Por lo general el punto es tomado entre 1 y  $n-1$  siendo  $n$  el número de individuos, con el fin de que efectivamente ocurra una recombinación en los hijos.
- Cruce de dos Puntos: En este caso se escogen dos posiciones aleatorias, y los segmentos de información cromosómica de cada individuo, entre estas dos posiciones son intercambiados. Es necesario que ninguno de los dos puntos elegidos sean iguales a los extremos del cromosoma, para evitar generar accidentalmente

un cruce de un solo punto. Este tipo de cruce tiene más capacidad recombinatoria que el cruce de un solo punto.

- **Cruce Multipuntos:** teniendo los 2 padres del cromosoma, el cromosoma hijo se crea eligiendo un bit aleatorio perteneciente a uno de los padres. La capacidad recombinatoria que tiene este tipo de cruce es amplia, pero debe usarse con cuidado porque presenta una mayor aleatoriedad que los otros métodos.
- **Cruce especializado:** Existen problemas que son representados de tal manera que el posicionamiento de las características del individuo en cuestión de bits está muy rígidamente definido. Para esos casos se utilizan cruces especializados que tomen en cuenta la representación al momento de hacer el cruce.

## Operador de Mutación

Es el operador que permite a un cromosoma variar en su representación independientemente de sus padres o de cualquier otro cromosoma. No se basa en la herencia como el operador de cruce, es más relacionado con el caos, o con la capacidad que tiene la naturaleza de variar el material genético propio de algunos pocos individuos.

Inicialmente la mutación no se consideraba como un operador genético por sí solo. Mitchell [4] explica en su libro, que el cruce era considerado “el instrumento mayor de variación e innovación en los algoritmos genéricos, siendo entonces la mutación una simple medida que asegurara la población en contra de cualquier fijación permanente de un estilo específico y jugando así un rol más secundario”.

Sin embargo, la apreciación de la utilidad de la mutación como operador ha evolucionado a la par con lo que crece la complejidad de los problemas que se busca resolver con algoritmos genéticos. Actualmente gracias a la amplia diversidad de implementaciones de la mutación en diferentes algoritmos genéticos, esta ha pasado a ser de igual o más importancia (en algunos casos) que el operador de cruce en sí, convirtiéndola así en un operador también.

La forma en que se implemente este operador depende en gran parte de la manera en la que esté representado el cromosoma. Dependiendo del área de estudio o espacio de búsqueda, un creador de algoritmos genéticos puede hacer uso o definir muchos “tipos de mutación”, logrando así un aumento en la capacidad creativa del algoritmo genético y garantizando la diversidad en las poblaciones obtenidas.

### Otros operadores

A lo largo del tiempo han existido otros operadores extra que de alguna manera u otra no llegan a estar al nivel de los 2 principales. Holland [3] definió en los inicios un tercer operador genético llamado *Inversión* el cual simplemente cambiaba los valores  $i$  por los valores  $n-i$  de una cadena de 0 a  $n$  caracteres, es decir “invertía” el cromosoma.

También dependiendo de la representación se pueden aplicar operadores de ordenamiento, los cuales simplemente eligen un pedazo del cromosoma y lo ordenan, ya sea ascendiente o descendientemente. Para esto la representación del cromosoma requiere usar caracteres ordenables (letras, números, entre otros).

Como último, cabe destacar que operadores mixtos, o versiones de los operadores específicas a la representación del cromosoma, entran dentro de esta clasificación. En especial, el operador de mutación es popularmente manipulado, para crear nuevos tipos de mutaciones, que potencien o afecten rasgos específicos codificados dentro de la información del cromosoma.

## 2.3. Usos y aplicaciones

Los algoritmos genéticos han sido ampliamente utilizados para resolver problemas de búsqueda y optimización en espacios complejos. Primariamente eran aplicados en la resolución de problemas computacionales y matemáticos clásicos, aunque con el paso del tiempo se han abierto campo en otras áreas de investigación y desarrollo.

Diversas implementaciones del problema de las 8 reinas, y la resolución de acertijos “8 puzzle” pueden ser observados en los trabajos [12], [13], [14]. En este sentido puede observarse como estos problemas han servido como punto de referencia para mejorar y probar técnicas nuevas, y para evaluar y estudiar el desempeño de los AG con el fin de mejorarlos.

## Capítulo 3

# La Música

Entre uno de los campos de mayor subjetividad entre las artes, está el campo musical. Con el pasar de los años, es posible apreciar como distintos artistas y músicos crean experiencias, describen sentimientos y narran historias a través de este medio. También es notable como el acompañamiento musical es de gran importancia en la mayoría de los medios actuales para darle “vida”, y “sentimiento” a un contenido.

En este capítulo se explica un poco acerca de la manera en que opera la música con el fin de comprender y ahondar como puede lograrse una simulación del proceso de creación y composición musical a través de un sistema automatizado.

### 3.1. Algunos conceptos musicales básicos

A continuación se abordará la descripción de los conceptos básicos, de la terminología musical, susceptibles de ser utilizados para la realización de esta investigación. No se trata de un abordaje profundo de la teoría musical, debido a su amplitud y diversidad. Básicamente se abarcarán los conceptos musicales más relevantes para el desarrollo de un sistema semi-interactivo para la evolución de melodías.

### 3.1.1. Música y Sonido

La *Música* es una forma de expresión que ha venido desarrollándose a la par con el mundo y con el ser humano a través del tiempo. En la antigüedad era utilizada como un elemento organizador del trabajo en serie. También resulta ser un mecanismo útil en la preservación de las especies, un medio de conquista todavía útil hoy en día, entre otras cosas. De una manera bastante genérica, Guevara[15] define la Música como “El arte de combinar sonidos agradablemente al oído según las leyes que lo rigen”.

De esta definición se puede concluir que existe en este ámbito una serie de reglas en base a las cuales se pueden combinar *sonidos* para lograr con esto agrandar al oyente. Cabe entonces preguntar ¿Qué es un sonido?

Definimos por sonido a la vibración que se propaga por un medio elástico. Más específicamente, un sonido audible, es la sensación percibida por el órgano auditivo humano (el oído), con respecto a estas vibraciones. En otras palabras, un sonido es el conjunto de ondas producidas por un cuerpo al vibrar, que crean una variación de presión en el medio que le rodea y pueden ser captadas por el oído.

Bennet [16] define el sonido como “un fenómeno que se produce en nuestro sistema nervioso al ser excitado nuestro aparato auditivo”. Como se ha mencionado anteriormente, estas señales captadas por nuestro sistema auditivo, provienen de “estímulos sonoros” o vibraciones propagadas por medios elásticos, como el aire.

De acuerdo a la naturaleza, rapidez, duración, periodicidad y muchas otras características de estos estímulos, podemos clasificar el sonido en 2 tipos [16]:

**Ruido:** Cuando los movimientos vibratorios son irregulares y complejos (no periódicos) y percibimos un tipo de sensación sonora que puede ser denominada “ruido”. En la actualidad, el ruido más que ser considerado como sonido o conjunto de sonidos desagradables e inestables, ha adquirido cierto nivel de importancia como elemento que introduce variabilidad y creatividad a una pieza musical. Cuando

se usa el ruido en este sentido puede ser denominado más certeramente como “sonido indeterminado” para diferenciarlo de los sonidos caóticos y sin valor musical alguno.

**Sonido Musical:** Cuando los movimientos vibratorios producidos por un cuerpo sonoro son regulares, periódicos, o estructurados, nuestro sistema nervioso registra una sensación diferente que suele ser denominada como “Sonido Musical”. Estos sonidos pueden ser clasificados, estudiados, y agrupados, lo que les otorga una capacidad de ser comprendidos y diferenciados por el ser humano. En este sentido, estos sonidos conforman el material principal de la comunicación humana y el arte musical.

### 3.1.2. Cualidades Principales del Sonido

En los sonidos, específicamente los sonidos musicales, se pueden definir cuatro propiedades principales: Altura, Amplitud o Intensidad, Duración y Timbre.

#### Altura

Se refiere a la “Velocidad Vibratoria” de un cuerpo sonoro. El sonido generado por un cuerpo sonoro que vibra rápidamente, será percibido como un sonido más agudo y mientras que al vibrar lentamente, el sonido es percibido como un sonido grave.

La altura puede ser considerada como una sensación subjetiva de los oídos humanos, pero también puede ser considerado como un valor objetivo (por ejemplo la frecuencia de una onda de sonido). Existe la determinación absoluta y subjetiva de altura. La relativa, está basada en la determinación de la altura en relación con un tono inicial (por ejemplo el tono D4, es más alto que el tono C4). La absoluta es el valor constante y objetivo (por ejemplo, la frecuencia del tono A4 es  $440Hz$ ) [6].

## **Amplitud o Intensidad**

Se refiere al tamaño de las crestas o picos de la onda sonora, la cual representa directamente la potencia de un sonido al ser reproducido. Se refiere también a la fuerza con la que un objeto sonoro se ejecuta. Las diferencias de intensidad sonora dependen de variaciones en la amplitud de los movimientos vibratorios realizados por la fuente sonora, y transmitidos por el aire.

La Intensidad del sonido es un elemento importante en la elaboración de estructuras musicales. El uso controlado y deliberado de sonidos de diferente intensidad, se llama Dinámica Musical. La intensidad se mide mediante una unidad física llamada decibel (Db) [16].

## **Duración**

La duración como su nombre lo indica es el tiempo en el que se mantiene en vibración un sonido. Guevara [15] describe acertadamente que la duración “Nos informa del espacio temporal que ocupa desde su aparición hasta su extinción, es equivalente al tiempo”. La duración de un sonido posee gran importancia musical pues es la que determina el ritmo, el cual es uno de los elementos básicos de la música.

La duración de una sensación sonora depende directamente de la duración del movimiento vibratorio que origina el sonido, aunque en algunos casos la sensación persiste después del cesar el estímulo.

Cuando la duración es muy pequeña afecta nuestra percepción de la altura: un sonido a una frecuencia de 1000 Hz cuya duración es de 0,01 segundos produce una sensación cuya altura es igual a la de un sonido de una frecuencia de 842 Hz cuya duración es de 1,5 segundos. Con la intensidad del sonido ocurren fenómenos similares.

## Timbre

Es el matiz característico de un sonido, también puede ser visto como la marca que identifica específicamente al cuerpo sonoro que ejecuto el sonido, teniendo en cuenta su espectro armónico. En la naturaleza muy pocas veces se puede experimentar sonidos realmente simples compuesto por una sola longitud de onda. A este tipo de sonidos se le llama *sonidos puros*.

La mayoría de los sonidos presentes en la naturaleza son *sonidos complejos* lo cual no es más que una composición de ondas de sonido con diversas cualidades sonoras. Cada uno de estos sonidos complejos tienen un timbre específico el cual identifica sin lugar a dudas la fuente de la cual proviene el sonido, por la forma de las ondas, asegurando que en las mismas condiciones el sonido producido será semejante al anterior [15].

Con espectro armónico se define a los gráficos obtenidos por medio de instrumentos electrónicos especializados, que muestran la intensidad relativa de cada sonido parcial existente dentro de un sonido complejo.

El timbre también es conocido como Color tonal, o Calidad tonal. Diferencia el sonido generado por distintos medios, así sea la misma frecuencia, y el mismo volumen o intensidad.

### 3.1.3. Conceptos Musicales Varios

Luego de estudiar la Música y los Sonidos como base del área musical, se estudiarán otros conceptos importantes relacionados con ellos. Este campo sin embargo, comprende de una base teórica bastante amplia y extensa, de conceptos poco triviales. A continuación se presenta una abstracción general, orientada a los intereses de este trabajo de investigación.

## Frecuencia

En Física de ondas, y movimientos vibratorios, se llama período ( $T$ ) al tiempo empleado en que la onda realice un ciclo completo, y es medido en segundos. Por consiguiente, el número de ciclos por segundo se llama frecuencia ( $f$ ) y se mide en ciclos por segundo. La frecuencia es inversa al período  $f = 1/T$ .

La frecuencia de un sonido representa la cantidad de veces que el cuerpo sonoro emite una vibración en un tiempo determinado. Se mide en una unidad llamada Hercios (Hertz, Hz) y se obtiene al contar el número de veces que la onda sonora oscila durante 1 minuto. Por ejemplo tenemos que la nota con altura A4, esta nota es mejor conocida como “*LA Natural*” y tiene una frecuencia de  $440\text{Hz}$ , es decir, oscila 440 veces por minuto.

Se define como frecuencia fundamental de un sonido compuesto a la frecuencia más baja, tal que las demás frecuencias armónicas presentes en su espectro de sonido sean múltiplos de esta.

Cuando la frecuencia del sonido se duplica decimos que estamos en una octava superior. El espectro de frecuencias que son audibles para el ser humano comprende un rango de 10 octavas. Este espectro comprende desde los sonidos de frecuencia 20 Hz, hasta los sonidos de frecuencia 18000 Hz. A los sonidos bajo este espectro se les conoce como infrasonidos, y a los que se encuentran después del espectro se les llama ultrasonidos.

En la figura 3.1 podemos observar el espectro de frecuencias en comparación con las frecuencias audibles por otros animales, o de diversos eventos naturales.

## Intervalo Musical

Es la diferencia o distancia existente entre 2 alturas o notas musicales. Los intervalos musicales son medidos cuantitativamente en grados o notas, y cualitativamente

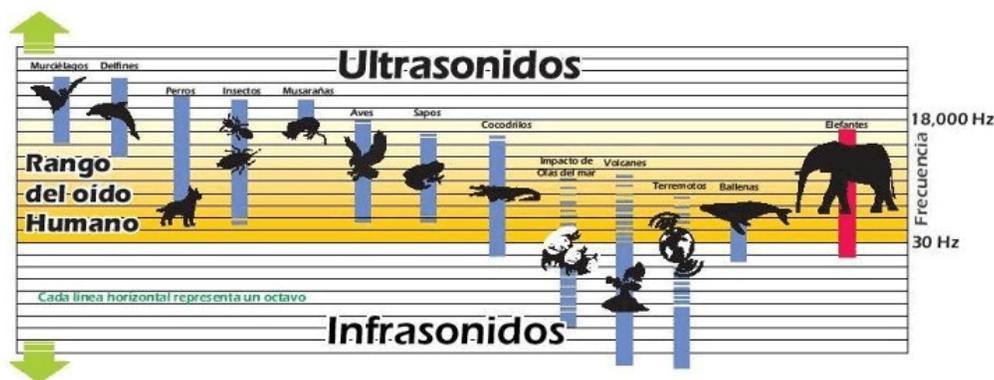


Figura 3.1: Comparación de espectros de frecuencias

en tonos y semitonos.

Dependiendo de si son agradables al oído humano o si producen una sensación subjetivo de tensión, los intervalos musicales pueden ser clasificados como consonantes o disonantes. De acuerdo al sistema estándar de notación europeo, existen 2 propiedades que clasifican a los intervalos: la calidad (Perfecto, mayor, menor, aumentado, disminuido) y el número de la distancia entre tonos (Unísono, segunda, tercera, cuarta, quinta, sexta, séptima, octava)

Son descritos también como “melódicos u horizontales” si se refieren a la relación entre 2 tonos sucesivos y “armónicos o verticales” si se refieren a la relación entre 2 tonos que suenan simultáneamente.

### Nota Musical

Una nota es definida universalmente como un sonido o vibración cuya frecuencia fundamental es constante. Según el Sistema europeo estándar de notación musical, para cualquier nota música, el doble de su valor representa la misma nota con una altura mayor. Por lo cual entre esta nota y la otra se pueden encontrar 8 notas básicas que son las definidas en el estándar actual: DO, RE, MI, FA, SOL, LA, SI, DO.

## Tono y Semitonos

La definición de tono es ambigua. En el campo de la acústica, el tono es una medida subjetiva de cuán agudo o grave es un sonido. En ese sentido podría verse como una definición alterna de altura musical, dando paso también a la definición de tonalidad y color tonal.

Sin embargo más con respecto a intervalos musicales, podemos definir al tono, como la representación unitaria de un sonido determinado, con respecto al sistema musical de notación occidental. Las alturas musicales son escritas y clasificadas por medio de notas musicales las cuales a su vez no son más que denominaciones de tonos o semitonos de acuerdo al caso.

Según el sistema estándar europeo, una octava se divide en 8 notas musicales, y en 12 semitonos igualmente distribuidos. Usualmente una octava, se distribuye como 5 tonos (DO, RE, FA, SOL, LA) y 2 semitonos (MI, SI) respectivamente y es representada por la nota con la cual comienza y termina.

## Acordes Musicales

Se llama acorde a una agrupación de sonidos que se ordenan por intervalos musicales de notas superpuestas, sonando todos los sonidos a la vez. Un acorde puede estar formado por tres, cuatro o más notas, entre los cuales los más usados son los de 3 notas. A este tipo de acordes se les llama triada. Por lo general los acordes son denominados de acuerdo a su nota fundamental, la cual suele ser la nota con frecuencia más baja.

Los acordes pueden ser *mayores*, *menores*, *de séptima* y *de quinta*, siendo los acordes mayores y menores los más utilizados en el sistema tonal occidental.

## Ritmo

Se refiere a la distribución, organización y repetición de sonidos y silencios en el tiempo. Bennet [16] define el ritmo como “la distribución temporal de los sonidos que integran una manifestación musical cualquiera”.

Es caracterizado como el “Aspecto fundamental y primordial de la música” ya que sin el las representaciones sonoras serían aleatorias y con poco orden al oído humano. El ser humano cuenta con una capacidad para memorizar eventos y ocurrencias, y al organizar una representación sonora de una manera coherente y repetitiva, será mucho más fácil para el humano común memorizar y asimilar este tipo de expresión.

Adicionalmente, es importante definir algunos conceptos referentes al ritmo:

**Figura Musical:** Fundamentalmente el ritmo se divide haciendo uso de las Figuras musicales, entre las cuales podemos encontrar las referentes a los tonos o notas, y las referentes a los silencios. Una *Figura musical*, es un símbolo utilizado comúnmente para representar cual es la nota y cuanta es su duración en un determinado momento de una composición musical.

**Silencio (figura musical):** Es aquella figura musical que no representa ninguna nota sino un espacio vacío en un tiempo determinado. Es una de las figuras más importantes en cuanto a la rítmica de la composición.

**Clave:** Es un símbolo musical usado en los pentagramas para denotar la altura donde se escribe o debe ejecutarse la música. Existen principalmente 3 claves: La clave de SOL para indicar sonidos más agudos y perceptibles por el oído humano, La clave de FA, para indicar tonos más graves, normalmente es usada como clave secundaria, y la clave de DO, la menos usada, pero de gran importancia para sonoridades especiales, como es el caso de la viola [15].

**Compás:** Entendemos por compás al símbolo musical utilizado para denotar la distribución de las figuras (específicamente la duración de cada figura) y define la

distribución de estas en la composición. En un compás, el denominador define la figura base a usar en la composición y el numerador define, cuantas figuras caben en cada compás. Los compases más utilizados pueden ser observados en la figura 3.2

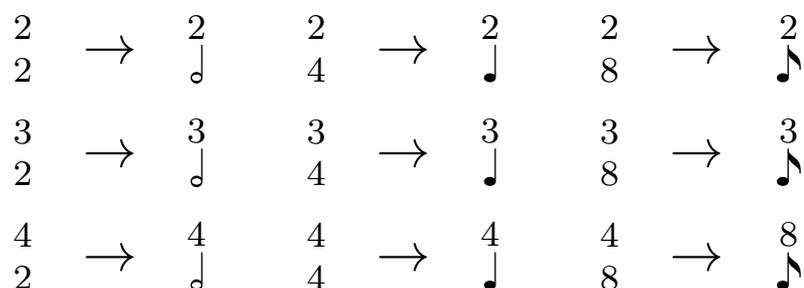


Figura 3.2: Compases musicales y sus figuras relacionadas

**Tempo:** El tempo es la velocidad en la que se ejecuta una composición musical. Esta velocidad es independiente al ritmo de la composición, ya que se pueden obtener versiones más rápidas o lentas de la composición alterando su tempo, y aun así se conservarían las relaciones temporales internas de la canción (ritmo). Usualmente es medido en pulsaciones por minuto.

## Melodías y Composición Musical

Una *melodía* se define como la composición de sonidos o alturas musicales de manera sucesiva o consecutiva. Con respecto a esta definición podría decirse que una melodía también es un conjunto caótico de sonidos sin ningún significado musical. En ese sentido, al agregar a tal concepto la vinculación con el concepto de ritmo, se crea la definición de un conjunto de sonidos como un a composición *melodiosa*.

Una tonada no siempre es “Melodiosa”. Las melodías que son fácilmente recordadas, o que despiertan en el oyente un deseo de cantar, constituyen ejemplos perfectos del elemento melódico de la música. Sin embargo también es estudiada como “melodía”, una sucesión de sonidos de altura fija. [16].

Una *composición musical* entonces no es más que una serie de melodías o una melodía específica como tal, con un ritmo y de carácter musicalmente agradable. En el ámbito musical se define por composición al producto el trabajo de los músicos en generar este tipo de melodías, los cuales suelen ser calificados como compositores musicales.

En este trabajo se utilizara el término “Ideas melódicas” para referirse al producto musical final del algoritmo, el cual puede ser utilizado para su refinamiento posterior y completado opcional por el usuario.

### **Armonía Musical**

Consiste en la composición de notas simultáneas y superpuestas, formando con esto sonidos complejos, que dan una sensación musical más completa y llena. También puede ser definida como la composición y evaluación de los acordes musicales dentro de una composición musical.

La armonía musical está basada en la música “polifónica” la cual no es más que música en la que se suele ejecutar distintas notas musicales en forma simultánea y coordinada. Este estilo musical es propio de la música occidental, y se encuentra ausente en otras culturas por lo cual estos conceptos tienen un alcance relativo a la cultura.

Este término da paso a los conceptos de Consonancia y Disonancia que se explicaran más adelante, pero puede decirse bien que una “composición armoniosa” busca mantener un balance entre armonías consonantes y disonantes, a manera de lograr el gusto del oyente.

## 3.2. Criterios usados para la composición y evaluación musical

Para que una composición sea aceptable al oído musical, debe cumplir con características propias innegables más no inflexibles. A lo largo de los años, de acuerdo con la capacidad auditiva de los seres humanos y de los estilos culturales existentes, se ha desarrollado un conjunto de estándares y reglas, que permiten evaluar y especificar cuando una composición puede ser calificada como música o no. Esta subjetividad da paso a ciertos elementos que conforman a una composición de tipo musical:

- La Melodía: o relación entre sonidos *sucesivos*
- La Armonía: o relación entre sonidos *simultáneos*
- El Ritmo: o disposición temporal de los sonidos en la composición
- La Instrumentación: o fuente que produce el sonido

La existencia de estas características en una composición musical no es necesaria completamente de forma directa, más si es discriminante [15]. Es decir, Una composición puede carecer de melodía y de ritmo pero aun así conservar una armonía para ser calificada como musical. Al carecer de todas estas características solo estamos en presencia del ruido.

Las definiciones explicadas en esta sección están directamente relacionadas con estas cualidades, permitiendo establecer criterios musicales evaluativos para especificar cuando una composición o melodía es adecuada musicalmente.

### 3.2.1. Consonancia y Disonancia

La Consonancia y Disonancia son los conceptos auditivos sobre los cuales se basa gran parte del estudio de la calidad musical. Pese que la sensación de gusto y agrado

por un sonido es un concepto subjetivo, estos términos son usados para referirse a los intervalos, acordes o tonalidades musicales que en su mayoría son percibidos de esa manera.

La Consonancia y disonancia pueden ser definidas de la siguiente manera:

**La Consonancia** se refiere a una relación de armonía, acorde o intervalo musical que es estable o agradable al oído y causa reposo auditivo. En otras palabras, el conjunto de notas se escucha agradable musicalmente. En otras palabras, la consonancia es el intervalo musical armónico o melódico que es aceptado por nuestro oído.

**La Disonancia** se refiere a una relación de armonía, acorde o intervalo musical que es inestable o que causa tensión. En otras palabras, el conjunto de notas se escucha desagradable o es incoherente musicalmente. En otras palabras, un intervalo musical disonante es aquel intervalo melódico o armónico que es rechazado por nuestro oído. Por lo general las disonancias crean una sensación como si necesitaran de una consonancia para liberar la tensión auditiva producida, a esto se le llama “resolver” una disonancia.

En cuanto a los intervalos, el sistema tonal estándar, considera que todos los intervalos que no son unísono, cuarta, quinta y octava justa, o tercera y sexta mayor, son disonantes. Por otro lado, la consonancia y disonancia también puede evaluarse y calificarse a diversos acordes musicales. En ese caso se considera acordes consonantes a los acordes triadas mayores y menores en cualquiera de sus formas.

Hasta finales del siglo XIX, la música se componía principalmente con consonancias, y se condimentaba con disonancias para añadir un cierto grado de tensión. En la música del siglo XX, sin embargo, se utilizan mucho más las disonancias que, en lugar de resolverse, se mezclan cada vez con más disonancias. De hecho, en algunas composiciones se evita deliberadamente toda consonancia.

### 3.2.2. Tonalidad Musical

Puede definirse la tonalidad musical como un conjunto de sonidos o notas que están en íntima relación entre sí. Matic [6] define la tonalidad como un sistema de notas en el cual, existen relaciones de jerarquía específicas entre alturas musicales, que están basadas en una nota central o tónica. Se le llama tónica a la nota que define la tonalidad, y le da su nombre. También se refiere con tónica a la primera nota de una escala musical.

El concepto de tonalidad, permite clasificar sonidos y relacionarlos entre sí, permite *componer* y agrupar sonidos, facilitando la enseñanza, estudio y creación de piezas musicales de gran belleza auditiva y utilidad. De acuerdo a esto entonces, una composición musical se ejecuta, por lo general, haciendo uso de una o 2 tonalidades musicales que sean consonantes, o si no lo son, que generen una sensación auditiva agradable.

El grado de consonancia se determina usando una “función tonal” o “Diatónica” la cual tiene como parámetro característico el intervalo que cada nota forma a partir de la tónica.

En la antigüedad, se le adjudicaban personalidades emocionales a los distintos modos (tonalidades) dados a la composición musical. De estos modos sobrevivieron fundamentalmente dos: el jónico o también llamado modo mayor, y el eólico o modo menor. A dichos modos se les adjudican las personalidades emocionales referentes a la alegría y la tristeza respectivamente. Más concretamente, se puede especificar más esta disgregación al agregar los conceptos de ritmo, tempo y timbre a la composición, más sin embargo esto es muy dependiente del contexto.

### 3.2.3. Escalas Musicales

Es una sucesión de sonidos por grados conjuntos en forma ascendente (DO, RE, MI, FA, SOL, LA, SI) o descendente (SI, LA, SOL, FA, MI, RE, DO) todo esto dentro

de una debida tonalidad. Bennet [16] define una escala como “un grupo de sonidos que han sido escogidos como material básico para hacer música”.

Las escalas se caracterizan mayormente por la cantidad de sonidos que contienen y sobre todo por las relaciones de intervalos que existan entre estos sonidos. Dependiendo de la locación la música se puede estructurar usando escalas de 5, 6 o más sonidos. Sin embargo en la música occidental se acostumbra usar escalas de 7 o más para la composición.

Generalmente y muy en relación con la tonalidad las escalas se denominan usando la primera nota con la que comienzan. Luego se les califica de acuerdo a su comportamiento. El comportamiento de una escala se describe por medio de la distribución de tonos y semitonos, a partir de la nota inicial. Como ejemplo, en la figura 3.3 vemos la Escala Cromática, que representa una sucesión de tonos y semitonos sin omitir ninguno entre ellos:

DO - DO# - RE - RE# - MI - FA - FA# - SOL - SOL# - LA - LA# - SI - DO

Figura 3.3: Escala cromática en Do

La mayor parte de la música occidental fue compuesta usando 2 escalas: La escala mayor y la escala menor. Ambas están formadas por siete sonidos (más la octava, que es un nuevo comienzo). La escala menor posee 3 formas: menor melódica ascendente, menor melódica descendente y menor armónica, que se diferencian entre sí por la distribución de sus tonos y semitonos.

**La transposición** Es una operación que fue ideada junto con el concepto de escala debido a que estas se basan en la relación entre sus intervalos y no específicamente en la nota musical en la cual comienzan. Trasponer una escala consiste en moverla hacia otra nota inicial preservando sus intervalos tonales. de esta forma podemos estudiar cualquier escala en cualquiera de los 12 tonos básicos de nuestro sistema tonal.

### 3.2.4. Diferencias y relaciones entre tonalidad musical y escalas

Como ha podido observarse, El concepto de tonalidad musical y el de escalas musicales pareciera que fuesen sinónimos. Sin embargo esto no es correcto. Pese a que ambos conceptos guardan estrecha relación con “un conjunto de notas representativas de una composición musical”, la tonalidad es un término más amplio y genérico.

La tonalidad es más vista como el conjunto de notas, de forma abstracta, mientras que también posee otros elementos que ayudan a describir la sensación musical de una pieza. Uno de esos elementos es la escala musical, la cual representa el mismo conjunto pero más que todo representa la relación de orden y permutaciones entre sus tonos y subtonos.

En este sentido la tonalidad y las escalas es posible observar como ambos conceptos guardan una relación estrecha, mas representan cosas distintas. Aun así, son el pilar que da paso a la composición musical como área de estudio, y a la evaluación de la calidad musical.

### 3.3. Musical Instrument Digital Interface (MIDI)

MIDI es un protocolo de comunicación estándar que permite a los computadores, sintetizadores, secuenciadores, entre otros dispositivos musicales electrónicos comunicarse y compartir información para la generación de sonidos.

Los aparatos MIDI pueden clasificarse en 3 tipos:

- **Controladores:** Son los que generan los “mensajes MIDI” (activación o desactivación de una nota, variaciones de tono, entre otros). El más común es el piano, sin embargo, actualmente todo computador cuenta con la capacidad de implementar un controlador MIDI.

- **Unidades generadoras de sonido:** reciben los mensajes MIDI y los transforman en señales sonoras.
- **Secuenciadores:** Están destinados a grabar, reproducir o editar mensajes MIDI. Pueden ser hardware, software, o estar incorporados en un sintetizador.

Según Sgroi[17] “Un Sistema electrónico musical MIDI, contiene un controlador MIDI y una gran variedad de generadores de sonido MIDI. Las señales de audio de los generadores son combinadas y transmitidas por un sistema de sonido para producir el sonido Audible Total”.

Funciona principalmente a través “mensajes MIDI” los cuales son comandos enviados desde un dispositivo MIDI a otro para que ejecute alguna acción. Los más comunes son:

- **NoteOn:** Comienza la reproducción de una nota
- **NoteOff:** Detiene la reproducción de una nota
- **ProgramChange:** Permite cambiar varias opciones internas del controlador (volumen, instrumento, tempo, entre otros)

## Capítulo 4

# Algoritmos Genéticos Musicales

Al usar algoritmos genéticos se busca mejorar de alguna forma el resultado obtenido de algoritmos simples de generación de contenido musical. En el capítulo previo, se expuso claramente una serie de criterios musicales que sirven como base para la evaluación musical. En este sentido, dichos criterios pueden ser usados para evaluar y optimizar melodías creadas por el computador.

En este capítulo serán cubiertos algunos trabajos recientes que aplican Algoritmos Genéticos para fines de composición musical, abordando con mas profundidad los que fueron de utilidad en el desarrollo de esta investigación.

### **4.1. Convergencia audible para la extensión óptima de melodías base con evaluación estadística de distancia interválica específica del género, Ronald Hochreiter[18]**

En este trabajo se usa un algoritmo genético, para calcular extensiones optimas de una melodía base de acuerdo a la minimización de la distancia entre los intervalos.

Como estructura de representación, se utilizo un vector  $v$  de 3 dimensiones en la cual su primera coordenada representa una nota correspondiente en el rango  $(0, 11)$ , su segunda coordenada representa la equivalencia enarmónica, un valor en el rango  $(0, 4)$

representando los modificadores musicales usables ( $\emptyset, \sharp, \flat, \times, \flat\flat$ ), y la tercera coordenada representa la octava de la nota, también expresada como un valor entero. Por ejemplo la nota  $C\sharp 4 = (1, 1, 4)$ .

Una melodía base  $m$  que contiene  $n_m$  notas. La melodía extensión, consistirá de  $n_e$  notas por cada barra de la melodía a extender. Así, la melodía final tiene

$$n = n_m \cdot (n_e + 1)$$

notas. Cada cromosoma de la población contiene determinada cantidad de datos melódicos, en los cuales, en los cuales los vectores de altura  $v$  son mapeados de  $\mathbb{N}^3$  a  $\mathbb{N}$  con  $\bar{v} = v_2 \cdot 12 + v_0$ .

## Evaluación de Aptitud

Para calcular la aptitud se necesitan ciertos valores:

- Un conjunto de categorías de intervalos, donde un valor (escalado ordinalmente) será asignado a cada intervalo
- Una estructura objetivo de media y varianza para cada barra, o para toda la melodía. Los valores de media y varianza deben estar en línea con los valores asignados a cada intervalo

La Aptitud  $f$  de un vector melodía generado es evaluado calculando la sumatoria ponderada de las distancias entre los valores pre especificados de medias  $\mu_i$  y varianzas  $\sigma_i^2$ .

## Operadores genéticos

Como cruce se implementó un cruce estándar de solo punto. Para la mutación se utilizaron los siguientes operadores:

- Intercambiar dos notas adyacentes
- Transponer una nota por un intervalo aleatorio
- Transponer una nota por una octava
- Revertir un grupo de notas dentro de un punto de inicio y un punto final determinado
- Invertir algunas notas dentro de un punto de inicio y un punto final determinado

## 4.2. Un algoritmo genético para componer música, Dragan Matic [6]

Este trabajo implementa un algoritmo genético que produce composiciones melódicas simples. Matic [6] hace uso de un ritmo predefinido durante la creación de los individuos de la población inicial, permitiendo la uniformidad entre los individuos generados. Se hace uso de operadores genéticos varios, junto con modificaciones de operadores genéticos existentes, que permiten la alta variabilidad de notas y tonos en las melodías evitando la convergencia prematura del algoritmo en soluciones de baja calidad musical.

### Estructura y Representación

Los individuos son representados como arreglos de notas musicales de una determinada tonalidad. Si se tiene que  $n$  es el número total de alturas musicales a incluir, se toma que la primera nota es 1, y será denominada nota referencial. De ahí en adelante se numeran hasta  $n$ , las siguientes alturas del conjunto. Luego de esto se halla el máximo común divisor entre las duraciones y lo denominamos “La duración más corta”, representado por  $k$ . Tenemos también que la composición cuenta de un número  $m$  de compases musicales o barras. Cada barra tiene  $p$  pulsos, y cada pulso tiene  $k$  “distancias más cortas”, entonces:

- Cada barra tiene  $pq$  “distancias más cortas”
- Cada tono puede durar  $tk$  “distancias más cortas” para algún tiempo  $t$
- La composición es de tamaño  $mpq$
- Se utiliza un arreglo en caracteres de tamaño  $mpq$  para representar la composición. Cada caracter es un número del  $[0, n + 1]$  siendo los números  $[1, n]$  las notas musicales comenzando desde la nota referencial, 0 un silencio de duración  $k$ , y  $n + 1$  una prolongación de la figura anterior durante un tiempo  $k$  o la “distancia más corta”

Un ejemplo de este tipo de representación para una composición de 14 tonos y 32 notas en total puede ser observado en la figura 4.1 y la tabla 4.1.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
0	3	6	7	8	15	15	7	8	7	6	5	4	15	15	15	0	4	5	6	7	15	15	6	7	6	5	4	3	15	15	15

Tabla 4.1: Representación del Cromosoma Representado en la figura 4.1



Figura 4.1: Cromosoma representado en el pentagrama.

## Función de Aptitud

Según el estado del arte actual, no existe una función de aptitud específica que diga cuando una solución musical es de mejor calidad que otra. En la mayoría de los casos Se utiliza una función de aptitud total, basada en una suma de diferentes criterios o funciones de aptitud parciales. Para este caso, dicha función de aptitud se

define como:

$$\sum_{i=1}^n \lambda_i f_i$$

En donde  $\lambda_i$  representa el peso (influencia) del valor  $f_i$  en la aptitud total, y  $n$  es el número de aptitudes parciales evaluadas.

Las funciones de aptitud utilizadas para el cálculo de la función de aptitud total usadas fueron:

- La “distancia” entre una composición dada y la composición referencial. Esto se hace calculando la media aritmética y la varianza de cada una de las 2 composiciones
- La relación entre el número de intervalos disonantes y todos los intervalos de la composición
- La relación entre el número de tonos fuera de la tonalidad dada con respecto al número total de tonos de la composición

Para esta implementación, cuando se habla de distancia entre una composición y la composición referencial, se refiere a que las relaciones entre los intervalos internos de la composición se mantengan en un mismo nivel de “calidad” y no por debajo de la composición referencial. Cuando se habla de “calidad” con respecto a los intervalos, se refiere al valor obtenido de la suma de los valores de disonancia de esta composición.

Los valores de disonancia de un intervalo son asignados, dependiendo del tipo de intervalo. Algunos valores propuestos en este trabajo pueden ser observados en la figura 4.2:

## El algoritmo genético

Matic considero los siguientes puntos en cuanto a la implementación del algoritmo genético:

Categoría de intervalos	Valores (1)	Valores (2)
Consonancias perfectas (unison, octave, 4ta, 5ta)	1	1
Consonancias imperfectas (3ras y 6tas mayores y menores)	2	3
Segundas (mayores y menores)	3	1
Septimas (mayores y menores)	3	3
Intervalos mas largos que una octava	5	5

Tabla 4.2: Tabla de valores de consonancia propuestos por Matic

- Al principio se genera una población inicial en base a los parámetros de entrada con un ritmo predefinido. Luego se aplican diversos operadores de iteración a iteración, mientras se busca el individuo que sobrepase el criterio de completitud esperado.
- La función de aptitud es una sumatoria de las imperfecciones que se van observando de cada individuo con respecto a su estructura interna musical. Esta función de aptitud utilizada es decreciente, significando esto que mientras menor es el valor de aptitud de un individuo, más calidad tiene.
- De los operadores genéticos existentes no se toma en cuenta al cruce como operador. Este trabajo se centra en el uso de diversos métodos de mutación basados en la teoría musical, y a la representación usada.
- Se calcula la aptitud de todos los individuos de la población, y se crean nuevos individuos aplicando diversos operadores genéticos musicales a los más aptos de la población actual. En base a esto una nueva población es creada seleccionando los mejores individuos entre los viejos y los nuevos.

En la figura 4.2 Se puede observar los elementos básicos del algoritmo propuesto por Matic así como su flujo de trabajo.

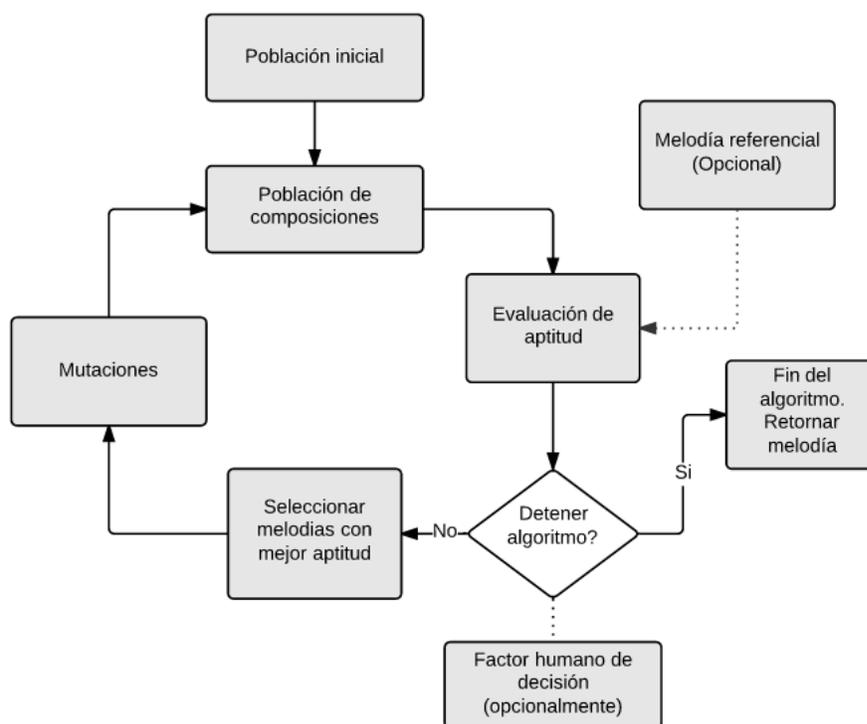


Figura 4.2: Esquema del Algoritmo genético usado para la composición musical

## Operadores genéticos

En este trabajo no se usó el operador de cruce. En su defecto se utilizaron diversos tipos de mutaciones bastante ligadas a la parte musical de la composición. Según el autor, esta decisión optimiza la búsqueda aplicada por este algoritmo, haciendo obsoleto el uso del operador de cruce en cuanto a optimización. Matic también estipula que no tiene sentido cruzar composiciones tan cortas como las estudiadas en este caso.

Los operadores de mutación utilizados fueron:

### Cambiar una nota por una octava (Superior o Inferior):

Reduce considerablemente la longitud de los intervalos dentro de la composición, o la aparición de intervalos largos. Consideramos como largo a un intervalo musical mayor que una octava.

**Cambiar una nota por otra:**

Esta mutación permite corregir la aptitud de un individuo previo. Modifica las relaciones interválicas en la vecindad de la nota, y permite generar intervalos más consonantes. Sin embargo existe un chance de cambiar un intervalo consonante por uno disonante, más esto no afectara al algoritmo, debido a que el individuo previo prevalecerá.

**Intercambiar una o dos notas consecutivas:**

Se escoge una nota y se cambia de posición con una nota vecina. Al igual que el operador anterior, también puede corregir las relaciones interválicas de la vecindad de las notas involucrada.

La selección como operador también es aplicada para elegir de acuerdo a la aptitud a los individuos que permanecerán en la próxima generación. La eliminación también es usada para remover individuos con baja aptitud y posibles réplicas de otros individuos de acuerdo a su aptitud.

**Detalles de implementación**

El algoritmo esta implementado en Java. Para el despliegue de los resultados en notación musical se utilizó el API Notation Musician, y para la programación musical se utilizó JFugue. El proyecto fue realizado y compilado en el IDE JCreator.

**4.3. GenJam: un algoritmo genético para generar solos de jazz, John A. Biles [8]**

Este trabajo describe a GenJam, un modelo algorítmico genético interactivo que busca simular a un músico jazzista novato que está aprendiendo a improvisar. GenJam usa progresiones de acordes y escalas para la creación de poblaciones de composiciones musicales.

A su vez al ser un algoritmo genético interactivo, un mentor humano provee una respuesta en tiempo real a las composiciones generadas, deduciendo de ello la aptitud de los individuos y permitiendo la evolución de la población mediante diversos operadores musicales.

## **Diseño**

Para improvisar una tonada, GenJam, lee un archivo de progresión, el cual provee al algoritmo del tempo, el estilo rítmico necesario (el compás), el número de coros que debe tomar, y la progresión de acordes a usar. La improvisación de la tonada es generada construyendo coros de eventos MIDI, decodificados de las secuencias obtenidas de cada instrumento.

Al obtener estas secuencias, el algoritmo produce varias improvisaciones en base a las progresiones de acordes que se han usado en los coros anteriores, y con esto construye individuos pertenecientes a una población. Luego de esto, el mentor escucha los solos generados y asignar los valores ‘g’ y ‘b’ a las porciones de cada individuo. La aptitud entonces es determinada en base a estos valores, incrementando un contador por cada ‘g’ asignada y disminuyéndolo por cada ‘b’ asignada.

GenJam opera en 3 modos: Modo Aprendizaje, Modo Reproducción y Modo Demo. El modo “Aprendizaje” está hecho para mejorar los valores de aptitud de las composiciones generadas, sin uso de operadores genéticos algunos. Frases musicales varias son presentadas para ser evaluadas por el mentor. En el modo de “Reproducción” el algoritmo genera nuevos individuos haciendo uso de diversos operadores genéticos. El modo “Demo” está destinado a la ejecución de las piezas musicales generadas. El algoritmo elige cuales son las composiciones más prominentes mediante una selección por torneo y las reproduce al oyente.

## Representación del Cromosoma

GenJam utiliza una representación binaria a dos niveles basada en la posición. Con 2 niveles se refiere a que para cada iteración del algoritmo genético en vez de contar con una única población, se usan dos poblaciones: Una población de “frases”, y la otra de “medidas”.

Un individuo en la población de “medidas” es una serie de eventos MIDI (notas musicales). Por su parte, un individuo en la población de frases es una serie de índices que mapean directamente a individuos en la población de “medidas”. En este sentido, el algoritmo trata de buscar optimizar, ambas poblaciones para construir solos haciendo uso de ambas simultáneamente.

Los individuos en la población de medidas, están compuestos por un valor de aptitud, y un cromosoma, conformado por una serie de valores de 8 eventos, cada nota con una duración de una medida del compás 4/4. Se tienen 3 tipos de eventos: *Nueva nota*, *Descanso* y *Sostenido*. Existen 14 diferentes notas que pueden ser representadas con el evento nueva nota. El evento Descanso, silencia la nota previa y genera una pausa de duración estándar (1 medida de 4/4). El evento Sostenido, no hace nada, lo cual prolonga tanto una nota como un descanso por la duración estándar.

Los cromosomas de ambas poblaciones son usualmente representados por cadenas de caracteres numéricos del 0 al 15, siendo 0 un Descanso, 15, un Sostenido, y los caracteres del 1 al 14 las notas musicales respectivas.

## El “Cuello de Botella” de la Aptitud

En los algoritmos genéticos interactivos, es muy común que al ser un ser humano el que evalúa y otorga feedback de las poblaciones, su opinión y criterio se vea reducido con la evaluación consecutiva, así como también, se vea afectado su desempeño de acuerdo con factores desgastantes como son, el cansancio, la monotonía, entre otras cosas.

A este proceso se le conoce como “Cuello de botella de la Aptitud” y es el principal inconveniente de los algoritmos genéticos interactivos. En GenJam este proceso se evidencia en la figura del mentor, al tener este que escuchar todos los individuos de la población generada, y evaluarlos.

## Operadores Genéticos

Nótese que la aplicación de los operadores genéticos solo ocurre en el modo “Reproducción” de GenJam. Aun así, el modo “Demo”, permite ejecutar el operador de selección.

Los operadores de selección y reemplazo se mezclan en una versión modificada de un proceso de selección por torneo. Cuatro individuos son escogidos aleatoriamente sin importar la aptitud, para formar una *familia*. De los cuatro individuos, los 2 con la aptitud más alta son usados como padres y los 2 peores individuos son reemplazados con los hijos del cruce de los padres.

Para acelerar el aprendizaje, creando no solo nuevos, sino mejores individuos, se usan “mutaciones dependientes del contexto” tanto para frases como para medidas. Esto viola las reglas de los algoritmos genéticos que dicen que los operadores genéticos deben de ser ignorantes del contexto sobre el cual operan. Sin embargo es utilizado al representar una optimización significativa y una ayuda al mentor como elemento evaluador del sistema. Algunas de las mutaciones utilizadas se mencionan a continuación:

- **Inversión (de frase y de medida):** Invierte el orden de los genes del cromosoma sea frase o medida
- **Rotación hacia la derecha (de frase y de medida):** se cambian *ruedan* los genes hacia la derecha  $n$  posiciones
- **Reparación Genética:** Se cambia una medida por otra

- **Frase Huérfana:** cambia toda una frase aleatoriamente
- **Ordenar ascendiente o descendientemente:** de acuerdo con el número de la frase o la medida

## Detalles de implementación

GenJam fue desarrollado en un entorno Macintosh/Think C, haciendo uso del CMU MIDI Toolkit Framework. Para la representación de las notas y alturas musicales se hizo uso de la interfaz de notación MIDI comúnmente usada en gran variedad de sintetizadores e instrumentos musicales electrónicos.

## Capítulo 5

# Desarrollo e Implementación

A continuación se describe el proceso a través del cual se implementó la solución propuesta. Se hará énfasis en las etapas del desarrollo desde conceptualización y estructuración, hasta los detalles que permitieron dar forma a lo que se presenta como una solución posible.

### 5.1. Conceptualización y estructuración de la solución

#### 5.1.1. Metodología Planteada

Como metodología de desarrollo se propuso hacer uso de un esquema de “Desarrollo por Prototipos”. Esta metodología plantea el desarrollo de prototipos funcionales y autoescalables de la aplicación. Esto quiere decir que dichos prototipos se contrastan directamente con los objetivos del trabajo de investigación se crean nuevos prototipos que mejoren los primeros hasta lograr el resultado esperado.

A su vez, se propuso el uso de una arquitectura basada en la programación orientada a objetos (POO). El sistema está conformado por diferentes módulos que se encargaran de otorgar al usuario las funcionalidades requeridas. Cada módulo está estructurado mediante clases y métodos que permitan la fácil interconexión entre las distintas partes de la aplicación.

### 5.1.2. Conceptualización

El programa fue concebido como un sistema compuesto por varios módulos. En su interfaz más básica contaría con la existencia de 3 módulos que se relacionarían entre sí:

**Un módulo “Generador”:** el cual se encarga de proveer melodías completas o pedazos de melodías en base a una representación específica.

**Un módulo “Evolutivo”:** que contiene el algoritmo genético y maneja la evolución de poblaciones de melodías con respecto a ciertos parámetros musicales.

**Un módulo “Reproductor”:** El cual reproduce y convierte desde melodías en la representación escogida a música en formato MIDI.

Para la primera versión del programa se pensó hacer que los módulos se relacionaran entre sí de la siguiente forma: El modulo “Evolutivo” crea una población de individuos usando el modulo “Generador”, el cual hace melodías con una representación planteada. Luego el modulo “Evolutivo” trabaja con dichas melodías, las opera y mezcla genéticamente a través de las generaciones, y en todo momento el módulo “Reproductor” le permitía al usuario escuchar y guardar las melodías de su preferencia.

El programa también contaría con 2 submódulos auxiliares que permitirían al usuario controlar mejor el programa, se pensó entonces crear:

- Un submódulo de inicialización para configurar el programa inicialmente.
- Un submódulo de configuración para re-configurar el programa y proveer otras opciones, una vez iniciado.

En etapas finales de desarrollo se creó también un módulo de pruebas que permite a los usuarios escuchar un conjunto de melodías y puntuarlas respectivamente teniendo

en cuenta sus preferencias personales, a fin de contrastar esto contra la función de aptitud elegida.

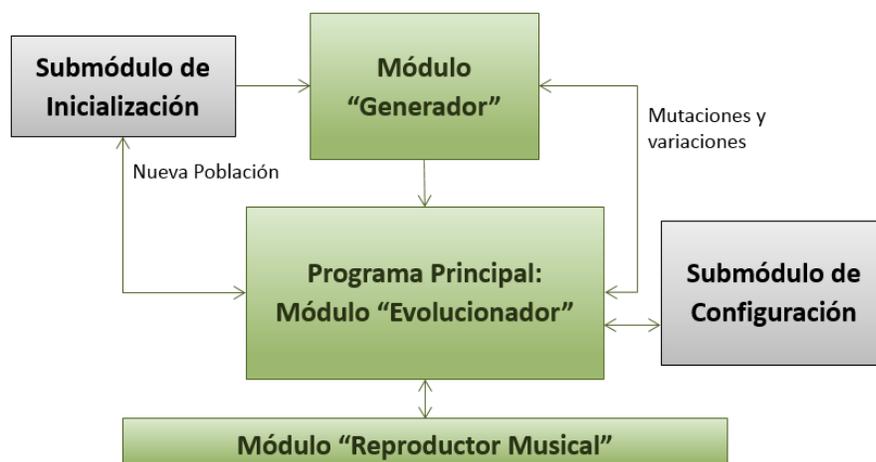


Figura 5.1: Diagrama de módulos de la aplicación

En la figura 5.1 se puede apreciar el diagrama de modelos de la aplicación final.

### 5.1.3. Implementación y estructura

El programa fue desarrollado usando el lenguaje de programación C#, haciendo uso del API gráfico de Microsoft “Windows Forms”. La implementación de las pantallas e interfaces gráficas en esta especificación esta realizado siguiendo el paradigma de programación orientada a objetos (POO) enfoque que a su vez fue elegido para realizar este trabajo.

Se desarrollaron varias clases base que constituyen la lógica de la aplicación estas son:

**Note:** una nota es básicamente una Altura musical, una Octava una duración. Esta clase contiene estos datos, y con ellos también calcula y guarda su valor MIDI y su representación en strings. Esta clase es considerada como el centro de

la aplicación ya que guarda la mayor cantidad de funciones que representan y manipulan información musical.

**Bar:** una melodía está compuesta por un conjunto de barras o compases los cuales contienen una cantidad de notas determinada por el tamaño de sus duraciones. Esta clase contiene un conjunto de notas e información varia de la barra, así como los métodos para manejarla.

**Melody:** como se mencionó antes una melodía es un conjunto de barras. Esta clase contiene un conjunto de objetos de la clase barra y otros valores musicales como el número de barras, entre otros. También contiene los métodos para manejar melodías.

**Helpers:** es una clase estática que guarda las variables y constantes que funcionan como parámetros de configuración musical y evolutiva del programa.

En la clase Note también se crearon los enumerados “NoteType” (Altura musical), “Octave” (Octava), “Figure” (Figura musical) para manejar la información musical de la nota.

Para el desarrollo del módulo “Generador” se creó la clase **MelodyGenerator**, es una clase estática con métodos para generar melodías, barras y notas aleatorias.

Igualmente para el desarrollo del módulo “Reproductor musical” se creó la clase **SequencePlayer**, es una clase que instancia un reproductor MIDI usando la librería C# MIDI toolkit. Esta clase cuenta con métodos para reproducir, detener, guardar, cargar melodías y para crear una secuencia y agregarla al reproductor dado un objeto del tipo **Melody**.

Finalmente, para el desarrollo del módulo “Evolutivo” se crearon dos clases:

**Composition:** esta clase es un envoltorio a la clase melodía y representa un individuo de la población. Contiene las variables de información evolutiva del individuo y los métodos que manejan la aptitud y las partes del individuo como tal.

**Evolver:** esta clase implementa el algoritmo genético usado en el problema. Posee los métodos que permiten cruzar, mutar y seleccionar individuos, así como los métodos que manejan el reemplazo de las poblaciones y el manejo de la aptitud.

En cuanto a la parte gráfica se implementaron 4 formularios de “Windows Forms” o pantallas:

**MainForm:** es la pantalla principal, del programa, muestra los individuos de la población actual. Permite seleccionar un individuo y mostrar sus detalles.

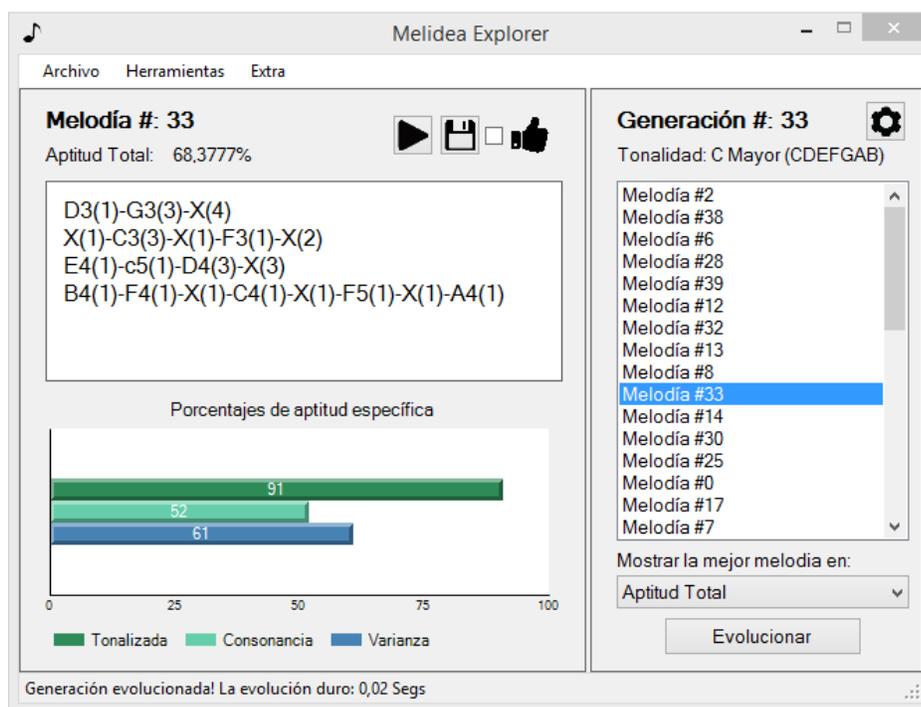
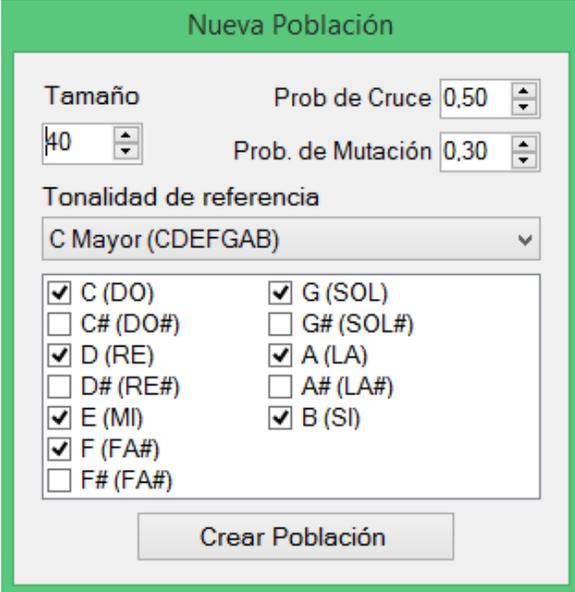


Figura 5.2: MainForm: pantalla principal de la aplicación

**InitializerForm:** es la primera pantalla del programa, la cual inicializa los parámetros para crear la primera población.



**Nueva Población**

Tamaño: 40

Prob de Cruce: 0,50

Prob. de Mutación: 0,30

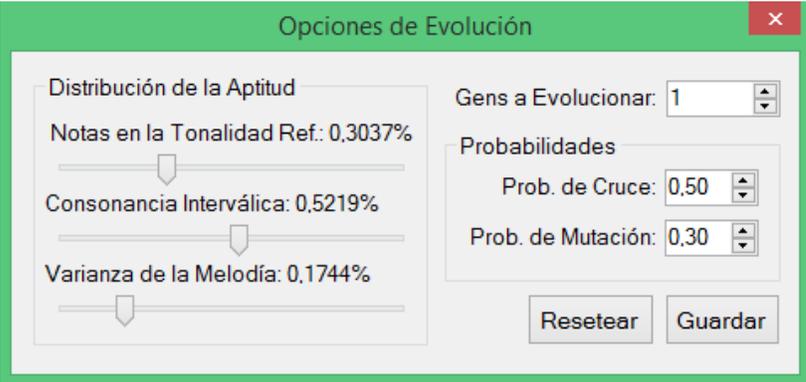
Tonalidad de referencia: C Mayor (CDEFGAB)

C (DO)       G (SOL)  
 C# (DO#)     G# (SOL#)  
 D (RE)         A (LA)  
 D# (RE#)      A# (LA#)  
 E (MI)         B (SI)  
 F (FA#)  
 F# (FA#)

Crear Población

Figura 5.3: InitializerForm: pantalla para crear una nueva población

**EvolverSettingsForm:** es una pantalla de configuración, que permite cambiar los parámetros operativos del algoritmo genético.



**Opciones de Evolución**

Distribución de la Aptitud

Notas en la Tonalidad Ref.: 0,3037%

Consonancia Interválica: 0,5219%

Varianza de la Melodía: 0,1744%

Gens a Evolucionar: 1

Probabilidades

Prob. de Cruce: 0,50

Prob. de Mutación: 0,30

Resetear    Guardar

Figura 5.4: EvolverSettingsForm: Pantalla de configuración de la aplicación

**TestingForm:** es una pantalla de pruebas que permite puntuar melodías de un conjunto de prueba de acuerdo a su calidad musical.

### 5.1.4. Diagrama de Clases y Formularios

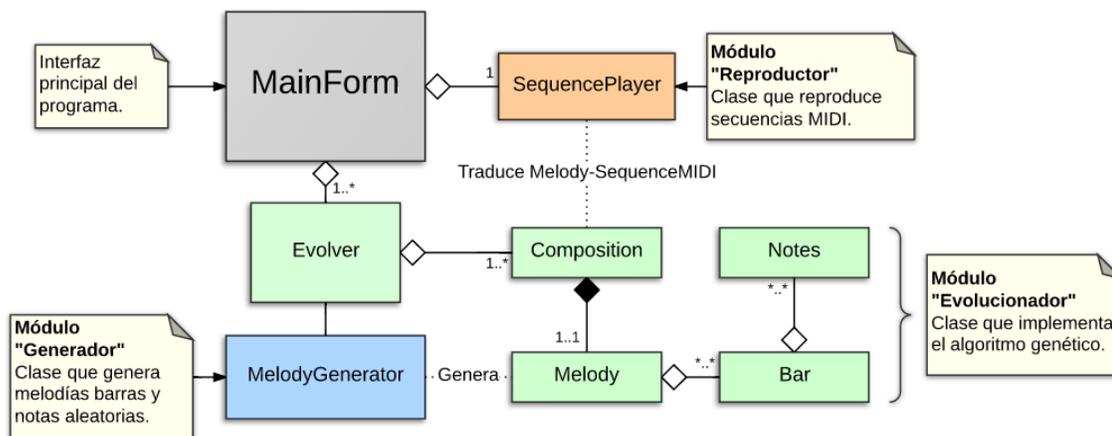


Figura 5.5: Diagrama de clases y formularios de la aplicación

En la figura 5.5 se puede ver el diagrama de clases del programa. Para facilidad de interpretación se incluyeron solo las clases principales del programa.

Puede observarse cada uno de los módulos implementados en el programa a modo de clases y la manera directa en que se relacionan. Un “Evolver” tiene entonces  $x$  “Composition”, cada una de la cual tiene un “Melody” que a su vez están compuestas por  $b$  “Bar”, con una cantidad específica de “Note”. Estas melodías son creadas por la clase “MelodyGenerator” al momento de crear un “Composition”.

En todo momento el “MainForm” puede usar el “SequencePlayer” para reproducir las “Melody” de la población actual del “Evolver”.

## 5.2. Módulo “Generador”

Este módulo se encarga de generar aleatoriamente melodías completas o barras o notas. Constituye la base de creación de datos utilizados por el programa.

### 5.2.1. Implementación

Fue el primer módulo que se implementó en el programa. Este módulo fue desarrollado como una clase estática que contuviera métodos que generaran notas, barras y melodías.

Para el primer prototipo de la aplicación, se desarrolló un generador aleatorio de strings usando la estructura de representación similar a la de Matic [6] para el manejo de melodías. Las notas eran representadas con el cifrado musical inglés: c(do), d(re), e(mi), f(fa), g(sol), a(la), b(si). También se utilizó el símbolo X para denotar los silencios y el símbolo S para denotar los sostenidos.

Las funciones principales para ese prototipo eran la función **RandomNote** la cual devolvía una nota aleatoria, un silencio o un sostenido, y la función **RandomMelody** la cual usaba la función **RandomNote** para obtener un número de notas específico y crear una melodía.

Luego para el próximo prototipo se tuvo que modificar este método para que generara melodías con una nueva representación (melodía  $\implies$  barra  $\implies$  nota) y para eso se implementó la función **RandomBar** que genera barras usando la función **RandomNote** y la función **RandomMelody** ahora usaría esta función **RandomBar** para generar sus barras.

De acuerdo a la nueva representación presentada en la sección 5.3.2, la función que genera las barras, debe generar una barra con un conjunto de notas que tuvieran una duración válida. Para esto se utilizó un conjunto de “permutaciones de duraciones” pre-generado por un programa hecho en Python. Estos archivos que contienen todas estas permutaciones se cargan en el programa mediante la clase “Helpers” y estos datos son usados para generar una barra eligiendo una permutación aleatoria y creando sus notas con dichas duraciones.

También se implementaron funciones que generaran barras y notas, fuera del proceso de creación de melodías.

### 5.3. Módulo “Evolutivo”

Es el módulo principal de la aplicación. Se encarga de conectar todos los módulos y formularios del programa para generar una población de individuos, y proveer las opciones necesarias para su evolución.

#### 5.3.1. Estructura del Algoritmo Genético

Antes de comenzar a desarrollar se definieron con claridad los componentes necesarios que conforman al algoritmo genético. Se definió entonces:

- **Representación:** implementación similar a la representación propuesta por Matic [6]
- **Función de Aptitud:** cantidad de notas dentro de una tonalidad especificada. Se utilizó la tonalidad C mayor como tonalidad de referencia (DO-RE-MI-FASOL-LA-SI-DO)
- **Estrategia de selección:** método de selección por ruleta
- **Cruce:** Cruce de un solo punto
- **Mutación(es) :**
  - **Mutación de una nota:** Se cambia una nota por otra aleatoriamente
  - **Mutación de una octava:** Se cambia la octava de una nota por otra octava aleatoriamente
- **Estrategia de reemplazo de la población:** Reemplazo completo de la población, ya sea por los padres seleccionados si no hay cruce ni mutación, o por los hijos luego de pasar por algún operador genético

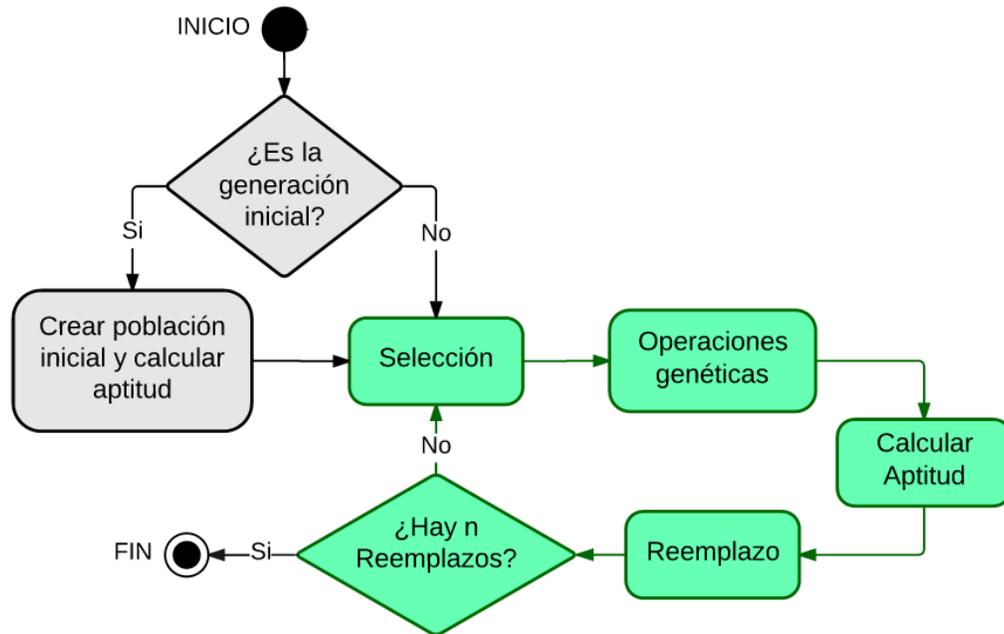


Figura 5.6: Diagrama de flujo del algoritmo genético

El diagrama de flujo del algoritmo genético implementado se puede observar en la figura 5.6. A continuación se expondrá con detalle el proceso de desarrollo e implementación de cada componente del algoritmo genético así como los diversos cambios que se hicieron para mejorar el desempeño de la aplicación.

### 5.3.2. Representación

Inicialmente para la representación cromosómica de los individuos de la población, se decidió utilizar la representación de melodías ya establecida para el generador en la sección 5.2, la cual fue la propuesta por Matic en [6].

En esta representación se utilizan números del 1 al  $n$  para representar  $n$  semitonos continuos desde la primera nota. También se usa el número 0 para representar un silencio, y se usa el número  $n+1$  para representar la prolongación de una nota o un silencio. Todos estos símbolos tenían como duración un tiempo “Shortest Length” calculado y definido con respecto al compás definido (Time Signature).

Para este prototipo se replicó esta representación, casi por completo. Se utilizaron enumerados de strings para denotar las diferentes notas musicales, los silencios y sostenidos, y enteros para denotar sus octavas. Las notas son representadas con el cifrado musical inglés: c(do), d(re), e(mi), f(fa), g(sol), a(la), b(si). También se utilizó el símbolo X para denotar los silencios y el símbolo S para denotar los sostenidos.

Una melodía según este esquema era entonces un conjunto de  $n$  notas, donde  $n$  es determinado por:

$$n = p * m \quad (5.1)$$

Siendo  $p$  el número de “Shortest Length” en una barra y  $m$  el número de barras en la melodía. En la figura 5.7 se puede observar una melodía bajo este esquema de representación.

**f4-g3-X-S-S-D3-C3-A2**  
**X-B4-S-S-A3-F2-S-X**  
**S-S-A4-G2-S-D3-S-X**  
**C3-d4-X-F3-a2-S-S-C3**

Figura 5.7: Primera representación. Cada línea es una barra de la melodía

Este esquema se utilizó en el primer prototipo para comenzar a evaluar la calidad musical de las melodías generadas. La idea del programa es que generara composiciones musicales básicas pero que a su vez al ser escuchadas por el usuario, este no percibiera la sensación del ruido de muchos sonidos rápidos y consecutivos y aleatorios. Con este prototipo, solo se obtenían melodías aleatorias de esta índole, insatisfactoria musicalmente, aun cuando se aplicaba una evolución de muchas generaciones.

Esto se debía a que con la representación utilizada hasta ese momento, era más aleatoriamente probable generar notas cortas independientes, que notas más largas que la duración mínima, ya que para eso se necesitaba generar una nota y múltiples “sostenidos” seguidos a ella. En teoría la probabilidad de aparición de una nota con duración  $x$  debería ser la misma que la probabilidad de aparición de una nota con duración  $y$ .

Se decidió entonces reformular la estructura de representación de una melodía. La nueva representación consta de:

- Un enumerado de tipo string, **NoteType** que define la altura de la nota
- Un enumerado de tipo entero, **Octave** que define la octava de la nota
- Un enumerado de tipo entero, **Figure** que define la duración de una nota

Las melodías bajo este esquema se representaron entonces como conjunto de  $b$  barras predefinido, donde cada barra tiene  $n$  notas cada una con una duración  $d_i$  tales que:

$$\sum^n d_i = N \quad (5.2)$$

Siendo  $N$  el número de “Shortest Length” en una barra. En la figura 5.8 se puede observar una melodía bajo este esquema de representación.

Para este cambio se decidió realizar una reestructuración de código que afectó el modelo de clases utilizado en el proyecto. Se cambiaron entonces las clases:

- **“Note”**: Ahora esta clase manejaba una duración. Esto cambió toda la representación al agregar melodías de tamaño variable.

D4(1)-X(1)-A5(2)-D4(1)-X(1)-D5(2)  
 X(3)-G4(1)-G2(1)-X(2)-D4(1)  
 A3(1)-D2(1)-G5(1)-F4(4)-A2(1)  
 F2(6)-C4(2)

Figura 5.8: Segunda representación. Los números entre paréntesis representan duraciones

- **“Melody”**: se separó en 2 clases: una clase “Bar”, la cual contiene un conjunto de notas, y “Melody” la cual contiene un conjunto de “Bar”. Esto permitió conservar el manejo estable de melodías de igual “tamaño” (tomando por tamaño la cantidad de barras).
- Todas las demás clases del programa para que funcionaran con la nueva clase “Melody” y la nueva clase “Note”.

### 5.3.3. Aptitud

Para el primer prototipo, se puntuó mejor a los individuos que tuviesen un mayor número de sus notas dentro de una debida tonalidad elegida. Se comenzó usando la tonalidad C mayor como tonalidad de referencia (DO-RE-MI-FA-SOL-LA-SI-DO). El valor de aptitud se discretizó a [0-1] para su manejo como un valor porcentual.

Con esta función de aptitud, se obtenían melodías con el 100% de aptitud muy rápidamente, y aparte de tener una buena aptitud, esta no representaba necesariamente que la melodía era musicalmente buena o agradable.

Se procedió a implementar una nueva estrategia con respecto a la función de aptitud. Se creó una función de aptitud global multi-objetivos distribuida por pesos, compuesta de sub-funciones de aptitud internas. La idea final era que en el programa el usuario fuera capaz de distribuir estos pesos a su conveniencia para generar composiciones más aptas de acuerdo a una función de aptitud determinada.

Teniendo ya definida la función de aptitud “**numero de notas dentro de una tonalidad dada**”, se implementaron dos nuevas sub-funciones:

- **El Índice de Consonancia Interválica:** según esta función un individuo es más apto mientras existan más intervalos musicales consonantes o disonantes resueltos, y es menos apto mientras existan más disonancias no-resueltas en la melodía.
- **La varianza de las notas y duraciones:** Es más apto si la composición no varía mucho ni varía poco con respecto a sus notas y a las duraciones de sus notas.

El implementar esto hizo que las melodías tuvieran un mayor valor musical a medida que van evolucionando. También se implementó una pantalla de configuración que permitía al usuario manipular la distribución de los pesos de las distintas funciones de aptitud con respecto la aptitud global.

### Índice de Consonancia Interválica (TCI)

Este fue calculado tomando en cuenta las relaciones de consonancia y disonancia de intervalos musicales internos presentes en la melodía. Las reglas de consonancia y disonancia utilizadas son expuestas por Danhauser [19] en su libro “Teoría de la Música”. Para calcularlo se hayan todas las relaciones de consonancia y disonancia, y se les asigna un puntaje de acuerdo a un esquema de puntuación como el usado por Matic [6]. Las puntuaciones son asignadas como se indica en la tabla 5.1.

En principio se utilizó la puntuación 1 al ser una distribución directamente proporcional a la importancia descrita por Danhauser [19], sin embargo luego se cambio esta puntuación por la segunda, dado que la aparición constante de octavas e unísonos tiende a afectar negativamente la calidad de las composiciones.

Tipos de intervalos	Puntaje 1	Puntaje 2
Unísono y Octava (perfectas)	1	2
Consonancias perfectas	1	1
Consonancias imperfectas	2	1
Consonancias mixtas	2	2
Disonancias atractivas	3	3
Disonancias Resueltas	3	3
Disonancias Normales	4	4
Notas con más de una octava de diferencia	5	5

Tabla 5.1: Puntuaciones de aptitud para el índice de consonancia total.

Luego el índice de consonancia interválica es calculado de usando:

$$TCI(m) = \frac{\sum CV_i}{N_{intervals}} \quad (5.3)$$

en donde  $m$  es la melodía dada, y  $C_i$  es el valor de consonancia dado por la tabla 5.1. Cabe acotar que este valor es discretizado a  $[0,1]$  para cuestiones de manejo porcentual y facilidad de uso.

### Varianza total de la melodía (TMV)

Este valor constituye un aproximado de que tan “variada” es una composición. Este valor se calcula tomando en cuenta la varianza de los valores MIDI existentes en toda la melodía en general, así como la varianza de los diferentes valores de duración que tiene cada nota de la melodía. Para normalizar el valor de las varianzas al intervalo  $[0,1]$ , se calculan sus coeficientes de Pearson,

Las varianzas son denotadas por  $Var$  y el coeficiente de Pearson por  $CP$ . Las

fórmulas para estas varianzas vienen dadas por:

$$Avg = \frac{\sum_{i=0}^n V_i}{n} \quad (5.4)$$

$$Var = \frac{\sum_{i=0}^n (V_i - Avg)^2}{n} \quad (5.5)$$

$$SD = \sqrt{Var}, \quad (5.6)$$

$$CP = \frac{SD}{Avg} \quad (5.7)$$

en las cuales,  $Avg$  es la media de cada caso, y  $SD$  la desviación estándar. El valor  $V$  representa el valor correspondiente al caso, es decir, el Valor MIDI de la nota o la duración respectivamente.

Para finalizar luego estos valores se mapean con una función básica para obtener un valor más cercano a 1 cuando estos valores son cercanos a la mitad del intervalo. La función utilizada es la siguiente:

$$TMV = 1 - \left| \frac{((CP_{tono}) + (1 - CP_{dur}))}{2} + 0,75 \right| \quad (5.8)$$

en donde  $CP_{tono}$  y  $CP_{dur}$  son los coeficientes de Pearson para el caso de los valores MIDI o tonos, y las duraciones, respectivamente.

### 5.3.4. Selección

Para fines del primer prototipo se implementó una selección por ruleta o “Roulette Selection”. Este método tiene como desventaja que se selecciona con mucha más frecuencia a los individuos más aptos que a los menos aptos al momento de cruzar.

Esto es bueno para problemas en los cuales se requiera encontrar rápidamente el mejor individuo de una población. Sin embargo para este caso, se requiere que el problema no converja de una manera muy rápida hacia individuos “muy aptos”, ya que pese a que un individuo sea muy apto, puede que la combinación de notas musicales

elegidas no sea satisfactoria para el usuario, en otras palabras no debe sacrificarse la exploración por la optimización.

Por esta razón, para el segundo prototipo del programa se implementaron nuevos métodos de selección que dieran más cabida a la exploración y abriesen un poco más el espectro de posibilidades para el usuario, a la vez que trataran de mejorar las soluciones más óptimas.

La estrategia de selección que mejores resultados obtuvo en la práctica para este fin, fue la estrategia de selección por torneo, o “Tournament selection”. Para esta estrategia se seleccionó una ventana de torneo fijo de 4 individuos, evitando con esto la aparición de muchos individuos aleatorios. Sin la disminución de la calidad de la población se protegió mediante las estrategias de reemplazo, explicadas en la sección [5.3.6](#).

### 5.3.5. Operadores Genéticos

El proceso de prototipaje con respecto a los operadores genéticos solo contó con creación de nuevos métodos y alteración de los ya existentes. Se describirá entonces el proceso y sus resultados tanto para el cruce como para las mutaciones.

#### Cruce

En investigaciones previas a este trabajo, como la de Matic [6], se menciona que un conjunto de mutaciones aporta más al intercambio genético que el cruce, para el área musical. Sin embargo a fines de conservar el concepto de algoritmo genético, para el primer prototipo se implementó un cruce simple de 1 punto.

Luego al ejecutar cambios en la representación del individuo, como se vio en la sección [5.3.2](#), se cambió ese cruce para cruzar usando las barras como punto de referencia en vez de las notas.

Como una última optimización se implementó un cruce multipuntos usando como puntos las barras de la melodía, es decir, un cruce “multi-barras”. Esto fue realizado a fines de mejorar la convergencia que aportaba un esquema de cruce tan rígido como el cruce simple de un solo punto.

## Mutación

Se generó una función de mutación central que recibe un individuo, calcula si debe o no debe mutar, y si debe mutar elige el tipo de mutación a utilizar. En este sentido el esquema de mutaciones es muy variado entre generaciones, logrando con esto una mayor variedad del material evolutivo.

Se implementaron las siguientes mutaciones:

- **Mutación de la altura de una nota:** Se cambia la altura de una nota por otra altura escogida aleatoriamente. Esto puede afectar a la consonancia interválica de la composición.
- **Mutación por una octava superior o inferior:** Se cambia la octava de una nota por su octava superior o inferior. Esto se hace con el fin de posiblemente corregir los valores de consonancia interválica.
- **intercambio entre notas vecinas:** Se realiza un intercambio entre 2 notas internas a una barra. Esta mutación afecta los valores de consonancia interválica de los intervalos vecinos.
- **Mutar el contenido interno de una barra:** Se cambia una barra por una nueva barra aleatoria. Esta mutación puede parecer un poco drástica con respecto a la calidad de los individuos de la población, ya que puede deteriorar un individuo prometedor. Sin embargo, este problema se ve contrarrestado con las estrategias de reemplazo seleccionadas y descritas en la sección [5.3.6](#).

### 5.3.6. Reemplazo

Al comienzo del desarrollo se eligió utilizar un esquema de reemplazo completo de la población, ya sea por los padres seleccionados si no hay cruce ni mutación, o por los hijos luego de pasar por algún operador genético.

Para esto se creaba un nuevo contenedor para guardar la población nueva, y se realizaban selecciones, cruces (de ser posible), mutaciones (de ser posible), y se guardaban esos individuos, hasta tener tantos individuos como la población pasada. La aplicación de esta estrategia generaba una sobrepoblación de cada generación los individuos seleccionados (más aptos) y sus hijos.

La estrategia de reemplazo se cambió entonces a Steady-State, reemplazando el 70 % de los individuos de la población, solo en el caso de que el individuo hijo resultante fuera más apto que el peor individuo de la población, y no existiese ya en la población actual.

Con este nuevo prototipo de la aplicación, la población estuvo por fin conformada por solo individuos diferentes, y las melodías generadas comenzaron a ser satisfactorias, pero el problema de la convergencia se conservó. En este caso se obtenían individuos “súper-aptos” en muy pocas generaciones.

En este punto del desarrollo fue cuando se descubrió que la convergencia de la población en muchos individuos bastante parecidos, venía dada por la función de selección. Se cambió a selección por torneo, como fue descrito en la sección 5.3.4, y esto mejoró la variedad en los individuos, pero gracias a esto, los mejores individuos de la población podían perderse y ser ignorados durante la selección.

Este problema se solucionó con una estrategia de reemplazo Elitista, la cual conserva el 10 % de los individuos más aptos y solo reemplaza aleatoriamente los demás. Utilizando esta estrategia se preservó algo de la calidad de la población, pero a su vez se promueve la diversidad y se evita la convergencia a un determinado individuo.

## 5.4. Módulo “Reproductor Musical”

Es el módulo que se encarga de transformar un objeto melodía en otro objeto que pueda ser reproducido y almacenado como un sonido MIDI. Este módulo es de vital importancia ya que para la evaluación de melodías por parte del usuario es requerido poderlas escuchar.

### 5.4.1. C# Midi Toolkit

Para el desarrollo de este módulo se hizo uso de la librería C# MIDI Toolkit diseñada por Leslie Sanford [20]. Este API contiene clases y métodos que implementan secuenciadores MIDI, y permiten enviar y recibir mensajes MIDI al controlador de la PC.

Las clases y métodos que se usaron con más frecuencia y sus usos son descritas a continuación a modo de referencia

**ChannelMessage:** Clase que representa un mensaje MIDI. Es la unidad atómica de reproducción y puede representar tanto un sonido como un comando que se envía al controlador.

**Track:** Clase para implementar una pista MIDI, la cual es simplemente una colección de objetos del tipo **ChannelCommand**. Para agregar un ChannelCommand al track, se hace uso del método “Insert” de la misma librería.

**Sequence:** Clase para representar una canción MIDI. Está compuesta por varios objetos del tipo **track** definidos. Para agregar un track a la secuencia, se hace uso del método “Add” de la misma librería.

**Sequencer:** Clase para implementar un reproductor MIDI, capaz de reproducir objetos del tipo **sequence**.

**OutputDevice:** Clase para el manejo de la interfaz MIDI de la PC. Es necesaria instanciarla para que la computadora reproduzca los sonidos.

**Métodos Play/Stop/Continue:** Permiten el manejo de la reproducción y pausado de la secuencia por medio del secuenciador.

**Métodos Load/Save:** cargar y guardar una secuencia a un archivo de tipo MIDI para su posterior reproducción fuera del programa.

### 5.4.2. Implementación

Este módulo fue desarrollado de la mano con el módulo “Generador” para probar las “melodías” que este crea. Contiene métodos que son simples re-implementaciones de los métodos de la librería, adaptándolo a los estándares del programa.

Por otro lado, se creó un método que permite los objetos de la clase “Melody” obtenido del módulo ‘Generador” a secuencias Sanford. Esto lo hace recorriendo cada nota de la melodía, analizando la duración y su valor MIDI, y utilizando las funciones de la librería para crear una secuencia que envíe los mensajes MIDI correspondientes.

## Capítulo 6

# Resultados

Con el fin de obtener resultados, se creó un módulo de pruebas que permitió contrastar varias distribuciones de aptitud y sus resultados en melodías con las preferencias de múltiples usuarios.

En este capítulo se presentan los resultados de este módulo y se analiza la factibilidad de la propuesta. También se muestran algunos valores musicales obtenidos y posibles mejoras y recomendaciones para trabajos futuros.

### 6.1. Ambiente de Trabajo

Antes de mostrar los resultados de la investigación, es importante denotar el entorno de trabajo en el cual fue desarrollada y probada la aplicación. Las pruebas se realizaron bajo un ambiente con las siguientes especificaciones:

- Sistema operativo: Windows 8.1 - 64 bits
- Procesador: Intel Core i5 650
- Memoria RAM: 4GB DDR3
- Tarjeta Gráfica: Nvidia Geforce GTX 750ti
- Tarjeta de Sonido: Integrada

Se utilizó Microsoft Visual Studio 2013 como entorno de desarrollo y C# como lenguaje de programación. Se utilizó el API de Windows Forms para la creación de la interfaz gráfica del programa. Como librerías externas solo se utilizó C# MIDI Toolkit desarrollada por Leslie Sanford [20] para el uso y manejo de la interfaz MIDI así como la reproducción y creación de sonidos en este estándar.

## 6.2. Módulo de pruebas

Fue el último módulo requerido y desarrollado. Tiene como objetivo generar la data de prueba del programa, para contrastar la función de aptitud usada en el módulo de evolución, contra la opinión de los usuarios.

Para el desarrollo de este módulo se implementó principalmente 2 funcionalidades: la creación de un lote de melodías de prueba y la ejecución de una prueba.

### 6.2.1. Creador de casos de prueba

Para crear un lote de melodías de pruebas se crearon varios “evolucionadores” variando distintos parámetros de distribución de aptitud y se evolucionaron exactamente 150 generaciones en cada uno.

Luego se extrae de cada una de las poblaciones resultado el mejor individuo, el individuo representando la mediana de la población, y el peor individuo.

Todo esto se ordena en una estructura de carpetas específica que luego el módulo de prueba puede consultar y utilizar para realizar las pruebas. Adicionalmente se crea un archivo que guarda los valores de aptitud para cada individuo y que también sirve para guardar los resultados de las pruebas a medida que se vayan ejecutando.

En la figura 6.1 se puede observar la estructura de carpetas necesaria así como el archivo de datos de pruebas (.tests).

Carpeta de pruebas		Carpeta de cada caso
Nombre	Fecha de modifica...	Nombre
 case_001	11-04-2015 03:07 a...	 Best.mid
 case_010	11-04-2015 03:07 a...	 Median.mid
 case_011	11-04-2015 03:06 a...	 Worst.mid
 case_100	11-04-2015 03:06 a...	
 case_101	11-04-2015 03:06 a...	
 case_110	11-04-2015 03:06 a...	
 case_111	11-04-2015 03:06 a...	
 case_ideal	11-04-2015 03:07 a...	
 .tests	11-04-2015 12:23 ...	

Figura 6.1: Carpetas de casos de prueba

### 6.2.2. Pantalla de pruebas de usuario

Es una pantalla especial diseñada para probar varios grupos de melodías generados previamente. En esta pantalla se le presenta al usuario varios botones de reproducción para que el escuche cada una de las melodías y le dé una puntuación a cada una dependiendo de que tanto considera el que esa melodía puede ser utilizada en un contexto real.

Las melodías pueden ser puntuadas del 1 al 5, teniendo que 1 representa una melodía desordenada, caótica, y de poco valor musical, y 5 representa una buena melodía, agradable al usuario.

Luego de puntuar todas las melodías del grupo, el usuario podrá pasar a realizar pruebas al siguiente grupo y así sucesivamente durante los 8 grupos a probar. Luego de finalizar, el programa guardara los resultados en el archivo de datos que se encuentra en el directorio de las melodías.

En la figura 6.2 puede verse la interfaz de pruebas. Se utilizó el objeto trackbar para discretizar la puntuación del usuario del 1 al 5.



Figura 6.2: Pantalla de pruebas de usuario

### 6.3. Resultados cuantitativos

La aplicación se probó cuantitativamente en cuanto al tiempo de ejecución de la evolución de generación. Para una población de 40 individuos, se evoluciono 100 veces independientemente, y se midió el tiempo de ejecución de cada una de las evoluciones. Los resultados pueden ser observados en la tabla 6.1

Tiempo de ejecución	1 generación	50 generaciones
Mínimo	0.016 s	0.795 s
Promedio	0.022 s	0.88 s
Máximo	0.036 s	0.97 s

Tabla 6.1: Mediciones de tiempos de ejecución para 40 individuos

En general puede observarse que los tiempos de ejecución son considerablemente bajos, inclusive en el caso de evolucionar múltiples generaciones a la vez. Esto se considera ideal para un programa interactivo.

## 6.4. Resultados cualitativos

Con el fin de contrastar la calidad de la función de aptitud del algoritmo genético, se realizaron pruebas con 20 usuarios de distinta clase, ninguno de ellos con amplia experiencia musical. Se les pidió a los usuarios que a través de la pantalla de prueba escucharan 8 grupos de melodías, y que a cada melodía de cada grupo le diesen una puntuación del 1 al 5.

Grupo	Función 1 (Tonalidad)	Función 2 (Consonancia)	Función 3 (Varianza)
100	1	0	0
010	0	1	0
001	0	0	1
110	0.5	0.5	0
011	0	0.5	0.5
101	0.5	0	0.5
111	0.3	0.4	0.3
Ideal	0.35	0.5	0.15

Tabla 6.2: Distribuciones de aptitud utilizadas para crear casos de prueba

Los grupos de melodías de prueba fueron creados usando el sistema de distribución de pesos para la aptitud que se puede ver en la tabla 6.2. El esquema ideal se utilizó debido a los buenos resultados que este otorgaba durante el desarrollo, en su mayoría para probar los resultados que esta distribución arrojaba.

Los resultados de las pruebas ejecutadas pueden verse en la tabla 6.3. Se obtuvo los puntajes en el rango [1,5] de 20 usuarios, y se obtuvo el promedio entre todos. Luego se obtuvo con esto un porcentaje de aptitud promedio que puede ser contrastado con el valor de aptitud estimado por el programa.

En la figura 6.3 puede observarse la comparación entre ambos valores de aptitud. Pese a que la opinión del usuario es un valor bastante subjetivo, Se observa que los

Grupo	Melodía	Aptitud	Puntuacion personas	Promedio	Aptitud (Usuarios)
G1	111_Best	90,30%	3 5 3 4 3 3 3 1 2 4 2 4 1 2 3 4 2 1 3 1	2,7	54,00%
	111_Median	74,84%	1 3 4 5 3 4 2 2 2 2 1 1 2 1 4 2 1 1 2 1	2,2	44,00%
	111_Worst	54,91%	2 3 2 5 4 1 4 4 4 3 4 4 3 3 5 3 1 3 4 1	3,15	63,00%
G2	110_Best	100,00%	2 2 5 5 4 3 1 3 3 5 2 4 3 3 5 4 2 2 4 1	3,15	63,00%
	110_Median	79,17%	1 5 4 5 4 4 5 4 3 4 3 5 3 2 4 5 2 2 2 2	3,45	69,00%
	110_Worst	42,06%	3 4 3 5 4 2 3 4 3 4 2 4 4 3 4 3 2 3 3 2	3,25	65,00%
G3	101_Best	91,88%	3 5 5 5 4 3 2 3 3 5 3 2 3 4 4 4 2 4 5 2	3,55	71,00%
	101_Median	90,41%	4 5 3 5 4 4 3 1 2 5 2 1 3 2 4 5 2 2 1 2	3	60,00%
	101_Worst	67,10%	4 3 2 5 5 2 2 4 2 5 2 2 3 3 5 5 2 4 4 2	3,3	66,00%
G4	011_Best	85,45%	4 4 5 5 3 1 2 5 3 3 2 2 4 2 4 5 2 5 4 2	3,35	67,00%
	011_Median	67,13%	2 4 4 4 3 3 4 4 2 4 3 1 3 4 3 5 1 2 2 3	3,05	61,00%
	011_Worst	44,27%	3 4 3 5 2 4 2 3 2 2 4 2 3 3 4 4 1 5 1 1	2,9	58,00%
G5	100_Best	100,00%	1 1 4 3 2 1 1 3 2 4 3 1 2 2 4 4 1 1 3 1	2,2	44,00%
	100_Median	100,00%	1 2 4 5 3 4 2 2 2 4 2 1 1 4 2 1 1 1 4 1	2,35	47,00%
	100_Worst	64,71%	2 1 3 5 2 3 1 2 2 1 1 1 1 3 3 3 1 3 2 2	2,1	42,00%
G6	010_Best	100,00%	1 3 2 5 3 2 5 5 2 2 1 1 2 4 3 5 1 5 5 2	2,95	59,00%
	010_Median	75,00%	2 3 4 5 5 1 2 4 2 3 1 2 4 4 3 5 1 5 4 2	3,1	62,00%
	010_Worst	26,09%	2 2 3 4 4 5 2 1 2 1 2 4 3 3 4 2 1 4 1 2	2,6	52,00%
G7	001_Best	96,10%	2 2 4 3 3 4 1 4 3 2 1 1 3 4 3 2 1 4 3 2	2,6	52,00%
	001_Median	82,76%	1 2 3 4 5 3 1 2 3 2 1 1 4 3 3 4 1 3 2 3	2,55	51,00%
	001_Worst	60,29%	2 2 4 4 4 3 1 3 2 2 1 1 3 2 4 2 1 2 2 4	2,45	49,00%
G8	ideal_Best	94,78%	4 4 4 5 3 5 2 4 3 2 1 3 3 4 4 3 2 4 5 5	3,5	70,00%
	ideal_Median	80,54%	5 4 4 5 4 4 2 2 1 1 2 4 3 4 3 4 2 4 3 3	3,2	64,00%
	ideal_Worst	52,20%	2 4 3 5 5 2 2 4 2 1 1 3 2 3 5 2 2 2 2 5	2,85	57,00%

Tabla 6.3: Datos resultados de la prueba del programa en usuarios

valores que se adhieren más fielmente al valor dictado por el programa son proporcionalmente los del caso ideal.

Ocurren casos como el del grupo 100 los valores el programa le dan a los individuos una aptitud del 100 %, más esto solo significa que para estas composiciones tienen todas sus notas dentro de la tonalidad especificada. Esto le confiere un juicio de menor nivel que el que otorga la función referente a la consonancia interválica. Esto puede observarse al ver que para este casi se obtienen en promedio los resultados más bajos.

También puede ser denotada la subjetividad del caso de estudio al observar que el caso que mejor tuvo apreciación por parte de los usuarios, fue la mejor melodía del

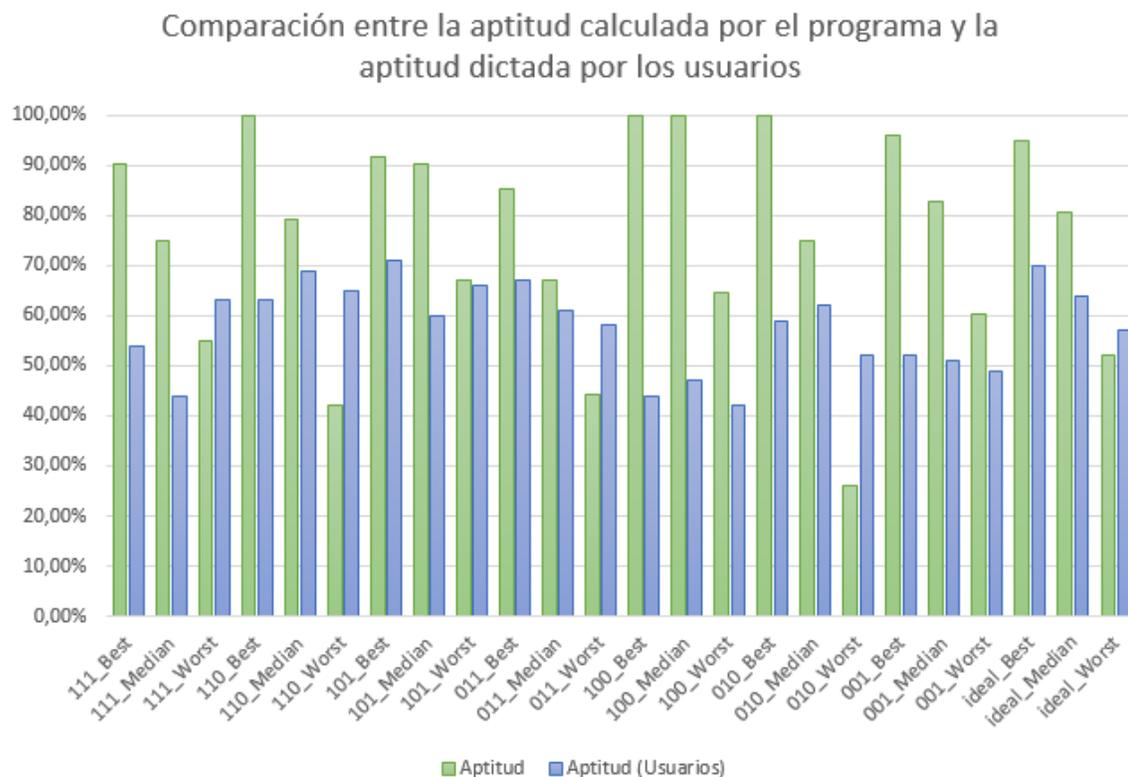


Figura 6.3: Comparación entre las aptitudes

caso 101, superando inclusive a la mejor del caso ideal y del caso que incluye a las 3 funciones de aptitud.

### 6.5. Resultados musicales

Con respecto a las melodías obtenidas, también es importante observar un poco los resultados que se ha logrado obtener, al ser estos bastante diversos e interesantes.

El algoritmo se corrió con la distribución ideal de pesos para las aptitudes, y se eligió el mejor individuo de la generación inicial (sin evolucionar), de la generación número 50, y de la generación número 100, los 3 individuos resultantes pueden apreciarse en la figura 6.4.

The figure displays three musical staves, each representing a different stage of an evolutionary process. Each staff begins with a tempo marking of a quarter note equal to 120. The top staff, titled "Best-Initial", shows a melody with several dissonant intervals, including notes on different octaves. The middle staff, titled "Best-50", shows a melody with more defined intervals and some coherence. The bottom staff, titled "Best-100", shows a melody with improved consonance and a more structured flow.

Figura 6.4: Representación de un individuo luego de varias etapas de evolución

Puede observarse como el individuo en la primera generación cuenta con notas que pese a estar ubicada una al lado de la otra, se encuentran varias octava una por encima de la otra, representando esto intervalos disonantes. Sin embargo esto es válido tomando en cuenta que este es un individuo generado aleatoriamente, que no ha sido sometido al proceso de evolución.

Para la generación 50 se observa una mejora notable en las características de los individuos. Los intervalos están mejor definidos y se empieza a observar más coherencia entre notas y silencios.

Ya en la 100 se tiene un ejemplo con un individuo que tiene como base el mismo individuo de la generación 50. La evolución entonces lo que logro fue variar los intervalos musicales de la melodía variando sus notas y logrando mejorar su consonancia. Pero si se escuchan, ambas pueden ser relativamente buenas.

Debido a esto se puede decir que una buena aproximación a una generación en la cual los individuos ya cuentan con una calidad aceptable es después de la generación

50. También se concluye que aparte de la calidad musical, el juicio entre si una melodía es mala o buena, depende de las necesidades del usuario final.

## Capítulo 7

# Conclusiones y trabajos futuros

### 7.1. Conclusiones

En este trabajo se implementó un sistema evolutivo semi-interactivo para la creación y evolución de Ideas melódicas o motivos musicales. El sistema le permite a un usuario observar y manipular la evolución de melodías de acuerdo a parámetros evolutivos y musicales

Puede decirse entonces que se logró el objetivo general planteado como solución al problema de la alta demanda de contenido musical básico.

Para la generación de las melodías se creó un esquema de representación que toma en cuenta características musicales varias como la altura musical, la octava y la figura de una nota, así como métricas de composición musical como el compás y el número de barras de la melodía. Con respecto a la evaluación de las mismas, se tomó en cuenta las relaciones interválicas de consonancia y disonancia de notas y distintos cálculos de varianza y preferencias tonales dictadas por el usuario.

Con respecto al algoritmo genético, Se implementó un algoritmo genético que usa una estrategia de selección por torneo, un esquema de aptitud multi-objetivo con participación opcional del usuario, operadores genéticos variados con preferencia en la mutación sobre el cruce, y una estrategia de reemplazo elitista. Todo este esquema está enfocado en la disminución maximizada de la convergencia de los individuos, sin

que eso represente que se pierda la calidad evolutiva del algoritmo, logrando así que el usuario siempre tenga opciones para explorar y elegir a medida que el algoritmo evoluciona y cambia.

Se realizaron pruebas con varios individuos y en las cuales se les pedía que puntuaran melodías generadas con distintas corridas inicializadas con diversos valores de distribución aptitud y se contrastaron los resultados en promedio con los resultados generales del algoritmo. Pese a que un campo de estudio como lo es la música, es un de naturaleza inherentemente subjetiva, se puede observar que se obtuvo resultados bastante satisfactorios en cuanto a las preferencias del usuario común.

## 7.2. Trabajos futuros y recomendaciones

Sobre la base del proceso de desarrollo y los resultados obtenidos en el presente trabajo, es posible derivar ideas, o propuestas de nuevas investigaciones útiles e interesantes. Algunas de ellas pueden ser:

- Implementación mejorada del algoritmo evolutivo utilizando estrategias de selección diferentes y probando diversos esquemas.
- Implementación de funciones de mutación y cruce que sean inteligentes musicalmente y generen contenido apropiado musicalmente. Pese a que esto puede llegar a ser limitativo, puede proveer de resultados interesantes.
- Inclusión un generador musical más potente y menos aleatorio usando otros métodos como cadenas de Markov y gramáticas regulares. Así el programa tendría una inicialización más musicalmente interesante.
- Con respecto a la parte musical, un muy valioso aporte puede ser el manejo del ritmo como una variable que pueda ser administrada y controlada en los individuos. Esto agregaría un potencial y un valor musical incalculable al resultado del programa.

- Implementación de controles para el manejo de variables MIDI, como los instrumentos, el tempo, el metrónomo entre otras cosas.
- También valdría la pena implementar una interfaz más amigable al usuario y menos técnica, que muestre una representación en partitura de la melodía.

## Bibliografía

- [1] Philip Ball. *El instinto musical: Escuchar, pensar y vivir la música*. Turner Publicaciones S.L. Madrid, 2012.
- [2] Jose David Fernández and Francisco Vico. AI methods in algorithmic composition: A comprehensive survey. *Journal of Artificial Intelligence Research*, 48:513–582, 2013.
- [3] John H Holland. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press, 1975.
- [4] Melanie Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.
- [5] David Edward Goldberg et al. *Genetic algorithms in search, optimization, and machine learning*, volume 412. Addison-wesley Reading Menlo Park, 1989.
- [6] Dragan Matić. A genetic algorithm for composing music. *The Yugoslav Journal of Operations Research ISSN: 0354-0243 EISSN: 2334-6043*, 20(1), 2013.
- [7] Randy L Haupt and Sue Ellen Haupt. *Practical genetic algorithms*. John Wiley & Sons, 2004.
- [8] John Biles. Genjam: A genetic algorithm for generating jazz solos. In *Proceedings of the International Computer Music Conference*, pages 131–131. International Computer Music Association, 1994.

- [9] Geraint A Wiggins, George Papadopoulos, Somnuk Phon-Amnuaisuk, and Andrew Tuson. Evolutionary methods for musical composition. *DAI RESEARCH PAPER*, 1998.
- [10] Carlos A Coello Coello and Col San Pedro Zacatenco. Introducción a la computación evolutiva. *Notas del curso. Departamento de Ingeniería Eléctrica, Sección de Computación, Instituto Politécnico Nacional, México*, 2004.
- [11] Peter JB Hancock. An empirical comparison of selection methods in evolutionary algorithms. In *Evolutionary Computing*, pages 80–94. Springer, 1994.
- [12] Fubito Toyama, Yukihiro Fujiki, Kenji Shoji, and Juichi Miyamichi. Assembly of puzzles using a genetic algorithm. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 4, pages 389–392. IEEE, 2002.
- [13] Jason M Barnes, Shaddi H Hasan, and Sanghwi Lee. Solving the 8-puzzle: A genetic programming approach. 2006.
- [14] Harsh Bhasin and Neha Singla. Genetic based algorithm for n-puzzle problem. *International Journal of Computer Applications*, 51, 2012.
- [15] Juan S Guevara. *Teoría de la música*. 2010.
- [16] Roy Bennett. *Léxico de música*, volume 4. Ediciones Akal, 2003.
- [17] J.J. Sgroi. Midi sound designer with randomizer function, October 18 1994. US Patent 5,357,048.
- [18] Ronald Hochreiter. Audible convergence for optimal base melody extension with statistical genre-specific interval distance evaluation. In *Applications of Evolutionary Computing*, pages 712–716. Springer, 2006.
- [19] A.L. Danhauser. *Teoría de la música, por A. Danhauser... traducida... por G. J. Llompart...* E. Lemoine, 1897.
- [20] Leslie Sanford. C# midi toolkit, 2007.

