

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA  
**UNIVERSIDAD**  
**CATÓLICA**  
DEL PERÚ

## COMPOSICIÓN MUSICAL ARTIFICIAL CON ALGORITMOS GENÉTICOS

Tesis para optar por el Título de Ingeniero Informático, que presentan los bachilleres:

**Francisco Vílchez Vargas**

**José Astuvilca Fuster**

**ASESOR: Dr. César BELTRÁN CASTAÑÓN**

Lima, Junio del 2015

## *Resumen del proyecto de fin de carrera*

La composición musical consiste en el uso y combinación adecuada de diversas notas, con el propósito de generar una melodía musical. Por otro lado, la improvisación musical, consiste en la composición instantánea, realizada en un momento determinado, ya sea en base a una composición ya existente o alguna totalmente nueva. Para realizar dicha tarea, se requiere el conocimiento de diversos patrones existentes en teoría musical y, de esta manera, poder juntarlas teniendo en consideración ciertos criterios armónicos, melódicos y rítmicos para que suene agradable al usuario final. Debido a la cantidad de conocimiento necesario para poder realizar una composición musical, nuestro proyecto tiene la intención de realizar un método que tenga la capacidad de poder combinar notas adecuadamente siguiendo las reglas de la teoría musical moderna. Este compositor (improvisador) tendrá como entrada una armonía y adaptará sus creaciones (improvisaciones) en base a ella. Se logrará realizar dicha tarea mediante un algoritmo genético. Para ello, se utilizará como población inicial pequeñas frases musicales creadas por expertos, las cuales son conocidos como licks. Estas serán juntadas y mejoradas mediante nuestros operadores genéticos, de tal manera que se logre una nueva composición totalmente distinta a la inicial cada vez que se aplique nuestro algoritmo. De esta manera, se desarrollará una metodología basada en algoritmos genéticos, incrementando así el estado del arte del área y finalmente disponer de una herramienta de ayuda y autoaprendizaje computarizada para este proceso de composición musical.

FACULTAD DE  
**CIENCIAS E  
INGENIERÍA**  
ESPECIALIDAD DE  
INGENIERÍA INFORMÁTICA



PONTIFICIA  
**UNIVERSIDAD  
CATÓLICA**  
DEL PERÚ

**TEMA DE TESIS PARA OPTAR EL TÍTULO DE INGENIERO INFORMÁTICO**

**TÍTULO:** COMPOSICIÓN MUSICAL ARTIFICIAL CON ALGORITMOS GENÉTICOS

**ÁREA:** CIENCIAS DE LA COMPUTACIÓN

**PROPONENTE:** Dr. César Beltrán Castañón

**ASESOR:** Dr. César Beltrán Castañón

**ALUMNO Y:** **Francisco Reynaldo Vilchez Vargas** 20080434

**CÓDIGO:** **Jose Daniel Astuvilca Fuster** 20089019

**TEMA N°:** 592

**FECHA:** San Miguel, 26 de mayo del 2015



**DESCRIPCIÓN**

La computación musical es un área de investigación que, entre otros objetivos, busca simular la creatividad artística de los humanos. Para ello se busca crear herramientas de software que puedan componer música a partir de información basada en teoría musical. Los métodos más comunes para implementar una solución para este problema están basados en el uso de Algoritmos Genéticos, Algoritmos Genéticos Interactivos y Cadenas de Markov.

Estas soluciones cuentan con algunas restricciones como por ejemplo: la continua intervención del usuario en la calificación de las melodías creadas; por otro lado, otro tipos de soluciones que intentan realizar una creación sin información previa, no cuentan con resultados muy agradables al oído. Estas restricciones impiden que los resultados sean eficientes. Las soluciones presentadas generalmente se basan en conocimiento previamente almacenado, pero el cual no logra incrementarse después de nueva creación realizada, lo cual puede volver algunas de las creaciones un tanto repetitivas. Además, los trabajos presentan algunas limitaciones a las creaciones que impide se asemeje a la creatividad humana, como por ejemplo, una secuencia de acordes precisa para su correcto funcionamiento, una cantidad límite de notas por compás, lo cual limita un máximo total de notas en una canción, además de una base de datos de conocimientos previo limitada, cuyo valor no se incrementa. Restricciones que se intenta mejorar en el presente trabajo.

El presente proyecto de fin de carrera busca implementar una herramienta que logre realizar una combinación adecuada de notas e identificación de patrones musicales para la próxima generación de la melodía. Además, tendrá un enfoque musical muy amplio, el cual tendrá muchas ventajas educativas para el usuario final. Existen diversos métodos de optimización combinatoria, pero la propuesta consiste en aplicar Algoritmos Genéticos para aproximar la búsqueda de estas melodías. Se utiliza este tipo de algoritmos debido a que en la composición musical se busca que el paso entre frases musicales se realice de una forma fluida y no sea notada por el oyente final



Av. Universitaria 1801  
San Miguel, Lima - Perú



Apartado Postal 1761  
Lima 100 - Perú

Teléfono:  
(511) 626 2000 Anexo 4801



FACULTAD DE  
**CIENCIAS E  
 INGENIERÍA**  
 ESPECIALIDAD DE  
 INGENIERÍA INFORMÁTICA

 PONTIFICIA  
**UNIVERSIDAD  
 CATÓLICA**  
 DEL PERÚ

mediante la minimización de la distancia entre notas musicales, y además, la creación nuevas frases musicales para la composición. Estos dos puntos mencionados son generados en los procesos de casamiento y mutación, razón por la cual se realiza la propuesta mencionada para este proyecto de fin de carrera.

#### OBJETIVO GENERAL

Desarrollar un software de composición musical automático basado en un modelo computacional, para la generación de música artificial que apoye a la enseñanza de improvisación a músicos novatos.

#### OBJETIVOS ESPECÍFICOS

Los objetivos específicos del presente proyecto son:

- Objetivo 1: Implementar el almacenamiento de datos de acordes con sus respectivas escalas restringidas a un conjunto de notas.
- Objetivo 2: Diseñar e implementar un repositorio que permita el almacenamiento de melodías previas para generar data inicial.
- Objetivo 3: Realizar la interfaz para lectura de la armonía de una composición para su manipulación y procesamiento.
- Objetivo 4: Construcción del modelo computacional que permitirá generar la composición musical, a partir de las armonías.
- Objetivo 5: Presentar los resultados a músicos novatos en improvisación musical.

#### ALCANCE

El presente proyecto está orientado a realizar una herramienta con la capacidad de poder generar melodías (improvisaciones) a partir de una armonía, ya dada, mediante el uso de un algoritmo genético, la cual servirá de apoyo para estudiantes de música y compositores profesionales. Para ello, se tuvo en cuenta el siguiente alcance:

- La cantidad de géneros específicos de música popular que abarca el proyecto estará limitada por las 200 frases musicales, del libro "1001 Licks", que serán escogidas para el proyecto.
- El proyecto considera un análisis armónico y melódico de la composición ingresada como reglas para poder limitar las melodías que se generen a través del algoritmo genético.
- El proyecto no está enfocado a realizar un entorno de composición. Se le brindará las ayudas y herramientas básicas que permita ingresar la armonía de la composición que el usuario desee.

Por otro lado, se tendrán en consideración las siguientes limitantes durante el desarrollo del proyecto:

- El proyecto no está enfocado en realizar un reproductor musical. La reproducción incluida en la herramienta sirve como una sencilla pre-



 Av. Universitaria 1801  
 San Miguel, Lima – Perú



 Apartado Postal 1761  
 Lima 100 – Perú

 Teléfono:  
 (511) 626 2000 Anexo 4801



FACULTAD DE  
**CIENCIAS E  
INGENIERÍA**  
ESPECIALIDAD DE  
INGENIERÍA INFORMÁTICA



PONTIFICIA  
**UNIVERSIDAD  
CATÓLICA**  
DEL PERÚ

visualización de la melodía generada, que permite a los músicos principiantes poder tener una guía mucho sencilla.

- El proyecto se encuentra limitado a la creación de melodías para composiciones en compás cuaternario de subdivisión binaria (4/4). Cualquier otro tipo de compás no está cubierto por el proyecto.
- La herramienta está orientada a la creación de melodías para música popular. La música 'clásica', la cual según la Real Academia Española recorre los períodos barroco, clasicismo y romanticismo, no están contempladas en el proyecto.
- La herramienta no se encarga de la creación de una composición instrumental completa. Está enfocada a realizar una composición melódica que vaya de acuerdo a la armonía de alguna canción seleccionada.

La distribución de responsabilidad del proyecto es la siguiente:

Responsable	Objetivos específicos
Francisco Reynaldo Vilchez Vargas	OE01 y OE02 y OE5
Jose Daniel Astuvilca Fuster	OE03 y OE04 y OE5

*Máximo: 100 páginas*



Av. Universitaria 1801  
San Miguel, Lima – Perú



Apartado Postal 1761  
Lima 100 – Perú

Teléfono:  
(511) 626 2000 Anexo 4801



# Índice general

<b>Resumen</b>	<b>I</b>
<b>Tema de tesis</b>	<b>II</b>
<b>1. Generalidades</b>	<b>1</b>
1.1. Problemática	1
1.1.1. Objetivo General	3
1.1.2. Objetivos Específicos	3
1.1.3. Resultados Esperados	4
1.2. Herramientas, metodologías y procedimientos	4
1.2.1. Introducción	4
1.2.2. Herramientas	5
1.2.2.1. HTML5	5
1.2.2.2. Euphony	6
1.2.2.3. jMusic	6
1.2.2.4. Java	6
1.2.2.5. MySQL	7
1.2.2.6. MuseScore	7
1.2.3. Métodos y Procedimientos	7
1.2.3.1. Metodologías para el producto	7
1.3. Alcance	9
1.3.1. Limitaciones	9
1.3.2. Riesgos	10
1.4. Justificación y viabilidad	11
1.4.1. Justificativa	11
1.4.2. Viabilidad	12
<b>2. Marco Conceptual</b>	<b>13</b>
2.1. Conceptualización	13
2.1.1. Introducción	13
2.1.2. Objetivos del Marco Conceptual	13
2.1.3. Conceptos Musicales	13
2.1.3.1. Música	13
2.1.3.2. Nota	14

2.1.3.3. Acorde . . . . .	15
2.1.3.4. Escala . . . . .	15
2.1.3.5. Lick . . . . .	16
2.1.3.6. Composición Musical . . . . .	16
2.1.4. Conceptos de Optimización Combinatoria . . . . .	18
2.1.4.1. Problema NP . . . . .	18
2.1.4.2. Problema NP-Completo . . . . .	18
2.1.4.3. Algoritmos de Aproximación . . . . .	19
2.1.4.4. Reconocimiento de Patrones . . . . .	19
2.1.4.5. Algoritmos Genéticos . . . . .	19
2.1.4.6. Cadenas de Markov . . . . .	20
2.2. Estado del Arte . . . . .	20
2.2.1. Aplicaciones en Reconocimiento de Patrones . . . . .	20
2.2.2. Aplicaciones de Algoritmos Genéticos . . . . .	21
2.2.2.1. GenJam . . . . .	21
2.2.2.2. Drangan Mátic's GA for composing music . . . . .	22
2.2.3. Aplicaciones de las Cadenas de Markov . . . . .	23
2.2.3.1. Redes Neuronales . . . . .	23
2.2.4. Conclusiones sobre el estado del arte . . . . .	24
<b>3. Generación de población inicial . . . . .</b>	<b>25</b>
3.1. Introducción . . . . .	25
3.2. Método de almacenamiento de Acordes Musicales con su escala respectiva . . . . .	25
3.2.1. Proceso de almacenamiento . . . . .	26
3.2.2. Ventajas del almacenamiento de la teoría musical . . . . .	29
3.3. Método de almacenamiento de Frases Musicales . . . . .	30
3.3.1. Representación del cromosoma . . . . .	30
3.3.2. Proceso de almacenamiento . . . . .	34
3.4. Modelo de Base de Datos . . . . .	36
3.5. Estructuras de datos utilizadas . . . . .	37
3.6. Discusión . . . . .	38
3.6.1. Almacenamiento de teoría musical . . . . .	38
3.6.2. Almacenamiento de frases musicales . . . . .	38
<b>4. Modelo Computacional que permita generar la composición musical . . . . .</b>	<b>39</b>
4.1. Introducción . . . . .	39
4.2. Modelo Computacional . . . . .	39
4.2.1. Función de fitness . . . . .	39
4.2.2. Pseudocódigo . . . . .	42
4.2.3. Inicialización de la población . . . . .	43
4.2.4. Torneo de selección . . . . .	44
4.2.5. Operadores genéticos . . . . .	44
4.2.5.1. Casamiento inteligente . . . . .	45

4.2.5.2. Mutación . . . . .	46
4.3. Discusión . . . . .	50
<b>5. Presentación de resultados a los usuarios finales</b>	<b>51</b>
5.1. Introducción . . . . .	51
5.2. Interfaz para la lectura de la armonía para su manipulación y procesamiento . . . . .	51
5.3. Generación de archivos MIDI de la creación del algoritmo y del acompañamiento . . . . .	53
5.4. Reproducción musical en la herramienta desarrollada . . . . .	55
5.5. Representación musical de la composición y sugerencias . . . . .	56
5.6. Arquitectura de la herramienta web desarrollada . . . . .	58
5.7. Verificación de los resultados . . . . .	60
5.8. Discusión . . . . .	62
5.8.1. Interfaz para el ingreso de armonía . . . . .	62
5.8.2. Intefaz para la reproducción de la música generada . . . . .	62
<b>6. Conclusiones y trabajos futuros</b>	<b>63</b>
6.1. Introducción . . . . .	63
6.2. Observaciones . . . . .	63
6.3. Resultados . . . . .	64
6.3.1. Workshops . . . . .	64
6.3.2. Composiciones disponibles . . . . .	69
6.3.3. Análisis musical de "Alice in Wonderland" . . . . .	71
6.4. Conclusiones . . . . .	74
6.5. Recomendaciones y trabajos futuros . . . . .	74
<b>Bibliografía</b>	<b>76</b>

# Capítulo 1

## Generalidades

### 1.1. Problemática

La composición musical es el arte de ordenar los sonidos con el fin de crear una determinada emoción en el oyente [1]. Con el desarrollo de la tecnología computacional, se han desarrollado muchas herramientas con el fin de otorgar la habilidad de poder tocar y manipular una gran cantidad de sonidos, este es el caso de los instrumentos electrónicos que tienen música pre-programada [2]. En las últimas tres décadas se intentó buscar algoritmos que puedan crear música a partir de diversos métodos, entre ellos: algoritmos randomizados, búsqueda de patrones y sistemas basados en conocimientos previos [3] [4]. Un caso particular es poder componer una melodía a partir de una secuencia armónica dada [3], similar a lo realizado por los músicos cuando improvisan. En otras palabras, cuando existe una secuencia de acordes predefinida y la melodía inventada debe adaptarse a ella siguiendo las reglas teóricas de música moderna.

La computación musical es un área de investigación que, entre otros objetivos, busca simular la creatividad artística de los humanos [5]. Para ello se busca crear herramientas de software que puedan componer música a partir de información basada en teoría musical. Los métodos más comunes para implementar una solución para este

problema están basados en el uso de Algoritmos Genéticos, Algoritmos Genéticos Interactivos y Cadenas de Markov[6] [4].

El enfoque de cadenas de Markov se basa en métodos estadísticos, que intentan clasificar cada una de las notas con una probabilidad de forma que "suenen" en una determinada melodía; de esta manera, se procederá a utilizar las notas que tengan un mayor puntaje, luego se procede a combinarlos de acuerdo a los criterios estadísticos, y de esta forma se generará una melodía [4]. Otro método, es el de los algoritmos genéticos, el cual consiste en generar notas de forma aleatoria y juntarlas para que formen una pequeña frase de la canción. Estas frases serán mejoradas mediante un algoritmo genético y la respectiva calificación del usuario [6]. Sin embargo, en estos métodos expuestos, los resultados que se obtienen logran ser poco factibles (debido al tiempo y la cantidad de intervención humana necesaria) para la generación de melodías agradables, además del poco uso (casi nulo) de teoría musical en la generación de éstas.

Estas soluciones cuentan con algunas restricciones como por ejemplo: la continua intervención del usuario en la calificación de las melodías creadas, lo cual es uno de los cuellos de botella de la solución implementada por Biles[7]; por otro lado, otro tipos de soluciones que intentan realizar una creación sin información previa, no cuentan con resultados muy agradables al oído humano [4]. Estas restricciones impiden que los resultados sean eficientes. Las soluciones presentadas generalmente se basan en conocimiento previamente almacenado, pero el cual no logra incrementarse después de nueva creación realizada, lo cual puede volver algunas de las creaciones un tanto repetitivas. Además, los trabajos presentan algunas limitaciones en las creaciones que impiden se asemejen a la creatividad humana, como por ejemplo, una secuencia de acordes precisa para su correcto funcionamiento, una cantidad límite de notas por compás, lo cual limita un máximo total de notas en una canción, además de una base de datos de conocimientos previo limitada, cuyo valor no se incrementa. Restricciones que se intenta mejorar en el presente trabajo.

En conclusión, el presente proyecto de fin de carrera busca implementar una herramienta que logre realizar una combinación adecuada de notas e identificación de patrones musicales para la próxima generación de la melodía. Además, tendrá un enfoque musical muy amplio, el cual tendrá muchas ventajas educativas para el usuario final. Existen diversos métodos de optimización combinatoria, pero la propuesta consiste en aplicar Algoritmos Genéticos para aproximar la búsqueda de estas melodías. Se utiliza este tipo de algoritmos debido a que en la composición musical se busca que el paso entre frases musicales se realice de una forma fluida y no sea notada por el oyente final mediante la minimización de la distancia entre notas musicales, y además, la creación nuevas frases musicales para la composición. Estos dos puntos mencionados son generados en los procesos de casamiento y mutación[8], razón por la cual se realiza la propuesta mencionada para este proyecto de fin de carrera.

### 1.1.1. **Objetivo General**

Desarrollar un software de composición musical automático basado en un modelo computacional, para la generación de música artificial que apoye a la enseñanza de improvisación a músicos novatos.

### 1.1.2. **Objetivos Específicos**

Los objetivos específicos del presente proyecto son:

- Objetivo 1: Implementar el almacenamiento de datos de acordes con sus respectivas escalas restringidas a un conjunto de notas.
- Objetivo 2: Diseñar e implementar un repositorio que permita el almacenamiento de melodías previas para generar data inicial.
- Objetivo 3: Realizar la interfaz para lectura de la armonía de una composición para su manipulación y procesamiento.

- Objetivo 4: Construcción del modelo computacional que permitirá generar la composición musical, a partir de las armonías.
- Objetivo 5: Presentar los resultados a músicos novatos en improvisación musical.

### 1.1.3. Resultados Esperados

Los resultados esperados del presente proyecto son:

- Resultado 1 para el objetivo 1: Almacenamiento de acordes musicales mapeados con su escala respectiva.
- Resultado 2 para el objetivo 2: Repositorio que almacene las frases musicales, que sirvan como población inicial.
- Resultado 3 para el objetivo 3: Interfaz capaz de leer la armonía de la composición ingresada por el usuario, lista para su manipulación y procesamiento.
- Resultado 4 para el objetivo 4: Melodía generada con el algoritmo genético.
- Resultado 5 para el objetivo 5: Presentación visual y auditiva de los resultados obtenidos.

## 1.2. Herramientas, metodologías y procedimientos

### 1.2.1. Introducción

En esta sección se definirá las metodologías empleadas en la elaboración de las partes específicas del proyecto.

Objetivo	Resultado Esperado	Herramienta a utilizarse
Objetivo 1	Repositorio que almacene las frases musicales, que sirvan como población inicial.	MySQL: Se encargará del almacenamiento de la información previa. Java: Nos permitirá realizar una aplicación que haga del ingreso de información previa, una tarea más sencilla.
Objetivo 2	Estructura de datos de acordes musicales mapeados con su escala respectiva.	Java: Cargará la información previa
Objetivo 3	Armonía y melodía de la composición ingresada por el usuario almacenada correctamente para su manipulación y procesamiento	HTML: Permite la comunicación entre el usuario y la herramienta a través de una agradable interfaz
Objetivo 4	Escala más adecuada para la armonía de la composición, con las notas musicales que la conforman, lo cual servirá como filtro para la población inicial del algoritmo genético.	Java: Se encarga del manejo de las estructuras para el algoritmo
Objetivo 4	Población inicial del algoritmo, la cual es el resultado de las frases musicales iniciales filtradas, procesadas y organizadas.	Java: Se encarga del manejo de las estructuras para el algoritmo
Objetivo 4	Melodías más adecuada generada a través del algoritmo genético.	Java: Se encarga del manejo de las estructuras para el algoritmo
Objetivo 5	Presentación visual y auditiva de los resultados obtenidos.	Euphony: Se encarga de mostrar la melodía y reproducirla al usuario. jMusic: Se encarga de la generación del midi con la composición. MuseScore: Se encarga de la generación de la partitura musical a partir del MIDI

CUADRO 1.1: Mapeo Resultados vs. Herramientas a utilizarse.

## 1.2.2. Herramientas

### 1.2.2.1. HTML5

Es la quinta versión del lenguaje básico de la World Wide Web. Además, puede integrar elementos multimedia y contiene nuevos componentes para la funcionalidad de representaciones gráficas [9].

El uso de esta herramienta permitirá que el proyecto pueda ser utilizado por cualquier usuario, ya que brinda una experiencia multiplataforma. Además, brinda facilidades para la comunicación entre el usuario compositor y la herramienta.

#### **1.2.2.2. Euphony**

Euphony es una librería creada en HTML5 que permite reproducir archivos midis en un entorno web.

El uso de esta herramienta permitirá mostrar al usuario final la composición creada por la herramienta de una manera clara y fácil de comprender para usuario expertos o novatos. [10]

#### **1.2.2.3. jMusic**

Es un librería de audio multiplataforma para la creación de melodías en un tiempo y frecuencia específica[11].

El uso de esta herramienta permitirá crear archivos midi la melodía generada por el algoritmo al usuario final. De esta manera, podrá ayudar a los estudiantes a lograr entender de una manera más sencilla la composición generada.

#### **1.2.2.4. Java**

Es un lenguaje de programación orientado a objetos. Permite ejecutarse en cualquier máquina virtual Java sin importar la arquitectura de la computadora subyacente. Su paradigma de Orientado a Objetos nos permitirá organizar el código de forma eficiente y asimismo hacer escalable la aplicación.[12]

Es una herramienta de sencillo uso y basta experiencia en el tema. Además de su flexibilidad para cualquier tipo de proyecto.

#### 1.2.2.5. MySQL

Es el sistema de gestión de bases de datos relacional más usado en le área de desarrollo de software. [13]

El uso de este gestor de base de datos permitirá un rápido y eficaz acceso a los datos, asimismo, como la seguridad e integridad de datos esta garantizada. Además de su libre disponibilidad para proyectos académicos.

#### 1.2.2.6. MuseScore

MuseScore es una herramienta gratuita que permite la creación de partituras musicales. [14]

Es una herramienta que nos permite una creación de las partituras a partir del archivo MIDI generado, por lo cual está relacionado a los otros objetivos y simplifica el trabajo.

### 1.2.3. Métodos y Procedimientos

#### 1.2.3.1. Metodologías para el producto

Entrada: Armonía de una composición.

Proceso: Generación de melodías adecuadas a la armonía y melodía dada por el usuario.

Salida: Melodía para la composición.

A continuación, se detallará el proceso que realizará el proyecto de fin de carrera y su ilustración respectiva.

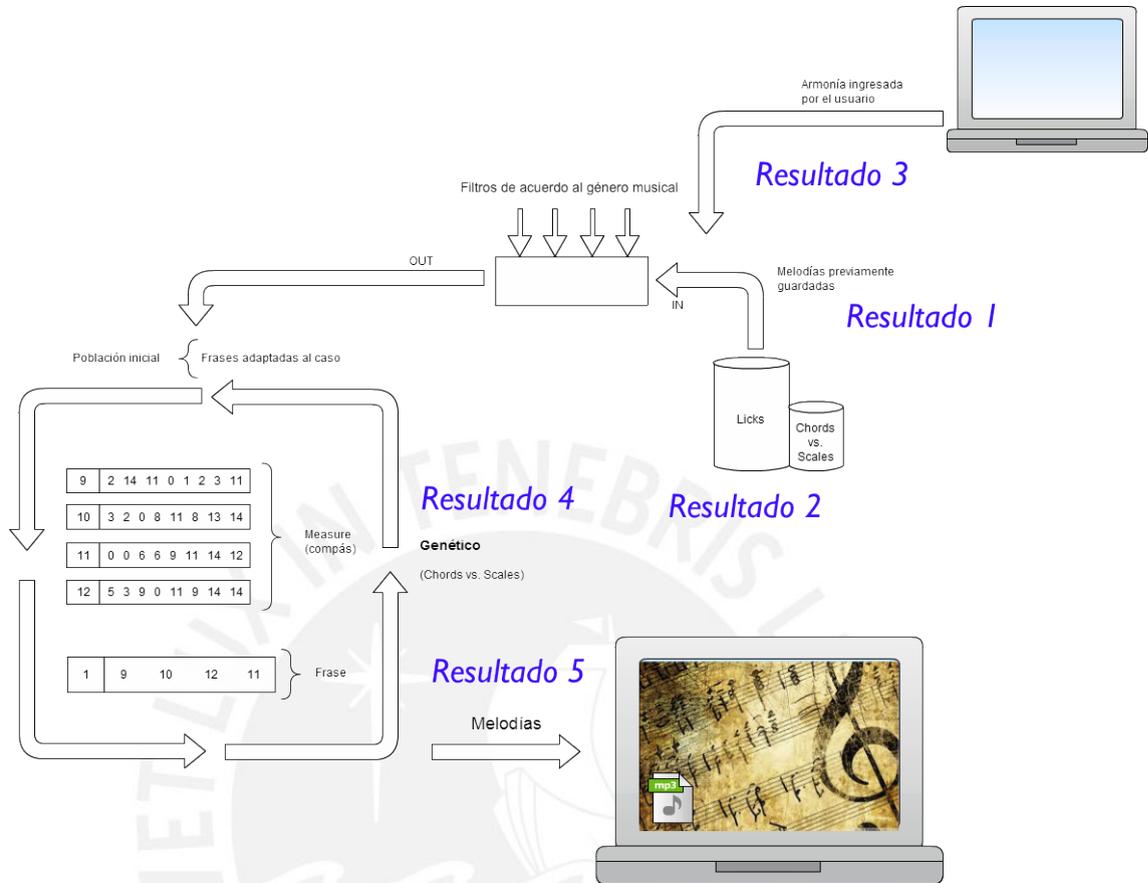


FIGURA 1.1: Metodología para el producto [7]

1. Construcción de un repositorio de datos, el cual servirá como estructura intermedia para el mapeo de los acordes 'bases' con su escala.
2. Construcción de un repositorio de frases musicales, las cuales servirán de población inicial para el algoritmo genético.
3. Se leerá las armonías ingresadas por el usuario para la generación de la improvisación.
4. Se procede a realizar la composición con el algoritmo genético. Para ello, se obtendrá la población inicial del repositorio de licks, las cuales son filtradas de acuerdo al género musical seleccionado por el usuario.
5. Finalmente, se entregan los resultados al usuario mediante un archivo midi y su contenido es visualizado a través de una interfaz.

### 1.3. Alcance

El presente proyecto está orientado a realizar una herramienta con la capacidad de poder generar melodías (improvisaciones) a partir de una armonía, ya dada, mediante el uso de un algoritmo genético, la cual servirá de apoyo para estudiantes de música y compositores profesionales.

- La cantidad de géneros específicos de música popular que abarca el proyecto estará limitada por las 200 frases musicales, del libro "1001 Licks"[15], que serán escogidas para el proyecto.
- El proyecto considera un análisis armónico y melódico de la composición ingresada como reglas para poder limitar las melodías que se generen a través del algoritmo genético.
- El proyecto no está enfocado a realizar un entorno de composición. Se le brindará la ayuda y herramientas básicas que permita ingresar la armonía de la composición que el usuario desee.

#### 1.3.1. Limitaciones

Para el presente proyecto se debe tener en consideración lo siguiente:

- El proyecto no está enfocado en realizar un reproductor musical. La reproducción incluida en la herramienta sirve como una sencilla pre-visualización de la melodía generada, que permite a los músicos principiantes poder tener una guía mucho sencilla.
- El proyecto se encuentra limitado a la creación de melodías para composiciones en compás cuaternario de subdivisión binaria (4/4). Cualquier otro tipo de compás no está cubierto por el proyecto.

- La herramienta está orientada a la creación de melodías para música popular. La música 'clásica', la cual según la Real Academia Española recorre los períodos barroco, clasicismo y romanticismo, no están contempladas en el proyecto.
- La herramienta no se encarga de la creación de una composición instrumental completa. Está enfocada a realizar una composición melódica que vaya de acuerdo a la armonía de alguna canción seleccionada.

### 1.3.2. Riesgos

Riesgo identificado	Impacto en el proyecto	Medidas correctivas para mitigar
Las melodías generadas no resultan ser conformes a los resultados esperados.	Al ser el objetivo principal de proyecto, generaría altos costos en tiempo.	Implementación de nuevos filtros y limitaciones a las melodías generadas, lo cual permita obtener una mejor calidad en las improvisaciones.
La cantidad de géneros musicales pertenecientes a la música popular no logrará ser cubierta en su totalidad.	Al ser un objetivo no vital en el proyecto, no generará altos costos.	Reducción en el alcance de los géneros pertenecientes a la música popular y focalización a un menor número de géneros.
El gestor de base de datos utilizado para el almacenamiento de la información previa no resulta ni eficiente ni eficaz en la solución del problema.	Al ser un objetivo vital, generará altos costos.	Investigación de un nuevo gestor que se adapte a las necesidades del proyecto y vaya de acuerdo a los objetivos planteados.

CUADRO 1.2: Riesgos identificados en el proyecto.

## 1.4. Justificación y viabilidad

### 1.4.1. Justificativa

El fin de este proyecto es implementar un software para la creación de melodías musicales que permitan a estudiantes novatos de música a aprender cómo improvisar. Esto se realizará mediante la combinación adecuada de notas y la identificación de patrones musicales. El problema de la composición automática es que requiere de la selección de diversos patrones existentes en teoría musical y poder juntarlos con cierto criterio armónico, melódico y rítmico para que suene agradable al oído humano. La intención del presente proyecto es realizar un software que pueda combinar notas adecuadamente siguiendo las reglas de la teoría musical moderna. Este compositor (improvisador) tendrá como base una armonía dada, y adaptará sus creaciones (improvisaciones) en base a ella.

Esta herramienta brindará distintas utilidades al usuario:

- En el caso del estudiante, le expondrá de manera clara las melodías generadas a partir de la armonía dada. Ya que podrá visualizar la combinación de notas generadas por el improvisador. Esto será mostrado en un pentagrama musical para que sea entendible por el músico. De esta forma, el alumno puede relacionar sus conocimientos previos con los expuestos por la máquina y aprender nuevas formas de combinaciones y métodos en composición.
- En el caso del compositor, podrá tener acceso a nuevas melodías musicales, las cuales podrá identificar rápidamente y usarlas de la manera que considere adecuada. De esta forma, el músico contará con una mayor cantidad de opciones disponibles para el desarrollo de su obra, lo que aportará mayor variedad en su repertorio musical.

Adicionalmente el trabajo permitirá demostrar la aplicación de técnicas informáticas en el campo de la música, con lo cual se amplía el marco de desarrollo aplicativo y la generación de herramientas poco convencionales.

### 1.4.2. Viabilidad

Este proyecto es viable como proyecto de fin de carrera, ya que además de contar con los conocimientos previos de Algoritmos Genéticos y composición musical, se ha identificado en el estado del arte una serie de investigaciones previas que tratan sobre la búsqueda de patrones en una composición musical.

En el aspecto aplicativo, la propuesta se muestra ad-hoc para efectos de enseñanza la improvisación musical, en vista que será de una herramienta que permita a los músicos novatos, contar con un tutor guía que les otorgue soporte.

Desde el punto de vista económico, este proyecto no demanda gran recurso, en vista que se orienta especialmente a la búsqueda de una solución computacional al problema de composición musical. En el caso que la propuesta demuestre ser óptima, esta podrá dar pie al desarrollo de una herramienta comercial que pueda generar réditos económicos. Desde el punto de vista social, ayudará al aprendizaje de estudiantes de música, especialmente en los colegios.

## Capítulo 2

# Marco Conceptual

### 2.1. Conceptualización

#### 2.1.1. Introducción

A continuación, se hace una breve descripción de los conceptos teóricos que se utiliza. Entre ellos se menciona definiciones sobre Teoría Musical, Algoritmos y Estructuras de Datos.

#### 2.1.2. Objetivos del Marco Conceptual

Con el presente marco conceptual se desea definir los conocimientos necesarios para poder comprender el tema desarrollado en este proyecto de tesis.

#### 2.1.3. Conceptos Musicales

##### 2.1.3.1. Música

La música cuenta con muchas definiciones, pero puede ser resumida en que es "el arte de ordenar los sonidos con el fin de crear una determinada emoción en el oyente"

[1]. Para lograr esta emoción en el oyente es necesario conocer algunos términos y procedimientos teóricos para la composición de una melodía, los cuales se intentará simular en el presente proyecto.

### 2.1.3.2. Nota

La nota representa un sonido musical [1]. Esta compuesta por ciertos atributos:

- Pitch (timbre): Es la propiedad que permite ordenar una nota según a una altura correspondiente.
- Tonality (tonalidad): Es el timbre jerárquico de una melodía, determina la relación de las notas hacia una nota (o acorde) central. Existen tonalidades mayores y menores, las cuales pueden ser interpretadas como alegres o tristes respectivamente.
- Melody (melodía): Es la combinación de timbre y ritmo. La unión de muchas notas genera una melodía o frase musical, al unir muchas de estas frases, se genera la melodía de la canción. En términos sencillos, es la parte de una canción que uno puede silbar.
- Harmony (armonía): Es el uso simultáneo de varios timbres y acordes. En otros términos, es un conjunto de reglas seguidas por un compositor para acompañar la melodía[4].



FIGURA 2.1: Ejemplo de nota musical  
[1]

### 2.1.3.3. Acorde

Un acorde es un "conjunto de 3 o más sonidos diferentes combinados armónicamente" [16]. Todos los acordes pertenecientes a una canción giran entorno a un acorde central, el cual es determinado por la tonalidad. Por lo tanto, cada acorde de la tonalidad tiene una intención distinta al resto (algunos pueden sonar más alegres, tristes o tensos con respecto al resto). En términos técnicos tiene una respectiva función armónica [1].

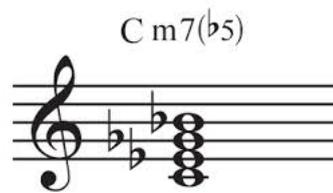


FIGURA 2.2: Ejemplo de acorde musical. La partes superior indica el nombre del acorde (Do menor 7 con quinta bemol)

[1]

### 2.1.3.4. Escala

La escala es una "sucesión de notas musicales" [16]. Cada acorde de la canción tiene asignada una escala correspondiente. Esta escala es la que limita las notas que se pueden utilizar en esa parte de la canción de tal forma que suene agradable al oído humano [1].



FIGURA 2.3: Ejemplo de la escala musical de Do mayor

[1]

### 2.1.3.5. Lick

Un lick es un patrón musical que consiste en una corta sucesión de notas que es frecuentemente utilizada en improvisaciones musicales. Se considera que dicha frase musical tiene un sonido agradable, por lo cual, una vez conocida, es usada por los músicos durante sus improvisaciones. [17]



FIGURA 2.4: Ejemplo de un lick usado constantemente en improvisaciones y composiciones [17]

### 2.1.3.6. Composición Musical

La composición musical consiste en la combinación específica de melodías, armonías y diversos ritmos musicales, de tal manera que genere una sensación agradable en el oyente [17]. El proceso de improvisación, consiste en realizar, de forma inmediata, una composición que se adecúe a una canción existente [7]. Para poder comprender mejor el proceso de composición, se comparará mediante el proceso de contar una historia cualquiera a través de un lenguaje, tomado de las clases de Gary Burton en Berklee College of Music basada en sus propias fuentes bibliográficas. Para ello, se tomará como ayuda el siguiente gráfico[17].



FIGURA 2.5: Lenguaje Hablado vs. Música Improvisada [17]

Para poder realizar, por ejemplo, la descripción de un objeto, uno comienza a recordar diversas palabras relacionadas a ello. En el caso de la música, lo que uno comienza a pensar son en los diversos sonidos que se pueden utilizar, identificando los acordes y escalas que se adecúen a la canción de acuerdo a la teoría. Después de ello, uno comienza a ordenar las palabras para que creen una oración, mientras que en el caso de la música, se comienza a unir los sonidos para que logren crear lo que se define como una frase musical. Finalmente, se procede a unir las oraciones creadas de tal forma que expliquen la historia de forma ordenada. Por otro lado, en el aspecto musical, se unen las frases de tal manera que expresen musicalmente un tema o una estructura composicional[17].

## 2.1.4. Conceptos de Optimización Combinatoria

### 2.1.4.1. Problema NP

Los problemas NP son aquellos que son verificables en tiempo polinomial. Por verificable se refiere a que podemos certificar que una solución dada es correcta en tiempo polinomial[18].

### 2.1.4.2. Problema NP-Completo

Son aquellos problemas que son reducibles a cualquier problema NP[18], es decir si son reducibles al problema circuit-SAT.[18]

Dado que la composición musical es el arte de agrupar y ordenar las armonías de manera adecuada, de tal forma que el resultado sea agradable al oído humano, se puede representar como una función de optimización de la siguiente forma:

Supongamos que tengamos a un compositor oyendo un conjunto de notas musicales, si esta le parece bien lo agregará a su composición, de no ser así, solo lo obviara. Si esto es llevado a lenguaje matemático lo podemos representar de la siguiente manera. Supongamos que se tiene una función objetivo, que en este caso será la armonía que se adecúe mejor a la melodía que se esta componiendo, y que cada una de las notas son la variables a cambiar. Entonces, todo esto se convierte en un problema de optimización, ya que constatemente el compositor buscara la mejor melodía para su composición.

Una vez que sabemos que el proceso de composición es un problema de optimización queda considerar lo siguiente, si se cuenta con una armonía dada y se sabe que de esta se pueden generar infinitas melodías, además entre acorde y acorde hay por lo menos X combinaciones; a eso agregar que en una canción por lo general participa más de un instrumento hace impracticable buscar un algoritmo en tiempo polinomial. [19]

Otra forma de confirmar que el proceso de composición musical es un problema que necesita de una solución aproximada o alguna meta-heurística es demostrando que la metodología de 'Especies de Contrapunto' pueden ser reducida al problema de 'Three-Dimensional Matching problem' que es un NP-Completo, y que también es un problema de optimización. [20]

#### **2.1.4.3. Algoritmos de Aproximación**

Son aquellos algoritmos que retornan una solución cercana al óptimo. Esto se encuentran asociados con los problemas NP-Completo debido a que estos aún no se sabe si cuentan con una solución de complejidad polinomial se buscan soluciones mediante aproximaciones, heurísticas, etc[18].

#### **2.1.4.4. Reconocimiento de Patrones**

Es la disciplina científica cuyo objetivo es el de la clasificación de objetos dentro de un número de categorías o clases, también llamado "Generalización". Otra de las aplicaciones de esta área es la de "extracción de características", que también es conocido como pre-procesamiento, lo que se busca es separar lo irrelevante del input y solo dejar lo importante. Otro es la de "regresión" que trata de asignar un valor real al input. [21]

#### **2.1.4.5. Algoritmos Genéticos**

Los algoritmos genéticos es un tipo de algoritmo de búsqueda basada en la Teoría Darwiniana de la Evolución. Fue propuesto por John Holland, profesor de la Universidad de Michigan, en la década de 1960. Su principal objetivo, en sus años de investigación, fue la de entender el fenómeno de "adaptación" que ocurre en la naturaleza y así poder llevarlo a sistemas computarizados. En 1975 en el libro "Adaptation in Natural and Artificial Systems" explicó los algoritmos genéticos.[22]

En los algoritmos genéticos se parte de una población de “cromosomas” (usualmente cadenas de bits), al cual se le aplica un conjunto de operadores( inspirados en la genética): Casamiento, Mutación e Inversión. Los cromosoma que tengan un mejor “fitness” serán aquellos que prevalecerán en el tiempo, mientras que los menos adaptados, no.

#### **2.1.4.6. Cadenas de Markov**

La Probabilidad Moderna se encuentra enfocada en el estudio de procesos aleatorios. Se basan en el estudio de los resultados anteriores, ya que estos influyen en los resultados de futuros experimentos. Es en 1907 que el matemático Andréi Márkov empezó el estudio de componentes aleatorios, los cuales próximamente serían conocidos como las Cadenas de Markov, que vendría a ser la formulación matemática que hace posible la predicción de los resultados. Existen una variedad de aplicaciones en disciplinas como Ingeniería, Economía, Investigación de Operaciones, etc. Un ejemplo, sería la predicción de notas de un alumno en una serie de exámenes a partir de su record histórico en un materia dada.[23]

## **2.2. Estado del Arte**

El objetivo de esta sección es la de presentar los trabajos realizados hasta el momento relacionados a la composición artificial de música contemporánea.

### **2.2.1. Aplicaciones en Reconocimiento de Patrones**

Las investigaciones actuales en Composición Musical se enfocan en el enfoque de encontrar patrones derivados de músicas que fueron ingresados como input. Una de las propuestas para la búsqueda de patrones fue la de utilizar técnicas de minería de

datos para descubrir reglas de composición musical, como se vió en el punto caracterizada por estructuras, melodías y motivos. Los doctores en Ciencias de la Computación Man-Kwan Shan y Shih-Chuan Chiu de la National Chiao-Tung University crearon un Sistema para encontrar patrones en un conjuntos de canciones de los cuales pasarían por un proceso de Extracción de la Melodía Principal[5]; luego de este pre-procesamiento, vendría tres análisis para encontrar patrones : búsqueda de Motifs, análisis de melodías y análisis de la estructura musical. A partir del aprendizaje de los patrones generar de acordes y melodías.[5]

Otra fue la propuesta hecha por los doctores Ning-Han Liu, Yi-Hung Wu y Arbee L. P. Chen que en su paper “Identifying Prototypical Melodies by Extracting Approximate Repeating Patterns from Music Works”[24] proponen buscar en segmentos musicales patrones que permitan proponer melodías. La idea básica es considerar cada segmento musical como un candidato que tiene similitud con otros candidatos.

## **2.2.2. Aplicaciones de Algoritmos Genéticos**

### **2.2.2.1. GenJam**

GenJam es un trabajo de investigación en computación evolutiva que simula a un improvisador de jazz mediante el uso de un algoritmo genético. GenJam se encarga de producir eventos únicos e irrepetibles, el cual representa la idea del diseñador. En música, estos eventos pueden ser notas individuales, frases melódicas o incluso una pieza musical completa. En GenJam, estos eventos consisten en pequeñas frases llamadas “licks”, los cuales se unen para formar la improvisación completa del software. [25].

GenJam consiste inicialmente en un algoritmo genético, el cual va mejorando poblaciones de frases melódicas a través de unos operadores previamente definidos. El “fitness” de cada frase es determinado por un mentor (usuario), quien evalúa cada frase de generación en generación hasta obtener una población de frases con un sonido aceptable.[3]

Posteriormente, GenJam pasa de ser un algoritmo genético a un Algoritmo Genético Interactivo. Se habilita la posibilidad de contar con múltiples mentores, lo cual aceleraría ligeramente en tedioso proceso de enseñanza al algoritmo.[6]

Finalmente, GenJam abandona totalmente el uso de mentores, y procede a utilizar una base de datos con melodías ya seleccionadas. De esta manera, el algoritmo irá mejorando de generación en generación las frases ya existentes, con lo que se simularía el proceso creativo de improvisación.[26]

#### **2.2.2.2. Dragan Matic's GA for composing music**

Dragan Matic presenta un algoritmo genético para la creación de música. Incluye una representación flexible para poder almacenar la composición realizada por el ordenar, en el cual almacena el ritmo y el timbre (altura o tonalidad) respectivo de cada nota.

Para poder lograr la creación de melodías, se realiza previamente una enseñanza de ritmos predeterminados. A partir de estos ritmos, se comenzarán a asignar timbres respectivos, y de esta manera, se logrará completar la melodía. Por esta razón, las melodías que se creen, podrán repetir los mismo ritmos, sin embargo, la variación y creatividad se dará a nivel de tonalidades.

Por otro lado, la selección de las mejores generaciones se dará a través de una función fitness. Esta función presenta una novedad con respecto a los anteriores intentos de función fitness. En este, se procede a definir ciertos criterios que pueden calificar a una frase musical como buena o mala. Estos criterios están basados en los intervalos que forma una nota con su sucesor. De esta forma, algunos intervalos (espacio entre notas) sumarán mayor puntaje que otro tipos de intervalos. En otras palabras, se define ciertos filtros que son aplicados a los outputs para de esta forma poder evaluarlos y determinar su valor musical.

Finalmente, los resultados obtenidos reflejan cierta armonía agradable al oído humano. Por otro lado, debido a que solo se limita a evitar sonidos desagradables, las melodías obtenidas no logran una creatividad similar al del talento humano. Sin embargo, se

considera que se pueden emplear otros criterios (metaheurística o hibridación) para la mejora de los resultados.[8]

### **2.2.3. Aplicaciones de las Cadenas de Markov**

Las diversas formas de aplicación, llevó al uso de las Cadenas de Markov en la Teoría Musical. La clasificación de “Eventos Musicales” dentro de un finito o contable número de categorías que ocurren en una secuencia temporal. La pregunta que se hizo fue si es que acaso es posible analizar señales musicales para poder predecir “qué es lo siguiente que sucede” en una melodía. Por ejemplo, se tiene una partitura musical “monofónica” y una señal acústica mostrada de la partitura; la intención es la de adecuar el acompañamiento para este “solo”. El matemático y oboísta profesional Christopher Raphael, de la Indiana University, lo hizo posible gracias a las Cadenas de Markov Hidden y Modelos Bayesianos. Implementó su método en un sistema computarizado llamado “Music Plus One”[27].

Otra intención fue la de los profesores de State University of New York at New Paltz en “Extracting Patterns in Music for Composition via Markov Chains” que intentan encontrar patrones en un conjunto de melodías musicales existentes en una secuencia musical vía Minería de Datos y Aprendizaje Máquina, y según eso, utilizar estos patrones ya identificados para componer música a través de la Cadenas de Markov.

Sin embargo, las intenciones por componer música artificial con cadenas de Markov, presentan dificultades debido a la naturaleza probabilística de los procesos de composición. Si bien son un buen método a corto plazo, a largo plazo “la novedad de aleatoriedad se acumula’ y lleva a que las composición vague sin rumbo. [28]

#### **2.2.3.1. Redes Neuronales**

Las redes neuronales son un método general y práctico para el aprendizaje de funciones reales, discretas y vectoriales a partir de ejemplos. Este método es también

bio-inspirado en las redes interconectadas de neuronas que se encuentran en el cerebro. [29]

- Aprendizaje de reconocimiento de caracteres escritos manualmente
- Aprendizaje de reconocimiento de palabras habladas
- Aprendizaje de Reconocimiento de rostros

#### 2.2.4. Conclusiones sobre el estado del arte

Luego de haber mencionado las soluciones más usadas y con mejores resultados, así como la técnica que utilizan, se puede concluir los siguientes puntos relevantes para el proyecto:

- Las mejores soluciones dejan totalmente de lado los criterios teóricos de música moderna, los cuales son muy importantes en el momento de creación musical. Por esta razón, le daremos un énfasis importante a este punto.
- La mayoría de trabajos parten de datos aleatorios para poder generar la data inicial, la cual mejora continuamente hasta encontrar una solución óptima. Esto es una deficiencia en la performance de las soluciones.
- Los trabajos realizados están orientados a generar música, sin embargo, no se busca un enfoque educativo a las soluciones. Es por eso que la solución tendrá como finalidad dar a conocer a los alumnos la relación entre la armonía dada y la melodía generada.
- Finalmente, no existe implementación alguna de libre uso que permita a los usuarios poder utilizar y aprender con esta herramienta, lo cual podrá ser aprovechado en el proyecto de fin de carrera.

Estas soluciones fueron vitales para el diseño e implementación de la solución planteada.

## Capítulo 3

# Generación de población inicial

### 3.1. Introducción

En este capítulo se explicará la forma de obtención de los dos primeros resultados: Almacenamiento de acordes musicales mapeados con su escala respectiva y el repositorio que almacene las frases que sirva como población inicial. Por esta razón, se explicará el método y el estándar definido para almacenar las escalas musicales con sus respectivos acordes y las frases musicales.

### 3.2. Método de almacenamiento de Acordes Musicales con su escala respectiva

Para poder realizar la generación de una composición musical en cualquier tonalidad, el algoritmo debe ser capaz de adaptar cada uno de los licks a la escala correspondiente para los acordes que ingresó el usuario. Por esta razón, se procederá a almacenar las escalas correspondiente para todos los acordes existentes. Este procedimiento ha sido realizado basándose en la teoría musical de música moderna.

### 3.2.1. Proceso de almacenamiento

Para que el algoritmo pueda generar una melodía sobre cualquier tonalidad que en usuario desee, debemos saber las notas que corresponden a cada uno de los posibles acordes y sus posibles alteraciones. Como se mencionó en el marco teórico, el acorde se encuentra en la parte superior de cada compás en los pentagramas, el cual es un dato de entrada para nuestro algoritmo. Sin embargo, una gran cantidad de acordes. En el siguiente cuadro, se puede apreciar todas las posibles alteraciones solamente para el acorde de Do (C).

Escala musical para cada acorde		
Acorde	Escala	Notas a usar según la teoría musical
Cmaj7/C6	Major (avoid 4th)	C D E G A B
C7	Myxolydian (avoid 4th)	C D E G A Bb
Cm7	Minor (avoid 6th)	C D Eb F G Bb
Cm7b5	Locrian (avoid 2nd)	C Eb F Gb Ab Bb
Cdim	W/H	C D Eb F Gb G# A B
C+	Lydian Augmented	C D E F# G# A B
C7+	Whole Tone	C D E F# G# Bb
C7#11	Lydian Dominant	C D E F# G A Bb
C7alt	Altered Scale	C Db D# E Gb G# Bb
C7#9	Mixolydian #2 (avoid 4th)	C Eb E G A Bb
C7b9	Harm Minor V (avoid 6th)	C Db E F G Bb
CmMaj7	Melodic Minor	C D Eb F G A B
Cm6	Dorian (avoid 7th)	C D Eb F G A
Cm7b9	Melodic Minor II mode	C Db Eb F G A Bb
Cmaj7#11	Lydian	C D E F# G A B
C7sus	Mixolydian	C D E F G A Bb
Cmaj7sus	Major	C D E F G A B
C7Bl	Blues	C Eb F Gb G Bb

FIGURA 3.1: Acordes musicales con su escala respectiva

Este cuadro está basado en teoría musical y será lo que se procederá a almacenar en base de datos para que nuestro algoritmo sea capaz de generar melodías en cualquier tonalidad. A la estructura de datos utilizada para manejar dicha información la conoceremos de ahora en adelante como "ChordvsScale". La descripción de cada uno de los campos en la base de datos y la estructura de la clase creada se encuentran explicados en el siguiente gráfico.

ChordvsScale		
Campo	Descripción	Estructura de datos
id	Identificador para cada acorde generado automáticamente por el motor de base de datos	No es utilizada en el proceso
chord	Contiene el acorde musical	String
scale	Contiene el nombre de la escala correspondiente al acorde. Dicha escala ha sido determinada de acuerdo a la teoría musical moderna.	String
notasTeoricas	Contiene las notas teóricas de la escala.	Array de Strings
notes	Contiene las secuencia de notas (14 notas en total) para dicha escala, y se indica su octava correspondiente.	Array de Strings

FIGURA 3.2: Manejo de la teoría musical durante el proceso de conversión a números

Hasta este momento, se encuentran almacenados todos los posibles casos de acordes para Do (C) con su respectiva escala; sin embargo, existen muchas otras posibles tonalidades. De la misma manera en que se almacenó todos los posibles acordes de Do (C), se debe realizar para Re (D), para Mi bemol (Eb), para Sol sostenido (G#) y para todas las otras tonalidades existentes. Dicho procedimiento resultaría demasiado laborioso de ser realizado manualmente; por lo tanto, se procedió a automatizar el proceso de generación de la escala musical correspondiente a cada acorde.

Para poder deducir las notas de la escala que le corresponden a cada acorde se realizó el siguiente procedimiento:

En primer lugar, se decidió asignarle un número a cada nota existente comenzando desde un Do (C), al cual le corresponde el número cero (0), hasta un Si (B). Después de la nota Si (B), le prosigue nuevamente la nota Do (C) y así infinitamente; por lo tanto, decidimos utilizar solamente 2 repeticiones de las notas, lo cual musicalmente se le conoce como utilizar un máximo de dos octavas musicales. Estos datos fueron almacenados en la estructura "Teclado" la cual se muestra a continuación:

Teclado			
Índice	Nota	Índice	Nota
0	C, B#	12	C, B#
1	C#, Db	13	C#, Db
2	D	14	D
3	D#, Eb	15	D#, Eb
4	E, Fb	16	E, Fb
5	F, E#	17	F, E#
6	F#, Gb	18	F#, Gb
7	G	19	G
8	G#, Ab	20	G#, Ab
9	A	21	A
10	A#, Bb	22	A#, Bb
11	B	23	B

FIGURA 3.3: Teclado musical en base a números

En segundo lugar, utilizamos nuestro caso ya conocido de acordes con sus respectiva escala para el caso de Do (C) y realizamos la conversión a número de acuerdo a lo definido en la estructura "Teclado". Por ejemplo, para el caso del primer acorde (Cmaj7/C6) se obtendrá el siguiente resultado:

Acorde	Escala	Notas musicales	Notas musicales convertidas a números
Cmaj7/C6	Major(avoid 4th)	C D E G A B	0 2 4 7 9 11

FIGURA 3.4: Representación de Cmaj7 en números

Dicho procedimiento se realizó para todos los acordes de la tonalidad de Do (C)

En tercer lugar, para poder obtener todos los acordes otra tonalidad expresado en

números, se procede a el valor que se le asignó a dicha nota en nuestra estructura "Teclado". Continuando con el ejemplo de Cmaj7, si deseamos obtener el acorde Dmaj7, se procede a sumar su valor en la estructura "Teclado", al cual le corresponde el valor 2, a cada una de las notas convertidas en número. El resultado para Dmaj7 se muestra en el siguiente cuadro:

Acorde	Escala	Notas musicales	Notas musicales convertidas a números
Dmaj7/D6	Major(avoid 4th)	? ? ? ? ? ?	2 4 6 9 11 13

FIGURA 3.5: Representación de Dmaj7 en números

Finalmente, ya que conocemos el valor numérico de las notas de la escala de Dmaj7, es sencillo obtener las notas musicales, ya que simplemente se utiliza la estructura "Teclado" para obtener la nota correspondiente a cada uno de los números. El resultado final será el siguiente:

Acorde	Escala	Notas musicales	Notas musicales convertidas a números
Dmaj7/D6	Major(avoid 4th)	D E F A B C	2 4 6 9 11 13

FIGURA 3.6: Notas obtenidas para Dmaj7 de forma automatizada

Dicho procedimiento fue repetido para todas las tonalidades existentes. Esta numeración no está relacionada con la representación del gen de nuestro algoritmo genético.

### 3.2.2. Ventajas del almacenamiento de la teoría musical

El propósito del almacenamiento de la escala correspondiente a cada acorde, brinda las siguientes ventajas:

- Es utilizada para determinar las notas correspondientes a un acorde respectivo al momento de realizar el almacenamiento de frases musicales. De esta forma, facilita la asignación de números a cada una de las notas.

- Limita al algoritmo al momento de realizar la composición, de esta forma, no podrá tocar alguna nota que teóricamente no corresponda a dicho acorde. De esta manera, intentamos evitar las disonancias, lo cual no es agradable al usuario.

Por esta razón, se procede a almacenar las escalas en la base de datos para poder ser utilizadas posteriormente y lograr generar una composición que respete las normas musicales establecidas en la música moderna.

### **3.3. Método de almacenamiento de Frases Musicales**

Durante este proceso, se procede a realizar la carga de las frases musicales validadas por usuarios expertos (licks) a la base de datos. La fuente de licks utilizada han sido realizadas por el músico Toby Wine en su libro "1001 jazz licks" [15]. Por esta razón, las músicas generadas están limitadas a este género musical.

#### **3.3.1. Representación del cromosoma**

Para explicar la estructura que almacenará las frases musicales en la base de datos (el cromosoma), primero se debe entender cómo es que se representan las notas musicales para el algoritmo. Para ello, a cada nota musical de acuerdo a su escala se le asigna un número. En la siguiente figura, se muestra dicha asignación para la escala de Cmaj7 utilizando sólo las notas mostradas en el punto anterior (punto 3.2); es por ello, que en esta representación no está incluida la nota musical FA.

Nota Musical	Tipo de Notación	
	ABC	Numérico
Vacío	V	0
Do	C4	1
Re	D4	2
Mi	E4	3
Sol	G4	4
La	A4	5
Sol	B4	6
Do	C5	7
Re	D5	8
Mi	E5	9
Sol	G5	10
La	A5	11
Si	B5	12
Do	C6	13
Re	D6	14

FIGURA 3.7: Tabla explicativa de las Notas Musicales con su respectiva representación

En la siguiente figura, se observa como dicha representación numérica de las notas musicales es vista en un pentagrama.

Vacío	DO	RE	MI	SOL	LA	SI	DO	RE	MI	SOL	LA	SI	DO	RE
V	C4	D4	E4	G4	A4	B4	C5	D5	E5	G5	A5	B5	C6	D6
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

FIGURA 3.8: Pentagrama musical con su representación numérica

Es así que la siguiente frase musical, con 4 compases cada uno con 8 medias de tiempo (8 corcheas), se representa numéricamente de la siguiente forma con 32 enteros:



FIGURA 3.9: Una frase musical del libro de "1001 jazz lick"[15]

Por otro lado, esta representación no permite hacer uso de estructuras musicales como los “tresillos” o “semicorcheas”, ya que para el caso del tresillo se tienen que tocar tres notas en un tiempo y la representación hasta ahora propuesta solo permite obligar a tocar dos notas en un tiempo [7]. En el caso de la semicorchea se tienen que tocar 4 notas en un tiempo. Para solucionar esta limitación se propone insertar las representaciones especiales “hold” y “tresillo”.

Nota Musical	Tipo de Notación	
	ABC	Numérico
Hold	H	15
Tresillo	T	16

FIGURA 3.10: Tabla de representaciones especiales

Entonces, para poder agregar las estructuras “tresillo” y “semicorchea”, se cambia la representación de la frase de 32 enteros a 64, de forma que en un tiempo se puedan tocar hasta cuatro notas, suficientes para representar las estructuras musicales planteadas. Para no alterar la estructura mínima de cada compás, el cual es igual a 1 negra, a cada nota tocada se le aumentara un “hold”. En la siguiente figura se muestra como varía la representación; pero debe quedar claro que al momento de reproducir ambas vienen a ser la misma figura musical:

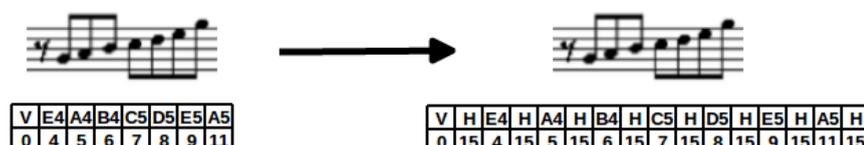


FIGURA 3.11: Variación de la representación

La representación de un “tresillo” vendría a ser como se muestra en la siguiente figura:

Tresillo	RE	RE	RE
T	D4	D4	D4
16	2	2	2

FIGURA 3.12: Representación numérica del tresillo

La representación de una “semicorchea” vendría a ser como se muestra en la siguiente figura:

RE	MI	SOL	LA
D4	E4	G4	A4
4	5	6	7

FIGURA 3.13: Representación numérica del tresillo

Un compás con la nueva representación definida sería la siguiente:

Tr	SI	SI	RE	Tr	LA	LA	SI	SOL	Ho	LA	Ho	SI	Ho	SOL	Ho
T	B4	B4	D5	Tr	A4	A4	B4	G4	H	A4	H	B4	H	G4	H
16	6	6	8	16	5	5	6	4	15	5	15	6	15	4	15

FIGURA 3.14: Representación numérica del tresillo

Para el algoritmo genético en construcción, la representación de una frase musical con 64 enteros sería la siguiente:

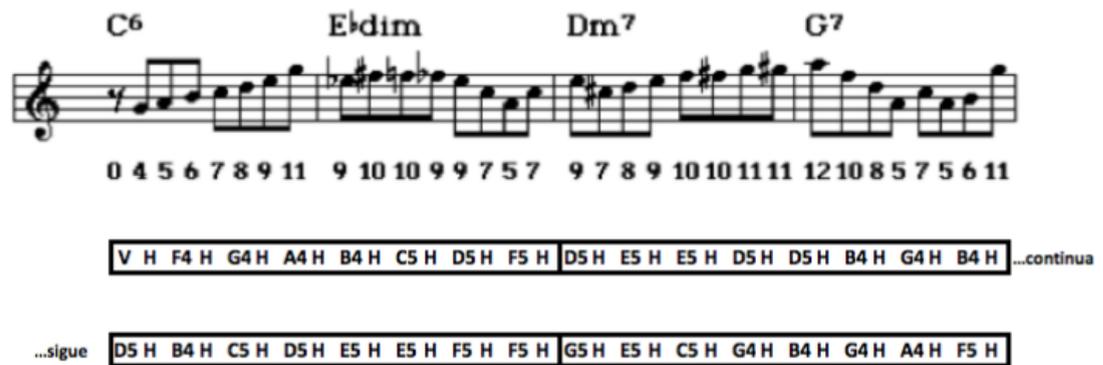


FIGURA 3.15: Representación numérica con 64 enteros de una frase musical

### 3.3.2. Proceso de almacenamiento

El propósito del almacenamiento de las frases musicales es permitir al algoritmo contar con información previa que le permita realizar una creación musical adecuada. A diferencia de la creación de música a partir de una población creada aleatoriamente (random), es que, a partir de una población de frases musicales ya validadas por un experto, nos garantiza la generación de frases con cierto criterio musical.

Para realizar el almacenamiento de frases, procedemos a realizar una conversión de las notas a números. Esto permite poder adaptar cada frase a cualquier armonía que solicite el usuario.

A continuación presentamos la estructura normal de una frase musical validada por un experto:



FIGURA 3.16: Frase extraída del libro 1001 licks [15]

La frase (Phrase) está compuesta por cuatro compases. Cada compás (Measure) cuenta con 4 tiempos. Para poder realizar la asignación de números, debemos considerar el acorde correspondiente a cada Measure. Por ejemplo, en el primer Measure, se tiene asignado un acorde C6, al cual, de acuerdo a la teoría musical almacenada y mostrada en la figura 3.2, le correspondería una escala mayor (Major avoid 4th). Por lo tanto, las notas disponibles serían las siguientes:

Nota Musical	Tipo de Notación	
	Numérico	ABC
Do	1	C4
Re	2	D4
Mi	3	E4
Fa	4	F4
Sol	5	G4
La	6	A4
Si	7	B4
Do	8	C5
Re	9	D5
Mi	10	E5
Fa	11	F5
Sol	12	G5
La	13	A5
Si	14	B5

FIGURA 3.17: Tabla explicativa de las Notas Musicales con su respectiva representación

A cada nota procedemos a agregarle un número, comenzando por el Do 4 (C4 en notación musical). Tomamos como unidad a la semicorchea. Esto significa que, en total, tendremos 16 números por compás. Para representa el silencio, utilizamos el 0. Para un "hold" usamos el 15. Finalmente para el inicio de un treillo, utilizamos el 16.

Una vez realizada la asignación de números para cada measure, se procede a almacenar individualmente cada uno, asignándole un "id" respectivo, para, posteriormente, almacenar el Phrase con los 4 ids respectivos de los measures por los que está compuesto.

Finalmente, se ha procedido a realizar una clasificación de las frases que están siendo ingresadas por el género musical al que corresponden.

### 3.4. Modelo de Base de Datos

Para el correcto funcionamiento del algoritmo, es necesario el almacenamiento de los siguientes elementos: En primer lugar, las frases musicales validadas por expertos, las cuales servirán como población inicial del algoritmo. En segundo lugar, la teoría musical necesaria para poder generar nuevas frases musicales correctas. Para ello, se realiza el almacenamiento de las escalas musicales correspondientes para cada acorde musical. Para poder cumplir dichos requerimientos, se ha utilizado una base de datos con tablas no relacionadas, las cuales almacenarán las frases musicales y la teoría musical. A partir de dichos elementos, se procederá a generar la población inicial para el algoritmo. A continuación, se muestra el modelo de la base de datos con sus respectivas tablas. Este modelo de base de datos y las estructuras que han sido utilizadas son explicadas durante este y los próximos capítulos.

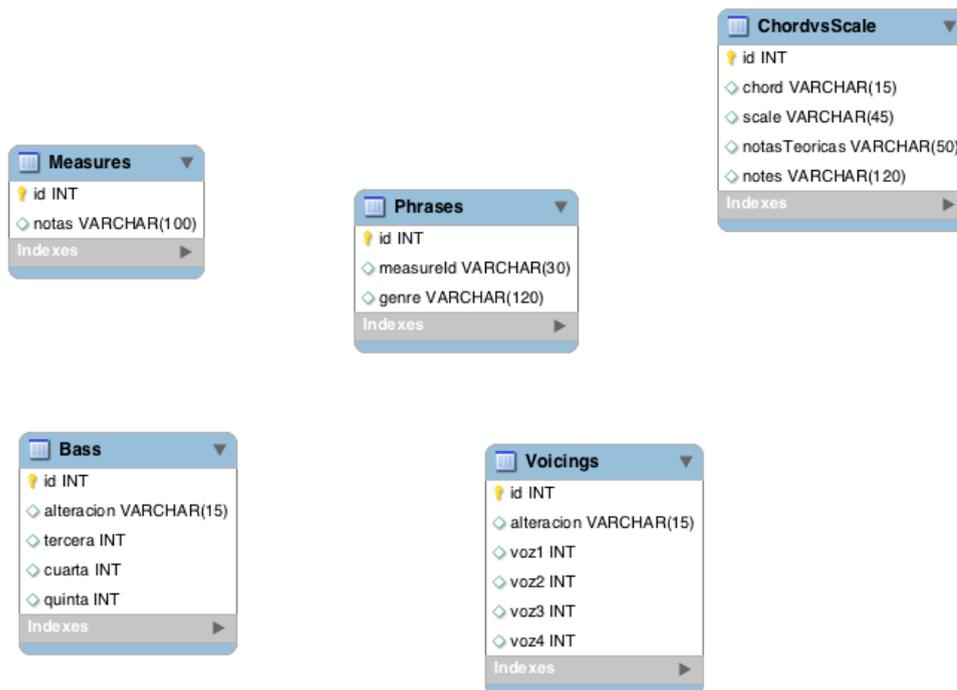


FIGURA 3.18: Modelo de la base de datos

### 3.5. Estructuras de datos utilizadas

Para poder cumplir este proceso, se utilizan 3 tablas de nuestra base de datos.

En primer lugar, se utiliza la tabla "ChordvsScale". En ella se encuentra almacenada la teoría musical necesaria para poder realizar la conversión de las frases musicales a sus respectivos números.

Para poder utilizar adecuadamente dichos valores durante el proceso de conversión, se utilizó la clase creada "ChordvsScale". La descripción de cada uno de los campos en la base de datos fueron explicados en el punto anterior. Está clase nos permite poder ubicar la escala correspondiente para cada lick que será convertido a números.

En segundo lugar, se utiliza la tabla "Measures". En ella se puede almacenar cada uno de los measures anteriormente descritos después de realizar el proceso de conversión a números. De esta forma, se contará con 4 measures almacenados por cada frase.

Para poder utilizar adecuadamente esos valores, se realizó la creación de la clase "Measure". La descripción de cada uno de los campos se explica en la siguiente tabla:

Measures		
Campo	Descripción	Estructura de datos
id	Identificador generado consecutivamente para poder ser refenciado a la frase a la que pertenece	Entero
notas	Contiene las notas (números) de un respectivo compás	Lista de enteros

FIGURA 3.19: Manejo de la los measures durante el proceso de conversión a números

## 3.6. Discusión

### 3.6.1. Almacenamiento de teoría musical

Para el logro de este resultado, se tuvo que realizar un estudio amplio sobre teoría musical y el uso de diferentes fuentes, entre ellas los libros de armonía musical de "Berklee College of Music" [17] y el libro de "Armonía musical moderna" de Enric Herrera [1]. De esta manera, se logró determinar las escalas más adecuadas y realizar la adaptación para cada uno de los acordes existentes.

### 3.6.2. Almacenamiento de frases musicales

Para la realización de esta tarea, se tuvo que recurrir al uso de un libro de frases musicales validadas por un experto. Dicho ingreso de las notas correspondiente a la frase se tuvo que realizar de manera manual. Se diseñó un programa el cual traduzca las notas ingresadas a su número correspondiente y las almacene en base de datos. La cantidad de frases va a influir en la capacidad del algoritmo para generar una mayor diversidad de composiciones.

## Capítulo 4

# Modelo Computacional que permita generar la composición musical

### 4.1. Introducción

En este capítulo se explicará el proceso de construcción del modelo computacional de algoritmos genéticos que utilizó para obtener la composición musical. Así mismo, se mencionará su interacción con cada uno de los objetivos anteriores y las diferentes estructuras de datos que utilizan para el proceso generación de la melodía.

### 4.2. Modelo Computacional

#### 4.2.1. Función de fitness

Para entender la función de fitness, primero se debe entender el concepto de distancia horizontal entre dos notas musicales[7]. La distancia horizontal se entiende como el

valor absoluto de la diferencia numérica que tienen dos notas musicales. La fórmula sería de la siguiente forma:

$$\text{distancia\_horizontal} = | \text{Nota}_1 - \text{Nota}_2 |$$

FIGURA 4.1: Distancia Horizontal entre notas

Es así, que el valor de la distancia horizontal de las notas DO4 y LA4 será.

$$\begin{aligned} \text{distancia\_horizontal} &= | \text{Do}_4 - \text{La}_4 | \\ \text{distancia\_horizontal} &= | 1 - 5 | \\ \text{distancia\_horizontal} &= 4 \end{aligned}$$

FIGURA 4.2: Ejemplo de distancia horizontal

Este valor de la distancia horizontal es representativo para el algoritmo, ya que mientras menor sea el valor de la distancia horizontal entre dos notas, mejor será la melodía generada[1]. Es por ello, que en el siguiente ejemplo, se muestra cómo una distancia de 1 es preferible sobre una de 7.

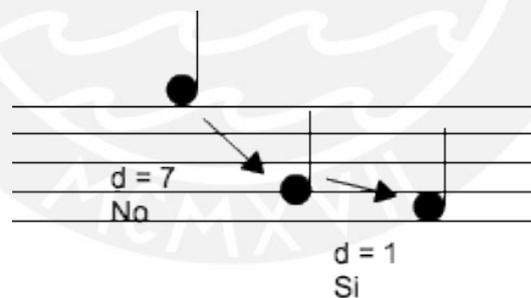


FIGURA 4.3: Evaluación de distancias agradables durante la improvisación

Tal como se explicó en el capítulo 3, un individuo viene a ser una “Frase” musical, la cual está compuesta por 4 “compases”. El paso de notas que hay entre compás y compás es considerado como un punto crítico, y son puntos dónde la operación de casamiento hará el cruce entre los padres.

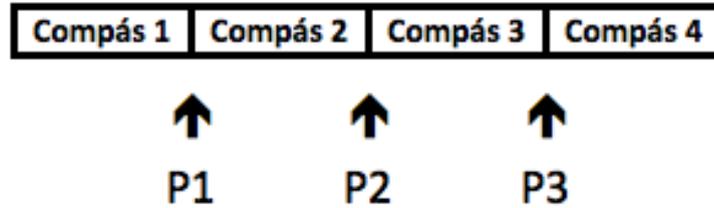


FIGURA 4.4: Los puntos críticos en el individuo

EL siguiente gráfico muestra como es hallada la distancia horizontal en los puntos críticos.

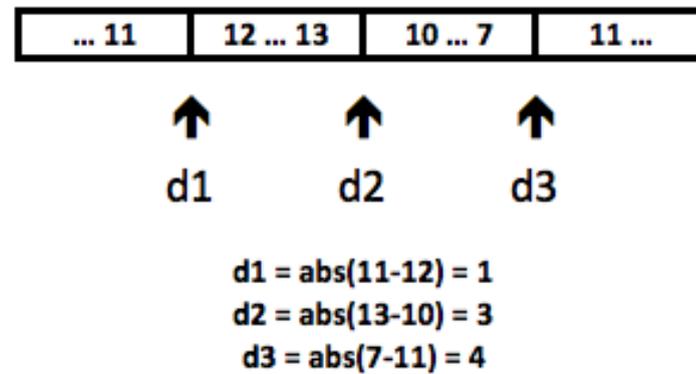


FIGURA 4.5: Cálculo de la distancia horizontal

Dichos puntos críticos son importantes, ya que serán en estas dónde el operador de casamiento hará el cruce para generar nuevos hijos. Para dos padres que serán casados, 4 compases cada uno, podrán generar 6 hijos, ya que se generan 2 hijos por cada punto crítico. En el siguiente gráfico se muestra 2 ejemplos que se generan con el operador de casamiento.

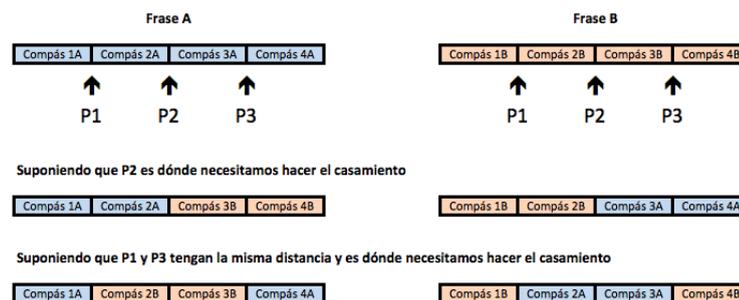


FIGURA 4.6: Los puntos críticos en el individuo

La función objetivo busca minimizar la distancia horizontal en cada punto crítico; es por ello que será el par de hijos que tenga menor distancia horizontal el que pase a la siguiente generación.

$$\text{Min} \left( \sum_{i=1}^3 \text{abs}(P_i) \right)$$

FIGURA 4.7: Función Objetivo

La razón por la cual se busca minimizar la distancia es la de lograr que al juntar dos partes de cromosomas, la transición de las notas musicales no sea notada por lo “fuerte” del paso de una a otra, sino de una manera suave, para que la melodía sea bien percibida por el usuario. En el caso de los operadores de mutación, se intenta minimizar dicha distancia horizontal, sin embargo, en el operador de casamiento, se intenta conservar la distancia horizontal de las frase musicales que están siendo casados, de tal forma que se conserve la idea musical original que intentaban transmitir.[30]

Es así que una vez especificado lo anterior, es posible entender el pseudocódigo del algoritmo genético.

#### 4.2.2. Pseudocódigo

A continuación se muestra el pseudocódigo del algoritmo genético, para después ser explicado individualmente:

```

1: procedure ALGORITMO_GENÉTICO
2:   poblacion_inicial ← Inicializacion_Poblacion_Inicial()
3:   i ← 0
4:   while i ≤ NUMERO_GENERACIONES do
5:     poblacion ← Torneo_Seleccion(poblacion_inicial)
6:     k ← 0
7:     for k ≤ tamano(poblacion) do
8:       padreX, padreY ← selecciona_padres(poblacion)
9:       if PROBABILIDAD_CASAMIENTO ≤ random() then

```

```

10:         hijoX, hijoY ← casamiento(padreX, padreY)
11:         if PROBABILIDAD_MUTACION ≤ random() then
12:             hijoX ← mutacion(hijoX)
13:         end if
14:         if PROBABILIDAD_MUTACION ≤ random() then
15:             hijoY ← mutacion(hijoY)
16:         end if
17:     end if
18:     k ← k + 1
19: end for
20:     i ← i + 1
21: end while
22: end procedure

```

#### 4.2.3. Inicialización de la población

Como se puede observar en el algoritmo, el primer paso es el de la selección. Para poder ser realizado se procede a sacar aleatoriamente 16 frases musicales de la Base de Datos. Después de ello, se crean 32 frases adicionales a partir de las 16 inicialmente seleccionadas. Esto es realizado debido a que toda la composición debe poseer la misma idea musical durante todo su desarrollo. Entonces, la idea detrás de la creación de estos 32 será de la siguiente manera: 16 nacerán del cruzamiento de la población obtenida de la Base de Datos y otros 16 nacerán a partir de obtener el resultado de los operadores de mutación a los 16 iniciales. Finalmente, éstas conformarán la población final que pasará por el algoritmo genético.

El pseudocódigo del proceso es el siguiente:

```

1: procedure INICIALIZACIÓN_POBLACIÓN_INICIAL
2:     poblacion_inicial ← sacar_frases_aleatoriamente_de_BD()
3:     poblacion_final ← vacio
4:     for par ∈ poblacion_inicial do
5:         hijoX, hijoY ← casamiento(par)

```

```
6:     insertar(poblacion_final, hijoX, hijoY)
7:   end for
8:   for individuo ∈ poblacion_inicial do
9:     hijoZ ← mutacion(individuo)
10:    insertar(poblacion_final, hijoZ)
11:  end for
12:  devolver(poblacion_final)
13: end procedure
```

#### 4.2.4. Torneo de selección

Una vez inicializada la población para el algoritmo genético, se procede a realizar un torneo entre pares y seleccionar 48 frases, que vendría a ser el número máximo de frases utilizadas en una composición[17]. Las frases a competir serán seleccionadas mediante un “random probabilístico”, entre las cuales se admitirán repeticiones. Este grupo seleccionado pasará a la fase de casamiento.

#### 4.2.5. Operadores genéticos

Una frase musical cuenta con una secuencia de notas, las cuales son las encargadas de expresar la idea melódica. Estas notas deben mantener una distancia mínima entre ellas, de tal forma que se eviten intervalos amplios e inesperados entre ellas. Las notas que presentan una distancia muy amplia entre ellas, lo cual es conocido como intervalo horizontal, no presentan generalmente un sonido muy agradable para los usuarios, por lo tanto, las composiciones que genere el algoritmo, debe de tratar de lograr el menor intervalo horizontal posible entre nota y nota, y además, minimizar el intervalo horizontal entre el fin de una frase y el comienzo de una nueva. Sin embargo, las frases actualmente validadas por expertos, presentan en varios casos intervalos horizontales de alta magnitud. Por lo tanto, en el caso de cruces de dichas frases, se deberá intentar conservar dicha distancia para evitar perder la idea que inicialmente se deseaba transmitir.

A continuación, se hablará sobre los operadores genéticos utilizados en el algoritmo.

#### 4.2.5.1. Casamiento inteligente

Como se comenta en el anterior punto 4.2.2, el casamiento se realiza en puntos claves y no aleatoriamente como se suele practicar en los Algoritmos Genéticos; puesto que el uso del crossover aleatorio no garantiza soluciones válidas para la composición, en las cuales la distancia horizontal no necesariamente será el mínimo. Por lo tanto, se ha procedido a utilizar un casamiento inteligente, analizando los puntos claves, el cual genere siempre soluciones válidas. Por lo tanto, este “crossover” especializado permitirá realizar mejoras a la población inicial. En cada cruce entre dos padres, los hijos tendrán compases de los padres de forma que siempre cada hijo tendrá al menos un compás de uno de los padres, esto se realiza con la intención de lograr que todas las combinaciones posibles permitan hallar el par de hijos con mejor función de fitness.

La función matemática que se regirá para la función de fitness será la de minimizar la distancia horizontal entre los compases. A continuación, se muestra una ilustración de dichos puntos:

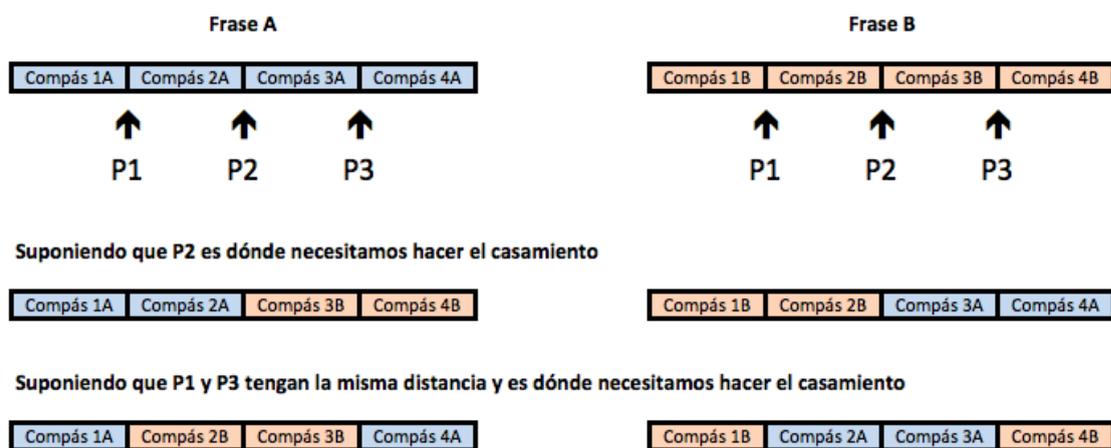


FIGURA 4.8: Operación de Casamiento

Es así que la función de minimización se basará bajo el siguiente pseudocódigo:

```

1: function CASAMIENTO_INTELIGENTE(padreX, padreY)
2:   min_dist, punto  $\leftarrow$  MAX_VALOR, VACIO
3:   for punto_clave  $\in$  frase_musical do
4:     distancia_padres  $\leftarrow$  distancia_horizontal(padreX, padreY)
5:     hijoX, hijoY  $\leftarrow$  posibles_hijos(padreX, padreY, punto_clave)
6:     distancia_hijos  $\leftarrow$  distancia_horizontal(hijoX, hijoY)
7:     diferencia  $\leftarrow$  valor_absoluto(distancia_padres - distancia_hijos)
8:     if diferencia < min_dist then
9:       min_dist  $\leftarrow$  diferencia
10:      punto  $\leftarrow$  punto_clave
11:    end if
12:  end for
13:  devolver(min_dist, punto_clave)
14: end function

```

#### 4.2.5.2. Mutación

La población inicial está conformada por las frases iniciales validadas por expertos. Para que el algoritmo logre realizar composiciones diferentes cada vez que se ejecute, es necesario dar diversidad a la población, de lo contrario, las composiciones podrán sonar repetidas o siempre basadas en las mismas frases.

Los operadores genéticos utilizados permiten lograr la diversidad que deseamos en la población. Sin embargo, al igual que con el crossover, el uso de operadores aleatorios podrían dañar la frase, e incluso transformarla en una totalmente desagradable para el usuario. Debido a esta razón, se ha procedido a implementar operadores de mutación inteligentes, los cuales aseguran la consistencia musical de cada frase en relación a la función objetivo. De esta forma, conservamos el sonido agradable y no echa a perder el trabajo del crossover de minimizar el intervalo horizontal. Algunos de los operadores de mutación implementados son los siguientes:

■ Transpose 2 down operator

Realiza una transposición de toda la frase hacia 2 tonos abajo. Este operador es mayormente utilizado para brindar diversidad en las frases sin alterar la idea principal ni arriegarse a obtener posibles disonancias.

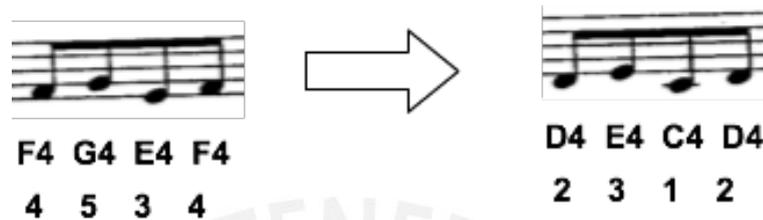


FIGURA 4.9: Operador Transpose 2

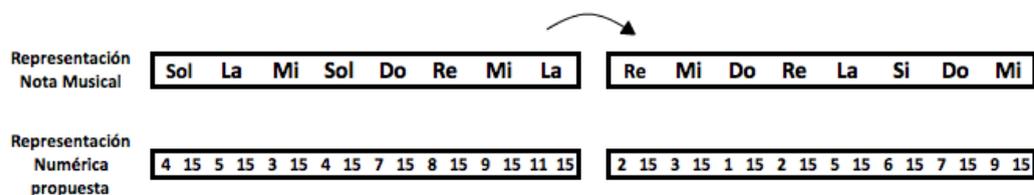


FIGURA 4.10: Ejemplo en cromosoma final

El pseudocódigo de este operador sería el siguiente:

```

1: function TRANSPOSE_TWO_DOWN(frase_musical)
2:   TWO ← 2
3:   nueva_frase ← VACIO
4:   for nota_musical ∈ frase_musical do
5:     if nota_musical > TWO then
6:       nueva_nota ← nota_musical - TWO
7:       frase_musical.append(nueva_nota)
8:     end if
9:   end for
10:  devolver(nueva_frase)
11: end function
    
```

■ Octave operator

Este operador cumple una función de mejorador y diversificador. Mediante él, se proceden a realizar modificaciones a la frase inicial, sin embargo, son realizadas buscando mejorarlas bajo el mismo principio de intentar disminuir el intervalo horizontal. Para ello, el operador procede a ubicar notas que presente un intervalo horizontal alto, lo cual posiblemente pueda resultar desagradable al usuario. Una vez ubicado, se procede a realizar una transposición de una octava, de tal forma que reduzca el intervalo horizontal anterior y logre eliminar dicho intervalo que podía resultar poco ventajoso para el algoritmo.

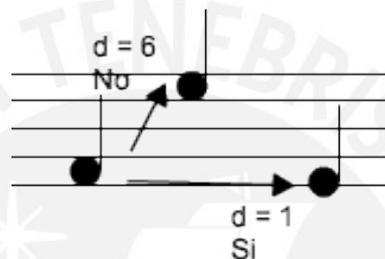


FIGURA 4.11: Ejemplo del operador "Octave"

En el ejemplo mostrado, se realizó una transposición del MI5 (2da nota) por un MI4 (3era nota), de esta forma, se conservó el mismo sonido, pero se redujo la distancia horizontal.

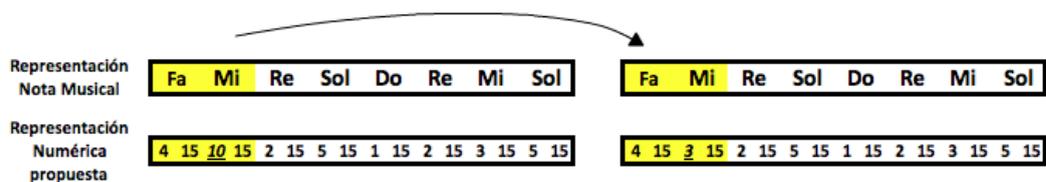


FIGURA 4.12: Ejemplo en cromosoma final

El pseudocódigo de este operador sería:

- 1: **function** OCTAVE(*frase\_musical*)
- 2:     *OCTAVE* ← 7
- 3:     *nueva\_frase* ← VACIO
- 4:     *nota\_inicial* ← *nota\_inicial\_frase*(*frase\_musical*)
- 5:     *frase\_musical.append*(*nota\_inicial*)
- 6:     **for** *nota\_act*, *nota\_sig* ∈ *frase\_musical* **do**

```

7:      nota_nueva ← VACIO
8:      if  $\text{abs}(\text{nota\_act} - \text{nota\_sig}) \geq \text{OCTAVE}$  then
9:          if  $(\text{nota\_act} - \text{nota\_sig}) \geq 0$  then
10:             nota_nueva ← nota_sig - OCTAVE
11:          else
12:             nota_nueva ← nota_sig + OCTAVE
13:          end if
14:      end if
15:      frase_musical.append(nota_nueva)
16:  end for
17:  devolver(nueva_frase)
18: end function

```

- Hemiola operator

Este operador es el que brinda una mayor diversidad a las frases. Con el hemiola, se busca un grupo determinado de notas. Una vez ubicado dicho grupo, se procede a repetirlo durante toda la frase (los 4 compases). De esta forma, se logra repetir dicho grupo de notas con un sonido agradable durante todo el compás, lo cual brinda una sensación de alteración de ritmo a la composición. Es una técnica muy utilizada por los músicos durante la improvisación y que hemos logrado simular para que el algoritmo la aplique en la composición. En la práctica los músicos toman un compás musical y esta es repetida en toda la frase.



FIGURA 4.13: Ejemplo de Hemiola

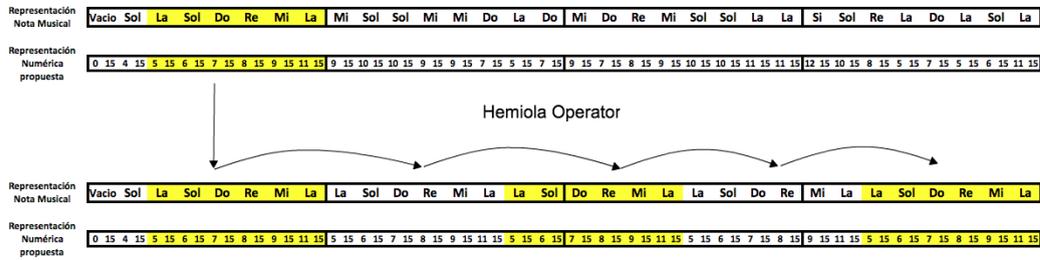


FIGURA 4.14: Ejemplo en cromosoma final

El pseudocódigo de este operador sería:

```

1: function HEMIOLA(compas_musical)
2:   nueva_frase ← VACIO
3:   for compas ∈ frase_musical do
4:     frase_musical.append(compas_musical)
5:   end for
6:   devolver(nueva_frase)
7: end function
    
```

### 4.3. Discusión

El algoritmo genético permitió la creación de nuevas frases musicales a partir de los licks del repositorio. Debido al riguroso uso de la teoría musical, no está permitida la generación de notas que no concuerden con los acordes ingresados por el usuario. De esta manera, se logra evitar todo tipo de disonancias en las frases musicales. Finalmente, el uso de los operadores genéticos inteligentes permitió mejorar las frases musicales iniciales y lograr la creación de nuevas frases a partir de ellas, lo cual impide que el algoritmo realice repetitivamente las mismas creaciones.

## Capítulo 5

# Presentación de resultados a los usuarios finales

### 5.1. Introducción

En este capítulo se explicará el proceso mediante el cual los resultados obtenidos por el algoritmo son presentados al usuario final y la interfaz utilizada para la obtención de la armonía musical por parte del usuario.

### 5.2. Interfaz para la lectura de la armonía para su manipulación y procesamiento

El propósito de la interfaz gráfica para la lectura de la armonía es que los usuarios finales puedan ingresar los acordes con los que se desea que la aplicación improvise. Esto brindará las siguientes ventajas:

- Los usuarios podrán ingresar los acordes sobre los que desean se realice la improvisación. Además, hemos incluido una lista de temas musicales, los cuales

pueden ser seleccionados para que sus acordes sean incluidos automáticamente y realizar una composición sobre ellos.

- Los usuarios podrán seleccionar la velocidad a la cual desean sea realizada la composición (tempo).
- Los usuarios tendrán la posibilidad de reproducir previamente la armonía que han ingresado, para poder darse una idea de como suenan los acordes que han ingresado.

A continuación, se muestra la imagen del panel realizado para poder brindarle dichas funcionalidades a los usuario:

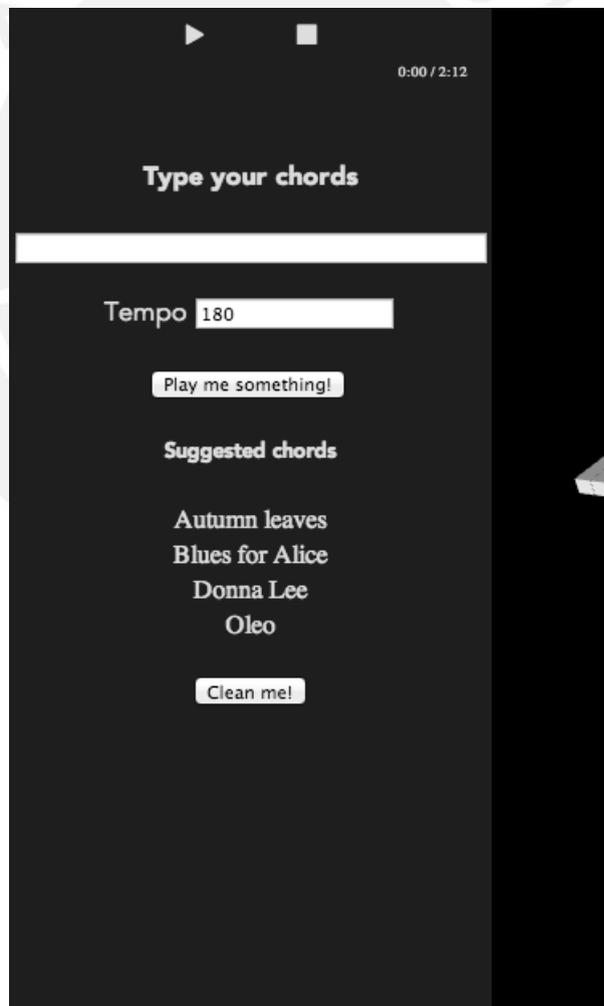


FIGURA 5.1: Panel para el ingreso de la armonía

### 5.3. Generación de archivos MIDI de la creación del algoritmo y del acompañamiento

Para que la herramienta desarrollada sirva como soporte para músicos novatos se realizó la generación de varios archivos MIDIs, de tal forma que el estudiante pueda utilizarlos de la manera más adecuada y que puedan apoyar durante su aprendizaje. Los archivos fueron generados con el apoyo de la librería jMusic, y contiene las partes de batería, bajo y piano.

La estructura del archivo MIDI está determinada por los canales con los que cuenta y los instrumentos que tiene asignados. El siguiente cuadro muestra los canales determinados para la generación de los archivos MIDI:

Estructura archivos MIDI			
Canal	Nombre	Instrumento	Descripción
0	Improvisación - Corchea Straight - Mute	Piano	Improvisación generada por el algoritmo. Es almacenada en un canal del archivo midi con sus tiempo reales de duración (corchea straight). Se utilizada para la generación de la partitura con los tiempos originales. Este canal no es reproducido. Se encuentra en silencio (mute).
1	Walking de bajo	Bajo	Acompañamiento del bajo generado automáticamente de acuerdo a la armonía ingresada por el usuario.
2	Voicings	Piano Rhodes	Acompañamiento de acordes generado automáticamente de acuerdo a la armonía ingresada por el usuario.
3	Improvisación - Corchea Swing	Piano Rhodes	La misma improvisación original generada por el algoritmo es adaptada a la corchea Swing, lo cual le da el estilo Jazz que se quiere lograr.
9	Percusión	Batería (Snare + Ride)	Generación automática de la percusión de acuerdo a la duración de la improvisación.

FIGURA 5.2: Estructura del archivo MIDI generado

Se ha realizado la generación de varios archivos MIDIs, cada uno de ellos puede utilizar, o no, los canales anteriormente indicados. A continuación, se mencionan los archivos MIDIs creados con el uso de los canales.

- Acompañamiento musical

Para que el estudiante pueda comprobar si la armonía que ha ingresado suena agradable, se realiza la generación de un archivo MIDI en el cual se excluye la

improvización (canal 3). De esta manera, puede corregir algún error que identifique escuchando la armonía que ingresó.

- Población Inicial

La generación de este archivo tiene varias intenciones. En primer lugar, sirve como ilustración de la forma en que se alteraron las frases musicales y se obtuvo una melodía totalmente distinta. En segundo lugar, permite al estudiante poder escuchar y analizar algunos licks que probablemente no tenía en su repertorio musical. Para este MIDI, se utilizan los mismos canales que el objetivo anterior y se agrega el canal de improvisación con la diferencia de que, en este momento, la melodía aún no ha pasado por el algoritmo genético.

- Música generada

Este archivo contiene la improvisación realizada por el algoritmo a partir de la población inicial que se pudo mostrar en el anterior MIDI. Para ello, se utilizan todos los canales.

- Play Along

Este archivo es una alteración de la improvisación generada. Se procede incluir espacios en el que sólo se realiza un acompañamiento, y otros en el que se incluye la improvisación, de esta forma, le da un espacio al alumno para que pueda practicar su propia improvisación. Posteriormente, se reducen esos espacios a sólo 4 compases en los que alternan el algoritmo y el estudiante. En términos musicales, a esto se le conoce como "trade fours" y le permite al alumno interactuar con el algoritmo y mantener lo que se conoce como una "conversación musical".

- Play Along versión Piano

El archivo contiene la misma idea que el anterior, con la única diferencia que está diseñado especialmente para improvisación de piano, por lo que se silencia el canal de Voicings.

- Solo Improvisación

En este archivo se escucha el canal de improvisación, el bajo y el ritmo de batería. En esta se busca que el alumno pueda ver y escuchar con mayor facilidad la improvisación en sí y evitar los adornos como los voicings, los cuales pueden dificultar a algunos alumnos seguir con mayor detalle el solo.

#### **5.4. Reproducción musical en la herramienta desarrollada**

La finalidad de este punto es el de mostrar el resultado del algoritmo al usuario para poder identificar los patrones que le permitan resolver las improvisaciones en las escalas correspondientes para los acordes que se han ingresado.

- Para poder mostrar al usuario final los resultados, se realiza la construcción un archivo de extensión .midi mediante la herramienta que es el resultado final, el cual será devuelto al usuario. Para ello, se debe de adaptar cada una de las melodías a su escala correspondiente, la cuales son determinandas de acuerdo al acorde ingresado por el usuario.
- Además de la generación de la melodía creada por el algoritmo genético, se realiza la creación automática de un acompañamiento a la melodía, el cual está conformado por una sección rítmica generada de acuerdo a los acordes ingresados por el usuario. Para ello fue necesario el uso de la librería jMusic.
- El archivo midi es enviado desde el servidor para ser enviado a la web cliente.
- Una vez que el midi ya se encuentre en la web, esta es reproducida y mostrada en la web mediante un piano. Cada nota musical de la melodía final será mostrada mediante un toque en el piano. Esto permitirá que la composición sea fácil de entender, tanto para expertos como para músicos novatos. Para ello, fue necesaria la adaptación de la librería Euphony hacia el objetivo del proyecto.

A continuación, se muestra una imagen de la herramienta reproduciendo una de las melodías generadas.

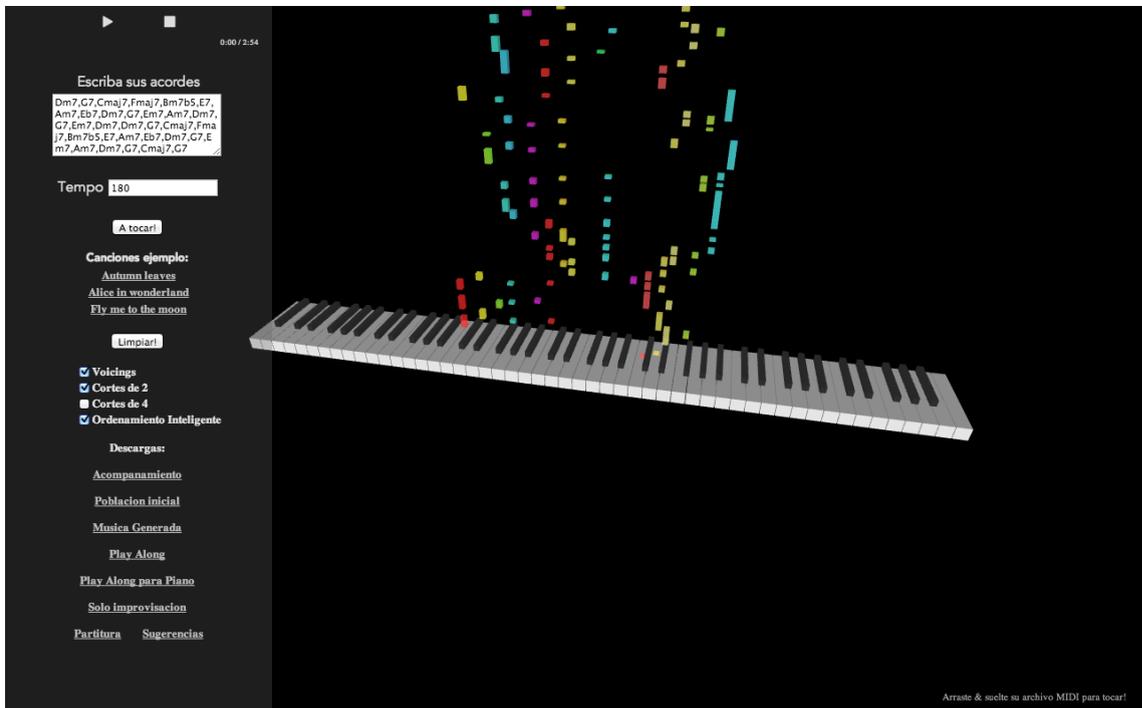


FIGURA 5.3: Herramienta musical reproduciendo una composición generada automáticamente

La herramienta desarrollada está totalmente orientada a apoyar el aprendizaje de los músicos novatos, por lo cual, la improvisación es mostrada de la manera más sencilla de entender para un músico, el cual es a través de un piano, donde se pueden apreciar claramente las notas. Los archivos MIDI's generados son compatibles con esta herramienta, lo cual les permite poder analizar cualquiera de los archivos generados con ella. Para una mayor facilidad de comprensión, la herramienta no reproduce el acompañamiento de la batería (canal 9), sin embargo, puede ser escuchado reproduciendo el archivo como cualquier otro archivo de sonido.

## 5.5. Representación musical de la composición y sugerencias

Para apoyar la improvisación de la música generada por la herramienta, se procede a mostrar las notas musicales y su respectiva duración mediante el sistema de notación

musical mundialmente más utilizado: la partitura. A continuación, se muestra el ejemplo de una partitura generada por nuestra herramienta.



FIGURA 5.4: Improvisación sobre "Alice in wonderland"

Finalmente, para poder apoyar el desarrollo de improvisación musical de los músicos, se brinda un archivo de texto con sugerencias para la improvisación, especialmente para la armonía ingresada por el usuario. A continuación, se muestra una imagen del archivo otorgado al músico a través de la herramienta.

Sugerencias sobre la Improvisación

Para el acorde : Dm7 se le recomienda usar la escala : Minor (avoid 6th) la cual esta conformada por las notas : D E F G A C

Para el acorde : G7 se le recomienda usar la escala : Mixolydian (avoid 4th) la cual esta conformada por las notas : G A B D E F

Para el acorde : Cmaj7 se le recomienda usar la escala : Major (avoid 4th) la cual esta conformada por las notas : C D E G A B

Para el acorde : Fmaj7 se le recomienda usar la escala : Major (avoid 4th) la cual esta conformada por las notas : F G A C D E

Para el acorde : Bm7b5 se le recomienda usar la escala : Locrian (avoid 2nd) la cual esta conformada por las notas : B D E F G A

Para el acorde : E7 se le recomienda usar la escala : Mixolydian (avoid 4th) la cual esta conformada por las notas : E F# G# B C# D

Para el acorde : Am7 se le recomienda usar la escala : Minor (avoid 6th) la cual esta conformada por las notas : A B C D E G

Para el acorde : Eb7 se le recomienda usar la escala : Mixolydian (avoid 4th) la cual esta conformada por las notas : Eb G A Bb C Db

Para el acorde : Dm7 se le recomienda usar la escala : Minor (avoid 6th) la cual esta conformada por las notas : D E F G A C

Para el acorde : G7 se le recomienda usar la escala : Mixolydian (avoid 4th) la cual esta conformada por las notas : G A B D E F

Para el acorde : Em7 se le recomienda usar la escala : Minor (avoid 6th) la cual esta conformada por las notas : E F# G A B D

Para el acorde : Am7 se le recomienda usar la escala : Minor (avoid 6th) la cual esta conformada por las notas : A B C D E G

Para el acorde : Dm7 se le recomienda usar la escala : Minor (avoid 6th) la cual esta conformada por las notas : D E F G A C

Para el acorde : G7 se le recomienda usar la escala : Mixolydian (avoid 4th) la cual esta conformada por las notas : G A B D E F

Para el acorde : Em7 se le recomienda usar la escala : Minor (avoid 6th) la cual esta conformada por las notas : E F# G A B D

Para el acorde : Dm7 se le recomienda usar la escala : Minor (avoid 6th) la cual esta conformada por las notas : D E F G A C

Para el acorde : Dm7 se le recomienda usar la escala : Minor (avoid 6th) la cual esta conformada por las notas : D E F G A C

Para el acorde : G7 se le recomienda usar la escala : Mixolydian (avoid 4th) la cual esta conformada por las notas : G A B D E F

Para el acorde : E7 se le recomienda usar la escala : Mixolydian (avoid 4th) la cual esta conformada por las notas : E F# G# B C# D

Para el acorde : Am7 se le recomienda usar la escala : Minor (avoid 6th) la cual esta conformada por las notas : A B C D E G

Para el acorde : Eb7 se le recomienda usar la escala : Mixolydian (avoid 4th) la cual esta conformada por las notas : Eb G A Bb C Db

Para el acorde : Dm7 se le recomienda usar la escala : Minor (avoid 6th) la cual esta conformada por las notas : D E F G A C

Para el acorde : G7 se le recomienda usar la escala : Mixolydian (avoid 4th) la cual esta conformada por las notas : G A B D E F

Para el acorde : Em7 se le recomienda usar la escala : Minor (avoid 6th) la cual esta conformada por las notas : E F# G A B D

Para el acorde : Am7 se le recomienda usar la escala : Minor (avoid 6th) la cual esta conformada por las notas : A B C D E G

Para el acorde : Dm7 se le recomienda usar la escala : Minor (avoid 6th) la cual esta conformada por las notas : D E F G A C

Para el acorde : G7 se le recomienda usar la escala : Mixolydian (avoid 4th) la cual esta conformada por las notas : G A B D E F

Para el acorde : Cmaj7 se le recomienda usar la escala : Major (avoid 4th) la cual esta conformada por las notas : C D E G A B

Para el acorde : G7 se le recomienda usar la escala : Mixolydian (avoid 4th) la cual esta conformada por las notas : G A B D E F

FIGURA 5.5: Sugerencias de improvisación sobre "Alice in wonderland"

## 5.6. Arquitectura de la herramienta web desarrollada

Para la realización de esta tarea, se dividió en dos áreas Cliente y Servidor.

El lado Cliente es la parte gráfica de la página web, en el cual se utilizó las siguientes herramientas:

- HTML : Para construir la interfaz gráfica
- JavaScript : Para el manejo de eventos y envío de información mediante una llamada XMLHttpRequest al servidor, en esta llamada se envía los datos que serán de entrada para el algoritmo: los acordes, el tiempo, el instrumento, el tipo de algoritmo.

El lado Servidor, esta se encarga del envío de datos que el usuario ingrese, que sea enviada al algoritmo y retornada a la página para su reproducción, se utilizó las siguientes herramientas:

- PHP : El servidor recibe la llamada XMLHttpRequest con los datos de entrada para el algoritmo y esta las envía al programa en JAVA para ejecutarla y devolver el midi, que será finalmente reproducido en la página web.

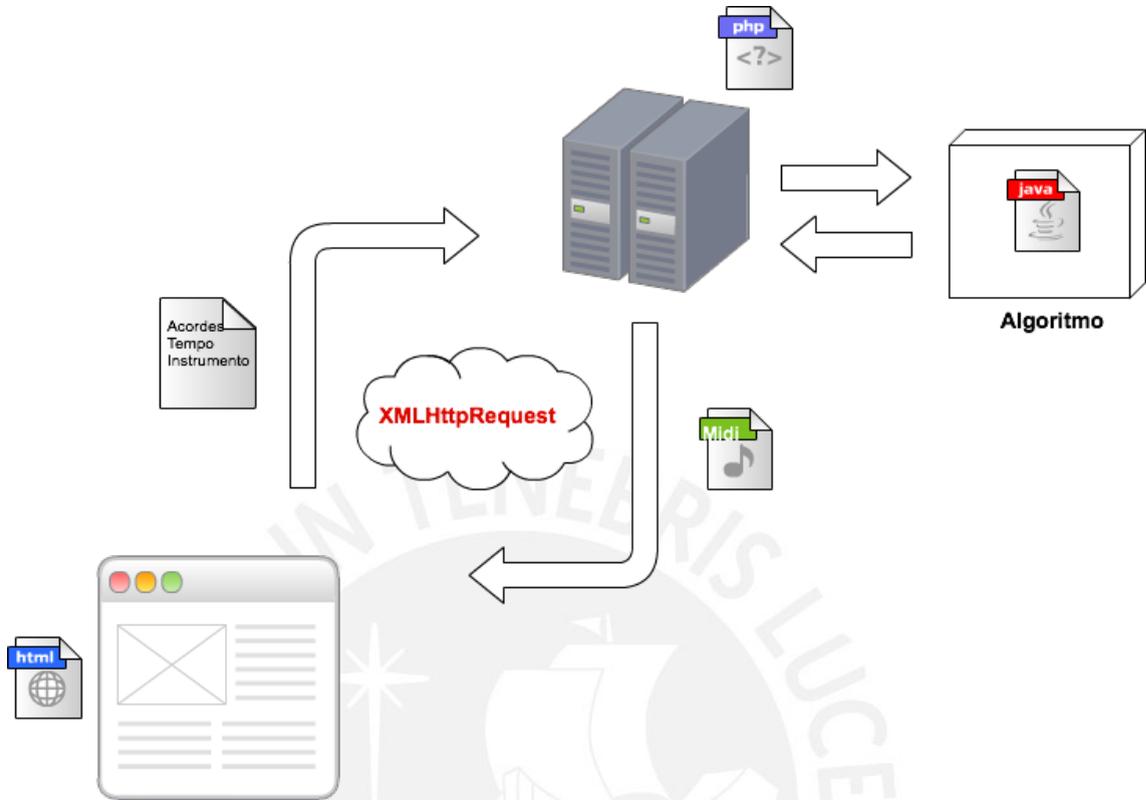


FIGURA 5.6: Esquema de trabajo de los datos en cliente y en el servidor

La representación interna del proceso sería como se muestra en la siguiente imagen.

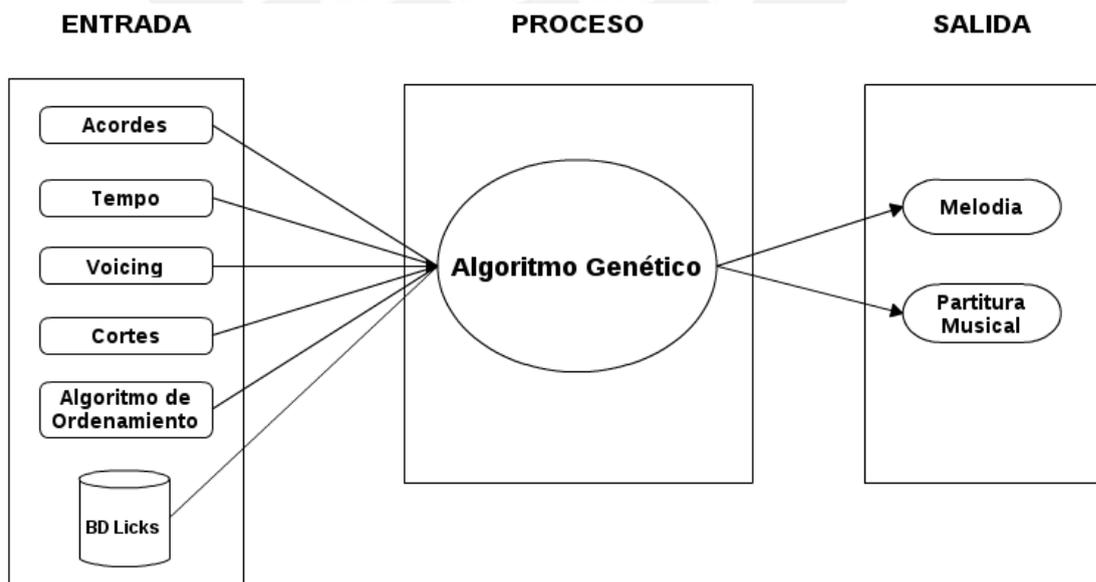


FIGURA 5.7: Representación interna del proceso

## 5.7. Verificación de los resultados

El proceso de validación de una pieza musical es algo complejo. Durante este proyecto, se intentó validar una frase musical con un buen sonido como aquella cuya distancia horizontal es mínima, tal como se explicó en capítulos anteriores. Sin embargo, esto puede resultar agradables para unos, como desagradables para otros. Por lo tanto, la validación de una composición es algo subjetivo y la única forma de poder decir si una composición es buena o mala es escuchándola. [30] [25] [8]. Debido a que ello es soportado por todas las fuentes consultadas, se realizó la validación de resultados mediante dicho método (validación con expertos), o en simples palabras, escuchando. No se han encontrado fuentes sobre métodos para la validación automática de una composición musical, de tal forma que decida si fue buena o mala. Por otro lado, es posible adaptarse ideas de trabajos realizados en otras áreas. Por ejemplo, en el trabajo de Ahmed Elgammal y Babak Saleh en su paper "Quantifying Creativity in Art Networks"[31], se realiza una calificación automática de pinturas mediante su comparación con una base de datos de obras de arte de conocidos expertos en el campo. Es posible adaptar dicho método al campo musical y realizar la comparación de las melodías generadas contra una base de datos con improvisaciones de expertos. Sin embargo, no se logró encontrar más fuentes en el tema, por lo cual se decidió realizar la validación de la melodía de forma manual mediante el análisis musical de igual manera como lo realizan nuestras fuentes.

Se realizó la consulta a un experto para poder realizar el análisis de la improvisación mostrada anteriormente. Para ello, se procedió a colorear las notas de la composición, dividiendo las notas pertenecientes al acorde (notas estables) con color azul, las notas pertenecientes a la escala con color verde, las notas que no pertenecen a la teoría musical correcta para el acorde con el color rojo y los silencios con color amarillo. El resultado se muestra en la siguiente imagen.

# Alice in wonderland

Solo

Euricide.java

Dm7 G7 CMaj7 FMaj7 Bm7b5

E7 Am7 Eb7 Dm7 G7

Em7 Am7 Dm7 G7 Em7

Dm7 Dm7 G7 CMaj7 FMaj7

Bm7b5 E7 Am7 Eb7 Dm7

G7 Em7 Am7 Dm7 G7 CMaj7

G7 CMaj7

FIGURA 5.8: Improvisación analizada

Finalmente, se consideraron distintos factores musicales para determinar la validez de la composición generada por la herramienta. Dichos factores y sus respectivas observaciones se aprecian en el análisis realizado al final del proyecto.

## **5.8. Discusión**

### **5.8.1. Interfaz para el ingreso de armonía**

Lo explicado servirá para como nexo de comunicación entre la página web y el algoritmo. Este nexo brindará facilidad de uso a los usuarios finales para poder probar combinaciones de acordes que ellos consideren experimentar para probar como es que se pueda improvisar sobre estos acordes.

### **5.8.2. Intefaz para la reproducción de la música generada**

Para la realización de este objetivo, se tuvo que recurrir al uso de una librería de un tercer usuario y adaptarla de mejor forma al objetivo. Posteriormente, tuvo que modificarse las API's para que puedan adaptarse al proyecto. Finamente, para el proceso de validación se tuvo que realizar la consulta a un experto en el tema y se comprobó la validez teórica y sonora de la composición.

## Capítulo 6

# Conclusiones y trabajos futuros

### 6.1. Introducción

En este capítulo se presentan las observaciones y conclusiones obtenidas en el proyecto. Además, se presentan las recomendaciones pertinentes para trabajos futuros.

### 6.2. Observaciones

El algoritmo desarrollado en el presente proyecto ha demostrado su fidelidad a la teoría musical tal como algunos de los algoritmos presentados en el estado del arte, con la diferencia de que ha sido desarrollado con la intención de ser una herramienta de libre uso. Además, permite ser usado para cualquier secuencia armónica que se solicite.

En primer lugar, se analizaron los algoritmos más utilizados y de propósito general y se detectaron sus debilidades en el ámbito de la teoría musical, que principalmente eran la generación de una composición sin una idea musical concreta, la cual no consideraba los patrones musicales conocidos. Esto permitió realizar el nuevo algoritmo de manera que intente mejorar este problema, por lo cual se decidió utilizar algoritmos genéticos con operadores genéticos basados exclusivamente en teoría musical, lo cual presentó un buen rendimiento.

Para el desarrollo del algoritmo se utilizaron las ideas extraídas principalmente de los trabajos de AI Biles y Dragan Mátic, lo cual permitió una adecuada estructuración y facilitó el cumplimiento de los resultados durante esta etapa.

### 6.3. Resultados

Se logró obtener los siguientes resultados a partir del proyecto de tesis.

#### 6.3.1. Workshops

Se presentó un artículo basado en el proyecto de tesis al "XXVII Workshop of Undergraduate Works - SIBRAPI 2014" (<http://emap.fgv.br/sibgrapi-2014/call-for-WUW.html>). Además, el paper del proyecto fue aceptado en el "I Workshop on Pattern Recognition and Applied Artificial Intelligence - WRPIAA 2014" (<http://grpiaa.inf.pucp.edu.pe/wrpiaa2014/>). Finalmente, este año el paper fue enviado al "II Latin American Congress on Computational Intelligence 2015" en Curitiba, Brasil, en donde también fue aceptado y publicado. (<http://la-cci.org/>).

# Artificial musical pattern generation with genetic algorithms

Francisco Vilchez Vargas, Jose Astuvilca Fuster and César Beltrán Castañón  
Pontificia Universidad Católica del Perú  
Sección Ingeniería Informática, Dpto. de Ingeniería  
Grupo de Reconocimiento de Patrones e Inteligencia Artificial Aplicada  
Av. Universitaria 1801, San Miguel, Lima, Perú  
{vilchez.francisco, j.astuvilca, cbeltran}@pucp.pe

**Abstract**—This paper describes an implementation for generating artificial musical patterns using genetic algorithms. This algorithm widely considers the musical theory and the use of smart operators for creating new melodies. We focused on creating jazz music and play it with a rhythmic accompaniment, so that we can get educational uses from it.

**Keywords**—Algorithm Composing; Combinatorial optimization; Genetic Algorithms.

## I. INTRODUCTION

The musical composition is the art of sorting sounds in order to create a certain emotion in the listener [1]. With the development of computer technology, many tools have been developed in order to provide the ability to touch and manipulate a lot of sounds, for example, electronic instruments with pre-programmed music [2]. In the last three decades people attempted to find algorithms that can create music from a variety of methods, including: randomized algorithms, searching patterns and systems based on previous knowledge [3] [4]. A particular case is to compose a melody from a previous given harmonic sequence [5], similar to what is done by the musicians when they improvise. In other words, when there is a predefined sequence of chords and the melody created must adapt to it following theoretical rules of modern music.

This research aims to implement a tool that can make a proper combination of musical notes and identification of patterns for the creation of melodies. Also, it will have a broad musical approach, which will have many educational advantages for the user. The search for these tunes becomes a problem of combinatorial analysis which depends on the number of chords. There are several methods of combinatorial optimization, but our proposal is to apply genetic algorithms to approximate the search for these melodies.

## II. TECHNICAL BACKGROUND

In this section, we detail the technique we used for the music generation.

### A. Important concept

Genetic algorithms are a class of heuristic which aims at finding a solution or optimization of an determined function. This kind of algorithms has a structure that simulates the

Darwinian Theory of Evolution. A population in a constant struggle for survival, in which only the fittest succeed. In this theory, the ability of a individual was defined by his abilities to adapt to the environment. [6]

Making a connection with the computational problem, the proposal of John Holland (1960) was to simulate this process with a heuristic called genetic algorithms, which he noted in Darwin's theory, where he tried to optimize the capabilities of individuals to improve their ability to survive and finally to find the optimal solution for the problem.

The general code for a genetic algorithm is as follows [5]:

```
Initialize the initial population
Until stopping condition is met
  Fitness of each individual is obtained
  We select parents to be married
  Selected parents marry
  The new population is constructed
End While
```

## III. NEW TECHNIQUE OR TECHNIQUE ADAPTATION

Our technique aims at obtaining a music that fits on any armony given. In this section, we detail the way we applied genetic algorithms to the problem of music generation.

### A. Formulation

The use of algorithms for musical composition is based on the continous effort of improving the melodies that are created, i.e, a composer creates a melody that came to his mind, but this one is not static, he constantly improves it following certain patterns that musicians know. These patterns in most cases are known by the experience and listening how good is the end result of the tune. In music the "licks" are musical phrase of 4 measures which is a small musical serie that sound good to the human ear, and had been validated by professional musicians. These phrases are constantly used by composers to "ensure" that their tunes will sound good. [7]

Everything explained above, is a process that an ordinary person needs to experience and that when we tried to modeled it, became a combinatorial optimization problem that seeks to minimize the horizontal distance of notes and find new phrases based on previous "licks" to finally create a original melody.

We use genetic algorithms because of the next reasons: Optimization and creation of new licks, which are generated by the crossover and mutation operators.

**B. Initialization and tuning**

1) *Licks Database*: In order to make the general population of our algorithm, we will proceed to use a number of musical phrases. However, performing an initialization with randomly created phrases does not guarantee that our creation will have a musical criteria according to the composition. For this, we proceeded to use phrases validated by experts to initialize our population (licks). Those phrases had been stored in a database on a special coding that allows them to adapt to any composition. In this way, we guarantee that the phrases created from the algorithm will retain some musical criteria, and even upgrades. [8]

For storage, we must consider each respective key and all its possible alterations. We considered the most possible alterations. For example, for the key of C (Do), we considered add the major 7th (maj7) which corresponds to a major scale (Major avoid 4th) [1]. Thus, the mapping scale will be as follows:

Cmaj7 - Major 4th - C D E G A B

Subsequently, we repeated this procedure for all existing tones, without forgetting any.

Here's an example of the result obtained for the key of C with their alterations considered.

chord	scale	notasTeóricas
Cmaj7	Major (avoid 4th)	C D E G A B
C7	Mixolydian (avoid 4th)	C D E G A Bb
Cm7	Minor (avoid 6th)	C D Eb F G Bb
Cm7b5	Locrian (avoid 2nd)	C Eb F Gb Ab Bb
Cdim	W/H	C D Eb F Gb G# A B
C+	Lydian Augmented	C D E F# G# A B
C7+	Whole Tone	C D E F# G# Bb
C7#11	Lydian Dominant	C D E F# G A Bb
C7alt	Altered Scale	C Db D# E Gb G# Bb
C7#9	Mixolydian #2 (avoid 4th)	C Eb E G A Bb
C7b9	Harm Minor V (avoid 6th)	C Db E F G Bb
CmMaj7	Melodic Minor	C D Eb F G A B
Cm6	Dorian (avoid 7th)	C D Eb F G A
Cm7b9	Melodic Minor II mode	C Db Eb F G A Bb
Cmaj7#11	Lydian	C D E F# G A B
C7sus	Mixolydian	C D E F G A Bb
Cmaj7sus	Major	C D E F G A B
C7bl	Blues	C Eb F Gb G Bb

Fig. 1. Chords vs. Scale mapping [3]

2) *Improved chromosome representation*: In order to can storage the phrases in the database, to apply the genetic operators and the final sound conversion, we have represented our musical phrases with numbers. This allows us to perform genetic operations for any phrase, regardless of his tone, and also allow to use any generated phrase for any composition requested by the user, whatever the tone is.

Here, we present as an example, the normal structure of a musical phrase validated by an expert and the method of conversion for further storage in the database:



Fig. 2. Lick validated [9]

The phrase is composed by four measures. Each measure has 4 times, that means, that is conformed by 4 quarters (note with a duration of 1 time). We will do the assignation of numbers for each sixteenth note, that means that for each quarter, we will have 4 numbers to represent it. In order to make the assignment of numbers, we must consider the corresponding chord for each Measure. For example, in the first Measure, has assigned a Cmaj7 chord, which, according to the musical theory, which is stored for disposal of our algorithm, would correspond a major scale (Major avoid 4th). Therefore, our available notes will be the next ones:

C - D - E - G - A - B - C - D - E - G - A - B - C - D  
 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14

Fig. 3. Notes and respective numbers

We proceed to add a number to each note, starting with the C4. Because of the reason that we took the sixteenth as our unit, we will have 16 numbers per measure. For representing silence, we used the 0. For a "hold" we used 15. Finally for starting a triplet, we used 16.

Some of the sentences doesnt obey all the theoretical rules that we have established, therefore, we proceeded to make approximations to those notes . For this, we made a replacement of those notes for others that accomplishes their similar tonal function.

Once we realized the assignation of numbers for each measure, we proceed to store each individual and assign them an id, then store the Phrase with his 4 respective ids of Measures that conforms it.

Finally, we proceeded to perform a classification of the phrases by their genre to which. This allows us to have a better population initialization.

An example of the final result would be:  
 Phrase with id 3:

id	measureid	genre
▶ 3	1 2 3 4	Jazz

Fig. 4. Phrase in database

His measures:

id	notas
1	0 0 0 0 0 9 15 10 15 9 15 8 15 7 15
2	8 15 15 15 15 15 8 15 9 15 9 15 7 15 6 15
3	7 15 15 15 15 15 7 15 9 15 7 15 6 15 5 15
4	4 15 5 15 3 15 3 15 4 15 3 15 3 15 4 15

Fig. 5. Measure in database

#### IV. IMPLEMENTATION

In order to get the best results for our method, we use the following implementation strategy.

##### A. Overview of the genetic algorithm

As we can see in the algorithm, the first step is the selection. To do this, first we proceeded to randomly choose 16 musical phrases database. After that, we will create 32 additional sentences from the 16 initially selected. This is done because the whole composition must have the same musical idea throughout its development. That is why these 32 new phrases have become "children" of the first 16. Finally, those will be part of the initial population.

##### B. Fitness selection

Once the population has been initialized for our genetic algorithm, we will run a tournament between pairs and select 48 phrases. The phrases will be selected with a probabilistic random, and repetitions will be allowed. This selected group will move to the crossover.

##### C. Genetic Operators

A phrase has a sequence of notes, which are responsible for expressing the melodic idea. These notes should maintain a minimum distance between them, so that large and unexpected intervals between them are avoided [10]. Most of the notes that have a very wide distance between them, which is known as horizontal interval, doesn't have a very nice sound for users, so, the compositions generated by the algorithm should try to achieve the lowest horizontal interval possible from note to note and also minimize the horizontal interval between each phrase. However, the phrases currently validated by experts, have in several cases horizontal intervals of high magnitude. Therefore, in the case of phrase level crossover, we will try to keep this distance to avoid losing the initial idea.

We have implemented the following genetic operators:

1) *Intelligent Crossover*: The use of a randomized crossover doesn't guarantee us valid solutions for our composition, because the horizontal interval is not necessarily the minimum, therefore, we have proceeded to use a intelligent crossover, which always generate valid solutions. Therefore, this specialized crossover will allow us to make improvements to our initial population. The intelligent crossover pseudocode being used is the next one [11]:

In each of the three bars of the phrase

Get the horizontal interval of the parents

Get the horizontal interval of children

Get the difference between the two intervals

Return the difference as the fitness for that point

Select the crossover point, which is the lowest fitness

Perform crossover

2) *Mutation*: Our initial population consists of initial phrases validated by experts (licks). In order to make our algorithm create different compositions each time it runs, we need to make a diversity of our population, otherwise composition will sound repetitive or always based on the same phrases.

Genetic operators allow us to get the diversity that we want in our population. However, as with the crossover operators, the use of random operators could damage our phrases into an unpleasant sensation for the user. Due this reason, we have proceeded to implement intelligent mutation operators, which ensure us the consistency of each sentence. That way, we retain the pleasant sound and doesn't spoils the crossover work to minimize the horizontal interval. Some implemented mutation operators are:

- *Transpose 2 down operator*: Make a transposition of the entire phrase to 2 tone below. This operator is most commonly used to provide diversity in phrases without altering the main idea or get the chance of possible dissonances. [8]
- *Octave operator*: This operator plays a role in improvement and diversifying. Through it, we will change the initial phrase, however, we do it by looking for an improvement based on the same principle of trying to decrease the horizontal interval. To do this, the operator must locate notes that present a high horizontal interval, which may possibly be unpleasant to the user. Once located, we will proceed to perform a transposition of an octave, so that reduces the horizontal interval and tries to eliminate this interval that could be disadvantageous to our algorithm. [10]
- *Hemiola operator*: This operator is the one that offers the most diversification of phrases. With the hemiola, we first look for a certain group of notes. Once the group is located, we proceed to repeat it throughout all the phrase (4 bars). That way, the group of notes with a pleasant sound is repeated throughout the phrase, which provides a sense of rhythm disturbance of the composition. It is a technique that is widely used by musicians during improvisation and we proceeded to teach it to the algorithm. [10]

#### V. RESULTS

To show our results, we proceed to create a beautiful tool that shows the users the specific notes they will need to play. It is shown in a piano, so it will be easy to

understand for beginner users. It also gives a .midi file, so the users can see the sheet of the composition that was created.

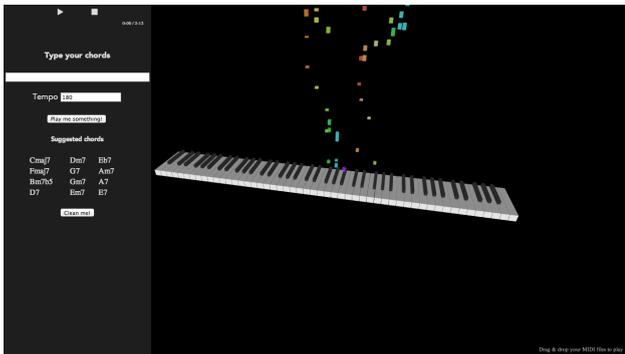


Fig. 6. Tool christened "Euricide"

Our results are available in this website:

<http://grpiaa.inf.pucp.edu.pe/euricide/>

## VI. CONCLUSION

In this paper, we introduced this technique based on previous works and we tried to add more music theory to improve results. We adapted our genetic operators to improve our licks database. The intelligent crossover allowed us to improve and mix the different licks. The mutations operators improved our phrases and gave diversity to the composition. As a result, we obtained great melodies displayed in a beautiful interface which will be very useful for beginners in music improvisation.

## ACKNOWLEDGMENT

The authors would like to thanks to Francisco's grandparents Miguel and Tana, and his mom Elvirita. Also, special thanks to Jose's family. Thanks to Phd Al Biles, for his recommendations.

## REFERENCES

- [1] E. Herrera, *Teoría musical y armonía moderna: Vol. I.* Antoni Bosch editor, 1990.
- [2] A. Johnston and E. Edmonds, "Creativity, music and computers: Guidelines for computer-based instrumental music support tools," 2004.
- [3] J. A. Biles, "Straight-ahead jazz with genjam: A quick demonstration," in *MUME 2013 Workshop*, 2013.
- [4] K. Verbeurgt, M. Dinolfo, and M. Fayer, "Extracting patterns in music for composition via markov chains," in *Proceedings of the 17th international conference on Innovations in applied artificial intelligence*, ser. IEA/AIE'2004. Springer Springer Verlag Inc, 2004, pp. 1123–1132. [Online]. Available: <http://dx.doi.org/10.1007/b97304>
- [5] J. Biles, "Genjam: A genetic algorithm for generating jazz solos," 1994, pp. 131–137.
- [6] J. H. Holland, *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press, 1975.
- [7] F. Pease and R. Mattingly, *Jazz composition: theory and practice.* Berklee Press, 2003.
- [8] E. R. Miranda and J. Al Biles, *Evolutionary computer music.* Springer, 2007.
- [9] J. Shneidman, "1001 jazz licks," 2000.
- [10] D. Matic, "A genetic algorithm for composing music," *The Yugoslav Journal of Operations Research ISSN: 0354-0243 EISSN: 2334-6043*, vol. 20, no. 1, 2013.
- [11] J. A. Biles, "Genjam in transition: From genetic jammer to generative jammer," in *Generative Art*, vol. 2002, 2002.

### 6.3.2. Composiciones disponibles

A continuación, se muestra la partitura de uno de los midis generados por nuestra herramienta. La generación de la improvisación se realizó sobre la secuencia armónica de la composición "Alice in Wonderland" de Sammy Fain. La partitura fue generada utilizando la herramienta "GarageBand" de Apple, lo cual demuestra la compatibilidad de las composiciones generadas con herramientas musicales profesionales de terceros.

Los resultados están disponibles en el siguiente link:

<https://soundcloud.com/euricide>



## Alice in Wonderland (Solo)

Jazz Swing

Euricide.java

♩ = 180

D minor (avoid 6th)    G mixolydian (avoid 4th)    C Major (avoid 4th)    F major (avoid 4th)

B locrian (avoid 2nd)    E mixolydian (avoid 4th)    A minor (avoid 6th)    Eb mixolydian (avoid 4th)

D minor (avoid 6th)    G mixolydian (avoid 4th)    E minor (avoid 6th)    A minor (avoid 6th)

D minor (avoid 6th)    G mixolydian (avoid 4th)

E minor (avoid 6th)    D minor (avoid 6th)    C major (avoid 4th)

### 6.3.3. Análisis musical de "Alice in Wonderland"

A continuación, se muestra un cuadro con el análisis realizado por el experto en composición e improvisación musical Juan Carlos Pajuelo, graduado en "Berklee College of Music" sobre la composición "Alice in Wonderland".

#	Criterio	Valor	Porcentaje sobre el total	Observaciones
1	Cantidad de tiempos (negras) en total en la composición	132	100 %	Realiza una improvisación sobre 2 vueltas armónicas.
2	Cantidad de notas utilizadas en total	218	100 %	La improvisación realizada utiliza distintas figuras melódicas (treceillos, semicorcheas, corcheas, etc.), lo cual la hace más llamativa y agradable. El uso de la corchea de swing durante la reproducción de la improvisación hace resaltar el estilo de jazz en el solo, lo cual resulta favorable para la herramienta. La cantidad de notas es aproximadamente el doble de la cantidad de tiempos, lo cual demuestra el uso constante de corcheas, lo cual puede mejorarse tratándose/obligando a utilizar una mayor cantidad de figuras melódicas lo cual atrae más la atención del oyente.
3	Cantidad de notas pertenecientes al acorde	131	60 %	Es adecuado que gran parte de las notas sean pertenecientes al acorde, ya que esto le da estabilidad a la improvisación. Es por ello que es necesario que la improvisación resuelva constantemente en una nota del acorde. Por otro lado, el uso excesivo de esto puede darle una sensación de monotonía a la improvisación. Sin embargo, la improvisación analizada utiliza una cantidad aceptable de notas del acorde y no se siente una sobrecarga en el uso de éstas.

4	Cantidad de notas pertenecientes a la escala	87	40 %	El uso de las notas de la escala le da una cierta variedad a la improvisación, y le permite expresar una mayor cantidad de sentimientos, entre ellos, partes de tensión, alegría o tristeza. Estas notas son adecuadas de usar pero en momentos precisos, como en el pase de una acorde a otro. Durante la improvisación escuchada, el uso de este tipos de notas fue usado en gran parte en momentos adecuado. Sin embargo, algunas de las notas determinadas pueden no resultar las más apropiadas si es que se realiza un análisis musical mas profundo.
5	Cantidad de notas no pertenecientes ni a la escala ni al acorde	0	0 %	Ninguna de las notas utilizadas fueron externas a las determinandas como escala más adecuada. Sin embargo, algunas de las notas, a pesar que pertenecen a la escala determinada por la herramienta como más efectiva, podrían sonar inclusive mejor si se cambian por otras, las cuales son determinadas realizando un análisis musical más complejo y que la herramienta podría intentar incorporar.
6	Cantidad de escalas más apropiadas utilizadas	26	81 %	La improvisación realiza un uso correcto de las escalas, inclusive las que no son las más apropiadas suenan bien en la improvisación realizada, sin embargo, sonaría mejor si se pudiera realizar un análisis más musical durante la asignación de escalas ya que, un mismo acorde puede tener más de una escala correcta. Entonces, durante un patrón musical de acordes, uno de esos acordes puede tener asignada una escala y para otro patrón tener asignada otra escala diferente. Esto permitiría resaltar mucho más la intención de cada acorde (tensión, tristeza, alegría), sin embargo, las escalas utilizadas son aceptables.

7	Cantidad de tiempos de silencio	10.5	8 %	Realizar breves descansos durante la improvisación es uno de los criterios más importantes, ya que permite poder dividir nuestras frases y dar un mejor orden a las ideas expresadas. La cantidad de silencio utilizadas en la improvisación analizada es muy baja, y puede aturdir por momento al oyente. Es recomendable intentar mejorar dicho aspecto en la improvisación.
8	Resoluciones en notas del acorde	17	53 %	El sumamente importante que el uso de las notas del acorde sea principalmente al inicio de cada nuevo acorde. Esto le brinda la sensación de "tranquilidad" a la improvisación, ya que se está resolviendo la frase en una nota estable. Para el alto número de notas del acorde utilizadas, se esperaba una mayor cantidad de resoluciones, sin embargo, esto no está reflejado en este indicador. Por otro lado, el jazz es característico por realizar resoluciones solamente en los momentos precisos y no constantemente, por lo que dicho método propuesto por la herramienta puede ser aceptable. Sin embargo, es recomendable tener en cuenta esta sugerencia.
9	Resoluciones en notas de la escala	16	46 %	Muchas de las resoluciones a notas de la escala se realizaron en tiempos fuertes, en los cuales es casi indispensable resolver en una nota del acorde, como por ejemplo, en el 5to compás. Al igual que el indicador anterior, se recomienda tener en consideración este punto, lo cual no quiere decir que debamos abusar de las continuas resoluciones en notas del acorde, ya que la música se caracteriza por el constante cambio de sensaciones y de acciones inesperadas.

CUADRO 6.1: Análisis realizado por Juan C. Pajuelo - Berklee College of Music, Boston

## 6.4. Conclusiones

Las conclusiones a las que se llegaron al finalizar este trabajo de fin de carrera son las siguientes:

- Se realizó la investigación de frases musicales validadas, cuya estructura musical se adapte a la planteada en el algoritmo. Posteriormente, se almacenaron las frases musicales en un repositorio.
- Se realizó el almacenamiento de la teoría musical, basándonos en libros de armonía musical moderna, lo cual permitió que las creaciones realizadas en el algoritmo no generaran disonancias desagradables. De esta manera, el algoritmo no puede componer algo fuera de la teoría musical.
- Se realizó la creación de una interfaz, la cual permitía a los usuarios ingresar las secuencias armónicas sobre la cual deseaban se realice la composición.
- Se realizó la implementación del algoritmo, para ello, se basó en trabajos previos, principalmente de Al Biles y Dragan Matic. Además, se realizó la creación de operadores genéticos inteligentes, los cuales mejoraron la población inicial y a partir de ellas se creaban nuevas composiciones.
- Se realizó la creación de una interfaz que permitiera mostrar la composición realizada para un usuario sin mucho conocimiento musical. Además, se realizó la creación de un archivo .midi sobre la composición generada.

## 6.5. Recomendaciones y trabajos futuros

Debido a que no se ha puesto mucho énfasis en el análisis melódico de la improvisación, se podría incluir dichos temas en la teoría musical considerada para que puedan ser tomados en cuenta durante la improvisación.

Por otro lado, las asignaciones de escala pueden ser mejoradas si realizamos un análisis armónico más amplio, en el cual se haga una búsqueda de patrones musicales que

se han sido utilizado para dicha armonía, y determinar una escala aún mejor que se pueda adaptar a ello.

Además, puede darse una enfoque aún más educativo realizando una explicación sobre la forma en que se determinó la escala más adecuada y el uso de notas elegidas. También puede darse sugerencias de patrones musicales que puedan ser utilizados en la composición y una mayor información sobre las frases musicales en las que se basó el algoritmo para la composición.

Finalmente, se puede realizar mejoras en el almacenamiento de las frases y en la lectura de la armonía, de tal forma que permita el uso de más figuras musicales y más tipos de compases.



## Bibliografía

- [1] Enric Herrera. *Teoría musical y armonía moderna: Vol. I*. Antoni Bosch editor, 1990.
- [2] Andrew Johnston and Ernest Edmonds. Creativity, music and computers: Guidelines for computer-based instrumental music support tools. 2004.
- [3] John Biles. Genjam: A genetic algorithm for generating jazz solos. pages 131–137, 1994.
- [4] Karsten Verbeurgt, Michael Dinolfo, and Mikhail Fayer. Extracting patterns in music for composition via markov chains. In *Proceedings of the 17th international conference on Innovations in applied artificial intelligence, IEA/AIE'2004*, pages 1123–1132. Springer Springer Verlag Inc, 2004. ISBN 3-540-22007-0. doi: 10.1007/b97304. URL <http://dx.doi.org/10.1007/b97304>.
- [5] Man-Kwan Shan and Shih-Chuan Chiu. Algorithmic compositions based on discovered musical patterns. *Multimedia Tools Appl.*, 46(1):1–23, January 2010. ISSN 1380-7501. doi: 10.1007/s11042-009-0303-y. URL <http://dx.doi.org/10.1007/s11042-009-0303-y>.
- [6] William Eign John A. Biles. Genjam populi: Training an iga via audience-mediated performance. In *Proceedings of the 1995 International Computer Music Conference, ICMA*, 1995.
- [7] John A Biles. Straight-ahead jazz with genjam: A quick demonstration. In *MUME 2013elgammalquantifying Workshop*, 2013.

- [8] Dragan Matic. A genetic algorithm for composing music. *The Yugoslav Journal of Operations Research ISSN: 0354-0243 EISSN: 2334-6043*, 20(1), 2013.
- [9] Ian Hickson and David Hyatt. *HTML5: A vocabulary and associated APIs for HTML and XHTML. W3C Working Draft edition*, 2011.
- [10] Xueqiao Xu. Euphony. <https://github.com/qiao/euphony/>, 2012.
- [11] Andrew R. Brown. *Making music with Java : an introduction to computer music, Java programming, and the jMusic library*. Lulu, Raleigh, North Carolina, 2005. URL <http://eprints.qut.edu.au/20320/>.
- [12] James Gosling. *The Java language specification*. Addison-Wesley Professional, 2000.
- [13] AB MySQL. *MySQL reference manual*. 2001.
- [14] Maxwell Shinn. *Instant MuseScore*. Packt Publishing Ltd, 2013.
- [15] Jack Shneidman. *1001 jazz licks*, 2000.
- [16] Real Academia Española. *Real Academia Española*. Real Academia Española, 1991.
- [17] Frederick Pease and Rick Mattingly. *Jazz composition: theory and practice*. Berkeley Press, 2003.
- [18] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 3 edition, 2009.
- [19] Zong Woo Geem and Jeong-Yoon Choi. Music composition using harmony search algorithm. In *Proceedings of the 2007 EvoWorkshops 2007 on EvoCoMnet, EvoFIN, EvoIASP, EvoINTERACTION, EvoMUSART, EvoSTOC and EvoTransLog: Applications of Evolutionary Computing*, pages 593–600, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-71804-8. doi: 10.1007/978-3-540-71805-5\_65. URL [http://dx.doi.org/10.1007/978-3-540-71805-5\\_65](http://dx.doi.org/10.1007/978-3-540-71805-5_65).
- [20] Nigel Gwee. *Complexity and Heuristics in Rule-based Algorithmic Music Composition*. PhD thesis, 2002. AAI3069711.

- [21] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.
- [22] John H Holland. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press, 1975.
- [23] Charles Grinstead and Laurie Snell. *Introduction to Probability*. American Mathematical Society, 2003.
- [24] Ning-Han Liu, Yi-Hung Wu, and Arbee L. P. Chen. Identifying prototypical melodies by extracting approximate repeating pattern from music works. *J. Inf. Sci. Eng.*, 26(4):1181–1198, 2010. URL <http://dblp.uni-trier.de/db/journals/jise/jise26.html#LiuWC10>.
- [25] John A Biles. Performing with technology: Lessons learned from the genjam project. In *MUME 2013 Workshop*, 2013.
- [26] John A Biles. Genjam in transition: From genetic jammer to generative jammer. In *Generative Art*, volume 2002, 2002.
- [27] Christopher Raphael. Christopher raphael's music plus one. [http://www.music.informatics.indiana.edu/~craphael/music\\_plus\\_one/](http://www.music.informatics.indiana.edu/~craphael/music_plus_one/), 2010.
- [28] Peter M. Todd and Gregory M. Werner. Musical networks. chapter Frankensteinian methods for evolutionary music composition, pages 313–339. MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-07181-9. URL <http://dl.acm.org/citation.cfm?id=346573.346629>.
- [29] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997. ISBN 0070428077, 9780070428072.
- [30] Eduardo R Miranda and John A Biles. *Evolutionary computer music*. Springer, 2007.
- [31] Ahmed Elgammal and Babak Saleh. Quantifying creativity in art networks.